

**République Algérienne Démocratique & Populaire**  
***Ministère de l'Enseignement Supérieur & de la Recherche***  
***Scientifique***

**Université Dr Moulay Tahar de Saida**



**Faculté de Technologie**  
**Département d'informatique**

**Polycopié de cours**

---

# **Développement d'Applications Web**

## **(DAW)**

---

**DR Hadj Ahmed BOUARARA**

**Année universitaire**

**2022-2023**

**Deuxième Année Licence**

# Table des matières

Table de figures .....	5
<b>Chapitre 0 : Avant cours</b>	
- Bienvenue au cours des développement des application web .....	7
- Avant-propos .....	7
- Objectif du cours .....	8
- Annotations .....	8
- Organisation du cours.....	9
- Quelles sont les conditions pour terminer le cours .....	9
<b>Chapitre 1 : introduction au World Wide Web (WWW)</b>	
- Web et internet .....	10
- Server web . .....	11
- Front end VS back end.....	12
- Site web statique VS site web dynamique. ....	14
- Domain name system (DNS) .....	15
- Les outils de développement .....	16
<b>Chapitre 2 : HyperText Markup Language (HTML)</b>	
- Introduction et structure de base d'un document HTML .....	18
- C'est quoi balise, attribut et élément HTML .....	19
- Ma première page web .....	19
- Exemple d'une page web simple .....	20
- Les titres en HTML .....	20
- Exemple : création des différent titres en HTML .....	21
- Un saut de ligne et ligne de séparation en HTML .....	21
- Les balises de formatage de texte .....	22
- Les listes en HTML .....	23
- Les liens en HTML .....	24
- Les balises sémantiques en HTML .....	24
- Les balises de formatage de texte .....	25
- Les tableaux en HTML .....	26
- Exercice sur les tableaux .....	26
- Embedding contents.....	27
- Les formulaires en HTML.....	29

- Création de formulaire de contact .....	31
- Les divisions et parties en HTML .....	33
- Les sous formulaires .....	33
- Liste de tous les balise HTML .....	34
<b>Chapitre 3 :Cascadin Style Sheets (CSS)</b>	
- CASS basic .....	35
- Internal CSS .....	37
- Commentaire CSS .....	38
- External CSS .....	38
- Inline CSS.....	39
- Exemple de css simple .....	41
- Sélecteurs en CSS .....	42
- Les règles en CSS .....	45
- Les couleurs et text en CSS .....	49
- CSS Box .....	51
- Positionnement en CSS .....	53
<b>Chapitre 4 : java script</b>	
- Introduction java script .....	44
- Internal java script .....	45
- External java script .....	47
- Internal java script .....	48
- Variables et primitive .....	49
- Les opérateurs en java script .....	51
- Java script statement .....	54
- Fonctions, objects et événements en JS .....	59
<b>Chapitre 5 : PHP</b>	
- PHP basic .....	67
- Installation du PHP .....	69
- Variable PHP .....	71
- Les listes en PHP .....	76
- Les expressions PHP .....	78
- Les structures conditionnelles .....	81
- Les boucles en PHP .....	85
- Les tableaux en PHP .....	86

## Chapitre 6 : PHP et MYSQL

- Introduction.....	88
- Requêtes de base de données .....	89
- Ouvrir une connection avec mysql .....	90
- Créer une base de données MySql avec php .....	91
- Créer une table mysql avec php .....	91
- Insérer des données mysql avec php .....	92
- Insérer plusieurs enregistrement MySql .....	92
- Sélectionner des données à partir d'une base de données MySql .....	95
- Sélectionner et filtrer les données d'une base de données MySql .....	96
- Supprimer les données d'une table MySql .....	97

## Chapitre 7 : les travaux pratiques

- TP 1 : web et internet .....	99
- TP 2 : HTML .....	102
- TP 3: CSS .....	104
- TP 4 : Java script .....	106
- TP 5 : PHP.....	107

## Table de Figures

Figure 1: explication d'un protocole de communication .....	25
Figure 2: communication entre deux systèmes.....	29
Figure 3 : les services d'internet et protocoles de communication .....	31
Figure 4 : exemple d'utilisation du protocole TCP/IP .....	34
Figure 5 : protocole HTTP et comment accéder aux données du serveur .....	35
Figure 6 : processus du hyper text transfer protocole (HTTP) .....	36
Figure 7 : questions réponses entre requester et provider .....	38
Figure 8 : communication entre un serveur web et un navigateur.....	42
Figure 9 : principe du serveur DNS .....	43
Figure 10 : Adresse IP du site google .....	46
Figure 11 : site web dynamique VS site web statique .....	53
Figure 12 : exemple d'un nom du domaine .....	56
Figure 13 : Fonctionnement du web .....	63
Figure 14 : Fonctionnement du serveur local Apache.....	67
Figure 15 : Répertoire du logiciel WAMP .....	69

# Chapitre 0 :

# Avant cours

---

« La vie c'est comme une bicyclette,  
Il faut avancer pour ne pas perdre l'équilibre »

**ALBERT Einstein**

## **0.1. Bienvenue au cours du développement d'applications web (DAW):**

Je vous souhaite la bienvenue au cours développement des applications web (DAW). Je m'appelle BOUARARA Hadj Ahmed, enseignant à l'université DR Molay Tahar, SAIDA Algérie, et je serai votre hôte pour ce cours. Je suis heureux de vous avoir parmi nous et j'espère que vous apprécierez ce cours au fur et à mesure que vous commencerez votre voyage vers la programmation des applications web. Ce cours est destiné aux étudiant du 2em année licence informatique.

## **0.2. Avant-propos :**

Depuis sa naissance dans les années 1990, le domaine du développement web a connu de nombreux changements. En pleine croissance, le métier de développeur web *full stack*, ou spécialisé comme un développeur *front-end* ou *back-end*, est également influencé par les exigences de plus en plus spécifiques des internautes. Heureusement, les outils sont aussi de plus en plus nombreux et permettent de créer des sites interactifs et sur mesure, qui répondent de manière pertinente à la demande des utilisateurs.

À la fin de ce cours, vous devez acquérir suffisamment de connaissances de base pour pouvoir approfondir vos compétences, que ce soit en tant que carrière de chercheur ou en appliquant les outils du développement web dans d'autres domaines intéressants comme la médecine, la biologie, l'éducation.....etc.

**Hadj Ahmed BOUARARA**

### **0.3. Objectif du cours DAW :**

Dans ce polycopié, nous apprendrons les outils de base que chaque codeur de page Web doit connaître. Nous commencerons par le bas en apprenant comment implémenter des pages Web modernes avec HTML et CSS. Nous passerons ensuite à apprendre à coder nos pages de telle sorte que ses composants se réorganisent et se redimensionnent automatiquement en fonction de la taille de l'écran de l'utilisateur. Vous pourrez coder une page Web qui sera tout aussi utile sur un téléphone mobile que sur un ordinateur de bureau. Enfin et surtout, nous obtiendrons une introduction approfondie au langage le plus omniprésent, populaire et incroyablement puissant du Web: Javascript. En utilisant Javascript, vous pourrez créer une application Web entièrement fonctionnelle qui utilise PHP pour exposer les fonctionnalités et les données côté serveur à l'utilisateur final.

### **0.4. Annotations**

- **WWW** : World wide web
- **HTML** : hypertext markup language
- **CSS** : cascading style sheets
- **JS** : java script
- **PHP** : hypertext preprocessing
- **XML** : extensible markup language
- **DTD** : data type definition
- **DAW** : developpement des applications web
- **HTTP** : hypertext transfer protocole
- **TCP/IP** : transmission contrôle protocole / internet protocole
- **FTP** : file transfer protocol

### **0.5. Organisation du cours :**

Ce cours est organisé comme suit :

- **Chapitre 1 : introduction au WEB**

Dans ce chapitre nous allons parler de la différence entre web et internet, site web dynamique et site web statique, adresse IP privée et adresse IP publique, domaine name system (DNS) et comment la traduction entre un nom du domaine et une adresse IP est réalisée.

## - **Chapitre 2 : HTML**

Cette section donne une vue générale sur le HTML qui est un langage de programmation qui permet aux développeurs Web de créer une page HTML statique pouvant être visualisée sur un navigateur Web. Le langage HTML est essentiellement utilisé pour développer des pages Web. Ce livre vous aidera à comprendre les bases du langage HTML et comment le mettre en pratique pour créer des sites Web.

## - **Chapitre 3 : CSS**

Cette section explique la feuille de style en cascade qui permet aux développeurs Web de concevoir la page HTML pouvant être visualisée sur un navigateur Web. Les CSS sont des règles de style appliquées sur les documents HTML. Ce chapitre vous aidera à comprendre les bases de la syntaxe CSS et comment la mettre en pratique pour créer des sites Web.

## - **Chapitre 4 : Java script**

Dans ce chapitre nous allons détailler le JAVASCRIPT qui est un langage de programmation côté client qui permet aux développeurs Web de créer des scripts pouvant s'exécuter sur le navigateur client. Le langage JAVASCRIPT est essentiellement utilisé pour exécuter un programme côté client. Ce livre vous aidera à comprendre les bases du langage JAVASCRIPT et comment le mettre en pratique pour créer des sites Web.

## - **Chapitre 5 : PHP**

Cette partie permet l'illustration du PHP Hypertext Preprocessor (PHP) est un langage de programmation qui permet aux développeurs Web de créer du contenu dynamique qui interagit avec les bases de données. PHP est essentiellement utilisé pour développer des applications logicielles basées sur le Web. Ce tutoriel vous aidera à comprendre les bases de PHP et comment le mettre en pratique.

## - **Chapitre 6 : Mysql avec PHP**

Cette partie permet de faire connaissances sur les détails de création des bases et de données et manipulation des requêtes MySql à l'aide de php.

## - **Chapitre 7 : les travaux pratiques**

Pour permettre aux étudiants de mieux pratiquer l'utilisation des fondamentaux DAW un ensemble de travaux pratiques et de projets ont été regroupés dans ce chapitre comme la création d'un site web statique, concevoir un site web à l'aide des plateformes en ligne gratuit comme WIX et Wordpress, création d'un site web dynamique ....ect).

### **0.6. Quelles sont les conditions pour terminer le cours ?**

Le cours est conçu pour l'étude auto-rythmée d'environ 8-12 heures par semaine pendant 14 semaines, y compris des quiz / vérifications des connaissances, des tutoriels/ activités d'apprentissage pratiques, des devoirs, des tests pour vous aider à vous préparer aux évaluations finals et d'autres lectures.

# Chapitre 1 :

## Introduction Au World Wide Web (WWW)

---

### 1. Introduction

#### 1.1. Réseau : machines connectées ensemble

#### 1.2. Internet :

Un réseau global qui permet de relier tous les appareils ayant un accès aux services internet (ordinateur, téléphone,.....ect) dans le monde. C'est d'ailleurs de là que vient son nom: «interconnected network» réseau interconnecté en anglais. Il est la structure qui permet d'envoyer des données d'un ordinateur à un autre, grâce à un système complexe d'adresses qui font parvenir ces informations au bon endroit [1].

##### 1.2.1. L'origine de l'internet

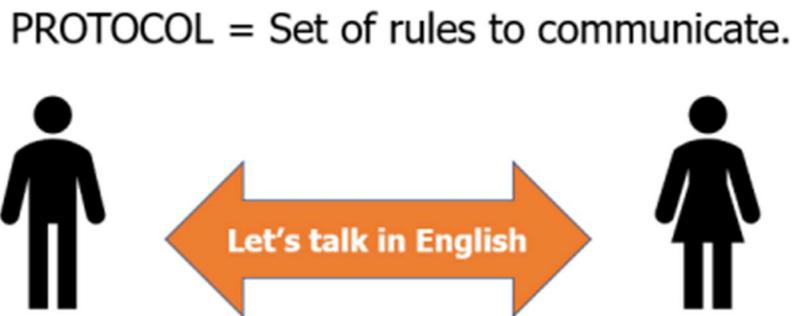
Il y'a plusieurs années au début des année 1970 la conception d'internet à commencer. Il était le résultat d'une expérience appelée ARPANET « adncaed research projects agency network » projet du département americain de la défense. Il a d'abord été utilisé à des fins militaires, puis pour faciliter la transmission d'informations entre des universités et des centres de recherche [1].

##### 1.2.2. Les services de l'internet

- **La visualisation des pages web et les news :** Les utilisateurs peuvent faire une visualisation de plusieurs pages web. Cette visualisation de page web s'avère d'ailleurs être le service le plus connu, le plus récent et le plus utilisé en ce qui concerne Internet.
- **Telnet, SSH et FTP :** des services incontournables Telnet et SSH sont des services offerts par Internet qui permettent aux internautes de se connecter. Le FTP est un service de transfert de fichiers. Il permet de disposer des fichiers sur une machine distante, de télécharger ou de transférer des fichiers et de faire une diffusion de plusieurs fichiers [2].
- **Le service de messagerie et de l'email :** Ce service a été mis en place en suivant plusieurs protocoles dont le SMTP ou Simple Mail Tranfert Protocol qui sont des protocoles utilisés dans l'envoi des messages. Il y a également le POP ou Post Office Protocol, pour la réception [2].

#### 1.3. Protocole :

Un protocole en langage courant signifie ensemble de règles pour communiquer. C'est une sorte de règles qui doivent être suivies pour obtenir les résultats souhaités [2].



**Figure 1:** explication d'un protocole de communication

Imaginez, si je ne parle que la langue anglaise et que vous voulez communiquer avec moi. Alors le protocole que nous pouvons utiliser pour communiquer les uns avec les autres pourraient être :

- Langage corporel
- Langue des signes
- Langue anglaise

Ils sont tous appelés Protocole. C'est le moyen par lequel nous nous engageons à communiquer les uns avec les autres. Donc, si vous connaissez l'anglais, vous pouvez communiquer avec moi en anglais et c'est notre protocole pour communiquer [2]. Un protocole est une convention de communication entre deux ordinateurs. Lorsque deux ordinateurs communiquent entre eux, ils utilisent à chaque fois plusieurs protocoles en même temps, et ces protocoles sont inclus les uns dans les autres [3].

### **1.3.1. TCP/IP**

Le protocole de base qui structure Internet (pas seulement le web, mais aussi les autres composantes d'Internet, comme le FTP et le mail) s'appelle TCP/IP. De fait, il s'agit déjà de deux protocoles imbriqués : TCP et IP. La couche supérieure: Transmission Control Protocol gère le découpage d'un message ou un fichier en petits paquets qui sont transmis sur Internet et reçues par une couche TCP qui rassemble les paquets dans le message original rassemble les paquets dans le message original de la couche inférieure: Internet Protocol gère la partie adresse de chaque paquet (adresse IP). Chaque ordinateur sur le réseau vérifie cette adresse pour transmettre le message [3].

### **1.3.2. HTTP**

Au dessus du TCP/IP, le web repose sur le protocole HTTP. Il est le protocole spécifique du web. HTTP signifie : Hyper-Text Transfert Protocole (http) est donc le langage dans laquelle le serveur et le client dialogue. Il est utilisé pour la communication entre les clients Web et serveurs. Pour demander une page Web, un navigateur envoie un message HTTP GET Pour envoyer des données vers un serveur Web, un navigateur peut utiliser un message HTTP POST[4].

## **1.4. Comment deux systèmes communiquent-ils ?**

Imaginez que deux systèmes veulent communiquer entre eux et échanger des données. Comment communiquent-ils avec les autres ? Tout ce qu'ils savent, c'est 0 et 1. C'est là que le protocole vient et aide.

PROTOCOL = Set of rules to communicate.



Figure 2 : communication entre deux systems [4]

Deux systèmes doivent d'abord se mettre d'accord sur un protocole commun. Ils doivent savoir quelles règles ils vont suivre pour communiquer entre eux. Il devrait y avoir une sorte d'instructions prédéfinies lorsqu'une communication peut être établie. Ainsi, avant que deux ordinateurs ne décident de communiquer, ils conviennent d'un protocole commun et sur la base de cette définition de protocole, ils communiquent entre eux.

Quel genre de règles ou d'instructions suivent-ils pour communiquer ?

PROTOCOL = Set of rules to communicate.

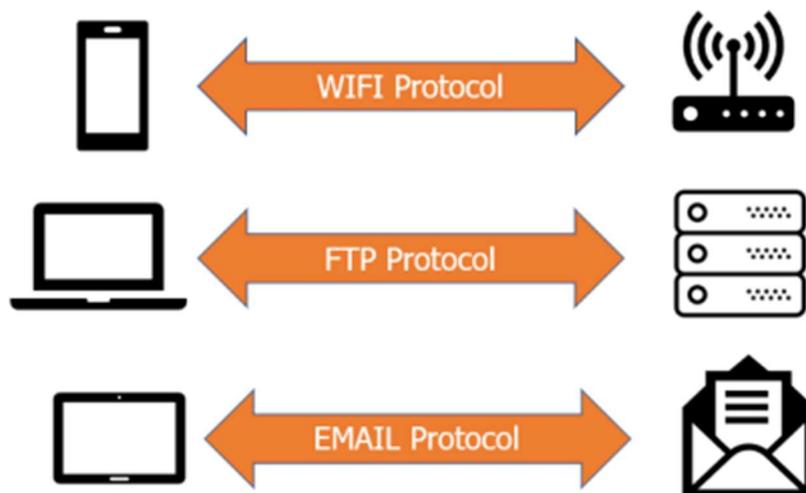


Figure 3 : les services d'internet et protocoles de communication [5].

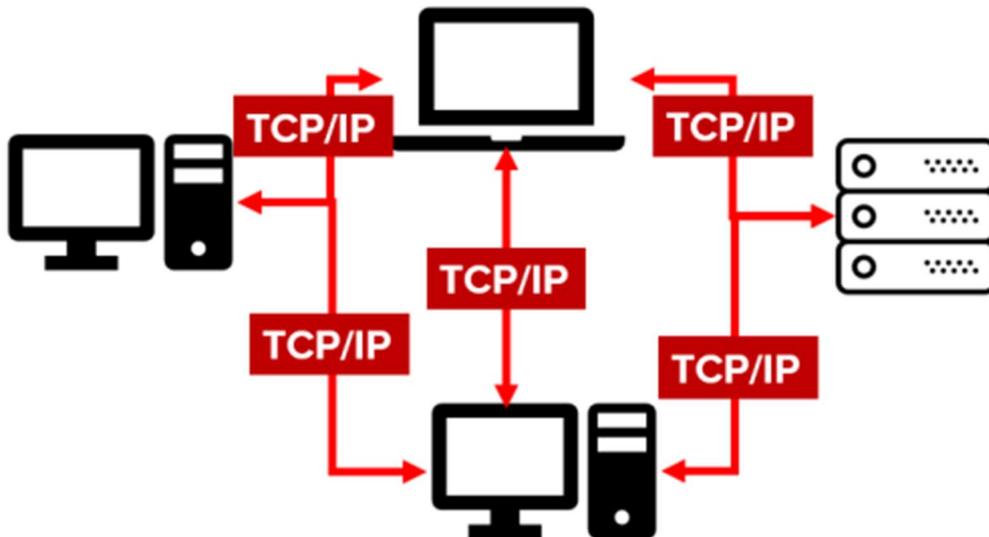
Voici quelques exemples qui montrent comment les deux appareils utilisent un protocole spécifique pour communiquer entre eux.

- **Protocole Wifi :** Ce protocole permet de connecter votre téléphone mobile à vos appareils Wifi et d'accéder à Internet.

- **Protocole FTP** : ce protocole FTP peut aider à accéder au système de fichiers de n'importe quel serveur et à effectuer les opérations de fichiers sur celui-ci, telles que télécharger, télécharger, supprimer et renommer les fichiers.

#### 1.4.1. Comment les données sont-elles échangées sur Internet ?

Les données sur Internet sont échangées avec le protocole TCP/IP. C'est le protocole que chaque ordinateur utilise pour échanger des données. C'est comme une langue que chaque ordinateur de ce réseau parle pour communiquer.



**Figure 4** : exemple d'utilisation du protocole TCP/IP [7]

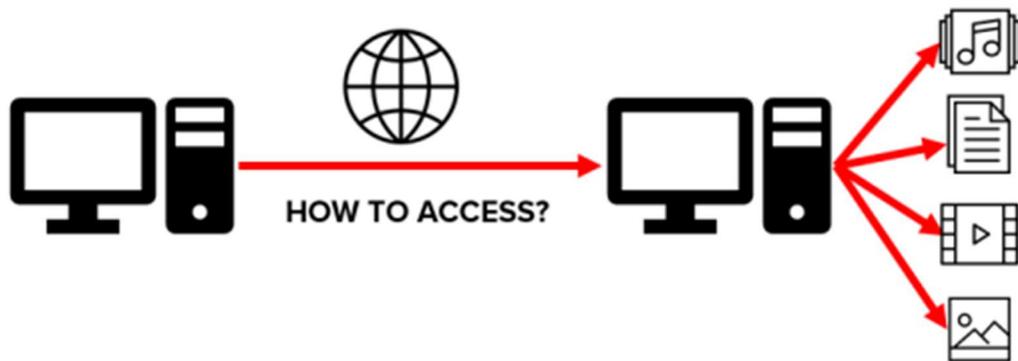
Il s'agit d'une exigence selon laquelle chaque ordinateur connecté au réseau mondial (Internet) doit parler le langage du protocole TCP/IP pour communiquer. De nombreux autres protocoles sont construits sur la base de ce protocole TCP/IP [8].

#### 1.5. Le world wide web :

Un système qui permet de naviguer de pages en pages en cliquant sur des liens hypertexte grâce à un navigateur. Le Web est né en 1989 par Tim Berners-Lee, qui est un physicien au CERN, le Conseil européen pour la recherche nucléaire. Le Web permet de consulter des pages sur des sites, grâce à des navigateurs adaptés, et via des machines comme des ordinateurs ou des smartphones. Il s'agit un peu d'une immense bibliothèque de documents, qui repose sur les fondations complexes d'Internet. Pour conclure nous pouvons dire que le web représente un service d'internet [9].

##### 1.5.1. HTTP et Serveur WEB :

Le Web n'est pas Internet. L'internet est un moyen de connecter des ordinateurs entre eux et le Web est le moyen d'accéder à ces informations via Internet. Le Web est une méthode permettant d'accéder à ces informations sur Internet.



**Figure 5 :** Protocole HTTP et comment accéder aux données du serveur [9]

Le Web est la façon dont les informations sont partagées sur Internet. Les informations sont généralement représentées par du texte, des images, de la vidéo, de l'audio et d'autres types de médias. Le World Wide Web (WWW) ou Web sont en fait la même chose qui permet d'accéder à ces informations. Le Web réalise deux choses :

- 1) Accédez à la ressource depuis n'importe quel ordinateur sur Internet.
- 2) Reliez ces ressources [10].

