République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Ecole Normale Supérieure de Constantine ASSIA DJABAR
Département Science Exacte et Informatique

Technologie Web 1

Par : Mme. Naouel Ouafek

Cours dédié aux étudiants de troisième année informatique

2021/2022


بسم الله الرحمن الرحيم
قالوا سبحانك لا علم لنا إلا ما علمتنا
إنك أنت العليم الحكيم
صدق الله العظيم

Mots clés : Technologie web, PHP, HTML,CSS.

Table des matières

Table des figures

Liste des tableaux

Introduction

Ce cours est dédié aux étudiants de la troisième année Informatique. C'est un cours sur deux semestres, et qui présente les concepts liés au thème du **Web** et propose les bases permettant d'approfondir **les connaissances** permettant la maîtrise de la construction des pages Web.

— Le code du cours est :**I346**.
— Coefficient : **3**.
— Volume horaire hebdomadaire : **1h30 cours et 1h30 TP**.

0.1 Objectifs de cours et pré-requis :

Les objectifs de ce cours Technologie Web1 sont :

1. Connaitre l'historique du web et savoir les éléments nécessaires pour créer des sites web statique.
2. Apprendre les de bases du **HTML5, et CSS** qui permettent **l'élaboration des pages Web**.
3. Apprendre le langage **PHP afin de pouvoir ensuite créer de manière autonome des sites internet dynamiques et interactifs.**
4.

Les chapitres HTML et CSS ne nécessite aucun pré-requis. Par contre le

chapitre PHP nécessite une connaissances de bases en programmation impérative.

0.2 Organisation de cours

Ce cours est structuré en deux grandes parties :

Partie I : Cette partie traite le web statique :
Le premier chapitre présente l'histoire de **l'internet**, le **web**, et les principes essentiels de la **technologie web**.
Le deuxième chapitre est une introduction sur le langage à balise et présentation des éléments de base **HTML** .
Enfin un troisième chapitre présente le langage CSS (des règles CSS les plus utilisée) et comment intégrer le code **CSS** avec le contenu **HTML**.

Partie II : Cette partie présente le web dynamique.
Le premier chapitre est introduction sur PHP qui est un langage de programmation qui ajoute de la dynamicité à un document web.
Le deuxième chapitre présente le langage PHP en tant que langage de programmation. il traite : les variables et les fonctions, les structures de contrôle et certaines particularités de PHP par rapport à d'autres langages à savoir l'étude des formulaires et les fichiers.

0.3 Planning

— Chapitre1 : Introduction à Intenet **(une séance)**
— Chapitre2 : les bases de HTML **(5 séances)**.
— Chapitre3 : Les bases de CSS : **(3 séances).**

— Chapitre4 : Introduction à PHP : **(une séance)**.

— Chapitre5 : Les base de PHP : **(7 séances)** .

0.4 Évaluation

— Une note de participation dans les séances de TP **5% de la note finale**.

— Compte-rendu à rendre chaque fin des deux chapitres : Chapitre CSS (*chapitre HTML inclue*) et le Chapitre PHP **5% de la note finale.**

— 2 Examens (un examen pour chaque semestre.)

Première partie : Web statique

Chapitre 1

Introduction à Internet

Sommaire

1.1 Historique et Définitions

1952 : Début de la recherche par ARPA (Advanced Recearch Projects Agency), une agence du ministère de la Défense américain, où ils ont crée avec succès à un réseau global d'ordinateurs pour des objectifs de sécurité de Etat nommé **ARBANET**.
1983 : L'utilisation de ARBANET par l'université Stanford pour la recherche scientifique. 1989 : l'aparition de World Wide Web par Tem Berners-lee qui est le service fourni sur Internet qui permet des internaute(l'utilisateur de l'internet) consulte des page web (des site web). 1995 : l'explosion de l'internet.

Definition 1.1.1. Internet : (INTERconnection of NETworks) en français : interconnexion des réseaux, c'est à dire un réseau mondial d'ordinateurs in-

terconnetés. C'est donc le réseau des réseaux.

Definition 1.1.2. Un Intranet est un réseau informatique utilisé à l'intérieur d'une entreprise ou de toute autre entité organisationnelle et qui utilise les mêmes protocoles qu'Internet (TCP, IP, HTTP, etc.).

Internet s'appuis sur des protocoles qui permettent la communication sous ce réseau, en particulier le protocole TCP-IP. de nombreuses applications peuvent être fêtent sur internet telque : le courrier électronique, transfert de fichiers (FTP), messagerie instantanée, World Wide Web ou **le Web**.

Definition 1.1.3. Le World Wide Web (WWW), littéralement la "toile mondiale", communément appelé le Web, et parfois la Toile (toile d'araignée), Le Web est une application internet présente des pages web (des documents web) consultable par le publique d'Internet.

Un "document hypertexte" (d'où le nom du protocole HTTP) permettant de naviguer entre les documents web (entre les pages web a travers des liens hypertexte).

1.2 Protocoles Internet

Bas niveau :

TCP/IP : Transmission Control Protocol/Internet Protocol. Protocole utilisé sur le réseau Internet pour transmettre des données entre deux machines. Protocole de transport, TCP prend à sa charge l'ouverture et le contrôle de la liaison entre deux ordinateurs. Procole d'adressage, IP assure le routage des paquets de données. A voir comme un langage universel permettant à deux machines de communiquer entre elles peu importe leur système d'exploitation.

Haut niveau (applicatif) :

1- HTTP : signifie HyperText Transfer Protocol ou Protocole de transfert

hypertexte. Ces documents hypertextes (pages ou fichiers) sont transférés entre un navigateur internet (le client) et un serveur web où ils sont stockés. Ils sont localisés grâce à une chaîne de caractères appelée URL(Uniform Resource Locator) et qui comprend nécessairement le nom du serveur (Adresse web), le chemin d'accès (Dossier et Fichier). L'autre, https pour HyperText Transfer Protocol Secure), est sécurisé et permet au visiteur de vérifier l'identité du site auquel il y accède grâce à un certificat d'authentification.

FIGURE 1.1 – URL

2- SMTP (mail) : Signifie « Simple Mail Transfer Protocol ». Il s'agit du protocole utilisé pour envoyer des e-mails sur Internet. Votre client de messagerie (comme Outlook, Eudora ou Mac OS X Mail) utilise SMTP pour envoyer un message au serveur de messagerie, et le serveur de messagerie utilise SMTP pour relayer ce message vers le serveur de messagerie destinataire approprié. Fondamentalement, SMTP est un ensemble de commandes qui authentifient et dirigent le transfert de courrier électronique. Lors de la configuration des paramètres de votre programme de messagerie, vous devez généralement définir le serveur SMTP sur les paramètres SMTP de votre fournisseur d'accès Internet local (c'est-à-dire "smtp.yourisp.com").

3-FTP : File Transfer Protocol, protocole de communication destiné à l'échange de fichiers sur un réseau TCP / IP. • Permet de « mettre en ligne » son site web • Une sorte d'explorateur sur un serveur à distance

1.3 Terminologie d'Internet

- Un protocole : désigne l'ensemble de règles et conventions régissant l'échange de données entre les ordinateurs.

- ISP : Inernet Service Provider : la société qui fourni l'Internet c-à-d elle permet à un internaute de se connecté au réseau mondial (Internet) pour un prix mensuelle.

- Site web : un ensemble de page web relier entre-elles par des liens hypertexte.

-Serveur Web Les développeur de site web stockent leur sites Web dans des machine spécialisée , des machine très puissante qui travaille sans cesser 24h/24h et 7/7jour pour fournir le site web à des visiteurs dans le monde entier.

- Un navigateur Web : c'est le logiciel qui traduit les pages web codées en HTML à un contenu comprhéensif par l'utilisateur (du texte des image etc.) le navigateur web est un logiciel ou une application qui permet de consulter le Web. Citons par exemple : Chrome Microsoft edge Firefox Opéra

-Adresse IP : est une suite de chiffres et de lettres attribuée à chaque appareil connecté à un réseau informatique, par exemple le réseau Internet. Cette étiquette identifie et distingue les milliards d'appareils connectés à Internet : ordinateurs, téléphones mobiles, imprimantes. Elle sert à établir un

FIGURE 1.2 – Client Serveur

lien entre le client et les sites web qu'il les consulte.

-Le DNS est utilisé par les navigateurs internet pour trouver l'adresse IP d'un ordinateur à partir de son nom.

1.4 Outils nécessaires

Les sites Web sont développé par un programmeur sous différent langages, dans ce cours nous allons traiter les suivant : HTML, CSS, ET PHP. Pour écrire ces langage nous allons besoins les outils suivante :

1- Notepad++ téléchargeable gratuitement sur https ://notepad-plus-plus.org/downloads/ . Ce logiciel sera utilisé comme éditeur de code HTMl,

FIGURE 1.3 – Navigateur

CSS, et PHP.

2- Pour exécuter le code nous allons utiliser un navigateur web telque

FIGURE 1.4 – Le logiciel Notepad ++

Google chrome.

Pour le web dynamique nous allons besoin de créer une structure client serveur comme le montre la figure (1.5)

1- Une fois le développeur termine la création du site web en code PHP, il sera stocker dans un Serveur Web (cette opération appelé hébergement) , le serveur est le responsable à diffuser le site pour le monde entier.

2-Le serveur génére du code HTML pour qu'il soit lu par le navigateur web au niveau de la machine client.

3-Le navigateur au niveau de la machine (l'ordinateur du visiteur)traduit ce

FIGURE 1.5 – Requête Client Serveur

code HTML en un contenu lisible pour le visiteur.

Et pour construire cette structure client serveur dans un même ordinateur nous allons besoin d'une plateforme nommé **WAMP SERVER** qui est gratuite, et elle regroupe :

• Apache : un serveur web. Il est chargé de délivrer les pages web aux visiteurs. Cependant, Apache ne gère que les sites web statiques (il ne peut traiter que des pages HTML). Il faut donc le compléter avec d'autres programmes.

• PHP : c'est un plug-in pour Apache qui le rend capable de traiter des pages web dynamiques en PHP.

• MySQL : c'est le logiciel de gestion de base de données. Il permet d'enregistrer des données de manière organisée (sous forme d'une BD).

Chapitre 2

Syntaxe HTML

Sommaire

2.1 Introduction

Ce chapitre présente la syntaxe du langage (HTML HyperText Markup Langage). Ce langage est définit par le W3C (World Wide Web consortium), organisme indépendant chargé de la normalisation et de la recherche sur la technologie Web .[1]
Ce langage HTML est basé sur la notion de **Balise (Tags)** pour définir les différents composants du document web.

1- Les Balise : sont des éléments de base du langage (ou mots clés) borné par les caractères < et >.
Touche < du clavier pour ouvrir et touche shift + < pour fermer c-à-d pour avoir le caractère >.
Le nom des balises est prédéfini dans les spécifications HTML, on ne peut donc **PAS** en inventer !
Il existe deux types de balises :

— **Les balises existant par paire** : ce sont les plus courantes. On écrit la première balise, on tape du texte, puis on ferme la balise en la réécrivant précédé d'un caractère slash "/". Par exemple :

<titre>Bienvenue !</titre>

La première paire de la balise qui est : <titre> (la balise ouvrante) indique le début du titre, et elle est refermée plus loin avec </titre> (la balise fermante). Le navigateur comprend que ce qui est entre <titre> et </titre> est le titre de votre site web, et que celui-ci est "Bienvenue !" .

— **Les balises seules :** elles sont un peu plus rares, mais il y en a quand même. On s'en sert en général pour insérer un élément dans une page. Ce type de balise se termine toujours par un slash "/", mais cette fois le slash se trouve à la fin de la balise. Par exemple la balise qui permet d'insérer une image :

<img />

Cette balise indique juste qu'il y a une image à cet endroit.

Toutes les commandes (les éléments HTML)qui seront présentées dans ce chapitre vont être incluse dans une balise principale nommée : *BODY* **<body> </body>**.

Tous les noms de balise doivent être écrites en minuscule.

2- Les attributs : ce sont des propriétés pour un élément (une balise) autrement-dit un moyen pour donner des précisions sur une balise. les attributs se déclarent dans la balise ouvrante et ses valeurs sont entourés par des guillemets **""** voir la figure (2.1). Par exemple :

, la balise <img /> indique que à cette endroit on insère une image, mais faut-il indiquer laquelle, c'est à laide de l'attribut **nom** avec sa **valeur= "oiseau .jpg"** on précise l'image voulue.

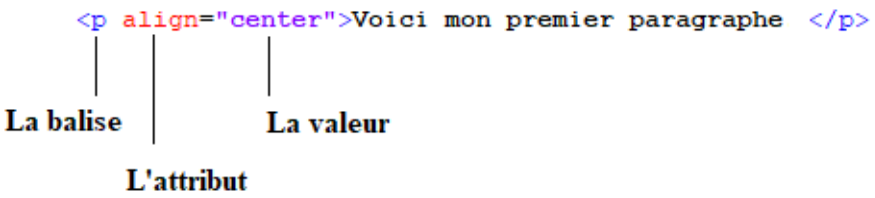


FIGURE 2.1 – L'attribut et sa valeur

3- Les commentaires sont des informations que le developeur du code HTML les écrive pour comprendre son code. Les commentaires n'apparaissent pas dans la page web.

Un commentaire est une balise un peu spéciale comme ceci :

< !– Ceci est commentaire –>

Remarque : pour vérifier qu'un code HTML est valide nous allons utiliser le validateur fourni par le consortium W3C http ://validator.w3.org/

2.2 Structure générale d'un document HTML

la structure d'un document html est présenté dans la figure (2.2).

```
<!doctype html>
<html lang="fr">
    <head>
    <title>  Titre apparaissant sur la barre de titre </title>
    <meta charset="UTF-8"/>
    </head>
    <body>
    <h1> Titre apparaissant dans la page web </h1>


    </body>
</html>
```

FIGURE 2.2 – la structure générale d'un document HTML

1. **La balise <!doctype html>** : Cette balise permet au navigateur de savoir quelle version de HTML est utilisée sur la page.
   Obligatoire pour valider une page (validateur W3C).

2. **La balise <html>** : Tout document HTML commence par la balise <html> qui se ferme en fin de document : </html>.
   l'attribut **lang** dans la balise <html lang="fr"> permet de spécifier la langue de traitement du document global.
   La balise <html> contient nécessairement deux balises définissant l'entête (<head>) et le corps (<body>) du document.

3. **La balise <head>** : elle détermine l'entête du document et contient les méta-informations telque :
   — La balise **<title>Titre de la page</title>** qui sera le titre du document et sera affiché dans la barre de titre de la fenêtre.
   — On ajoutera également les appels pour les feuilles de style CSS (l'objet du chapitre 2).
   — <meta charset="UTF-8"/> permet de définir l'encodage de caractère de la page par exemple dans le langage français il ya des

caractères spéciaux avec des accents.

4. **La balise <Body>** : elle délimite le contenu de la page.

