

الجمهورية الجزائرية الديمقراطية الشعبية

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTRE DE L'ENSEIGNEMENT
SUPERIEUR ET DE LA RECHERCHE
SCIENTIFIQUE

UNIVERSITE 8 MAI 1945
GUELMA

Faculté de Mathématiques,
D'Informatique et de Sciences de la Matière
Département : D'Informatique



وزارة التعليم العالي
و البحث العلمي
جامعة 8 ماي 1945
قلمة

كلية الرياضيات، الإعلام الآلي
و علوم المادة
قسم: الإعلام الآلي

Développement des Applications Web

Cours avec exemples et exercices corrigés

2^{ème} Année Licence Académique



Par : Dr. Khaled HALIMI

* Année universitaire : 2022/2023 *

SYLLABUS (Plan de cours)

Unité d'Enseignement : UEF1 Matière : Développement d'applications Web

Domaine/Filière: Informatique, Licence 2

Semestre : 4, **Année Universitaire :** 2022/2023

Crédits : 4, **Coefficient :** 2

Volume Horaire Hebdomadaire Total : 3h00

- Cours Magistral (*Nombre d'heures par semaine*) : 1h30
- Travaux Pratiques (*Nombre d'heures par semaine*) : 1H30

Enseignant responsable : Dr. Khaled HALIMI, **Grade :** Maitre de Conférences -A.

Email : halimi.khaled@univ-guelma.dz

Objectifs de l'enseignement de la matière :

Présenter les systèmes d'information dans le contexte Internet. Le module initie à la programmation Web via les langages HTML, JavaScript et PHP. En plus, il initie au développement des services Web. Une étude pratique renforce les concepts acquis.

Programme de cours:

Chapitre 1: introduction au World Wide Web

1. Définition et historique
2. Architecture Client/Serveur
3. Protocole http
4. notions de base de Web 2.0 (X.0)

Chapitre 2 : Langages de programmation pour le Web

1. Généralités : page statique, page dynamique et applications Web
2. Langages de balise : définition et historique
3. HTML
4. CSS
4. XML

Chapitre 3. Langage de programmation coté serveur (PHP)

1. Introduction
2. Syntaxe de base
3. Types, variables et opérateurs
4. Structures de contrôles
5. Classes et objets

Chapitre 4. Services Web : notions de base

1. Introduction
2. Architecture orientée services (SOA)
3. Caractéristiques des services Web

4. Standards de base pour les services Web

5. Plateformes de développement des services Web

Chapitre 5. Etude de cas : développement d'une application Web sous forme de service Web (coté fournisseur et puis coté consommateur)

Evaluation : Contrôles des connaissances & Pondérations

| Contrôle | Pondération (%) |
|---|-----------------|
| Examen final | 60 % |
| Travaux Dirigés (Présence & Participation) | |
| Travaux Pratiques | 40% |
| Micro-Interrogations | |
| Autres | |
| Total | 100% |

Références bibliographiques :

La suite des ouvrages disponibles dans la bibliothèque centrale de l'université 8 Mai 1945 Guelma :

| Code | Titre | Auteur | Maison d'édition |
|-----------|---|-------------------------------------|---|
| 02/73725 | Le World -Wide Web: Création de serveurs sur Internet. | <u>Claude Bajard</u> <u>Jean</u> | <u>Addison Wesley</u> |
| 03/80623 | Web design: Améliorez vos sites par l'exemple. | <u>يونس ديفيد</u> | <u>Osman Eyrolles</u> <u>Multimedia.</u> |
| 03/80429 | HTML | <u>Scambray Joel</u> | <u>CookBook.</u> |
| 15/266539 | HTML & CSS : design et création Web | / | <u>دار الراية للنشر والتوزيع</u> |
| 15/266546 | HTML & Javascript pour les nuls | / | <u>دار الراية للنشر والتوزيع</u> |
| 10/159682 | Html le guide complet | <u>Lemainque</u> <u>Fabrice</u> | <u>Mico Application.</u> |
| 15/266560 | HTML 5 par la pratique | / | <u>دار الراية للنشر والتوزيع</u> |
| 03/80631 | XML | <u>Ayers Frank</u> | <u>Ediscience</u> |
| 09/149309 | XML cours et exercices | <u>Brillant</u> <u>Alexandre</u> | <u>Osman Eyrolles</u> <u>Multimedia.</u> |
| 12/195186 | XML : synthèse de cours et exercices corrigés | <u>محمد صاحب سلطان</u> | <u>دار الراية للنشر والتوزيع</u> |
| 02/73712 | PHP | <u>Wielsch Michael</u> | <u>Mico Application.</u> |
| 10/161349 | PHP & My SQL et CSS | <u>Carfantan Jean</u> | <u>Mico Application.</u> |
| 12/195064 | Apprendre à développer un site Web avec PHP et MySQL : exercices pratiques et corrigés | <u>محمد صاحب سلطان</u> | <u>دار الراية للنشر والتوزيع</u> |
| 13/200747 | Les Web services : concevoir et utiliser des applications 2.0 C# | / | <u>API : PHP</u> |
| 07/115389 | Les Web Services : Techniques, Démarches Et Outils XML, WSDL, SOAP, UDDI, Rosettanet, UML | <u>Kadima Hubert</u> | <u>Paris [France] : Dunod</u> |

Avant-Propos

L'objectif de ce cours est de vous amener très rapidement à la capacité de programmer des pages Web dynamiques. Le cours se compose d'une partie théorique, qui présente les principes généraux du Web, et d'une série de exemples pratiques progressifs, qui permettent de découvrir pas à pas les différentes composantes du Web dynamique. Il est important de comprendre les principes généraux du Web avant de commencer à programmer des pages Web dynamiques. En effet, cela vous permettra de comprendre le fonctionnement des pages Web et les différents outils utilisés pour créer des pages Web dynamiques.

Les pages Web dynamiques permettent d'afficher du contenu qui change en fonction des actions de l'utilisateur ou de données provenant d'une source externe, telle qu'une base de données ou un API. Pour créer des pages Web, vous devrez d'abord apprendre les langages de programmation côté client tels que le HTML, CSS et le JavaScript. Et pour commencer d'apprendre à programmer des pages Web dynamiques, ce cours vous propose un ensemble des concepts théoriques et pratiques concernent le XML, le PHP, la gestion des Bases de Données, et les Services Web. Il vous aide également à apprendre comment organiser les pages Web dynamiques simples pour vous familiariser avec les différentes composantes des applications Web.

Ce cours se trouve divisé en trois chapitres. Le premier chapitre sert comme introduction à au domaine du WWW. Il s'agit ici de faire apprendre aux étudiants les notions indispensables pour comprendre le contexte du Web. Nous présentons donc l'historique du Web, l'évolution du Web, le fonctionnement du Web, et les langages du Web.

Dans le deuxième chapitre, un aperçu sur le langage de balisage, le HTML, est bien détaillé. Puis nous présentons l'essentiel du langage XML.

Dans le troisième chapitre, nous abordons le langage CSS. Dans ce chapitre, nous introduisons les feuilles de styles, les règles CSS, les classes et les Div, et un aperçu sur les options de mise en page.

Dans le chapitre 04, nous présentons les concepts de programmation fondamentaux de PHP et acquérir les compétences nécessaires pour développer des programmes en PHP. Où on mettra l'accent sur la syntaxe de PHP, la gestion des bases de données, le langage SQL, la gestion des cookies et de sessions, etc.

Dans le chapitre 05, nous présentons les services Web, où l'objectif est d'enseigner aux étudiants les concepts fondamentaux des services Web, notamment les protocoles, les formats de données et les technologies utilisées pour créer des services Web.

Et on termine par le chapitre 6 qui présente un ensemble des exercices corrigés sur les différents concepts vus dans ce cours.

Pourquoi apprendre le Web ?

Il y a plusieurs objectifs pour lesquels il peut être bénéfique d'apprendre le cours du Web, notamment :

- Créer et gérer son propre site Web : Si vous avez une entreprise, un blog, ou tout simplement une passion que vous souhaitez partager avec le monde, apprendre le cours du Web vous permettra de créer et de gérer votre propre site Web.
- Travailler dans l'industrie du Web : Le Web est un secteur en constante croissance qui offre de nombreuses opportunités d'emploi pour les développeurs, les designers, les spécialistes du marketing numérique, et bien d'autres professions.
- Améliorer votre présence en ligne : Apprendre le cours du Web peut vous aider à améliorer votre présence en ligne en vous permettant de mieux comprendre comment fonctionnent les sites Web, les réseaux sociaux et les moteurs de recherche.
- Se tenir à jour avec les dernières tendances : Le cours du Web est un domaine en constante évolution et en apprenant constamment, vous pouvez rester à jour avec les dernières tendances et technologies, ce qui peut être bénéfique pour votre entreprise ou votre carrière.

Booster la carrière de l'étudiant

Le Web offre de nombreuses opportunités de carrière pour les étudiants, que ce soit dans le développement de sites Web, la conception, la gestion de contenu, la publicité en ligne, le marketing numérique et bien d'autres domaines.

En apprenant le cours du Web, les étudiants peuvent acquérir des compétences précieuses qui peuvent les aider à se démarquer sur le marché du travail. Par exemple, les compétences en développement Web, comme le HTML, le CSS et le JavaScript, sont très recherchées par les employeurs dans de nombreux secteurs. Les étudiants peuvent également utiliser leurs compétences en Web pour démarrer leur propre entreprise ou projet en ligne. De nombreuses startups ont commencé comme de simples sites Web et ont connu un grand succès grâce à une stratégie efficace de marketing numérique.

En outre, les étudiants qui apprennent le cours du Web peuvent également bénéficier d'un avantage concurrentiel sur le marché du travail. Les entreprises recherchent souvent des candidats ayant des compétences en développement Web, en conception de sites Web et en marketing numérique, car ces compétences sont essentielles pour toute entreprise ayant une présence en ligne.

En résumé, apprendre le cours du Web est très bénéfique pour la carrière des étudiants, que ce soit pour travailler dans l'industrie du Web ou pour améliorer leurs compétences en communication et en marketing numérique.

SOMMAIRE

| | |
|--|----|
| Sommaire | 6 |
| Introduction Générale..... | 12 |
| 1 Le World Wide Web..... | 13 |
| 1.1 Historique du Web | 13 |
| 1.2 Termes rattachés au Web | 14 |
| 1.3 Consultation et mesure..... | 16 |
| 1.4 Métiers du Web..... | 16 |
| 1.5 Évolution de l'usage | 17 |
| 1.6 Le Web 2.0..... | 17 |
| 1.7 Le Web 3.0..... | 18 |
| 1.8 L'Architecture du Web | 18 |
| 1.9 Exploration du Web et Web profond | 19 |
| 1.10 Document HTML..... | 19 |
| 1.11 Multimédia..... | 20 |
| 1.11.1 Images & videos | 20 |
| 1.11.2 Scripts et animations..... | 21 |
| 1.12 Décentralisation | 21 |
| 1.13 Comment fonctionnent les pages Web..... | 22 |
| 1.14 Noms de domaine et hébergement | 23 |
| 1.15 Le langage du Web | 24 |
| 2 Le HTML | 25 |
| 2.1 L'évolution du Web et des normes Web | 26 |
| 2.2 Qu'est-ce qu'une balise HTML ?..... | 27 |
| 2.3 Les balises élémentaires..... | 29 |
| 2.4 Les couleurs en html | 29 |
| 2.5 Les Titres | 30 |
| 2.6 Les listes en HTML | 31 |
| 2.6.1 Les attributs des listes..... | 33 |
| 2.7 Les tableaux | 33 |
| 2.7.1 colgroup | 34 |
| 2.7.2 Tableau à colonne de taille différente | 35 |
| 2.7.3 Couleur de fond et alignement du contenu de colonne..... | 35 |

| | | |
|-------|--|----|
| 2.8 | Les liens hypertextes..... | 35 |
| 2.9 | Le langage XML..... | 36 |
| 2.9.1 | Le XPath..... | 37 |
| 2.9.2 | Le XML Schema..... | 37 |
| 2.10 | Quizz..... | 38 |
| 3 | Le CSS..... | 40 |
| 3.1.1 | Stylisation d'un titre..... | 42 |
| 3.1.2 | Couleurs hexadécimales..... | 44 |
| 3.2 | Les règles CSS..... | 44 |
| 3.3 | Comprendre les classes CSS..... | 45 |
| 3.4 | Le span en CSS..... | 46 |
| 3.5 | Feuilles de style internes et externes..... | 48 |
| 3.5.1 | Création d'une feuille de style externe..... | 48 |
| 3.6 | Positionnement et Flux en CSS..... | 49 |
| 3.6.1 | Le Flux normal..... | 49 |
| 3.6.2 | Le Flux flottant..... | 50 |
| 3.7 | Un bref historique des techniques de mise en page sur le Web..... | 50 |
| 3.8 | Aperçu des options de mise en page..... | 53 |
| 3.9 | Comprendre les <div>..... | 54 |
| 3.10 | Quizz..... | 55 |
| 4 | Le PHP..... | 57 |
| 4.1 | Les sites Web dynamiques..... | 58 |
| 4.2 | Langage du Web dynamique..... | 58 |
| 4.3 | Comment fonctionne un site Web ?..... | 60 |
| 4.3.1 | Cas d'un site statique : Lorsque le site est statique, le schéma est très simple. Cela se passe en deux temps :..... | 61 |
| 4.3.2 | Cas d'un site dynamique :..... | 61 |
| 4.4 | éléments essentiels pour faire fonctionner un site Web PHP..... | 61 |
| 4.4.1 | Le serveur Web..... | 62 |
| 4.4.2 | Le langage PHP..... | 62 |
| 4.4.3 | Le système de gestion de base de données (SGBD)..... | 62 |
| 4.4.4 | Créer son premier script php..... | 64 |
| 4.5 | Premiers pas avec PHP..... | 64 |
| 4.6 | Les balises PHP..... | 64 |

| | | |
|--------|--|----|
| 4.7 | Syntaxe de base de php | 65 |
| 4.7.1 | Les variables en PHP | 66 |
| 4.7.2 | Les chaînes de caractères | 67 |
| 4.7.3 | Les entiers | 68 |
| 4.7.4 | Float | 69 |
| 4.7.5 | Boolean | 69 |
| 4.7.6 | Le type array | 70 |
| 4.7.7 | Le type object..... | 70 |
| 4.7.8 | Le type Null | 71 |
| 4.8 | Les opérateurs arithmétiques | 71 |
| 4.9 | Les opérateurs de comparaison | 72 |
| 4.10 | Les opérateurs logiques..... | 73 |
| 4.11 | Les classes..... | 73 |
| 4.12 | Structures de contrôle de flux | 74 |
| 4.12.1 | If-else..... | 74 |
| 4.12.2 | Switch-Case | 74 |
| 4.12.3 | While | 75 |
| 4.12.4 | For | 75 |
| 4.12.5 | Foreach..... | 75 |
| 4.12.6 | do-while | 75 |
| 4.12.7 | break | 76 |
| 4.12.8 | Continue..... | 76 |
| 4.13 | Les formulaires en PHP | 77 |
| 4.13.1 | Méthodes d'envoi Get et Post | 78 |
| 4.13.2 | Récupération des paramètres de formulaire en PHP | 79 |
| 4.14 | Les cookies..... | 79 |
| 4.14.1 | Principe | 80 |
| 4.15 | Les sessions..... | 81 |
| 4.16 | Interfaçage avec une base de données..... | 81 |
| 4.16.1 | Le langage sql | 81 |
| 4.16.2 | Les requêtes SQL..... | 83 |
| 4.17 | Les en-têtes HTTP | 84 |
| 4.18 | La bibliothèque GD..... | 85 |
| 4.19 | Configuration de Serveur XAMPP | 85 |

| | | |
|-------|--------------------------------------|-----|
| 5 | Les Services Web | 89 |
| 5.1 | But des services Web..... | 90 |
| 5.2 | L'architecture des services Web..... | 90 |
| 5.3 | Le protocole SOAP..... | 91 |
| 5.4 | Le protocole WSDL..... | 92 |
| 5.5 | Le Registre UDDI..... | 93 |
| 5.6 | Déployer Webservice en php..... | 94 |
| 6 | Exercices Corrigés..... | 96 |
| 6.1 | Exercices sur Le HTML..... | 96 |
| 6.1.1 | Exercice 1 | 96 |
| 6.1.2 | Exercice 2 | 97 |
| 6.1.3 | Exercice 3 | 98 |
| 6.1.4 | Exercice 4 | 100 |
| 6.2 | Exercices sur Le CSS..... | 100 |
| 6.2.1 | Exercice 1 | 100 |
| 6.2.2 | Exercice 2 | 101 |
| 6.2.3 | Exercice 3 | 102 |
| 6.3 | Exercices sur le PHP..... | 104 |
| 6.3.1 | Exercice 1 | 104 |
| 6.3.2 | Exercice 2 | 104 |
| 6.3.3 | Exercice 3 | 105 |
| 6.3.4 | Exercice 4 | 107 |
| 6.4 | Exercices sur les services Web..... | 109 |
| 6.4.1 | Exercice 1 | 109 |
| | Conclusion Générale | 111 |
| | Références Bibliographiques | 112 |

Table des figures

| | |
|--|----|
| Figure 1. Exemple d'une page HTML | 20 |
| Figure 2. Comment les fichiers se transmettent sur Internet..... | 23 |
| Figure 3. Code HRML pour afficher un titre..... | 25 |
| Figure 4. La partie entête de document HTML..... | 28 |
| Figure 5. La partie corps de document HTML..... | 28 |
| Figure 6. Les balises élémentaires de l'HTML | 29 |
| Figure 7. Exemple d'utilisation des couleurs avec le HTML..... | 30 |
| Figure 8. Les différents niveaux de la balise Heading | 31 |
| Figure 9. Une liste ordonnée..... | 32 |
| Figure 10. Liste non ordonnée | 32 |
| Figure 11. La liste de définitions | 32 |
| Figure 12. Liste ordonnée avec attribut..... | 33 |
| Figure 13. Création d'une table | 33 |
| Figure 14. La personnalisation de la table | 34 |
| Figure 15. L'utilisation de la balise COLGROUP | 34 |
| Figure 16. Tableau a colonne de taille différente | 35 |
| Figure 17. Personnalisation de la table..... | 35 |
| Figure 18. L'utilisation de la balise <a> | 36 |
| Figure 19. La balise <a> avec les attributs | 36 |
| Figure 20. Exemple d'un fichier XML | 36 |
| Figure 21. Exemple de code CSS pour changer la couleur de fond | 40 |
| Figure 22. Résultat de l'application d'un style..... | 41 |
| Figure 23. Une règle CSS | 42 |
| Figure 24. Explication de la règle CSS | 42 |
| Figure 25. Les métas donnés dans la partie entête | 43 |
| Figure 26. Regle CSS a l'interieur d'un fichier HTML | 44 |
| Figure 27. Sélecteur d'une règle CSS | 44 |
| Figure 28. Règle avec classe..... | 45 |
| Figure 29. Une classe CSS..... | 45 |
| Figure 30. Règle avec Span | 46 |
| Figure 31. Feuille de style externe | 48 |
| Figure 32. Comment lire le fichier CSS externe | 49 |
| Figure 33. Exemple d'un flux normal en CSS | 50 |
| Figure 34. Le flux flottant en CSS | 50 |
| Figure 35. tableau composé de trois lignes et de deux colonnes..... | 51 |

| | |
|--------------------|--------------------|
| Cellule fusionnée | |
| Ligne 2, Colonne 1 | Ligne 2, Colonne 2 |

Figure 36. Un tableau avec deux cellules fusionnées dans la première ligne. 51

Figure 37. Un modèle pour la mise en page basée sur des tableaux

| | |
|--|----|
| Figure 38. Code de la mise en page | 53 |
| Figure 39. Une mise en page flexible | 54 |
| Figure 40. Création des DIV | 55 |
| Figure 41. La demande d'une page statique..... | 61 |
| Figure 42. Demande d'une page dynamique..... | 61 |
| Figure 43. Script PHP a l'intérieur d'un document HTML..... | 65 |
| Figure 44. Un code PHP avec une syntaxe de base..... | 66 |
| Figure 45. Les différents types de variable | 67 |
| Figure 46. Utilisation de type Boolean..... | 70 |
| Figure 47. Utilisation de type object..... | 71 |
| Figure 48. Utilisation des opérateurs arithmétiques | 72 |
| Figure 49. Utilisation des opérateurs de comparaison..... | 73 |
| Figure 50. Exemple d'une classe en PHP..... | 74 |
| Figure 51. La syntaxe de IF-ELSE | 74 |
| Figure 52. La syntaxe de SWITCH-CASE | 75 |
| Figure 53. La syntaxe de la boucle WHILE..... | 75 |
| Figure 54. La syntaxe de la boucle FOR..... | 75 |
| Figure 55. La syntaxe de la boucle Foreach | 75 |
| Figure 56. La syntaxe de la boucle DO-WHILE | 76 |
| Figure 57. L'utilisation de la fonction BREAK | 76 |
| Figure 58. L'utilisation de la fonction CONTINUE | 76 |
| Figure 59. Exemple d'un formulaire | 77 |
| Figure 60. Récupération des données soumises par le formulaire | 77 |
| Figure 61. Redirection par la fonction HEADER | 78 |
| Figure 62. Utilisation des Sessions | 81 |
| Figure 63. Script de connexion avec la base de données | 83 |
| Figure 64. Téléchargement et installation de Serveur XAMPP | 86 |
| Figure 65. Lancement du Serveur XAMPP | 86 |
| Figure 66. La page d'accueil XAMPP | 87 |
| Figure 67. Le fichier php.ini..... | 87 |
| Figure 68. Lancement du PhpMyAdmin..... | 88 |
| Figure 69. Fonctionnement des services Web selon l'architecture SOA..... | 91 |
| Figure 70. Exemple d'un fichier SOAP..... | 91 |
| Figure 71. Exemple de fichier WSDL | 93 |

INTRODUCTION GENERALE

Le World Wide Web (WWW), littéralement la « toile (d'araignée) mondiale », communément appelé le Web, et parfois la Toile, est un système hypertexte public fonctionnant sur Internet. Le Web permet de consulter, avec un navigateur, des pages accessibles sur des sites. L'image de la toile d'araignée vient des hyperliens qui lient les pages Web entre elle.

Le Web n'est qu'une des applications d'Internet ; distincte d'autres applications comme le courrier électronique, la messagerie instantanée, et le partage de fichiers en pair à pair. Le Web a été inventé par Tim Berners-Lee et Robert Cailliau plusieurs années après Internet, mais c'est lui qui a rendu les médias grand public attentifs à Internet. Depuis, le Web est fréquemment confondu avec Internet; en particulier, le mot Toile est souvent utilisé dans les textes non techniques sans qu'il désigne clairement le Web ou Internet.

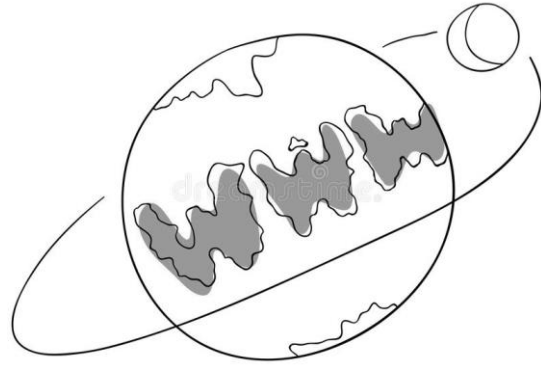
En inventant le Web, Tim Berners-Lee avait aussi pensé à d'autres noms, comme *Information Mesh* (maillage d'informations), *Mine of Information* ou encore *The Information Mine* (la mine d'informations, dont le sigle serait *Tim*). Le sigle WWW a été largement utilisé pour abrégé *World Wide Web* avant que l'abréviation Web prenne le pas. À l'écrit, les lettres *www* restent très utilisées dans les adresses Web et quelques autres conventions techniques. WWW est parfois abrégé en *W3*, abréviation qu'on retrouve dans le sigle *W3C* du *World Wide Web Consortium*. Dans la seconde moitié des années 1990, une blague répandue prétendait que WWW signifiait *World Wide Wait*, soit « attente mondiale », car le réseau Internet était engorgé par la popularité grandissante du Web.

Au fil des années, le Web a connu de nombreuses avancées, notamment dans le développement de sites Web, l'expérience utilisateur, la sécurité, l'optimisation pour les moteurs de recherche, les réseaux sociaux et la publicité en ligne. Les tendances actuelles incluent l'utilisation de l'intelligence artificielle, le développement de sites Web progressifs, l'optimisation mobile et la réalité virtuelle.

Le Web comprend plusieurs langages de programmation essentiels tels que HTML, CSS et PHP. Ces langages sont utilisés pour créer des sites Web interactifs et dynamiques avec des styles, des images, des formulaires, des animations et des fonctionnalités avancées. En utilisant ces langages de programmation, les développeurs Web peuvent créer des sites Web fonctionnels, interactifs et esthétiquement plaisants. La maîtrise de ces langages est essentielle pour tout étudiant ou professionnel du Web qui souhaite créer des sites Web professionnels et performants.

Le World Wide Web

" Le World Wide Web n'est pas un concept flou et lointain, mais une réalité quotidienne pour des millions de personnes dans le monde entier. "



Tim Berners-Lee'

1 LE WORLD WIDE WEB

1.1 HISTORIQUE DU WEB

Le World Wide Web (WWW), communément appelé le Web, est un système d'information permettant à des documents et autres ressources Web d'être consultés sur Internet. Les documents et les médias téléchargeables sont mis à disposition du réseau via des serveurs Web et peuvent être consultés par des programmes tels que les navigateurs Web.

Le World Wide Web est connu sous plusieurs noms et abréviations synonymes tels que World Wide Web, Web, WWW et W3. Le nom original du projet était WorldWideWeb mais il a été rapidement séparé en World Wide Web pour améliorer la lisibilité. Le World Wide Web Consortium (W3C) préconise maintenant d'écrire les trois mots séparément sans trait d'union. Tim Berners-Lee avait également envisagé d'autres noms pour le Web tels qu'Information Mesh ou Mine of Information. L'abréviation WWW a été largement utilisée avant que l'abréviation Web ne devienne plus populaire. L'usage de la minuscule pour écrire « le Web » est de plus en plus courant.

Le World Wide Web a été inventé en 1989 par le chercheur britannique Tim Berners-Lee lorsqu'il travaillait au CERN (l'Organisation Européenne pour la Recherche Nucléaire). À l'origine, le projet a été conçu et développé pour que des scientifiques travaillant dans des universités et instituts du monde entier puissent s'échanger des informations instantanément. Le premier site est devenu accessible en dehors du CERN un an plus tard. Le premier serveur Web aux États-Unis a été mis en ligne en décembre 1991 au Centre de l'Accélérateur Linéaire de Stanford (SLAC), en Californie

Le développement du Web jusqu'en 1993 était principalement dirigé par Tim Berners-Lee et Robert Cailliau. Cependant, l'arrivée de NCSA Mosaic a changé la donne. Ce navigateur Web a été développé par Eric Bina et Marc And reessen au National Center for Super Computing Applications (NCSA) en Illinois. Il a jeté les bases de l'interface graphique des navigateurs modernes en intégrant des images au texte, ce qui a entraîné une augmentation exponentielle de la popularité du Web. Plus tard, certains développeurs de Mosaic ont créé Netscape Navigator qui a introduit le parallélisme pour permettre l'utilisation de pages avec images sur des connexions bas débit. Ce navigateur est finalement devenu Mozilla Firefox après plusieurs évolutions. Le NCSA a également produit le NCSA HTTPd qui est devenu Apache HTTP Server, le serveur HTTP le plus utilisé depuis 1996. Le 30 avril 1993, le CERN a mis toutes les technologies développées autour du www dans le domaine public et pour marquer l'occasion, il a mis en ligne ce qui peut être considéré comme étant le premier site en "www", hébergé sur la station NeXT de TBL.

En 1995, le Web se développait à une vitesse folle et deux navigateurs qui ont marqué l'histoire du Web ont fait leur apparition : Netscape et Internet Explorer. C'était également l'année où Jeff Bezos a lancé la librairie en ligne Amazon.com depuis son garage de Bellevue dans la banlieue de Seattle.

En l'an 2000, le moteur de recherche de Google avait déjà indexé plus d'un milliard de pages et avait lancé Google AdWords avec 250 clients en octobre 2000. C'était également l'année où le nom de domaine "twitter.com" a été créé le 21 janvier 2000.

L'expression "Web 2.0" a commencé à être largement popularisée au milieu des années 2000 pour désigner la transition dans le flux de l'information et la manière d'utiliser le Web. De plus, des technologies pour le développement de sites Web interactifs (langage JavaScript) et dynamiques (langage PHP: Hypertext Preprocessor) ont été mises à disposition cette année-là.

En 2010, le HTML (HyperText Markup Language) a connu une douce révolution avec l'émergence du HTML5, toujours sous l'impulsion de Tim Berners-Lee. Et en En 2020, près de la moitié de la population mondiale, soit 3,8 milliards de personnes, utilisait les médias sociaux. Cela représente une hausse de 9% en un an (soit 321 millions de nouveaux internautes).

1.2 TERMES RATTACHES AU WEB

La page Web désigne l'unité élémentaire d'un site Web, lui-même constitué d'un nombre plus ou moins important de pages Web. Pour les internautes, la page Web est accessible via un navigateur Web (Firefox, Mozilla, Safari, Internet Explorer, etc.).

Une ressource est en ligne lorsqu'elle est connectée et disponible sur le réseau. Pour accéder à une ressource distante, il faut respecter un protocole de communication qui peut varier en fonctionnalités. Hypertext Transfer Protocol (HTTP) est le protocole communément utilisé pour transférer les ressources du Web et HTTPS est sa variante avec authentification et chiffrement.

L'expression « en ligne » signifie être connecté à un réseau tel qu'Internet ou un réseau téléphonique. Un hôte est un ordinateur connecté à un réseau. Une ressource du World Wide Web est une entité informatique accessible indépendamment d'autres ressources. Si une

ressource est en accès public, elle peut être librement accessible depuis tout Internet. Une ressource peut être locale si elle se trouve sur l'hôte de l'utilisateur ou distante si elle se trouve sur un autre hôte.

Un lien est un élément d'une ressource qui est lié à une URL (Uniform Resource Locator). Les liens hypertextes sur le Web sont directionnels : ils permettent de passer d'une source à une destination. Seule la ressource source contient les données définissant le lien hypertexte. Ainsi, il est possible de créer un lien hypertexte sans aucune intervention ni coopération de la part de la ressource cible du lien hypertexte. Ce modèle unilatéral adopté par le Web n'est pas présent dans tous les systèmes hypertextes. Il présente l'avantage majeur, dans un réseau public mondial, de permettre la création d'un nombre incalculable de liens hypertextes entre des partenaires décentralisés et non coordonnés.