### 1.5.2. Hypertexte signifie un texte spécial qui a :

- Un lien vers d'autres ressources sur Internet.
- Il peut inclure des vidéos, des images et des sons.
- Le HTTP est le protocole à l'aide duquel Hyper Text est téléchargé à partir du serveur et affiché sur le navigateur.
- Une fois cet hypertexte stocké sur le serveur connecté à Internet, elle peut être échangé entre les ordinateurs à l'aide du protocole HTTP.

En bref, HTTP transporte l'Hyper Texte entre les deux ordinateurs pour établir une communication.

# Hyper Text Transfer Protocol (HTTP)



Figure 6 : process du hyper text transfer protocole (HTTP) [10].

- C'est un langage que deux machines parlent.
- **Comment fonctionne HTTP ?**  
La règle de HTTP est (DEMANDER ET SERVIR). Les hypertextes sont disponibles dans les systèmes connectés à Internet. Quelqu'un doit faire une demande à cette machine et demander à la récupérer. Ensuite, une fois la demande effectuée, elle est servie en envoyant l'hypertexte via la réponse. Deux rôles sont impliqués dans le processus HTTP. Le premier est appelé demandeur et le second fournisseur.

Rules of HTTP = Ask and it is served.

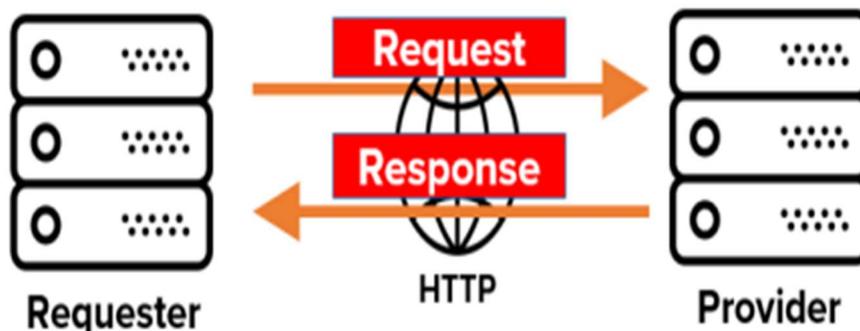


Figure 7: questions/réponses entre requester et provider [10]

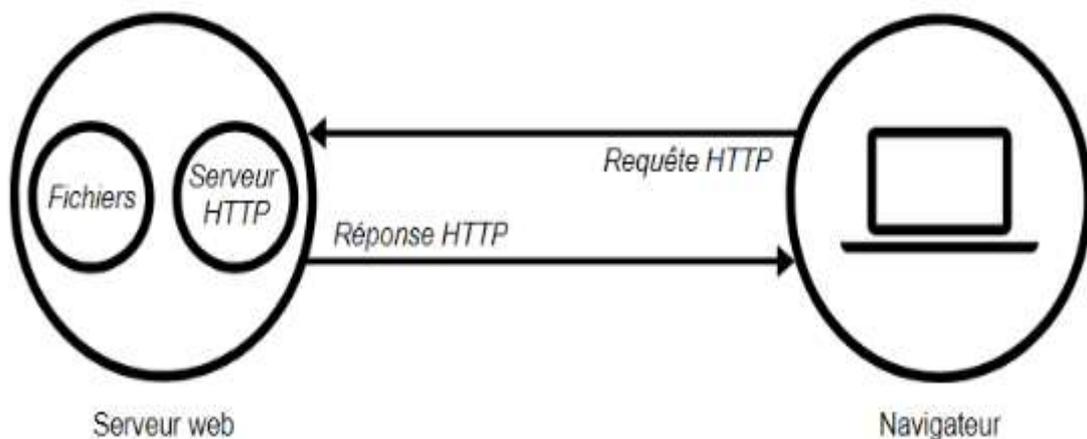
- Le demandeur est l'initiateur de la demande. Celui qui demande la ressource. Le fournisseur sert la demande et fournit la réponse au demandeur. Ainsi, pour que le protocole HTTP fonctionne, le demandeur doit initier la demande en demandant une ressource spécifique sur ce serveur, puis le fournisseur une fois la demande reçue, il recherchera et renverra cette ressource au demandeur. S'il n'y a pas de demandeur, le fournisseur ne fait rien [10].

**1.5.3. Est-ce que quelqu'un contrôle Internet ?** Personne ne contrôle Internet. C'est un réseau où les ordinateurs sont connectés avec le protocole TCP/IP

## 1.6. Un serveur web

Un serveur informatique qui répond à des requêtes du World Wide Web, en utilisant principalement le protocole HTTP. Un « serveur web » fait référence à des composants logiciels (software) et/ou matériels (hardware) [11].

- Au niveau des composants matériels, un ordinateur qui stocke les fichiers qui composent un site web (par exemple les documents HTML, les images, les feuilles de style CSS, les fichiers JavaScript) et qui les envoie à l'appareil de l'utilisateur qui visite le site. Cet ordinateur est connecté à Internet et est généralement accessible via un nom de domaine [12].
- Au niveau des composants logiciels, un serveur web contient différents fragments qui contrôlent la façon dont les utilisateurs peuvent accéder aux fichiers hébergés. On trouvera un serveur http qui est un logiciel qui comprend les URL et le protocole HTTP. Au niveau le plus simple, à chaque fois qu'un navigateur a besoin d'un fichier hébergé sur un serveur web, le navigateur demande (on dit qu'il envoie une requête) le fichier via HTTP. Quand la requête atteint le bon serveur web (matériel), le serveur HTTP (logiciel) renvoie le document demandé, également grâce à http [12].



**Figure 8:** Communication entre un serveur web et un navigateur [10].

## 1.7. Principe du client/serveur

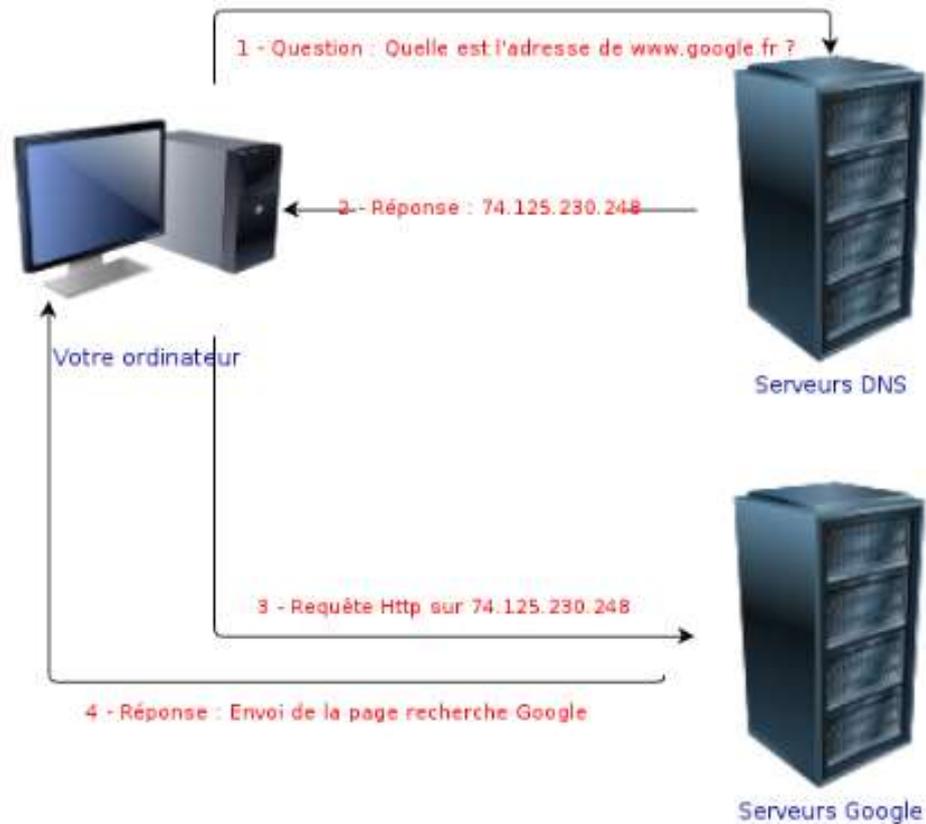
Les caractéristiques d'un client sont les suivantes : il est d'abord actif (ou maître), il envoie des requêtes au serveur, il attend et reçoit les réponses du serveur. Un serveur : Un serveur est initialement passif, il attend, il est à l'écoute, prêt à répondre aux requêtes envoyées par des clients. Dès qu'une requête lui parvient, il la traite et envoie une réponse. Le dialogue : Le client et le serveur doivent bien sûr utiliser le même protocole de communication. Un serveur est généralement capable de servir plusieurs clients simultanément [11].

## 1.8. Serveur DNS (domain name system) :

Le DNS est un acronyme signifiant Domain Name System. C'est un protocole permettant d'associer à des noms de domaine (par exemple `www.chezmoi.fr`) une adresse IP (par exemple `140.78.132.45`). Les machines se connectent entre-elles à l'aide d'adresses IP, mais les noms de domaine servent à faciliter la mémorisation et l'utilisation pour les humains. Reprenons en modèle ce que vous avez fait pour arriver sur cette page, et regardons comment le serveur DNS est

intervenu sans que vous le sachiez. Vous avez lancé votre navigateur Internet et vous avez taper « www.google.fr » comme moteur de recherche par défaut. Donc la page GOOGLE s'est affichée. Voici le mécanisme complet et regardez à quel endroit le serveur DNS est appelé.

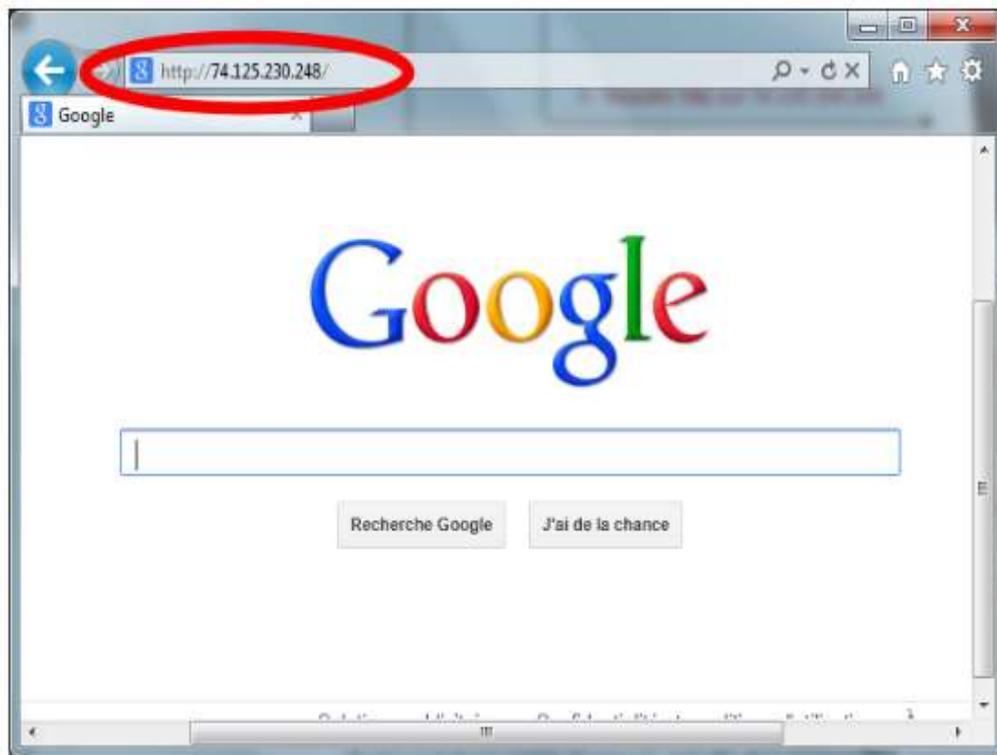
### Principe d'une requête DNS



**Figure 9** : principe du serveur DNS [13].

Dans l'exemple ci-dessus, on voit que la requête "quelle est l'adresse de www.google.fr" a répondu 74.125.230.248. Cette requête s'appelle une résolution de nom de domaine. Si l'on poursuit, on peut constater que le serveur DNS n'est utilisé que sur la partie 1-Question et 2-Réponse. Une fois que l'ordinateur a récupéré l'adresse du serveur à joindre, il n'a plus besoin du serveur DNS [14].

Cette adresse IP n'est pas une adresse au hasard. Elle correspond bien à un des serveurs de Google. (il y en a évidemment un grand nombre). Donc, si dans votre dans la barre d'adresse de votre navigateur, vous tapez : <http://74.125.230.248>, la page de recherche de Google doit s'afficher.

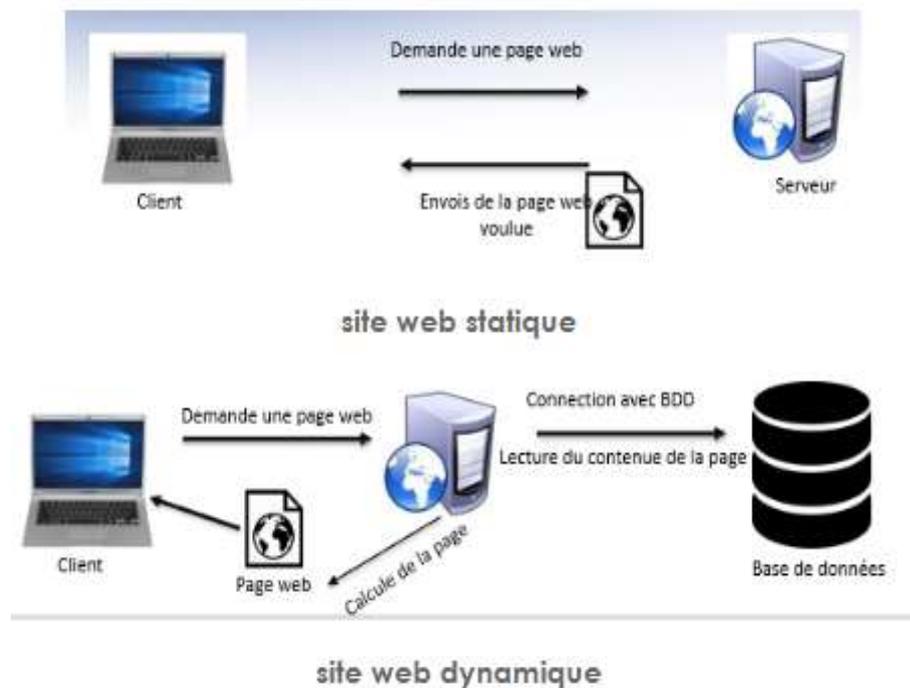


**Figure 10 :** Adresse IP du site web Google

### **1.9. Site web statique vs site web dynamique**

Un serveur peut « servir » du contenu statique ou dynamique. Un contenu « statique » signifie qu'il est servi tel quel. Les sites web statiques sont les plus simples à mettre en œuvre et il sera donc préférable de commencer par un site statique.

Un site « dynamique » signifie que le serveur traite le contenu ou le génère à la volée depuis les informations contenues dans une base de données. Cette solution est plus flexible mais beaucoup plus complexe à mettre en œuvre.



**Figure 11:** site web dynamique VS site web statique

Les avantages de l'architecture client-serveur sont :

- Une économie d'espace mémoire (tous les postes clients vont chercher l'information en fonction de leurs besoins, et ne doivent pas la stocker durablement)
- Une garantie de cohérence (à chaque instant, le serveur constitue un référentiel unique, identique pour tous les clients, alors qu'une duplication sur chaque poste client entraîne généralement une divergence entre les différentes versions, et une perte du référentiel commun). Cette garantie de cohérence est particulièrement souhaitable lorsque l'information subit des mises à jour [11].
- Une maîtrise du service (le gestionnaire du serveur contrôle ce que les programmes clients peuvent faire ou ne pas faire sur le serveur). Le premier avantage est de nature physique. Les deux autres touchent directement à l'organisation du travail et des processus [11].

Lorsqu'on parle de « client » et de « serveur », il peut s'agir à la fois de l'ordinateur et du programme qui tourne sur cet ordinateur. Dans la suite de ce cours, « client » et « serveur » désigneront les programmes. Les ordinateurs seront explicitement désignés par « ordinateur client » et « ordinateur serveur ». Internet est l'exemple extrême d'architecture client-serveur : chaque serveur est potentiellement accessible par des millions de clients. On peut dire que le Web illustre de manière aiguë les avantages de l'architecture client-serveur : chacun est libre de décider de ce qu'il met en ligne, chacun possède en permanence le droit d'accéder à un volume de données plus d'un million de fois supérieur à ses capacités de stockage personnelles. Sur le web [13].

- le serveur s'appelle « serveur internet ». Le programme le plus utilisé est Apache [13].
- le client s'appelle navigateur, (ou bien browser, ou butineur). Les programmes les plus utilisés sont Chrome, Internet Explorer et Firefox[12].

### 1.9.1. C'est quoi un nom de domaine :

Une adresse Internet ou nom de domaine est l'équivalent de votre adresse postale sur Internet. C'est la manière dont vos contacts et clients vont trouver votre site Internet sur le web. Un nom de domaine est donc indispensable lors de la création de votre site web[9]. Une **adresse Internet** se compose d'un préfixe "www" (world wide web) et d'un nom de domaine. Ce nom de domaine est lui-même composé d'une chaîne de caractères et d'une extension (TLD - Top Level Domain) [10]. Dans l'exemple ci-dessous, l'extension utilisée est relative à la France : le **.fr**.



Figure 12 : exemple d'un nom de domaine

Ainsi votre nom de domaine vous permet d'exprimer toute votre créativité, par le choix des mots et des extensions qui font sens pour votre public [3].

### 1.9.2. Comment trouver un nom de domaine :

Très important, le nom de domaine : il ne faut pas négliger cette étape car vous ne pourrez/devrez plus en changer. Si votre site est une marque, alors vous pouvez/devez prendre le nom de cette marque si le domaine est disponible. Sinon le choix de votre nom de site est primordial [16].

- Choisir un nom de domaine parmi les top 5 des mots clés de votre site
- Rendre le domaine unique
- Pensez toujours à vérifier que ce nom de domaine soit disponible avec les extensions dont vous avez besoin. Essayer d'avoir un nom du domaine avec extension .com pour que votre site soit visitable ou mémorable par l'écrasante majorité des consommateurs.
- Assurer-vous que votre nom de domaine soit facile à taper et facile à retenir
- Garder le nom le plus court possible
- Éviter la violation du copyright
- Rejeter les tirets et les numéros

Utiliser ces deux liens suivant pour trouver un nom de domaine libre

- <https://instantdomainsearch.com/>
- <https://leandomainsearch.com/>

### 1.9.3. Application web :

Une application web désigne un logiciel applicatif hébergé sur un serveur et accessible via un navigateur web

#### 1.9.3.1. Site web :

Un site web est l'ensemble des pages web et des ressources inter reliées entre elles par des liens hypertextes, auxquelles l'internaute peut accéder par une adresse web appelée Url, le tout enregistré sous le même nom de domaine.

Une page web peut être défini comme étant l'ensemble des documents html structurés, stockés sur un serveur connecté au réseau mondial qu'est internet [15].

#### 1.9.3.2. Hébergeur de site web :

Lorsque vous vous rendez sur un site web depuis votre navigateur, votre ordinateur reçoit un message écrit dans les langages du web (html css javascript). Votre navigateur traduit ce langage sous la forme que vous connaissez : images, textes, boutons CTA, tout ce qui compose une page web. Ce code provient de différents fichiers qui ont besoin d'un espace de stockage : cet espace est ce qu'on appelle un serveur. Ce dernier est composé d'un ordinateur qui contient de la mémoire, un disque dur et un système d'exploitation. L'hébergeur web est l'entreprise qui est propriétaire des serveurs où sont stockées les informations liées à votre site web. Généralement, en plus de vendre de l'espace sur des serveurs, les hébergeurs web proposent aussi des services de gestion de serveurs, de soutien informatique, d'antivirus et de sauvegarde de sites web [11].

#### 1.9.3.3. Les différents types de sites web

Il existe différents types de **sites web**, chaque type correspond à un objectif[12]. On peut citer:

- Les **sites e-commerce** ou sites marchands qui pour principal objectif la vente en ligne.
- Les **sites vitrines**, utilisés majoritairement pour présenter et exposer les services d'une organisation ou une marque [13].
- Les **sites institutionnels**, qui Présentent une organisation et ses valeurs, à travers la description de son activité, de ses chiffres clés et des informations indispensables au public ciblé [14].
- Les **sites portails**, qui proposent des services de messagerie, d'actualités [15].
- Les **sites personnels**, réalisés au profit des particuliers, qui désirent partager en ligne leur passion pour un sujet précis[14].

Cette liste est loin d'être exhaustive, on retrouve d'autres sites tels que : Les Blogs, les sites communautaires, les sites collaboratifs [11].

### 1.10. Les outils de développements :

Les outils de développement web Pour notre travail nous avons besoin des outils suivants : - Installation du l'éditeur du code Visual studio code lien du téléchargement

<https://code.visualstudio.com/download>

- Téléchargement d'un navigateur web comme google chrom, mozilla, opera (exécution du code coté client). L'objectif d'un navigateur web c'est de lire le document html et affiche les résultats. Le navigateur n'affiche pas les balises mais utilise les éléments html, CSS et JS pour afficher le contenu. - Un serveur apache (exécution du code coté serveur) pour cela il suffit d'installer XAMP [9].

- Un système de gestion de base de données MySQL (PHP My Admin Intégrer dans XAMP).
- **FIGMA** : pour la construction visual d'une page web
- Redimensionner les images utilisant Paint ou adobephotoshop.
- Valider votre code html :  
<https://validator.w3.org/>
- Lien pour des images et des icônes de qualité :  
<https://fontawesome.com/>
- Sélectionner un host web :  
<https://www.siteground.com/tutorials/webhosting/choosehost/>
- Site pour trouver un nom du domaine libre :  
<https://instantdomainsearch.com/>

### Des ressources externes

- Le centre de developpement mozilla : <https://developer.mozilla.org/fr/>
- Livre enligne html et css <https://learn.shayhowe.com/html-css/>.
- Guide sur l'hébergeur du site web : <https://www.webhostingsecretrevealed.net/webhosting-beginner-guide/>
- Vous pouvez vérifier si le navigateur que vous utilisez prend en charge la plupart des balises HTML5 en allant sur <http://html5test.com/index.html>. Ce site vous donnera une note pour le navigateur que vous utilisez.

Vous pouvez également cliquer sur le lien "Autres navigateurs" pour avoir une idée de la façon dont les autres navigateurs fonctionnent. - Site de tester votre code interactif : <https://codepen.io/shayhowe/pen/nvlqy>.