2.3 Division de la page web

La balise qui permet de diviser la page en un ensemble de block :
Code html :
** Le premier block **
** Un deuxième block **
** Un troisième block **
L'objectif principale de cette balise est de regrouper d'autres balises HTML (paragraphes, tableaux, etc.) et de leur donner un style CSS commun (chose que nous allons traiter dans le chapitre 2).

2.4 Les éléments de mise en formes

2.4.1 les niveaux de titre

Il existe plusieurs niveaux de titre en HTML de niveau 1 jusqu'au niveau 6. Sachant que l'élément le plus haut niveau est le niveau 1. La balise HTML qui permet d'ajouter un titre est <h..> les (..) seront remplacés par un chiffre de 1 à 6, par exemple <h1> pour un titre de niveau 1.
La figure(2.4) représente le code HTML des différents niveaux de titre possible en HTML. La figure (2.5) représente le résultat obtenu.

2.4.2 Écrire des paragraphes

En html il faut préciser le commencement et la terminaison d'un paragraphe à l'aide de la balise <p>.

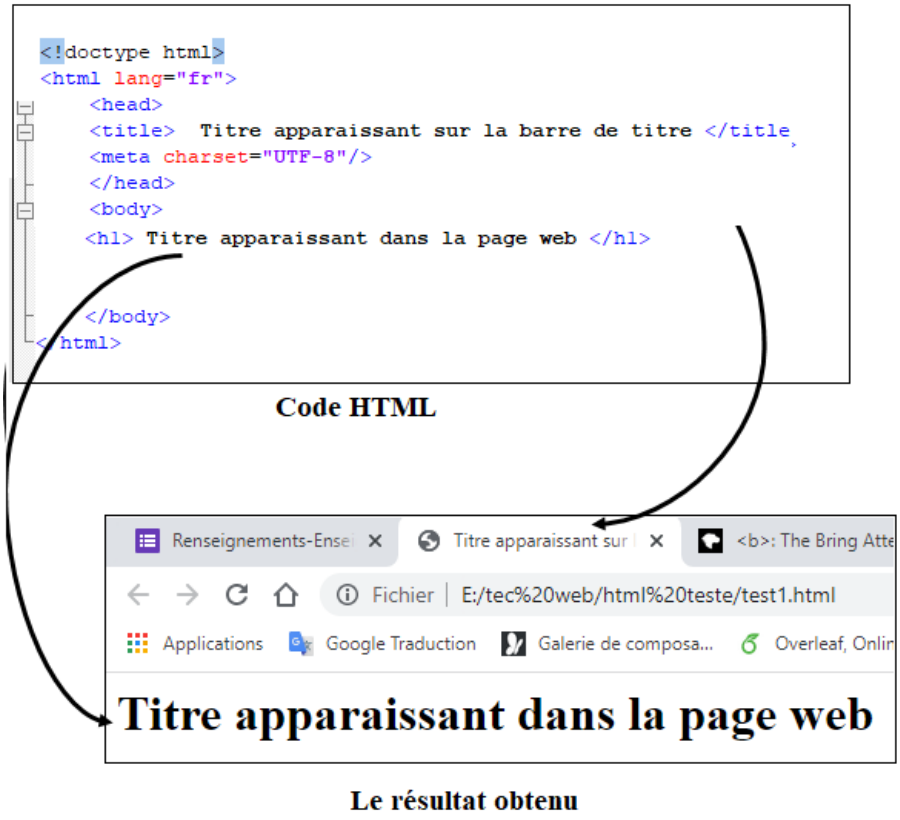


FIGURE 2.3 – Différence entre la balise de titre <h1> et la balise <tille>

Exemple :

 Voici mon premier paragraphe.

Pour mettre un paragraphe dans le centre de la page web , il faut utiliser l'attribut **align** avec la valeur **"center"** comme ceci (Figure (2.6)) :

2.4.3 Retour à la ligne

Pour faire un retour à la ligne il ne suffit pas de cliquer sur le bouton entrer du clavier (Figure (2.7)), il faut utiliser la balise
 "Break Line" qui force le passage à la ligne suivante, comme ceci (Figure (2.8), Le résultat est présenté dans la (Figure (2.9))) :

```
<html lang="fr">
    <head>
    <title>  Ma première page </title>
    <meta charset="UTF-8"/>
    </head>
    <body>
    <h1> Ceci est un titre en niveau 1 </h1>
    <h2> Ceci est un titre en niveau 2 </h2>
    <h3> Ceci est un titre en niveau 3 </h3>
    <h4> Ceci est un titre en niveau 4 </h4>
    <h5> Ceci est un titre en niveau 5 </h5>
    <h6> Ceci est un titre en niveau 6 </h6>
    </body>
</html>
```

FIGURE 2.4 – Le code des différents niveaux de titre

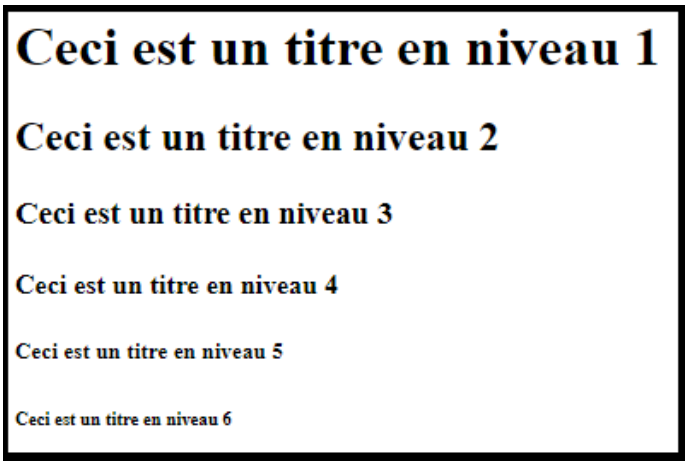


FIGURE 2.5 – Le résultat obtenu

Remarque : la forme
 est la dernière norme des structures de pages web

2.4.4 Mettre en valeur

— Pour mettre le texte en gras on l'encadre par la balise <b>.

— La balise <u> est utilisé pour souligner un texte.

— La balise <i> est utiliser pour mettre le texte en italique

La figure (2.10) montre le code et le résultat obtenu des balises de mise en

```
<p align="center"> Voici mon premier paragraphe.</p>
```

FIGURE 2.6 – Paragraphe Centré

```
<p align="center">
Parghraphe centré sur trois lignes
la 2 ligne obtene par le bouton ENTRER
la c'est la troisième ligne.
</p>
```

Code HTML

Parghraphe centré sur trois lignes la 2 ligne obtene par le bouton ENTRER la c'est la troisième ligne.

Résultat Obtenu

(Pas de saut de la ligne avec le bouton ENTRER)

FIGURE 2.7 – Teste du saut de la ligne avec le bouton ENTRER

valeur.

2.5 Les listes

Les listes sont des éléments très utilisées sur les pages Web.Il existe plusieurs types :

— Liste non ordonnée ou liste à puce.

— Liste ordonnée.

— Liste de définitions.

```
<p align="center">
Parghraphe centré sur trois lignes<br/>la 2 ligne obtene par la balise br <br/>
la c'est la troisième ligne.
</p>
```

FIGURE 2.8 – Code paragraphe avec la balise

Parghraphe centré sur trois lignes
la 2 ligne obtene par la balise br
la c'est la troisième ligne.

FIGURE 2.9 – Résultat avec la balise

2.5.1 Les listes non ordonnées :

Pour faire une liste non ordonnée en HTML en utilisant la balise <ul>, et la balise <li> pour chaque élément. Voir la figure (2.11)

2.5.2 Les listes ordonnées :

Pour faire une liste non ordonnée en HTML en utilisant la balise <ol>, et la balise <li> pour chaque élément. Voir la figure (2.11)

2.5.3 Les listes de définitions :

La balise <dl> est la balise html qui permet d'insérer une lise de définitions c'est à dire plusieurs éléments à définir dans une page web. Chaque élément de cette liste est insérer avec la balise <dt> et le texte de la définition est écrit par la balise <dd>. Dans le cas ou le mot à définir a plus q'une seule définition ,une balise <dd> est nécessaire pour chaque d'elles.

FIGURE 2.10 – Le code HTML et le résultat obtenu des balises de mise en valeur

La figure 2.13 illustre en détails le code HTML d'une liste de définitions

2.6 Image

l'insertion d'une image dans une page web se fait à l'aide de la balise seule <img/>, Elle admet différents attributs, dont les plus importants sont *src* et *alt*. la balise <img/> est présentée comme ceci :

****

L'attribut *src* permet de renseigner le chemin vers l'image et l'attribut *alt* affiche un texte alternatif à la place de l'image dans le cas où, pour une raison ou autre, celle-ci ne peut pas être chargée. Il aide aussi les moteurs de recherche à référencer l'image.

```
<ul> La liste
<li>premier élement</li>
<li>deuxième élement </li>

</ul>
```

Code HTML

La liste
- premier élement
- deuxième élement

Résultat obtenu

FIGURE 2.11 – Le code HTML pour les listes non ordonnée et le résultat obtenu

2.6.1 Stockage de l'image :

L'endroit où se trouve l'image à affichée par rapport au fichier HTML est très important pour l'attribut *src*.

— Dans le premier cas l'image se trouve dans le même dossier contenant le fichier HTML qui va l'afficher, la partie 1 de la figure (2.14) représente le code HTML correcte, et la partie 2 représente une erreur sur le chemin qui a généré l'affichage du texte de l'attribut *alt*.

— Le deuxième cas lorsque l'image se trouve dans un dossier et ce dernier et du même niveau que le fichier HTML (figure (2.15)). Le code HTML nécessaire est le suivant :

****

```
<ol> La liste ordonnée
    <li>premier élement</li>
    <li>deuxième élement </li>

</ol>
```

Code HTML

La liste ordonnée
1. premier élement
2. deuxième élement

Résultat

FIGURE 2.12 – Le code HTML pour les listes ordonnée et le résultat obtenu

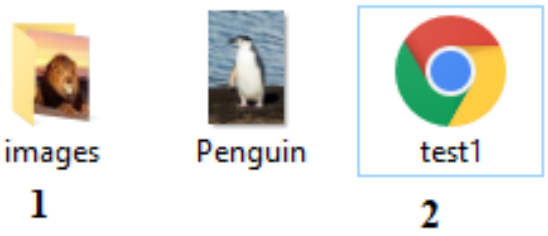


FIGURE 2.15 – Stockage de l'image dans un dossier : 1 : le dossier qui contient l'image de lion. 2 : le fichier HTML qui va l'afficher.

2.6.2 Infobule :

C'est un attribut facultatif qui permet d'afficher une aide flottante sur l'image comme un nuage dans lequel une description sur l'image est écrite. Cet attribut est l'attribut ***title*** figure (2.16).

```
<dl> Terminologie d'Internet
    <dt>- Un protocole:
        <dd>Définition 1: désigne l'ensemble de règles et conventions régissant
        l'échange de données entre les ordinateurs. </dd>
        <dd>Définition 2: c'est le moyen de communication
        entre les ordinateurs. </dd>
    </dt>
    <dt>- ISP:
        <dd>Définition 1: Inernet Service Provider : la société qui fourni l'Internet </dd>
        <dd>Définition 2: Un fournisseur d'Internet pour un prix mensuelle. </dd>
    </dt>

</dl>
```

Code HTML

Terminologie d'Internet
- Un protocole:
  Définition 1: désigne l'ensemble de règles et conventions régissant l'échange de données entre les ordinateurs.
  Définition 2: c'est le moyen de communication entre les ordinateurs.
- ISP:
  Définition 1: Inernet Service Provider : la société qui fourni l'Internet
  Définition 2: Un fournisseur d'Internet pour un prix mensuelle.

Résultat final

FIGURE 2.13 – Le code HTML pour une liste de définitions et le résultat obtenu

```
<img src="images\lion.jpg" alt="c'est un lion!" title="Un lion à la savane" />
```

Le code HTML

Le résultat est obtenu lorsqu'on raproche le curseur sur l'image

FIGURE 2.16 – Le code nécéssaire pour afficher un infobule.

2.7 Les liens

La balise qui permet d'ajouter un lien est la balise <**a**/></**a**>. Il faut cependant lui ajouter un attribut, **href**, pour indiquer vers quelle page nous

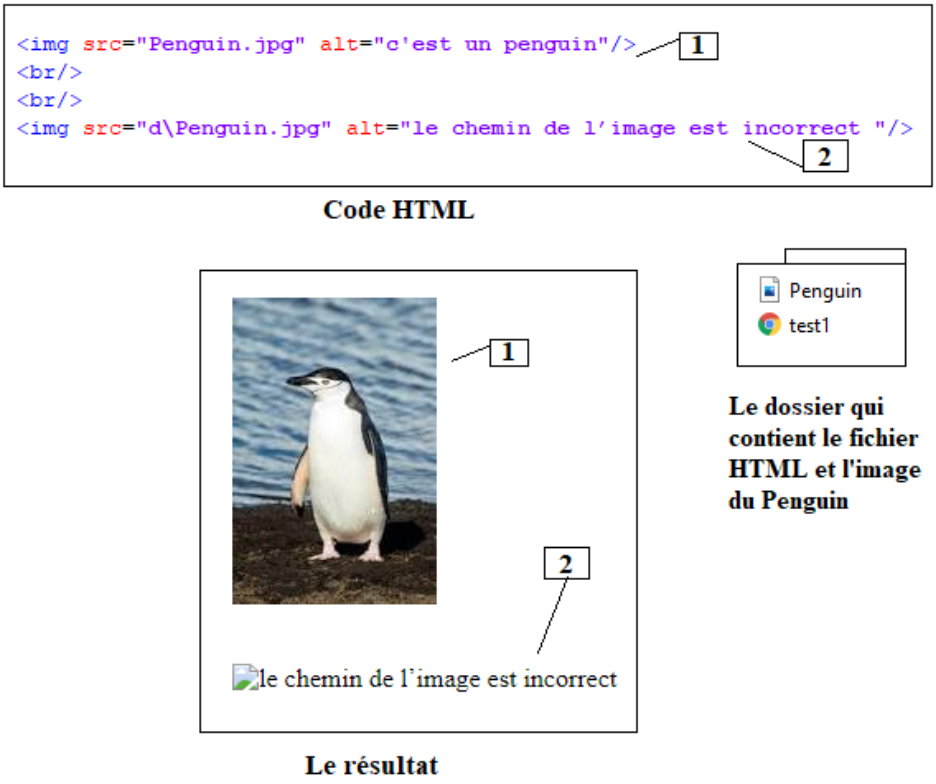


FIGURE 2.14 – Affichage de l'image, 1 : l'url vers l'image (le chemin) est correcte. 2 : le chemin est incorrecte

souhaitons visiter. Entre les deux paire <a/>et nous mettons le texte qui sera afficher dans la page courant, et en cliquant sur ce texte la page à visité apparaisse.