Une URL (pour *Uniform Resource Locator*, signifiant littéralement « localisateur de ressource uniforme ») est une chaîne de caractères décrivant l'emplacement d'une ressource. Elle contient généralement dans l'ordre : le nom du protocole, un deuxpoints (:), deux barres obliques (//), un nom d'hôte, une barre oblique (/), et un chemin composé de texte séparés par des barres obliques. Par exemple: <http://www.univ-guelma.dz/fmism/dInf/index.html>

HTML (Hypertext Markup Language) et XHTML (Extensible Hypertext Markup Language) sont tous deux des langages utilisés pour écrire des pages Web. HTML est basé sur SGML (Standard Generalized Markup Language), tandis que XHTML est basé sur XML (eXtensible Markup Language). XHTML a été dérivé de HTML pour se conformer aux normes XML et est donc plus strict que HTML en ce qui concerne la codification et la structure. Les deux langages incluent une large gamme de fonctionnalités pour créer des pages Web et des applications riches et interactives, telles que la prise en charge du multimédia, du style et du script.

Le navigateur Web est un logiciel principalement gratuit qui permet de consulter les pages Internet. À l'aide d'un navigateur Web, des textes, des images, des vidéos, mais aussi des liens et d'autres fonctions d'un site Internet sont affichés. En gros, c'est une application logicielle qui permet d'accéder à Internet.

Un serveur Web peut être soit un logiciel de service de ressources Web (serveur HTTP), soit un serveur informatique (ordinateur) qui répond à des requêtes du World Wide Web sur un réseau public (Internet) ou privé (intranet) en utilisant principalement le protocole HTTP. Au niveau des composants matériels, un serveur Web est un ordinateur qui stocke les fichiers qui composent un site Web et qui les envoie à l'appareil de l'utilisateur qui visite le site.

Un client HTTP est un logiciel conçu pour se connecter à un serveur HTTP (Hypertext Transfer Protocol). Dans le cas le plus courant, il s'agit d'un navigateur Web. Les autres clients sont généralement des programmes automatisant des transferts. Par exemple, un robot d'indexation permet à un moteur de recherche d'explorer le contenu des sites Web.

Un éditeur HTML (ou éditeur Web) est un logiciel conçu pour faciliter l'écriture de documents HTML et de pages Web en général.

Un site Web est un ensemble de pages Web et de ressources reliées par des hyperliens, défini et accessible par une adresse Web. L'adresse Web d'un site correspond en fait à l'URL (Uniform Resource Locator) d'une page Web prévue pour être la première consultée : la page d'accueil.

1.3 CONSULTATION ET MESURE

Visiter un site Web signifie accéder à ses pages Web en utilisant un navigateur Web et en saisissant son adresse URL ou en cliquant sur un lien qui y mène. Une fois sur le site, vous pouvez naviguer entre les différentes pages en suivant les liens hypertextes qui les relient.

L'expression « surfer sur le Web » signifie « consulter le Web ». Elle a été inventée pour mettre l'accent sur le fait que consulter le Web consiste à suivre de nombreux hyperliens de page en page. Elle est principalement utilisée par les médias; elle n'appartient pas au vocabulaire technique.

- On appelle audience : la fréquentation d'un site.
- On appelle fil d'Ariane : le parcours effectué par un visiteur sur un site.
- On appelle journal d'accès : la liste des accès .

L'User-Agent est une référence fournie par le client HTTP au serveur HTTP pour identifier sa nature. Il peut inclure des informations telles que le nom et la version du client HTTP, le moteur de rendu HTML utilisé, les préférences linguistiques de l'utilisateur ou de l'outil client et d'autres critères tels que les modules d'extension installés sur le client. Ces informations permettent au serveur HTTP de déterminer quelles ressources sont les plus adaptées à ce client. Des technologies dynamiques telles que JavaScript et CSS permettent également aux pages de prendre en charge des clients différents et de s'adapter dynamiquement en utilisant leurs propres ressources plutôt que celles du serveur.

La mesure de l'audience est l'étude des consultations effectuées sur un site, elle vise à comprendre les désirs des utilisateurs, en fonction des ressources les plus utilisées et du fil d'Ariane de ceux-ci. La mesure d'audience peut se faire de différentes manières :

- Analyse des journaux du serveur. Les calculs de mesure de l'audience peuvent alors être effectués en temps réel au moment de la visite, ou bien en différé.
- Utilisation de scripts sur le serveur Web lui-même.
- Utilisation d'images invisibles déportées sur un serveur Web tiers (mesure d'audience simple).
- Utilisation de script en langage javascript et envoi des informations au serveur Web ou a un serveur tiers (mesure d'audience plus complète).

1.4 METIERS DU WEB

Les métiers du Web et du numérique sont très variés et couvrent des domaines tels que le commerce, la communication, le graphisme, l'informatique et la logistique. Le marketing digital comprend des spécialisations telles que le SEO (référencement naturel), le SEA (référencement payant), le SMO (optimisation des réseaux sociaux) et l'achat d'espaces publicitaires sur le

Web. D'autres métiers du Web incluent les développeurs Web (front-end, back-end et full-stack), les développeurs d'applications mobiles et de jeux vidéo, les trafics managers, les experts SEO, les rédacteurs Web et les Community Managers.

1.5 ÉVOLUTION DE L'USAGE

Tim Berners-Lee a créé le premier navigateur Web et éditeur Web en développant le logiciel WorldWideWeb. Son objectif était de faire du Web un média collaboratif où les utilisateurs pouvaient consulter et créer des informations. Cependant, le Web est rapidement devenu un média de diffusion d'informations plutôt que de collaboration.

Au début des années 1990, l'idée d'un site Web associé à un nom de domaine stable n'était pas encore établie. Les sites étaient souvent créés par des employés et des étudiants dans des départements techniques et les URL changeaient fréquemment en fonction des changements de personnel et d'infrastructure. De plus, il n'y avait pas encore de moteurs de recherche efficaces, donc beaucoup de pages consistaient en listes de liens vers les pages préférées de l'auteur.

Les créateurs de Google ont utilisé la propriété du Web pour calculer la pertinence des pages en exploitant les liens entre elles. Cette technique a ensuite été détournée par les fermes de liens. Avant cela, en janvier 1994, Yahoo! a été créé et est rapidement devenu le plus grand annuaire Web. Son URL originale était <http://akebono.stanford.edu/yahoo> car ses créateurs étaient des étudiants de l'université Stanford. Le domaine yahoo.com n'a été créé qu'en janvier 1995.

Dans la seconde moitié des années 1990, le Web est devenu de plus en plus populaire et toutes les grandes entreprises, organisations, écoles et administrations ont créé leur propre site Web. Les moteurs de recherche sont devenus plus efficaces avec l'arrivée d'Altavista en décembre 1995 et finalement de Google en 1998. À cette époque, la diffusion d'informations était principalement unidirectionnelle : les sites Web étaient utilisés pour diffuser les informations de leur propriétaire. Les interactions se limitaient souvent à la recherche et au commerce en ligne. Bien que le courrier électronique, la messagerie instantanée et les forums de discussion existaient avant le Web, ils utilisaient principalement des protocoles et des logiciels spécifiques.

1.6 LE WEB 2.0

Le Web 2.0 désigne la nouvelle étape de l'évolution d'Internet à partir des années 2000. Il est également connu sous les noms de Web participatif ou social. Le Web 2.0 met l'accent sur le contenu généré par les utilisateurs, la facilité d'utilisation, la culture participative et l'interopérabilité pour les utilisateurs finaux. Cela a été rendu possible grâce à l'apparition d'interfaces innovantes et à une utilisation plus facile de l'outil Web par les internautes malgré la complexification de la technologie.

Au cours des années 2000, les blogs, les wikis (comme Wikipédia lancé en anglais en 2001) et les réseaux sociaux (comme Myspace en 2003 et Facebook en 2004) sont devenus populaires. Le contenu généré par les utilisateurs s'est répandu grâce à des plateformes comme Wikipédia, YouTube (lancé en 2005) et Twitter (lancé en 2006). La technologie Ajax, développée en 1998

et théorisée en 2005, a commencé à être largement utilisée pour créer des applications complètes dans une seule page Web, comme Google Maps lancé en 2004.

L'expression Web 2.0 a été largement popularisée au milieu des années 2000 pour désigner cette transition dans le flux d'information et la manière d'utiliser le Web. Le succès de cette expression a conduit de nombreuses personnes à utiliser des termes tels que Web 2.5, Web 3.0 ou Web 4.0 pour décrire leur vision du futur du Web.

Le HTML5 est la dernière révision majeure du langage HTML (HyperText Markup Language), qui est un format de données conçu pour représenter les pages Web. Cette version a été finalisée le 28 octobre 2014 et spécifie deux syntaxes d'un modèle abstrait défini en termes de DOM: HTML5 et XHTML5. HTML5 représente une rupture majeure avec les pratiques précédentes de balisage et a pour but de standardiser les nombreuses nouvelles façons dont les développeurs utilisaient le langage, ainsi que d'encourager un ensemble unique de meilleures pratiques en matière de développement Web 2. Le terme HTML est essentiellement un mot à la mode qui désigne un ensemble de technologies Web modernes, y compris la norme vivante HTML ainsi que des API JavaScript pour améliorer le stockage, le multimédia et l'accès aux matériels 3.

1.7 LE WEB 3.0

Le Web 3.0 est un terme utilisé pour désigner l'idée d'un Web décentralisé exploitant la technologie des chaînes de blocs (blockchain). Il se veut ainsi le successeur du Web 2.0, qui désigne le Web « social ». Certains spécialistes évoquent le Web 3.0 comme étant la prochaine étape d'évolution du World Wide Web actuel. Selon ces analystes, cette prochaine étape pourrait désigner soit l'Internet des objets, soit l'émergence du Web sémantique. L'expression Web 3.0 est utilisée en futurologie à court terme pour désigner l'Internet qui suit le Web 2.0 et constitue l'étape à venir du développement du World Wide Web. Son contenu réel n'est pas défini de manière consensuelle, chacun l'utilisant pour désigner sa propre vision du futur d'Internet.

1.8 L'ARCHITECTURE DU WEB

Le World Wide Web peut être représenté comme un graphe orienté contenant des cycles, avec les ressources comme nœuds et les liens hypertextes comme arcs. Étant orienté, il peut y avoir des nœuds qui ne sont pas reliés à d'autres, ce qui les rend des "cul-de-sac", tandis que d'autres peuvent être des sources. Le modèle de réseau invariable d'échelle présent dans la plupart des réseaux sociaux est également présent dans la structure du Web, avec des hubs vers lesquels convergent les liens hypertextes, constituant les sites les plus importants du Web. Techniquement, le World Wide Web n'est pas différent des autres Web qui utilisent les mêmes technologies. Bien qu'il existe d'innombrables Web privés, une page d'un site Web populaire est considérée comme faisant partie du Web, car il peut être découvert récursivement à partir de cette page, ce qui exclut les sources et les Web privés.

1.9 EXPLORATION DU WEB ET WEB PROFOND

Les robots d'indexation des moteurs de recherche utilisent la méthode de l'exploration récursive du Web à partir de ressources judicieusement choisies. En 2004, environ 4 milliards de ressources ont été indexées par les moteurs de recherche. Le Web profond, ou Web invisible, est la partie du Web qui ne peut pas être trouvée par les moteurs de recherche généralistes car elle n'est pas indexée. Selon une étude publiée en 2001, le Web profond représente plus de 99% du Web. Le Web profond inclut plusieurs types de ressources telles que les pages administratives ou payantes protégées par un mot de passe, les ressources qui ne sont pas communiquées par les protocoles de communication pris en charge par les robots, celles dont le format de données n'est pas pris en charge, celles listées dans un fichier d'exclusion des robots, celles exclues par le robot pour spamdexing ou parce qu'elles sont considérées comme peu pertinentes. En outre, le Web profond inclut des ressources vers lesquelles les hyperliens sont créés dynamiquement en réponse aux requêtes des visiteurs, qui proviennent généralement de bases de données et constituent la partie la plus importante du Web profond.

1.10 DOCUMENT HTML

Le document HTML est composé de balises, qui sont des éléments de code qui indiquent au navigateur Web comment afficher le contenu de la page. Chaque balise est entourée de crochets angulaires et peut contenir des attributs qui fournissent des informations supplémentaires sur la balise.

Les balises HTML peuvent être utilisées pour créer des titres, des paragraphes, des listes, des liens hypertexte, des images et bien d'autres éléments. Elles peuvent également être utilisées pour spécifier la mise en forme et la mise en page de la page Web. Le document HTML est souvent combiné avec d'autres langages tels que CSS (Cascading Style Sheets) et JavaScript pour améliorer la présentation et le comportement de la page Web. En utilisant ces langages en conjonction, il est possible de créer des sites Web dynamiques et interactifs avec des fonctionnalités avancées telles que des animations, des formulaires et des effets visuels.



Figure 1. Exemple d'une page HTML

Le document HTML est un élément clé de la structure et de la présentation des ressources sur le Web. Il permet de lier et de structurer le contenu multimédia en utilisant des hyperliens. Contrairement aux ressources multimédia, un document HTML ne contient que du texte, qui est ensuite interprété par un navigateur Web.

La présentation des documents HTML est la principale fonctionnalité des navigateurs Web. Les langages de style et de script peuvent être utilisés pour améliorer la présentation, mais l'HTML laisse la plupart des décisions de mise en page au navigateur. Ainsi, la police de caractère, la longueur des lignes de texte et les couleurs peuvent être adaptées en fonction du périphérique de sortie (écran, imprimante, etc.).

1.11 MULTIMEDIA

Les ressources multimédias sont souvent indépendantes des documents HTML et sont liées par des hyperliens. Ainsi, ces ressources peuvent être distribuées sur Internet. Lorsqu'une page Web est présentée, les éléments multimédias correspondants sont automatiquement téléchargés. Les images et les animations de petite taille sont les seuls éléments multimédias standardisés, tandis que d'autres types tels que la vidéo, le son ou l'espace tridimensionnel ne le sont pas. Pour les prendre en charge, certains navigateurs Web permettent d'ajouter des logiciels (plugins) pour étendre leurs fonctionnalités. Toutefois, la gestion de flux audio et vidéo requiert souvent des protocoles de communication différents de HTTP, ce qui rend leur intégration dans les pages Web difficile et explique le besoin de plugins spécifiques.

1.11.1 Images & videos

Le format de données PNG est recommandé pour les images synthétiques telles que les logos et les éléments graphiques. Pour les images naturelles, il est préférable de l'utiliser uniquement si la qualité de l'image est plus importante que la durée de transfert. Le format de données GIF est approprié pour les petites animations, mais pour les images synthétiques, il est souvent privilégié

par défaut en raison de sa popularité passée. Toutefois, le format GIF présente quelques inconvénients tels que la limitation du nombre de couleurs et une compression moins efficace. De plus, il y a eu une controverse autour de son utilisation de 1994 à 2004, car Unisys a revendiqué un brevet sur la méthode de compression. Enfin, l'utilisation d'images au format de données XBM est désuète.

Avant les années 2000, pour visionner des contenus musicaux et vidéo en ligne, il était nécessaire d'installer un plugin dédié sur le navigateur Web. L'usage courant du plugin Flash Player a finalement simplifié le visionnage de vidéos à un niveau similaire à celui des images. C'est finalement l'intégration de la vidéo dans la cinquième version du langage HTML (HTML 5) qui a permis une utilisation aisée et généralisée de la vidéo en ligne.

1.11.2 Scripts et animations

Le langage de script permet d'écrire des programmes exécutés par un logiciel. Sur le Web, les scripts sont exécutés par les navigateurs Web pour répondre aux actions de l'utilisateur sur la page Web. Les scripts peuvent être intégrés au document HTML ou provenir de ressources externes. JavaScript fut le premier langage de script pour le Web, suivi par une variante concurrente de Microsoft nommée JScript. La norme ECMAScript a finalement été proposée pour la syntaxe du langage, tandis que les normes DOM ont été établies pour l'interface avec les documents.

Initialement limité dans ses capacités, le langage JavaScript est devenu capable d'exécuter toutes sortes d'applications, telles que le traitement de texte, les jeux vidéo, les émulateurs, etc. En plus de la vidéo, les animations étaient largement diffusées en utilisant le plugin Flash Player, parfois même pour des sites entiers créés en Flash. La gestion des autres types de ressource dépend des logiciels installés sur l'hôte client et de leurs réglages. Lorsque le logiciel correspondant est disponible, les documents et images de tout type sont généralement automatiquement présentés, selon des modalités (fenêtrage, dialogues) dépendant du navigateur Web et du logiciel gérant le type. Lorsque le type de la ressource n'est pas géré, il est généralement possible de la sauver dans un fichier local.

Le Web a été conçu pour être accessible à un large éventail d'équipements informatiques, notamment les ordinateurs personnels, les téléphones portables, les terminaux informatiques en mode texte, etc. Cette accessibilité universelle repose principalement sur l'universalité des protocoles Internet et la flexibilité de présentation des pages Web offerte par HTML. De plus, HTTP permet aux navigateurs de négocier le type de chaque ressource, tandis que CSS permet de proposer différentes présentations adaptées à l'équipement utilisé. Le W3C a créé des normes pour garantir l'interopérabilité entre les outils de création de contenu et les outils de lecture, afin de permettre l'accessibilité du Web aux personnes handicapées. Des initiatives telles que la Web Accessibility Initiative se consacrent spécifiquement à cet objectif..

1.12 DECENTRALISATION

Le Web permet une grande liberté d'organisation pour les pages et sites Web, avec la possibilité de créer des hyperliens vers toute autre ressource accessible sur Internet sans nécessiter une

action de la ressource pointée. Il n'y a pas de registre centralisé pour les hyperliens, pages ou sites, avec seule la base de données distribuée du DNS pour traduire les noms de domaine en adresses IP. Cette conception décentralisée a favorisé l'essor rapide du Web et l'émergence de sites spécialisés tels que les annuaires et les moteurs de recherche pour faciliter la recherche d'informations. Les portails Web cherchent quant à eux à rassembler un maximum d'informations et de services en un seul site. Cependant, cette décentralisation peut poser des problèmes lorsqu'une ressource est déplacée ou supprimée, ce qui peut entraîner des liens cassés et des erreurs 404.

1.13 COMMENT FONCTIONNENT LES PAGES WEB

Les pages Web sont créées à partir de codes HTML (HyperText Markup Language), CSS (Cascading Style Sheets) et JavaScript. Le HTML fournit la structure et le contenu de base de la page, tandis que le CSS est utilisé pour définir l'apparence et la mise en page de la page, y compris les couleurs, les polices de caractères, les marges, les bordures et autres éléments visuels. Le JavaScript est utilisé pour ajouter de l'interactivité à la page, tels que des animations, des boutons cliquables, des menus déroulants et des formulaires interactifs.

Lorsqu'un utilisateur demande une page Web, le serveur Web envoie les fichiers HTML, CSS et JavaScript au navigateur de l'utilisateur. Le navigateur analyse ensuite les fichiers et affiche la page Web en conséquence. La plupart des pages Web sont également dynamiques, ce qui signifie qu'elles sont générées à la volée en fonction des données fournies par l'utilisateur ou par le serveur. Cela peut inclure des fonctionnalités telles que des résultats de recherche en temps réel, des mises à jour en temps réel des messages ou des notifications, ou encore des recommandations de produits personnalisées en fonction des préférences de l'utilisateur.

Les noms de domaine Internet et World Wide Web Internet repose sur le concept fondamental selon lequel tous les ordinateurs devraient pouvoir se joindre à l'aide d'une adresse. Tout comme votre maison, appartement ou école a une rue adresse où le bureau de poste peut vous joindre, où vous avez un numéro de téléphone où quelqu'un peut vous appeler, tous les ordinateurs connectés à Internet ont une adresse connue sous le nom de TCP/IP adresse. Le TCP/IP signifie Transmission Control Protocol/Internet Protocol, mais ce n'est pas d'importance critique. Ce que vous devez savoir, c'est que TCP/IP autorise les paquets de données numériques des informations, telles que votre site Web, à envoyer sur les réseaux, puis à réassembler une fois qu'elles atteignent sa destination. Les adresses TCP/IP sont communément appelées adresses IP.

Lorsqu'un fichier est transmis sur Internet, il est d'abord décomposé en petits paquets de données. Ces paquets sont ensuite envoyés sur Internet vers leur destination, où ils sont réassemblés dans le fichier d'origine.

Pour transmettre les paquets, les données sont d'abord converties en code binaire, qui est une série de 0 et de 1. Ce code binaire est ensuite transmis sur Internet à l'aide d'un protocole tel que TCP/IP (Transmission Control Protocol/Internet Protocol).

Lorsqu'un paquet est envoyé sur Internet, il est d'abord étiqueté avec ses adresses IP source et de destination, ainsi qu'un identifiant unique qui permet à l'ordinateur récepteur de réassembler les

paquets dans le bon ordre. Les paquets sont ensuite envoyés sur Internet vers leur destination via une série de routeurs et de commutateurs. Ces appareils aident à diriger les paquets vers leur destination en utilisant les adresses IP et d'autres informations contenues dans les en-têtes de paquets.

Une fois que tous les paquets ont été transmis, ils sont réassemblés dans le fichier d'origine par l'ordinateur récepteur. L'identifiant unique inclus avec chaque paquet permet à l'ordinateur récepteur de remettre les paquets dans le bon ordre, et le fichier est alors prêt à être ouvert et utilisé par le destinataire.



Figure 2. Comment les fichiers se transmettent sur Internet

1.14 NOMS DE DOMAINE ET HEBERGEMENT

Les noms de domaine aident les utilisateurs à se repérer sur Internet. Vous connaissez déjà le domaine noms parce qu'ils sont généralement entourés de `www` au début et de `.com` à la fin. Les domaines peuvent également inclure diverses terminaisons telles que `.dz`, `.org`, `.net` et `.gov`. Les noms de domaine existent parce qu'il n'est pas très pratique pour vous d'utiliser ou de mémoriser des adresses IP. Les serveurs de noms de domaine (DNS) traduisent des noms de domaine faciles à comprendre en adresses IP. Un DNS convertit une chaîne de lettres familière, le « nom de domaine », en adresse IP numérotée.

Au lieu de saisir l'adresse IP `172.16.0.2` dans un navigateur Web, vous pouvez saisir le domaine nom, tel que `www.univ-guelma.dz`. Un DNS sur Internet convertit votre demande domaine dans l'adresse IP appropriée, qui achemine votre demande vers le serveur Web approprié.

Le serveur Web est un ordinateur qui ressemble beaucoup à un ordinateur de bureau. Il exécute généralement soit une version de Microsoft Windows Web Server ou UNIX, mais il peut avoir une puissance de traitement supplémentaire et des systèmes redondants pour gérer le trafic de milliers d'utilisateurs en même temps. Les Serveurs Web maintenir une connexion permanente à Internet, afin que vos sites Web soient disponibles 24 heures sur 24.

Parce que la plupart des entreprises veulent que leurs serveurs Web soient disponibles toute la journée, tous les jours, ils sont souvent maintenus par des sociétés d'hébergement Web. Ces entreprises sont payées pour entretenir votre site serveur Web afin qu'il soit toujours accessible et

en cours d'exécution. Si vous gérez un petit site Web, ils peuvent mettre votre site sur un serveur partagé avec d'autres sites. Pour les chantiers plus exigeants, ou les chantiers avec informations sensibles, une entreprise paiera des frais plus élevés pour un serveur dédié. Même grand les entreprises se tournent souvent vers des entreprises d'hébergement pour maintenir leurs serveurs Web, bien que certains les entreprises peuvent choisir de placer leurs serveurs Web au sein de leur propre entreprise.

1.15 LE LANGAGE DU WEB

Le langage du Web fait référence à l'ensemble des langages de programmation utilisés pour créer des pages Web. Ces langages permettent aux développeurs de concevoir et de mettre en ligne des sites Web interactifs et dynamiques. Les principaux langages de programmation du Web sont :

- HTML (Hypertext Markup Language) : Il est utilisé pour structurer le contenu de la page Web en utilisant des balises qui définissent les éléments de la page.
- CSS (Cascading Style Sheets) : Il est utilisé pour ajouter de la mise en forme et de la présentation à la page HTML en définissant des règles pour la police, les couleurs, les images et autres éléments de style.
- JavaScript : Il est utilisé pour ajouter de l'interactivité à la page Web en permettant la manipulation de la page, la validation de formulaires, l'affichage de messages d'erreur et autres fonctionnalités.
- PHP (Hypertext Preprocessor) : Il est utilisé pour la programmation côté serveur pour créer des pages Web dynamiques qui se connectent à des bases de données et permettent l'affichage de contenu personnalisé.
- SQL (Structured Query Language) : Il est utilisé pour la gestion des bases de données et permet de stocker et de récupérer des données à partir de sites Web.

Le HTML

*"Le HTML est la base de tout site Web.
Sans lui, il n'y aurait pas de site Web."*

Tim Berners-Lee'



2 LE HTML

Le HTML (Hypertext Markup Language) est un langage de balisage utilisé pour structurer et afficher du contenu sur le Web. Il permet aux développeurs de créer des pages Web en définissant des éléments tels que le texte, les images, les liens, les formulaires, les listes et autres éléments. Les balises HTML définissent la structure et le contenu d'une page Web en utilisant des paires de balises, qui indiquent où commence et où se termine chaque élément. Par exemple, voici le code HTML qui affiche un titre et un paragraphe :

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Titre de la page</title>
5 </head>
6 <body>
7   <h1>Titre principal</h1>
8   <p>Ceci est un paragraphe.</p>
9 </body>
10 </html>
11
```

Figure 3. Code HTML pour afficher un titre

Dans cet exemple, la balise <!DOCTYPE html> définit le type de document HTML utilisé, la balise <html> marque le début de la page Web, la balise <head> contient des informations sur la page (comme le titre), et la balise <body> contient le contenu de la page. La balise <h1> est

utilisée pour créer un titre de niveau 1, la balise <p> est utilisée pour créer un paragraphe. Les balises <title> et <head> ne sont pas visibles à l'écran, mais sont utilisées pour fournir des informations importantes sur la page aux moteurs de recherche et aux navigateurs Web.

Le langage de balisage hypertexte, ou HTML, utilise des balises qui entourent du texte brut. Les balises décrivent la façon dont le texte doit apparaître et la fonction du texte. Le navigateur Web examine les balises et les affiche en conséquence. Voici un exemple simple de texte HTML :

```
<p>Voulez-vous apprendre ? </p>
```

Le texte à afficher, Voulez-vous apprendre ? est entouré de deux balises indiquant qu'il s'agit d'un paragraphe. La première balise est la balise d'ouverture <p> et la seconde est la balise de fermeture </p>. Ces balises ne sont généralement pas affichées dans le navigateur, qui lit le texte à partir du serveur Web et le formate sous forme de paragraphe pour l'afficher sur l'écran du spectateur.

Le langage HTML permet également aux concepteurs de créer des hyperliens. Les hyperliens sont des zones de texte, des images, des boutons ou d'autres parties d'une page sur lesquelles l'utilisateur peut cliquer pour accéder à un contenu supplémentaire. Un clic sur un lien peut ouvrir une nouvelle page Web, un nouveau site, un nouveau document, une nouvelle vidéo ou une nouvelle animation.

Le HTML est un langage fondamental pour la création de sites Web. Il est souvent utilisé en conjonction avec d'autres langages de programmation, tels que CSS et JavaScript, pour créer des pages Web dynamiques et interactives.

2.1 L'EVOLUTION DU WEB ET DES NORMES WEB

Le code HTML est interprété par les navigateurs Web, tels qu'Internet Explorer, Firefox, Safari et Chrome. Les concepteurs Web ont découvert que le même code HTML pouvait s'afficher différemment sur différents navigateurs Web. Comme les navigateurs Web peuvent interpréter le code HTML différemment, vous devrez tenir compte des tests de navigateur dans vos considérations de conception.

Aux premiers jours du Web, certains développeurs de navigateurs ont créé des balises HTML propriétaires. Ils créaient des balises qui n'étaient prises en charge que par leur navigateur, car ils espéraient que ces capacités uniques attireraient davantage d'utilisateurs. Les concepteurs ont rapidement découvert qu'ils ne pouvaient pas compter sur le même code HTML pour tous les navigateurs.

Ils ont donc ajouté des "hacks", c'est-à-dire du code supplémentaire, aux pages, pour s'assurer que les mises en page fonctionnaient dans différents navigateurs. Certains concepteurs allaient jusqu'à créer deux versions d'un site, la version appropriée s'affichant en fonction du navigateur utilisé par l'internaute. D'autres concepteurs ajoutaient des badges à leurs sites, indiquant aux visiteurs que le site fonctionne mieux avec un navigateur particulier. Aujourd'hui encore, on peut voir sur certains sites des indications telles que "Ce site est optimisé pour Internet Explorer" ou pour un autre navigateur.

Les concepteurs, les entreprises et les sociétés qui développent les navigateurs ont fini par découvrir que l'incohérence et les balises propriétaires qui ne fonctionnaient qu'avec leur navigateur nuisaient à l'expérience des utilisateurs et à la vitalité et à la croissance du Web. Les différents fabricants de navigateurs ont commencé à reconnaître l'importance de la cohérence et la plupart d'entre eux travaillent avec le World Wide Web Consortium (W3C) et des organismes de test indépendants pour valider la conformité de leur navigateur aux normes acceptées. Les développeurs de navigateurs font désormais la promotion de leur conformité aux normes et de leur rapidité d'affichage des pages basées sur les normes. Maintenant que vous avez compris l'histoire et les concepts du Web, vous allez commencer à examiner le code HTML qui alimente les pages Web.

2.2 QU'EST-CE QU'UNE BALISE HTML ?

Une balise HTML est un élément qui permet de spécifier comment le navigateur doit afficher le contenu d'une page Web. Contrairement à du texte ordinaire, une balise HTML n'est pas affichée telle quelle dans le navigateur, mais est interprétée par celui-ci pour appliquer un formatage, une structure ou une action spécifique. Elle est toujours encadrée par des signes de chevrons '<' et '>' pour indiquer où commence et où se termine la balise, par exemple la balise : `<html>`.

La plupart des balises HTML ont une balise de fermeture correspondante avec le même nom, précédée d'un caractère "/". En utilisant ces balises ouvrantes et fermantes, la commande spécifiée ne s'applique qu'au texte situé entre les deux balises, exemple : `<HTML> ... </HTML>`. Entre ces deux balises, on définit deux zones :

- L'en-tête, spécifié par les commandes : `<HEAD>` et `</HEAD>`,
- Le corps, délimité par : `<BODY>` et `</BODY>`

Et voici la structure de base d'un document html :

```
<HTML>
<HEAD>
...
</HEAD>
<BODY>
...
</BODY>
</HTML>
```

L'entête (ou header) d'un document HTML est la partie située entre les balises `<head>` et `</head>` et qui précède le contenu de la page défini dans la balise `<body>`. Il contient des informations telles que le titre de la page, les liens vers des fichiers CSS et JavaScript, les métadonnées (comme la description de la page), les icônes de favoris, etc.