# Chapitre 2 :

## Hypertext Markup Language (HTML)

---

### 2. Introduction

L'Hypertexte signifie un texte spécial qui a :

- Un lien vers d'autres ressources sur Internet.
- Il peut inclure des vidéos, des images et des sons.

La partie la plus importante d'HyperText est la possibilité de lier d'autres ressources sur le serveur accessible via le lien. Les hypertextes sont affichés sur l'écran de l'ordinateur par le navigateur comme Google Chrome, Edge de Microsoft et Firefox [9].

#### 2.1. Fonctionnement du web :

Le but principal du Web ou World Wide Web (WWW) est d'échanger des informations d'un ordinateur à un autre ordinateur. Le WWW est le moyen de connecter tous les documents et de les afficher sur le navigateur. Le Web utilise le protocole HTTP pour transférer les informations sur Internet. Les hypertextes sont transférés d'un ordinateur à un autre via le protocole HTTP et affiché sur le navigateur [10].

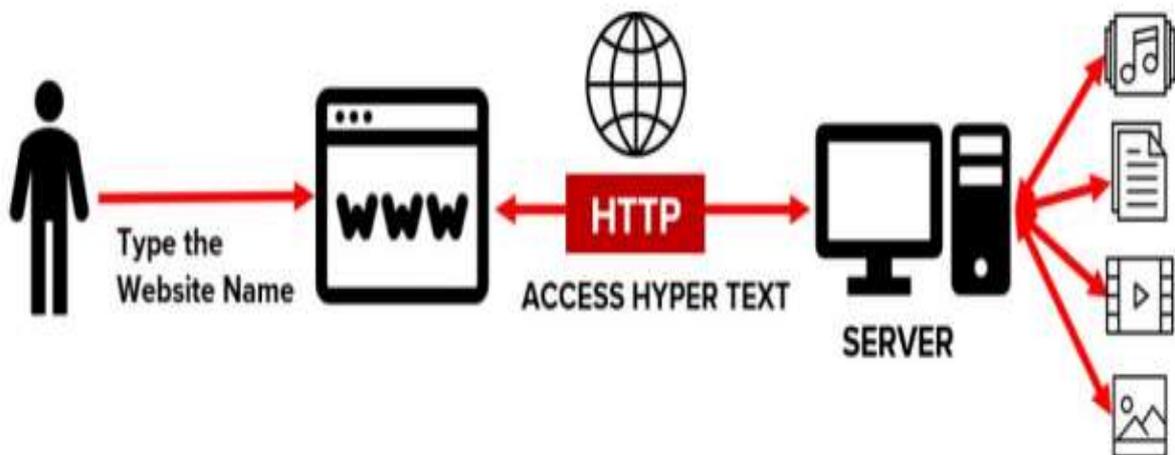


Figure 13: Fonctionnement du web [13].

- Vous tapez le nom du site Web dans le navigateur. Par exemple. [www.univ-saida.dz](http://www.univ-saida.dz)

- Le navigateur envoie la demande au serveur DNS.
- Le serveur DNS recherche l'adresse IP pour ce nom de domaine.
- Une fois l'adresse IP trouvée, elle se connecte ensuite au serveur spécifique sur Internet.
- Le serveur reçoit la requête via le protocole HTTP.
- Le serveur recherchera alors l'hypertexte demandé et le renverra au navigateur.
- Le navigateur recevra l'hypertexte et l'affichera sur l'écran de l'ordinateur.

### 2.1.1. Quel est le but du langage de balisage ?

- Les hypertextes sont généralement échangés entre les serveurs pour permettre la communication entre les machines, mais pour que cet hypertexte soit converti en un langage et afin que nous puissions écrire le langage de telle manière que nous puissions accomplir une tâche spécifique [11].
- Le langage de balisage est utilisé pour représenter la façon d'organiser les données. Ce type de langage se concentre davantage sur la manière dont les données doivent être utilisées et définit le but des données [10].

Afin d'organiser l'hypertexte qui se trouve partout sur l'ordinateur dans un format compréhensible par l'homme, un langage de balisage était nécessaire. HTML résout ce problème de balisage des données requises du serveur de manière à ce que nous en comprenions le but. Par exemple, nous avons un fichier vidéo enregistré sur le serveur. Maintenant, nous voulons dire à tout le monde quel est le but de cette vidéo et aussi une description de la vidéo lorsque quelqu'un accède à ce contenu. Emballer ces informations dans un exemple de langage de balisage pourrait ressembler à ceci [13].

---

```
<title>Comment perdre du poids en 10 jours</title>
<description>Un moyen rapide et facile de perdre du poids sans aller à salle de sport
</description>
<videolocation>/assets/video/looseweight.mpeg</videolocation>
```

---

Il s'agit d'un langage de balisage mais pas de HTML. Nous pouvons emballer notre contenu dans un balisage Langage permettant de comprendre l'hypertexte que nous obtenons du serveur [15].

## 2.2. Qu'est-ce que le HTML ?

HTML signifie Hyper Text Markup Language. HTML est un langage de balisage utilisé pour marquer le contenu, puis dire au navigateur comment les afficher à l'écran. HTML est un langage de balisage que notre navigateur comprend et sait comment les afficher [13].

En langage HTML, vous ferez les choses suivantes :

- Écrivez comment afficher le contenu sur le navigateur.
- Liens vers une autre ressource sur le serveur. (Hyperliens)
- Intégrez des vidéos et de l'audio à partir du serveur.
- Disposez le contenu.
- Le but principal du langage de balisage HTML est de dire au navigateur comment les données sont affichées et à quel endroit.

### 2.2.1. Histoire du HTML ?

Le scientifique anglais Tim Berners-Lee a inventé le World Wide Web en 1989. Il a construit un navigateur à l'aide duquel la ressource a été consultée sur Internet et affichée sur le navigateur. En 1991, il a été rendu public. À partir de là, toutes les grandes entreprises ont sorti leurs propres navigateurs comme Google l'a fait Chrome et Windows ont publié Edge. Berners Lee a publié le premier exemple de page HTML contenant 18 balises et affiché sur son propre navigateur [11].

### 2.3. Qu'est-ce qu'une application Web ?

Les applications développées et déployées sur le serveur Web sont appelées applications Web. Les applications Web sont de petits projets avec un tas de fichiers et de médias accessibles via le serveur Web [11].

#### 2.3.1. Qu'est-ce qu'une page Web ?

Les pages Web sont les documents qui peuvent être affichés dans le navigateur comme Google Chrome, Edge ou Safari. Les pages Web sont écrites à l'aide du langage de balisage HTML et sont stockées sur le serveur Web. Ensuite, les pages Web sont accessibles sur Internet via un navigateur pour afficher la sortie sur l'écran de l'ordinateur.

#### 2.3.2. Qu'est-ce qu'un site Web ?

Le site Web est une collection de pages Web. Chaque site Web est accessible avec un nom de domaine. Un regroupement logique de pages Web pour effectuer une fonctionnalité souhaitée est appelé site Web.

#### 2.3.3. Qu'est-ce que la balise ?

Une balise est le texte entre le crochet gauche (<) et le crochet droit (>).

Il y a des balises de début (telles que <name>) et il y a des balises de fin (</name>)

`<TAG>` = Starting Tag / Opening Tag

`</TAG>` = Ending Tag / Closing Tag

#### 2.3.4. Qu'est-ce que l'élément ?

Un élément est la balise d'ouverture, la balise de fermeture et tout ce qui se trouve entre les deux.

Element : `<balise>` valeur de l'element `</balise>`

#### Qu'est-ce qu'un élément vide ?

L'élément qui n'a pas de données entre les balises est appelé élément vide [15].

Element vide : `<balise></balise>`

`<BALISE />` est également appelé élément à fermeture automatique.

#### 2.3.5. Qu'est-ce que l'attribut ?

Un attribut est une paire nom=valeur à l'intérieur de l'élément [11].

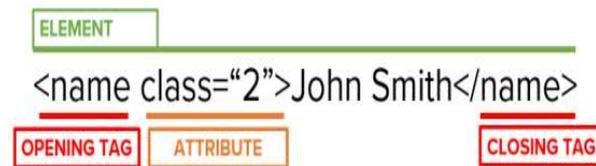
`<balise nom_attribut= « value »>` texte `</balise>`

Aide sur les attributs pour afficher des détails supplémentaires sur l'élément.

Exemple: <chien>berger allemand</chien> = ELEMENT

<dog color="brown">Shepard allemand</dog> = ELEMENT avec ATTRIBUT

Color = "brown" est appelé comme attribut.



## Objectif du HTML ?

HTML a été écrit pour organiser les données de manière à ce qu'elles puissent être correctement affichées sur le navigateur [9].

- Structure des données ? Le langage de balisage HTML utilise une structure de balisage pour organiser les éléments de la page. Considérez cet exemple :
- 

---

```
<html>
<head>
<title>Hello World!</title>
</head>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph</p>
</body>
</html>
```

---

Ne vous souciez pas du HTML mais concentrez-vous davantage sur la façon dont les éléments sont organisés pour indiquer au navigateur le but de chaque élément et la structure de chaque élément. L'objectif principal est d'indiquer au navigateur comment afficher le contenu sur l'ordinateur l'écran pour que vous organisiez les annotations de la même manière qu'elles devraient être affichées. Le séquençage est également important et détermine la façon dont les éléments sont organisés sur la page. Il existe également une relation parent-enfant entre les éléments. Les éléments head et body sont des éléments enfants pour HTML p et h1 sont des éléments enfants de l'élément parent body [13].

### 2.4. Les titres en HTML

Les titres sont définis avec les <h1> à <h6> balises. <h1> définit le cap le plus important. <h6> définit la moins importante cap [16].

---

```
<!DOCTYPE html>
<html>
<body>
<h1>horizon titre 1</h1>
```

---

```
<h2>horizon titre 2</h2>
<h3>Horizon titre 3</h3>
<h4>Horizon titre 4</h4>
<h5>Horizon titre 5</h5>
<h6>Horizon titre 6</h6>
</body>
</html>
```

---

Les moteurs de recherche utilisent les titres pour indexer la structure et le contenu de vos pages Web. Par exemple : <h1>Les titres doivent être utilisés pour les titres principaux, suivi par <h2> titres, les moins important <h3>, et ainsi de suite [17].

## 2.5. Lignes horizontales :

La ligne horizontale agit comme un séparateur entre les sections et le paragraphe. Ceci est utilisé lorsque vous souhaitez ajouter de l'espace entre les paragraphes et également afficher une ligne indiquant un sujet différent [13].

- TAG : <hr />

N'oubliez pas que <hr> est une balise à fermeture automatique que vous n'avez pas besoin de fermer.

### Attributs de la balise hr :

- width = pourcentage (%)
- align = gauche, droite, centre

---

```
<h1> heading 1 </h1>
<hr>
<h2> heading 2 </h2>
<hr>
<h3> heading 3 </h3>
<hr>
<h4> heading 4 </h4>
<hr>
```

---

## 2.6. Paragraphes en HTML :

Les paragraphes sont l'endroit où vous mettez la plupart de votre contenu à afficher sur le navigateur. Les balises de paragraphe aident à bien organiser le contenu dans un petit conteneur qui rend le contenu facile à lire et à modifier[11].

- Balise : <p>
- ÉLÉMENT : <p>un texte</p>

**Exercice 1** : Créez une page HTML et écrivez les titres 1 à 3 et sous chaque titre, écrivez un paragraphe de texte.

## Today's News

1. It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout. The point of using Lorem Ipsum is that it has a more-or-less normal distribution of letters, as opposed to using 'Content here, content here', making it look like readable English. Many desktop publishing packages and web page editors now use Lorem Ipsum as their default model text, and a search for 'lorem ipsum' will uncover many web sites still in their infancy. Various versions have evolved over the years, sometimes by accident, sometimes on purpose (injected humour and the like).

## Today's News

2. It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout. The point of using Lorem Ipsum is that it has a more-or-less normal distribution of letters, as opposed to using 'Content here, content here', making it look like readable English. Many desktop publishing packages and web page editors now use Lorem Ipsum as their default model text, and a search for 'lorem ipsum' will uncover many web sites still in their infancy. Various versions have evolved over the years, sometimes by accident, sometimes on purpose (injected humour and the like).

## Today's News

3. It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout. The point of using Lorem Ipsum is that it has a more-or-less normal distribution of letters, as opposed to using 'Content here, content here', making it look like readable English. Many desktop publishing packages and web page editors now use Lorem Ipsum as their default model text, and a search for 'lorem ipsum' will uncover many web sites still in their infancy. Various versions have evolved over the years, sometimes by accident, sometimes on purpose (injected humour and the like).

**Exercice 2 :** Créez une page HTML et écrivez les titres 1 et un paragraphe en dessous. Ensuite, tracez une ligne en dessous, puis affichez un peu plus de texte.

**Saut de ligne en html :** Les sauts de ligne aident à rompre le lien comme la nouvelle ligne (`\r\n`) que nous avons dans le fichier texte normal. Les pauses sont utilisées lorsque nous voulons rompre une ligne en cours et commencer une nouvelle ligne. Plus vous avez de pauses, plus de nouvelles lignes sont ajoutées.

- Balise : `<br>`
- ÉLÉMENT : `<br>`
- `<br>` est une balise impaire

**Exercice 3 :** Créer une page HTML et ajouter une ligne de rupture dans les titres 1.

**Exercice 4 :** Créez une page HTML et décomposez le paragraphe ligne par ligne jusqu'à 10 lignes [13].

### Texte strong :

Une balise strong aide à mettre le texte en gras et à mettre en évidence le texte pour mettre en évidence certaines lignes dans le paragraphe. Il est souvent utilisé dans les paragraphes [11].

- Balise : `<strong>`
- ÉLÉMENT : `<strong>quelque chose</strong>`

**Exercice 5:** Créez une page HTML et créez trois lignes en gras dans un paragraphe.

**Exercice 6:** Créer une page HTML et mettre en gras le premier et le dernier mot des paragraphes

### 2.7. Texte d'accentuation emphasis :

L'emphase est utilisée pour mettre en évidence un mot ou une ligne dans les paragraphes. C'est similaire à italiques, mais cela est plus utilisé pour mettre l'accent sur un mot ou une ligne. Il est souvent utilisé dans les paragraphes [14].

- Balise : `<em>`
- ÉLÉMENT : `<em>quelque chose</em>`

**Exercice 8:** Créer une page HTML et mettre l'accent sur l'ensemble du paragraphe

### 2.7.1. Texte souligné

Underline soulignera simplement le texte. Il est souvent utilisé dans les paragraphes[13].

- balise : `<u>`
- ÉLÉMENT : `<u>quelque chose</u>`

### 2.7.2. Text Italique

La balise Italique mettra simplement le texte en italique. Il est souvent utilisé dans les paragraphes[14].

- Balise : `<i>`
- ÉLÉMENT : `<i>quelque chose</i>`

### 2.7.3. Code en HTML :

La balise Code est utilisée pour afficher le code source de programmation sur la page Web. Le navigateur traite la balise de code comme spéciale et affiche le code tel qu'il est écrit dans le code HTML. Cependant, il ignore les nouvelles lignes. Ainsi, le code est affiché sur une seule ligne [2].

- balise : `<code>`
- ÉLÉMENT : fonction `add(x, y){ return a + b }`

**Exercice 9 :** Créez une page HTML et écrivez un code Java simple.

### 2.7.4. Balise pré-formatée (pre) :

La balise « PRE » est utilisée pour afficher l'espace blanc et elle conservera l'indentation du format qui est écrit dans le fichier html. Le navigateur ne formate rien à l'intérieur de la balise pre, mais à la place, il essaiera d'imprimer tel quel, y compris l'espacement et la mise en forme [13].

- Balise : `<pre>`
- ÉLÉMENT : `<pre>ligne1 ligne2 ligne3</pre>`

## 2.8. Les listes en HTML :

### 2.8.1. Liste de définition :

Utilisation de la liste de définitions. La liste de définition est utilisée pour définir la définition de la liste. Il est souvent utilisé comme en-tête [11].

- Balises : `<dl>`, `<dt>`, `<dd>`
- ÉLÉMENT:
 

```

<dl>
<dt>titre</dt>
<dd>définition</dd>
</dl>
```

**Exercice 10:** Créez une page HTML et créez quelque chose comme ci-dessous avec h1, dl, p et hr.

# Definition List

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

## HTML5 eBook

All your want to learn about HTML5

It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

---

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

## 2.8.2. Liste ordonnée

La liste ordonnée est utilisée pour afficher la liste des éléments avec des numéros. Il est souvent utilisé pour lister les points comme 1, 2, 3. Les nombres seront générés automatiquement par la balise.

- Balises : `<ol>`, `<li>`
- ÉLÉMENT: 

```
<ol>
<li>titre</li>
<li>description</li>
</ol>
```

---

Exercice 11 : Créez une page HTML et créez quelque chose comme ci-dessous

# Ordered List Example

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.

---

---

1. New List Item 1

---

2. New List Item 2

---

3. New List Item 3

---

## 2.8.3. Liste non ordonnée

La liste non ordonnée est utilisée pour afficher la liste des éléments avec un point circulaire. Il est souvent utilisé pour lister les points sans numéro. Vous pouvez modifier la valeur par défaut liste du point du cercle à tout autre symbole ou image.

- Balises : `<ul>`, `<li>`
- ÉLÉMENT:

---

```
<ul>
<li>titre</li>
<li>description</li>
</ul>
```

---

Exercice 12 : Créez une page HTML et créez quelque chose comme ci-dessous

- UnOrdered List Items - 1
  1. List Item 1
  2. List Item 2
  3. List Item 3

- 
- UnOrdered List Items - 2
    1. List Item 4
    2. List Item 5
    3. List Item 6

## 2.8.4. Utilisation de l'article

L'article est utilisé pour définir un groupe de contenu qui a un titre, des liens et un paragraphe pour constituer un contenu indépendant sur la page. Une page de blog peut avoir la liste de tous les articles affichés en petit résumé. L'article peut avoir un en-tête, un paragraphe et un pied de page.

Cela constitue un regroupement logique de choses dans cet article. La balise <section> est utilisée pour regrouper les articles en sections.

- Balise : <article>, <section>, <en-tête>, <pied de page>

## 2.9. Intermédiaire HTML :

**2.9.1. Table :** Le tableau est utilisé pour organiser les données dans un format de ligne et de colonne. C'est similaire à montrer les données dans un format de base de données ou un format de feuille Excel.

### 2.9.2. Balises de tableau : <table>, <th>, <tr>, <td>

- <table> est utilisée pour définir la table
- <th> est utilisé pour définir le titre.
- <tr> est utilisé pour mentionner la ligne.
- <td> représente la colonne dans la table.

### 2.9.3. ÉLÉMENT du tableau :

---

```
<tableau>
<th>Titre</th>
<tr>
<td>Colonne</td>
</tr>
</table>
```

---

**Exercice 13 :** Créez une page HTML et créez un tableau comme celui-ci avec une ligne et une colonne simples.

#	Name	Course
1	John	WordPress
2	Kelly	C
3	Kenny	C++

### 2.10.2. Tableau avec bordures :

border est un attribut du tableau. Elle spécifie la largeur de la bordure pour tracer la ligne autour du tableau. Border aidera à afficher les lignes du tableau pour comprendre la largeur et la hauteur de chaque cellule. Vous pouvez mentionner la largeur de la bordure avec cette valeur.

**Exercice 14:** Créez une page HTML et créez quelque chose comme ci-dessous.

Heading 1	Heading 2
Col 1	
	Col 2

### 2.10.3. Tableau avec en-tête et pied de page

Le tableau a deux sections importantes en dehors de la ligne et de la colonne et c'est l'en-tête et pied de page. En-tête et pied de page tels qu'ils sont utilisés pour spécifier du texte avant la ligne et après la ligne.

Balises : <table>, <caption>, <thead>, <tfoot>, <tbody>

#### ELEMENT:

---

```
<table>
<caption>Caption</caption>
<thead>
<tr>
<td>Heading 1</td>
</tr>
</thead>
<tfoot>
<tr>
<td>Footer 1</td>
</tr>
</tfoot>
<tbody>
<tr>
<td>Column 1</td>
</tr>
</tbody>
</table>
```

---

**Exercice 15:** Créer une page HTML et créer un tableau comme celui-ci :

Heading 1
Column 1
Footer 1

Heading 1	Heading 2	Heading 3
Column 1		
Footer 1	Footer 2	Footer 3

### 2.10.4. Tableau avec RowSpan et ColSpan

RowSpan et ColSpan consistent à ajouter un espacement à la cellule. Si vous souhaitez fusionner deux cellules ensemble, cet attribut vous aidera. rowspan et colspan sont des attributs que vous pouvez utiliser dans le <td> ou <tr>.