Il existe deux types de lien, un lien vers d'autre site web. Et des liens sur des pages qui existe dans le même site web.

2.7.1 Liens vers d'autre sites web

Pour ajouter dans la page courante un lien pour faire la recherche sur google, le code html nécessaire est le suivant :

Cliquer ici pour chercher

2.7.2 Liens vers une page dans le même site

Pour ce cas, nous avons plusieurs possibilités selon l'emplacement de stockage de la page a visité (la destination du lien) par rapport à la page actuelle (la page ou se trouve le lien). Supposant que la page actuelle est nommée page1 es la page à visitée est nommée *page2*.

1. Si la page a visité existe dans le même dossier contenant la page courante (voir figure(2.17)) le code nécessaire est :

    la page 2

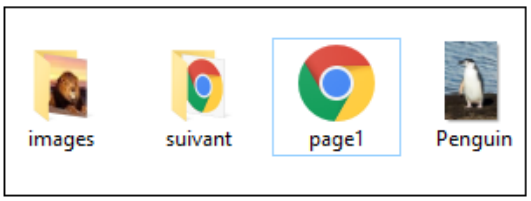
FIGURE 2.17 – 1 : le code HTML. 2 : le fichier page1.html et le fichier page2.html existent dans le même dossier. 3 : Le résultat obtenu le lien sera en bleu

2. Si la page2 existe dans un sous dossier (figure(2.18)) le code nécéssaire est :

    la page 2

```
<html lang="fr">
    <head>
    <title>  La page 1 </title>
    <meta charset="UTF-8"/>
    </head>
    <body>
    <h2> Voici un lien vers une page stocker dans le meme dossier</h2>
    <a href=suivant\page2.html> cliquer sur la page suivante</a>

    </body>
</html>
```

1



2

FIGURE 2.18 – 1 : le code HTML. 2 : le fichier page2.html existe dans un sous dossier nommé *suivant*

2.7.3 Liens vers une ancre

Une ancre joue le rôle d'un point de repère sur des liens vers une partie du document courant Pour ajouter une ancre il suffit d'ajouter un « # » dans la valeur de l'attribut href et ajouter un attribut id dans la balise de l'élément qui représente le point de l'encre. L'exemple suivant détaille la façon pour ajouter une ancre :

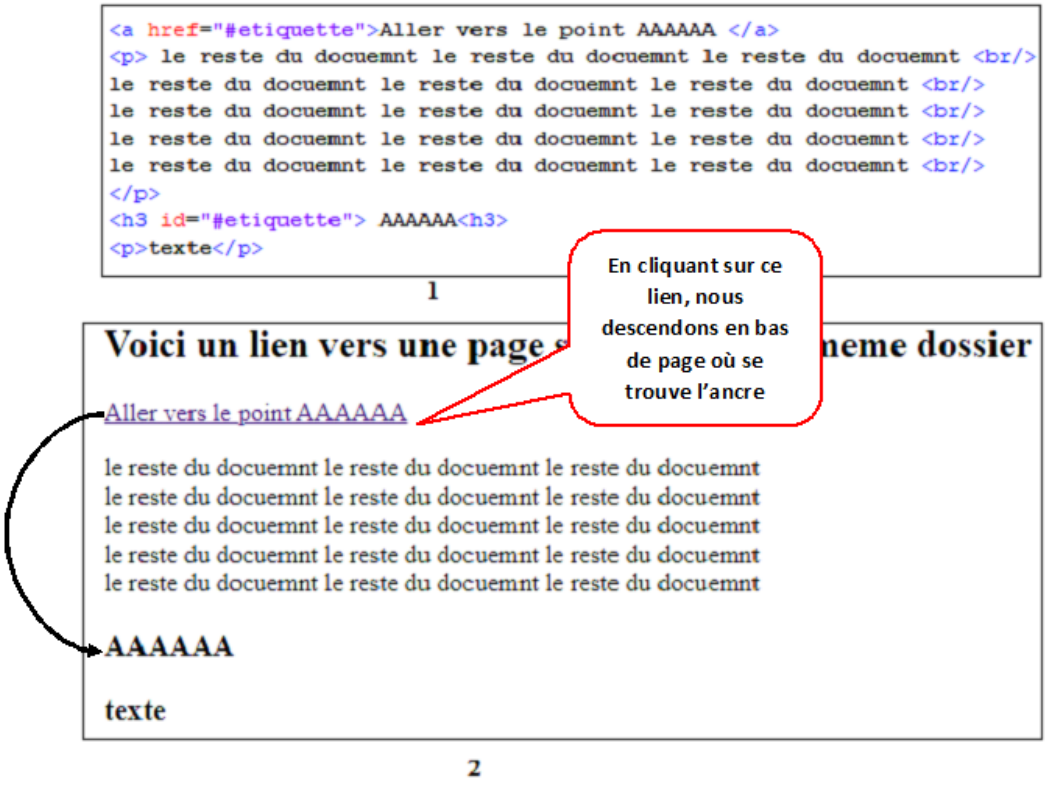


FIGURE 2.19 – 1 : le code HTML. 2 : le lien d'une ancre permet de se déplacer vers des points de repère dans la même page. *suivant*

Exercice : Ecrire le code html qui permet de revenir en haut de page après chaque paragraphe dans une page web (notez qu'il y a deux paragraphes).

Solution :

<body>

<h1 id="haut"> Ici le haut de page </h1>

<h3 >Paragraphe 1</h3>

<p> texte du paragraphe 1 texte du paragraphe 1 texte du paragraphe 1 texte du paragraphe 1 texte du paragraphe 1 texte du paragraphe 1 texte du paragraphe 1 </br>

** retour en haut**

<h3 >Paragraphe 2</h3>

 texte du paragraphe 2 texte du paragraphe 2 texte du paragraphe 2 texte du paragraphe 2 texte du paragraphe 2 texte du paragraphe 2 texte du paragraphe 2 texte du paragraphe 2 </br>

** retour en haut**

</body>

2.7.4 Liens vers une ancre dans une autre page

C'est un lien qui emmène à un emplacement (une ancre) situé plus bas ou plus haut sur autre page web.

Pour le faire il suffit d'indiquer le nom de la page suivi d'un dièse puis un nom de l'ancre.

Exemple :

```
<a href="page2.html#etiquette3"> aller à l'étiquette 3</a>
```

1. Code Html de la page1

```
<h3 id="#etiquette3"> un titre<h3>
```

2. Code Html de la page2

FIGURE 2.20 – Ancre dans une autre page

2.7.5 Ouvrir un lien dans une nouvelle fenêtre

Pour ouvrir un lien dans une nouvelle fenêtre il suffit d'ajouter l'attribut **target="_blank"** dans la balise <a>.

Exemple :

 Page2

2.8 Les Tableaux

Un tableau est un ensemble de lignes et de colonnes qui forment des cellules.

Les tableaux servent à organiser nos informations.

En HTML Le tableau se créer par la balise <table>, et pour insérer une ligne en ajoutant la balise <tr>, et la balise <td> pour chaque colonne.

un exemple de la structure générale d'un bateau est montré dans la figure (2.21).

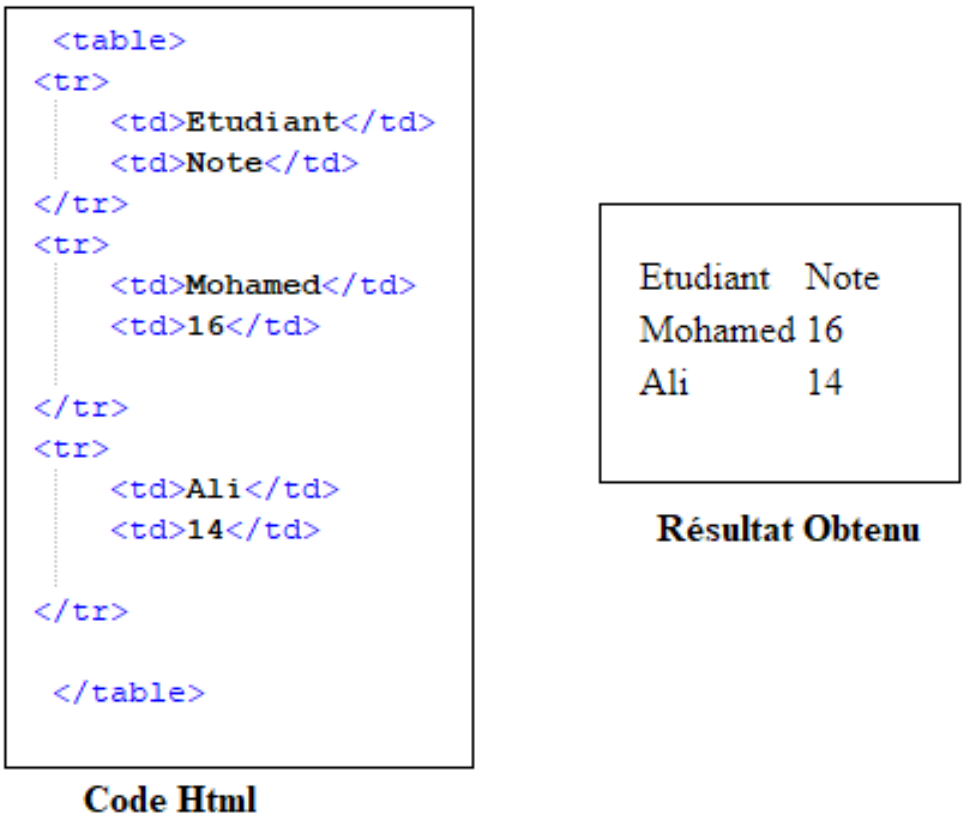


FIGURE 2.21 – Structure générale d'un tableau

Par défaut la bordure est invisible, ce type de tableau est très utile pour ***la mise en page d'un document***. Maintenant pour faire apparaître la bordure vous pouvez utiliser l'attribut border.

Code HTML :

<table border="1">

Etudiant	Note
Mohamed	16
Ali	14

FIGURE 2.22 – Résultat en ajoutant border="1" dans la balise <table>.

2.8.1 Titre du tableau

Pour donner un titre au tableau il suffit d'ajouter la balise <caption>, juste après l'ouverture du tableau (c-à-d après la balise <table>). Et pour déterminer l'endroit où sera placer ce titre, nous utilisrons l'attribut align qui accepte comme valeurs "top" (par défaut), "bottom" en bas du tableau, "left" à gauche et "right" à droite. Exemple pour le tableau précident nous pouvons utiliser le titre suivant :

<table border="1"> <caption align="top">Les notes de module Informatique</caption>

Le résultat comme le montre la figure(2.23).

Les notes de module Informatique

Etudiant	Note
Mohamed	16
Ali	14

FIGURE 2.23 – Titre d'un tableau

2.8.2 Entête du tableau

Souvent le contenu de la première ligne est considéré comme définition des valeurs du reste des lignes en dessous, elle joue le rôle d'entête du tableau. Pour la première ligne spécial la balise <td> par la balise <th> comme le montre la figure(2.24)

Les notes de module Informatique

Etudiant	Note
Mohamed	16
Ali	14

FIGURE 2.24 – La ligne d'entête en gras

2.8.3 Les différentes parties d'un tableau :

Les tableaux de grande taille sont plus difficiles à gérer. Pour cela on peut les diviser en plusieurs parties : partie entête, partie corps et partie pied du tableau respectivement représentées par les balises <thead>,<tbody> et <tfoot> figure (2.25).

```html
<table border="1">
        <caption align="top">les notes </caption>
        <thead>
        <tr>
            <th>étudiant</th>
            <th>note</th>
        </tr>
        </thead>
        <tbody>
        <tr>
            <td>Mohamed</td>
            <td>16</td>
        </tr>
        <tr>
            <td>Ali</td>
            <td>14</td>
        </tr>
        <tr>
            <td>sara</td>
            <td>17</td>

        </tr>
        <tbody>
        <tfoot>
        <tr>
            <th>étudiant</th>
            <th>note</th>
        </tr>
        </tfoot>

    </table>
```

Code

les notes

étudiant	note
Mohamed	16
Ali	14
sara	17
maha	12
étudiant	note

Résultat

FIGURE 2.25 – Les différents parties du tableau

Noter que ces balises sont facultatives mais elles aident à bien organiser le code du tableau.

Exercice : Ecrire le code HTML qui permet d'afficher la page suivante présentée dans la figure (2.26) :

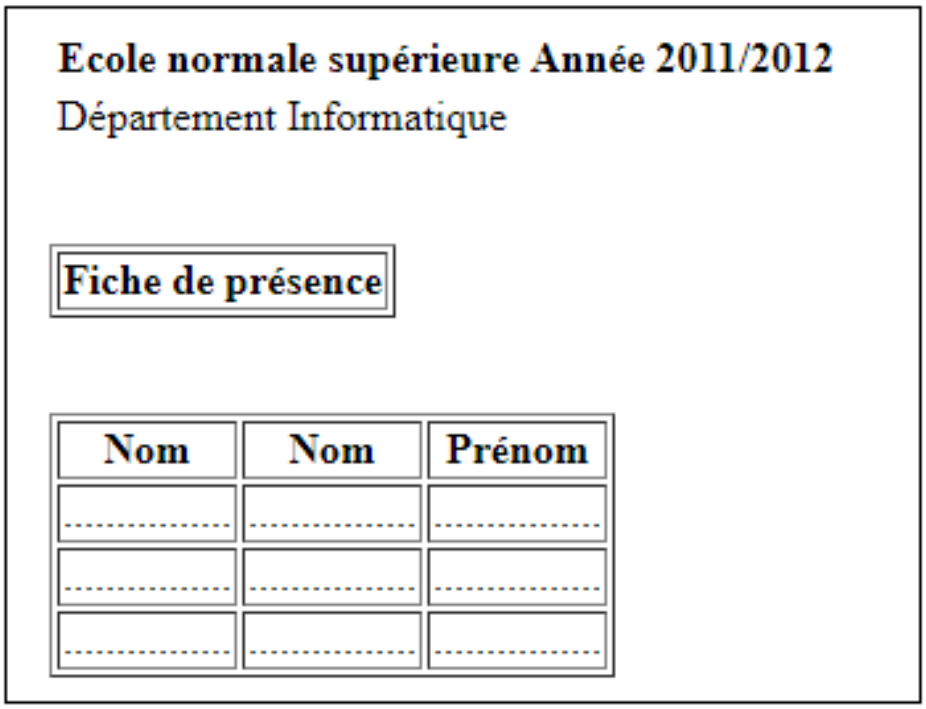


FIGURE 2.26 – Exercice tableau

2.8.4 Fusionner les cellules

Il existe deux types de fusion. Fusion de colonnes et fusion de lignes.

Pour faire la fusion de colonnes on utilise l'attribut **colspan** dans la balise <td>.

Pour fusionner les lignes on utilise l'attribut **rowspan** dans la balise <td>.