```
<!DOCTYPE html>
<html>
<head>
  <title>Titre de la page</title>
  <meta charset="utf-8">
  <link rel="stylesheet" href="style.css">
  <script src="script.js"></script>
  <!-- autres balises d'entête -->
</head>
<body>
  <!-- contenu de la page -->
</body>
</html>
```

Figure 4. La partie entête de document HTML

Dans cet exemple, la balise `<title>` définit le titre de la page qui apparaîtra dans l'onglet du navigateur. La balise `<meta>` définit les méta-données de la page, telles que la description de la page, le type de caractère utilisé pour l'encodage, etc. Les balises `<link>` et `<script>` permettent de lier des fichiers CSS et JavaScript à la page HTML.

Le corps (ou body) d'un document HTML est la partie située entre les balises `<body>` et `</body>` et qui contient le contenu visible de la page Web. C'est dans cette partie que l'on ajoute le texte, les images, les liens, les formulaires et autres éléments qui composent la page.

```
<!DOCTYPE html>
<html>
<head>
  <title>Titre de la page</title>
</head>
<body>
  <h1>Titre principal</h1>
  <p>Ceci est un paragraphe.</p>
  
  <a href="https://www.example.com">Lien vers un autre site</a>
  <!-- autres éléments du corps de la page -->
</body>
</html>
```

Figure 5. La partie corps de document HTML

Dans cet exemple, la balise `<h1>` définit un titre de niveau 1, la balise `<p>` définit un paragraphe et la balise `` insère une image. La balise `<a>` permet de créer un lien vers une autre page Web.

Le corps de document HTML est la partie la plus importante de la page car c'est là où se trouve le contenu visible de la page Web. C'est donc dans cette partie que l'on doit apporter une attention particulière pour créer une page Web bien structurée et facilement compréhensible par les utilisateurs.

2.3 LES BALISES ELEMENTAIRES

HTML est constitué d'un ensemble de balises qui permettent de définir la structure et le contenu d'une page Web. Voici quelques exemples de balises HTML élémentaires :

- **<html>** : balise racine qui enveloppe toute la page Web
- **<head>** : balise qui contient les métadonnées et les informations sur la page (titre, liens CSS, etc.)
- **<body>** : balise qui contient le contenu visible de la page
- **<h1>** à **<h6>** : balises pour les titres de niveaux différents
- **<p>** : balise pour les paragraphes de texte
- **<a>** : balise pour les liens hypertextes
- **** : balise pour les images
- **** et **** : balises pour les listes à puces
- **** et **** : balises pour les listes numérotées
- **<table>** : balise pour les tableaux
- **<tr>**, **<th>** et **<td>** : balises pour les rangées, les en-têtes et les cellules d'un tableau
- **<form>** : balise pour les formulaires interactifs
- **<input>** : balise pour les champs de saisie, les boutons radio et les cases à cocher

Il existe bien sûr de nombreuses autres balises HTML qui permettent de créer des pages Web plus complexes et plus riches en fonctionnalités. Chaque balise a une fonction spécifique et peut être utilisée de différentes manières pour créer des pages Web de qualité.

```

1 ▼ <HTML>
2 ▼ <HEAD>
3 <TITLE> Titre du document</TITLE>
4 </HEAD>
5 ▼ <BODY>texte simple<BR>
6 <B>texte en gras</B><BR>
7 <STRONG>texte en gras</STRONG><BR>
8 <I>texte en italique</I><BR>
9 <EM>texte en italique</EM><BR>
10 <B><I>texte en gras et en italique</I></B><BR>
11 <FONT SIZE=5>texte</FONT>
12 <FONT COLOR="#0000FF">en bleu</FONT>
13 <!--C'est fini-->
14 </BODY>
15 </HTML>
16

```

Figure 6. Les balises élémentaires de l'HTML

2.4 LES COULEURS EN HTML

En HTML, les couleurs peuvent être définies en utilisant différentes méthodes :

Les couleurs nommées : il est possible d'utiliser le nom de la couleur pour définir sa valeur. Par exemple, *red* pour rouge, *blue* pour bleu, *green* pour vert, etc.

Les couleurs hexadécimales : il s'agit d'une combinaison de six chiffres ou lettres (0 à 9 et A à F) précédée du symbole dièse (#). Chaque paire de chiffres ou lettres correspond à une composante

de la couleur (rouge, vert et bleu). Par exemple, #FF0000 correspond à la couleur rouge pure, #00FF00 à la couleur verte pure et #0000FF à la couleur bleue pure.

Les couleurs RGB : il est possible de définir une couleur en utilisant les valeurs de ses trois composantes (rouge, vert et bleu) sur une échelle de 0 à 255. Par exemple, rgb(255, 0, 0) correspond à la couleur rouge pure, rgb(0, 255, 0) à la couleur verte pure et rgb(0, 0, 255) à la couleur bleue pure.

Les couleurs RGBA : similaire aux couleurs RGB, mais avec une quatrième valeur pour la transparence de la couleur (alpha). La valeur alpha varie entre 0 (complètement transparent) et 1 (complètement opaque). Par exemple, rgba(255, 0, 0, 0.5) correspond à une couleur rouge semi-transparente. Voici quelques exemples d'utilisation de couleurs en HTML :

- color: red; : définit la couleur du texte en rouge
- background-color: #00FF00; : définit la couleur de fond en vert pur
- border-color: rgb(255, 0, 0); : définit la couleur de bordure en rouge pur
- box-shadow: 0 0 10px rgba(0, 0, 255, 0.5); : définit une ombre portée avec une couleur bleue semi-transparente.

Il est important de choisir des couleurs qui s'accordent bien entre elles pour créer des pages Web harmonieuses et agréables à l'œil. Il est également important de tenir compte de la lisibilité du texte en choisissant des couleurs suffisamment contrastées avec le fond.

```
1 <HTML>
2 <HEAD>
3 <TITLE> Titre du document </TITLE>
4 </HEAD>
5 <BODY>texte simple<BR>
6 <B>texte en gras</B><BR>
7 <STRONG>texte en gras</STRONG><BR>
8 <I>texte en italique</I><BR>
9 <EM>texte en italique</EM><BR>
10 <B><I>texte en gras et en italique</I></B><BR>
11 <FONT SIZE=5>texte</FONT>
12 <FONT COLOR="#0000FF">en bleu</FONT><BR>
13 <FONT COLOR="#00FF00">en Vert</FONT><BR>
14 <FONT COLOR="#FFFFFF">en blanc</FONT><BR>
15 <FONT COLOR="#8000FF">en violet</FONT><BR>
16 <FONT COLOR="#FF0000">en rouge</FONT><BR>
17 <FONT COLOR="#FFFF00">en jaune</FONT><BR>
18 <FONT COLOR="#C0C0C0">en gris clair</FONT><BR>
19 <FONT COLOR="#000000">en noir</FONT><BR>
20 </BODY>
21 </HTML>
```

Figure 7. Exemple d'utilisation des couleurs avec le HTML

2.5 LES TITRES

Pour donner une taille aux caractères d'un titre dans un texte avec un saut de ligne à la fin, en utilise la balise <Hn></Hn>, où n peut prendre une valeur comprise entre 1 et 6 (dans l'ordre décroissante des tailles)

```

1 ▼ <HTML>
2 ▼ <HEAD>
3   <TITLE> Titre du document</TITLE>
4 </HEAD>
5 ▼ <BODY>texte simple<BR>
6 <B><I><H1>texte en gras et en italique et H1</H1></I></B>
7 <B><I><H2>texte en gras et en italique et H2</H2></I></B>
8 <B><I><H3>texte en gras et en italique et H3</H3></I></B>
9 <B><I><H4>texte en gras et en italique et H4</H4></I></B>
10 <B><I><H5>texte en gras et en italique et H5</H5></I></B>
11 <B><I><H6>texte en gras et en italique et H6</H6></I></B>
12 </BODY>
13 </HTML>

```

Figure 8. Les différents niveaux de la balise Heading

En HTML, certains caractères ont une signification particulière et ne peuvent pas être utilisés tels quels dans le texte d'une page Web. Pour insérer ces caractères spéciaux dans le contenu d'une page HTML, il est nécessaire d'utiliser des codes de caractères spéciaux, qui sont des séquences de symboles prédéfinis. Voici quelques exemples de caractères spéciaux couramment utilisés en HTML :

- < : représente le symbole < (inférieur à)
- > : représente le symbole > (supérieur à)
- & : représente le symbole & (esperluette)
- " : représente le symbole " (guillemet double)
- ' : représente le symbole ' (apostrophe)

Il existe de nombreux autres codes de caractères spéciaux en HTML, chacun représentant un caractère spécifique qui ne peut pas être utilisé directement dans le texte. Il est important d'utiliser ces codes de caractères spéciaux pour garantir que le texte sera affiché correctement dans tous les navigateurs Web, quel que soit le système d'exploitation ou le jeu de caractères utilisé.

2.6 LES LISTES EN HTML

En HTML, il est possible de créer des listes de plusieurs types différents. Voici les trois types de listes les plus couramment utilisés :

La liste à puces (pour "unordered list") : cette liste est composée d'une série d'éléments qui sont affichés avec des puces devant chaque item. Les éléments de la liste sont définis à l'aide de la balise (pour "list item"). Exemple de code HTML :

```
1 ▼ <HTML>
2 ▼ <HEAD>
3 <TITLE> Titre du document
4 </TITLE>
5 </HEAD>
6 ▼ <BODY>texte simple<BR>
7 ▼ <OL>
8 <LI> élément 1,
9 <LI> élément 2,
10 <LI> élément 3.
11 </OL>
12 </BODY>
13 </HTML>
```

Figure 9. Une liste ordonnée

La liste numérotée (pour "ordered list") : cette liste est similaire à la liste à puces, sauf que les éléments sont numérotés dans l'ordre. Les éléments de la liste sont également définis à l'aide de la balise . Exemple de code HTML :

```
8 ▼ <UL>
9 <LI> élément 1,
10 <LI> élément 2,
11 <LI> élément 3.
12 </UL>
```

Figure 10. Liste non ordonnée

La liste de définitions (<dl> pour "definition list") : cette liste est utilisée pour définir des termes et leurs définitions. Chaque terme est défini à l'aide de la balise <dt> (pour "definition term"), et chaque définition est définie à l'aide de la balise <dd> (pour "definition description"). Exemple de code HTML :

```
8 ▼ <dl>
9 <dt>Terme 1</dt>
10 <dd>Définition de terme 1</dd>
11 <dt>Terme 2</dt>
12 <dd>Définition de terme 2</dd>
13 <dt>Terme 3</dt>
14 <dd>Définition de terme 3</dd>
15 </dl>
16
```

Figure 11. La liste de définitions

Notez qu'il est possible d'imbriquer des listes du même type ou de types différents.

2.6.1 Les attributs des listes

En HTML, les listes ordonnées () et les listes non ordonnées () peuvent être personnalisées à l'aide de différents attributs. Voici quelques-uns des attributs les plus couramment utilisés :

- **type** : cet attribut permet de spécifier le type de puce ou de numérotation utilisé dans la liste. Les valeurs possibles pour cet attribut sont "1" pour une numérotation décimale, "A" pour une numérotation alphabétique en majuscules, "a" pour une numérotation alphabétique en minuscules, "I" pour une numérotation romaine en majuscules et "i" pour une numérotation romaine en minuscules.
- **start** : cet attribut permet de spécifier le numéro de départ pour une liste ordonnée.
- **reversed** : cet attribut permet d'inverser l'ordre de numérotation dans une liste ordonnée.
- **class** et **id** : ces attributs permettent de définir des classes et des identifiants CSS pour une liste. Cela permet de personnaliser l'apparence de la liste à l'aide de styles CSS.

Exemple de code HTML :

```
8 <ol type="I">
9   <li>Item 1</li>
10  <li>Item 2</li>
11  <li>Item 3</li>
12 </ol>
```

Figure 12. Liste ordonnée avec attribut

2.7 LES TABLEAUX

En HTML, les tableaux sont créés à l'aide des balises <table>, <tr> et <td>. Voici comment créer un tableau simple :

```
8 <table>
9 <tr>
10  <td>Colonne 1, ligne 1</td>
11  <td>Colonne 2, ligne 1</td>
12 </tr>
13 <tr>
14  <td>Colonne 1, ligne 2</td>
15  <td>Colonne 2, ligne 2</td>
16 </tr>
17 </table>
```

Figure 13. Création d'une table

Dans cet exemple, la balise <table> crée un tableau, la balise <tr> crée une ligne dans le tableau et la balise <td> crée une cellule dans la ligne. Les cellules sont remplies de texte ou d'autres éléments HTML.

Il est possible de personnaliser les tableaux à l'aide de différents attributs. Voici quelques-uns des attributs les plus couramment utilisés :

- **border** : cet attribut permet de spécifier l'épaisseur de la bordure du tableau.

- cellspacing : cet attribut permet de spécifier l'espace entre les cellules du tableau.
- cellpadding : cet attribut permet de spécifier l'espace entre le contenu des cellules et leur bordure.
- width : cet attribut permet de spécifier la largeur du tableau.
- height : cet attribut permet de spécifier la hauteur du tableau.

Voici un exemple de code HTML pour un tableau personnalisé :

```
8 <table border="1" cellspacing="0" cellpadding="5" width="100%" height="200">
9   <tr>
10    <th>Colonne 1</th>
11    <th>Colonne 2</th>
12  </tr>
13  <tr>
14    <td>Contenu 1</td>
15    <td>Contenu 2</td>
16  </tr>
17 </table>
```

Figure 14. La personnalisation de la table

Dans cet exemple, la bordure du tableau a une épaisseur de 1 pixel, l'espace entre les cellules est de 0 pixel et l'espace entre le contenu des cellules et leur bordure est de 5 pixels. Le tableau a une largeur de 100% de la page et une hauteur de 200 pixels. Les balises <th> sont utilisées pour les en-têtes de colonnes.

2.7.1 colgroup

En HTML, la balise <colgroup> permet de regrouper des éléments <col> au sein d'un tableau et d'appliquer des styles à l'ensemble de ces éléments.

Voici un exemple de code HTML avec <colgroup> et <col> :

```
8 <table>
9   <colgroup>
10    <col style="background-color: yellow;">
11    <col style="background-color: pink;">
12  </colgroup>
13  <tr>
14    <td>Colonne 1</td>
15    <td>Colonne 2</td>
16  </tr>
17  <tr>
18    <td>Contenu 1</td>
19    <td>Contenu 2</td>
20  </tr>
21 </table>
22
```

Figure 15. L'utilisation de la balise COLGROUP

Dans cet exemple, la balise <colgroup> est utilisée pour regrouper deux éléments <col>, qui ont chacun un attribut style pour définir la couleur de fond. Les styles sont appliqués à l'ensemble des colonnes, ce qui permet d'éviter de devoir spécifier le style pour chaque cellule. Les éléments <tr> et <td> sont utilisés pour créer les lignes et les cellules du tableau, comme dans un tableau classique.

Il est également possible d'utiliser des attributs sur la balise `<colgroup>` pour définir des propriétés de colonnes, comme la largeur ou l'alignement :

- `span` : cet attribut permet de spécifier le nombre de colonnes que doit occuper le groupe de colonnes.
- `width` : cet attribut permet de spécifier la largeur de toutes les colonnes du groupe.
- `align` : cet attribut permet de spécifier l'alignement horizontal des colonnes du groupe.

2.7.2 Tableau à colonne de taille différente

La balise `<col>` avec le paramètre `width` permet de définir la taille de chacune des colonnes d'un tableau.

```

8 <table border>
9 <colgroup >
10 <col width= 90>
11 <col width= 140>
12 <col width= 250>
13 </colgroup>
14 <tr>
15 <th> Titre1 </th>
16 <th> Titre2 </th>
17 <th> Titre3 </th>
18 </tr>
19 <tr>
20 <td> ligne 1, Colonne1</td>
21 <td> ligne 1, Colonne2</td>
22 <td> ligne 1, Colonné3 </td>
23 </tr>
24 <tr>
25 <td> ligne 2, Colonne1</td>
26 <td> ligne 2, Colonne2</td>
27 <td> ligne 2, Colonné3 </td>
28 </tr>
29 </table>

```

Figure 16. Tableau a colonne de taille différente

2.7.3 Couleur de fond et alignement du contenu de colonne

Le paramètre `bgcolor`, de la balise `<col>` permet de définir la couleur de fond d'une colonne.

Le paramètre `align`, de la balise `<col>` permet d'aligner l'écriture d'être à gauche (left), à droite (right) ou centrer (center).

```

8 <table border>
9
10 <tr>
11 <td bgcolor = #8FFFE5> ligne 1, Colonne1</td>
12 <td align = right> ligne 1, Colonne2</td>
13 <td align = center> ligne 1, Colonné3 </td>
14 </tr>
15
16 </table>
17

```

Figure 17. Personnalisation de la table

2.8 LES LIENS HYPERTEXTES

Un lien HTML est un élément qui permet à un utilisateur de cliquer sur un texte ou une image pour être redirigé vers une autre page ou un autre emplacement sur la même page. Voici un exemple de code HTML pour créer un lien :

```

6 <BODY>
7
8 <a href="https://www.univ-guelma.dz">Cliquez ici pour visiter l'université de Guelma</a>
9
10 </BODY>

```

Figure 18. L'utilisation de la balise <a>

Dans ce code, la balise <a> est utilisée pour créer le lien. L'attribut href spécifie l'URL vers laquelle le lien doit rediriger. Le texte qui apparaît entre les balises <a> et est le texte cliquable affiché sur la page. Il est également possible de créer un lien qui redirige vers un emplacement spécifique sur la même page en utilisant l'attribut href avec un identifiant de section. Voici un exemple de code pour un tel lien :

```
7 <BODY>
8
9 <a href="#section2">Cliquez ici pour accéder à la section 2</a>
10 ...
11 <h2 id="section2">Section 2</h2>
12 <p>Ceci est le contenu de la section 2.</p>
13
14 </BODY>
```

Figure 19. La balise <a> avec les attributs

Dans ce code, le lien utilise l'attribut href="#section2" pour rediriger l'utilisateur vers l'élément ayant l'identifiant "section2" sur la même page. L'élément correspondant est une balise <h2> avec un attribut id="section2".

2.9 LE LAGAGE XML

XML (eXtensible Markup Language) est un langage de balisage qui permet de structurer et d'organiser des données de manière hiérarchique. Il est utilisé pour l'échange de données entre des applications hétérogènes, car il est indépendant de toute plateforme ou langage de programmation. Voici un exemple de document XML :

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <livre>
3   <titre>Le seigneur des anneaux</titre>
4   <auteur>J.R.R. Tolkien</auteur>
5   <dateParution>1954-1955</dateParution>
6   <personnages>
7     <personnage>Gandalf</personnage>
8     <personnage>Frodon</personnage>
9     <personnage>Legolas</personnage>
10  </personnages>
11 </livre>
12
```

Figure 20. Exemple d'un fichier XML

Dans cet exemple, on a un élément racine <livre> qui contient plusieurs éléments enfants tels que <titre>, <auteur>, <dateParution> et <personnages>. L'élément <personnages> contient lui-même plusieurs éléments enfants <personnage>.

XML permet de créer des structures de données personnalisées et de les organiser de manière hiérarchique, ce qui facilite l'échange de données entre des applications qui ne partagent pas le même langage de programmation ou la même plateforme.

Le langage XML est utilisé pour de nombreuses raisons, en voici quelques-unes :

- Structure de données : Le XML permet de structurer les données de manière hiérarchique grâce à l'utilisation de balises. Cette hiérarchie permet une lecture et une compréhension plus aisée des données.

- Indépendance : Le XML est un langage indépendant de toute plate-forme ou système, ce qui facilite l'échange de données entre différents systèmes et applications.
- Lisibilité : Le XML est un langage lisible par les humains, ce qui facilite la lecture et la compréhension du contenu des documents XML.
- Extensibilité : Le XML est un langage extensible qui permet aux développeurs de définir des balises spécifiques pour leurs propres besoins, ce qui permet une grande flexibilité pour la représentation des données.
- Facilité d'utilisation : Le XML est facile à utiliser et peut être traité par de nombreux outils logiciels, notamment des outils pour la validation, la transformation, la recherche et la visualisation des données XML.
- Compatibilité avec les standards : Le XML est largement utilisé dans l'industrie et est compatible avec de nombreux standards et technologies connexes tels que XPath, XSLT, RDF, SOAP, etc.

En résumé, le XML est utilisé car il fournit une structure flexible, indépendante de la plate-forme et lisible par les humains pour la représentation des données. Il permet également une grande compatibilité avec les standards et les technologies connexes.

2.9.1 Le XPath

XPath (XML Path Language) est un langage de requête utilisé pour extraire des informations spécifiques à partir d'un document XML. Il permet de naviguer dans la structure hiérarchique d'un document XML en utilisant des expressions de chemin pour identifier des éléments ou des attributs. Voici quelques exemples d'expressions XPath :

- `/livres/livre`: sélectionne tous les éléments livre qui sont des enfants directs de l'élément livres.
- `//auteur`: sélectionne tous les éléments auteur dans le document, quel que soit leur emplacement.
- `/livres/livre[1]`: sélectionne le premier élément livre qui est un enfant direct de l'élément livres.
- `//livre[@id="123"]`: sélectionne tous les éléments livre qui ont un attribut id avec la valeur "123".
- `//livre[annee/text() > 2000]`: sélectionne tous les éléments livre qui ont un élément année avec un contenu supérieur à 2000.

XPath est largement utilisé dans de nombreuses applications et bibliothèques de traitement XML, telles que XSLT, XQuery, DOM, SAX, etc. Il permet de sélectionner, filtrer et manipuler les données XML de manière efficace et précise.

2.9.2 Le XML Schema

L'XML Schéma est une spécification qui définit la structure, le contenu et les contraintes d'un document XML. Il est également appelé XML Schema Definition (XSD).

Les XML Schéma sont utilisés pour valider la structure d'un document XML, c'est-à-dire vérifier si les éléments, les attributs et les valeurs sont conformes à la spécification définie dans le schéma. Ils permettent également de définir les types de données qui peuvent être utilisés dans le document XML, tels que des chaînes de caractères, des nombres, des dates, etc. Un XML Schéma peut contenir les éléments suivants :

- Des éléments et des attributs : définis avec leurs noms, leurs types de données, leurs valeurs par défaut, etc.
- Des contraintes : spécifiées avec des expressions XPath pour vérifier les valeurs et les relations entre les éléments et les attributs.
- Des types de données : définis pour les éléments et les attributs avec des règles de validation spécifiques.
- Des espaces de noms : utilisés pour identifier les éléments et les attributs dans un document XML.
- Les XML Schéma sont souvent utilisés pour la validation des données entrantes dans une application, par exemple pour vérifier si une requête SOAP est correctement formatée et si les données qu'elle contient sont valides.

Les XML Schéma sont largement utilisés dans l'industrie pour la définition de formats de données standardisés, tels que le format de document Office Open XML (OOXML) utilisé dans Microsoft Office. Les applications peuvent interpréter les données XML à l'aide de bibliothèques ou de parseurs XML, tels que DOM ou SAX en Java, SimpleXML en PHP ou ElementTree en Python.

2.10 QUIZZ

1. Quelle est la signification de l'acronyme "HTML" ?
 - a) Hyperlinks and Text Markup Language
 - b) Hypertext Markup Language
 - c) Home Text Markup Language
 - d) Hyperlinking and Text Markup Library
2. Quelle est la balise HTML utilisée pour définir un lien hypertexte ?
 - a) <a>
 - b) <p>
 - c)
 - d) <div>
3. Comment peut-on inclure une image dans une page HTML ?
 - a) En utilisant l'élément <image>
 - b) En utilisant l'élément
 - c) En utilisant l'élément <picture>
 - d) En utilisant l'élément <graphics>
4. Quelle est la balise HTML utilisée pour définir une liste non ordonnée ?
 - a)
 - b)
 - c)
 - d) <list>
5. Comment peut-on cibler un élément avec un identifiant en HTML ?
 - a) En utilisant l'attribut "class"
 - b) En utilisant l'attribut "id"

- c) En utilisant l'attribut "name"
 - d) En utilisant l'attribut "target"
6. Quelle balise HTML est utilisée pour définir un en-tête de section ?
 - a) <h1>
 - b) <header>
 - c) <section>
 - d) <title>
 7. Quelle est la balise HTML utilisée pour définir une ligne de tableau ?
 - a) <tr>
 - b) <td>
 - c) <th>
 - d) <table>
 8. Comment peut-on créer une ligne de séparation horizontale en HTML ?
 - a) En utilisant l'élément <line>
 - b) En utilisant l'élément <rule>
 - c) En utilisant l'élément <hr>
 - d) En utilisant l'élément <separator>
 9. Quelle balise HTML est utilisée pour définir une zone de texte éditable ?
 - a) <input>
 - b) <textarea>
 - c) <select>
 - d) <form>
 10. Comment peut-on inclure du CSS dans un document HTML ?
 - a) En utilisant l'élément <style>
 - b) En utilisant l'attribut "style" sur chaque élément
 - c) En utilisant l'élément <link>
 - d) En utilisant l'élément <script>

Réponses :

1. b) Hypertext Markup Language
2. a) <a>
3. b) En utilisant l'élément
4. a)
5. b) En utilisant l'attribut "id"
6. a) <h1>
7. a) <tr>
8. c) En utilisant l'élément <hr>
9. b) <textarea>
10. a) En utilisant l'élément <style>

Le CSS

"CSS est la magie qui donne vie à l'ensemble de la mise en page d'un site Web."

Dave Shea



3 LE CSS

Le CSS (Cascading Style Sheets) est un langage de balisage utilisé pour décrire la présentation visuelle des pages Web. Il permet de séparer la présentation du contenu, ce qui facilite la maintenance et la mise à jour des pages Web.

En utilisant CSS, vous pouvez définir des styles pour différents éléments HTML, tels que les couleurs, les polices, les marges, les bordures, les arrière-plans et bien plus encore. Vous pouvez également définir des styles pour les différents états d'un élément, tels que le survol, le clic et le focus. Voici un exemple de code CSS pour changer la couleur de fond d'un élément HTML :

```
body {  
  background-color: #f2f2f2;  
}
```

Figure 21. Exemple de code CSS pour changer la couleur de fond

Dans ce code, nous sélectionnons l'élément `body` et nous définissons la couleur de fond en utilisant la propriété `background-color` et la valeur `#f2f2f2`.

Il existe plusieurs façons d'ajouter des styles CSS à une page Web. Vous pouvez les inclure dans une balise `<style>` dans le fichier HTML, ou vous pouvez les mettre dans un fichier CSS séparé et les lier à la page HTML en utilisant une balise `<link>`.

CSS offre également des fonctionnalités avancées telles que les sélecteurs de classe, les sélecteurs d'ID, les sélecteurs d'élément, les sélecteurs de groupe, les sélecteurs d'attribut et les pseudo-classes. Ces sélecteurs vous permettent de cibler des éléments spécifiques de manière plus précise et de leur appliquer des styles différents.

Les feuilles de style en cascade (CSS) utilisent un langage distinct du HTML. Les feuilles de style en cascade vous permettent d'appliquer un style cohérent aux éléments sur toutes les pages de votre site, de sorte que tous les titres, listes et paragraphes aient le même aspect et le même comportement sur toutes les pages d'un site.

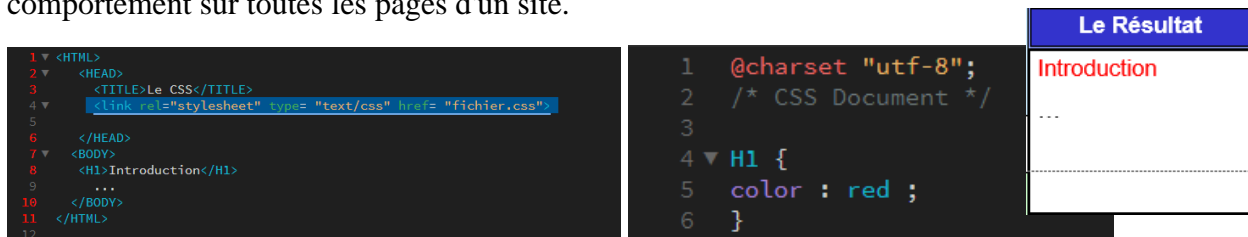


Figure 22. Résultat de l'application d'un style

Un document html est toujours composé d'un corps où se place le contenu et d'un en-tête (entre balise HEAD) dans lequel va se placer le lien entre html et une feuille de style css grâce à la balise `link`. `rel` permet de spécifier une relation, `stylesheet` se rapporte à une feuille de style externe, le type de document lié est le style css mode texte, et `href` définit l'adresse du document css.

Ici, on appelle la feuille *fichier.css* car la structure du document html et sa présentation sont gérées séparément. Dans cette feuille de style, on ajoute une règle de couleur à une balise html existante `h1` qui sert à définir un titre de niveau 1.

On obtient ainsi comme résultat, le titre `introduction` s'écrivant en rouge. Il s'agit d'un sélecteur simple, on a simplement appliqué un style (couleur rouge) à une balise html. Mais il existe d'autres types de sélecteurs pour définir des règles.

Nous nous référons à la syntaxe CSS dans ce cours avant de commencer à travailler avec CSS, nous devons expliquer comment nous nous référons aux différentes parties de la syntaxe CSS dans ce cours. Néanmoins, voici les principes de base : tout le code suivant est ce que nous appelons une règle en CSS :

```
4 ▼ h1{  
5   color:blue;  
6   margin-top:1em;  
7 }
```

Figure 23. Une règle CSS

Cette règle comporte plusieurs éléments, comme suit :

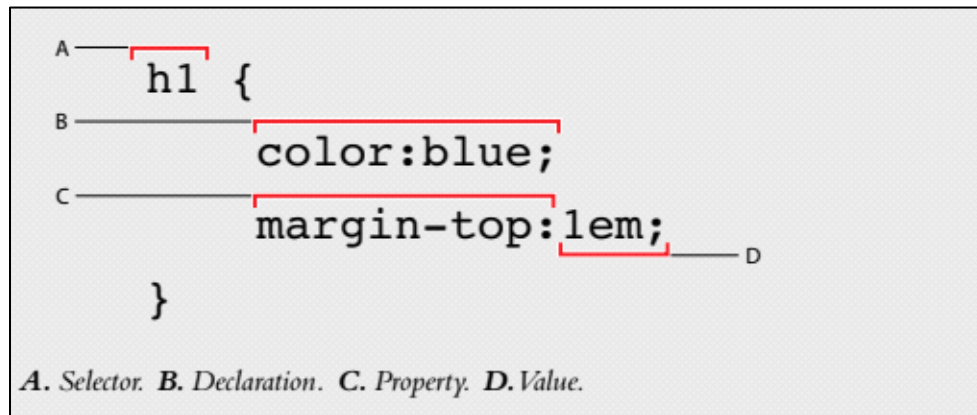


Figure 24. Explication de la règle CSS

Donc si nous vous demandons de changer la valeur "blue" en "red", vous devriez savoir quoi faire. Ou si nous vous demandons de localiser et de remplacer le sélecteur h1 par un sélecteur h2, cela devrait avoir un sens.