---

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```

    <meta name="description" content="Page Description">
    <title>Table with RowSpan and ColSpan</title>
</head>
<body>
  <h1>Table with RowSpan and ColSpan Example</h1>
  <table border="1">
    <tr>
      <th>Heading 1</th>
      <th>Heading 2</th>
      <th>Heading 3</th>
    </tr>
    <tr>
      <td>Col 1</td>
      <td rowspan="2">Col 2</td>
      <td>Col 3</td>
    </tr>
    <tr>
      <td>Col 1</td>
      <td>Col 3</td>
    </tr>
  </table>
  <hr>
  <table border="1">
    <tr>
      <th>ID</th>
      <th>Name</th>
    </tr>
    <tr>
      <td>1</td>
      <td>John Smith</td>
    </tr>
    <tr>
      <td colspan="2">Total: 2</td>
    </tr>
  </table>
</body>
</html>

```

## 2.10. Embedded contents :

Les balises d'intégration sont utilisées pour inclure des ressources externes dans la page html. Ces ressources peuvent se trouver sur votre serveur ou se trouver à un autre endroit. Le but principal de WWW était de connecter tous les contenus avec un lien et cela rend le Web si puissant qu'il a la capacité de connecter des choses partout sur le Web. Ce concept rend l'ensemble du Web très puissant [13].

# Chapitre 3 :

## Cascading Style Sheets (CSS) et XML eXtensible Markup language (XML)

---

### 3. Introduction

CSS signifie feuille de style en cascade. Il est utilisé pour décrire comment le contenu doit être affiché sur le navigateur, l'impression ou l'écran. Avec CSS Language, vous pouvez contrôler la mise en page de la page, la couleur du texte, la taille de la police, l'espacement entre le texte, la largeur et la hauteur des éléments et la présentation complète de la page Web. En bref, CSS gère l'apparence de la page Web. HTML est utilisé pour décrire le contenu et CSS est utilisé pour afficher le contenu de manière présentable.

#### 3.1. Utilisation de CSS

Les CSS sont écrits dans un fichier avec l'extension .css et sont liés à la page HTML. Définissez le style une fois, puis utilisez-le n'importe où sur votre site. Chargez le CSS une fois par page et il gèrera l'ensemble de la mise en page et de la présentation. Aide à modifier la mise en page en fonction de l'écran sur lequel le site est visualisé, comme un mobile, une tablette ou un écran d'ordinateur.

CSS aide à séparer le travail de présentation de la page HTML et le développeur peut se concentrer sur la création du contenu et l'afficher séparément. Les normes mondiales suggèrent également d'utiliser CSS et de ne pas utiliser d'attributs HTML pour styliser la balise. Réutilisez le même CSS pour plusieurs sites WordPress pour avoir la même apparence.

##### 3.1.1. Page HTML Sans CSS

Sans CSS, la page HTML sera affichée selon le formatage et la coloration par défaut du navigateur. La plupart du temps, c'est en noir et blanc avec une taille de police définie par le navigateur.

< body >

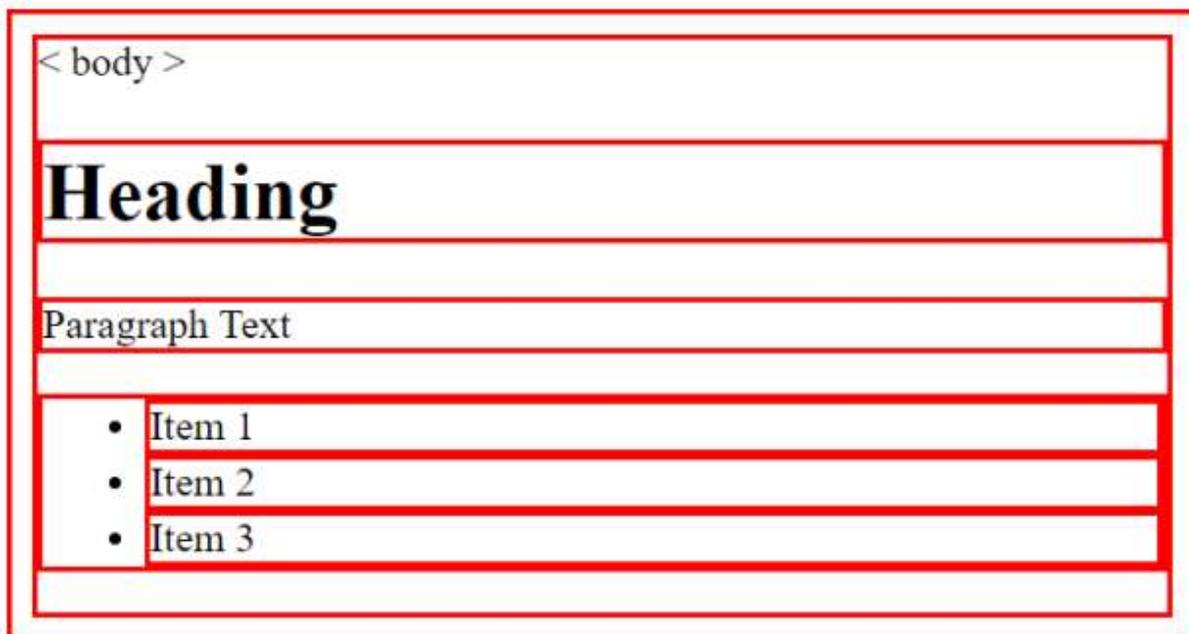
# Heading

Paragraph Text

- Item 1
- Item 2
- Item 3

### 3.1.2. Page avec CSS

Avec CSS, vous pouvez ajouter des styles à chacun des éléments de la balise HTML. CSS peut contrôler chaque élément et le formater.



La syntaxe du CSS :

SELECTORS

DECLARATION BLOCK

**h1 { color: red; font-size: 12px; }**

DECLARATION

DECLARATION

**{ property: value; }**

Dans cet exemple CSS ci-dessus, nous modifions la propriété de la balise h1 pour l'afficher en couleur ROUGE et la taille de police à 12 points.

### 3.2. Langage et termes CSS :

- propriété représente le nom des sélecteurs sont les balises HTML qui vous souhaitez appliquer le style.
- Les sélecteurs personnalisés sont les sélecteurs dont le nom ne correspond pas avec le nom de la balise HTML.

Exemple : h1 est un sélecteur et h1a est un sélecteur personnalisé.

- l'attribut que vous souhaitez modifier. valeur est la valeur de cette propriété.
- { } est appelé en tant que bloc.
- { valeur de la propriété; } est appelé Déclaration.
- La collection de déclaration est appelée bloc de déclaration.
- Vous pouvez séparer chaque déclaration avec ; à l'intérieur du bloc.
- La propriété de déclaration est prédéfinie comme la couleur, la taille de la police ne peut pas être un nom personnalisé.

**Résumer du CSS :** CSS est une collection de sélecteurs qui a de nombreux blocs de déclaration pour modifier les balises HTML et vous pouvez définir des sélecteurs personnalisés qui peuvent être appliqués sur les balises HTML.

#### 3.2.1. Internal CSS :

La feuille de style en cascade (CSS) peut être écrite à de nombreux endroits différents dans la page HTML. L'une des méthodes de définition du CSS est à l'intérieur de la même page HTML. Ce type d'inclusion CSS est limité au niveau de la page uniquement, ce qui signifie que vous ne pouvez pas réutiliser ce code dans d'autres pages. L'avantage d'utiliser ce css interne est lorsque vous souhaitez que des modifications spécifiques s'appliquent uniquement à ce niveau de page.

Exemple :

```
<head>  
<style type="text/css">
```

---

```
h1 { color: red; }
</style>
</head>
```

---

- La balise <style> est utilisée pour écrire le CSS à l'intérieur de cette balise.
- Cette balise <style> est définie à l'intérieur du <head>

Les données écrites à l'intérieur de la balise <style> ne sont pas affichées dans le navigateur mais elles sont utilisées comme instruction au navigateur pour faire quelque chose sur la page.

**Exercice 16:** colorier le paragraphe en bleu et la liste en vert Couleur.

**Exercice 17:** Modifier la taille de la police des balises h1, p et li.

---

```
h1 – { font-size: 12 px; }
p – { font-size: 24 px; }
li – { font-size: 48 px; }
```

---

Suivez la syntaxe correcte pour ajouter le CSS interne sur la page HTML.

### 3.2.2. Commentaire en CSS :

Vous pouvez utiliser la notation spéciale pour commenter le code à l'intérieur du CSS. Les commentaires HTML et les commentaires CSS ne sont pas les mêmes. Ne vous confondez pas avec les commentaires en CSS par rapport aux commentaires en HTML. Les commentaires en CSS sont des commentaires sur plusieurs lignes.

- /\* est utilisé pour indiquer que les commentaires commencent.
- \*/ est utilisé pour indiquer que les commentaires sont terminés.

Tout ce qui se trouve entre /\* et \*/ ne sera pas exécuté par le navigateur.

### 3.3.3. External CSS :

La feuille de style en cascade (CSS) peut être écrite à de nombreux endroits différents dans Page HTML. L'une des méthodes de définition du CSS est externe, ce qui signifie que vous pouvez écrire du CSS dans un fichier et l'importer dans la page HTML. CSS externe signifie que le CSS est écrit en externe dans un autre fichier qui est ensuite lié dans la page. Ce type de CSS externe est une technique très puissante et utile qui est couramment utilisée dans chaque développement de site Web. L'avantage d'utiliser ce CSS externe est que vous avez un fichier CSS qui est inclus dans toutes les pages du site Web.

Exemple :

---

```
<head>
  <!-- Make sure styles.css file exists in the same folder -->
  <link rel="stylesheet" type="text/css" href="styles.css"
  media="screen" />
</head>
```

---

La balise <link> est utilisée pour lier la ressource à la page HTML. L'attribut de balise de lien permettra au navigateur de savoir de quel type de ressource il s'agit. L'attribut rel est utilisé pour indiquer au navigateur de quel type de ressource il s'agit. rel = "stylesheet" signifie qu'il s'agit d'un fichier avec CSS à l'intérieur.

- L'attribut type indique le type de contenu dans le fichier. Dans ce cas, c'est text/css
- L'attribut href est similaire à la balise <a> href pour mapper l'emplacement du fichier dans le serveur avec le chemin et le nom du fichier.
- L'attribut media indique au navigateur d'intégrer le fichier pour l'écran objectif.
- Il y a deux fichiers : index.html et styles.css
- Le fichier styles.css est lié à l'intérieur du fichier index.html avec la balise <link> dans la section <tête>. Nom du fichier : index.html

### Exercice 18:

Créer style1.css et style2.css et liés eux dans la page html. style1.css h1 { color: red; }. style2.css h1 { color: blue; }

### 3.2.3. Inline CSS

Le CSS en ligne est défini à l'intérieur de la balise elle-même comme un attribut. Il remplace tous les styles définis dans le CSS interne et le CSS externe. Ceci est défini dans la balise HTML en tant qu'attribut.

SYNTAXE: <h1 style="color: red;">Ce titre doit être de couleur rouge.</h1>

## 2. Sélecteurs CSS

### 2.1. Sélecteurs & Déclaration

Une règle CSS composée de deux parties : Sélecteur et Déclaration

SELECTORS  


---

**h1 { color: red; }**  


---

 DECLARATION

Le sélecteur est utilisé pour indiquer à quel élément cette règle doit être appliquée. Vous pouvez même écrire plusieurs éléments en les séparant par ","

SELECTORS  


---

**h1, p, li { color: red; }**  


---

 DECLARATION

Les éléments h1, p, li seront de couleur rouge avec cette seule règle CSS. La déclaration indique quelle est la règle à appliquer sur cet élément. La règle est indiquée par la propriété : valeur.

SELECTORS  


---

**h1 { color: red; font-size: 12px }**  


---

 DECLARATION  
 { property: value; }

Les déclarations multiples sont séparées par ";"

---

```
<head>
<style type="text/css">
h1 { color: red; }
p,
li { color: blue; font-size: 12px }
</style> </head>
```

---

3.3. **Universal Selector** : Le sélecteur universel est la règle que vous souhaitez appliquer pour tous les éléments de la page. Le caractère générique '\*' est utilisé pour représenter le sélecteur universel.

**UNIVERSAL  
SELECTORS**

\* { color: red; }

**DECLARATION**

Les sélecteurs universels sont appliqués sur tous les éléments de la page.

---

```
<head>
<style type="text/css">
{
border-style: solid;
border-color: red;
}
</style>
</head>
```

---

La propriété border-style est utilisée pour définir le style de la bordure border-color est utilisée pour indiquer la couleur de la bordure. Si vous observez correctement, chaque élément de la page HTML est entouré d'une boîte magique à l'aide de laquelle vous pouvez contrôler cet élément.

### 3.3.1. Sélecteur de types

Les sélecteurs de type sont définis avec des balises HTML uniquement. Lorsque vous trouvez un sélecteur auquel une règle est appliquée sur les balises HTML, il est appelé sélecteur de type. Les sélecteurs de type s'appliquent uniquement aux balises HTML.

SYNTAX:

---

```
<head>
<style type="text/css">
h1
{
border-style: solid;
border-color: red;
}
</style>
</head>
```

---

### 3.3.2. Sélecteur de classe

Class Selector est un sélecteur personnalisé que vous pouvez créer dans les règles CSS et appliquez-le à n'importe quelle balise HTML avec l'attribut de classe. Les noms de sélecteur de classe sont personnalisés, vous ne devez donc PAS utiliser les noms de balises HTML prédéfinis. Les noms de sélecteurs personnalisés sont des sélecteurs de classe qui peuvent être appliqués à n'importe quelle balise HTML. Le sélecteur de classe peut être défini sur n'importe quelle balise HTML spécifique et appliqué à tout spécifier les balises. Vous définissez le sélecteur de classe avec "." devant elle.

## CLASS SELECTORS

---

# .redcolor { color: red; }

Si le sélecteur commence par "." et son nom n'est pas une balise HTML, il est alors appelé Sélecteur de classe.

Syntaxe :

---

```
<head>
<style type="text/css">
.redcolor { color: red; }
</style>
</head>
<body>
<h1 class="redcolor">This is red color heading</h1>
</body>
```

---

## CLASS SELECTORS

---

# .redcolor { color: red; }



```
<h1 class="redcolor">Red Color</h1>
```

Si vous souhaitez créer un sélecteur de classe uniquement spécifique aux balises HTML, utilisez cette notation.

## CLASS SELECTORS

---

# h1.redcolor { color: red; }

Le sélecteur de classe `red :color` ne fonctionnera que pour la balise `h1`.

**Exercice 19:** Écrire la règle CSS avec le sélecteur de classe. Créez une règle appelée `".headings"` et faites en sorte que la couleur soit bleue, aligner le centre et l'italique. Créez une autre règle avec le même nom `".headings"` et modifiez la couleur au rouge.

### 3.3.3. Sélecteur multiple de classe

Vous pouvez appliquer le sélecteur à la balise HTML parente et tous les éléments enfants hériteront de la propriété des propriétés des balises parentes.

#### SYNTAX:

---

```
<head>
<style type="text/css">
div.redcolor { color: red; }
</style>
</head>
<body>
<div class="redcolor"> <p>This is red color paragraph</p> </div>
</body>
```

---

## 3.4. Couleur et Texte en CSS :

### 3.4.1. Couleurs

Vous pouvez représenter les couleurs en CSS de différentes manières. Les couleurs peuvent être appliquées à presque toutes les balises HTML comme l'arrière-plan, texte, bordure et remplissez la case. Méthode la plus couramment utilisée pour représenter les couleurs : Valeur RVB, Valeur du code hexadécimal, Nom de la couleur.

- RVB est représenté par la forme courte `rgb` (rouge, vert, bleu) et des chiffres à l'intérieur.
- RVB (255, 0, 0) - Rouge, RVB (0, 255, 0) - Vert

La valeur du code hexadécimal est également utilisée pour représenter les couleurs spécifiques. La valeur commence par `#`, puis suivie de chiffres et de caractères en hexadécimaux. Il est généralement composé de 6 chiffres.

- Couleur de fond : `#ff0000` ;

Vous pouvez utiliser MAJUSCULES – FF ou minuscules ff pour représenter la valeur HEX. Il est bon d'utiliser des minuscules.

### 3.4.2. Texte en CSS

La police ou le mot de texte est utilisé pour modifier le comportement du texte sur la page.

Voici quelques-unes des propriétés de police et de texte.

- `font-family` – Spécifiez le nom de la police à utiliser.
- `font-size` – Spécifiez la taille de la police en pixels ou px. C'est le même
- pixel utilisé dans MS Word.

- font-style - Utilisé pour appliquer l'italique, normal ou oblique
- font-weight - Le poids est utilisé pour représenter l'épaisseur du trait de la police. Généralement gras, léger, moyen
- text-transform – Contrôle la casse du texte – Majuscule ou minuscule.
- text-decoration – Décoration du texte souligné ou surligné.
- text-shadow – Ajoute une ombre à l'arrière du texte.

La modification des polices dans CSS n'est pas différente de la modification d'un document Word et de la modification du contenu. La plupart des mots correspondront aux mots-clés que nous utilisons dans le logiciel Microsoft Word.

### **3.5. Extensible Markup Language :**

Extensible markup language est un langage Standardiser par le W3C en 1998. Il a comme version le XML 1.0 et le XML 1.1 mais généralement la version 1.0 est la plus utilisé. Il permet le stockage l'échange, la description et la struction des données. Dans ce module le XML est utiliser pour la description syntaxique des pages web ainsi que le stockage des données.

#### **3.5.1. La difference entre XML et base de données (BDD) :**

- Le XML présente une vitesse de lecture très rapide, si vous devez accéder à une base de données, il faut d'abord exécuter le script PHP, ouvrir la base de données, rechercher dans la table, ça prend du temps, alors qu'avec un fichier XML vous le lisez très rapidement avec le langage PHP, et vous avez accès directement à l'information,
- La connexion avec une base de données différent d'un langage à un autre tout dépend de l'environnement de développement utiliser.
- Le xml est Compatible avec plusieurs langages de programmation.
- Fonctionne avec la majorité des systèmes d'exploitation.
- La facilité de convertir un document xml au format pdf html excel...ect
- Le xml permet de copier et sauvgarder facilement
- Facile à comprendre y'a pas de modèle de conception ni notion d'évènement ni contrainte.
- Le XML veut également compatible avec le web afin que les échanges de données puissent se faire facilement à travers le réseau Internet.
- Le XML se veut donc standardisé, simple, mais surtout extensible et configurable afin que n'importe quel type de données puisse être décrit.

### 3.5.2. Les caractéristiques du XML :

- La déclaration : un document XML doit toujours commencer par un prologue :

`<?xml version="1.0" encoding="UTF-8"? Standalone= yes >`

- **Les balises et les attributs (corps du document) :** Le corps du document XML constitué de l'ensemble des balises qui décrivent les données. Bien sur les noms des balises sont générique et non pas prédéfinie comme le HTML. Nous avons aussi la possibilité d'ajouter les attributs qui représente une information cachée supplémentaire.
- **Un document XML bien formé :** on dit qu'un document est bien formé si il respecte les critères suivants :
  - Respecter les règles de nommage des balises et des attributs
  - La présence d'une balise racine
  - La présence de la déclaration
  - Chaque balise ouvrante a une balise fermante et la hiérarchie des balises est respectée où il y'a pas de chevauchement.
- **Un document xml valide :** un document est valide si il suit son DTD et il est bien formé.

### 3.6. C'est quoi DTD (data type definition) :

Le DTD est un grammaire pour la structuration et la conception des documents XML. Il n'est pas obligatoire il peut être facultative, interne ou externe au document XML. IL contient des déclarations pour les éléments, attributs, entités, notations utilisées. On peut distinguer deux avantages principaux du DTD :

- Partage d'une même structure entre plusieurs documents, structures « standard » pour une communauté
- Vérification stricte et automatisable de la correction des documents

Voici un exemple d'un document XML et de son DTD :

```

<?xml version="1.0" encoding="ISO-8859-1" standalone="no" ?>
<!DOCTYPE document SYSTEM "accueil.dtd">
<document type='exemple'>
  <salutation> Bonjour! </salutation>
</document>

<!-- fichier accueil.dtd. Exemple de DTD simple -->

<!-- Définition de l'élément racine -->
<!ELEMENT document (salutation)>
<!-- Définition de l'attribut type pour l'élément document -->
<!ATTLIST document type CDATA #IMPLIED>

<!-- Un élément salutation ne contient que du texte -->
<!ELEMENT salutation (#PCDATA)>

```

XML

DTD

Les propriétés des éléments du DTD : les éléments, balise et attributs en xml sont représentés comme suite en DTD :

Les Sous-éléments : plusieurs façons de les combiner

- séquence : une balise chapitre est représenté end td comme suit :

```
<!ELEMENT chapitre (titre, intro, section)>
```

- Remarque: l'ordre des éléments est important

- alternative : l'ordre des balises est très important

```
<!ELEMENT chapitre (titre, intro, (section|sections))>
```

L'exemple précédent veut dire que dans le document xml on utilise section ou sections.

- indicateurs d'occurrence: \* (0-n), + (1-n), ? (0-1)

```
<!ELEMENT chapitre (titre*, intro?, section+)>
```

\*: 0 ou plusieurs. +:1 ou plusieurs. ?: optional

- Données :

```
texte : <!ELEMENT chapitre (#PCDATA) >
```

#PCDATA: chaîne de caractères.

Les attributs avec DTD : <ATTLIST nom-élément nom-attribut type-attribut declaration-default>

– Pour un élément donné on décrit la liste de ses attributs

– Chaque attribut: un nom, un type et une valeur par défaut

– Remarque: l'ordre des attributs n'est pas important

Exemple : <!ELEMENT ex (#PCDATA)>

<!ATTLIST cible ID #REQUIRED

Nb (1|2|3) '1'

Propriétaire CDATA #fixed 'moi' >

Valeur par défaut d'un attribut

- La valeur en question
- #REQUIRED : attribut obligatoire, valeur à être précisée dans le document
- #IMPLIED : attribut facultatif, valeur à être précisée dans le document
- #FIXED (suivi de la valeur) : valeur de l'attribut fixée pour tout élément instance

# Chapitre 4 :

# Java script (JS)

---

## 4. Introduction :

JavaScript est le langage de programmation le plus populaire au monde. JavaScript est le langage de programmation du Web. JavaScript est facile à apprendre. JavaScript peut modifier le contenu HTML. L'une des nombreuses méthodes JavaScript HTML est `getElementById()`. L'exemple ci-dessous "trouve" un élément HTML (avec `id="demo"`) et change le contenu de l'élément (`innerHTML`) en "Hello JavaScript":

- Ex : `document.getElementById("demo").innerHTML = "Hello JavaScript";`

### 4.1. Emplacement du JS :

En HTML, le code JavaScript est inséré entre les balises `<script>` et `</script>`

---

```
<script>
document.getElementById("demo").innerHTML = "My First JavaScript";
</script>
```

---

#### 4.1.1. Fonctions et événements JavaScript :

Une fonction en JavaScript est un bloc de code JavaScript, qui peut être exécuté lorsqu'il est "appelé". Par exemple, une fonction peut être appelée lorsqu'un **événement** se produit, comme lorsque l'utilisateur clique sur un bouton.

- JavaScript dans `<head>` ou `<body>`

Vous pouvez placer n'importe quel nombre de scripts dans un document HTML. Les scripts peuvent être placés dans le `<body>`, ou dans la `<head>` section d'une page HTML, ou dans les deux.

- **JavaScript dans <Head>** : Dans cet exemple, une fonction JavaScript est placée dans le `<head>` section d'une page HTML. La fonction est invoquée (appelée) lorsqu'un bouton est cliqué :

---

```
<!DOCTYPE html>
<html>
<head>
<script>
function myFunction() {
```

```
    document.getElementById("demo").innerHTML = "Paragraph changed.";
}
</script>
</head>
<body>
<h2>Demo JavaScript in Head</h2>
<p id="demo">A Paragraph</p>
<button type="button" onclick="myFunction()">Try it</button>
</body>
</html>
```

---

#### 4.1.2. JavaScript dans <body>

Dans cet exemple, un JavaScript fonctionest placé dans la <body>section d'une page HTML. La fonction est invoquée (appelée) lorsqu'un bouton est cliqué :

```
<!DOCTYPE html>
<html>
<body>
<h2>Demo JavaScript in Body</h2>
<p id="demo">A Paragraph</p>
<button type="button" onclick="myFunction()">Try it</button>
<script>
function myFunction() {
    document.getElementById("demo").innerHTML = "Paragraph changed.";
}
</script>
</body>
</html>
```

---

#### 4.1.3. JavaScript externe

Les scripts peuvent également être placés dans des fichiers externes :