Après ou avant la fusion de cellules, il faut faire attention du nombre des lignes et des colonnes adjacentes (figures(2.27 et 2.28)).

```
<h3>Exemple de fusion de colonne:</h3>
<table border="1">
<tr>
<td colspan="3">une seule Cellule sur trois colonnes</td>
</tr>
<tr>
<td>Colonne 1</td>
<td>Colonne 2</td>
<td>Colonne 3</td>
</tr>
</table>
```

Code Html

Exemple de fusion de colonne:

une seule Cellule sur trois colonnes		
Colonne 1	Colonne 2	Colonne 3

Résultat

FIGURE 2.27 – Fusion de colonnes

```
<h3>Exemple de fusion de lignes :</h3>
<table border="1">
<tr>
<td>Ligne 1 - Colonne 2</td>
<td>Ligne 1 - Colonne 3</td>
<td>Ligne 1 - Colonne 4</td>
<td rowspan="2">Cellule sur deux lignes</td>

</tr>
<tr>
<td>Ligne 2 - Colonne 2</td>
<td>Ligne 2 - Colonne 3</td>
<td>Ligne 2 - Colonne 4</td>
</tr>
</table>
```

Code

Exemple de fusion de lignes :

Ligne 1 - Colonne 2	Ligne 1 - Colonne 3	Ligne 1 - Colonne 4	Cellule sur deux lignes
Ligne 2 - Colonne 2	Ligne 2 - Colonne 3	Ligne 2 - Colonne 4	

Résultat

FIGURE 2.28 – Fusion de lignes

2.9 Les Formulaires

Le formulaire permet à un visiteur de renseigner des données : remplir un champ de commentaire, s'inscrire dans un site, envoyer un émail etc. Donc ils permettent au visiteur d'interagir avec la page web qu'il est en train de consulter.
Les données remplies doivent être envoyé au serveur pour les analyser, cette opération ne peut pas être effectué par le langage html ça nécessite un autre langage plus puissant tel que le php. Cette étape de traitement fait l'objet du chapitre 3.

2.9.1 Création du formulaire

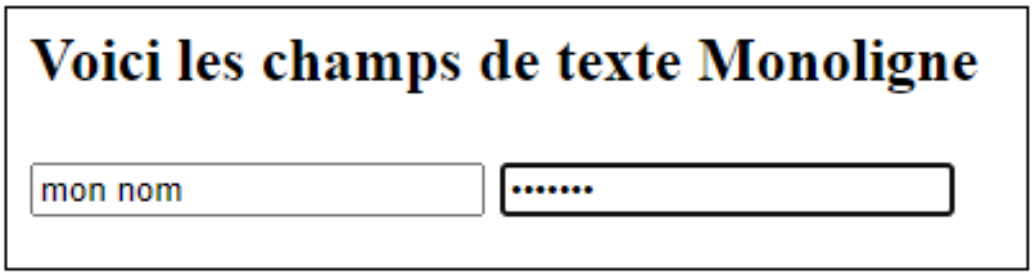
Pour créer un formulaire on utilise la balise <form>. C'est la balise principale, qui indique le début et la fin du formulaire. Cette balise accepte principalement deux attributs :
L'attribut action : pour indiquer l'adresse de la page ou le programme qui va traiter les données saisies par le visiteur.
Et l'attribut method : qui indique le mode d'envoi de ces données. Elle prend deux valeurs différentes qui sont **"get"** et **"post"**. La valeur **get** elle nous limite à 255 caractères et l'adresse du programme qui va traiter les données doit être une URL càd (http :/ /). Tandis que, la valeur **poste** elle nous permet d'envoyer un grand nombre de caractère et d'une manière transparente sans passer par la barre d'adresse.
Pour ces différentes raisons on utilise souvent la valeur poste pour l'attribut method.

2.9.2 Champ de texte monoligne

Comme son nom l'indique, on ne peut écrire qu'une seule ligne à l'intérieur. Elle sert à faire entrer un nom ou un mot de passe par exemple .
Pour insérer ce type de champ de texte, utiliser la balise <**input**/>. Cette balise a un attribut nommé **type** qui peut prendre deux valeur "**text**" ou "**password**".
Le **type text** pour faire entrer par exemple un pseudo ou un nom d'utilisateur, et le **type password** pour renseigner un mot de passe.
La figure (2.29)montre les deux type texte et password.

```
<h2> Voici les champ de texte Monoligne <h2/>
<form action="pagetraitement.php" method="post" >
        <input type="text" >
        <input type="password">
        </form>
```

Code Html



le résultat

FIGURE 2.29 – Le type **text** montre le texte entré par l'utilisateur, par contre le type **password** chiffre le texte en points

Il existe un attribut qu'on doit l'ajouter pour la balise <input> pour toute les balises de type formulaire. C'est L'attribut **name** qui seprésente le nom du champ, cet attribut n'apparaît pas sur la page, mais va permettre par la suite la page de traitement de retrouver d'où vienne la valeur de la balise, comme dans l'exemple précédent nous avons la même balise qui'est la balise **input** mais pour différencier les deux contenus comme dans le code suivant :

<form action="pagetraitement.php" method="post" >
<input type="text" name="login" />
<input type="password" name="mdp"/>
</form>

Comme nous avons dit l'attribut **name** est utile l'ors du traitement, il n'a aucun effet sur l'affichage de la page web. Maintenant pour clarifier la nature de contenu à placer pour chaque champ de texte, une indication est donnée à l'utilisateur via la balise <label> <label/ > Exemple :
<form action="pagetraitement.php" method="post" >
<label> entrer votre pseudo </label>
<input type="text" name="login" >
<label> entrer votre mot de passe</label>
<input type="password" name="mdp">
</form> Vous voyez dans l'exemple que rien ne relie la balise <label> et la balise au quelle elle rattaché.
Pour le faire on rajoute un attribut « **for** » à la balise label et un attribut « **id** » à l'autre balise avec la même valeur pour les deux attributs **for** et **id**.
Voir l'exemple :

<form action="pagetraitement.php" method="post" >

<label for="pseudo">entrer votre pseudo</label>

<input type="text" name="login" id="pseudo"/>

<label for="pwd">entrer votre mot de passe</label>

<input type="password" name="mdp" id="pwd"/>

</form>

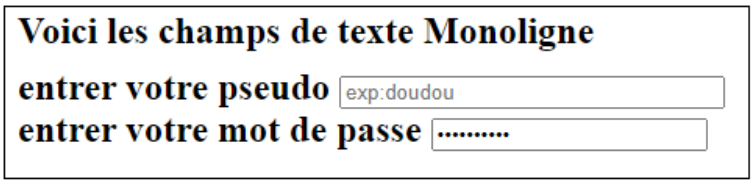
D'autres attributs sont disponibles, comme par exemple :

1. **Value** : qui permet de pré-remplir le champ par une valeur par défaut.
2. **Size** : qui permet de préciser la taille du champ en nombre de caractères.
3. **maxlength** : qui sert à limiter le nombre de caractères possibles.
4. **Placeholder** : qui sert à donner une indication sur le contenu du champ.

La figure (2.30)présente un exemple de champ monoligne avec les attributs nécessaires.

```
<h2> Voici les champs de texte Monoligne <h2/>
<form action="pagetraitement.php" method="post" >
        <label for="pseudo">entrer votre pseudo</label>
        <input type="text" name="login" placeholder="exp:doudou" size="30" id="pseudo">
        <br/>
        <label for="pwd">entrer votre mot de passe</label>
        <input type="password" name="mdp" value="motdepasse" id="pwd">
         </form>
```

Code Html



Résultat

FIGURE 2.30 – Exemple de champ monoligne avec les attribut nécessaire

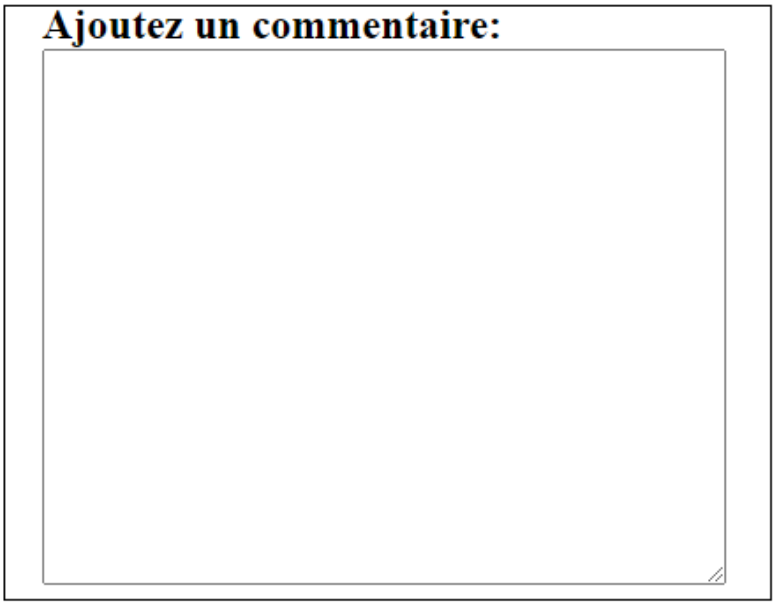
2.9.3 Champ de texte multiligne

Il permet à l'utilisateur d'écrire plusieurs lignes, comme par exemple : Rédigez un commentaire, ou un message sur plusieurs lignes. Pour permettre à un visiteur de saisir un texte sur plusieurs lignes utiliser la balise <textarea> <textarea/>

Nous pouvons changer la taille de la zone de texte en utilisant les deux attributs rows et cols (respectivement ligne et colonne).

```
<form action="pagetraitement.php" method="post" >
    <label for="com">Ajoutez un commentaire:<br/></label>
    <textarea name="commentaire" rows="20" cols="50" id="com" > </textarea>
</form>
```

Code Html



Résultat

FIGURE 2.31 – Exemple de champ multiligne

2.9.4 Case à coché

Les cases à cocher permet au visiteur de faire un choix multiple, Pour les faire, on utilise toujours la balise **<input>** dont l'attribut **type** prend la valeur « **checkbox** »

```
<form action="pagetraitement.php"method="post" >
<h3>
Cochez les moyens de transports que vous aimez les prendre :</h3><br/>
<input type="checkbox" name="train" id="train" />
<label for="train"> train<br/> </label>
<input type="checkbox" name="avion"id="avion"/>
<label for="avion"> avion<br /> </label>
<input type="checkbox" name="bus" id="bus"/>
<label for="bus">bus<br/></label>
<input type="checkbox"name="vélo" id="vélo"/>
<label for="vélo">vélo</br> </label>
<input type="checkbox"name="voiture" id="voiture" />
<label for="voiture">voiture </br> </label>

</form>
```

Code HTML

Cochez les moyens de transports que vous aimez les prendre

- ☐ **train**
- ☐ **avion**
- ☐ **bus**
- ☐ **vélo**
- ☐ **voiture**

Résultat

FIGURE 2.32 – Les cases à cochée

Maintenant pour cocher une case par défaut utiliser l'attribut **checked** voir la figure (2.33) **<input type="checkbox" name="train" id="train" checked/><label for="train"> train
</label>**

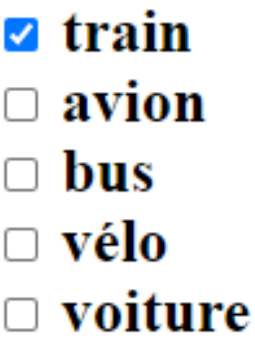


FIGURE 2.33 – L'attribut **checked**

2.9.5 Les boutons radio

Dans les boutons radio le choix est unique, pour les faire nous utilisons la même balise <input/> dont la valeur de l'attribut type est « **radio** » mais ce type fonctionne en groupe c-à-d chaque groupe a un attribut **nom** avec un nom identique comme dans l'exemple de la figure (2.34) nous avons deux groupes : groupe **"age"** et groupe **"sexe"**.
Maintenant pour récupérer la valeur sélectionnée par l'utilisateur un nouveau attribut est utilisé nommé **value**. La valeur de cette attribut va être transmise au serveur.

```
<form action="pagetraitement.php"method="post" >
<h3>Indiquer votre sexe :<br /></h3>
<input type="radio" name="sexe" value=" homme" id="hom"/>
<label for="hom"> homme<br /></label>
<input type="radio" name="sexe" value="femme" id="fem"/>
<label for="fem">femme<br /></label>


<h3>Quel est votre age  :<br /></h3>
<input type="radio" name="age" value="entre15et20" id="15-20"checked />
<label for="15-20"> entre 15 et 20 <br /></label>
<input type="radio" name="age"  value ="entre20et25" id="20-25"/>
<label for="20-25"> entre 20 et 25<br /></label>
<input type="radio" name="age"value=" entre25et30" id="25-30"/>
<label for="25-30"> entre 25 et 30<br /></label>
<input type="radio" name="age"  value ="plusde30" />
<label for="plusde30">plus de 30<br /></label>

</form>
```

Code HTML

Indiquer votre sexe :

- ○ homme
- ○ femme

Quel est votre age :

- ◉ entre 15 et 20
- ○ entre 20 et 25
- ○ entre 25 et 30
- ○ plus de 30

Résultat

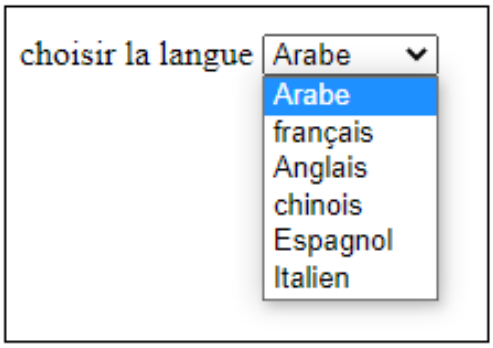
FIGURE 2.34 – Les boutons radio

2.9.6 Liste déroulante

Elle permet l'utilisateur de faire un choix parmi une liste de possibilité, pour l'insérer nous utilisons la balise <**select**></**select**>, et pour chaque choix (element de la liste) on utilise la balise <**option**>, avec un attribut **value** pour pouvoir identifier ce que le visiteur a choisi. Voir la figure (2.35).

```
<form action="pagetraitement.php"method="post" >
        <label for="langue"> choisir la langue</label>
        <select name="langue" id="langue">
            <option value="arabe">Arabe</option>
            <option value="français "> français </option>
            <option value="anglais" > Anglais </option>
            <option value="chinois"> chinois </option>
            <option value="Espagnol">Espagnol</option>
            <option value="italien"> Italien </option>
        </select>
</form>
```

Code HTML



Résultat

FIGURE 2.35 – Une option par défaut

Pour sélectionner une option par défaut , utilisez l'attribut **selected** :
<option value="anglais" selected> anglais </option>

```
<option value="anglais" selected> Anglais </option>
```

choisir la langue Anglais

FIGURE 2.36 – Liste déroulante

2.9.7 Les boutons

1. **Le bouton envoyer** : comme son nom l'indique, il sert à envoyer les données du formulaire au serveur. Pour insérer ce bouton à une page web nous utilisons toujours la balise <input> avec la valeur "submit" dans l'attribut type. Voir la figure . Exemple : **<input type="submit" name="envoyer" value="Envoyer">**

```
    <form action="pagetraitement.php"method="post" >
        <label for="langue"> choisir la langue</label>
        <select name="langue" id="langue">
            <option value="arabe">Arabe</option>
            <option value="français "> français </option>
            <option value="anglais" selected> Anglais </option>
            <option value="chinois"> chinois </option>
            <option value="Espagnol">Espagnol</option>
            <option value="italien"> Italien </option>
        </select>
        <input type="submit" name="envoyer"  value="Envoyer">
</form>
```

code Html

Résultat

FIGURE 2.37 – Le bouton **Envoyer**

2. il permet à réinitialiser toutes les valeurs du formulaire. C.-à-d. il mit tout le formulaire dans les valeurs initiales. Pour le faire donner la valeur « reset » pour l'attribut type de la balise <input> Exemple :

```
</select>
<br/>

<input type="reset" name="Recommencer " value="Effacer toutes les valeurs">

<input type="submit" name="envoyer"  value="Envoyer">

</form>
```

Code Html

choisir la langue Anglais

Effacer toutes les valeurs | Envoyer

Résultat

FIGURE 2.38 – Le bouton **Effacer**

2.10 Conclusion

Ce chapitre a présenté les balises essentielles dans le langage HTML qui servent principalement à organiser un document web, par contre sa mise en forme es étudier en détail dans le chapitre suivant.