Au quotidien, la plupart des concepteurs ne sont cependant pas toujours aussi précis. Par exemple, la règle ci-dessus peut être appelée "style", "règle de style", "règle h1" ou "règle CSS pour h1". De même, comme vous pouvez le voir ci-dessus, le nom officiel de la paire propriété/valeur s'appelle une déclaration. Encore une fois, dans l'usage quotidien, l'utilisation du terme "déclaration" n'est pas courante et la plupart des concepteurs utilisent indifféremment le terme "propriété" ou "propriétés".

3.1.1 Stylisation d'un titre

Pour vous familiariser avec le fonctionnement des CSS, vous allez créer une règle CSS simple qui modifie le style d'un titre dans votre page. Dans votre page index.html, vous avez déjà le contenu "Département d'Informatique" imbriqué dans une balise <h1>. L'une des meilleures façons de commencer à réfléchir au fonctionnement des CSS est d'examiner comment le style par défaut de ce titre est rendu dans le navigateur.

1 Examinez l'en-tête du fichier que vous avez visualisé à la dernière étape de l'exemple précédent. Les instructions de style et de formatage sont fournies par le navigateur. La taille, la couleur et la police sont fournies par le navigateur car les instructions de formatage

exactes ne sont pas spécifiées. Le navigateur sait seulement qu'il s'agit d'un titre. Vous allez redéfinir ce style avec une règle CSS.

2 Dans votre code, localisez la balise <title>, cliquez une fois à la fin de la ligne, après la balise de fermeture, puis appuyez sur retour pour ajouter une nouvelle ligne de code. Tapez ce qui suit : <style type="text/css">

3 Appuyez trois fois sur Retour, puis tapez </style>, Il s'agit d'un élément de style que vous utiliserez pour placer votre règle pour le style de l'élément <h1>. L'élément <style> est imbriqué dans la section *head* de la page, et c'est là que seront placées les règles CSS.

L'élément <style> est imbriqué dans les balises <head> de votre page. En HTML, tout ce qui est imbriqué dans la section <head> n'est pas rendu par le navigateur sur la page. Par exemple, il y a également un élément <title> à l'intérieur de cette section ; ce titre apparaît en haut du navigateur Web, et non sur la page elle-même.

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Untitled Document</title>
<style type="text/css">

</style>

```

Figure 25. Les métas données dans la partie entête

4 Dans la ligne vide située sous la balise d'ouverture <style>, tapez ce qui suit: **h1 {**. Il s'agit de votre sélecteur. Le sélecteur est l'élément HTML que vous voulez styliser, dans ce cas l'élément Heading 1.

5 Appuyez sur la touche Retour, puis sur la touche Tab pour placer votre curseur sous l'accolade. La tabulation est facultative mais elle permet de rendre votre CSS plus lisible. Comme vous le verrez bientôt, le nombre de lignes de cette règle va augmenter et il vaut la peine de garder le code facile à lire.

6 Tapez le code suivant sous le h1 { : **color:purple;** Le mot *couleur* est appelé une propriété dans la syntaxe CSS et le mot *pourpre* est une valeur.

La paire combinée d'une propriété et d'une valeur s'appelle une déclaration. La combinaison de la propriété (couleur) et de la valeur (violet) est souvent appelée règle de style.

```

<title>Untitled Document</title>
<style type="text/css">
h1 {
    color:purple;

</style>

```

Figure 26. Règle CSS à l'intérieur d'un fichier HTML

7 Appuyez à nouveau sur la touche Return et, sur la ligne suivante, tapez } qui est le caractère d'accolade droit. Cela ferme l'accolade que vous avez ajoutée à l'étape 4. Vous avez maintenant trois lignes dans cette règle : `h1 {color:purple; }`

8 Choisissez Fichier > Enregistrer, puis visualisez votre page dans le navigateur. L'en-tête est maintenant de couleur violet clair et vous avez réussi à créer votre première règle CSS. Fermez votre navigateur et revenez à votre éditeur de texte.

9 Dans le fichier HTML, sélectionnez le mot violet et tapez ce qui suit pour la valeur de la couleur : `#800080`. Cette couleur hexadécimale est l'équivalent du violet. Vous pouvez utiliser des couleurs nommées ou des couleurs hexadécimales lorsque vous définissez des couleurs à l'aide de CSS.

Enregistrez votre fichier, puis visualisez-le dans le navigateur. La couleur reste la même. Les couleurs hexadécimales sont une méthode plus courante pour décrire les couleurs.

3.1.2 Couleurs hexadécimales

La couleur, tant en HTML qu'en CSS, est désignée par un code de six caractères précédés d'un dièse. Ce code, appelé code hexadécimal, est le système utilisé pour identifier les couleurs appliquées aux éléments. Vous pouvez reproduire presque toutes les couleurs à l'aide d'un code hexadécimal unique. Par exemple, le code suivant est rouge foncé : `#CC0000`.

La première, la moyenne et la dernière paire de chiffres du code hexadécimal correspondent à des valeurs dans le spectre RVB. Par exemple, le blanc, qui est représenté en RVB par R :255 G:255 B:255, est représenté en HTML par `#FFFFFF` (255|255|255). Un programme comme Photoshop vous permettra de choisir une couleur RVB spécifique dans l'éditeur de couleurs et vous donnera la couleur hexadécimale équivalente à utiliser dans votre code.

Il existe également des références en ligne que vous pouvez utiliser pour localiser ou "mélanger" des couleurs hexadécimales, par exemple :

3.2 LES REGLES CSS

Une règle CSS est une instruction qui définit le style d'un élément HTML ou d'un groupe d'éléments HTML. Elle se compose d'un sélecteur et d'un bloc de déclarations.

```
div {  
  background-color: blue;  
}
```

Figure 27. Sélecteur d'une règle CSS

Dans ce code, le sélecteur est `div`, ce qui signifie que cette règle sera appliquée à tous les éléments `div` sur la page. Le bloc de déclarations est entre les accolades `{ }` et contient une propriété `background-color` avec la valeur `blue`, qui définit la couleur de fond de l'élément.

Il est également possible de combiner plusieurs sélecteurs pour créer une règle plus spécifique. Par exemple, vous pouvez créer une règle qui s'applique uniquement aux éléments div avec la classe exemple :

```
div.example {  
  background-color: blue;  
}
```

Figure 28. Règle avec classe

Dans ce code, le sélecteur est div.example, ce qui signifie que cette règle sera appliquée uniquement aux éléments div qui ont la classe exemple.

Les règles CSS peuvent également inclure des propriétés supplémentaires telles que la couleur du texte, la taille de la police, la marge, le rembourrage, la bordure, etc. Vous pouvez inclure autant de propriétés que nécessaire pour définir le style de l'élément.

Nous allons maintenant vous présenter une autre catégorie de styles CSS, appelée classe. Vous allez également travailler avec l'élément qui sépare et contrôle le contenu en ligne, comme une phrase dans un paragraphe ou un mot individuel dans une phrase.

3.3 COMPRENDRE LES CLASSES CSS

Les classes CSS sont des noms donnés à un ou plusieurs éléments HTML pour leur appliquer un style spécifique. Elles permettent de regrouper des éléments en fonction de leur fonctionnalité, de leur contenu ou de leur style, et de leur appliquer un ensemble de propriétés CSS commun. Voici un exemple de définition de classe CSS :

```
.example-class {  
  color: red;  
  font-size: 18px;  
  font-weight: bold;  
}
```

Figure 29. Une classe CSS

Dans ce code, nous avons créé une classe CSS nommée .example-class qui applique les propriétés color, font-size et font-weight. Pour appliquer cette classe à un élément HTML, vous devez ajouter l'attribut class à la balise et lui attribuer la valeur example-class, comme ceci :

```
8  
9   <p class="example-class">Ceci est un exemple de texte avec la classe example-class</p>  
10
```

Cela appliquera les propriétés définies dans la classe CSS à l'élément <p> avec la classe example-class. Il est possible d'attribuer plusieurs classes à un élément HTML en séparant les noms de classe par des espaces. Par exemple, si vous avez une classe example-class et une classe

another-class, vous pouvez appliquer les deux classes à un élément en utilisant l'attribut class comme ceci :

```
8
9 <p class="exemple-class another-class">Ceci est un exemple de texte avec deux classes</p>
10
```

Cela appliquera les propriétés définies dans les deux classes à l'élément <p>. L'utilisation de classes CSS permet de maintenir un code CSS organisé et facile à gérer, en regroupant les éléments qui ont un style similaire sous une même classe. Cela rend également le code HTML plus lisible et plus facile à comprendre, car il indique clairement quels éléments ont quelles fonctions ou styles.

3.4 LE SPAN EN CSS

En CSS, span est un sélecteur d'élément qui permet de cibler tous les éléments présents dans la page HTML et d'appliquer un style spécifique à ces éléments.

Les éléments sont des éléments HTML en ligne qui sont utilisés pour délimiter un texte ou une section de texte spécifique à laquelle un style CSS peut être appliqué. Voici un exemple de code CSS qui cible tous les éléments et applique une couleur de texte rouge :

```
4 span {
5   color: red;
6 }
7
```

Dans ce code, le sélecteur est span, ce qui signifie que toutes les balises de la page HTML seront ciblées et se verront appliquer la propriété CSS color avec la valeur red.

Il est également possible d'utiliser des sélecteurs de classe et des sélecteurs d'ID avec le sélecteur span pour cibler des éléments spécifiques. Par exemple, vous pouvez créer une règle CSS qui cible uniquement les éléments qui ont la classe highlight :

```
3
4 span.highlight {
5   background-color: yellow;
6   font-weight: bold;
7 }
8
```

Figure 30. Règle avec Span

Dans ce code, le sélecteur est span.highlight, ce qui signifie que cette règle ne sera appliquée qu'aux éléments qui ont la classe highlight. Cette règle définit une couleur de fond jaune et une épaisseur de police en gras pour ces éléments.

En résumé, le sélecteur span permet de cibler tous les éléments dans une page HTML, tandis que les sélecteurs de classe et d'ID permettent de cibler des éléments spécifiques en fonction de leur attribut class ou id.

Les sélecteurs de balises sont fréquemment utilisés, mais ils ne peuvent être appliqués qu'aux éléments HTML. Lorsque vous voulez donner un style à quelque chose qui ne correspond pas directement à une balise, par exemple, changer la couleur d'un seul mot dans un paragraphe, les balises HTML standard ne sont pas une bonne option. Dans ce cas, vous pouvez utiliser un sélecteur de classe, qui est une règle CSS que vous pouvez appliquer à un nombre quelconque d'éléments sur une page. Les sélecteurs de classe peuvent être nommés de différentes façons, mais vous devez choisir des noms qui décrivent ce qu'ils font. Par exemple, vous pouvez nommer les sélecteurs de classe caption, imageborder, ou redtext.

1 Placez votre curseur sur la ligne située immédiatement sous l'accolade de fermeture de la règle h1, puis tapez ce qui suit :

```
3
4 ▼ .violet {
5   couleur: violet;
6 }
7
```

Notez le point au début du sélecteur de classe. Le texte qui suit le point est le nom de la classe. Vous pouvez utiliser le nom de votre choix, mais le point est nécessaire au début pour l'identifier en tant que classe. La règle est la même que dans le dernier exemple, mais ici le sélecteur n'est pas l'élément h1. Le nom de la classe peut être ce que vous voulez, mais il doit comporter le point au début pour l'identifier comme une classe. Ensuite, vous allez appliquer cette classe au texte afin de lui donner un style violet. Pour ce faire, vous utiliserez une balise HTML .

2 Dans le paragraphe de la balise <body>, localisez le texte, cliquez une fois à sa gauche, puis tapez :

```
<p>Ce texte est <span>mis en évidence</span> grâce à la balise span.</p>
```

La balise vous permet de définir la partie d'un paragraphe que vous souhaitez styliser.

3 Cliquez à droite du texte et ajoutez une balise de fermeture span . Votre code devrait ressembler à ceci : <p>text autre text

Enregistrez votre fichier. Si vous deviez prévisualiser la page dans le navigateur, vous ne verriez aucun changement.

La balise en HTML est une balise vide ; elle ne fait rien par elle-même et doit être associée à un style. La balise définit le début et la fin de l'endroit où le style sera appliqué dans le paragraphe sera appliqué dans le paragraphe, mais elle n'applique pas le style par elle-même et ne définit pas le style.

4 Fermez le navigateur et revenez à votre éditeur de texte. Repérez la balise d'ouverture que vous avez insérée avant le texte. Cliquez une fois après le mot span mais avant la parenthèse >, puis tapez ce qui suit : **class="purple"**

Le code devrait maintenant se lire text .

3.5 FEUILLES DE STYLE INTERNES ET EXTERNES

Les feuilles de style internes sont des règles CSS contenues directement dans un document, à l'aide de la balise <style>. L'intégralité de la feuille de style est contenue dans les balises <style> ouvrante et fermante.

Les feuilles de style externes sont des règles CSS enregistrées dans un document distinct portant l'extension de fichier .css. Avec les feuilles de style internes, les règles CSS s'appliquent uniquement au HTML du document actuel. Par exemple, si vous avez un site Web de 20 pages et que vous utilisez des feuilles de style internes, vous devrez créer une feuille de style distincte dans chacune des pages. Une modification du style nécessiterait la mise à jour des styles internes dans chacune des 20 pages distinctes.

Les feuilles de style externes placent toutes les règles CSS d'un site dans un seul document. Vous pouvez joindre le fichier .css à un nombre illimité de pages HTML. Cela offre une plus grande souplesse. Si une règle de style est modifiée dans la feuille de style externe, tous les paragraphes du site sont modifiés en une seule étape. Vous allez créer une feuille de style externe, puis l'attacher à une nouvelle page HTML.

3.5.1 Création d'une feuille de style externe

Pour créer une feuille de style CSS externe, vous devez créer un fichier séparé qui contiendra toutes les règles CSS pour votre site Web. Voici les étapes à suivre :

1. Créez un fichier avec l'extension .css. Vous pouvez utiliser n'importe quel éditeur de texte pour cela, tel que Notepad, Sublime Text, Atom ou Visual Studio Code. Ajoutez vos règles CSS dans ce fichier, en utilisant la même syntaxe que vous utiliseriez dans une balise <style> dans le code HTML. Voici un exemple simple :

```
4 body {
5   background-color: #f2f2f2;
6   font-family: Arial, sans-serif;
7   font-size: 16px;
8 }
9
10 h1 {
11   color: #333;
12   font-size: 28px;
13   font-weight: bold;
14 }
15
16 p {
17   color: #666;
18   line-height: 1.5;
19 }
20
```

Figure 31. Feuille de style externe

2. Enregistrez le fichier avec un nom significatif, tel que styles.css. Assurez-vous que le fichier est enregistré dans le même dossier que votre fichier HTML.
3. Liez votre fichier CSS externe à votre page HTML en ajoutant une balise <link> dans l'en-tête de votre fichier HTML. Voici un exemple :


```
1 <html>
2 <head>
3   <title>Mon site web</title>
4   <link rel="stylesheet" href="styles.css">
5 </head>
6 <body>
7   <h1>Bienvenue sur mon site web</h1>
8   <p>Ceci est un paragraphe de texte.</p>
9 </body>
10 </html>
```

Figure 32. Comment lire le fichier CSS externe

Dans ce code, nous avons ajouté une balise `<link>` à la section `<head>` de notre document HTML, qui lie notre fichier CSS externe à notre page HTML en utilisant l'attribut `href` pour spécifier l'emplacement de notre fichier CSS. Une fois que vous avez créé votre feuille de style externe et lié votre fichier CSS à votre page HTML, toutes les propriétés de style que vous avez définies dans votre fichier CSS seront appliquées.

3.6 POSITIONNEMENT ET FLUX EN CSS

Le positionnement et les flux sont deux concepts importants en CSS qui déterminent comment les éléments HTML sont disposés sur une page Web.

Le positionnement en CSS permet de contrôler l'emplacement des éléments sur la page en utilisant différentes propriétés de positionnement telles que "position", "top", "bottom", "left" et "right". Par exemple, en utilisant la propriété "position: absolute", on peut positionner un élément à un endroit précis par rapport à un parent contenant. Les flux, quant à eux, décrivent la façon dont les éléments sont placés les uns par rapport aux autres sur la page en utilisant des propriétés telles que "display", "float" et "clear". Le flux par défaut des éléments est appelé "flux normal" ou "flux en ligne" et suit l'ordre de lecture de gauche à droite. En utilisant la propriété "float", on peut retirer un élément du flux normal et le placer à droite ou à gauche de son conteneur.

Il est important de comprendre ces concepts pour pouvoir créer des mises en page efficaces et réactives. En utilisant judicieusement les propriétés de positionnement et de flux, on peut créer des designs intéressants et adaptatifs.

3.6.1 Le Flux normal

Le flux normal en CSS est le comportement par défaut des éléments HTML dans une page Web. Il suit l'ordre de lecture de gauche à droite et de haut en bas, ce qui signifie que les éléments sont positionnés l'un après l'autre sur une ligne horizontale et que les lignes suivantes sont placées sous la ligne précédente.

Le flux normal est géré par les propriétés "display" et "position" en CSS. Les éléments en "display: block" prennent toute la largeur disponible et sont empilés les uns sous les autres, tandis que les éléments en "display: inline" sont positionnés les uns à côté des autres. Cependant, il est important de noter que les éléments en "display: inline-block" sont également inclus dans le flux normal, car ils ont à la fois les propriétés "block" et "inline". Le flux normal peut être modifié en utilisant des propriétés telles que "float", "clear" et "vertical-align". Par exemple, en

utilisant la propriété "float: left", on peut retirer un élément du flux normal et le placer à gauche de son conteneur.

```

23 <div class="conteneur">
24 <div class="normalA">
25   Bloc A
26 </div>
27 <div class="normalB">
28   Bloc B
29 </div>
7 .normalA {
8   width:150px;
9   height:150px;
10  background-color:red;
11  border:1px solid black;
12 }
13 .normalB {
14  width:250px ;
15  height:100px ;
16  background-color:green;
17  border:1px solid black;
18 }

```

Figure 33. Exemple d'un flux normal en CSS

3.6.2 Le Flux flottant

Le flux flottant en CSS est un comportement qui permet à un élément de se déplacer vers la gauche ou la droite d'un conteneur jusqu'à ce qu'il touche le bord de celui-ci ou qu'il rencontre un autre élément flottant. Cela est généralement accompli en utilisant la propriété "float".

Lorsqu'un élément est défini en "float: left" ou "float: right", il est retiré du flux normal et positionné le plus à gauche ou le plus à droite possible de son conteneur, respectivement. Les autres éléments se réorganisent autour de l'élément flottant, en remplissant l'espace laissé vacant. Les éléments flottants sont souvent utilisés pour créer des mises en page de style magazine où le texte et les images sont positionnés de manière dynamique. Cependant, il peut y avoir des problèmes de disposition et d'alignement lorsqu'on utilise des éléments flottants, surtout si la taille des éléments n'est pas uniforme.

Pour corriger ces problèmes, on peut utiliser la propriété "clear", qui permet à un élément de forcer l'élément suivant à apparaître sur une nouvelle ligne. Par exemple, en utilisant "clear: left", on peut forcer un élément à apparaître sur une nouvelle ligne sous tous les éléments flottants à gauche. En général, le flux flottant en CSS peut être un outil utile pour créer des mises en page flexibles et réactives, mais il est important de comprendre comment il fonctionne et comment l'utiliser correctement pour éviter les problèmes de disposition.

```

28 <div class="conteneur">
29 <div class="flotteA">
30   Boîte A
31 </div>
32 <div class="flotteB">
33   Boîte B
34 </div>
35 </div>
7 .conteneur {
8   width:800px;
9   border:1px solid black;
10 }
11 .flotteA {
12  float:left;
13  width:650px;
14  background-color:yellow;
15  border:1px solid black;
16 }
17 .flotteB {
18  float:left;
19  width:100px;
20  background-color:blue;
21  border:1px solid black;
22 }

```

Figure 34. Le flux flottant en CSS

3.7 UN BREF HISTORIQUE DES TECHNIQUES DE MISE EN PAGE SUR LE WEB

Même si vous allez apprendre à construire votre mise en page à l'aide de styles CSS, vous devez savoir que cela n'a pas toujours été une pratique standard. Lorsque la conception Web s'est développée au milieu des années 90, la seule méthode disponible pour une mise en page

sophistiquée, telle que l'ajout de plusieurs colonnes à une page, consistait à utiliser la balise HTML `<table>`. Le tableau HTML a été conçu à l'origine pour présenter des données dans un format logique, en utilisant des lignes, des colonnes et des cellules.

Les concepteurs ont adopté cet élément de tableau et l'ont utilisé comme base de la structure de leurs pages. À l'époque, cette technique était parfaitement logique : les tableaux étaient le seul outil disponible pour créer le type de conception requis à l'époque. Les concepteurs utilisaient souvent des techniques telles que l'imbrication des tableaux. Par exemple, le code d'une page standard à deux colonnes peut commencer par un tableau composé de trois lignes et de deux colonnes.

| | |
|--------------------|--------------------|
| Ligne 1, Colonne 1 | Ligne 1, Colonne 2 |
| Ligne 2, Colonne 1 | Ligne 2, Colonne 2 |
| Ligne 3, Colonne 1 | Ligne 3, Colonne 2 |

Figure 35. tableau composé de trois lignes et de deux colonnes.

Cependant, comme la première ligne devenait une section d'en-tête, la colonne en cours de définition était gênante. La balise HTML `<colspan>` a permis au concepteur de fusionner les deux cellules.

| | |
|--------------------|--------------------|
| Cellule fusionnée | |
| Ligne 2, Colonne 1 | Ligne 2, Colonne 2 |

Figure 36. Un tableau avec deux cellules fusionnées dans la première ligne.

Dans cette première rangée fusionnée, un concepteur pourrait vouloir une section indépendante à trois colonnes pour un logo et d'autres éléments, tels que la navigation ou une connexion utilisateur. Pour ajouter cette section, le concepteur ajouterait un nouveau tableau (à trois colonnes) dans la rangée supérieure.

Pour donner une structure à ce tableau, le concepteur peut définir une largeur et une hauteur fixes pour le tableau original. Supposons que le concepteur souhaite également une fine bordure noire autour de l'ensemble du tableau. Cependant, la propriété de bordure des tableaux HTML est très basique et ne permet pas l'ajout de couleurs. Une solution courante consiste à insérer le tableau existant dans un autre tableau, qui consisterait en une seule cellule avec une couleur de fond noire. En modifiant le remplissage et la couleur d'arrière-plan et en fusionnant des cellules supplémentaires, le concepteur a pu créer une mise en page basée sur un tableau avec un style de base.

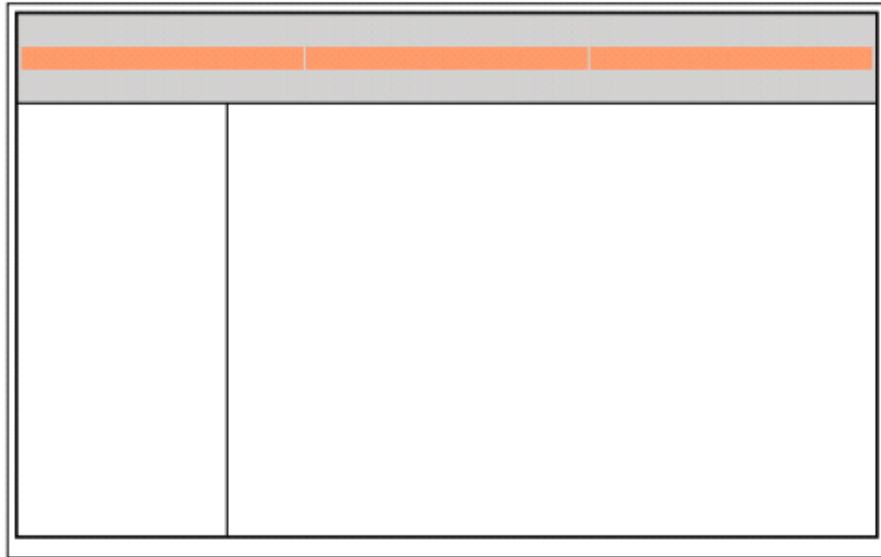


Figure 37. Un modèle pour la mise en page basée sur des tableaux

Cet examen de la mise en page Web est pertinent aujourd'hui car un grand nombre de sites Web ont été construits et continuent de l'être en utilisant la méthode des tableaux. Les CSS ont remplacé l'utilisation des tableaux pour la mise en page, mais le processus est lent et progressif. Les tableaux ont l'avantage d'être rétro compatibles avec les anciens navigateurs, mais cet avantage a diminué au fur et à mesure que les utilisateurs passent à des navigateurs plus récents.

Les mises en page CSS n'étaient pas non plus bien prises en charge (voire pas du tout) dans les premiers navigateurs Web, de sorte que les concepteurs Web n'étaient pas vraiment incités à abandonner les techniques de mise en page sur tableau au profit de la mise en page CSS. Cependant, l'un des inconvénients des mises en page basées sur des tableaux était la quantité et le type de code requis pour construire une page. La mise en page décrite dans les paragraphes précédents aurait nécessité un code similaire à ce qui suit :

```
28 <table width="799" border="0" cellspacing="1" cellpadding="1">
29 <tr>
30 <td bgcolor="#000000">
31 <table width="800" height="485" border="0">
32 <tr>
33 <td height="81" colspan="2" bgcolor="#CCCCCC">
34 <table width="100%" border="0">
35 <tr>
36 <td bgcolor="#FF9966"> </td>
37 <td bgcolor="#FF9966"> </td>
38 <td bgcolor="#FF9966"> </td>
39 </tr>
40 </table>
41 </td>
42 </tr>
43 <tr><br>
44 <td width="191" bgcolor="#FFFFFF"> </td>
45 <td width="599" bgcolor="#FFFFFF"> </td> </td>
46 </tr>
47 </table>
48 </td>
49 </tr>
50 </table>
```

Figure 38. Code de la mise en page

Il s'agit d'une mise en page relativement simple, sans contenu ni navigation. Cependant, la définition de la relation entre tous les différents éléments est très confuse, et elle nécessite plusieurs lignes de code.

Rappelez-vous que l'un des principaux objectifs de CSS était de séparer le style de la structure du HTML. Dans le code du tableau ci-dessus, notez que les valeurs de largeur et de hauteur, ainsi que la couleur d'arrière-plan et quelques autres valeurs, sont intégrées dans le code HTML. Bien que cette pratique était inévitable avant les CSS, vous pouvez désormais définir ces valeurs à l'aide des CSS. L'élément tableau HTML revient peu à peu à sa fonction initiale, qui est de présenter des données, et non d'être utilisé pour la mise en page.

3.8 APERÇU DES OPTIONS DE MISE EN PAGE

Avant de créer une mise en page, vous devez prendre quelques décisions. La première est la largeur de la mise en page. Il existe deux grandes catégories de largeur de mise en page : les *mises en page à largeur fixe* et les *mises en page à largeur variable*.

Les mises en page à largeur fixe sont beaucoup plus courantes : dans une mise en page à largeur fixe, tous les éléments de la page sont imbriqués dans un conteneur qui a une largeur explicite (dans cet exemple, vous utiliserez 960 pixels, mais l'unité de mesure est souvent en ems également). Une mise en page à largeur fixe est utile pour le concepteur car elle permet de positionner de manière fiable les différents éléments de mise en page (tels que les en-têtes, les barres latérales et les pieds de page). Elle fournit également une structure fiable pour les éléments, tels que la largeur d'un paragraphe sur une page ou le placement des images.

Les mises en page flexibles sont appelées ainsi car elles sont conçues pour s'adapter à la largeur de la fenêtre du navigateur. Ce style de mise en page est utile lorsque les utilisateurs ont des résolutions d'écran différentes, ce qui rend impossible la création d'une mise en page à largeur fixe ayant le même aspect sur tous les écrans. Une mise en page flexible correctement conçue peut s'adapter automatiquement à la fenêtre du navigateur de l'utilisateur.

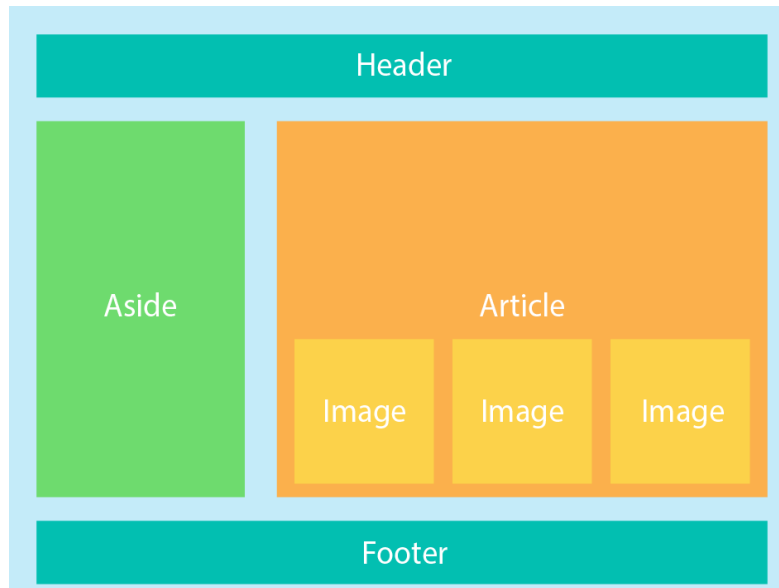


Figure 39. Une mise en page flexible

Les mises en page flexibles s'adaptent aux changements de taille de la fenêtre du navigateur. On pourrait dire que les mises en page flexibles sont plus appropriées pour le Web. Après tout, il ne s'agit pas d'un document imprimé : le texte et les images d'une page Web peuvent se réajuster. Maintenant que les appareils mobiles représentent une part importante des navigateurs Web, une mise en page flexible pourrait être mieux adaptée à ces nouvelles interfaces qu'une mise en page à largeur fixe. Les mises en page à largeur variable sont beaucoup plus difficiles à réaliser. Le concepteur doit prendre davantage de décisions et envisager davantage d'options.

3.9 COMPRENDRE LES <DIV>

Les divs en HTML (Division Element) sont des éléments de balisage utilisés pour diviser une page Web en sections logiques ou groupes de contenu. Les divs sont souvent utilisés comme conteneurs pour d'autres éléments HTML, tels que des images, du texte, des formulaires ou d'autres éléments de mise en forme.

Les divs sont créés en utilisant la balise HTML <div>, qui est généralement placée dans le code HTML avec une classe ou un identifiant. Une classe ou un identifiant peut être utilisé pour appliquer des styles CSS spécifiques à ce div, ou pour cibler le contenu de ce div à l'aide de JavaScript.