```
function myFunction() {
    document.getElementById("demo").innerHTML = "Paragraph changed.";
}
```

---

Les scripts externes sont pratiques lorsque le même code est utilisé dans de nombreuses pages Web différentes. Les fichiers JavaScript ont l'extension de fichier **.js** . Pour utiliser un script externe, mettez le nom du fichier de script dans l' `src`attribut (source) d'une `<script>`balise :

- `<script src="myScript.js"></script>`

Vous pouvez placer une référence de script externe dans `<head>`ou `<body>`comme vous le souhaitez. Le script se comportera comme s'il se trouvait exactement à l'endroit où `<script>`se trouve la balise.

#### 4.1.4. Avantages de JavaScript externe

Placer des scripts dans des fichiers externes présente certains avantages :

- Il sépare HTML et code
- Il rend HTML et JavaScript plus faciles à lire et à maintenir
- Les fichiers JavaScript mis en cache peuvent accélérer le chargement des pages

Pour ajouter plusieurs fichiers de script à une page, utilisez plusieurs balises de script :

---

```
<script src="myScript1.js"></script>
<script src="myScript2.js"></script>
```

---

Références externes : Un script externe peut être référencé de 3 manières différentes :

- Avec une URL complète (une adresse Web complète)
- Avec un chemin de fichier (comme /js/)
- Sans aucun chemin

Cet exemple utilise une **URL complète** pour créer un lien vers myScript.js :

- `<script src="https://www.w3schools.com/js/myScript.js"></script>`

Cet exemple utilise un **chemin de fichier** pour créer un lien vers myScript.js :

- `<script src="/js/myScript.js"></script>`

Cet exemple n'utilise aucun chemin pour créer un lien vers myScript.js :

- `<script src="myScript.js"></script>`

#### 4.2. Output dans java script : Possibilités d'affichage JavaScript [1]

JavaScript peut "afficher" les données de différentes manières :

- Écrire dans un élément HTML, en utilisant innerHTML.
- Écrire dans la sortie HTML à l'aide de document.write().
- Écrire dans une boîte d'alerte, en utilisant window.alert().
- Écrire dans la console du navigateur, en utilisant console.log().

**4.2.1. Utiliser innerHTML :** Pour accéder à un élément HTML, JavaScript peut utiliser la document.getElementById(id)méthode.

L' id attribut définit l'élément HTML. La innerHTML propriété définit le contenu HTML :

---

```
<!DOCTYPE html>
<html>
<body>
<h1>My First Web Page</h1>
<p>My First Paragraph</p>
```

```
<p id="demo"></p>
<script>
document.getElementById("demo").innerHTML = 5 + 6;
</script>
</body>
```

---

**4.2.2. Utilisation de document.write() :** A des fins de test, il est pratique d'utiliser document.write():

---

```
<!DOCTYPE html>
<html>
<body>
<h1>My First Web Page</h1>
<p>My first paragraph.</p>
<script>
document.write(5 + 6);
</script>
</body>
</html>
```

---

### 4.3. Instructions JavaScript

Un **programme informatique** est une liste "d'instructions" à "exécuter" par un ordinateur. Dans un langage de programmation, ces instructions de programmation sont appelées **instructions** . Un **programme JavaScript** est une liste d' **instructions** de programmation [17] .

---

```
let x, y, z; // Statement 1
x = 5;      // Statement 2
y = 6;      // Statement 3
z = x + y;  // Statement 4
```

---

Les instructions JavaScript sont composées de :

- Valeurs, opérateurs, expressions, mots-clés et commentaires. Cette instruction indique au navigateur d'écrire "Hello Dolly". dans un élément HTML avec id="demo":
- document.getElementById("demo").innerHTML = "Hello Dolly.";

La plupart des programmes JavaScript contiennent de nombreuses instructions JavaScript. Les instructions sont exécutées, une par une, dans le même ordre qu'elles sont écrites. Points-virgules ; Les points-virgules séparent les instructions JavaScript [14]. Ajoutez un point-virgule à la fin de chaque instruction exécutable :

---

```
let a, b, c; // Declare 3 variables
a = 5;      // Assign the value 5 to a
b = 6;      // Assign the value 6 to b
c = a + b;  // Assign the sum of a and b to c
```

---

Lorsqu'elles sont séparées par des points-virgules, plusieurs instructions sur une même ligne sont autorisées :

- a = 5; b = 6; c = a + b;

Espace blanc JavaScript : JavaScript ignore plusieurs espaces. Vous pouvez ajouter un espace blanc à votre script pour le rendre plus lisible.

**Les lignes suivantes sont équivalentes :**

- let person = "Hege";
- let person="Hege";

Longueur de ligne JavaScript et sauts de ligne : Pour une meilleure lisibilité, les programmeurs aiment souvent éviter les lignes de code de plus de 80 caractères. Si une instruction JavaScript ne tient pas sur une seule ligne, le meilleur endroit pour la casser est après un opérateur :

- document.getElementById("demo").innerHTML = "Hello Dolly!";

## Syntaxe JavaScript

---

```
// How to create variables:  
var x;  
let y;  
// How to use variables:  
x = 5;  
y = 6;  
let z = x + y;
```

---

Valeurs JavaScript : La syntaxe JavaScript définit deux types de valeurs :

- Valeurs fixes
  - Valeurs variables
- Les valeurs fixes sont appelées **Literals** et les valeurs variables sont appelées **Variables Littéraux JavaScript**. Les deux règles de syntaxe les plus importantes pour les valeurs fixes sont : **Les nombres** s'écrivent avec ou sans décimales :

4.4. **Variables JavaScript** : Dans un langage de programmation, **les variables** sont utilisées pour **stocker** des valeurs de données. JavaScript utilise les mots clés **var** et **let** pour **déclarer** les variables. Un signe égal est utilisé pour attribuer des valeurs aux variables. Dans cet exemple, x est défini comme une variable. Ensuite, x est affecté (donné) la valeur 6 :

```
let x;  
x = 6;
```

4.5. **Opérateurs JavaScript** : JavaScript utilise des opérateurs arithmétiques (+ - \* /) pour calculer les valeurs :

(5 + 6) \* 10

JavaScript utilise un **opérateur d'affectation** (=) pour **affecter des** valeurs aux variables :

---

```
let x, y;  
x = 5;  
y = 6;
```

---

**4.6. Expressions JavaScript :** Une expression est une combinaison de valeurs, de variables et d'opérateurs, qui calcule une valeur. Le calcul s'appelle une évaluation [18].

Par exemple,  $5 * 10$  est évalué à 50 :

-  $5 * 10$

Les expressions peuvent également contenir des valeurs variables :

-  $x * 10$

Les valeurs peuvent être de différents types, tels que des nombres et des chaînes. Par exemple, "John" + " " + "Doe", est évalué à "John Doe":

- "John" + " " + "Doe"

**4.7. Mots-clés JavaScript :** Les **mots-clés** JavaScript sont utilisés pour identifier les actions à effectuer. Le mot **let** clé indique au navigateur de créer des variables :

---

```
let x, y;  
x = 5 + 6;  
y = x * 10;
```

---

Le mot **var** clé indique également au navigateur de créer des variables :

---

```
var x, y;  
x = 5 + 6;  
y = x * 10;
```

---

#### **4.8. Commentaires JavaScript**

Toutes les instructions JavaScript ne sont pas "exécutées". Le code après les doubles barres obliques //ou entre /\*et \*/est traité comme un commentaire. Les commentaires sont ignorés et ne seront pas exécutés [19] :

#### **4.9. Opérateurs JavaScript :**

**4.9.1.** L'opérateur d' **addition** + ( ) ajoute des nombres :

---

```
let x = 5;  
let y = 2;  
let z = x + y;
```

---

**4.9.2.** L'opérateur de **multiplication**\* ( ) multiplie les nombres.

---

```
let x = 5;  
let y = 2;  
let z = x * y;
```

---

**4.9.3. Types d'opérateurs JavaScript Il existe différents types d'opérateurs JavaScript :**

- Opérateurs arithmétiques
- Opérateurs d'affectation
- Opérateurs de comparaison
- Opérateurs logiques
- Opérateurs conditionnels
- Opérateurs de type

Opérateurs arithmétiques JavaScript : Les opérateurs arithmétiques sont utilisés pour effectuer des opérations arithmétiques sur des nombres [20] :

Operator	Description
+	Addition
-	Subtraction
*	Multiplication
**	Exponentiation ( <a href="#">ES2016</a> )
/	Division
%	Modulus (Division Remainder)
++	Increment
--	Decrement

Opérateurs d'affectation JavaScript : Les opérateurs d'affectation attribuent des valeurs aux variables JavaScript.

Operator	Example	Same As
=	x = y	x = y
+=	x += y	x = x + y
-=	x -= y	x = x - y
*=	x *= y	x = x * y
/=	x /= y	x = x / y
%=	x %= y	x = x % y
**=	x **= y	x = x ** y

## 4.10 déclarations

### 4.10.1. Si Déclarations

Si l'instruction est utilisée pour vérifier une condition et prendre une décision en fonction du résultat de la condition. Il peut choisir une action lorsque la condition est vraie et également prendre une action lorsqu'elle est fausse. Vous pouvez imbriquer plusieurs conditions ensemble et décider de choisir une action en fonction de plusieurs conditions [20].

Syntaxe :

---

```

if ( condition1 ){
// Statements
} else if( condition1 || condition2 ){
// Statements
} else {
// Statements
}

```

---

#### Example 1:

---

```

if( marks > 35 ){
alert("You are Passed!");
}else{ alert("Try Again!"); }

```

Example 2:

```

if( marks > 35 && marks < 60 ){
alert("You are Passed with Grade C");
}else if ( marks > 60 && marks < 80 ){
alert("You are Passed with Grade B");
}else if ( marks > 80 ){

```

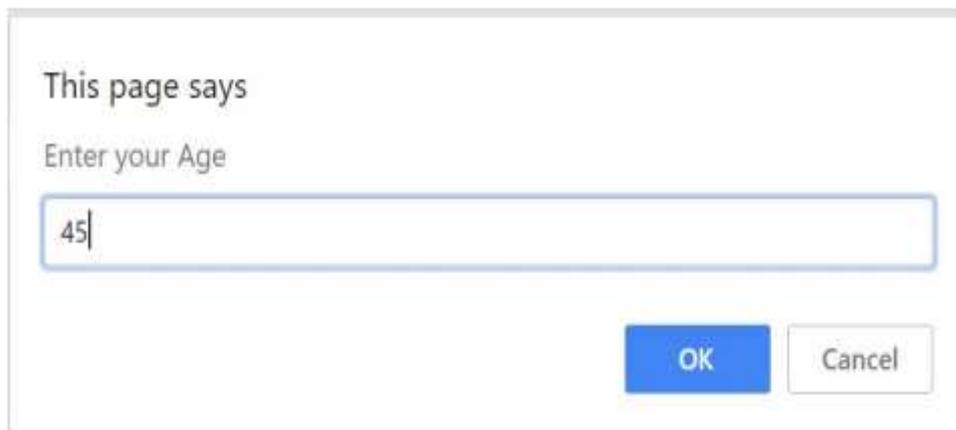
```
alert("You are Passed with Grade A");  
}else{  
alert("Try Again!");}
```

---

**Exercise 20:** Accept number 1 to 10 and check if the user entered correctly between 1 to 10 as requested with the if Statement.



**Exercise 21:** Ask user to enter the age and decides if he is kid, man (age > 21) or senior citizen (age > 55).



#### 4.10.2. switch case

l'instruction switch est utilisée pour vérifier une valeur et prendre une décision en fonction du résultat de la correspondance de valeur. Syntaxe: `switch( variable ){ case value: //Statement break;`

Syntaxe:

---

```
switch( variable ){  
case value:  
//Statement  
break;  
case value:  
//Statement
```

```
break;
default:
//Statement
break;
}
```

---

Example 1:

---

```
switch ( dayOfWeek ){
case 'Mon':
alert("Welcome Monday");
break;
case 'Tuesday':
alert("Welcome Tuesday");
break;
default:
alert("Try Again!");
break;
}
```

---

#### 4.10.3 Déclarations while

Les instructions While sont utilisées pour boucler un code de bloc et l'exécuter jusqu'à ce qu'une condition soit remplie. Exécution du même bloc de code jusqu'à ce que la condition soit satisfaite.

Syntaxe:

---

```
while( condition ) {
//Statements
}
```

---

Example 1: var counter = 0; while ( counter <= 10 ){ counter++; }

**Exercise 22:** Accept two number from user and print all the numbers between them. Print only maximum of 10 numbers. break; – is a keyword that can be used in the loop to break from the loop. continue; – is a keyword that can be used in the loop to continue the loop by skipping the below statements. Tip: if( counter >= 10){ break; }

**Exercise 23:** Accept two number from user and print EVEN numbers only between them. Print only maximum of 10 numbers.

#### 4.10.4 pour statement

for Loop sont utilisés pour boucler un code de bloc et l'exécuter jusqu'à ce qu'une condition soit remplie. Exécuter le même bloc de code jusqu'à ce que la condition soit satisfaite

Syntax: for( counter initialization; condition; increments ) { //Statements }

# Chapitre 5 :

# Preprocessing

# hypertext (PHP)

---

## 5. Introduction :

Pour exécuter PHP sur votre système local, vous devez installer HTTP Web Server. PHP n'est pas comme Java ou C où vous pouvez installer les bibliothèques et exécuter PHP à l'invite de commande. PHP est une programmation côté serveur et utilisé dans le web. Il est conçu pour fonctionner à partir d'un serveur Web comme IIS ou Apache HTTP Server [20].

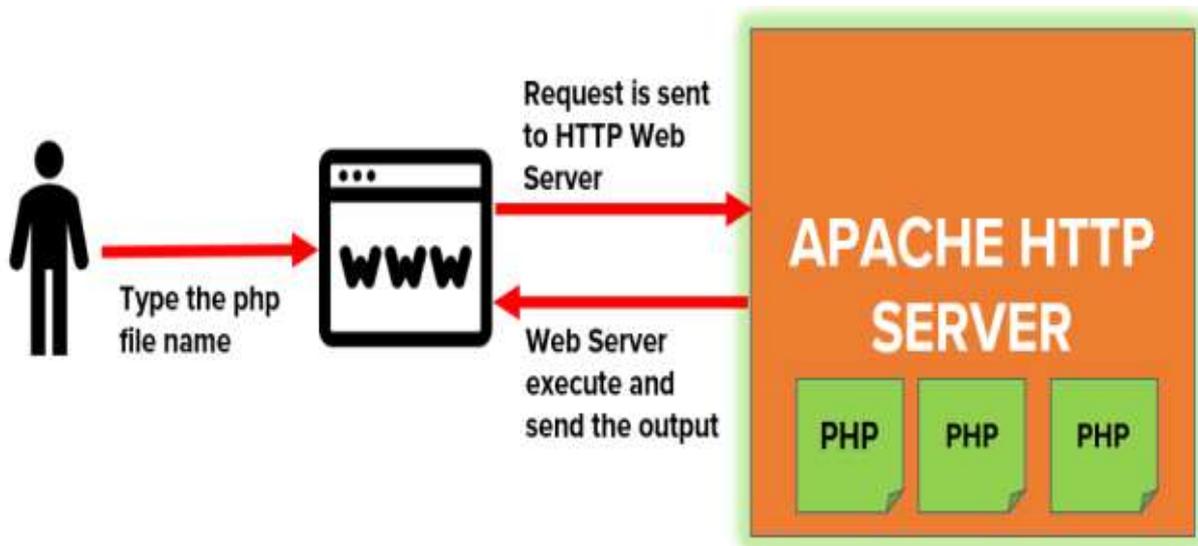


Figure 14 : fonctionnement du serveur local Apach

### 5.1. Comment fonctionne PHP ?

Lorsque l'utilisateur saisit le chemin filename.php dans l'URL du navigateur :

Le navigateur ira au serveur sur lequel les serveurs HTTP sont en cours d'exécution. Web Server écoutera la demande. Il exécutera le code PHP sur le serveur. Prenez la sortie générée par le serveur PHP. Renvoyez-le au navigateur. Le navigateur affiche la sortie [19].

### 5.1.1. Choses à noter :

Vous avez besoin d'un serveur Web pour exécuter du code PHP. Le serveur exécute php sur le serveur et renvoie la sortie du code. Vous ne pourrez pas voir le code PHP sur le navigateur. Le navigateur transmet les données utilisateur du navigateur au serveur et récupère les données du serveur. L'insertion dans la base de données, l'envoi d'e-mails, la vérification des informations d'identification de connexion sont toutes effectuées par le programme php sur le serveur. Personne ne peut voir votre code php [16].

### 5.1.2. Pourquoi avoir besoin d'un serveur Web ?

**Exercice 24:** Créer un exemple de nom de fichier index.php. Ajoutez le code PHP suivant.

```
<?php echo "Bonjour PHP !" ; ?>
```

Ouvrez le fichier index.php dans le navigateur. Regardez comment le navigateur affiche simplement le code PHP. Le navigateur ne comprend pas PHP ni comment l'interpréter. C'est le travail d'un serveur Web. Le serveur Web interprète le code PHP et envoie la sortie au navigateur. C'est la raison pour laquelle vous avez besoin d'un serveur Web pour exécuter du code PHP [12].

### 5.1.3. Comprendre le serveur WAMP

Vous pouvez choisir d'installer les logiciels suivants individuellement :

Apache – <https://httpd.apache.org/>

- PHP – <http://php.net/software.php>
- MySQL – <https://www.mysql.com/>
- phpMyAdmin – <https://www.phpmyadmin.net/>

Apache est le serveur Web. PHP est la bibliothèque qui aide à exécuter le code php. MySQL est la base de données pour stocker les données. phpMyAdmin est le client d'administration pour accéder à votre base de données [17]. Au lieu d'installer tous les logiciels, il existe un package qui combine tous les logiciels et donne un environnement Web pour créer des applications Web.

## 5.2. Premier programme PHP

Le nom du fichier php doit se terminer par l'extension ".php". Vous devez écrire le code php avec dans ces symboles de début et de fin.

- <?php
- ?>
- <?php – tells server to interpret the code from here
- ?> – server will stop interpreting the code.

Vous devez écrire le code PHP à l'intérieur de ce bloc

---

```
<?php  
//PHP CODE
```

---

---

?>

---

### SYNTAXE:

---

```
<body>
<?php
?>
</body>
```

---

Toutes les déclarations que vous écrivez entre le bloc php doivent se terminer par ";"

#### Étape 1 : Créer le fichier php

- Aller au dossier : C:\wamp64\www
- Créer un dossier : \php\basics\echo\ex1
- Dossier final : C:\wamp64\www\php\basics\echo\ex1
- Créer un fichier avec le nom : index.php
- Dossier : C:\wamp64\www\php\basics\echo\ex1

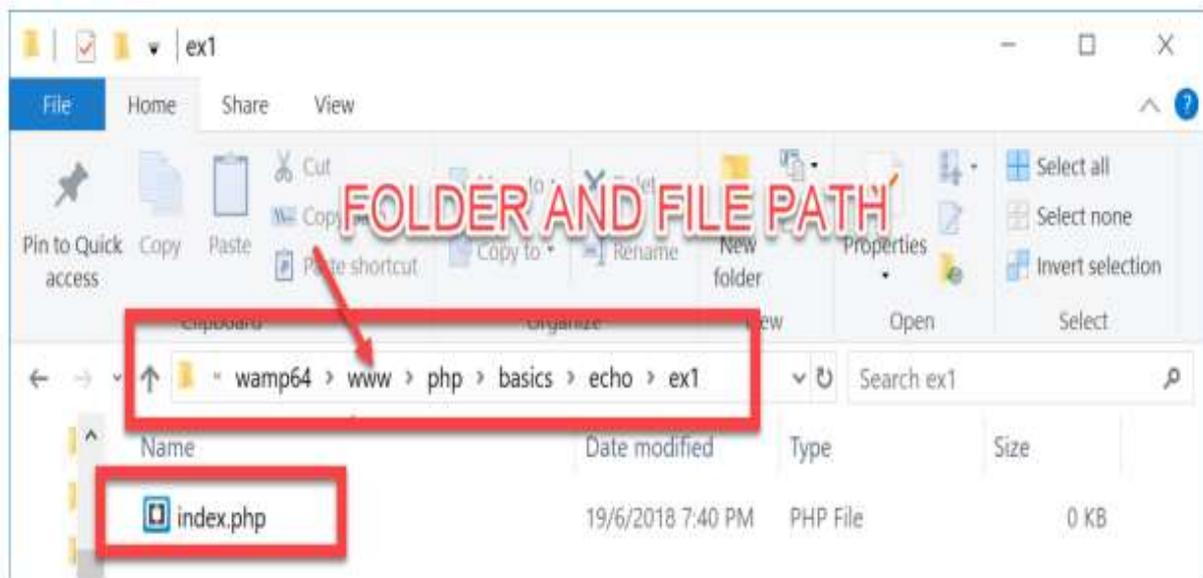


Figure 15: répertoire du logiciel WAMP

#### Étape 2 : Ajouter le code dans index.php

Vous pouvez utiliser le logiciel Brackets pour ouvrir le fichier.

---

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initialscale=1.0">
<meta name="description" content="Page Description">
<title>Page Title</title>
</head>
```

---

---

```
<body>
<h1>Heading</h1>
<p>Paragraph Text</p>
<!-- PHP Code -->
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initialscale=1.0">
<meta name="description" content="Page Description">
<title>Page Title</title>
</head>
<body>
<h1>Heading</h1>
<p>Paragraph Text</p>
<!-- PHP Code -->
```

---

### 5.2.1. Commentaires en PHP

Un commentaire dans le code PHP est une ligne qui n'est pas exécutée dans le cadre du programme. Son seul but est d'être lu par quelqu'un qui regarde le code. Les commentaires peuvent être utilisés pour :

Rappelez-vous ce que vous avez fait - La plupart des programmeurs ont dû revenir à leur propre travail un an ou deux plus tard et avoir à repenser ce qu'ils avaient fait. Les commentaires peuvent vous rappeler ce à quoi vous pensiez lorsque vous avez écrit le code PHP prend en charge plusieurs façons de commenter :

---

```
<!DOCTYPE html>
<html>
<body>
<?php
// This is a single-line comment
# This is also a single-line comment
?>
</body>
</html>
```

---

5.3. **Variables PHP :** Les variables sont des "conteneurs" pour stocker des informations. Créer (déclarer) des variables PHP. En PHP, une variable commence par le \$ signe, suivi du nom de la variable [18]:

---

```
<?php
$txt = "Hello world!";
$x = 5;
$y = 10.5;
?>
```

---

Après l'exécution des instructions ci-dessus, la variable \$txt contiendra la valeur Hello world!, la variable \$x contiendra la valeur 5 et la variable \$y contiendra la valeur 10.5.