Chapitre 3

CSS (Cascading Style Sheets) Feuille de style

Sommaire

3.1 Introduction

Le CSS c'est un langage qui complète le langage HTML, dans le quel on déclare toute la mise en forme des pages : la couleur de fond, les polices de caractère, leurs couleurs, etc. [2], [3]

3.2 L'endroit où on mit le code CSS :

Il existe plusieurs possibilités pour insérer le code .css (exp : fs.css).

— Créer un nouveau fichier dans **Notepad++** et lui porter l'extension .css et il doit être enregistré dans le même répertoire ou il y a le code html qui lui correspondre.

— Ajouter la ligne suivante dans le <head> du code html pour relier les deux fichiers : **<link rel="stylesheet" href="fs.css" />**

Cette méthode est la méthode la plus utilisée. D'autre méthode existent mais moins utilisées par le webmaster tel-que :

1. Dans le corps du code HTML càd dans la balise concerné
   Exemple : **<p style="color :orange ; font-size :150% ;">salut !<p>**
2. Dans l'en-tête de la page, voir la figure (3.2)

L'avantage d'utiliser une feuille de style dans un fichier séparé :

• Le style est définit une seul fois dans le fichier .Css et il est utilisé plusieurs fois par les différentes pages du site. figure(3.1).

• Cohérence de la présentation sur tout le site et les pages futures.

• Une mise à jour simple du style : si on veut modifier le style d'apparence de notre site web on le modifie une seul fois.

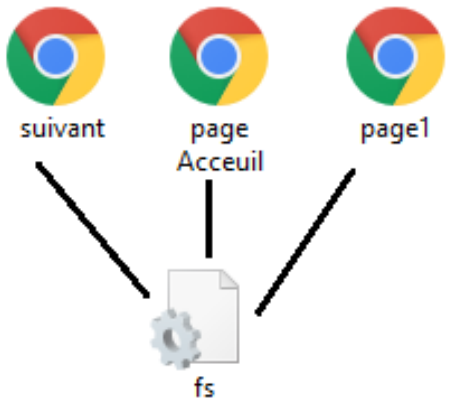


FIGURE 3.1 – La même feuille de style utilisée par plusieurs pages du site web

```html
<!doctype html>
<html lang="fr">
        <head>
        <meta charset="utf-8" />
        <style>
            p
            {
                color: blue;
            }
             h3
            {
                color: red;
            }
        </style>
        <title>Contenu CSS</title>
    </head>
    <body>
    <h3> Code html avec du style </h3>
    <p> Hello word</p>

    </body>
</html>
```

Code HTML

Code html avec du style

Hello word

Le résultat

FIGURE 3.2 – Le code CSS dans la balise <head> du document HTML

3.3 Format du code CSS :

1. Le code CSS contient 3 éléments différents :
   — **Les balises :** on cite les différentes balises html qu'on veut modi-

fier leurs apparences.

— **Les propriétés CSS** : se sont les attributs effet qu'on veut l'appliquer.

— **les valeurs** : à chaque propriété CSS on doit l'appliquer une valeur

Exemple d'une structure CSS :

Nom-balise-html1
{
propriete : valeur ;
}
Nom-balise-html2
{
Propriété1 : valeur ;
Propriété2 : valeur ;
}

Exemple :

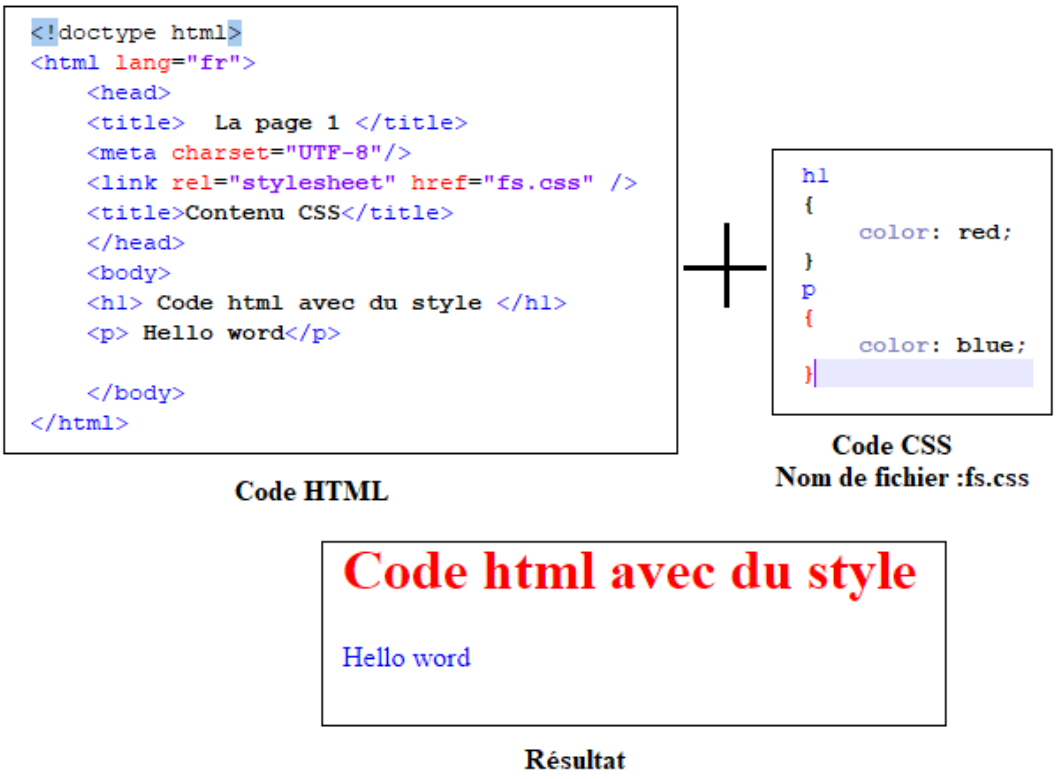


FIGURE 3.3 – Code HTML

2. Pour appliquer un même style sur plusieurs balises, on fait comme ceci

   Balisehtml1, Balisehtml2

   {

   propriété : valeur ;

   }

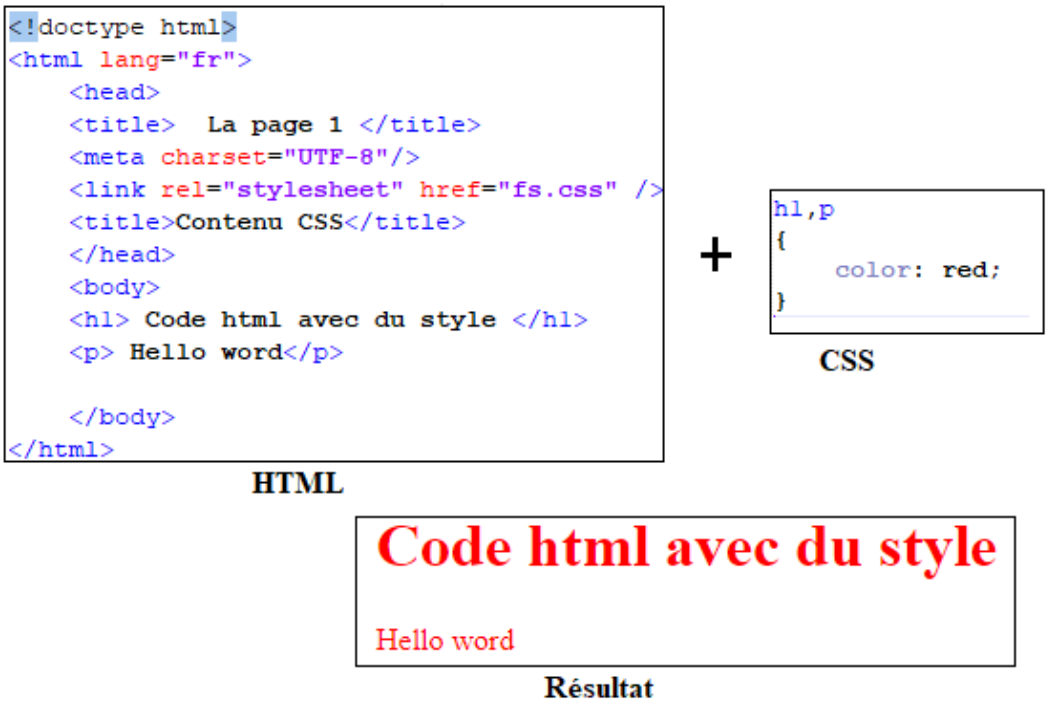


FIGURE 3.4 – Appliquer le même style sur plusieurs balises

3.4 Le sélecteur class :

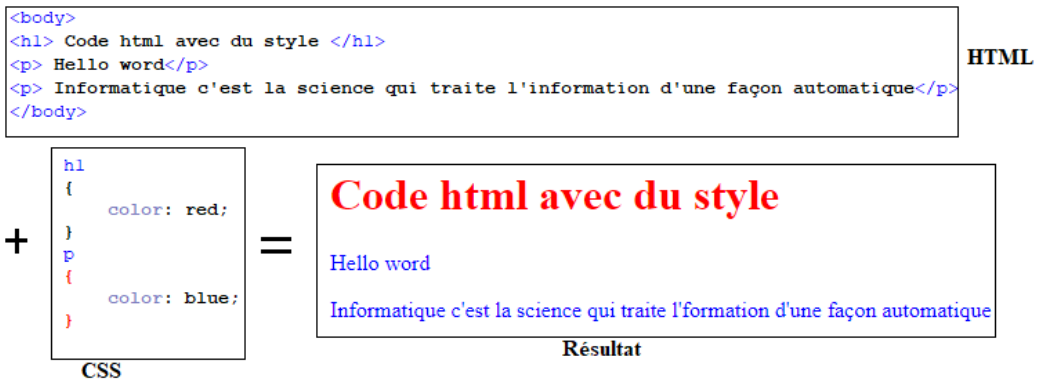


FIGURE 3.5 – comment séparer les deux paragraphe pour appliquer deux style différents

Comme le montre la figure (3.5) le code CSS indique que tous les paragraphes qui existent dans la page web seront en bleu Maintenant si vous voulez appliquer une mise en forme particulière à un paragraphe spécifique, il faut ajouter l'attribut '**class**' qui sépare ce paragraphe aux autres, pour que vous pouver changer son apparence.

Cette attribut on peut l'ajouter dans n'importe qu'elle balise du code html et sa valeur sera ajouter dans le code CSS précédé par un point à la place du nom de balise HTML, comme le montre la figure ()3.6.

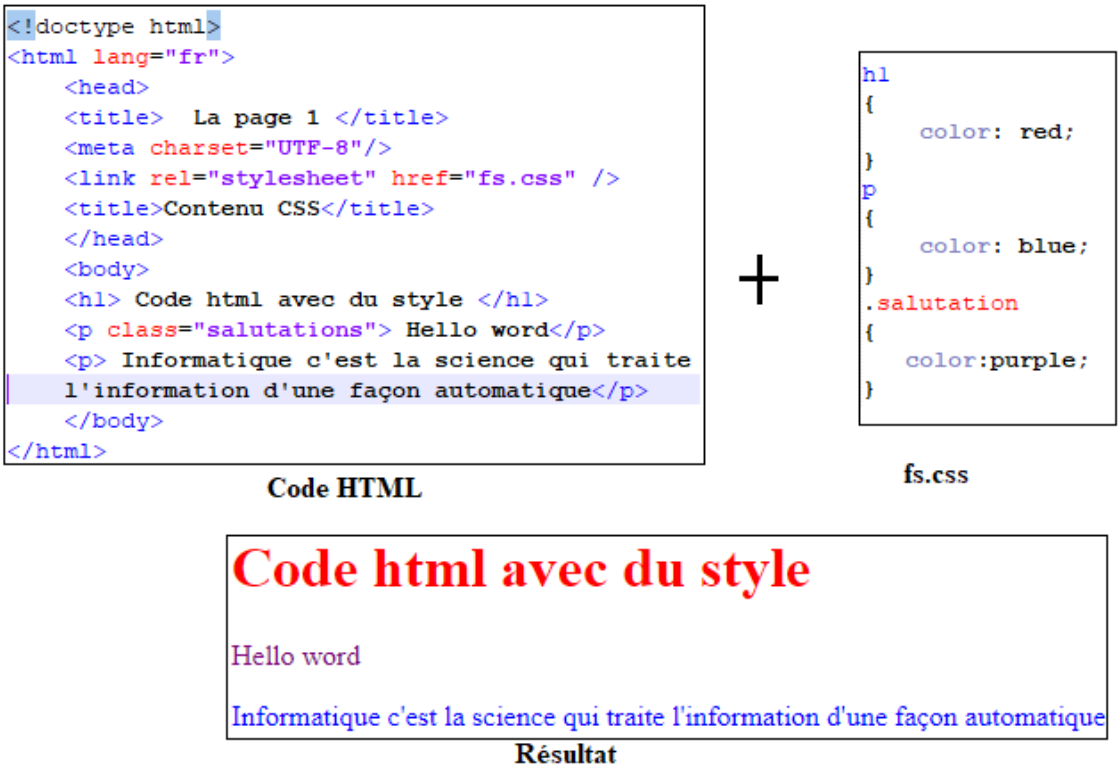


FIGURE 3.6 – l'attribut **class**

3.5 Les balises Div et Span de l'HTML

Pour modifier l'apparence des morceaux de texte au seine du même paragraphe, nous utilisons la balise <Span>

Exemple : Si vous voulez que le mot "informatique" soit avec une couleur différente du reste de la définition , encadrer ce mot avec la balise . Le résultat est montré dans la figure (3.7).