```

<body>
  <div class="container">
    <h1>Titre de la page</h1>
    <div class="content">
      <p>Ceci est un exemple de paragraphe dans un div.</p>
    </div>
    <div class="content">
      <p>Ceci est un autre exemple de paragraphe dans un div.</p>
    </div>
  </div>
</body>
  <style>
    .container {
      width: 80%;
      margin: 0 auto;
      padding: 20px;
      background-color: #fff;
      border: 1px solid #ccc;
      border-radius: 5px;
    }
    .content {
      font-size: 18px;
      line-height: 1.5;
      margin-bottom: 20px;
    }
  </style>

```

Figure 40. Création des DIV

Dans cet exemple, nous avons créé deux divs avec la classe "content" pour contenir deux paragraphes, et un div avec la classe "container" pour contenir les deux divs "content" et un titre de page. Nous avons également appliqué des styles CSS différents à chaque classe pour les distinguer visuellement.

3.10 QUIZZ

1. Que signifie CSS ?
 - a) Creative Style Sheet
 - b) Cascading Style Sheet
 - c) Content Style Sheet
2. Comment insérer du CSS dans une page HTML ?
 - a) Avec la balise <script>
 - b) Avec la balise <style>
 - c) Avec la balise <link>
3. Comment sélectionner tous les éléments de type <p> dans une page HTML avec CSS ?
 - a) p {}
 - b) .p {}
 - c) #p {}
4. Comment modifier la couleur de fond d'un élément avec CSS ?
 - a) background-color
 - b) color
 - c) font-color
5. Comment ajouter une bordure à un élément avec CSS ?
 - a) border
 - b) border-style
 - c) border-color
6. Quelle est la signification de l'acronyme "CSS" ?
 - a) Cascading Style Sheets
 - b) Creative Style Sheets
 - c) Colorful Style Sheets
7. Quelle propriété CSS est utilisée pour définir la couleur de fond d'un élément ?
 - a) background-color
 - b) color
 - c) font-family
 - d) font-size
8. Comment peut-on inclure du CSS dans un document HTML ?
 - a) En utilisant l'élément <style>
 - b) En utilisant l'attribut "style" sur chaque élément
 - c) En utilisant l'élément <link>
 - d) En utilisant l'élément <script>
9. Comment peut-on cibler un élément avec une classe en CSS ?
 - a) En utilisant le symbole "." suivi du nom de la classe
 - b) En utilisant le symbole "#" suivi du nom de la classe

- c) En utilisant le symbole ">" suivi du nom de la classe
 - d) En utilisant le symbole "+" suivi du nom de la classe
10. Quelle propriété CSS est utilisée pour définir la taille d'un élément ?
- a) width
 - b) height
 - c) size
 - d) scale

Réponses :

1. b) Cascading Style Sheet
2. b) Avec la balise <style>
3. a) p { }
4. a) background-color
5. a) border
6. a) Cascading Style Sheets
7. a) background-color
8. a) En utilisant l'élément <style>
9. a) En utilisant le symbole "." suivi du nom de la classe
10. a) width

Le PHP

"PHP est un langage incroyable qui vous permet de faire des choses incroyables avec seulement quelques lignes de code."



Rasmus Lerdorf, créateur de PHP

4 LE PHP

Les sites Web statiques ont plusieurs limites en termes de fonctionnalités et d'interactivité par rapport aux sites Web dynamiques. Voici quelques-unes des limites les plus courantes :

1. Contenu statique : Les sites Web statiques affichent généralement du contenu fixe qui ne change pas en fonction des actions de l'utilisateur. Par conséquent, ils ne peuvent pas offrir une expérience utilisateur personnalisée ou des fonctionnalités interactives telles que la recherche en temps réel.
2. Maintenance difficile : Les sites Web statiques peuvent être difficiles à maintenir, car chaque modification apportée au contenu doit être effectuée manuellement en éditant le code HTML. Cela peut prendre du temps et être difficile pour les personnes qui ne sont pas des développeurs Web professionnels.
3. Limites de conception : Les sites Web statiques peuvent avoir des limites en termes de conception, car les mises en page et les styles doivent être définis manuellement dans le code HTML. Cela peut limiter la créativité et rendre difficile la création de sites Web modernes et esthétiques.
4. Pas d'interaction : Les sites Web statiques ne permettent pas aux utilisateurs d'interagir avec le contenu, ce qui peut être un obstacle pour les entreprises qui ont besoin de recueillir des commentaires ou de fournir des outils d'analyse.
5. Difficulté à intégrer des bases de données : Les sites Web statiques ont des difficultés à intégrer des bases de données ou des applications de gestion de contenu, car cela nécessite souvent une programmation personnalisée.

En somme, bien que les sites Web statiques soient plus simples à mettre en place et à héberger, ils ont des limites en termes de fonctionnalités et d'interactivité par rapport aux sites Web

dynamiques. Suite à ces limites et inconvénients des sites statiques, il est évidemment nécessaire de faire se diriger vers la conception des sites Web dynamiques.

4.1 LES SITES WEB DYNAMIQUES

Les sites Web dynamiques sont des sites Web qui permettent aux utilisateurs d'interagir avec le contenu et d'obtenir des résultats en temps réel en fonction de leurs actions. Ces types de sites Web sont devenus populaires pour plusieurs raisons :

1. **Interactivité** : Les sites Web dynamiques permettent aux utilisateurs d'interagir avec le contenu. Par exemple, un site Web de commerce électronique peut permettre aux utilisateurs de rechercher des produits, de les ajouter à leur panier et de passer une commande en temps réel.
2. **Personnalisation** : Les sites Web dynamiques permettent de personnaliser le contenu en fonction des données utilisateur. Par exemple, un site Web de médias sociaux peut afficher du contenu personnalisé en fonction des intérêts et des interactions de l'utilisateur.
3. **Maintenance facile** : Les sites Web dynamiques peuvent être mis à jour facilement en utilisant des CMS (systèmes de gestion de contenu) tels que WordPress, Drupal ou Joomla. Ces CMS permettent aux propriétaires de sites Web de modifier facilement le contenu de leur site Web sans avoir besoin de connaissances en programmation.
4. **Fonctionnalités avancées** : Les sites Web dynamiques peuvent offrir des fonctionnalités avancées telles que des graphiques interactifs, des animations et des fonctionnalités de recherche avancées.
5. **Meilleure expérience utilisateur** : Les sites Web dynamiques peuvent offrir une expérience utilisateur plus fluide et plus agréable en raison de leur interactivité et de leur personnalisation.

En résumé, les sites Web dynamiques offrent de nombreux avantages, tels que l'interactivité, la personnalisation, la facilité de maintenance, les fonctionnalités avancées et une meilleure expérience utilisateur. C'est pourquoi ils sont devenus très populaires pour la création de sites Web modernes.

4.2 LANGAGE DU WEB DYNAMIQUE

Le langage le plus couramment utilisé pour les sites Web dynamiques est le langage de programmation PHP. Il est utilisé pour générer du contenu dynamique côté serveur, tel que des pages Web personnalisées, des formulaires de contact, des systèmes de gestion de contenu, des forums de discussion, des sites de commerce électronique, etc. D'autres langages de programmation Web dynamiques couramment utilisés comprennent :

- **JavaScript** : un langage de programmation côté client qui permet d'ajouter de l'interactivité aux pages Web, comme des animations, des effets visuels, des notifications et des fonctionnalités en temps réel.
- **Python** : un langage de programmation polyvalent qui peut être utilisé pour créer des applications Web dynamiques, des systèmes de gestion de contenu et des applications de commerce électronique.

- Ruby : un langage de programmation qui peut être utilisé pour développer des sites Web dynamiques, des applications de commerce électronique et des systèmes de gestion de contenu.
- Java : un langage de programmation orienté objet qui peut être utilisé pour développer des applications Web dynamiques, des applications de commerce électronique et des systèmes de gestion de contenu.

Malgré la diversité des langages du Web dynamique, le PHP reste le langage de programmation le plus couramment utilisé pour le développement de sites Web dynamiques en raison de sa simplicité, de sa popularité, de sa fiabilité, de sa grande communauté de développeurs, de sa compatibilité et de sa disponibilité.

1. Simplicité : PHP est un langage de programmation relativement simple à apprendre, ce qui en fait un choix populaire pour les débutants en programmation Web.
2. Popularité : PHP est largement utilisé dans l'industrie de la programmation Web, avec une part de marché d'environ 80 % pour les sites Web dynamiques.
3. Fiabilité : PHP est une technologie mature et éprouvée qui a été utilisée pour développer de nombreux sites Web populaires et de grande envergure.
4. Communauté : PHP dispose d'une grande communauté de développeurs actifs qui partagent des connaissances et des ressources pour résoudre les problèmes et améliorer la technologie.
5. Compatibilité : PHP est compatible avec de nombreux systèmes d'exploitation, serveurs Web et bases de données, ce qui le rend très flexible.
6. Disponibilité : De nombreux hébergeurs Web prennent en charge PHP, ce qui facilite la mise en ligne de sites Web dynamiques.

Le PHP est un langage de programmation côté serveur, utilisé pour créer des pages Web dynamiques. Le PHP peut être utilisé pour effectuer des tâches telles que la collecte et la validation des données de formulaire, la gestion des sessions utilisateur, l'interaction avec les bases de données et la création de contenu dynamique à partir de modèles. Voici quelques caractéristiques du PHP :

- Le PHP est gratuit et open-source.
- Le PHP est un langage de script, ce qui signifie qu'il est interprété à chaque fois que la page est chargée, plutôt que compilé en code machine.
- Le PHP peut être utilisé avec une grande variété de bases de données, notamment MySQL, Oracle et PostgreSQL.
- Le PHP peut être intégré dans des fichiers HTML, ce qui permet de créer des pages Web dynamiques qui peuvent générer du contenu personnalisé en fonction des données de l'utilisateur.
- Le PHP peut être utilisé pour créer des applications Web complexes, telles que des forums, des sites de commerce électronique et des réseaux sociaux.

En général, pour exécuter des scripts PHP sur un serveur Web, vous devez installer un serveur Web tel qu'Apache, Nginx ou IIS, et un moteur PHP tel que PHP-FPM. Ensuite, vous pouvez créer des fichiers PHP et les placer dans le répertoire approprié sur le serveur, afin qu'ils soient

accessibles via une URL. Le serveur Web exécutera alors le script PHP et renverra le résultat au navigateur de l'utilisateur.

4.3 COMMENT FONCTIONNE UN SITE WEB ?

Un site Web est une collection de fichiers électroniques (code HTML, CSS, JavaScript, images, etc.) stockés sur un serveur Web et accessibles via Internet. Lorsque vous voulez visiter un site Web, vous tapez son adresse dans votre navigateur Web, que ce soit Mozilla Firefox, Internet Explorer, Opera, Safari ou un autre. Mais ne vous êtes-vous jamais demandé comment faisait la page Web pour arriver jusqu'à vous ? Lorsqu'un utilisateur accède à un site Web, voici ce qui se passe :

- Le navigateur Web de l'utilisateur envoie une demande de connexion au serveur Web du site. Cette demande est envoyée en utilisant le protocole HTTP (Hypertext Transfer Protocol) ou HTTPS (HTTP Secure) pour une connexion sécurisée.
- Le serveur Web reçoit la demande de connexion et renvoie une réponse contenant les fichiers nécessaires pour afficher la page Web demandée. Ces fichiers peuvent inclure du code HTML, des images, des scripts, etc.
- Le navigateur Web reçoit les fichiers et les interprète pour afficher la page Web demandée. Le code HTML définit la structure de la page, le code CSS définit la mise en forme et le style, et le code JavaScript permet d'ajouter de l'interactivité à la page.
- Si la page Web contient des éléments interactifs (comme un formulaire de contact), le navigateur peut envoyer des informations au serveur Web en utilisant à nouveau le protocole HTTP ou HTTPS.
- Le serveur Web peut stocker ces informations dans une base de données ou effectuer d'autres actions en fonction des instructions du code PHP ou d'autres langages de programmation utilisés pour développer le site Web.
- Le navigateur Web peut continuer à afficher des pages Web supplémentaires en répétant les étapes ci-dessus, en envoyant des demandes de connexion au serveur Web pour accéder aux pages supplémentaires.

Il faut savoir qu'Internet est un réseau composé d'ordinateurs. Ceux-ci peuvent être classés en deux catégories :

- **Les clients** : ce sont les ordinateurs des internautes comme vous. Votre ordinateur fait donc partie de la catégorie des clients. Chaque client représente un visiteur d'un site Web.
- **Les serveurs** : ce sont des ordinateurs puissants qui stockent et délivrent des sites Web aux internautes, c'est-à-dire aux clients. La plupart des internautes n'ont jamais vu un serveur de leur vie. Pourtant, les serveurs sont indispensables au bon fonctionnement du Web. La plupart du temps, le serveur est dépourvu d'écran : il reste allumé et travaille tout seul sans intervention humaine, 24h/24, 7j/7. Un vrai forçat du travail.

C'est justement là que se fait la différence entre un site statique et un site dynamique. Voyons voir ensemble ce qui change.

4.3.1 Cas d'un site statique :

Lorsque le site est statique, le schéma est très simple. Cela se passe en deux temps :

1. Le client demande au serveur à voir une page Web.
2. Le serveur lui répond en lui envoyant la page réclamée.

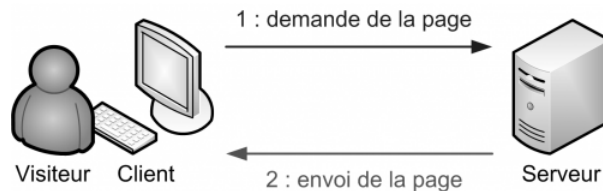


Figure 41. La demande d'une page statique

Sur un site statique, il ne se passe rien d'autre. Le serveur stocke des pages Web et les envoie aux clients qui les demandent sans les modifier.

4.3.2 Cas d'un site dynamique :

Lorsque le site est dynamique, il y a une étape intermédiaire : la page est générée.

1. Le client demande au serveur à voir une page Web.
2. Le serveur prépare la page spécialement pour le client.
3. Le serveur lui envoie la page qu'il vient de générer.

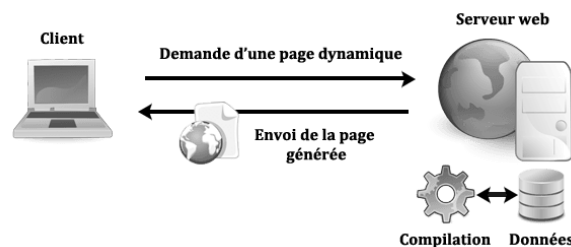


Figure 42. Demande d'une page dynamique

Les sites Web dynamiques sont constamment en train de générer de nouvelles pages en réponse aux demandes des clients. Cette caractéristique confère une certaine vivacité aux sites, car le contenu d'une même page peut varier d'une seconde à l'autre. C'est grâce à ce fonctionnement que certains sites sont capables d'afficher votre pseudonyme sur toutes les pages visitées. En effet, étant donné que chaque page est générée en temps réel, le serveur est en mesure de personnaliser son contenu en fonction des préférences et des goûts du visiteur, ce qui peut inclure l'affichage de son pseudonyme.

4.4 ELEMENTS ESSENTIELS POUR FAIRE FONCTIONNER UN SITE WEB PHP

Pour faire fonctionner un site Web PHP, vous aurez besoin des éléments suivants :

1. Un serveur Web : C'est un logiciel qui s'exécute sur un ordinateur et sert des pages Web aux clients qui les demandent via Internet. Il existe de nombreux serveurs Web disponibles, tels

qu'Apache, Nginx et Microsoft IIS. La plupart des hébergeurs Web proposent un serveur Web dans le cadre de leur service.

2. PHP : C'est le langage de programmation utilisé pour écrire le contenu dynamique du site Web. Vous devrez avoir PHP installé sur le serveur Web pour exécuter des scripts PHP.
3. Un système de gestion de base de données : Si votre site Web nécessite une base de données (comme MySQL, PostgreSQL ou SQLite), vous devrez avoir un système de gestion de base de données installé sur votre serveur Web. Cela vous permettra de stocker et de récupérer des données à partir de la base de données.
4. Un éditeur HTML, CSS et JavaScript : Ce sont les langages utilisés pour créer l'apparence visuelle et la fonctionnalité interactive du site Web. Vous pouvez utiliser n'importe quel éditeur de texte pour écrire du code HTML, CSS et JavaScript, mais il existe de nombreux éditeurs spécialisés disponibles.

4.4.1 Le serveur Web

Un serveur Web est un logiciel qui s'exécute sur un ordinateur et qui est chargé de servir des pages Web aux clients qui les demandent via Internet. Le serveur Web reçoit les demandes de pages Web des navigateurs Web des clients, puis récupère les fichiers associés à la page Web demandée et les envoie au navigateur pour affichage. Il existe plusieurs serveurs Web populaires, tels qu'Apache, Nginx et Microsoft IIS, qui sont utilisés pour héberger des sites Web de toutes tailles et pour tous les types d'applications Web.

4.4.1.1 Apache

Apache est l'un des serveurs Web les plus populaires et les plus utilisés dans le monde entier. Il a été développé par la fondation Apache et est disponible gratuitement sous licence open-source. Apache est compatible avec plusieurs systèmes d'exploitation, tels que Windows, Linux et macOS, et peut être utilisé pour héberger différents types de sites Web, y compris des sites Web statiques et dynamiques.

Apache est connu pour être fiable, performant et extensible, et offre une grande flexibilité grâce à ses modules et extensions. Il prend en charge de nombreux protocoles, tels que HTTP, HTTPS, FTP et SSL/TLS, et peut être configuré pour répondre aux besoins spécifiques de chaque site Web. En outre, Apache dispose d'une communauté de développeurs très active qui fournissent des mises à jour régulières pour corriger les bugs et améliorer les performances.

4.4.2 Le langage PHP

PHP est un langage de script côté serveur, ce qui signifie que le code PHP est exécuté sur le serveur Web avant que la page ne soit envoyée au navigateur Web du client. Il peut être utilisé pour interagir avec des bases de données, créer des sessions utilisateur, envoyer des e-mails, traiter des formulaires Web et effectuer de nombreuses autres tâches.

4.4.3 Le système de gestion de base de données (SGBD)

Un système de gestion de base de données (SGBD) est un logiciel qui permet de stocker, d'organiser et de gérer des données de manière structurée. Dans le contexte des sites Web, un SGBD est utilisé pour stocker et gérer les données du site, telles que les informations des utilisateurs, les articles de blog, les produits de commerce électronique, etc.

Les SGBD les plus couramment utilisés pour les sites Web sont MySQL, PostgreSQL, Oracle et Microsoft SQL Server. Ils sont conçus pour gérer de grands volumes de données et pour offrir des fonctionnalités telles que la sauvegarde et la restauration des données, la réplication, la sécurité des données, etc.

Les SGBD sont souvent utilisés en conjonction avec un langage de programmation côté serveur, comme PHP, pour fournir une interaction dynamique avec les données stockées. Les données sont stockées dans des tables, qui peuvent être interrogées et modifiées à l'aide de requêtes SQL. Les SGBD permettent également de stocker des données dans des formats tels que JSON et XML, qui sont utilisés pour les échanges de données entre les différents systèmes.

4.4.3.1 *Mysql*

MySQL est un système de gestion de base de données relationnelles (SGBDR) open source très populaire, qui est largement utilisé pour les applications Web. Il est conçu pour stocker, organiser et récupérer des données à partir de tables structurées. MySQL est un logiciel gratuit et est distribué sous la licence publique générale GNU.

MySQL peut être utilisé avec de nombreux langages de programmation côté serveur, tels que PHP, Python, Ruby, etc. Il est compatible avec de nombreux systèmes d'exploitation, y compris Windows, Linux et macOS. Les avantages de MySQL incluent la facilité d'utilisation, la fiabilité, la rapidité et la flexibilité.

MySQL est utilisé pour stocker une grande variété de données, telles que les informations d'utilisateur pour les sites Web, les données de commerce électronique, les données de blog, etc. Il est également souvent utilisé pour les applications Web qui nécessitent une manipulation de données en temps réel, telles que les applications de suivi en temps réel, les systèmes de réservation en ligne, etc.

4.4.3.2 *Editeur de code*

Un éditeur HTML est un logiciel qui permet de créer, modifier et publier des pages Web au format HTML. Il existe plusieurs types d'éditeurs HTML, notamment les éditeurs de texte simples, les éditeurs WYSIWYG (What You See Is What You Get) et les environnements de développement intégrés (IDE).

Les éditeurs de texte simples sont des programmes qui permettent d'éditer le code HTML directement. Ils sont souvent utilisés par les développeurs expérimentés qui préfèrent écrire leur code manuellement. Les éditeurs WYSIWYG, quant à eux, sont des programmes qui permettent de créer des pages Web en utilisant une interface graphique conviviale. Ils sont souvent utilisés par les débutants ou les utilisateurs qui ne connaissent pas le langage HTML.

Les IDE sont des environnements de développement qui incluent un éditeur de texte, un débogueur, un compilateur et d'autres outils de développement. Ils sont souvent utilisés par les développeurs professionnels pour créer des applications Web complexes.

Il existe de nombreux éditeurs HTML disponibles, tels que Sublime Text, Atom, Visual Studio Code, Dreamweaver, etc. Chacun a ses propres avantages et inconvénients, et le choix dépendra des préférences de l'utilisateur et de l'objectif du projet Web.

4.4.4 Créer son premier script php

Voici les étapes pour créer un premier script PHP :

1. Ouvrez un éditeur de texte (comme Notepad, Sublime Text, Visual Studio Code, etc.) et créez un nouveau fichier.
2. Dans ce fichier, écrivez la première ligne de votre script PHP, qui doit être `<?php`. Cette ligne indique au serveur Web que le code suivant doit être interprété comme du code PHP.
3. Ajoutez du code PHP entre les balises `<?php` et `?>`. Par exemple, vous pouvez écrire :

```
3 <?php
4 echo "Bonjour, ceci est mon premier script PHP !";
5 ?>
6
```

Cette ligne de code utilise la fonction `echo` pour afficher le message "Bonjour, ceci est mon premier script PHP !" dans la page Web.

4. Enregistrez votre fichier avec une extension `.php` (par exemple, "mon_script.php").
5. Déplacez ce fichier dans le répertoire racine de votre serveur Web.
6. Ouvrez votre navigateur Web et entrez l'URL de votre script PHP (par exemple, "http://localhost/mon_script.php"). Le serveur Web exécutera le script PHP et affichera le résultat dans votre navigateur.

Et voilà, vous avez créé votre premier script PHP !

4.5 PREMIERS PAS AVEC PHP

Dans le début du chapitre, nous avons découvert le principe du fonctionnement de PHP. Ici, nous allons passer au concret et réaliser notre toute première page Web en PHP.

Ne vous attendez pas à un résultat extraordinaire (en fait la page que nous allons créer ne va rien faire de spécial) mais ça va vous permettre de prendre nos marques. Vous allez en particulier comprendre comment on sépare le code XHTML classique du code PHP.

4.6 LES BALISES PHP

Les balises PHP sont utilisées pour insérer du code PHP dans une page Web. Il y a deux types de balises PHP:

Balises ouvrantes standard: `<?php` - elles sont utilisées pour indiquer le début du bloc de code PHP.

Balises courtes: `<?` ou `<?='`, la dernière étant une forme raccourcie de la balise `<?php echo`. Les balises courtes sont parfois désactivées dans les serveurs PHP pour des raisons de sécurité, alors il est préférable d'utiliser les balises ouvrantes standard. A l'intérieur, on écrira donc du code source PHP :

Code : PHP

```
1 <?php /* Le code PHP se met ici */ ?>
```

On peut sans problème écrire la balise PHP sur plusieurs lignes. En fait, c'est même indispensable car la plupart du temps le code PHP fera plusieurs lignes. Cela donnera quelque chose comme :

Code : PHP

```
1 <?php
2 /* Le code PHP se met ici
3 Et ici
4 Et encore ici */
5 ?>
```

La balise PHP que nous venons de découvrir s'insère au milieu du code XHTML comme je vous l'ai dit plus tôt. Pour reprendre l'exemple qu'on a vu au chapitre précédent :

```
1 <html>
2 <head> <title> Mon script PHP </title> </head>
3 <body>
4 //ligne de code HTML
5 <?php
6 //ligne de code PHP
7 ?>
8 //ligne de code HTML
9 ...
10 </body> </html>
```

Figure 43. Script PHP a l'intérieur d'un document HTML

Bien entendu cette page ne fonctionne pas vu que nous n'avons pas encore écrit de vrai code PHP (ce sont juste des balises d'exemple). Tout ce qu'il vous faut retenir ici, c'est que dès que vous voulez mettre du code PHP, hop, vous ouvrez une balise PHP : <?php ?>. On peut placer une balise PHP n'importe où dans le code.

4.7 SYNTAXE DE BASE DE PHP

Voici la syntaxe de base de PHP:

- Balises PHP: <?php et ?>. Tout le code PHP doit être inclus entre ces balises.
- Commentaires: // pour un commentaire sur une ligne et /* */ pour un commentaire sur plusieurs lignes.
- Variables: pour créer une variable, vous devez utiliser le signe \$ suivi d'un nom de variable. Les noms de variable commencent par une lettre ou un underscore (_), suivie de lettres, de chiffres ou d'underscores. Les noms de variable sont sensibles à la casse.
- Types de données: PHP est un langage de programmation dynamique, ce qui signifie que les types de données sont déterminés dynamiquement. Les types de données les plus couramment utilisés en PHP sont les suivants :
- string: une chaîne de caractères
- integer: un nombre entier
- float: un nombre à virgule flottante
- boolean: true ou false
- array: une collection de valeurs

- object: une instance d'une classe
- null: une valeur nulle
- Opérateurs: PHP prend en charge les opérateurs arithmétiques (+, -, *, /, %), les opérateurs de comparaison (==, !=, <, >, <=, >=) et les opérateurs logiques (&&, ||, !).
- Structures de contrôle de flux: les structures de contrôle de flux permettent de modifier le comportement d'un programme. Les structures de contrôle de flux les plus couramment utilisées en PHP sont les suivantes:
 - if/else: permet de conditionner l'exécution d'un bloc de code
 - switch: permet de sélectionner un bloc de code à exécuter en fonction de la valeur d'une expression
 - while: permet d'exécuter un bloc de code tant qu'une condition est vraie
 - for: permet d'exécuter un bloc de code un nombre déterminé de fois
 - foreach: permet de parcourir un tableau

Voici un exemple de code PHP utilisant la syntaxe de base :

```
<?php
// déclaration d'une variable de type string
$message = "Bonjour, comment ça va ?";

// affichage du message
echo $message;

// utilisation d'une structure de contrôle de flux
if (strlen($message) > 10) {
    echo "Le message est trop long";
} else {
    echo "Le message est de la bonne taille";
}
?>
```

Figure 44. Un code PHP avec une syntaxe de base

4.7.1 Les variables en PHP

En PHP, les variables sont utilisées pour stocker des données. Voici quelques informations à savoir sur les variables en PHP :

Déclaration de variables : pour créer une variable en PHP, vous devez utiliser le signe \$ suivi d'un nom de variable. Les noms de variable doivent commencer par une lettre ou un underscore (_), suivie de lettres, de chiffres ou d'underscores. Les noms de variable sont sensibles à la casse. Voici un exemple de déclaration de variable :

```
$nomDeVariable = valeur;
```

Types de données : en PHP, les types de données sont déterminés dynamiquement. Cela signifie que vous n'avez pas besoin de spécifier le type de données lors de la déclaration de la variable. Les types de données les plus couramment utilisés en PHP sont :

- string: une chaîne de caractères

- integer: un nombre entier
- float: un nombre à virgule flottante
- boolean: true ou false
- array: une collection de valeurs
- object: une instance d'une classe
- null: une valeur nulle

Voici un exemple de déclaration de variables avec différents types de données :

```
$nom = "Jean";  
$age = 30;  
$salaire = 2500.50;  
$estEmploye = true;  
$tableau = array("rouge", "vert", "bleu");  
$personne = new Personne();  
$nulleVariable = null;
```

Figure 45. Les différents types de variable

Modification de variables : pour modifier la valeur d'une variable, il suffit d'affecter une nouvelle valeur à la variable. Concaténation de chaînes : pour concaténer des chaînes en PHP, vous pouvez utiliser l'opérateur de concaténation (.). Voici un exemple :

```
$nom = "Jean";  
$message = "Bonjour " . $nom;
```

Variables superglobales: en PHP, il existe plusieurs variables superglobales qui sont accessibles de n'importe où dans le code. Ces variables sont utilisées pour stocker des informations telles que les paramètres de requête HTTP, les informations sur le serveur, etc. Voici quelques exemples de variables superglobales:

- \$_POST: les données envoyées via une requête HTTP POST
- \$_GET: les données envoyées via une requête HTTP GET
- \$_SERVER: les informations sur le serveur

Voici un exemple d'utilisation d'une variable superglobale:

```
if ($_SERVER["REQUEST_METHOD"] == "POST") {  
    // code à exécuter si la méthode de requête est POST  
}
```

4.7.2 Les chaînes de caractères

En PHP, les chaînes de caractères sont définies comme une séquence de caractères, qui peuvent être des lettres, des chiffres, des espaces, des symboles, des signes de ponctuation, etc. Voici quelques informations à savoir sur les chaînes de caractères en PHP :

Définition de chaînes de caractères : Les chaînes de caractères peuvent être définies à l'aide de guillemets doubles ("...") ou de guillemets simples ('...'). Voici un exemple :

```
$chaine1 = "Ceci est une chaîne de caractères.";
$chaine2 = 'Ceci est une autre chaîne de caractères.';
```

Concaténation de chaînes de caractères : Pour concaténer des chaînes de caractères en PHP, vous pouvez utiliser l'opérateur de concaténation (.)>

Interpolation de chaînes de caractères : PHP permet également d'interpoler des variables dans des chaînes de caractères définies avec des guillemets doubles. Pour ce faire, vous pouvez utiliser la syntaxe `${nom_de_variable}`. Voici un exemple :

```
$nom = "Jean";
$message = "Bonjour ${nom}, comment vas-tu ?";
```

Longueur d'une chaîne de caractères : Pour connaître la longueur d'une chaîne de caractères en PHP, vous pouvez utiliser la fonction `strlen()`.

Recherche de sous-chaînes : Pour rechercher une sous-chaîne dans une chaîne de caractères en PHP, vous pouvez utiliser la fonction `strpos()`. Cette fonction renvoie la position de la première occurrence de la sous-chaîne dans la chaîne principale. Voici un exemple :

```
$chaine = "Ceci est une chaîne de caractères.";
$position = strpos($chaine, "une");
```

Extraction de sous-chaînes : Pour extraire une sous-chaîne d'une chaîne de caractères en PHP, vous pouvez utiliser la fonction `substr()`. Cette fonction prend en paramètre la chaîne principale, la position de départ et la longueur de la sous-chaîne à extraire.

Remplacement de sous-chaînes : Pour remplacer une sous-chaîne par une autre dans une chaîne de caractères en PHP, vous pouvez utiliser la fonction `str_replace()`. Cette fonction prend en paramètre la sous-chaîne à remplacer, la nouvelle sous-chaîne et la chaîne principale. Voici un exemple :