**Remarque :** Lorsque vous affectez une valeur de texte à une variable, placez des guillemets autour de la valeur [17].

Une variable peut avoir un nom court (comme x et y) ou un nom plus descriptif (age, carname, total\_volume). Règles pour les variables PHP :

- Une variable commence par le \$ signe suivi du nom de la variable
- Un nom de variable doit commencer par une lettre ou le caractère de soulignement
- Un nom de variable ne peut pas commencer par un chiffre
- Un nom de variable ne peut contenir que des caractères alphanumériques et des traits de soulignement (Az, 0-9 et \_)
- Les noms de variables sont sensibles à la casse ( \$age et \$AGE sont deux variables différentes)

### 5.3.1. Variables de sortie

L'instruction PHP echo est souvent utilisée pour afficher des données à l'écran. L'exemple suivant montre comment afficher du texte et une variable [16]:

---

```
<?php
$txt = "W3Schools.com";
echo "I love $txt!";
?>
```

---

L'exemple suivant produira le même résultat que l'exemple ci-dessus :

---

```
<?php
$txt = "W3Schools.com";
echo "I love " . $txt . "!";
?>
```

---

L'exemple suivant affichera la somme de deux variables :

---

```
<?php
$x = 5;
$y = 4;
echo $x + $y;
?>
```

---

**5.3.2. PHP est un langage faiblement typé :** Dans l'exemple ci-dessus, notez que nous n'avons pas eu à indiquer à PHP le type de données de la variable. PHP associe automatiquement un type de données à la variable, en fonction de sa valeur. Étant donné que les types de données ne sont pas définis au sens strict, vous pouvez par exemple ajouter une chaîne à un entier sans provoquer d'erreur. En PHP 7, les déclarations de type ont été ajoutées. Cela donne une option pour spécifier le type de données attendu lors de la déclaration d'une fonction, et en activant l'exigence stricte, il lancera une "erreur fatale" sur une incompatibilité de type [11].

5.4. **Écho PHP et print** : Avec PHP, il existe deux méthodes de base pour obtenir une sortie : echo et print. Dans cette partie, nous utilisons echo ou print dans presque tous les exemples. Ainsi, ce chapitre contient un peu plus d'informations sur ces deux déclarations de sortie. Echo et print sont plus ou moins les mêmes. Ils sont tous deux utilisés pour afficher des données à l'écran. Les différences sont minimes : echo n'a pas de valeur de retour alors qu'il print une valeur de retour de 1, il peut donc être utilisé dans des expressions. echo peut prendre plusieurs paramètres (bien qu'une telle utilisation soit rare) alors qu'il print peut prendre un argument. echo est légèrement plus rapide que print [14].

5.4.1. **La déclaration d'écho PHP** : L'echo instruction peut être utilisée avec ou sans parenthèses : echo ou echo(). L'exemple suivant montre comment générer du texte avec la echo commande (notez que le texte peut contenir du balisage HTML) :

---

```
<?php
echo "<h2>PHP is Fun!</h2>";
echo "Hello world!<br>";
echo "I'm about to learn PHP!<br>";
echo "This ", "string ", "was ", "made ", "with multiple parameters.";
?>
```

---

5.4.2. **Variables d'affichage** : L'exemple suivant montre comment générer du texte et des variables avec l'echo instruction :

---

```
<?php
$txt1 = "Learn PHP";
$txt2 = "W3Schools.com";
$x = 5;
$y = 4;
echo "<h2>" . $txt1 . "</h2>";
echo "Study PHP at " . $txt2 . "<br>";
echo $x + $y;
?>
```

---

5.4.3. **La déclaration d'impression PHP**

L'print instruction peut être utilisée avec ou sans parenthèses : print ou print(). L'exemple suivant montre comment générer du texte avec la print commande (notez que le texte peut contenir du balisage HTML) :

---

```
<?php
print "<h2>PHP is Fun!</h2>";
print "Hello world!<br>";
print "I'm about to learn PHP!";
?>
```

---

5.5. **Variables d'affichage print** : L'exemple suivant montre comment générer du texte et des variables avec l'print instruction :

---

```
<?php
$txt1 = "Learn PHP";
$txt2 = "W3Schools.com";
$x = 5;
$y = 4;
print "<h2>" . $txt1 . "</h2>";
print "Study PHP at " . $txt2 . "<br>";
print $x + $y;
?>
```

---

**5.6. Types de données PHP :** Les variables peuvent stocker des données de différents types, et différents types de données peuvent faire différentes choses. PHP prend en charge les types de données suivants :

- Chaîne de caractères
- Entier
- Float (nombres à virgule flottante - également appelés doubles)
- booléen
- Déployer
- Objet
- NUL
- Ressource
- Chaîne PHP

**5.6.1. Chaîne de caractères :** Une chaîne est une séquence de caractères, comme "Hello world!". Une chaîne peut être n'importe quel texte entre guillemets. Vous pouvez utiliser des guillemets simples ou doubles :

---

```
<?php
$x = "Hello world!";
$y = 'Hello world!';
echo $x;
echo "<br>";
echo $y;
?>
```

---

**5.6.2. Entier PHP :** Un type de données entier est un nombre non décimal compris entre - 2 147 483 648 et 2 147 483 647.

**5.6.2.1. Règles pour les entiers :**

- Un entier doit avoir au moins un chiffre
- Un entier ne doit pas avoir de point décimal
- Un entier peut être positif ou négatif
- Les nombres entiers peuvent être spécifiés en notation décimale (base 10), hexadécimale (base 16), octale (base 8) ou binaire (base 2)

Dans l'exemple suivant, \$x est un entier. La fonction PHP `var_dump()` renvoie le type de données et la valeur :

---

```
<?php
$x = 5985;
var_dump($x);
?>
```

---

### 5.6.3. Float PHP

Un flottant (nombre à virgule flottante) est un nombre avec un point décimal ou un nombre sous forme exponentielle. Dans l'exemple suivant, \$x est un flottant. La fonction PHP var\_dump() renvoie le type de données et la valeur :

---

```
<?php
$x = 10.365;
var_dump($x);
?>
```

---

**5.6.4. PHP booléen :** Un booléen représente deux états possibles : VRAI ou FAUX.

```
$x = true;
$y = false;
```

Les booléens sont souvent utilisés dans les tests conditionnels. Vous en apprendrez plus sur les tests conditionnels dans un chapitre ultérieur de ce didacticiel.

**5.6.5. Tableau PHP :** Un tableau stocke plusieurs valeurs dans une seule variable. Dans l'exemple suivant, \$cars est un tableau. La fonction PHP var\_dump() renvoie le type de données et la valeur :

---

```
<?php
$cars = array("Volvo","BMW","Toyota");
var_dump($cars);
?>
```

---

**5.6.6. Objet PHP :** Les classes et les objets sont les deux principaux aspects de la programmation orientée objet. Une classe est un modèle pour les objets et un objet est une instance d'une classe. Lorsque les objets individuels sont créés, ils héritent de toutes les propriétés et comportements de la classe, mais chaque objet aura des valeurs différentes pour les propriétés. Supposons que nous ayons une classe nommée Car. Une voiture peut avoir des propriétés comme model, color, etc. Nous pouvons définir des variables comme \$model, \$color, etc., pour contenir les valeurs de ces propriétés. Lorsque les objets individuels (Volvo, BMW, Toyota, etc.) sont créés, ils héritent de toutes les propriétés et comportements de la classe, mais chaque objet aura des valeurs différentes pour les propriétés. Si vous créez une fonction \_\_construct(), PHP appellera automatiquement cette fonction lorsque vous créez un objet à partir d'une classe.

---

```
<?php
class Car {
    public $color;
    public $model;
```

```

public function __construct($color, $model) {
    $this->color = $color;
    $this->model = $model;
}
public function message() {
    return "My car is a " . $this->color . " " . $this->model . "!";
}
}

$myCar = new Car("black", "Volvo");
echo $myCar -> message();
echo "<br>";
$myCar = new Car("red", "Toyota");
echo $myCar -> message();
?>

```

---

**5.6.7. Valeur NULL PHP :** Null est un type de données spécial qui ne peut avoir qu'une seule valeur : NULL. Une variable de type de données NULL est une variable à laquelle aucune valeur n'est affectée. **Conseil :** Si une variable est créée sans valeur, la valeur NULL lui est automatiquement affectée. Les variables peuvent également être vidées en définissant la valeur sur NULL :

```

<?php
$x = "Hello world!";
$x = null;
var_dump($x);
?>

```

---

5.7. **Instructions conditionnelles PHP:** Très souvent, lorsque vous écrivez du code, vous souhaitez effectuer différentes actions pour différentes conditions. Vous pouvez utiliser des instructions conditionnelles dans votre code pour ce faire. En PHP, nous avons les instructions conditionnelles suivantes :

- ifinstruction - exécute du code si une condition est vraie
- if...elseinstruction - exécute du code si une condition est vraie et un autre code si cette condition est fausse
- if...elseif...elsedéclaration - exécute des codes différents pour plus de deux conditions
- switchinstruction - sélectionne l'un des nombreux blocs de code à exécuter

**PHP - L'instruction if :** L'instruction if exécute du code si une condition est vraie.

### Syntaxe

```

if (condition) {
    code to be executed if condition is true;
}

```

---

## Exemple

Sortie "Bonne journée !" si l'heure actuelle (HOUR) est inférieure à 20 :

---

```
<?php
$t = date("H");
if ($t < "20") {
    echo "Have a good day!";
}
?>
```

---

**PHP - L'instruction if...else :** L' if...else instruction exécute du code si une condition est vraie et un autre code si cette condition est fausse.

## Syntaxe

---

```
if (condition) {
    code to be executed if condition is true;
} else {
    code to be executed if condition is false;
}
```

---

**Exemple :** Sortie "Bonne journée !" si l'heure actuelle est inférieure à 20, et "Bonne nuit !" Par ailleurs

---

```
<?php
$t = date("H");
if ($t < "20") {
    echo "Have a good day!";
} else {
    echo "Have a good night!";
}
?>
```

---

PHP - L'instruction if...elseif...else : L' if...elseif...else instruction exécute des codes différents pour plus de deux conditions.

## Syntaxe

---

```
if (condition) {
    code to be executed if this condition is true;
} elseif (condition) {
    code to be executed if first condition is false and this condition is true;
} else {
    code to be executed if all conditions are false;
}
```

---

## Exemple

Sortie "Bonjour !" si l'heure actuelle est inférieure à 10, et "Bonne journée !" si l'heure actuelle est inférieure à 20. Sinon, il affichera "Bonne nuit !" :

---

```
<?php
$t = date("H");
if ($t < "10") {
    echo "Have a good morning!";
} elseif ($t < "20") {
    echo "Have a good day!";
} else {
    echo "Have a good night!";
}
?>
```

---

**4.8. Boucles PHP :** Souvent, lorsque vous écrivez du code, vous souhaitez que le même bloc de code s'exécute encore et encore un certain nombre de fois. Ainsi, au lieu d'ajouter plusieurs lignes de code presque égales dans un script, nous pouvons utiliser des boucles. Les boucles sont utilisées pour exécuter le même bloc de code encore et encore, tant qu'une certaine condition est vraie. **En PHP, nous avons les types de boucles suivants :**

- while- parcourt un bloc de code tant que la condition spécifiée est vraie
- do...while- boucle une fois sur un bloc de code, puis répète la boucle tant que la condition spécifiée est vraie
- for- parcourt un bloc de code un nombre de fois spécifié
- foreach- boucle à travers un bloc de code pour chaque élément d'un tableau

#### 4.8.1. La boucle PHP pour

La forboucle est utilisée lorsque vous savez à l'avance combien de fois le script doit s'exécuter.

##### Syntaxe

---

```
for (init counter; test counter; increment counter) {
    code to be executed for each iteration;
}
```

---

Paramètres:

- *init counter* : Initialise la valeur du compteur de boucle
- *compteur de test* : Évalué à chaque itération de boucle. S'il est évalué à TRUE, la boucle continue. S'il est évalué à FALSE, la boucle se termine.
- *compteur d'incrément* : augmente la valeur du compteur de boucle

##### Exemples

L'exemple ci-dessous affiche les nombres de 0 à 10 :

---

```
<?php
for ($x = 0; $x <= 10; $x++) {
    echo "The number is: $x <br>";
}
?>
```

---

## Exemple expliqué

- `$x = 0` ; - Initialiser le compteur de boucle (`$x`), et mettre la valeur de départ à 0
- `$x <= 10` ; - Continuer la boucle tant que `$x` est inférieur ou égal à 10
- `$x++` - Augmente la valeur du compteur de boucle de 1 pour chaque itération

Cet exemple compte jusqu'à 100 par dizaines :

---

```
<?php
for ($x = 0; $x <= 100; $x+=10) {
    echo "The number is: $x <br>";
}
?>
```

---

## Exemple expliqué

- `$x = 0` ; - Initialiser le compteur de boucle (`$x`), et mettre la valeur de départ à 0
- `$x <= 100` ; - Continuer la boucle tant que `$x` est inférieur ou égal à 100
- `$x+=10` - Augmente la valeur du compteur de boucle de 10 pour chaque itération

### 4.8.2. La boucle PHP foreach

La foreachboucle ne fonctionne que sur les tableaux et est utilisée pour parcourir chaque paire clé/valeur d'un tableau.

## Syntaxe

---

```
foreach ($array as $value) {
    code to be executed;
}
```

---

Pour chaque itération de boucle, la valeur de l'élément courant du tableau est affectée à `$value` et le pointeur du tableau est déplacé de un, jusqu'à ce qu'il atteigne le dernier élément du tableau.

## Exemples

L'exemple suivant affichera les valeurs du tableau donné (`$colors`) :

---

```
<?php
$colors = array("red", "green", "blue", "yellow");

foreach ($colors as $value) {
    echo "$value <br>";
}
?>
```

---

L'exemple suivant affichera à la fois les clés et les valeurs du tableau donné (`$age`) :

---

```
<?php
$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");
foreach($age as $x => $val) {
    echo "$x = $val<br>";
}
?>
```

---

## 4.9. PHP Pause et Continuer

4.9.1. **Pause PHP :** Vous avez déjà vu l'breakinstruction utilisée dans un chapitre précédent de ce didacticiel. Il a été utilisé pour "sauter" d'une switchdéclaration. L'breakinstruction peut également être utilisée pour sortir d'une boucle. Cet exemple sort de la boucle lorsque x est égal à 4 :

---

```
<?php
for ($x = 0; $x < 10; $x++) {
    if ($x == 4) {
        break;
    }
    echo "The number is: $x <br>";
}
?>
```

---

## 4.9.2. PHP Continuer

L'continueinstruction interrompt une itération (dans la boucle), si une condition spécifiée se produit, et continue avec l'itération suivante dans la boucle. Cet exemple ignore la valeur de 4 :

---

```
<?php
for ($x = 0; $x < 10; $x++) {
    if ($x == 4) {
        continue;
    }
    echo "The number is: $x <br>";
}
?>
```

---

Interrompre et continuer dans la boucle While, Vous pouvez également utiliser breaket continuedans les whileboucles :

---

```
<?php
$x = 0;
while($x < 10) {
    if ($x == 4) {
        break;
    }
    echo "The number is: $x <br>";
    $x++;
}
```

```
}  
?>
```

---

### Continuer Exemple

---

```
<?php  
$x = 0;  
while($x < 10) {  
    if ($x == 4) {  
        $x++;  
        continue;  
    }  
    echo "The number is: $x <br>";  
    $x++;  
}  
?>
```

---

## 4.10. Fonctions PHP

La vraie puissance de PHP vient de ses fonctions. PHP a plus de 1000 fonctions intégrées, et en plus vous pouvez créer vos propres fonctions personnalisées.

**4.10.1. Fonctions intégrées PHP :** PHP a plus de 1000 fonctions intégrées qui peuvent être appelées directement, depuis un script, pour effectuer une tâche spécifique. Veuillez consulter notre référence PHP pour un aperçu complet des [fonctions intégrées de PHP](#) .

**4.10.2. Fonctions PHP définies par l'utilisateur :** Outre les fonctions PHP intégrées, il est possible de créer vos propres fonctions.

- Une fonction est un bloc d'instructions qui peut être utilisé à plusieurs reprises dans un programme.
- Une fonction ne s'exécute pas automatiquement lors du chargement d'une page.
- Une fonction sera exécutée par un appel à la fonction.

Créer une fonction définie par l'utilisateur en PHP. Une déclaration de fonction définie par l'utilisateur commence par le mot `function`:

### Syntaxe

---

```
function functionName() {  
    code to be executed;  
}
```

---

Dans l'exemple ci-dessous, nous créons une fonction nommée "writeMsg()". L'accolade ouvrante ({} ) indique le début du code de la fonction et l'accolade fermante ({} ) indique la fin de la fonction. La fonction affiche "Hello world!". Pour appeler la fonction, écrivez simplement son nom suivi de crochets ( ):

---

```
<?php  
function writeMsg() {  
    echo "Hello world!";
```

```
}  
writeMsg(); // call the function  
>
```

---

**Arguments de la fonction PHP :** Les informations peuvent être transmises aux fonctions via des arguments. Un argument est comme une variable. Les arguments sont spécifiés après le nom de la fonction, entre parenthèses. Vous pouvez ajouter autant d'arguments que vous le souhaitez, séparez-les simplement par une virgule. L'exemple suivant a une fonction avec un argument (\$fname). Lorsque la fonction familyName() est appelée, nous transmettons également un nom (par exemple Jani), et le nom est utilisé à l'intérieur de la fonction, qui affiche plusieurs prénoms différents, mais un nom de famille égal :

---

```
<?php  
function familyName($fname) {  
    echo "$fname Refsnes.<br>";  
}  
familyName("Jani");  
familyName("Hege");  
familyName("Stale");  
familyName("Kai Jim");  
familyName("Borge");  
>
```

---

L'exemple suivant a une fonction avec deux arguments (\$fname et \$year) :

---

```
<?php  
function familyName($fname, $year) {  
    echo "$fname Refsnes. Born in $year <br>";  
}  
familyName("Hege", "1975");  
familyName("Stale", "1978");  
familyName("Kai Jim", "1983");  
>
```

---

## 4.11. Tableaux PHP

Un tableau stocke plusieurs valeurs dans une seule variable :

---

```
<?php  
$cars = array("Volvo", "BMW", "Toyota");  
echo "I like " . $cars[0] . ", " . $cars[1] . " and " . $cars[2] . ".";  
>
```

---

### 4.11.1. Qu'est-ce qu'un tableau ?

Un tableau est une variable spéciale, qui peut contenir plusieurs valeurs à la fois. Si vous avez une liste d'éléments (une liste de noms de voitures, par exemple), le stockage des voitures dans des variables uniques pourrait ressembler à ceci :

```
$cars1 = "Volvo";  
$cars2 = "BMW";  
$cars3 = "Toyota";
```

Cependant, que se passe-t-il si vous voulez parcourir les voitures et en trouver une en particulier ? Et si vous n'aviez pas 3 voitures, mais 300 ? La solution est de créer un tableau ! Un tableau peut contenir plusieurs valeurs sous un seul nom, et vous pouvez accéder aux valeurs en vous référant à un numéro d'index.

**Créer un tableau en PHP :** En PHP, la fonction `array()` est utilisée pour créer un tableau :

```
array();
```

En PHP, il existe trois types de tableaux :

- **Tableaux indexés** - Tableaux avec un index numérique
- **Tableaux associatifs** - Tableaux avec des clés nommées
- **Tableaux multidimensionnels** - Tableaux contenant un ou plusieurs tableaux

Obtenir la longueur d'un tableau - La fonction `count()`. La `count()` fonction est utilisée pour renvoyer la longueur (le nombre d'éléments) d'un tableau :

---

```
<?php  
$cars = array("Volvo", "BMW", "Toyota");  
echo count($cars);  
?>
```

---

# Chapitre 6 :

# My SQL avec PHP

---

## 6. Introduction

MySQL est un système de base de données utilisé sur le Web qui s'exécute sur un serveur. Il est idéal pour les petites et les grandes applications, très rapide, fiable et facile à utiliser. Il utilise le SQL standard, compile sur un certain nombre de plates-formes, gratuit à télécharger et à utiliser. MySQL est développé, distribué et pris en charge par Oracle Corporation. Il porte le nom de la fille du co-fondateur Monty Widenius : My. Les données d'une base de données MySQL sont stockées dans des tables. Une table est une collection de données liées, et elle se compose de colonnes et de lignes. Les bases de données sont utiles pour stocker des informations de manière catégorique. Une entreprise peut avoir une base de données avec les tables suivantes :

- Des employés
- Des produits
- Clients
- Ordres

### 6.1. Requêtes de base de données

Une requête est une question et nous pouvons interroger une base de données pour obtenir des informations spécifiques et renvoyer un jeu d'enregistrements. Examinez la requête suivante (en SQL standard) :

```
« SELECT Nom FROM employes »
```

La requête ci-dessus sélectionne toutes les données de la colonne "LastName" de la table "employes".

## 6.2. Télécharger la base de données MySQL

Si vous n'avez pas de serveur PHP avec une base de données MySQL, vous pouvez le télécharger gratuitement ici : <http://www.mysql.com>

## 6.3. Etablir une connexion entre PHP et MySQL :

Le PHP 5 et les versions ultérieures peuvent fonctionner avec une base de données MySQL en utilisant :

- Extension MySQLi (le "i" signifie (improve) amélioré)
- PDO (objets de données PHP)

Les versions antérieures de PHP utilisaient l'extension MySQL. Cependant, cette extension a été dépréciée en 2012.