FIGURE 3.7 – l'attribut **span**

Cette balise est vu dans le chapitre 2 , elle permet de diviser la page en un ensemble de block :

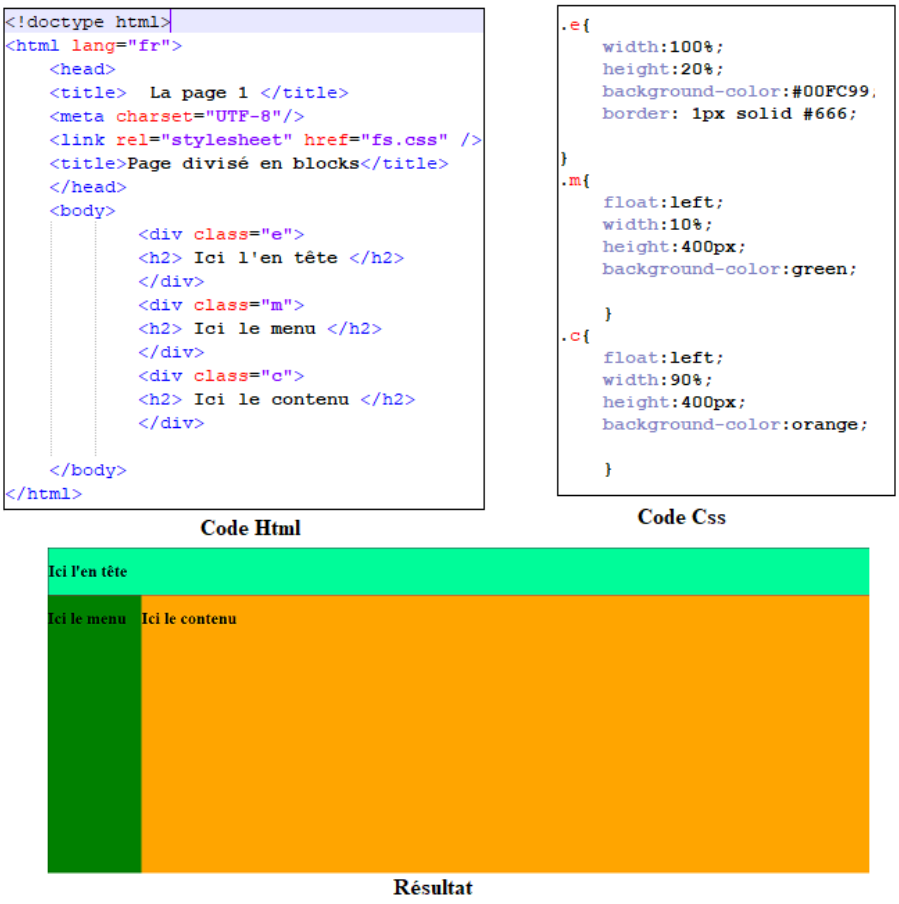


FIGURE 3.8 – la balise **div**

3.6 Regroupement des sélecteurs et l'héritage dans CSS

3.6.1 le regroupement des sélecteurs :

1. On peut regrouper les sélecteurs de même type ou de différents types au sein du code CSS, il faut juste les séparer par une virgule comme le montre la figure (3.9).

```css
h1, .salutations,.def
{
   color: red;
}
```

Code Css

```html
<body>
<h1> Code html avec du style </h1>
<p class="salutation"> Hello word</p>
<p> <span class="def">Informatique:</span>c'est la science qui traite
l'information d'une façon automatique</p>
```

Code HTML

FIGURE 3.9 – Regroupement des sélecteurs

2. Pour appliquer sur un seul élément (la même balise) plusieurs styles, on sépare les styles par un espace au niveau de la classe. Voir la figure(3.10).

```html
<body>
<h1 class="couleur centrer">Cours Application CSS </h1>
<h3 class="centrer engras">Chapitre:3</h3>
```

Code Html

```css
.couleur
{
color:blue;
}
.engras{
font-weight:bold;
}
.centrer{
text-align : center;
}
```

Code Css

Résultat

FIGURE 3.10 – Regroupement des sélecteurs au niveau de la classe

3.6.2 L'héritage dans CSS

Dans le cas d'emboîtement des balises, le navigateur respect une certain stratégie pour faire la mise en forme.

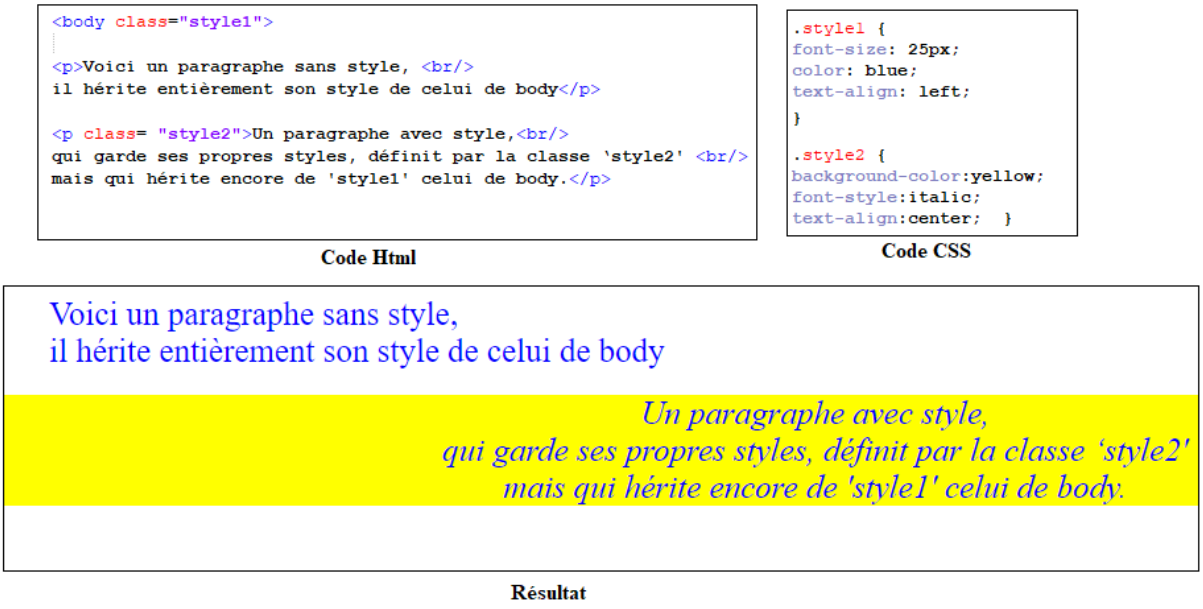


FIGURE 3.11 – L'héritage de style en CSS

Explication de l'exemple : Le premier paragraphe n'a pas de style, il hérite directement son style de la classe '*style1*' celui du body. Le deuxième paragraphe il a son stype définit par la classe '*style2*', il hérite du '*style1*' les propriétés suivante **font-size : 25px ; color : blue ;** qui ne sont pas définit dans son style.

Par contre la propriété : **text-align** est définit dans les deux styles, le navigateur exécute celle du style de la balise elle même, donc **text-align :** prend la valeur **center** qui est définit dans '*style2*'.

En plus il a d'autre propriétés propre à son style qui sont **font-style :italic** ; **background-color :yellow.**

En conclusion :

L'enfant hérite son **Style** des propriétés de ses parents. Mais S'il y a d'autres valeurs différentes pour la même propriété, l'enfant garde les siennes, sinon on remonte l'arborescence pour connaître le style associé.

3.7 Quelques propriétés CSS

3.7.1 Propriétés pour le texte

1. **La propriété CSS** qui permet d'indiquer la police à utiliser est **font-family**.

   Code CSS :

   balise{

   font-family : police ;

   }

   - Il existe une multitude de polices de caractères qui ont chacune leur forme. Et a vous de choisir la police selon votre préférence. En effet, Il se peut que la police que vous avez spécifiée n'existe pas sur le système du visiteur. Vous avez donc la possibilité de lui proposer d'autres choix alternatifs.

   Le navigateur essaiera d'abord de mettre la police1. Si le visiteur ne l'a pas, il essaiera de mettre la police2 et ainsi de suite. Il faut séparer par une virgule la liste des polices.

   Code CSS :

   Balise{

   font-family : police1, police2, police3, police4 ;

   }

   - Si le nom de la police contient un espace, il faut le mettre entre guillemets comme "**Arial Black**".

   - Enfin, on a intérêt à mettre en tout dernier une police standard. C'est-à-dire une police dont on est certain qu'elle existe partout, avec cette option on est sûr que l'affichage sera toujours correct, au moins pour cette police.

Code CSS :

p
{ font-family : "Agency FB", Cambria, Calibri, Andalus, serif }

2. **Taille de caractères :** La taille des caractères se déclare par l'attribut **font-size**, Il y a deux façons de spécifier sa valeur :
   — Indiquer **une taille fixe** : en pixels, en centimètres ou millimètres. Cette méthode est très précise mais il est conseillé de ne l'utiliser que si c'est absolument nécessaire, car Celles-ci ne sont pas redimensionnés à l'écran, or on doit toujours laisser la possibilité au visiteur de zoomer la page. La figure (3.12).
   — Indiquer **une taille relative** : en pourcentage, "em" ou "ex", cette technique a l'avantage d'être plus souple. Elle s'adapte plus facilement aux préférences de taille des visiteurs. Par défaut, les tailles sont en général réglés à 16px.
   — On peut spécifier de façon absolue la taille de la police en utilisant des mots clés . la valeur absolue consiste à juste dire si c'est petit, moyen ou grand sans spécifier la taille exacte et sans donner une référence. Les mots clés possibles par ordre croissant de la valeur sont :
     • xx-small
     • x-small
     • small
     • medium
     • large
     • x-large

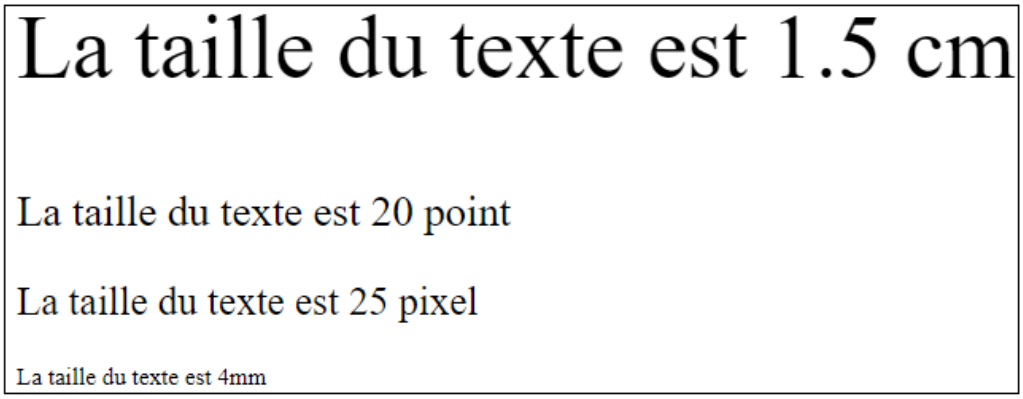
- xx-large

```html
<p class="taille2cm">La taille  du texte est 1.5 cm</p>
<p class="taille7pt"> La taille  du texte est 20 point</p>
<p class="taille50px"> La taille  du texte est 25 pixel</p>
<p class="taille4mm"> La taille  du texte est 4mm </p>
```

Code Html

```css
p.taille2cm{font-size: 1.5cm;}
p.taille7pt{font-size: 20pt;}
p.taille50px {font-size: 25px;}
p.taille4mm{font-size: 4mm;}
```

Code Css



Résultat

FIGURE 3.12 – Exemple de taille fixe.

```html
<p>Voici un paragraphe avec une taille standard celle de body</p>
<div class="parent">
  <p class= "paradiv"> Ici un paragraphe de taille 150% de la taille de  div</p>
</div>
```

Code Html

```css
.parent {
 font-family:arial, sans-serif;
 font-size:100%;
 }
.paradiv {
 font-size:150%;
 }
```

Code CSS

Voici un paragraphe avec une taille standard celle de body

Ici un paragraphe de taille 150% de la taille de div

Résutat

FIGURE 3.13 – Exemple de taille Relative.

En effet, le paragraphe hérite des propriétés de son parent, il faut donc bien faire attention aux valeurs déclarées. Il est recommander de déclarer la taille à 100% (.qui correspond à 16 pixel) et de ne la réduire (ou l'augmenter) que ponctuellement.
Même principe pour l'unité de mesure em, et ex, sauf ici les nombres décimaux sont autorisés, mais il faut tout simplement remplacer la virgule par un point.

3. **Alignement du texte :** L'alignement des textes en CSS est défini par la propriété '**text-align**'. Cette propriété peut prendre différentes valeurs à savoir :
   - left : aligner à gauche
   - right : aligner à droite

- center : centrer
- justify : justifier

Code CSS :

```
h1
{
text-align : center ;
}
```

4. **Gras, italique et souligner :**
   - **Mettre en Gras** La propriété CSS qui permet de mettre un texte en gras est **'font-weight'**, et prend les valeurs suivantes :
     - *bold* : le texte sera en gras.
     - *normal* : le texte sera écrit par défaut.

     Code CSS :

     ```
     h1
     {
     font-weight : bold ;
     }
     ```

   - **Mettre en italique** : La propriété CSS qui permet de mettre un texte en italique est font -style, et prend les valeurs suivantes :
     - *italic* : le texte sera mis en italique.
     - *normal* : le texte sera en format normale (par défaut).

     Code CSS :

     ```
     .normale
     {
     ```

font-style :normal ;

}

.enitalique

{

font-style :italic ;

}

Code HTML

** Police normale **

** Police italique **

— **Souligner et autres décorations :**

(a) La propriété CSS **text-decoration** permet entre autres de souligner le texte, et en plus d'autres fonctionnalités présenté sus les différentes valeurs qu'elle peut prendre :

- *underline* : souligné.
- *line-through* : barré.
- *overline* : ligne au-dessus.
- *blink* : clignotant. Ne marche pas sur tous les navigateurs (Internet Explorer et Google Chrome notamment).
- *none* : normal (par défaut).

(b) La propriété **text-transform** qui peut prendrer :

- *none* : normal.
- *capitalize* : met en majuscule la 1ère lettre d'un mot
- *Uppercase* : met en majuscules.

- *lowercase* : met en minuscules.

3.7.2 la couleur et le fond

1. **la gestion de la couleur :**

   En CSS pour colorer un texte, Utiliser la propriété **color** et indiquer la valeur de la couleur.

   Code CSS :

   H1

   {

   color : red ;

   }

   Il existe différentes façons d'indiquer la couleur en CSS :

   (a) **Indiquer le nom de la couleur :** Il existe seulement 17 couleurs qui peuvent être nommées directement par leur nom anglais comme valeurs de la propriété **color** comme le montre la figure(3.14).