```
$chaine = "Ceci est une chaîne de caractères.";
$nouvelle_chaine = str_replace("une", "deux", $chaine);
```

4.7.3 Les entiers

En PHP, les entiers sont des nombres entiers sans partie décimale. Voici quelques informations à savoir sur les entiers en PHP :

Définition d'un entier : En PHP, vous pouvez définir un entier en utilisant simplement un nombre entier.

Taille des entiers : En PHP, la taille des entiers dépend de la plateforme sur laquelle PHP est installé. Par défaut, PHP utilise des entiers signés sur 32 bits, ce qui signifie que les entiers peuvent être compris entre -2147483648 et 2147483647. Si vous avez besoin de manipuler des entiers plus grands, vous pouvez activer le support des entiers 64 bits en compilant PHP avec le support des entiers 64 bits, ou en utilisant une extension comme GMP.

Opérations mathématiques : En PHP, vous pouvez effectuer des opérations mathématiques sur les entiers, telles que l'addition, la soustraction, la multiplication et la division. Voici quelques exemples :

```
$a = 10;
$b = 5;
$c = $a + $b; // $c contient 15
$d = $a - $b; // $d contient 5
$e = $a * $b; // $e contient 50
$f = $a / $b; // $f contient 2
```

Conversion de chaînes en entiers : En PHP, vous pouvez convertir une chaîne de caractères en entier en utilisant la fonction `intval()`. Cette fonction prend en paramètre la chaîne de caractères à convertir et renvoie l'entier correspondant. Conversion d'entiers en chaînes : En PHP, vous pouvez convertir un entier en chaîne de caractères en utilisant la fonction `strval()`. Cette fonction prend en paramètre l'entier à convertir et renvoie la chaîne de caractères correspondante. Voici un exemple :

```
$nombre = "42";
$entier = intval($nombre); // $entier contient 42
```

```
$nombre = 42;
$chaine = strval($nombre); // $chaine contient "42"
```

4.7.4 Float

En PHP, float est un type de données utilisé pour représenter les nombres à virgule flottante (également appelés nombres décimaux ou nombres à virgule flottante).

Un float peut contenir un nombre décimal, et sa précision dépend de l'architecture du système et de l'implémentation du langage - En PHP, float et double sont synonymes, donc ils peuvent être utilisés indifféremment-. On peut déclarer une variable de type float et on peut également utiliser la fonction `gettype()` pour obtenir le type de données d'une variable, par exemple :

```
$numero = 3.14;
echo gettype($numero); // affiche "double"
```

4.7.5 Boolean

En PHP, le type boolean est un type de données qui représente une valeur booléenne, c'est-à-dire une valeur qui est soit true (vrai) soit false (faux). On peut déclarer une variable de type boolean en utilisant les mots-clés true ou false ou encore en convertissant d'autres types de données en booléen à l'aide de la fonction `boolval()` ou de la fonction de conversion (`bool`).

En outre, dans PHP, il existe également des opérateurs de comparaison qui renvoient des valeurs booléennes, tels que `==` (égal à), `!=` (différent de), `<` (inférieur à), `>` (supérieur à), `<=` (inférieur ou égal à), et `>=` (supérieur ou égal à). Voici quelques exemples de déclaration de variables booléennes en PHP:

```
1 <?php
2 $estVrai = true;
3 $estFaux = false;
4 $unNombre = 0;
5 $unAutreNombre = 10;
6
7 $variableBool = boolval($unNombre); // $variableBool est false car $unNombre est égal à 0
8 $autreVariableBool = (bool)$unAutreNombre; // $autreVariableBool est true car $unAutreNombre est différent de 0
9 ?>
```

Figure 46. Utilisation de type Boolean

4.7.6 Le type array

En PHP, le type array est un type de données qui permet de stocker plusieurs valeurs dans une même variable. Les éléments d'un tableau peuvent être de différents types de données (tels que des chaînes de caractères, des nombres, des booléens, des tableaux imbriqués, etc.) et sont accessibles à l'aide d'un index numérique qui commence à 0.

On peut également ajouter ou supprimer des éléments d'un tableau, ainsi que parcourir un tableau avec une boucle for, foreach, ou encore la fonction `array_walk()`, qui permet d'appliquer une fonction sur chaque élément du tableau. En outre, PHP offre de nombreuses fonctions pour manipuler des tableaux, telles que `count()` pour compter le nombre d'éléments d'un tableau, `array_push()` pour ajouter un élément à la fin du tableau, `array_pop()` pour supprimer le dernier élément du tableau, et bien d'autres. On peut déclarer un tableau en PHP de différentes manières. Voici quelques exemples :

```
1 <?php
2 // Déclaration d'un tableau vide
3 $tableauVide = array();
4 // Déclaration d'un tableau avec des éléments
5 $tableau = array("élément 1", "élément 2", "élément 3");
6 // Déclaration d'un tableau associatif (avec des clés)
7 $tableauAssociatif = array("clé1" => "valeur1", "clé2" => "valeur2", "clé3" => "valeur3");
8 ?>
```

4.7.7 Le type object

Le type object est un type de données qui permet de créer des instances de classes. Une classe est un modèle de données qui définit les propriétés et les méthodes d'un objet. On peut également passer des arguments au constructeur de la classe lors de la création de l'objet, pour initialiser ses propriétés. Une fois qu'un objet a été créé, on peut accéder à ses propriétés et méthodes en utilisant l'opérateur `->`.

On peut également copier un objet avec l'opérateur de clonage `clone`. En outre, PHP permet la définition d'interfaces, qui sont des modèles de méthodes pour les classes, et l'héritage, qui permet à une classe d'hériter des propriétés et des méthodes d'une autre classe. Pour créer une instance d'une classe, on utilise le mot-clé `new` suivi du nom de la classe. Par exemple :


```

1 <?php
2 // Définition de la classe Personne
3 class Personne {
4     public $nom;
5     public $age;
6     public function __construct($nom, $age) {
7         $this->nom = $nom;
8         $this->age = $age;
9     }
10    public function parler() {
11        echo "Bonjour, je m'appelle " . $this->nom . " et j'ai " . $this->age . " ans.";
12    }
13 }
14 // Création d'une instance de la classe Personne avec des arguments
15 $personne = new Personne("Mohamed", 35);
16 // Appel de la méthode parler()
17 $personne->parler(); // affiche "Bonjour, je m'appelle Mohamed et j'ai 35 ans."
18 ?>

```

Figure 47. Utilisation de type object

4.7.8 Le type Null

En PHP, le type null est une valeur spéciale qui représente l'absence de valeur. Une variable de type null ne contient aucune valeur, contrairement aux autres types de données tels que les entiers, les chaînes de caractères, les tableaux, etc. On peut également vérifier si une variable contient une valeur null en utilisant l'opérateur ===, qui compare non seulement la valeur, mais aussi le type de données. Dans ce cas, le code affichera "La variable est nulle." car la variable contient une valeur null.

Il est important de vérifier si une variable contient une valeur null avant de l'utiliser, car cela peut causer des erreurs dans le code. Par exemple, si on essaie d'appeler une méthode sur une variable contenant une valeur null, cela provoquera une erreur de type "Call to a member function X() on null".

En outre, il est possible de convertir une variable contenant une valeur null en un autre type de données en utilisant les fonctions de conversion de type telles que intval(), strval(), ou boolval(). Dans ce cas, la variable sera convertie en la valeur par défaut correspondante pour ce type de données (0 pour les entiers, une chaîne vide pour les chaînes de caractères, false pour les booléens, etc.).

```

1 <?php
2 $variable = null;
3
4 if ($variable === null) {
5     echo "La variable est nulle.";
6 } else {
7     echo "La variable n'est pas nulle.";
8 }
9 ?>

```

4.8 LES OPERATEURS ARITHMETIQUES

En PHP, les opérateurs arithmétiques sont utilisés pour effectuer des opérations mathématiques sur des valeurs numériques. Voici les opérateurs arithmétiques disponibles en PHP :

- + : addition
- - : soustraction
- * : multiplication

- / : division
- % : modulo (reste de la division)
- ** : exponentiation (puissance)

Il est également possible de combiner plusieurs opérateurs arithmétiques pour effectuer des calculs plus complexes, dans cet exemple, on effectue une série d'opérations arithmétiques sur les variables \$a, \$b et \$c. Le résultat final est stocké dans la variable \$resultat. Notez que les parenthèses sont utilisées pour indiquer l'ordre dans lequel les opérations doivent être effectuées.

```

1 <?php
2 $a = 10;
3 $b = 3;
4 // addition
5 $c = $a + $b; // $c contient 13
6 // soustraction
7 $c = $a - $b; // $c contient 7
8 // multiplication
9 $c = $a * $b; // $c contient 30
10 // division
11 $c = $a / $b; // $c contient 3.333...
12 // modulo
13 $c = $a % $b; // $c contient 1
14 // exponentiation
15 $c = $a ** $b; // $c contient 1000
16 $resultat = ($a + $b) * $c / $b % 2; // $resultat contient 0
17 ?>

```

Figure 48. Utilisation des opérateurs arithmétiques

4.9 LES OPERATEURS DE COMPARAISON

En PHP, les opérateurs de comparaison sont utilisés pour comparer deux valeurs et renvoyer un résultat booléen (vrai ou faux) en fonction de la comparaison. Voici les opérateurs de comparaison disponibles en PHP :

- == : égalité (vérifie si les deux valeurs sont égales)
- === : identité (vérifie si les deux valeurs sont égales et du même type)
- != ou <> : inégalité (vérifie si les deux valeurs sont différentes)
- !== : non-identité (vérifie si les deux valeurs sont différentes ou de types différents)
- < : inférieur (vérifie si la première valeur est inférieure à la seconde)
- > : supérieur (vérifie si la première valeur est supérieure à la seconde)
- <= : inférieur ou égal (vérifie si la première valeur est inférieure ou égale à la seconde)
- >= : supérieur ou égal (vérifie si la première valeur est supérieure ou égale à la seconde)

Dans ces exemples, les opérateurs de comparaison sont utilisés pour comparer les valeurs des variables \$a et \$b. En fonction de la comparaison, un message est affiché à l'écran. Notez que pour les opérateurs d'égalité et d'identité, == et !== peuvent entraîner des comportements inattendus lorsqu'ils sont utilisés avec des types de données différents, il est donc recommandé d'utiliser === et !== pour des comparaisons plus précises :

```

1 <?php
2 $a = 10; $b = 3;
3 // égalité
4 if ($a == $b) {
5     echo "Les valeurs sont égales.";
6 } else { echo "Les valeurs sont différentes.";
7 }
8 // identité
9 if ($a === "10") {
10    echo "Les valeurs sont identiques.";
11 } else { echo "Les valeurs sont différentes.";
12 }
13 // inférieur
14 if ($a < $b) {
15    echo "La première valeur est inférieure à la seconde.";
16 } else {echo "La première valeur n'est pas inférieure à la seconde.";
17 }
18 ?>

```

Figure 49. Utilisation des opérateurs de comparaison

4.10 LES OPERATEURS LOGIQUES

En PHP, il existe plusieurs opérateurs logiques qui permettent de combiner des expressions booléennes. Les opérateurs logiques les plus courants sont :

- L'opérateur NOT : représenté par le symbole "!" (point d'exclamation), cet opérateur inverse la valeur d'une expression booléenne. Par exemple, si \$a est égal à false, alors !\$a sera égal à true.
- L'opérateur AND : représenté par le symbole "&&" (deux et commerciaux), cet opérateur renvoie true si toutes les expressions booléennes qu'il relie sont vraies. Par exemple, si \$a est égal à true et \$b est égal à true, alors \$a && \$b sera égal à true.
- L'opérateur OR : représenté par le symbole "||" (deux barres verticales), cet opérateur renvoie true si au moins l'une des expressions booléennes qu'il relie est vraie. Par exemple, si \$a est égal à true et \$b est égal à false, alors \$a || \$b sera égal à true.

Il existe également des opérateurs logiques plus complexes, comme l'opérateur XOR (symbole "^"), qui renvoie true si une et une seule des expressions booléennes qu'il relie est vraie.

4.11 LES CLASSES

En PHP, les classes sont des structures de programmation orientées objet qui permettent d'encapsuler des données et des fonctions dans un objet. Cela permet de structurer son code et de rendre plus facile la gestion de celui-ci. Voici un exemple simple de définition de classe en PHP :

```

<?php
class Person {
    public $name;
    public $age;

    function __construct($name, $age) {
        $this->name = $name;
        $this->age = $age;
    }

    function greet() {
        echo "Hello, my name is " . $this->name . " and I am " . $this->age . " years old.";
    }
}
?>

```

Figure 50. Exemple d'une classe en PHP

Dans cet exemple, la classe s'appelle MaClasse et contient deux propriétés : \$nom et \$age, ainsi qu'une méthode direBonjour(). La méthode utilise les propriétés pour afficher une phrase qui se présente. Pour créer une instance de cette classe, il faut utiliser le mot-clé new :

```
<?php
$person1 = new Person("Mohamed", 35);
$person1->greet();

?>
```

Ensuite, on peut affecter des valeurs aux propriétés et appeler la méthode :

```
<?php
$objet->nom = "Mohamed";
$objet->age = 35;
$objet->direBonjour(); // affiche "Bonjour, je m'appelle Mohamed et j'ai 35 ans."

?>
```

C'est un exemple simple, mais les classes en PHP peuvent être beaucoup plus complexes et comporter des méthodes et des propriétés privées ou protégées, des constructeurs et des destructeurs, etc.

4.12 STRUCTURES DE CONTROLE DE FLUX

En PHP, il existe plusieurs structures de contrôle de flux qui permettent de conditionner l'exécution de certaines parties de code en fonction de l'évaluation d'expressions booléennes ou de la valeur de variables. Les principales structures de contrôle de flux en PHP sont :

4.12.1 If-else

La structure "if-else" : cette structure permet d'exécuter un bloc de code si une expression booléenne est vraie et un autre bloc de code si elle est fausse. Voici un exemple :

```
if ($a > 10) {
    echo "La variable \$a est supérieure à 10.";
} else {
    echo "La variable \$a est inférieure ou égale à 10.";
}
```

Figure 51. La syntaxe de IF-ELSE

4.12.2 Switch-Case

La structure "switch-case" : cette structure permet de tester la valeur d'une variable et d'exécuter un bloc de code en fonction de cette valeur. Voici un exemple :

```
switch ($a) {
    case 1:
        echo "La variable \$a vaut 1.";
        break;
    case 2:
        echo "La variable \$a vaut 2.";
        break;
    default:
        echo "La variable \$a ne vaut ni 1 ni 2.";
        break;
}
```

Figure 52. La syntaxe de SWITCH-CASE

4.12.3 While

La structure "while" : cette structure permet d'exécuter un bloc de code tant qu'une expression booléenne est vraie. Voici un exemple :

```
while ($a < 10) {
    echo "La variable \$a vaut $a.";
    $a++;
}
```

Figure 53. La syntaxe de la boucle WHILE

4.12.4 For

La structure "for" : cette structure permet d'exécuter un bloc de code un nombre déterminé de fois en utilisant une variable d'itération. Voici un exemple :

```
for ($i = 0; $i < 10; $i++) {
    echo "La variable \$i vaut $i.";
}
```

Figure 54. La syntaxe de la boucle FOR

4.12.5 Foreach

La structure "foreach" : cette structure permet d'itérer sur un tableau et d'exécuter un bloc de code pour chaque élément du tableau. Voici un exemple :

```
$tab = array("un", "deux", "trois");
foreach ($tab as $val) {
    echo "La valeur est $val.";
}
```

Figure 55. La syntaxe de la boucle Foreach

4.12.6 do-while

La structure "do-while" : cette structure permet d'exécuter un bloc de code au moins une fois, puis de répéter cette exécution tant qu'une expression booléenne est vraie. Voici un exemple, où

le bloc de code est exécuté une première fois sans vérification de la condition, puis répété tant que la variable \$i est inférieure à 10.

```
$i = 0;
do {
    echo "La variable \$i vaut $i.";
    $i++;
} while ($i < 10);
```

Figure 56. La syntaxe de la boucle DO-WHILE

4.12.7 break

La structure "break" : cette structure permet de sortir d'une boucle (for, while, do-while) ou d'une structure de contrôle de flux (switch) de manière anticipée. Voici un exemple où la boucle for est interrompue lorsque la variable \$i atteint la valeur 5, grâce à l'utilisation de la structure "break":

```
for ($i = 0; $i < 10; $i++) {
    if ($i == 5) {
        break;
    }
    echo "La variable \$i vaut $i.";
}
```

Figure 57. L'utilisation de la fonction BREAK

4.12.8 Continue

La structure "continue" : cette structure permet de passer à l'itération suivante d'une boucle en sautant le reste du bloc de code de l'itération actuelle. Voici un exemple où le bloc de code est exécuté pour les valeurs de \$i de 0 à 4 et de 6 à 9, car le bloc de code est ignoré lorsque \$i vaut 5 grâce à l'utilisation de la structure "continue".

```
for ($i = 0; $i < 10; $i++) {
    if ($i == 5) {
        continue;
    }
    echo "La variable \$i vaut $i.";
}
```

Figure 58. L'utilisation de la fonction CONTINUE

4.13 LES FORMULAIRES EN PHP

Les formulaires en PHP sont utilisés pour collecter des données à partir d'un utilisateur sur un site Web. Ils sont généralement utilisés pour les commentaires, les demandes de contact, les formulaires d'inscription, les enquêtes, etc.

Voici les étapes de base pour créer un formulaire en PHP :

1. Créer un fichier HTML contenant le formulaire. Utilisez la balise <form> pour définir le formulaire et les différents champs d'entrée (input) pour collecter les données.

```
<form action="traitement.php" method="post">
  <label for="nom">Nom :</label>
  <input type="text" name="nom" id="nom">

  <label for="email">Email :</label>
  <input type="email" name="email" id="email">

  <input type="submit" value="Envoyer">
</form>
```

Figure 59. Exemple d'un formulaire

2. Créer un fichier PHP pour traiter les données soumises par le formulaire. Utilisez la méthode \$_POST pour récupérer les données envoyées par le formulaire. Exemple :

```
<?php
  $nom = $_POST['nom'];
  $email = $_POST['email'];

  // traitement des données ici

  // redirection vers une page de confirmation ou d'erreur
?>
```

Figure 60. Récupération des données soumises par le formulaire

3. Traiter les données soumises par le formulaire. Cela peut inclure la validation des données, l'envoi d'un e-mail de confirmation, l'enregistrement des données dans une base de données, etc.
4. Rediriger l'utilisateur vers une page de confirmation ou d'erreur. Exemple :

```
<?php
    if ($donnees_valides) {
        // traitement des données ici

        header('Location: confirmation.php');
        exit;
    } else {
        header('Location: erreur.php');
        exit;
    }
?>
```

Figure 61. Redirection par la fonction HEADER

Notez que la validation des données est importante pour garantir que les données soumises par l'utilisateur sont correctes et cohérentes. Cela peut inclure la vérification des formats d'adresse e-mail, la vérification des champs obligatoires, etc.

4.13.1 Méthodes d'envoi Get et Post

En PHP, les méthodes d'envoi les plus courantes pour les formulaires HTML sont GET et POST.

La méthode GET : est utilisée pour envoyer les données du formulaire dans l'URL. Les données soumises sont visibles dans la barre d'adresse du navigateur et peuvent être stockées en cache par le navigateur ou le serveur Web. Elle est principalement utilisée pour des actions non-critiques, telles que la recherche ou l'accès à une page. Voici un exemple d'utilisation de la méthode GET :

```
<form action="page.php" method="get">
    <input type="text" name="nom">
    <input type="submit" value="Envoyer">
</form>
```

Dans cet exemple, les données soumises par l'utilisateur sont envoyées à la page "page.php" via la méthode GET.

La méthode POST : en revanche, est utilisée pour envoyer les données du formulaire en tant que corps de la requête HTTP. Les données soumises sont cachées à l'utilisateur et ne sont pas stockées en cache par le navigateur ou le serveur Web. Elle est principalement utilisée pour des actions critiques, telles que la création ou la mise à jour de données. Voici un exemple d'utilisation de la méthode POST :

```
<form action="page.php" method="post">
    <input type="text" name="nom">
    <input type="submit" value="Envoyer">
</form>
```


Dans cet exemple, les données soumises par l'utilisateur sont envoyées à la page "page.php" via la méthode POST. En résumé, la méthode GET est utilisée pour des actions non-critiques et la méthode POST est utilisée pour des actions critiques nécessitant un niveau de sécurité plus élevé.

4.13.2 Récupération des paramètres de formulaire en PHP

En PHP, les données envoyées via un formulaire HTML peuvent être récupérées à l'aide des superglobales \$_GET et \$_POST en fonction de la méthode d'envoi utilisée. Voici comment récupérer les données soumises. Récupération des données avec la méthode GET :

```
$nom = $_GET['nom'];  
$email = $_GET['email'];
```

Récupération des données avec la méthode POST :

```
$nom = $_POST['nom'];  
$email = $_POST['email'];
```

Il est important de noter que la récupération des données soumises par l'utilisateur doit être accompagnée d'une vérification et d'une validation adéquates pour garantir que les données sont correctes et ne posent pas de risque de sécurité. Pour éviter les erreurs de "Undefined index" lors de l'accès aux superglobales, vous pouvez utiliser la fonction isset() pour vérifier si une clé existe dans l'array. Par exemple :

```
if(isset($_POST['nom'])) {  
    $nom = $_POST['nom'];  
} else {  
    $nom = '';  
}
```

Il est également possible de récupérer toutes les données envoyées par le formulaire, quel que soit la méthode utilisée, en utilisant la superglobale \$_REQUEST. Cependant, il est recommandé d'utiliser \$_GET et \$_POST en fonction de la méthode d'envoi pour plus de sécurité.

4.14 LES COOKIES

En PHP, les cookies sont utilisés pour stocker des informations sur le navigateur de l'utilisateur. Les cookies sont de petits fichiers texte qui sont stockés sur le disque dur de l'utilisateur et qui peuvent être utilisés pour stocker des informations telles que les préférences utilisateur, les informations de session, les paniers d'achat, etc.

4.14.1 Principe

- Un cookie est un fichier texte créé par un script et stocké sur l'ordinateur des visiteurs d'un site
- Les cookies permettent de conserver des renseignements utiles sur chaque utilisateur, et de les réutiliser lors de sa prochaine visite
- Un cookie est utilisé pour Personnaliser la page d'accueil ou les autres pages du site
- Un cookie est utilisé comme un message personnel comportant par exemple son nom, la date de sa dernière visite, ou tout autre particularité.
- Les cookies étant stockés sur le poste client, l'identification est immédiate et ne concernent que les renseignements qui le concernent
- Pour des raisons de sécurité, les cookies ne peuvent être lus que par des pages issues du serveur qui les a créés.
- Le nombre de cookies qui peuvent être définis sur le même poste client est limité à 20 et la taille de chacun est limitée à 4ko.
- Un navigateur peut stocker un maximum de 300 cookies.
- La date d'expiration des cookies est définie de manière explicite par le serveur Web chargé de les mettre en place.
- Les cookies disponibles sont importés par PHP sous forme de variables identifiées sous les noms utilisés par ces cookies.
- La variable globale du serveur `$_COOKIES` enregistre tous les cookies qui ont été définis

Voici comment créer et lire des cookies en PHP :

- Création d'un cookie :

```
<?php
setcookie("nom_du_cookie", "valeur_du_cookie", temps_expiration, chemin, domaine, sécurisé, httponly);
?>
```

- "nom_du_cookie" : le nom du cookie
- "valeur_du_cookie" : la valeur du cookie
- temps_d'expiration : le temps d'expiration du cookie (en secondes). Si le temps d'expiration n'est pas défini, le cookie expirera à la fin de la session.
- chemin : le chemin sur lequel le cookie est valide (par défaut, le cookie est valide sur tout le site)
- domaine : le domaine sur lequel le cookie est valide (par défaut, le cookie est valide sur le domaine actuel)
- sécurisé : indique si le cookie doit être envoyé uniquement sur une connexion HTTPS (valeur par défaut : false)
- httponly : indique si le cookie doit être accessible uniquement via HTTP (valeur par défaut : false).

- Lecture d'un cookie :

```
<?php
$nom_du_cookie = $_COOKIE['nom_du_cookie'];
?>
```

Il est important de noter que la lecture des cookies n'est pas possible avant la prochaine visite du site. De plus, les cookies peuvent être désactivés dans les paramètres du navigateur, il est donc important de prévoir une alternative si les cookies ne sont pas disponibles.

Enfin, il est important de prendre en compte les risques liés à la sécurité lorsque l'on utilise des cookies. Il est recommandé d'utiliser des cookies sécurisés (protocole HTTPS) et d'utiliser des pratiques de codage sécurisées pour éviter toute vulnérabilité à l'injection de code malveillant ou à la falsification de cookies.

4.15 LES SESSIONS

En PHP, les sessions sont utilisées pour stocker des informations sur le serveur pour un utilisateur donné. Les sessions sont utiles pour stocker des informations de connexion, des préférences utilisateur, des informations de panier d'achat, etc. Les sessions sont stockées sur le serveur et sont associées à un identifiant unique appelé "ID de session", qui est stocké sur le navigateur de l'utilisateur sous la forme d'un cookie. Voici comment utiliser les sessions en PHP :

Une fois que la session est démarrée, vous pouvez stocker des données dans la session à l'aide de l'array superglobal \$_SESSION, on peut accéder à une variable de session et on peut la détruire :

```
<?php
session_start();
$_SESSION['nom'] = 'Mohamed';
$_SESSION['age'] = 35;
$nom = $_SESSION['nom'];
session_destroy();
?>
```

Figure 62. Utilisation des Sessions

Il est important de noter que la session_start() doit être appelée avant tout code HTML ou texte envoyé au navigateur. De plus, pour des raisons de sécurité, il est recommandé de régler les paramètres de session comme l'ID de session, le temps d'expiration de la session, etc. avant de stocker des données dans la session. Enfin, il est important de prendre en compte les risques liés à la sécurité lorsque l'on utilise des sessions. Il est recommandé d'utiliser des pratiques de codage sécurisées pour éviter toute vulnérabilité à l'injection de code malveillant ou à la falsification de sessions.

4.16 INTERFAÇAGE AVEC UNE BASE DE DONNEES

4.16.1 Le langage sql

SQL (Structured Query Language) est un langage de programmation utilisé pour communiquer avec les bases de données relationnelles. Il permet d'effectuer des opérations telles que la création de tables, la sélection, la modification et la suppression de données. SQL est utilisé dans de nombreux domaines tels que les applications Web, les systèmes de gestion de bases de données, les analyses de données et les applications mobiles. Voici quelques exemples de commandes SQL couramment utilisées :

- Créer une table :

```
CREATE TABLE table_name (  
    column1 datatype,  
    column2 datatype,  
    column3 datatype,  
    ...  
);
```

Dans cet exemple, nous créons une table avec les colonnes column1, column2 et column3 et leurs types de données respectifs.

- Insérer des données dans une table :

```
INSERT INTO table_name (column1, column2, column3, ...)  
VALUES (value1, value2, value3, ...);
```

Dans cet exemple, nous insérons des données dans la table en spécifiant les colonnes et leurs valeurs respectives.

- Sélectionner des données d'une table :

```
SELECT column1, column2, ...  
FROM table_name;
```

Dans cet exemple, nous sélectionnons les colonnes column1 et column2 de la table.

- Mettre à jour des données dans une table :

```
UPDATE table_name  
SET column1 = value1, column2 = value2, ...  
WHERE condition;
```

Dans cet exemple, nous mettons à jour les valeurs des colonnes column1 et column2 dans la table en fonction d'une condition donnée.

- Supprimer des données d'une table :

```
DELETE FROM table_name WHERE condition;
```

Dans cet exemple, nous supprimons les données de la table en fonction d'une condition donnée.

Il est important de noter que SQL est un langage de base de données standardisé, mais que chaque système de gestion de base de données peut avoir ses propres extensions et fonctionnalités spécifiques. Il est donc important de se familiariser avec la documentation du système de gestion de base de données utilisé pour connaître les fonctionnalités et les extensions disponibles.

PHP propose de nombreux outils permettant de travailler avec la plupart des SGBDR : Oracle, Sybase, Microsoft SQL Server, PostgreSQL ou encore MySQL. Lorsqu'une base de données n'est pas directement supportée par PHP, il est possible d'utiliser un driver ODBC (pilote standard) pour communiquer avec les bases de données. Php fournit un grand choix de fonctions

permettant de manipuler les bases de données, à savoir : La fonction de connexion au serveur ; La fonction de choix de la base de données ; La fonction de requête ; La fonction de déconnexion

En PHP, il est courant d'interfacer avec une base de données pour stocker et récupérer des informations. La plupart des sites Web dynamiques utilisent une base de données pour stocker des informations telles que des utilisateurs, des produits, des commandes, etc. Voici les étapes pour interagir avec une base de données en PHP :

- Connexion à la base de données :

```
<?php
$serveur = "localhost";
$utilisateur = "nom_utilisateur";
$mot_de_passe = "mot_de_passe";
$base_de_donnees = "nom_base_de_donnees";
$connexion = mysqli_connect($serveur, $utilisateur, $mot_de_passe, $base_de_donnees);
?>
```

Figure 63. Script de connexion avec la base de données

Dans cet exemple, nous utilisons MySQLi, une extension PHP pour interagir avec les bases de données MySQL.

- Exécution d'une requête :

```
$resultat = mysqli_query($connexion, "SELECT * FROM table");
```

Dans cet exemple, nous exécutons une requête SELECTE pour récupérer toutes les données de la table.

- Traitement des résultats :

```
while ($ligne = mysqli_fetch_assoc($resultat)) {
    // traitement de chaque ligne
}
```

Dans cet exemple, nous parcourons chaque ligne du résultat à l'aide d'une boucle while et stockons chaque ligne dans un tableau associatif.

- Fermeture de la connexion :

```
mysqli_close($connexion);
```

Il existe également d'autres méthodes pour exécuter des requêtes SQL, telles que PDO et MySQLi orienté objet, qui offrent des fonctionnalités supplémentaires telles que la préparation de requêtes et la gestion des erreurs.

Il est important de prendre en compte les risques liés à la sécurité lors de l'interaction avec une base de données. Il est recommandé d'utiliser des pratiques de codage sécurisées pour éviter toute vulnérabilité à l'injection de code malveillant ou à la falsification de requêtes. Il est également recommandé d'utiliser des requêtes préparées pour éviter l'injection SQL.

4.16.2 Les requêtes SQL

En PHP, les requêtes SQL sont utilisées pour interagir avec une base de données. Il existe plusieurs méthodes pour exécuter des requêtes SQL en PHP, mais la plus courante est

l'utilisation de la bibliothèque MySQLi. Voici comment exécuter une requête SQL en PHP à l'aide de MySQLi :

- *Connexion à la base de données*
- *Exécution de la requête*

Dans cet exemple, nous exécutons une requête SELECTE pour récupérer toutes les données de la table.