## 6.4. Ouvrir une connexion avec MySQL

---

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
// Create connection
$conn = new mysqli($servername, $username, $password);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
echo "Connected successfully";
?>
```

---

## 6.5. Fermer la connexion

La connexion sera fermée automatiquement à la fin du script. Pour fermer la connexion avant, utilisez ce qui suit : `mysqli_close($conn);`

## 6.6. Créer une base de données MySQL avec PHP

Une base de données se compose d'une ou plusieurs tables. Vous aurez besoin de privilèges CREATE spéciaux pour créer ou supprimer une base de données MySQL. L'instruction CREATE DATABASE est utilisée pour créer une base de données dans MySQL. Les exemples suivants créent une base de données nommée "myDB":

---

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
```

```

// Create connection
$conn = new mysqli($servername, $username, $password);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Create database
$sql = "CREATE DATABASE myDB";
if ($conn->query($sql) === TRUE) {
    echo "Database created successfully";
} else {
    echo "Error creating database: " . $conn->error;
}

$conn->close();
?>

```

---

## 6.7. Création d'une table mysql avec php :

L'instruction CREATE TABLE est utilisée pour créer une table dans MySQL. Nous allons créer une table nommée "MyGuests", avec cinq colonnes : "id", "firstname", "lastname", "email" et "reg\_date" :

```

CREATE TABLE MyGuests (
id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
firstname VARCHAR(30) NOT NULL,
lastname VARCHAR(30) NOT NULL,
email VARCHAR(50),
reg_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP
)

```

---

### 6.7.1. Remarques sur la table ci-dessus :

Le type de données spécifie le type de données que la colonne peut contenir. Après le type de données, vous pouvez spécifier d'autres attributs facultatifs pour chaque colonne :

- NOT NULL - Chaque ligne doit contenir une valeur pour cette colonne, les valeurs nulles ne sont pas autorisées
- Valeur DEFAULT - Définissez une valeur par défaut qui est ajoutée lorsqu'aucune autre valeur n'est transmise

- UNSIGNED - Utilisé pour les types de nombres, limite les données stockées aux nombres positifs et zéro
- AUTO INCREMENT - MySQL augmente automatiquement la valeur du champ de 1 à chaque fois qu'un nouvel enregistrement est ajouté
- CLÉ PRIMAIRE - Utilisée pour identifier de manière unique les lignes d'une table. La colonne avec le paramètre PRIMARY KEY est souvent un numéro d'identification et est souvent utilisée avec AUTO\_INCREMENT

Chaque table doit avoir une colonne de clé primaire (dans ce cas : la colonne "id"). Sa valeur doit être unique pour chaque enregistrement de la table.

---

```

<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// sql to create table
$sql = "CREATE TABLE MyGuests (
id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
firstname VARCHAR(30) NOT NULL,
lastname VARCHAR(30) NOT NULL,
email VARCHAR(50),
reg_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP
)";

if ($conn->query($sql) === TRUE) {
    echo "Table MyGuests created successfully";
} else {
    echo "Error creating table: " . $conn->error;
}

$conn->close();
?>

```

---

## 6.8. Insérer des données mysql avec php :

Une fois qu'une base de données et une table ont été créées, nous pouvons commencer à y ajouter des données. Voici quelques règles de syntaxe à suivre :

- La requête SQL doit être entre guillemets en PHP
- Les valeurs de chaîne à l'intérieur de la requête SQL doivent être entre guillemets
- Les valeurs numériques ne doivent pas être entre guillemets
- Le mot NULL ne doit pas être entre guillemets

L'instruction INSERT INTO est utilisée pour ajouter de nouveaux enregistrements à une table MySQL :

---

```
INSERT INTO table_name (column1, column2, column3,...)
```

```
VALUES (value1, value2, value3,...)
```

---

Dans le chapitre précédent, nous avons créé une table vide nommée "MyGuests" avec cinq colonnes : "id", "firstname", "lastname", "email" et "reg\_date". Maintenant, remplissons le tableau avec des données.

---

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$sql = "INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('John', 'Doe', 'john@example.com)";

if ($conn->query($sql) === TRUE) {
    echo "New record created successfully";
} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}

$conn->close();
?>
```

---

### 6.8.1. Obtenir l'ID du dernier enregistrement inséré

Si nous effectuons un INSERT ou UPDATE sur une table avec un champ AUTO\_INCREMENT, nous pouvons obtenir immédiatement l'ID du dernier enregistrement inséré/mis à jour. Dans la table "MyGuests", la colonne "id" est un champ AUTO\_INCREMENT :

---

```
CREATE TABLE MyGuests (  
id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
firstname VARCHAR(30) NOT NULL,  
lastname VARCHAR(30) NOT NULL,  
email VARCHAR(50),  
reg_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP ON UPDATE  
CURRENT_TIMESTAMP  
)
```

---

Nous renvoyons également le dernier ID inséré :

---

```
<?php  
$servername = "localhost";  
$username = "username";  
$password = "password";  
$dbname = "myDB";  
  
// Create connection  
$conn = new mysqli($servername, $username, $password, $dbname);  
// Check connection  
if ($conn->connect_error) {  
    die("Connection failed: " . $conn->connect_error);  
}  
  
$sql = "INSERT INTO MyGuests (firstname, lastname, email)  
VALUES ('John', 'Doe', 'john@example.com)";  
  
if ($conn->query($sql) === TRUE) {  
    $last_id = $conn->insert_id;  
    echo "New record created successfully. Last inserted ID is: " . $last_id;  
} else {  
    echo "Error: " . $sql . "<br>" . $conn->error;  
}  
  
$conn->close();  
?>
```

---

## 6.9. Insérer plusieurs enregistrement mysql à l'aide de mysqli :

Plusieurs instructions SQL doivent être exécutées avec la `mysqli_multi_query()` fonction. Les exemples suivants ajoutent trois nouveaux enregistrements à la table "MyGuests" :

---

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$sql = "INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('John', 'Doe', 'john@example.com')";
$sql .= "INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('Mary', 'Moe', 'mary@example.com')";
$sql .= "INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('Julie', 'Dooley', 'julie@example.com')";

if ($conn->multi_query($sql) === TRUE) {
    echo "New records created successfully";
} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}

$conn->close();
?>
```

---

## 6.10. Sélectionner des données à partir d'une base de données MySQL :

L'instruction `SELECT` est utilisée pour sélectionner des données dans une ou plusieurs tables :

**SELECT column\_name(s) FROM table\_name**

ou nous pouvons utiliser le caractère `*` pour sélectionner toutes les colonnes d'une table :

**SELECT \* FROM table\_name**

L'exemple suivant sélectionne les colonnes `id`, `firstname` et `lastname` de la table `MyGuests` et les affiche sur la page :

---

```

<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
$sql = "SELECT id, firstname, lastname FROM MyGuests";
$result = $conn->query($sql);
if ($result->num_rows > 0) {
    // output data of each row
    while($row = $result->fetch_assoc()) {
        echo "id: " . $row["id"]. " - Name: " . $row["firstname"]. " " . $row["lastname"]. "<br>";
    }
} else {
    echo "0 results";
}
$conn->close();
?>

```

---

## 6.11. Sélectionner et filtrer les données d'une base de données MySQL

La clause WHERE est utilisée pour filtrer les enregistrements. Elle est utilisée pour extraire uniquement les enregistrements qui remplissent une condition spécifiée.

**SELECT column\_name(s) FROM table\_name WHERE column\_name operator value**

L'exemple suivant sélectionne les colonnes id, firstname et lastname de la table MyGuests où le nom est "Doe", et l'affiche sur la page :

---

```

<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";
// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

```

```

$sql = "SELECT id, firstname, lastname FROM MyGuests WHERE lastname='Doe'";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    // output data of each row
    while($row = $result->fetch_assoc()) {
        echo "id: " . $row["id"]. " - Name: " . $row["firstname"]. " " . $row["lastname"]. "<br>";
    }
} else {
    echo "0 results";
}
$conn->close();
?>

```

---

## 6.12. Supprimer des données d'une table MySQL à l'aide de MySQLi

L'instruction DELETE est utilisée pour supprimer des enregistrements d'une table :

**DELETE FROM table\_name WHERE some\_column = some\_value**

---

```

<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
// sql to delete a record
$sql = "DELETE FROM MyGuests WHERE id=3";
if ($conn->query($sql) === TRUE) {
    echo "Record deleted successfully";
} else {
    echo "Error deleting record: " . $conn->error;
}
$conn->close();
?>

```

---

# Chapitre 7 :

# Travaux pratiques

---

## TP n°1: Serveur web et les outils de développement

---

### Exercice 1 :

1)- Un client tape l'adresse web <http://www.google.com> dans un navigateur

Détailler les étapes de connections entre le client et le serveur web de google.

2)- Un client entre dans un navigateur l'adresse [www.google.com](http://www.google.com) et l'adresse [mail.google.com](mailto:mail.google.com)

Quelle est la différence entre ces deux adresses et quelle est la différence entre le web et l'internet

3)- citez les étapes nécessaires pour rendre un site web en ligne.

### Exercice 2 : URL

l'uniforme resource locator (URL) est l'adresse universelle d'un site. Les url vous les utilisez en permanence. Un site internet, une page web, c'est d'abord une URL. Un nom de domaine bien choisis permet au site de gagner plus vite en popularité. Le terme URL est toujours utilisé comme synonyme de l'expression adresse web. La structure de base d'une URL

protocole://nom-du-serveur.nom-du-domaine/répertoire/nom-du-fichier

Ex : [http://fr.wikipedia.org/wiki/uniform\\_resource\\_locator.html](http://fr.wikipedia.org/wiki/uniform_resource_locator.html)

**http://** protocole technique de communication entre client et serveur web. **wikipedia** : nom du domaine. **.org** : extension du site web. **Wiki** : nom du sous domaine. **uniform\_resource\_locator** : nom du fichier. **.html** : extension du fichier.

### Répondu vrais ou faux :

Q1)- Les URLs peuvent comporter des espaces

Q2)- Les URL peuvent comporter des points, des tiret (-) des esperluettes (~) ou des underscores ()

Q3)- Le slash (/) permet d'organiser les dossiers et sous dossiers

Q4)- Il faut se méfier des URL qui ne commencent pas par HTTPS://

**Q5)-** le protocole ftp permet de : 1- créer un groupe d'échange ; 2- d'afficher une page au format HTML ; 3- de transférer des fichiers sur un serveur web

**Q6)** une URL pourrait être remplacée par l'adresse IP du serveur

**1- Les outils de développement**

- Télécharger et installer Visual studio code : <https://code.visualstudio.com/>
- Télécharger et installer XAMP : <https://www.apachefriends.org/fr/index.html>
- Télécharger et installer google chrome

**Travail à faire :** un site hébergé par un hébergeur gratuit possède généralement une adresse du type : nomduite.nomdehebergeur.extension. ex : daw.wordpress.com ou daw.wix.com. Créez un site web gratuit utilisant wordpress ou wix

- Créez une adresse électronique
- Accédez au site
  - <https://fr.wordpress.com/>
  - <https://fr.wix.com/>
  - un blog avec blogger : <https://www.blogger.com/>
- Choisissez un nom de domaine disponible et le type du site.
- Choisissez un thème template à votre site
- Ajouter un article avec des images
- Gérer vos commentaires

## TP n°2 : HTML

---

Un fichier HTML est un fichier texte simple, parfaitement lisible, respectant un ensemble de convention d'écriture. Un fichier HTML peut être écrit à l'aide de n'importe quel éditeur de texte simple (des éditeurs plus évolués existent également). Lorsque vous consultez un site web, les fichiers HTML (extension .htm ou .html) sont envoyés à votre navigateur et le code html est interprété sur VOTRE machine par VOTRE navigateur. Le code html est interprété par le navigateur pour visualiser le résultat. Ce code est constitué de texte et d'indications de mise en page. Ces indications sont appelées éléments ou balises.

**Objectifs du TP :** se familiariser avec le langage html, et création de page web en HTML

### I. Se familiariser avec le langage HTML

**Q1-** Ouvrez le visual code studio (ou Notepad++), recopiez-y le document suivant :

```
<!doctype html>
<html>
<head>
<title>Ma premiere page web !!</title>
</head>
<body>
Bonjour, <br>
c'est le premier TP de la matière <b> DAW</b>
</body>
</html>
```

**Q2-** Enregistrez le fichier sous le nom « test.html ».

A présent, pour visualiser le résultat, allez dans votre navigateur web, et choisissez dans le menu :

fichier->ouvrir et sélectionnez votre fichier test.html. Observez et commentez le résultat.

**Q3.** Dans la section body de votre fichier test.html, essayez et testez les effets des balises suivantes :

---

```
<b>Bonjour</b>
<i> Bonjour </i>
<u> Bonjour </u>
<mark> bonjour </mark>
<p> bon<sub>jou</sub>r</p>
<p> bon<sup>jou</sup>r</p>
<strong>bonjour </strong>
<em>bonjour </em>
<br>
<p> Bonjour
<h1>Bonjour</h1>
```

---

Changez ensuite le numéro de 1 à 6 de la dernière balise.

## Q4. Liens

### *Liens externes :*

`<a HREF="url" TARGET="type de fenêtre">Ceci est un lien !</a>`

url : Universal Resource Locator, décrit l'adresse d'une ressource web

type de fenêtre :

`_blank` : une nouvelle fenêtre

`_self` : la fenêtre elle-même

### **Exemple 1 :**

`<a href="https://www.univ-saida.dz/ar/">site de l'université de Saida</a>`

### **Exemple 2 :**

`<a href="page2.html">lien vers la page 2 dans le même site web</a>`

### *Liens internes ou ancre*

Ces liens guident vers un autre endroit du même document et sont qualifiés de liens internes.

1- définir de position de référence : `<a id="nom"></a>`, exemple :

`<a id= "TOP">point d'ancrage</a>`

2- `<a href= "#TOP">` lien vers le haut de la page `</a>`

Recopier le texte de '1' dans le fichier test.html en 1<sup>ère</sup> ligne (après la balise `<body>`)

Recopier le texte de '2' dans le fichier test.html avant la fermeture de la balise `<body>` (avant `</body>`)

Lien de téléchargement : `<a href= icon4.png download= « pglogo »>telecharger image </a>`

Mailling : `<a href= "mailto:bouararaahmed1990@gmail.com" >contact me </a>`

## Q5. Image

La balise pour insérer une image est

IMG. Elle s'utilise selon le modèle suivant (à adapter a votre nom de fichier image) :

``

Ajouter ces balises dans la section body :

`<IMG SRC="saida.jpg">`

`<br>`

``

## Q6. Les listes

Ajouter ces balises dans la section body :

---

```
<dl>
<dt>graphe partiel</dt>
<dd>Un graphe partiel d'un graphe orienté ou non est le graphe obtenu en supprimant certains arcs ou arêtes.</dd>
<dt>Un sous-graphe</dt>
<dd>Un sous-graphe d'un graphe orienté ou non est le graphe obtenu en supprimant certains sommets et tous les arcs ou arêtes incidents aux sommets supprimés.</dd>
</dl>
```

---

Ensuite :

liste des courses:<br>

---

```
<ul>
<li> farine </li>
<li> sucre </li>
<li> oeufs </li>
<li> confiture
<li> beurre
</ul>
```

---

Ensuite : liste des courses:<br>

---

```
<ol>
<li> farine </li>
<li> sucre </li>
<li> oeufs </li>
<li> confiture
<li> beurre
</ol>
```

---

L'attribut type peut prendre plusieurs valeurs :

Pour les listes numérotées <ol>

- "i" pour numéroter en chiffres romains minuscules i,ii,iii...
- "I" pour les chiffres romains majuscules I,II,III...
- "a" pour numéroter avec a,b,c...
- "A" pour numéroter avec A,B,C...
- Pour les listes énumératives <ul>
- "circle" pour utiliser une puce ronde .
- "disc" pour utiliser une puce carrée .
- "square" pour utiliser une puce circulaire pleine.

**Q7.** en utilisant les images de icon 1 à icon 4 écrire un code html qui permet de réaliser l'affichage dans l'image suivante

- [index.html](#)
- Développement des application web
  - [visite la page facebook 2em année informatique 2019/2020](#)
- plan du cours
  -  [HTML](#)
  -  css
    - sélecteur
    - positionnement
    - arrière plan
-  php
  - mysql
- [Contact](#)

## Q8. Réaliser le code html qui permet d'afficher l'image suivante :

### I. HTML

1. Historique
2. Structure
3. Validation
4. Nouveautés HTML5

### II. CSS

- Introduction
- Les propriétés css
- Nouveautés CSS3

## Q9. Tableaux

Tableau : `<table> ... </table>`

- Ligne : `<tr> ... [</tr>]`

- Cellule : `<td> ... [</td>]`

- Cellule d'en-tête : `<th> ... [</th>]`

La balise `<thead>` est utilisée pour regrouper le contenu d'en-tête dans un tableau HTML.

L'élément `<thead>` est utilisé conjointement avec les éléments `<tbody>` et `<tfoot>` pour spécifier chaque partie d'un tableau (en-tête, corps, pied de page). La balise `<thead>` doit être utilisée dans le contexte suivant: En tant qu'enfant d'un élément `<table>`, après tout élément `<caption>` et `<colgroup>`, et avant tout `<tbody>`, `<tfoot>` et `<tr>` éléments.

### Exemple 1:

---

```
<table>
<tr> <td>1</td> <td>2</td> </tr>
<tr> <td>3</td> <td>4</td> <td>5</td> </tr>
</table>
```

---

### Exemple 2:

---

```
<table>
<tr> <td>1 <td>2
<tr> <td>3 <td>4 <td>5
</table>
```

---

### Exemple 3:

---

```
<table>
<tr> <td>1 <td colspan=2>2
<tr> <td>3 <td>4 <td>5
</table>
```

---

#### Exemple 4 :

```
<table>
<tr> <td>1 <td rowspan=2>2
<tr> <td>3 <td>4 <td>5
</table>
```

#### Exemple 5 :

```
<TABLE ALIGN="center" WIDTH="60%" BGCOLOR="silver" BORDER="1">
  <TR> <TH COLSPAN="2">Couleur de fond "silver"</TH> </TR>
  <TR><TH BGCOLOR="yellow">Fond "yellow"</TH>
  <TH BGCOLOR="lime">Fond "lime"</TH>
</TR>
</TABLE>
```

#### Exemple 6:

```
<TABLE ALIGN=center BORDER=1 CELLSPACING=0>
<TR onMouseOver="this.bgColor='yellow'"
  onMouseOut="this.bgColor='white'">
  <TD>un</TD>
  <TD>deux</TD>
</TR>
<TR>
  <TD onMouseOver="this.bgColor='lime'"
    onMouseOut="this.bgColor='white'">
    trois</TD>
  <TD onMouseOver="this.bgColor='aqua'"
    onMouseOut="this.bgColor='white'">
    quatre</TD></TR></TABLE>
```

Exécuter ces exemples et commenter l'affichage.

#### Q10. réaliser le code html qui permet d'afficher le tableau suivant

dev web 2em année

Matières	Note par matière	Coefficient	Note * Coefficient
réseau	17	4	68
ASD	15	3	45
logique	16	2	32
Totals		9	145
Moyenne			16.12

#### Q11. Ligne horizontale :

```
<H3 align=center>cours Web</H3>
<HR align=center size=8 width="50%">
```

#### Q12. L'arrière-plan:

```
<BODY BGCOLOR="#000088" TEXT="#FFFF00" LINK="#FF0000" VLINK="#8000FF"
ALINK="#FFFF00">
<DIV align=left> <H1>Bonsoir</H1></DIV>
<DIV align=right>
```

```
<H5>Bonsoir</H5>
X<SUB>2 </SUB>
t<SUP>3</SUP>
</DIV>
</BODY>
```

---

**Q13. Les frames :** Pour diviser l'écran en plusieurs fenêtres : Zone avec des fenêtres

---

```
<FRAMESET> Début de zone avec des fenêtres
</FRAMESET> Fin de zone avec des fenêtres
Agencement des fenêtres
<FRAMESET ROWS="..."> Fenêtres horizontales
<FRAMESET COLS="..."> Fenêtres verticales
```

---

**Exemple 1:** recopier les balises suivantes dans test3.html

---

```
<HTML>
<HEAD></HEAD>
<FRAMESET ROWS="30%,70%">
<FRAME>
<FRAME>
</FRAMESET>
</HTML>
```

---

Essayer le même code mais `<FRAMESET COLS="30%,70%">`

Mélanger les deux :

**Exemple 2:** recopier les balises suivantes dans test4.html

---

```
<FRAMESET ROWS="30%,70%">
<FRAME>
<FRAMESET COLS="30%,70%">
<FRAME>
<FRAME>
</FRAMESET>
</FRAMESET>
```

---

**Exemple 3:**

On construit 3 fichiers Html élémentaires que l'on place dans le même répertoire que le fichier de frames :

---

a.html contient : `<HTML><BODY><H4>Bonjour, c'est le fichier a.html</H4></BODY></HTML>`

b.html contient `<HTML><BODY><H1> Bonsoir, c'est le fichier b.html </H1></BODY></HTML>`

c.html contient `<HTML><BODY><H1> Salut, c'est le fichier c.html </H1></BODY></HTML>`

recopier les balises suivantes dans test5.html

```

<FRAMESET ROWS="30%,70%">
<FRAME SRC="A.htm">
<FRAMESET COLS="30%,70%">
<FRAME SRC="B.htm">
<FRAME SRC="C.htm">
</FRAMESET>
</FRAMESET>

```

---

Commenter l'exécution des fichiers test3.html...

#### Q14. Les balises sémantique :

Réaliser la page ci-dessous en utilisant les balises de structure html5 (header , nav , section, article footer...)

### Titre de l'entete de la page

- [Accueil](#)
- [Services](#)
- [Partenaires](#)
- [Contact](#)

### Titre de la section

#### Titre de l'article 1

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

#### Titre de l'article 2

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

© 2015 All Rights Reserved

#### Q15. Les formulaires:

##### Exemple 1: formulaire1.html

---

```

<html> <head> <title>les formulaires !!</title> </head>
<body>
<FORM action="f1.php" method="post">
  <P>
    Prénom : <INPUT type="text" name="prenom"><BR>
    Nom : <INPUT type="text" name="nom"><BR>
    E-mail: <INPUT type="text" name="email"><BR>
    N°TEL: <INPUT type="text" name="tel"><BR>
  Fonction :
  <SELECT name="fonction">
    <OPTION VALUE="enseignant">Enseignant</OPTION>

```

```

<OPTION VALUE="etudiant">Etudiant</OPTION>
<OPTION VALUE="ingenieur">Ingénieur</OPTION>
<OPTION VALUE="retraite">Retraité</OPTION>
<OPTION VALUE="autre">Autre</OPTION>
</SELECT><br>
  <INPUT type="radio" name="genre" value="homme"> Homme<BR>

  <INPUT type="radio" name="genre" value="femme"> Femme<BR>
<INPUT type="CHECKBOX" NAME="tache" VALUE="1" CHECKED> A: <br>
<INPUT type="CHECKBOX" NAME="tache" VALUE="2" CHECKED> B <br>
<INPUT type="CHECKBOX" NAME="tache" VALUE="4"> C <br>
Commentaires : <br>
<TEXTAREA rows="3" name="commentaires"> Tapez ici vos commentaires </TEXTAREA>
<br>
<INPUT type="submit" value="Envoyer"> <INPUT type="reset"> </P>

</FORM> </body> </html>

```

---

### Exemple 2: formulaire2.html

---

```

<html> <head> <title> les formulaires !!</title> </head>

<body>

<FORM method=post action="fl.php">
Enregistrement d'un utilisateur
<TABLE BORDER=0>
<TR> <TD>Nom</TD> <TD> <INPUT type=text name="nom"> </TD> </TR>
<TR> <TD>Prénom</TD> <TD> <INPUT type=text name="prenom"> </TD></TR>
<TR> <TD>Sexe</TD> <TD>
Homme : <INPUT type=radio name="sexe" value="M">
<br> Femme : <INPUT type=radio name="sexe" value="F">
</TD></TR>
<TR> <TD>Fonction</TD> <TD>
<SELECT name="fonction">
<OPTION VALUE="enseignant">Enseignant</OPTION>
<OPTION VALUE="etudiant">Etudiant</OPTION>
<OPTION VALUE="ingenieur">Ingénieur</OPTION>
<OPTION VALUE="retraite">Retraité</OPTION>
<OPTION VALUE="autre">Autre</OPTION>
</SELECT> </TD></TR>
<TR> <TD>Commentaires</TD> <TD> <TEXTAREA rows="3" name="commentaires">
Tapez ici vos commentaires</TEXTAREA> </TD></TR>
<TR> <TD COLSPAN=2> <INPUT type="submit" value="Envoyer"> </TD></TR>
</TABLE> </FORM> </body> </html>

```

---

## II. Travail demandé:

Comme réalisation du TP 1, créer des pages web qui comportent du texte, images, images cliquables, frames, liens externes et internes, tableaux, formulaires, élément audio, élément video, mail element, download image ....