FIGURE 3.14 – Les noms de couleurs

(b) **Notation hexadécimale des couleurs**

Dans la liste ci-dessus, 17 couleurs, c'est un nombre limité quand on sait qu'il ya des millions de couleurs mais ne peuvent pas être spécifiées par leur nom anglais. la notation hexadécimale xp (#ffA500). C'est une combinaison de lettres et de chiffres qui indiquent une

couleur. Elle est composée de 6 caractères qui contiennent des lettres de 0 à 9 ou des chiffres de A à F. Ces caractères sont toujours précédés du signe #. Ces lettres ou chiffres fonctionnent deux par deux. Les 2 premiers indiquent une quantité de rouge, les 2 suivants une quantité de vert, et les 2 derniers une quantité de bleu. En mélangeant ces quantités (qui sont les composantes Rouge-Vert-Bleu de la couleur) pour obtenir la couleur voulue.

(c) **La couleur en RGB :**

RGB(Red_Green_BLUE) Cela consiste à définir les couleurs avec des valeurs décimales comprises entre 0 et 255. Exemple : Si vous voulez que le texte de l'élément H1 soit en rouge, vous noterez :

Code CSS :

```
p {
color : rgb(255,0,0) ;
}
```

la notation RGB est beaucoup plus pratique que hexadécimale et qu'avec un logiciel de dessin tout simple comme Paint, vous pouvez trouver la valeur de la couleur que vous désirez.

Utiliser les pourcentages On peut aussi indiquer en pourcentages la valeur des composantes rouges, vertes et bleues d'une couleur. Il suffit juste de copier ces valeurs dans vos codes CSS comme ceci :

Code CSS :

```
p {
color : rgb(100%,0%,0%) ;
}
```

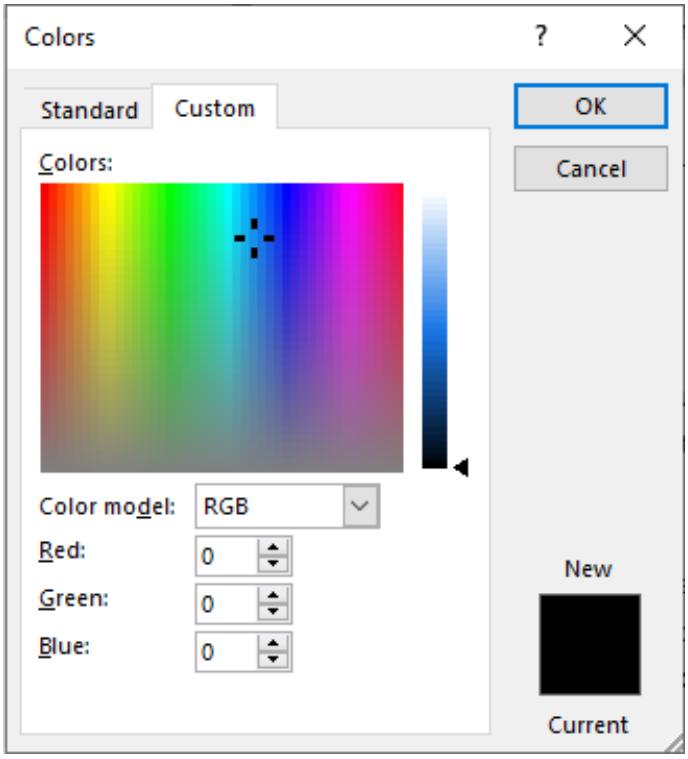


FIGURE 3.15 – La palette de couleurs dans Word

2. **La gestion de fond en CSS :** En CSS, il est possible de mettre un fond couleur ou image non seulement sur une page web mais aussi de mettre des fonds sur une partie de la page (les titres, les paragraphes, etc.) Les propriétés pour les fonds sont généralement précédées du mot background.
   (a) **La couleur de fond :** La propriété **background-color** sert à mettre une couleur de fond, Elle s'utilise de la même manière que la propriété color, c'est-à-dire que vous pouvez taper le nom d'une couleur, l'écrire en notation hexadécimale ou encore utiliser la méthode RGB. Pour mettre un fond gris sur toute la page web par exemple, on utilise **background-color** dans la balise **body**.
   Exemple :

body {

background-color : grey ; /* fond de page en gris*/

color : black ; /* couleur du texte en noir */

}

Si on désire mettre du fond uniquement sur une partie de la page web, il suffit d'utiliser les classes .

Exemple :

```
.fondcolorer{
            background-color: black;
            color: White ;
        }
h1
{
    color: red;
}
```

Code CSS

```
<h1 class="fondcolorer">Mon Titre </h1>
<h1> un autre titre h1  </h1>
```

Code Html

Résultat

FIGURE 3.16 – La couleur du fond

Uniquement le titre **h1** avec la classe 'fondcolorer' sera blanc sur un fond noir.

(b) **Image de fond :**

La propriété permettant d'indiquer une image de fond est **background-image**. Comme valeur, on doit lui donner le chemin de l'image url("@_de_l_image. jpg").

Code CSS :

body {

background-image : url("fond.jpg") ;

}

L'adresse indiquant où se trouve l'image de fond peut être écrite en absolu *(http ://...)* ou en relatif *(fond.png)* par rapport au *fichier .css* et non pas par rapport au *fichier .html*.

Il est conseille de placer l'image de fond dans le même dossier que *le fichier .css*

(c) **Image de fond fixé ou non :**

Quand on clique sur la barre de défilement, il est possible de rendre l'image de fond immobile. C'est-à-dire quel que soit le mouvement de la scrollbar, seuls les textes glisse. L'image de fond reste fixe. la propriété CSS pour avoir ce fond fixe est background-attachment ,elle peut prendre 2 valeurs :

• **scroll** : l'image de fond défile avec le texte (par défaut).

• **fixed** : l'image de fond reste fixe.

Code CSS :

body {

background-image : url("fond.jpg") ;

background-attachment : fixed ; /* Le fond restera fixe */

}

(d) **Répétition d'image :**

En CSS si l'image de fond de petite taille par défaut elle sera répétée en mosaïque.

Avec la **background-repeat** nous pouvons contrôler le mode de répétition selon les valeurs suivantes : • **no-repeat** : le fond ne sera pas répété. L'image sera donc unique sur la page.

• **repeat-x** : le fond sera répété uniquement sur la première ligne, horizontalement.

• **repeat-y** : le fond sera répété uniquement sur la première colonne, verticalement.

• **repeat** : le fond sera répété horizontalement et verticalement (par défaut).

Code CSS :

```
body {
background-image : url("fond.jpg") ;
background-repeat : repeat-y ;
}
```

Ce code CSS donnera une page avec un fond répété verticalement comme le montre la figure (3.17).

FIGURE 3.17 – La couleur du fond

(e) **Positionnement de l'image :**

On peut avoir différentes façons pour positionner l'image de l'arrière-plan à l'aide de la propriété **background-position**. Cette propriété est combinée directement avec **background-repeat : no-repeat ;** (un fond qui ne se répète pas).

La position du fond est calculée par rapport au coin supérieur gauche de l'élément (page ou paragraphe selon l'emplacement de l'insertion de l'image).

Pour mettre un fond placé à 40 pixels et 50 pixels en haut à gauche, la propriété aura comme valeur : **background-position :40px 50px ;**.

Il est aussi possible d'indiquer les valeurs à l'aide des mots clés suivants :

- **top** : en haut
- **bottom** : en bas
- **left** : à gauche
- **center** : centré
- **right** : à droite

On peut aussi combiner les mots Clés comme par exemple, **background-position : top left** ; ceci permet d'aligner une image en haut à gauche.

```css
body {
background-image: url("rosefond.jpg");
background-repeat: no-repeat;
background-position: top left;
}
h1{
color: white;

}
```

Code CSS

Résultat

FIGURE 3.18 – Image de fond placé en haut à gauche

3. **La transparence :**

   Il existe deux types de transparence des éléments. Une transparence complète de l'élément avec son fond en utilisant la propriété **opacity** et une transparence du fond de l'élément avec **la notion rgba**.

   La propriété **opacity**, elle peut prendre deux valeurs :

   • Avec une valeur de 1, l'élément sera totalement **opaque** : (par défaut).

   • Avec une valeur de **0**, l'élément sera totalement transparent.

   Code CSS :

   p

   {

   background-color :red ;

   color : black ;

opacity : 0.2 ;

}

Dans l'exemple précédant, tout le contenu du paragraphe sera transparent y compris le texte noir et le fond rouge. Voire la figure(3.19).

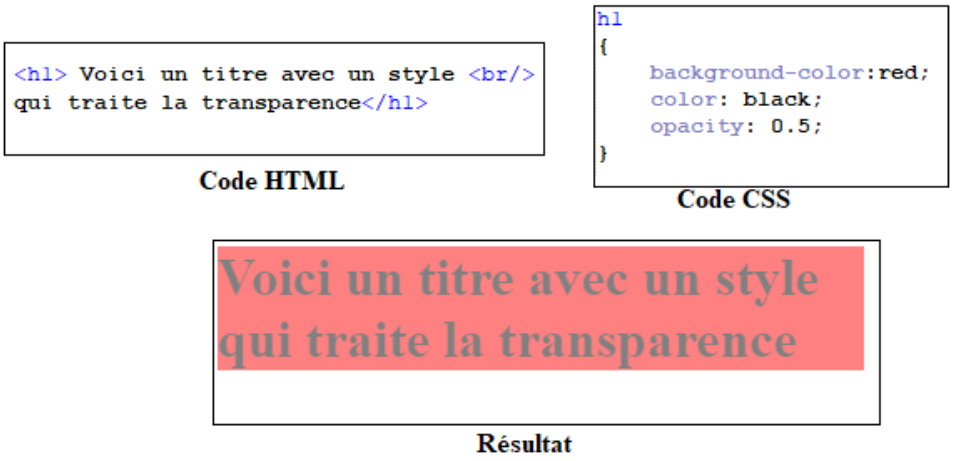


FIGURE 3.19 – Le texte est le fond est transparant avec la propriété opacity

La notion RGBA :

Il s'agit de la notation **RGB**, mais avec un quatrième paramètre : le niveau de transparence « alpha ». le même principe de opacity, la valeur 1 pour le fond opaque et 0 pour un fond transparent.

Code CSS :

h1{

color : black ;

background-color : rgba(255, 0, 0, 0.2) ;

}

le code précédant donne un résultat différent à celui de la propriété **opacity**. Ici le texte reste opaque et seulement le fond sera transparent, comme le montre la figure(3.20).

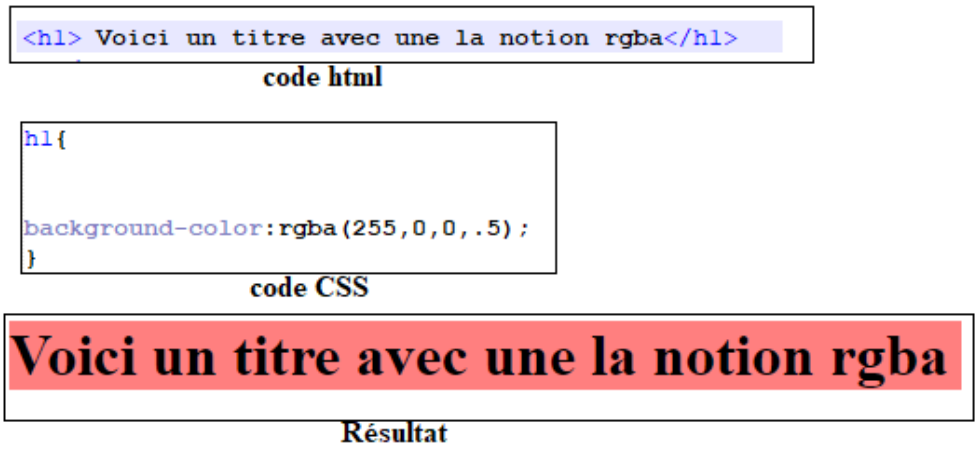


FIGURE 3.20 – Le texte reste opaque est le fond est transparant avec la propriété **rgba**

3.7.3 Les bordures et style

La propriété Border permet tracer des bordures, elle fonctionne comme la propriété **background**, on peut trouver **border-width, border-color, border-style.**

(a) **Border-style** Cette propriété nous permet de définir le style de bordure, elle peut prendre plusieurs valeurs :

- *none* : pas de bordure
- *solid* : un trait simple
- *dotted* : pointillés
- *dashed* : tirets
- *double* : bordure double
- *groove* : en relief
- *ridge* : effet 3D
- *inset* : Effet 3D mais le block est vu comme un creux
- *outset* : Effet 3D mais le block a l'air d'être surélevé)

Example : Code CSS :

```
h1 {
border-style : solid ;
}
```

```
<h1> Voici un titre encadré par une bordure</h1>
```

Code Html

```
h1{
border-style : solid ;
}
```

Code CSS

Voici un titre encadré par une bordure

Résultat

FIGURE 3.21 – Un titre avec une bordure solide

(b) **Border-color :** La couleur par défaut des bordures est en noire si on veut modifier sa couleur on utilise la propriété **Border-color** qui fonctionne du même principe que color (nom de la couleur, hexadécimale, ou RGB), voir la figure(3.22)

Code CSS : **.avecbordure {**

border-style : solid ;

border-color : blue ;

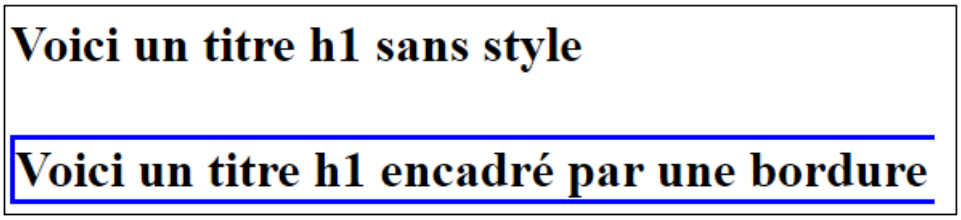
}

```
<h1> Voici un titre h1 sans style</h1>
<br/>
<br/>
<br/>
<h1 class="avecbordure"> Voici un titre h1 encadré par une bordure</h1>
```

Code Html

```
.avecbordure{
border-style : solid;
border-color : blue;
}
```

Code CSS



Résultat

FIGURE 3.22 – Un titre par défaut et un autre avec une bordure solide bleu

(c) **Border-width :** Vous avez aussi la possibilité de spécifier l'épaisseur de la bordure. On utilisant la propriété **border-width** , comme le montre la figure(3.23).