```
$resultat = mysqli_query($connexion, "SELECT * FROM table");
```

Traitement des résultats :

Dans cet exemple, nous parcourons chaque ligne du résultat à l'aide d'une boucle while et stockons chaque ligne dans un tableau associatif.

```
while ($ligne = mysqli_fetch_assoc($resultat)) {  
    // traitement de chaque ligne  
}
```

Il est également possible d'exécuter d'autres types de requêtes SQL tels que INSERT, UPDATE et DELETE.

Exemple d'exécution d'une requête INSERT :

```
<?php  
mysqli_query($connexion, "INSERT INTO table (colonne1, colonne2, colonne3) VALUES ('valeur1', 'valeur2', 'valeur3')");  
?>
```

Exemple d'exécution d'une requête UPDATE :

```
<?php  
mysqli_query($connexion, "UPDATE table SET colonne1='nouvelle_valeur' WHERE id=1");  
?>
```

Exemple d'exécution d'une requête DELETE :

```
mysqli_query($connexion, "DELETE FROM table WHERE id=1");
```

Il est important de noter que l'utilisation de requêtes SQL dynamiques peut poser des problèmes de sécurité tels que l'injection SQL. Pour éviter cela, il est recommandé d'utiliser des requêtes préparées. Les requêtes préparées utilisent des paramètres pour exécuter une requête, ce qui empêche les attaques d'injection SQL en les rendant inutiles.

4.17 LES EN-TÊTES HTTP

En PHP, les en-têtes HTTP peuvent être utilisés pour envoyer des informations supplémentaires avec les requêtes et les réponses HTTP. Les en-têtes HTTP sont des informations qui sont incluses dans la partie supérieure de la demande ou de la réponse HTTP. Ils peuvent être utilisés pour spécifier des informations telles que le type de contenu, la langue, l'encodage et les cookies. Voici comment ajouter un en-tête HTTP dans PHP :

```
header('Content-Type: text/html; charset=utf-8');
```

Dans cet exemple, nous ajoutons un en-tête HTTP Content-Type qui spécifie que le type de contenu est du texte HTML encodé en UTF-8.

Il est également possible d'ajouter plusieurs en-têtes HTTP à la fois en utilisant la fonction `header()` plusieurs fois. Voici quelques exemples d'en-têtes HTTP couramment utilisés :

- **Content-Type** : spécifie le type de contenu de la réponse HTTP, tel que "text/html", "application/json" ou "image/png".
- **Content-Length** : spécifie la longueur du contenu de la réponse HTTP en octets.
- **Cache-Control** : spécifie les directives de cache pour la réponse HTTP, telles que "no-cache" ou "max-age=3600".
- **Set-Cookie** : spécifie un cookie à inclure dans la réponse HTTP.
- **Location** : spécifie l'URL de redirection pour la réponse HTTP.

Il est important de noter que les en-têtes HTTP doivent être envoyés avant la sortie de tout contenu. Si des données ont déjà été envoyées au navigateur avant l'envoi de l'en-tête HTTP, cela peut entraîner des erreurs ou des comportements inattendus.

4.18 LA BIBLIOTHEQUE GD

La bibliothèque GD est une bibliothèque graphique open-source pour PHP. Elle permet aux développeurs PHP de créer et de manipuler des images en utilisant une variété de fonctions et de méthodes. GD prend en charge de nombreux formats d'image courants, notamment JPEG, PNG, GIF, BMP et TIFF.

En utilisant GD, vous pouvez créer des images dynamiques, ajouter du texte à des images, appliquer des filtres et des effets, redimensionner des images, créer des miniatures, générer des graphiques et des diagrammes, et bien plus encore. Cette bibliothèque est souvent utilisée dans des applications telles que la génération de graphiques en temps réel, la création de cartes, les systèmes de gestion de photos, les applications de retouche photo et les systèmes de surveillance de la qualité de l'air, entre autres.

L'utilisation de la bibliothèque GD avec PHP est assez simple. Il suffit de charger la bibliothèque et d'appeler les fonctions appropriées pour manipuler les images. Voici un exemple simple pour créer une image JPEG à partir de zéro en utilisant GD:

```
// création d'une image de 200x200 pixels
$image = imagecreate(200, 200);
```

4.19 CONFIGURATION DE SERVEUR XAMPP

Pour configurer XAMPP avec PHP et MySQL, vous pouvez suivre ces étapes :

1. **Télécharger et installer XAMPP** : Vous pouvez télécharger XAMPP depuis le site Web d'Apache Friends et l'installer sur votre ordinateur.

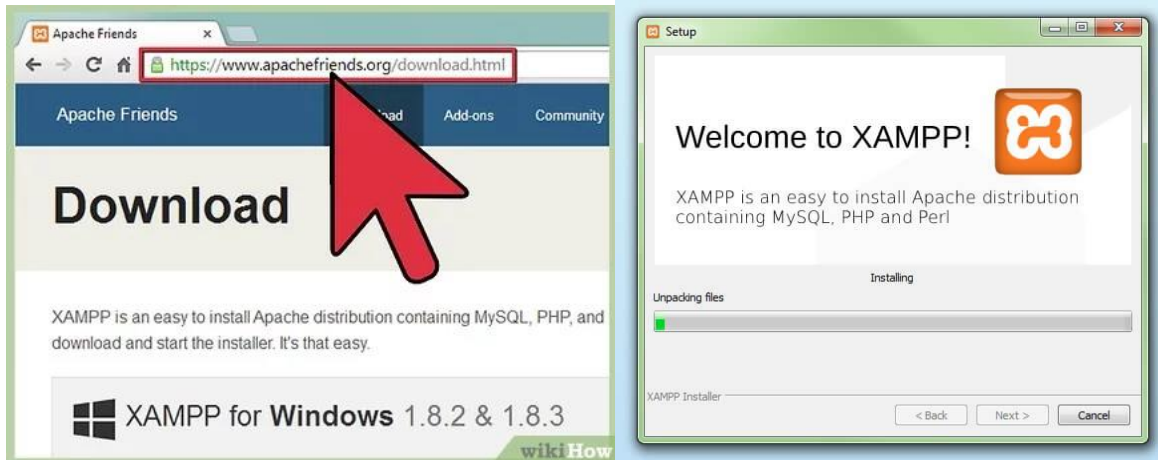


Figure 64. Téléchargement et installation de Serveur XAMPP

2. Démarrer Apache et MySQL : Après avoir installé XAMPP, démarrez les services Apache et MySQL depuis le panneau de contrôle XAMPP.

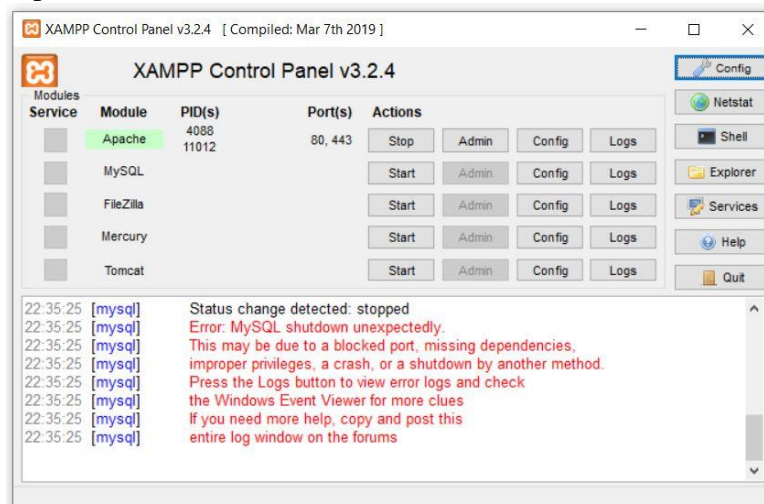


Figure 65. Lancement du Serveur XAMPP

3. Tester votre installation : Ouvrez votre navigateur Web et allez sur "http://localhost". Si tout est configuré correctement, vous devriez voir une page d'accueil XAMPP.

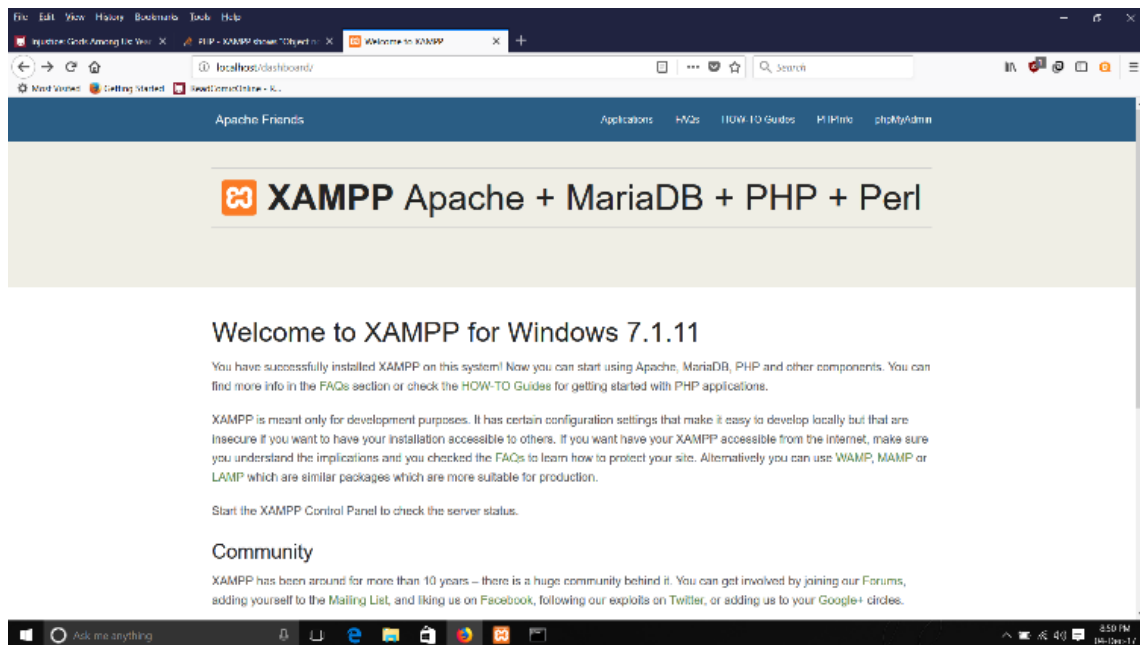


Figure 66. La page d'accueil XAMPP

4. Configurer PHP : Si vous souhaitez modifier la configuration de PHP, vous pouvez le faire en modifiant le fichier "php.ini" dans le dossier XAMPP.

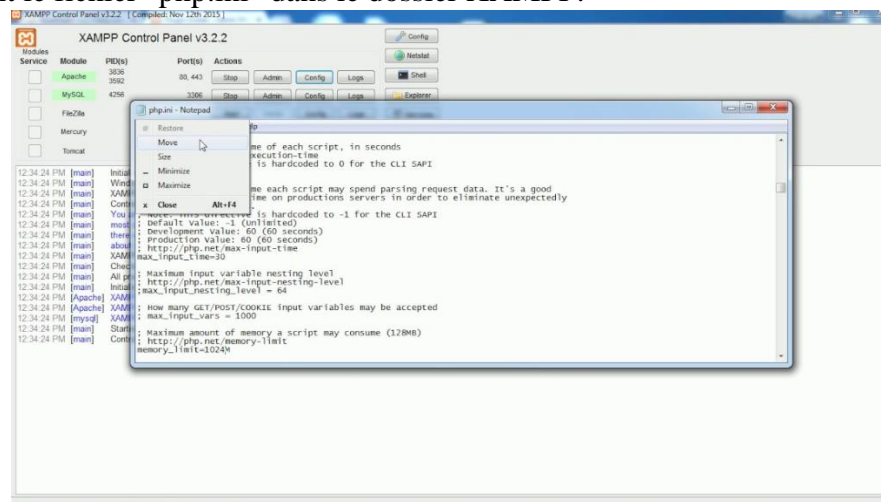


Figure 67. Le fichier php.ini

5. Configurer MySQL : Si vous souhaitez modifier la configuration de MySQL, vous pouvez le faire en modifiant le fichier "my.ini" dans le dossier XAMPP.

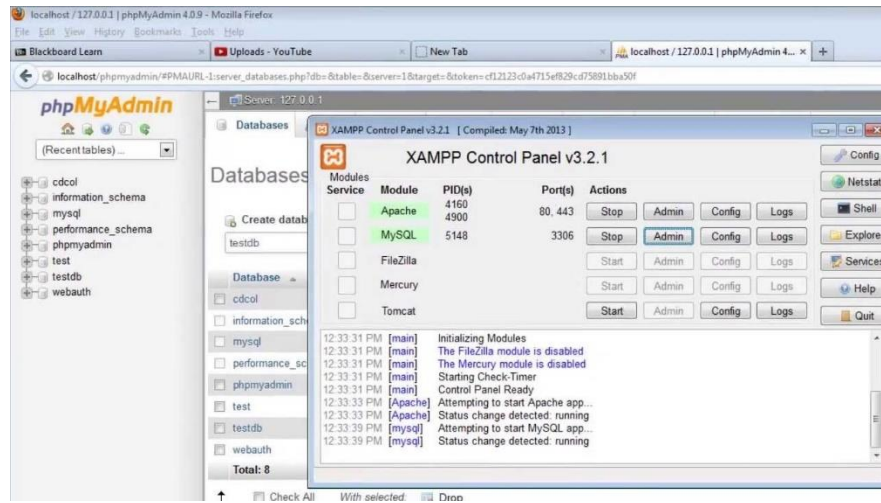


Figure 68. Lancement du PhpMyAdmin

6. Assurez-vous de prendre en compte les mesures de sécurité appropriées en configurant XAMPP pour éviter tout accès non autorisé à votre serveur Web local.

Les Services Web

" Les services Web sont une technologie clé pour l'intégration d'applications d'entreprise et la création de nouvelles solutions métier. "

Anne Thomas Manes



5 LES SERVICES WEB

Un service Web est un système logiciel identifié par un URI, dont les interfaces publiques et les incarnations sont définies et décrites en XML. Sa définition peut être découverte [dynamiquement] par d'autres systèmes logiciels. Ces autres systèmes peuvent ensuite interagir avec le service Web d'une façon décrite par sa définition, en utilisant des messages XML transportés par des protocoles Internet. Un service Web est une application logicielle qui utilise un protocole standardisé pour permettre l'échange de données entre différents systèmes ou applications via un réseau, généralement Internet. Les services Web reposent sur des technologies standardisées telles que XML, SOAP, WSDL et UDDI.

Les services Web permettent aux applications de communiquer et de partager des données, indépendamment de la plate-forme ou du langage de programmation utilisé. Ils permettent également l'intégration de systèmes hétérogènes, de différentes organisations ou entreprises, dans un environnement distribué. Il existe deux types de services Web :

Les services Web basés sur SOAP : ces services utilisent le protocole SOAP (Simple Object Access Protocol) pour l'échange de messages structurés basés sur XML. Le message SOAP contient des informations sur la méthode ou la fonction à exécuter, les paramètres et les valeurs de retour.

Les services Web basés sur REST : ces services utilisent l'architecture REST (Representational State Transfer) pour fournir des services Web simples et légers. Les services REST utilisent les opérations HTTP telles que GET, POST, PUT et DELETE pour accéder aux ressources et fournir des résultats.

Les services Web peuvent être utilisés dans de nombreuses applications, telles que le commerce électronique, la gestion de la chaîne d'approvisionnement, les services bancaires en ligne, etc. Les services Web sont largement utilisés dans les applications d'entreprise et sont une technologie clé pour l'intégration de systèmes distribués.

5.1 BUT DES SERVICES WEB

- Fournir une architecture générale pour les applications réparties sur Internet ;
- Inter-opérables : basé sur des standards ouverts sans composant spécifique à un langage ou un système d'exploitation
- Faiblement couplées : limiter au maximum les contraintes imposées sur le modèle de programmation des différents éléments de l'application par exemple ne pas imposer un modèle objet supportant la montée en charge : par exemple en n'imposant pas un modèle de type RPC, etc.

5.2 L'ARCHITECTURE DES SERVICES WEB

L'architecture des services Web est basée sur un modèle client-serveur, dans lequel le client envoie une demande au serveur, qui répond avec une réponse appropriée. Les services Web sont conçus pour être distribués et indépendants des plates-formes et des langages de programmation.

Trois acteurs principaux définissent le fonctionnement des services Web :

1. Le fournisseur de service (service provider) : définit le service publie sa description dans l'annuaire réalise les opérations
2. L'annuaire (discovery agency) : reçoit et enregistre les descriptions de services publiées par les fournisseurs reçoit et répond aux recherches de services lancées par les clients
3. Le client (service requestor) : obtient la description du service grâce à l'annuaire utilise le service

Et en voici les composants clés de l'architecture des services Web :

- Le protocole de communication : les services Web utilisent généralement le protocole HTTP (Hypertext Transfer Protocol) pour communiquer avec les clients. Les messages échangés entre le client et le serveur sont basés sur XML (eXtensible Markup Language) et encodés dans un format standard appelé SOAP (Simple Object Access Protocol).
- Le registre de services : un registre de services est un répertoire centralisé qui répertorie tous les services Web disponibles sur un réseau. Les registres de services peuvent être publics ou privés, et ils peuvent être accessibles à travers des interfaces Web, des API ou d'autres protocoles standardisés.
- La description de services : une description de services est une spécification détaillée de l'interface et des fonctionnalités d'un service Web. Elle est souvent écrite en utilisant le langage de description de services WSDL (Web Services Description Language), qui décrit les opérations que le service Web peut effectuer, les types de données qu'il accepte et renvoie, et les adresses de points d'accès réseau pour communiquer avec le service.
- La sécurité des services Web : les services Web peuvent être sécurisés à l'aide de différents mécanismes, tels que l'authentification et l'autorisation, le cryptage des messages, le contrôle d'accès basé sur les rôles, etc.

L'architecture des services Web est flexible et extensible, permettant aux développeurs d'intégrer des services Web dans des applications complexes et distribuées.

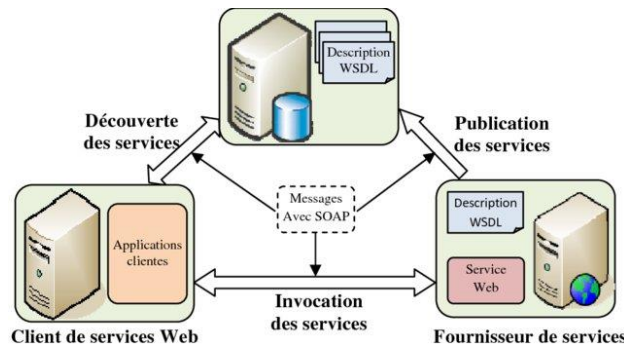


Figure 69. Fonctionnement des services Web selon l'architecture SOA

5.3 LE PROTOCOLE SOAP

SOAP (Simple Object Access Protocol) est un protocole de communication basé sur XML (eXtensible Markup Language) pour échanger des informations structurées entre applications distribuées. Le protocole SOAP définit un format de message standardisé pour représenter des données et des fonctions qui peuvent être appelées à distance.

Le format de message SOAP est basé sur XML et il est utilisé pour encapsuler des données et des appels de fonctions dans un format standardisé. Chaque message SOAP est composé d'un en-tête et d'un corps. L'en-tête contient des informations supplémentaires sur le message, telles que des informations de sécurité, des informations de routage, des informations d'identification, etc. Le corps contient les données et les appels de fonctions qui sont échangés entre les applications.

Le protocole SOAP est utilisé pour créer des services Web basés sur une architecture client-serveur. Les clients peuvent appeler des fonctions distantes sur un serveur en utilisant un message SOAP, et le serveur renvoie une réponse SOAP contenant les résultats de l'appel de fonction. Les services Web basés sur SOAP sont généralement décrits à l'aide du langage de description de services WSDL (Web Services Description Language).

Le protocole SOAP est largement utilisé dans les applications d'entreprise pour permettre l'intégration de systèmes distribués. Il offre un moyen standardisé et indépendant de plate-forme pour échanger des données et des fonctions entre les applications, ce qui facilite l'intégration et la collaboration entre les différents systèmes.

Voici un exemple de fichier SOAP qui illustre la structure d'un message SOAP :

```

1 <soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:examp="http://www.example.com/webservice">
2   <soapenv:Header/>
3   <soapenv:Body>
4     <examp:GetStockPrice>
5       <examp:StockSymbol>IBM</examp:StockSymbol>
6     </examp:GetStockPrice>
7   </soapenv:Body>
8 </soapenv:Envelope>

```

Figure 70. Exemple d'un fichier SOAP

Ce message SOAP encapsule un appel de fonction à distance à un service Web nommé "GetStockPrice" avec un paramètre "StockSymbol" ayant la valeur "IBM". Le message est encapsulé dans un élément "Envelope" qui contient un en-tête vide et un corps qui contient la demande d'appel de fonction.

L'élément "Envelope" est défini dans l'espace de noms XML "http://schemas.xmlsoap.org/soap/envelope/" et l'élément "GetStockPrice" est défini dans un espace de noms personnalisé "http://www.example.com/Webservice". Le langage SOAP utilise des espaces de noms XML pour organiser les éléments et les attributs dans un message SOAP et pour éviter les conflits de noms entre différents éléments.

Le message SOAP peut être envoyé à un service Web via différents protocoles de communication, tels que HTTP, SMTP, FTP, etc. Le service Web traitera la demande d'appel de fonction et renverra une réponse SOAP qui contiendra les résultats de l'appel de fonction.

5.4 LE PROTOCOLE WSDL

WSDL (Web Services Description Language) est un langage de description de services Web qui est utilisé pour décrire les interfaces d'un service Web. Le langage WSDL est basé sur XML (eXtensible Markup Language) et il est utilisé pour décrire les fonctions exposées par un service Web, les paramètres d'entrée et de sortie de chaque fonction, ainsi que les protocoles de communication utilisés pour accéder au service Web.

Le langage WSDL permet aux développeurs de décrire précisément les fonctionnalités d'un service Web, ce qui facilite l'intégration et la collaboration entre les différents systèmes. Un fichier WSDL contient une description complète de l'interface d'un service Web, y compris les opérations, les types de données, les protocoles de communication et les adresses de points d'accès.

Un fichier WSDL peut être utilisé pour générer automatiquement des clients et des serveurs de service Web dans différents langages de programmation, tels que Java, C#, PHP, etc. Les outils de génération de code peuvent utiliser le fichier WSDL pour générer du code qui encapsule les appels de fonctions et la communication avec le service Web, ce qui simplifie le développement de clients et de serveurs de service Web.

Le langage WSDL est essentiel pour les services Web basés sur SOAP, qui utilisent un format de message standardisé pour représenter des données et des fonctions qui peuvent être appelées à distance. Le fichier WSDL permet aux clients de découvrir les fonctions exposées par un service Web et de communiquer avec le service de manière transparente, indépendamment de la plateforme ou du langage de programmation utilisé.

Voici un exemple de fichier WSDL pour un service Web simple qui fournit des informations sur les pays

```
<?xml version="1.0" encoding="UTF-8"?>
<wSDL:definitions xmlns:wSDL="http://schemas.xmlsoap.org/wSDL/"
  xmlns:soap="http://schemas.xmlsoap.org/wSDL/soap/"
  xmlns:tns="http://example.com/countryService"
  targetNamespace="http://example.com/countryService">
  <wSDL:types>
  <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://example.com/countryService">
    <xsd:element name="CountryRequest">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="countryCode" type="xsd:string"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="CountryResponse">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="countryName" type="xsd:string"/>
          <xsd:element name="capital" type="xsd:string"/>
          <xsd:element name="population" type="xsd:int"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </xsd:schema>
</wSDL:types>
</wSDL:definitions>
```

Figure 71. Exemple de fichier WSDL

Ce fichier WSDL définit les éléments suivants :

- Un espace de noms cible pour le service, défini comme `http://example.com/countryService`.
- Un schéma XML qui définit les types de données pour les demandes et les réponses du service.
- Deux messages : un message de demande appelé `getCountryRequest` et un message de réponse appelé `getCountryResponse`.
- Un portType nommé `CountryServicePortType`, qui spécifie l'opération `getCountry` et les messages de demande et de réponse correspondants.

5.5 LE REGISTRE UDDI

Le UDDI (Universal Description, Discovery and Integration) est une spécification de registre de services qui fournit un annuaire pour les services Web. Il est conçu pour permettre la découverte, l'inscription et l'utilisation de services Web par les entreprises et les utilisateurs.

Le registre UDDI contient des informations sur les services Web, notamment leur emplacement, leur interface, leurs conditions d'utilisation et leurs politiques de sécurité. Il permet aux développeurs de services Web de publier des informations sur leurs services et de les rendre disponibles pour les autres utilisateurs, ce qui facilite la découverte et la consommation de ces services. Le registre UDDI est constitué de trois parties principales :

1. La partie blanche (White Pages) : cette partie contient des informations sur l'entreprise qui fournit le service, telles que son nom, son adresse et son numéro de téléphone.

2. La partie jaune (Yellow Pages) : cette partie contient des informations sur les types de services offerts, tels que les catégories et les mots-clés utilisés pour les décrire.
3. La partie verte (Green Pages) : cette partie contient des informations techniques sur les services, telles que leur emplacement, leur interface et leurs conditions d'utilisation.

Le registre UDDI permet aux entreprises de publier leurs services Web et de les rendre disponibles pour les autres utilisateurs qui souhaitent les consommer. Il facilite également la découverte de services Web pertinents en fournissant des informations détaillées sur les services et en permettant aux utilisateurs de rechercher des services en fonction de leur description et de leur emplacement.

Voici un exemple simple d'utilisation d'UDDI :

Supposons que vous soyez une entreprise qui offre des services de traitement de paiement en ligne. Vous voulez rendre votre service facilement accessible aux autres entreprises qui ont besoin de traiter des paiements en ligne. Voici comment vous pouvez utiliser UDDI pour atteindre cet objectif:

1. Créez un compte sur un registre UDDI public, tel que uddi.org.
2. Créez une entreprise dans le registre UDDI, en fournissant des informations telles que votre nom, votre adresse et une description de votre entreprise.
3. Créez une entrée de service dans le registre UDDI, en fournissant des informations telles que le nom du service, sa description et son emplacement.
4. Inscrivez votre service sous une catégorie appropriée, telle que "traitement de paiement en ligne".
5. Publiez votre entrée de service sur le registre UDDI.
6. Maintenant, les autres entreprises peuvent trouver votre service en recherchant des services de traitement de paiement en ligne dans le registre UDDI. Ils peuvent utiliser les informations que vous avez fournies pour se connecter à votre service et commencer à traiter des paiements en ligne avec votre entreprise.

5.6 DEPLOYER WEBSERVICE EN PHP

1. Créer un fichier PHP pour votre service Web : Tout d'abord, créez un fichier PHP qui contiendra votre code de service Web. Ce fichier devra être nommé en fonction de la fonctionnalité qu'il fournit.
2. Écrire le code de votre service Web : Dans le fichier PHP, écrivez le code pour votre service Web. Vous pouvez utiliser les fonctions PHP natives pour effectuer les opérations requises. N'oubliez pas que votre service Web doit accepter les paramètres nécessaires et renvoyer la réponse appropriée.
3. Héberger le fichier PHP : Pour que votre service Web soit disponible en ligne, vous devez héberger le fichier PHP sur un serveur Web. Vous pouvez utiliser des services d'hébergement gratuits ou payants pour cela. Assurez-vous que le serveur dispose des configurations PHP nécessaires pour exécuter votre code.
4. Tester votre service Web : Une fois que votre fichier PHP est hébergé sur un serveur Web, vous pouvez tester votre service Web en appelant l'URL du fichier PHP dans un navigateur Web. Assurez-vous que vous recevez la réponse appropriée.

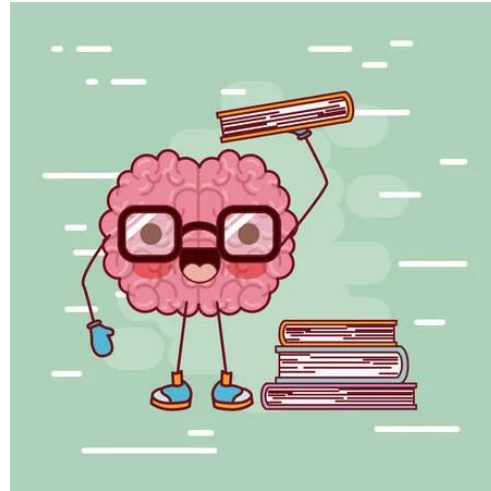
5. Consommer votre service Web : Vous pouvez maintenant consommer votre service Web à partir d'autres applications ou services en utilisant l'URL de votre service Web.

Il est important de noter que la sécurité est un aspect crucial lors du déploiement d'un service Web. Assurez-vous de prendre les mesures nécessaires pour protéger votre service Web contre les attaques malveillantes et les vulnérabilités de sécurité.

Exercices Corrígés

"C'est par la pratique que nous devenons capables"

Aristote



6 EXERCICES CORRIGES

6.1 EXERCICES SUR LE HTML

6.1.1 Exercice 1

Créez une page HTML qui affiche un titre "Mon site Web" et contient les éléments suivants :

1. Un paragraphe de texte qui décrit votre site Web
2. Une liste à puces de vos intérêts
3. Une image qui représente votre site Web

Solution

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Mon site Web</title>
5 </head>
6 <body>
7   <h1>Mon site Web</h1>
8   <p>Mon site Web est un site dédié à partager mes passions et mes intérêts avec le monde entier.</p>
9   <h2>Mes intérêts :</h2>
10  <ul>
11    <li>Voyages</li>
12    <li>Cuisine</li>
13    <li>Photographie</li>
14    <li>Nature</li>
15  </ul>
16  
17 </body>
18 </html>
19
```

Mon site Web

Mon site Web est un site dédié à partager mes passions et mes intérêts avec le monde entier.

Mes intérêts :

- Voyages
- Cuisine
- Photographie
- Nature



Notez que vous devrez remplacer le nom de fichier de l'image dans le code avec le nom de votre propre image. Assurez-vous également que l'image est bien placée dans le même dossier que votre fichier HTML.

6.1.2 Exercice 2

Créez un tableau HTML pour afficher les informations suivantes sur les fruits :

- Nom du fruit
- Couleur du fruit
- Prix du fruit

Incluez au moins trois fruits dans votre tableau.