**Indication :** vous pouvez prendre les cours des matières en L2 informatique comme contenu de ces pages web.

---

## TP n°3 : Cascading Style Sheets

---

Cascading Style Sheets (CSS) est un langage de feuille de style utilisé pour décrire la présentation d'un document écrit en HTML. CSS est l'un des langages principaux du Web et a été standardisé par le W3C. Ce standard évolue sous forme de niveaux (levels).

**Travail demandé :** Exécuter et commenter les exercices suivants

- **Etape 1: ajouter de la couleur & des fontes**

---

```
<html>

<head>

  <title>Ma première page avec du style</title>

  <!--css interne -->

  <style type="text/css">

    body {

      color: purple;

      background-color: #d8da3d }

    h1 {

      font-family: Helvetica, Geneva, Arial, SunSans-Regular, sans-serif }

    </style>

  </head> <body>

  <!-- Menu de navigation du site -->

  <ul class="navbar">

    <li><a href="index.html">Home page</a>

    <li><a href="reflexions.html">Réflexions</a>

    <li><a href="ville.html">Ma ville</a>

    <li><a href="liens.html">Liens</a>

  </ul>

  <!-- Contenu principal -->

  <h1>Ma première page avec du style</h1>

  <p>Bienvenue sur ma page avec du style!

  <p>Il lui manque des images, mais au moins, elle a du style. Et elle a des

    liens, même s'ils ne mènent nulle part.....;

  <p>Je devrais étayer, mais je ne sais comment encore.
```

---

```
<p>Je devrais étayer, mais je ne sais comment encore.  
<!-- Signer et dater la page, c'est une question de politesse! -->  
<address> Fait le 16 avril 2018<br>  
    par moi.</address>  
</BODY> </HTML>
```

---

## - Etape 2: la barre de navigation

---

```
body {  
    padding-left: 11px;  
    font-family: Georgia, "Times New Roman", Times, serif;  
    color: purple;  
    background-color: #d8da3d }  
ul.navbar {  
    position: absolute;  
    top: 2em;  
    left: 1em;  
    width: 9em }  
h1 {  
    font-family: Helvetica, Geneva, Arial, SunSans-Regular, sans-serif }
```

---

## - Etape 3: stylisez vos liens

---

```
_body {  
    padding-left: 11em;  
    font-family: Georgia, "Times New Roman", Times, serif;  
    color: purple;  
    background-color: #d8da3d }  
ul.navbar {  
    list-style-type: none;  
    padding: 0;  
    margin: 0;  
    position: absolute;
```

---

```
top: 2em;
left: 1em;
width: 9em }

h1 {
  font-family: Helvetica, Geneva, Arial, SunSans-Regular, sans-serif }
```

```
ul.navbar li {
  background: white;
  margin: 0.5em 0;
  padding: 0.3em;
  border-right: 1em solid black }
```

```
ul.navbar a {
  text-decoration: none }
```

```
a:link {
  color: blue }
```

```
a:visited {
  color: purple }
```

---

#### - Etape 4 : Line horizontale

L'ajout final à notre feuille de style est une ligne horizontale pour séparer le texte de la signature en bas.

---

```
body {
  padding-left: 11em;
  font-family: Georgia, "Times New Roman", Times, serif;
  color: purple;
  background-color: #d8da3d }

ul.navbar {
  list-style-type: none;
  padding: 0;
  margin: 0;
  position: absolute;
  top: 2em;
  left: 1em;
  width: 9em }

h1 {
  font-family: Helvetica, Geneva, Arial, SunSans-Regular, sans-serif }

ul.navbar li {
  background: white;
```

```

margin: 0.5em 0;
padding: 0.3em;
border-right: 1em solid black }
ul.navbar a {
text-decoration: none }
a:link {
color: blue }
a:visited {
color: purple }
address {
margin-top: 1em;
padding-top: 1em;
border-top: thin dotted }

```

---

### Etape 5: feuille de style externe

Nous disposons d'un fichier HTML avec une feuille de style intégrée. Mais si notre site se développe, nous voulons probablement que plusieurs pages partagent le même style. Il existe une meilleure méthode que de copier la feuille de style dans chaque page: si nous mettons la feuille de style dans un fichier à part, toutes les pages peuvent pointer sur celui-ci.

- **Fichier: monstyle.css**

---

```

body {
padding-left: 11em;
font-family: Georgia, "Times New Roman", Times, serif;
color: purple;
background-color: #d8da3d }
ul.navbar {
list-style-type: none;
padding: 0;
margin: 0;
position: absolute;
top: 2em;
left: 1em;
width: 9em }
h1 {
font-family: Helvetica, Geneva, Arial, SunSans-Regular, sans-serif }
ul.navbar li {
background: white;
margin: 0.5em 0;
padding: 0.3em;
border-right: 1em solid black }
ul.navbar a {
text-decoration: none }
a:link {

```

```
color: blue }
a:visited {
  color: purple }
address {
  margin-top: 1em;
  padding-top: 1em;
  border-top: thin dotted }
```

---

- **Le fichier HTML:**

---

```
<html>
<head>
  <title>Ma première page avec du style</title>
  <link rel="stylesheet" href="monstyle.css">
</head>
<body>
[etc.].....terminer le fichier HTML.....
```

---

- **Text format en CSS**

Ecrire un document css qui permet de changer

- La couleur du texte en rouge
- la taille du texte 16px
- la police du texte
- Souligner le texte
- Avec alignement du texte centrer
- Citer les types de spécification de couleurs possibles en CSS.
- border-style: dotted solid double dashed;
- ajouter une bordure ronde
- border: 5px solid red; // expliquer cette ligne de code
- rendre le texte en minuscule
- ajouter un espace de 5px au debut de chaque paragraphe du code html
- ajouter un espacement entre les caractères de 2px
- ajouter un espace de 1.8 entre les lignes
- rendre la police du texte en "Times New Roman"

- **Les Marges en CSS**

Ajouter une marge de :

- Gauch 20%, droite 5px, en haut 5px, en bas 2px
- margin: 25px 50px 75px; // commenter cette ligne
- ajouter des marge automatique
- margin-left: inherit; // commenter cette ligne

- **les liens en css :**

ajouter un lien à la balise html et commenter le résultat en appliquant le code css suivant :

---

```
a:link { color: red; }
a:visited { color: green; }
a:hover { color: hotpink; }
a:active { color: blue; }
a:link { text-decoration: none;}
a:visited { text-decoration: none;}
a:hover { text-decoration: underline;}
a:active { text-decoration: underline;}
a:link { background-color: yellow; }
a:visited { background-color: cyan; }
a:hover { background-color: lightgreen; }
a:active { background-color: hotpink;}
```

---

- **Les listes en css :**

ajouter une liste ordonnée et non ordonnée dans HTML et commenter le code CSS suivant :

---

```
ul{list-style-type: circle;} // remplacer circle par square
ol{list-style-type: upper-roman;} // remplacer upper-roman par lower-alpha
ul{list-style-image: url("icon4.png");}
ul{list-style-type: none;}
```

---

- **Les tableaux en css :**

Ajouter un tableau en html et commenter le code suivant

---

```
table { width: 100%; }
td { height: 50px; }
td { text-align: left;}
td { padding: 15px; text-align: left; }
tr:hover {background-color: #f5f5f5;}
```

---

**Exercice Background, Border, width et height :**

- Ajouter une image comme arrière-plan fixe avec une répétition vertical
- Ajouter la couleur red comme arrière-plan

Ajouter une bordure double et verte à gauche et à droite de taille 5px à l'élément p du document html

- padding: 25px 50px;
- border: 1px solid red; // commenter cette ligne

Ajouter une image et une division en html

- Changer la largeur de l'image et de la division
- Changer la longueur de l'image et de la division

**Les Couleurs en CSS :**

Ecrire un document html contenant des titres

- Mettre les titres en bleu
- Mettre le titre principal en italique ;
- Changer la couleur de fond de la page, tout en gardant un fond blanc pour les paragraphes.

- Ajouter une marge de 5px en haut et 2px en bas de page pour le contenu du document html

**Travail à faire :**

- Réaliser un menu déroulant utilisant HTML et CSS
- Réaliser un exposé sur le bootstrap 4 : <https://getbootstrap.com/>
- Diaporama en html et css

## TP n°5 PHP et MySql

---

Une page PHP porte l'extension « .php ». Une page PHP peut être entièrement programmée en PHP ou mélangée avec du code HTML. PHP est un langage « Embedded HTML », c'est à dire qu'il apparaît à n'importe quel endroit de la page HTML. Pour cela on le place dans des balises particulières : <?php et ?>.

**Outils :** après l'installation de « EasyPHP (ou wamp, ou lamp,...) », veuillez vérifier que les témoins des deux serveurs sont allumés (en vert).

### Exercice1:

1. Créez dans votre répertoire un fichier test1.php contenant le script suivant:

---

```
<html> <head> <title>Page de test</title> </head>
<body>
<h1>Page de test</h1>
<?php echo "Hello World! \n<br/>Bonjour à tous!"; ?>
</body>
</html>
```

---

2. ajouter la balise suivante et exécuter le fichier php:

---

```
<?php
phpinfo();
?>
```

---

3. ajouter le code suivant et exécuter le fichier php, commenter les résultats :

---

```
<br>
<?php
$ch1="bonjour?";
$ch2="bonsoir!!!";
$ch3=$ch2.$ch1."\n";
echo "ch3=".$ch3;
echo "<br>";
$ch2 .= $ch1;
echo "ch2=".$ch2;
echo "<br>";
if ($ch2==$ch3) echo "OK";
else echo "NOT OK!!";
?>
```

---

4. ajouter le code suivant et exécuter le fichier php:

---

```
<?php
$a=12;
if ($a == 5):
    echo "a = 5";
    echo "...";
endif;
```

---

---

```
elseif ($a == 6):
    echo "a = 6";
    echo "!!!";
else:
    echo "a ne vaut ni 5 ni 6";
endif;
?>
```

---

5. ajouter le code suivant et exécuter le fichier php:

---

```
<?php
function affiche($msg) {
    echo $msg;
}
function modif($msg) {
    $msg="coucou";
}
echo "<br>";
$msg="salut";
affiche($msg);
echo "<br>";
modif($msg);
echo $msg;
?>
```

---

Modifier la fonction modif : modif (&\$msg) et réexécuter à nouveau le fichier php :

7. ajouter le code suivant et exécuter le fichier php:

---

```
<br> définition d'une fonction fct<br>
<?php
function fct($n) {
    $val=1;
    for ($i=1;$i<=$n;$i++)
        $val *= $i;
    return $val;
}
?>
<br> appel de la fonction fct<br>
<?php
$val=fct(5);
echo $val;
?>
```

---

8. ajouter le code suivant et exécuter le fichier php:

---

```
<?php
$var=array(10,15,17,23,9);
echo $var[0]; // !!!!
echo "<br>";
```

---

---

```
echo $var[3]; // ???  
?>
```

---

### Exercice 2 :

Créer la page HTML f1.html dont le contenu est :

---

```
<form action="f1.php" method="post">  
Name: <input type="text" name="username"><br>  
Email: <input type="text" name="email"><br>  
<input type="submit" name="submit" value="Submit me!">  
</form>
```

---

Créer la page php f1.php suivante :

---

```
<?php  
echo $_POST['username'];  
echo "<br>";  
echo $_REQUEST['email'];  
?>
```

---

Visualiser f1.html, remplir les champs du formulaire et cliquer sur le bouton submit.  
Changer l'attribut method : remplacer post par get et réexécuter.

### Exercice 3.

1. Créez une page HTML de votre choix comprenant un formulaire avec les champs suivants:

- un champ texte(type="text")
- un champ mot de passe(type="password")
- une liste déroulante comprenant au moins 3 entrées(balise <select>...</select>)
- un groupe de boutons radio(type="radio")
- un bouton de réinitialisation du formulaire(type="reset")
- un bouton de soumission du formulaire(type="submit")

Vous utiliserez la méthode post pour l'envoi du formulaire, en positionnant la valeur De l'attribut method de la balise form à "post". L'attribut action du formulaire devra pointer vers un script, nommé affiche\_parametres.php, prenant en charge le traitement des données saisies par l'utilisateur dans les champs du formulaire.

2. Créez le script affiche\_parametres.php permettant d'afficher les valeurs saisies par l'utilisateur dans les différents champs du formulaire.

3. ajouter des scripts javascript au fichier HTML précédent pour vérifier que les champs du formulaire sont non vides.

#### Exercice 4 : requêtes SQL

##### - Création de la BDD db1

##### - Etape 01: création de la BDD db1:<br>

---

```
<?php
$con= mysqli_connect ("localhost","root","");//connexion mysql
if (!$con) {
    die("problem de connection: " . mysqli_connect_error());
}

// Create database
$sql = "CREATE DATABASE db1";
if (mysqli_query($con, $sql)) {
    echo "bdd crée avec succès ";
} else {
    echo "erreur lors de la creation de la BDD: " . mysqli_error($con);
}

mysqli_close($con);
?>
```

---

##### - Création la table etudiant de la BDD db1

##### - Etape 02: création de la table étudiant de la BDD db1:<br>

---

```
<?php
$servername = "localhost"; $username = "root"; $password = ""; $dbname = "db1";
$con = mysqli_connect($servername, $username, $password, $dbname);
if (!$con) { die("problem de connection : " . mysqli_connect_error()); }
// sql to create table
$sql="CREATE TABLE etudiant ( nom varchar(15),
prenom varchar(15),
age int )";

if (mysqli_query($con, $sql)) {
    echo "Table etudiant créer avec succès";
} else {
    echo "Erreur de création de table: " . mysqli_error($con); }

mysqli_close($con);
?>
```

---

**Remarque :** vous pouvez utiliser la rubrique Mysql dans easyphp pour créer la base et les tables.

##### - Insertion Mysql

---

```
<?php
```

```
$servername = "localhost"; $username = "root"; $password = ""; $dbname = "db1";  
$con = mysqli_connect($servername, $username, $password, $dbname);  
if (!$con) { die("Connection failed: " . mysqli_connect_error()); }
```

```
$sql="INSERT INTO etudiant(nom,prenom,age)VALUES('Oussama','Ahmed','35')";
```

```
if (mysqli_query($con, $sql)) {  
    echo "enregistrement creer avec succès";  
} else {  
    echo "Erreur " . mysqli_error($con);  
}
```

```
mysqli_close($con);
```

```
?>
```

---

- **Insertion avec un formulaire:**
- **Inclure ses balises dans un fichier html :**

---

```
<form action="insert.php" method="post">  
Nom: <input type="text" name="nomf"/>  
Prénom: <input type="text" name="prenomf"/>  
Age: <input type="text" name="agef"/>  
<input type="submit"/>  
</form>
```

---

- **Le fichier insert.php :**

---

```
<?php  
$servername = "localhost"; $username = "root"; $password = ""; $dbname = "db1";  
$con = mysqli_connect($servername, $username, $password, $dbname);  
if (!$con) { die("Connection failed: " . mysqli_connect_error()); }
```

```
$sql="INSERT INTO etudiant(nom,prenom,age)  
VALUES('$ _POST[nomf]','$ _POST[prenomf]','$ _POST[agef])  ";
```

```
if (mysqli_query($con, $sql)) {  
    echo "enregistrement creer avec succès";  
} else {  
    echo "Erreur " . mysqli_error($con);  
}
```

```
mysqli_close($con);
```

```
?>
```

---

- **Select MySQL**

---

```

<?php

$servername = "localhost";
$username = "root";
$password = "";
$dbname = "db1";
$conn = mysqli_connect($servername, $username, $password, $dbname);
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}
$sql = "SELECT nom, prenom, age FROM etudiant";
$result = mysqli_query($conn, $sql);
if (mysqli_num_rows($result) > 0) {
    while($row = mysqli_fetch_assoc($result)) {
        echo " - Name: " . $row["nom"]. " " . $row["prenom"]. " " . $row["age"]. "<br>";
    }
} else {
    echo "0 results";
}
mysqli_close($conn);

?>

```

---

- **Select MySQL avec where**

---

```

<?php

$servername = "localhost";
$username = "root";
$password = "";
$dbname = "db1";

// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}
$sql = "SELECT nom, prenom, age FROM student WHERE nom='Doe'";
$result = mysqli_query($conn, $sql);
if (mysqli_num_rows($result) > 0) {
    while($row = mysqli_fetch_assoc($result)) {
        echo "nom: " . $row["nom"]. " prenom: " . $row["prenom"]. " " . $row["age"]. "<br>";
    }
} else {
    echo "0 results";
}

mysqli_close($conn);

```

---

---

?>

---

## - Order by MySQL

---

<?php

```
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "db1";

// Create connection
$conn = mysqli_connect($servername, $username, $password, $dbname);
// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

$sql = "SELECT nom, prenom, age FROM student ORDER BY nom";
$result = mysqli_query($conn, $sql);
if (mysqli_num_rows($result) > 0) {
    while($row = mysqli_fetch_assoc($result)) {
        echo "nom: " . $row["nom"]. " prenom: " . $row["prenom"]. " " . $row["age"]. "<br>";
    }
} else {
    echo "0 results";
}
mysqli_close($conn);
```

?>

---

## - Update MySQL

---

<?php

```
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "db1";
$conn = mysqli_connect($servername, $username, $password, $dbname);
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

$sql = "UPDATE student SET nom='houari' WHERE prenom='ahmed'";

if (mysqli_query($conn, $sql)) {
    echo "enregistrement mis à jour avec succès";
} else {
```

---

---

```
    echo "Erreur de mise à jour: " . mysqli_error($conn);
}

mysqli_close($conn);

?>
```

---

### - Delete MySQL

---

```
<?php

$servername = "localhost";
$username = "root";
$password = "";
$dbname = "db1";

$conn = mysqli_connect($servername, $username, $password, $dbname);
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}
$sql = "DELETE FROM student WHERE prenom='ahmed'";

if (mysqli_query($conn, $sql)) {
    echo "enregistrement supprimer avec succès";
} else {
    echo "Erreur lors du suppression : " . mysqli_error($conn);
}

mysqli_close($conn);

?>
```

---

### **Mini projet :**

Conception et réalisation d'un site Web dynamique de réservations de chambres d'Hôtel. Votre site doit permettre :

1. L'ajout ou la modification de chambres d'Hôtel.
2. L'ajout, la modification ou la suppression de clients d'Hôtel.

En utilisant (html, css, java script, php, mysql)

## Références :

- [1] Lerdorf, R., Tatro, K., Kaehms, B., & McGredy, R. (2002). *Programming Php*. " O'Reilly Media, Inc."
- [2] Abdulloh, R. (2016). *Easy & Simple-Web Programming*. Elex Media Komputindo.
- [3] Levy, M. R. (1998). Web programming in Guide. *Software: Practice and Experience*, 28(15), 1581-1603.
- [4] Saad, M. I. (2020). *Otodidak Web Programming: Membuat Website Edutainment*. Elex Media Komputindo.
- [5] Abriata, L. A., Rodrigues, J. P., Salathé, M., & Patiny, L. (2018). Augmenting research, education, and outreach with client-side web programming. *Trends in biotechnology*, 36(5), 473-476.
- [6] Wicaksono, A., Florentinus, T. S., & Ahmadi, F. (2020). Development of E-learning in web programming subjects for moodle based vocational students. *Innovative Journal of Curriculum and Educational Technology*, 9(1), 1-9.
- [7] Tinoco, E. E. C., & Solís, I. S. (2014). *Programación web con CSS, JavaScript, PHP y Ajax*. Iván Soria Solís.
- [8] Schafer, S. M. (2007). *Web standards programmer's reference: HTML, CSS, JavaScript, Perl, Python, and PHP*. John Wiley & Sons.
- [9] Thakur, A., & Dhiman, K. (2021). Chat Room Using HTML, PHP, CSS, JS, AJAX. *arXiv preprint arXiv:2106.14704*.
- [10] Nazihovna, Y. G. (2022). CREATING A PLATFORM USING HTML, CSS AND JAVA SCRIPT METHODS AND STRENGTHENING EDUCATION WITH THIS STEAM. *Confrencea*, 5(5), 17-38.
- [11] Ranjan, A., Sinha, A., & Battewad, R. (2020). *JavaScript for Modern Web Development: Building a Web Application Using HTML, CSS, and JavaScript*. BPB Publications.
- [12] Ferguson, R. (2019). JavaScript and Development Tools. In *Beginning JavaScript* (pp. 11-24). Apress, Berkeley, CA.
- [13] Djirdeh, H., Murray, N., & Lerner, A. (2018). *Fullstack Vue: The Complete Guide to Vue.js and Friends*. Fullstack.
- [14] Nixon, R. (2021). *Learning PHP, MySQL & JavaScript*. " O'Reilly Media, Inc."
- [15] COE, S., & RC, Y. A Responsive E-Commerce Website Using PHP Framework.
- [16] Zandstra, M. (2021). *PHP 8 Objects, Patterns, and Practice: Mastering OO Enhancements, Design Patterns, and Essential Development Tools*. Apress.
- [17] Lerdorf, R., Tatro, K., & MacIntyre, P. (2006). *Programming Php*. " O'Reilly Media, Inc."
- [18] Beighley, L., & Morrison, M. (2008). *Head First PHP & MySQL: A Brain-Friendly Guide*. O'Reilly Media.

[19] Sklar, D. (2004). *Introduction à PHP 5*. O'Reilly Media, Inc..

[20] Stauffer, M. (2019). *Laravel: Up & running: A framework for building modern php apps*. O'Reilly Media.