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Tableau des fruits</title>
5 </head>
6 <body>
7   <h1>Tableau des fruits</h1>
8   <table>
9     <thead>
10    <tr>
11      <th>Nom du fruit</th>
12      <th>Couleur du fruit</th>
13      <th>Prix du fruit</th>
14    </tr>
15  </thead>
16  <tbody>
17    <tr>
18      <td>Pomme</td>
19      <td>Rouge</td>
20      <td>50.00 DA/pièce</td>
21    </tr>
22    <tr>
23      <td>Banane</td>
24      <td>Jaune</td>
25      <td>600.00 DA/kg</td>
26    </tr>
27    <tr>
28      <td>Fraise</td>
29      <td>Rouge</td>
30      <td>300.00 DA/barquette</td>
31    </tr>
32  </tbody>
33 </table>
34 </body>
35 </html>

```

Tableau des fruits

| Nom du fruit | Couleur du fruit | Prix du fruit |
|--------------|------------------|---------------------|
| Pomme | Rouge | 50.00 DA/pièce |
| Banane | Jaune | 600.00 DA/kg |
| Fraise | Rouge | 300.00 DA/barquette |

Dans cette solution, nous avons utilisé les éléments `<table>`, `<thead>`, `<tbody>`, `<tr>`, `<th>` et `<td>` pour créer notre tableau HTML. Les éléments `<thead>` et `<th>` sont utilisés pour créer l'en-tête du tableau, tandis que les éléments `<tbody>` et `<td>` sont utilisés pour créer le corps du tableau. Les balises `<tr>` représentent chaque ligne de notre tableau.

6.1.3 Exercice 3

Créez une liste non ordonnée (UL) qui contient les éléments suivants:

- Pommes
- Bananes
- Oranges
- Fraises

Créez ensuite une liste ordonnée (OL) qui contient les éléments suivants:

- Étape 1: Préchauffer le four à 180°C

- Étape 2: Mélanger les ingrédients dans un bol
- Étape 3: Verser la préparation dans un moule à gâteau
- Étape 4: Enfourner pendant 25 minutes

Solution

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Listes UL et OL</title>
5 </head>
6 <body>
7   <h2>Liste de fruits:</h2>
8   <ul>
9     <li>Pommes</li>
10    <li>Bananes</li>
11    <li>Oranges</li>
12    <li>Fraises</li>
13  </ul>
14
15  <h2>Recette de gâteau:</h2>
16  <ol>
17    <li>Étape 1: Préchauffer le four à 180°C</li>
18    <li>Étape 2: Mélanger les ingrédients dans un bol</li>
19    <li>Étape 3: Verser la préparation dans un moule à gâteau</li>
20    <li>Étape 4: Enfourner pendant 25 minutes</li>
21  </ol>
22 </body>
23 </html>
24
```

Liste de fruits:

- Pommes
- Bananes
- Oranges
- Fraises

Recette de gâteau:

1. Étape 1: Préchauffer le four à 180°C
2. Étape 2: Mélanger les ingrédients dans un bol
3. Étape 3: Verser la préparation dans un moule à gâteau
4. Étape 4: Enfourner pendant 25 minutes

Dans cette solution, nous avons créé une liste non ordonnée (UL) pour les fruits et une liste ordonnée (OL) pour la recette de gâteau. Chaque élément de la liste est créé en utilisant l'élément , qui représente un élément de la liste. Les éléments de la liste non ordonnée sont créés à

l'intérieur de l'élément et les éléments de la liste ordonnée sont créés à l'intérieur de l'élément .

6.1.4 Exercice 4

Créez un lien vers la page d'accueil de Google en utilisant un texte d'ancrage "Google".

Créez un lien vers la page de contact de votre site Web en utilisant un texte d'ancrage "Contactez-nous".

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Liens</title>
5 </head>
6 <body>
7   <h1>Exercice HTML sur les liens</h1>
8
9   <p>Visitez <a href="https://www.google.com">Google</a> pour effectuer une recherche.</p>
10
11  <p>Pour nous contacter, veuillez <a href="contact.html">nous contacter</a>.</p>
12 </body>
13 </html>
```

Exercice HTML sur les liens

Visitez [Google](https://www.google.com) pour effectuer une recherche.

Pour nous contacter, veuillez [nous contacter](#).

Dans cette solution, nous avons créé deux liens. Le premier lien pointe vers la page d'accueil de Google en utilisant l'attribut href avec la valeur "https://www.google.com". Le texte d'ancrage est défini comme "Google" en utilisant l'élément <a>. Le deuxième lien pointe vers la page de contact de notre site Web en utilisant l'attribut href avec la valeur "contact.html", qui est le chemin relatif de la page de contact dans notre site Web. Le texte d'ancrage est défini comme "nous contacter" en utilisant l'élément <a>.

6.2 EXERCICES SUR LE CSS

6.2.1 Exercice 1

Créez une page HTML avec un titre "Mon site Web" et un paragraphe qui contient "Bienvenue sur mon site Web!". Appliquez ensuite les styles CSS suivants:

- Définissez la couleur de fond de la page à "lightblue"
- Centrez le texte du titre
- Appliquez une couleur rouge au texte du paragraphe
- Définissez la police du texte à "Arial, sans-serif"
- Appliquez un espacement de 20 pixels entre le bord de la page et le contenu du paragraphe

Solution:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Mon site web</title>
5   <style>
6     body {
7       background-color: lightblue;
8     }
9
10    h1 {
11      text-align: center;
12    }
13
14    p {
15      color: red;
16      font-family: Arial, sans-serif;
17      padding: 20px;
18    }
19  </style>
20 </head>
21 <body>
22   <h1>Mon site web</h1>
23   <p>Bienvenue sur mon site web!</p>
24 </body>
25 </html>
26
```

Mon site web

Bienvenue sur mon site web!

Ce code est une page HTML qui définit un titre "Mon site Web" à l'aide de l'élément `<title>` dans la section `<head>`. Il contient également une section `<style>` qui définit le CSS pour la page.

Dans le CSS, le fond de la page est défini en bleu clair à l'aide de la propriété `background-color` pour l'élément `<body>`. Le titre de la page est centré à l'aide de la propriété `text-align` pour l'élément `<h1>`. Les paragraphes de la page sont définis en rouge à l'aide de la propriété `color` pour l'élément `<p>`. La police de caractères utilisée pour les paragraphes est Arial ou une police sans-serif similaire, définie à l'aide de la propriété `font-family` pour l'élément `<p>`. Enfin, un remplissage de 20 pixels est ajouté autour des paragraphes à l'aide de la propriété `padding` pour l'élément `<p>`.

Ensuite, la page contient un titre de niveau 1 `<h1>` "Mon site Web" et un paragraphe `<p>` "Bienvenue sur mon site Web!". Le texte "Bienvenue sur mon site Web!" sera affiché en rouge avec une police de caractères Arial ou similaire et aura un remplissage de 20 pixels autour. La page aura également un fond bleu clair et le titre "Mon site Web" sera centré.

6.2.2 Exercice 2

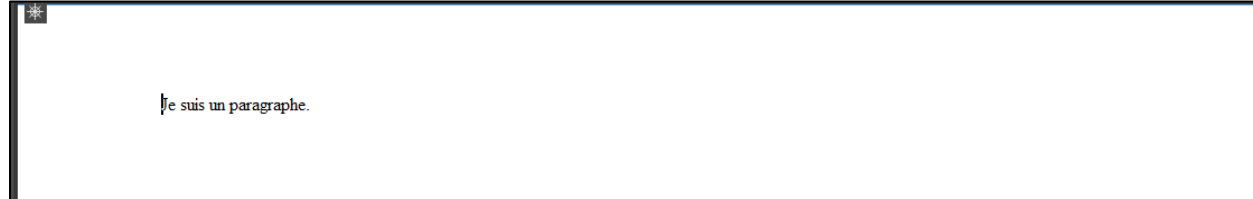
Créez une page HTML avec un titre "Exemple de CSS marges et rembourrage (padding)". Ajoutez un paragraphe contenant le texte "Je suis un paragraphe." Définissez la marge supérieure

et inférieure du paragraphe à 50 pixels et la marge gauche et droite à 100 pixels. Définissez également un remplissage de 20 pixels pour le paragraphe.

Solution

Créez un nouveau fichier HTML avec le nom "index.html" et ajoutez le code suivant:

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Exemple de CSS marges et rembourrage</title>
5  <style>
6      p {
7          margin: 50px 100px;
8          padding: 20px;
9      }
10 </style>
11 </head>
12 <body>
13     <p>Je suis un paragraphe.</p>
14 </body>
15 </html>
16
```



Dans cette solution, nous avons créé un nouveau fichier HTML et ajouté un titre "Exemple de CSS marges et rembourrage" à l'aide de l'élément <title>. Nous avons également ajouté une section <style> à l'en-tête du document pour inclure notre CSS. Nous avons ensuite ajouté un paragraphe contenant le texte "Je suis un paragraphe" à l'aide de l'élément <p>. Dans notre section <style>, nous avons ajouté une règle pour notre paragraphe. Nous avons défini la marge supérieure et inférieure à 50 pixels à l'aide de la propriété margin: 50px 100px; et la marge gauche et droite à 100 pixels. Nous avons également défini un remplissage de 20 pixels à l'aide de la propriété padding: 20px;

Enregistrez le fichier HTML et ouvrez-le dans votre navigateur. Vous devriez voir le paragraphe avec une marge supérieure et inférieure de 50 pixels et une marge gauche et droite de 100 pixels. Le texte du paragraphe aura également un remplissage de 20 pixels.

6.2.3 Exercice 3

Créez une page HTML avec un titre "Exemple de CSS classe et ID". Ajoutez deux paragraphes contenant du texte. Ajoutez une classe "important" au premier paragraphe et un identifiant "intro" au deuxième paragraphe. Définissez la couleur de texte du premier paragraphe à rouge et la couleur de fond du deuxième paragraphe à gris.

Solution :

Créez un nouveau fichier HTML avec le nom "index.html" et ajoutez le code suivant :


```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Exemple de CSS classe et ID</title>
5   <style>
6     .important {
7       color: red;
8     }
9
10    #intro {
11      background-color: gray;
12    }
13  </style>
14 </head>
15 <body>
16   <h1>Exemple de CSS classe et ID</h1>
17   <p class="important">Je suis un paragraphe important.</p>
18   <p id="intro">Je suis un paragraphe d'introduction.</p>
19 </body>
20 </html>
21
```

Exemple de CSS classe et ID

Je suis un paragraphe important.

Je suis un paragraphe d'introduction.

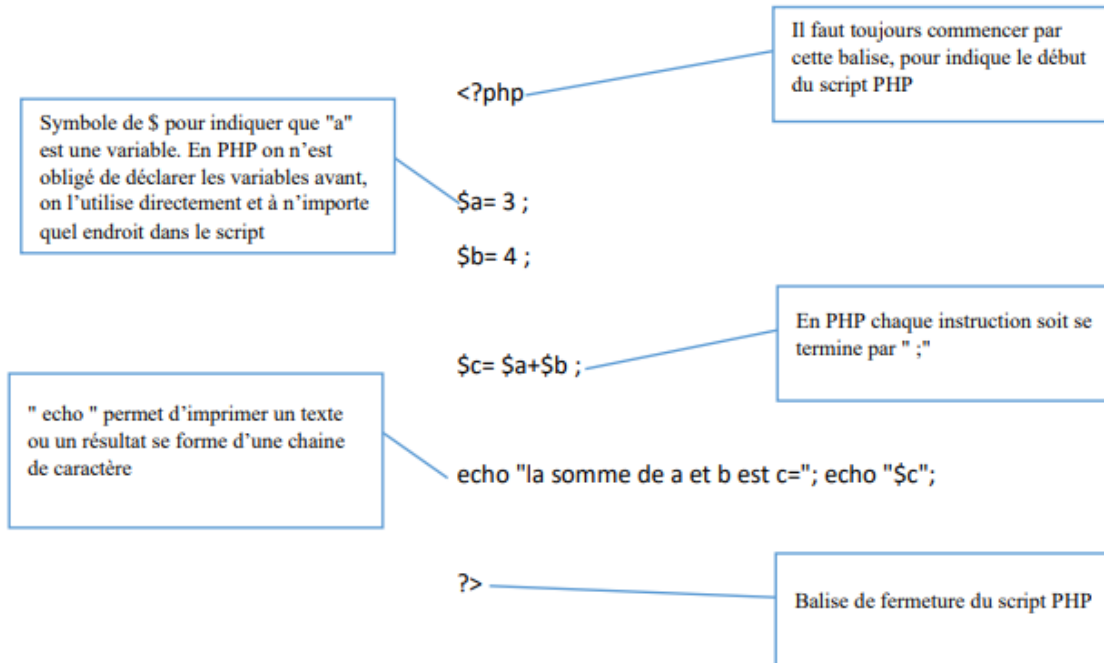
Dans cette solution, nous avons créé un nouveau fichier HTML et ajouté un titre "Exemple de CSS classe et ID" à l'aide de l'élément <title>. Nous avons également ajouté une section <style> à l'en-tête du document pour inclure notre CSS. Nous avons ensuite ajouté deux paragraphes à l'aide de l'élément <p>. Nous avons ajouté une classe "important" au premier paragraphe à l'aide de l'attribut class="important" et un identifiant "intro" au deuxième paragraphe à l'aide de l'attribut id="intro". Dans notre section <style>, nous avons ajouté deux règles. Nous avons défini la couleur de texte du premier paragraphe à rouge à l'aide de la règle .important { color: red; } et la couleur de fond du deuxième paragraphe à gris à l'aide de la règle #intro { background-color: gray; }.

Enregistrez le fichier HTML et ouvrez-le dans votre navigateur. Vous devriez voir les deux paragraphes avec des styles différents. Le premier paragraphe aura une couleur de texte rouge et le deuxième paragraphe aura une couleur de fond gris.

6.3 EXERCISES SUR LE PHP

6.3.1 Exercice 1

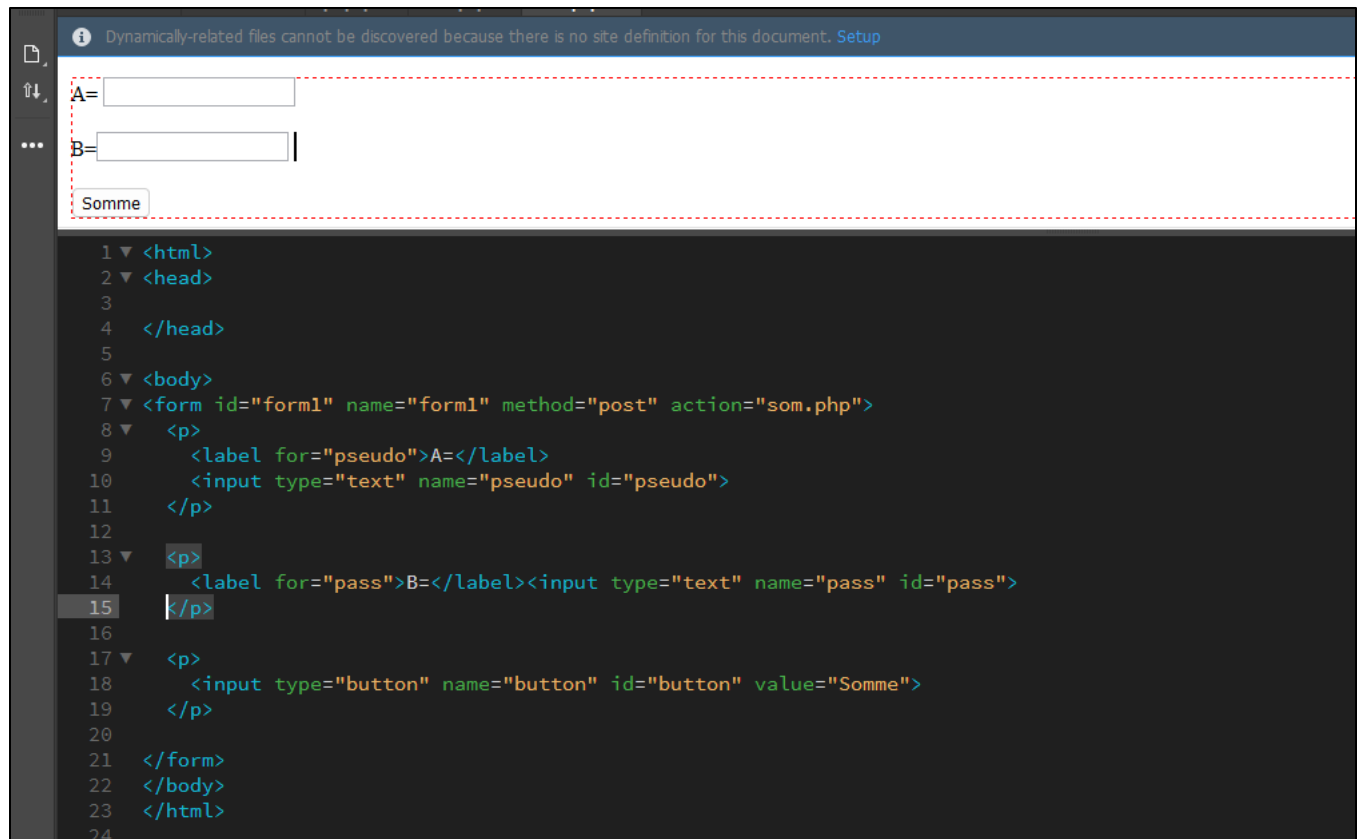
Voici un petit exemple de l'opération de SOMME en PHP



- Ecrire le code source dans un éditeur
- Créer un dossier dans C:\Program Files (x86)\EasyPHP\www, (par exemple TP-PHP) de votre serveur PHP (EasyPHP)
- Enregistrer la page sous le nom somme.php dans le dossier TP-PHP
- Lancer EasyPHP
- Pour exécuter le script, lancer votre navigateur, tapez : localhost/TP-PHP/somme.php

6.3.2 Exercice 2

Faire la somme de deux nombres avec un formulaire



```
1 <html>
2 <head>
3
4 </head>
5
6 <body>
7 <form id="form1" name="form1" method="post" action="som.php">
8 <p>
9 <label for="pseudo">A=</label>
10 <input type="text" name="pseudo" id="pseudo">
11 </p>
12
13 <p>
14 <label for="pass">B=</label><input type="text" name="pass" id="pass">
15 </p>
16
17 <p>
18 <input type="button" name="button" id="button" value="Somme">
19 </p>
20
21 </form>
22 </body>
23 </html>
24
```

6.3.3 Exercice 3

Créez une page PHP qui prend en entrée deux nombres entiers et permet à l'utilisateur de sélectionner une opération arithmétique à effectuer sur ces nombres (addition, soustraction, multiplication ou division). Ensuite, affichez le résultat de l'opération.

Solution :

Créez un nouveau fichier PHP avec le nom "index.php" et ajoutez le code suivant:

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4    <title>Les 4 opérations arithmétiques en PHP</title>
5  </head>
6  <body>
7    <form method="post">
8      <label for="num1">Entrez le premier nombre :</label>
9      <input type="number" name="num1" required><br><br>
10
11     <label for="num2">Entrez le deuxième nombre :</label>
12     <input type="number" name="num2" required><br><br>
13
14     <label for="operation">Sélectionnez une opération :</label>
15     <select name="operation">
16       <option value="addition">Addition</option>
17       <option value="soustraction">Soustraction</option>
18       <option value="multiplication">Multiplication</option>
19       <option value="division">Division</option>
20     </select><br><br>
21
22     <input type="submit" value="Calculer">
23   </form>

```

```

25 <?php
26   if ($_POST) {
27     $num1 = $_POST['num1'];
28     $num2 = $_POST['num2'];
29     $operation = $_POST['operation'];
30
31     if ($operation == 'addition') {
32       $resultat = $num1 + $num2;
33       echo "<p>Le résultat de l'addition est : " . $resultat . "</p>";
34     } elseif ($operation == 'soustraction') {
35       $resultat = $num1 - $num2;
36       echo "<p>Le résultat de la soustraction est : " . $resultat . "</p>";
37     } elseif ($operation == 'multiplication') {
38       $resultat = $num1 * $num2;
39       echo "<p>Le résultat de la multiplication est : " . $resultat . "</p>";
40     } elseif ($operation == 'division') {
41       if ($num2 == 0) {
42         echo "<p>Erreur : division par zéro.</p>";
43       } else {
44         $resultat = $num1 / $num2;
45         echo "<p>Le résultat de la division est : " . $resultat . "</p>";
46       }
47     }
48   }
49   ?>
50 </body>
51 </html>
52

```

Entrez le premier nombre :

Entrez le deuxième nombre :

Sélectionnez une opération :

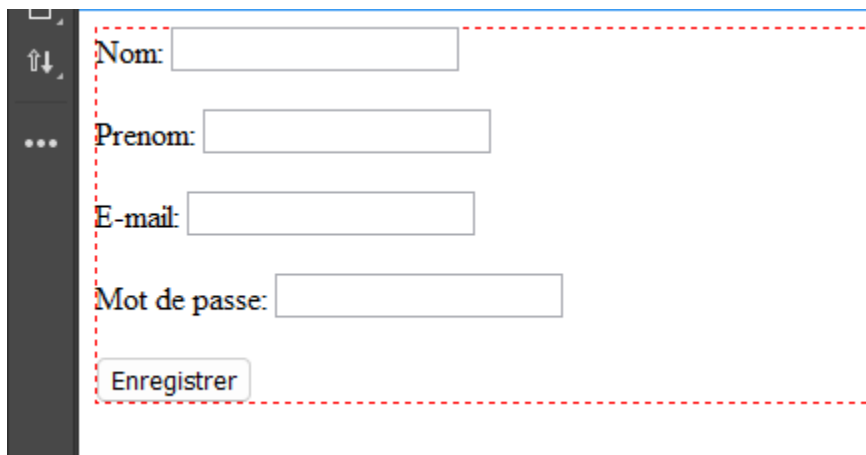
Dans cette solution, nous avons créé un nouveau fichier PHP et ajouté un titre "Les 4 opérations arithmétiques en PHP" à l'aide de l'élément <title>. Nous avons ensuite créé un formulaire avec deux champs numériques pour entrer les nombres et une liste déroulante pour sélectionner une

opération. Lorsque l'utilisateur soumet le formulaire, nous avons récupéré les valeurs de num1, num2 et operation à l'aide de la variable \$_POST. Nous avons ensuite utilisé une structure de test if pour déterminer quelle opération effectuer et afficher le résultat correspondant.

6.3.4 Exercice 4

Enregistrer les informations sur la base de données à travers un formulaire et les récupérer :

1. Création du formulaire



The image shows a web form with the following fields and a button:

- Nom:
- Prenom:
- E-mail:
- Mot de passe:
- Enregistrer

```
9 <form id="form1" name="form1" method="post" action="connex.php">
10 <p>
11 <label>Nom:
12 <input type="text" name="nom" id="nom" />
13 </label>
14 </p>
15 <p>
16 <label>Prenom:
17 <input type="text" name="prenom" id="prenom" />
18 </label>
19 </p>
20 <p>
21 <label>E-mail:
22 <input type="text" name="email" id="email" />
23 </label>
24 </p>
25 <p>
26 <label>Mot de passe:
27 <input type="text" name="mot" id="mot" />
28 </label>
29 </p>
30 <p>
31 <label>
32 <input type="submit" name="button" id="button" value="Enregistrer" />
33 </label>
34 </p>
35 </form>
36
```

2. Le script d'insertion dans la table :

```
1 <?php
2 mysql_connect("localhost", "root", "")or die("cannot connect");
3 mysql_select_db("bdd")or die("cannot select DB");
4
5 $nom= $_POST['nom'];
6 $prenom=$_POST['prenom'];
7 $email= $_POST['email'];
8 $mp= $_POST['mot'];
9
10 $req="INSERT INTO user (name, fname, email, password) VALUES ('$nom', '$prenom', '$email', '$mp)";
11
12 mysql_query($req);
13
14 echo "insertion reussite..";
15
16 ?>
17
```

3. Afficher les informations enregistrer sur la table

```
1 <?php
2 mysql_connect("localhost", "root", "")or die("cannot connect");
3 mysql_select_db("tp")or die("cannot select DB");
4     $sql="SELECT * FROM user";
5     $result=mysql_query($sql);
6     echo '<table width="50%" border="1" cellspacing="0" cellpadding="0">
7     <tr>
8         <td bgcolor="#66CCFF">Id</td>
9         <td bgcolor="#66CCFF">Nom</td>
10        <td bgcolor="#66CCFF">Pr&eacut;nom</td>
11        <td bgcolor="#66CCFF">E-mail</td>
12    </tr>
13    while($row = mysql_fetch_array($result)) {
14
15        $nom = $row["name"];
16        $prenom = $row["fname"];
17        $email = $row["email"];
18        $password=$row["password"];
19        echo "<tr>
20            <td>" ;echo "$name</td>
21            <td>" ;echo "$fname</td>
22            <td>" ;echo "$email</td>
23            <td>" ;echo "$password</td>
24        </tr>";
25    }
26    echo "</table>"; ?>
27
```

6.4 EXERCISES SUR LES SERVICES WEB

6.4.1 Exercice 1

Créez un Web service PHP simple qui prend en entrée une chaîne de caractères et renvoie une réponse JSON contenant la longueur de la chaîne et sa valeur en majuscules.

Solution :

Créez un nouveau fichier PHP avec le nom "Webservices.php" et ajoutez le code suivant :

```
1 <?php
2 if($_SERVER['REQUEST_METHOD'] == "POST"){
3     $input = json_decode(file_get_contents('php://input'), true);
4
5     $string = $input['string'];
6
7     $response = [
8         'length' => strlen($string),
9         'uppercase' => strtoupper($string)
10    ];
11
12    header('Content-Type: application/json');
13    echo json_encode($response);
14 }
```

Dans cette solution, nous avons créé un fichier PHP qui vérifie si la méthode HTTP utilisée est POST. Si c'est le cas, nous avons récupéré les données JSON de la requête à l'aide de `file_get_contents('php://input')` et nous les avons converties en un tableau associatif en utilisant `json_decode(..., true)`. Ensuite, nous avons extrait la chaîne de caractères de ce tableau associatif et nous avons créé une réponse JSON contenant la longueur de la chaîne et sa valeur en majuscules. Enfin, nous avons envoyé cette réponse JSON au client en utilisant `json_encode(...)` et `echo`.

Pour tester le Web service, vous pouvez utiliser l'outil Postman ou envoyer une requête HTTP POST depuis un autre script PHP. Voici un exemple de script PHP qui envoie une requête POST au Web service et affiche la réponse JSON :

```
1 <?php
2 $data = ['string' => 'Hello, world!'];
3 $options = [
4     'http' => [
5         'method' => 'POST',
6         'header' => 'Content-Type: application/json',
7         'content' => json_encode($data)
8     ]
9 ];
10 $context = stream_context_create($options);
11 $result = file_get_contents('http://localhost/webservices.php', false, $context);
12 $response = json_decode($result, true);
13
14 echo "Length: " . $response['length'] . "\n";
15 echo "Uppercase: " . $response['uppercase'] . "\n";
16
```

Dans cet exemple, nous avons créé un tableau associatif `$data` contenant une chaîne de caractères. Nous avons ensuite créé un tableau `$options` contenant la méthode HTTP POST, l'en-tête `Content-Type: application/json` et les données JSON encodées. Nous avons créé un contexte de flux à l'aide de `stream_context_create(...)` et nous avons envoyé une requête HTTP POST au

Web service en utilisant `file_get_contents(...)`. Enfin, nous avons affiché la réponse JSON décodée.

CONCLUSION GENERALE

En général, le cours du Web est un ensemble d'enseignements et de compétences pratiques qui permettent aux étudiants de comprendre les technologies et les concepts fondamentaux nécessaires pour concevoir, développer et déployer des applications Web.

Le cours de Web comprend des éléments tels que HTML, CSS, JavaScript, les frameworks Web, les bases de données et les services Web. Ces éléments sont essentiels pour construire des sites Web modernes et interactifs, qui sont aujourd'hui omniprésents dans notre vie quotidienne.

Les compétences acquises dans le cadre du cours de Web peuvent être appliquées dans une variété de domaines, tels que la création de sites Web personnels, la conception d'applications Web pour les entreprises, la création de plateformes de commerce électronique, ou la mise en place de systèmes de gestion de contenu pour des organisations.

Le cours du Web est un domaine en constante évolution, où de nouvelles technologies et tendances apparaissent régulièrement. Le Web est devenu une plateforme essentielle pour les entreprises et les individus pour communiquer, échanger, partager des informations et vendre des produits et services. Au fil des années, le Web a connu de nombreuses avancées, notamment dans le développement de sites Web, l'expérience utilisateur, la sécurité, l'optimisation pour les moteurs de recherche, les réseaux sociaux et la publicité en ligne. Les tendances actuelles incluent l'utilisation de l'intelligence artificielle, le développement de sites Web progressifs, l'optimisation mobile et la réalité virtuelle.

Le cours du Web est un domaine passionnant pour ceux qui souhaitent travailler dans le développement de sites Web, la conception, la gestion de contenu, la publicité en ligne et le marketing numérique. Cependant, il est important de rester constamment à jour sur les dernières tendances et technologies pour rester compétitif dans ce domaine en évolution rapide.

En conclusion, le cours de Web est un élément important de l'éducation en informatique, car il permet aux étudiants de se familiariser avec les technologies qui sont aujourd'hui indispensables pour la création de sites Web modernes et pour l'expansion de l'économie numérique.

REFERENCES BIBLIOGRAPHIQUES

1. Beard, J., Walker, A., & George, J. (2020). The principles of beautiful Web design. Sitepoint.
2. Cerami, E. (2002). Web services essentials: distributed applications with XML-RPC, SOAP, UDDI & WSDL. " O'Reilly Media, Inc.".
3. Duckett, J. (2014). Javascript and jquery: Interactive front-end Web development. Wiley Publishing.
4. Duckett, J., & Schlüter, J. (2011). HTML and CSS. Wiley.
5. Ejsmont, A. (2015). Web scalability for startup engineers. McGraw Hill Professional
6. Felke-Morris, T. (2015). Web development and design foundations with HTML5 (p. 672). Pearson education limited.
7. Goldstein, A., Lazaris, L., & Weyl, E. (2015). HTML5 & CSS3 for the Real World.
8. James Gillies, Robert Cailliau, How the Web was born: the story of the World Wide Web, Oxford, Oxford University Press, 2000, (ISBN 0-19-286207-3), p. 87.
9. Nixon, R. (2014). Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5. " O'Reilly Media, Inc.".
10. Richardson, L., & Ruby, S. (2008). RESTful Web services. " O'Reilly Media, Inc.".
11. Robbins, J. N. (2012). Learning Web design: A beginner's guide to HTML, CSS, JavaScript, and Web graphics. " O'Reilly Media, Inc.".
12. Site saint-genis-pouilly.fr, page "Histoire du CERN" [archive].
13. Tim Berners-Lee, Mark Fischetti, Weaving the Web: the past, present and future of the World Wide Web by its inventor, Londres, Texere, 2000, 272 p. [détail de l'édition] (ISBN 1-58799-018-0) p. 27.
14. Tim Berners-Lee's account of the exact locations at CERN where the Web was invented [archive], sur le site davidgalbraith.org.
15. Ullman, L. (2009). PHP for the Web: Visual QuickStart Guide. Peachpit Press.
16. Ullman, L. (2011). Php and mysql for dynamic Web sites: visual quickpro guide. Peachpit Press.
17. Verou, L. (2015). CSS Secrets: Better Solutions to Everyday Web Design Problems. " O'Reilly Media, Inc.".
18. Welling, L., & Thomson, L. (2003). PHP and MySQL Web development. Sams publishing.