Code CSS :

```
.avecbordure{
border-width : 9Spx ;
border-color : blue ;
border-style : solid ;
}
```

```
<h1 class="avecbordure"> Voici un titre h1 encadré par une bordure</h1>
```

Code html

```
.avecbordure{
border-style : solid;
border-color : red;
border-width: 9px;
}
```

Code CSS

Voici un titre h1 encadré par une bordure

Résultat

FIGURE 3.23 – Un titre par défaut et un autre avec une bordure solide rouge épaisse

Combinaison de propriétés En CSS vous pouvez combiner plusieurs propriétés en une seule, vous pouvez utiliser une sorte de "super-propriété" appelée **border** qui peut prendre plusieurs valeurs combinées des propriétés vues précédemment *border-width, border-color, border-style*. Il faut juste séparer les valeurs par un espaces comme l'exemple suivant :

Code CSS :

```
.avecbordure {
border : 4px red solid ;
}
```

(d) **Bordure non complète** En CSS vous pouvez changer le style de la bordure, vous pouvez la mettre dans un des 4 cotés : en bas, en haut, à gauche, ou à droite en utilisant les propriétés suivantes :

- *border-top* : bordure en haut.
- *border-bottom* : bordure en bas.

- *border-left* : bordure à gauche.
- *border-right* : bordure à droite.

Code CSS :

```
.hautetbas {
border-top : 3px solid red ;
border-bottom :1px solid red ;
}
```

```
<h1 class="hautetbas"> Voici un titre avec une bordure non complète</h1>
<br/>
<br/>
<h1 class="avecbordure"> Voici un titre h1 encadré par une bordure</h1>
<br/>
<br/>
<h1> Voici un titre h1 sans style</h1>
```

Code Html

```
.hautetbas {
border-top: 3px solid red ;
border-bottom:1px solid red ;
}
```

Code CSS

Voici un titre avec une bordure non complète

Voici un titre h1 encadré par une bordure

Voici un titre h1 sans style

Résulat

FIGURE 3.24 – Un titre avec une bordure non complète ,un titre par défaut et un autre avec une bordure solide rouge épaisse.

(e) **Bordure arrondi** En CSS, la propriété **Border-radius** permet de créer des bordures avec des coins arrondis, Il suffit d'indiquer la taille de l'arrondi en pixels dans la sa valeur.

Code CSS :

```
h1 {
Border-radius : 10px ;
}
```

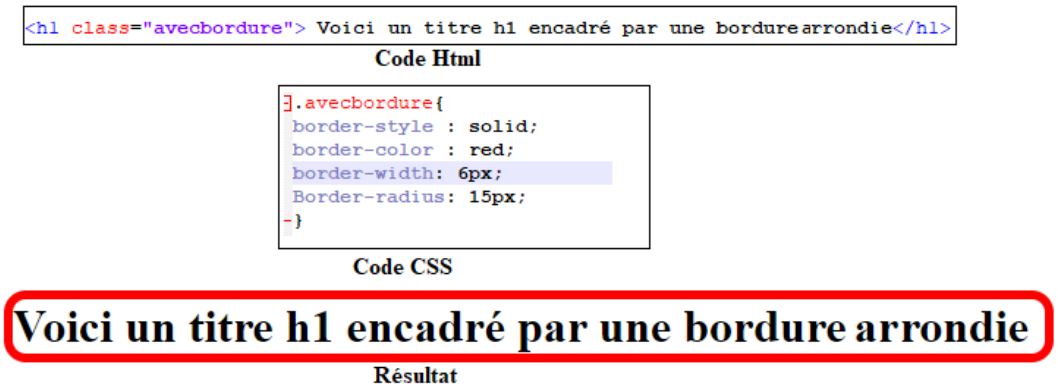


FIGURE 3.25 – Un titre avec une bordure arrondie rouge et épaisse.

3.7.4 Les marges :

Il existe deux types de marges :

• les marges intérieures : c'est L'espace entre le texte et la bordure.

• les marges extérieures :L'espace entre la bordure et le bloc suivant.

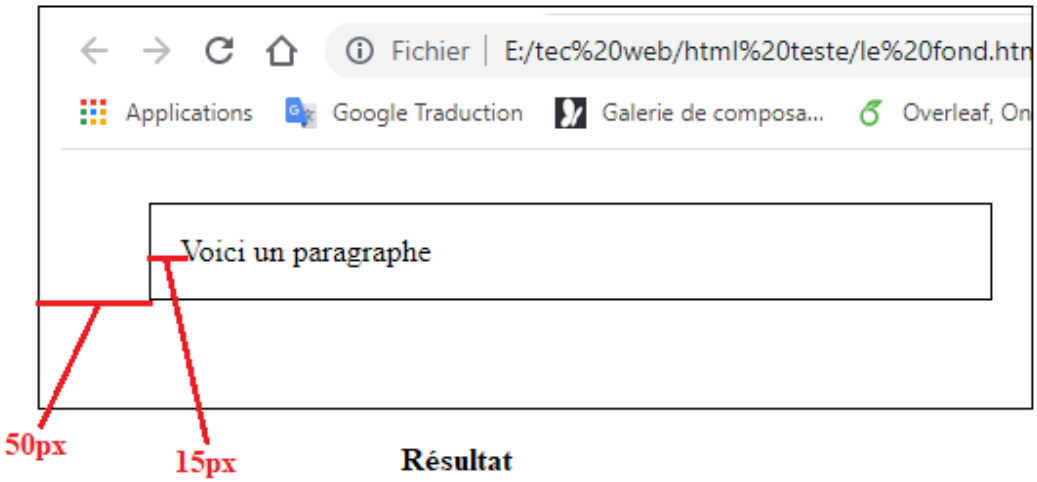
En CSS, on peut modifier la taille des marges avec les deux propriétés suivantes :

• **padding** : indique la taille de la marge intérieure. utiliser en général les pixels (px).

• **margin** : indique la taille de la marge extérieure. Là encore, on utilise le plus souvent des pixels.

```css
p
{
    width: 400px;
    border: 1px solid black;
    text-align: justify;
    padding: 15px; /* Marge interieur de 15px */
    margin: 50px; /* Marge extérieure de 50px */
}
```

Code CSS



Résultat

FIGURE 3.26 – Un titre avec une bordure arrondie rouge et épaisse.

Comme pour border il existe aussi des propriétés plus précises pour les quatre cotés.

Voici la liste pour margin :

- *margin-top* : marge extérieure en haut ;
- *margin-bottom* : marge extérieure en bas ;
- *margin-left* : marge extérieure à gauche ;
- *margin-right* : marge extérieure à droite.

Et la liste pour padding :

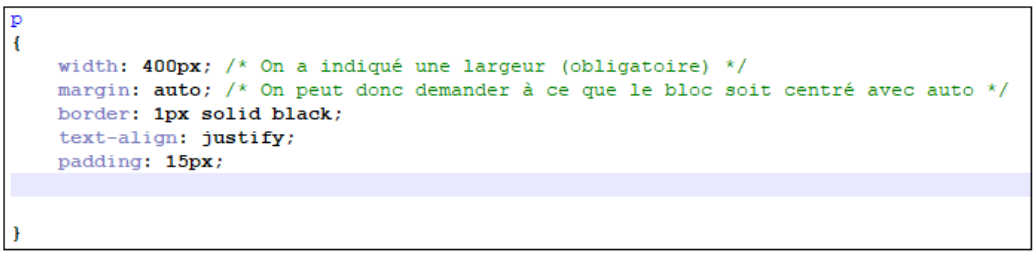
- *padding-top* : marge intérieure en haut ;
- *padding-bottom* : marge intérieure en bas ;
- *padding-left* : marge intérieure à gauche ;

• *padding-right* : marge intérieure à droite.

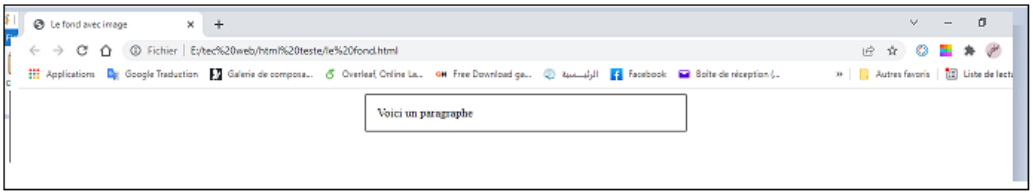
Pour rendre les blocs centré horizontalement utiliser la valeur **auto** pour **margin**

Code CSS :

```
p
{
width : 400px ; /* On a indiqué une largeur (obligatoire) */
margin : auto ; /* On peut donc demander à ce que le bloc soit centré horizontalement avec auto */
border : 1px solid black ;
text-align : justify ;
padding : 15px ;
}
```



Code Css



Résultat

FIGURE 3.27 – Le paragraphe est centré horizontalement avec la valeur **auto**

3.7.5 Création d'apparence dynamique :

A) **Au survol :**

Signifie : au passage de la souris sur l'élément, son apparence change, pour le réaliser utiliser le pseudo-format **:hover** .

```
<br/>
<a href="page1.html">Lien vers la page suivante</a>
<br/>
```

Code Html

```
a /* Le lien par défaut (non survolés) */
{
   text-decoration: none;
   color:blue;
   font-size:20px;
}
a:hover /* Apparence au survol des liens */
{
   text-decoration: underline;
   color: red;
   font-style: italic;
}
```

Code Css

1 Lien vers la page suivante

2 *Lien vers la page suivante*

Le résultat

FIGURE 3.28 – 1 : le lien sans survol, 2 : Au survol de la souris

B) **Au clic de la souris :**

C'est un effet pas trop visible car il très rapide, il s'applique juste au

moment du clic de la souris.

Pour le faire utiliser le **pseudo-format :active**.Voir figure (3.29).

```css
a:hover /* Apparence au survol des liens */
{
   text-decoration: underline;
   color:red;
   font-style: italic;
}
a /* Le lien par défaut (non survolés) */
{
   text-decoration: none;
   color:blue;
   font-size:20px;
}

 a:active /* Quand le visiteur clique sur le lien */
{
    background-color: gray;
}
```

Code Css

1 Lien vers la page suivante

2 *Lien vers la page suivante*

Résultat

FIGURE 3.29 – 1 : le lien par défaut, 2 : Au moment de clique de la souris

C) **Lorsque le lien a déjà été consulté**

le **pseudo-format :visited** permet de changer la couleur des liens déjà visités. Voir la figure(3.30)

```
a:visited /* Quand le visiteur a déjà vu la page concernée */
{
    color:green;}
```

Code CSS

Lien vers la page suivante

Résultat

FIGURE 3.30 – La couleur du lien se change dés la premiere viste

3.7.6 Positionnements en CSS :

Dans cette section nous allons voir les différentes façons pour changer le positionnement des éléments html avec des propriétés CSS.

1. La propriété **float :valeur** : permet d'extraire des éléments du flux de la page, ce qui signifie que le reste du contenu « coule »autour
   Elle prend 3 valeurs :
   -*left* :l'élément flottera à gauche.
   -*right* :l'élément flottera à droite.
   -*none* : (permet de remettre un élément dans le flux).

```html
<img src="wiki.jpg" class="imageflottante" alt="des" />
<p>Texte autour de l'image flottante. Texte autour de l'image flottante.<br/>
Texte autour de l'image flottante. Texte autour de l'image flottante.<br/>
Texte autour de l'image flottante. Texte autour de l'image flottante.<br/>
Texte autour de l'image flottante. Texte autour de l'image flottante.<br/>
Texte autour de l'image flottante. Texte autour de l'image flottante.<br/>
Texte autour de l'image flottante. Texte autour de l'image flottante.<br/>
```

Code Html

```css
.imageflottante
{
    float: left;
}
```

Code CSS

Texte autour de l'image flottante. Texte autour de l'image flottante.
Texte autour de l'image flottante. Texte autour de l'image flottante.
Texte autour de l'image flottante. Texte autour de l'image flottante.
Texte autour de l'image flottante. Texte autour de l'image flottante.
Texte autour de l'image flottante. Texte autour de l'image flottante.
Texte autour de l'image flottante. Texte autour de l'image flottante.
Texte autour de l'image flottante. Texte autour de l'image flottante.
Texte autour de l'image flottante. Texte autour de l'image flottante.
Texte autour de l'image flottante. Texte autour de l'image flottante.
Texte autour de l'image flottante. Texte autour de l'image flottante.
Texte autour de l'image flottante. Texte autour de l'image flottante.
Texte autour de l'image flottante. Texte autour de l'image flottante.
Texte autour de l'image flottante. Texte autour de l'image flottante.
Texte autour de l'image flottante. Texte autour de l'image flottante.

Le résultat

FIGURE 3.31 – Image qui flotte à gauche

Comme le montre la figure(3.31) le texte se positionne à coté de l'image et continu au dessous de l'image. Maintenant, pour que le texte ne soit plus à coté du flottant on utilise la propriété **clear** qui permet de poursuivre l'écriture du texte au dessous de l'élément flottant, elle peut prendre trois valeurs :

-*left* : le texte s'écrit en-dessous après un *float : left*, c-à-d il permet d'empêcher le contournement des blocs flottants à gauche.

-*right* : le texte se poursuit en-dessous après un *float : right* ;

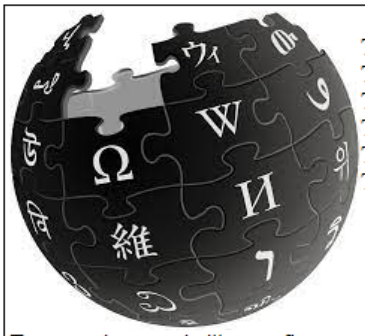
-*both* : le texte se poursuit en-dessous, que ce soit après un *float : left* ; ou après un *float : right* ;

```
<img src="wiki.jpg" class="imageflottante" alt="des" />
<p>Texte autour de l'image flottante. Texte autour de l'image flottante.<br/>
Texte autour de l'image flottante. Texte autour de l'image flottante.<br/>
Texte autour de l'image flottante. Texte autour de l'image flottante.<br/>
Texte autour de l'image flottante. Texte autour de l'image flottante.<br/>
Texte autour de l'image flottante. Texte autour de l'image flottante.<br/>
</p>
<p class="dessous">Texte en dessous de l'image flottante. <br/>
Texte en dessous de l'image flottante. Texte en dessous de l'image flottante.<br/>
Texte en dessous de l'image flottante. Texte en dessous de l'image flottante.<br/>
```

Code Html

```
.imageflottante
{
    float: left;
}
.dessous
{
    clear: both;
}
```

Code Css



Texte autour de l'image flottante. Texte autour de l'image flottante.
Texte autour de l'image flottante. Texte autour de l'image flottante.
Texte autour de l'image flottante. Texte autour de l'image flottante.
Texte autour de l'image flottante. Texte autour de l'image flottante.
Texte autour de l'image flottante. Texte autour de l'image flottante.
Texte autour de l'image flottante. Texte autour de l'image flottante.

Texte en dessous de l'image flottante. Texte en dessous de l'image flottante.
Texte en dessous de l'image flottante. Texte en dessous de l'image flottante.

Résultat

FIGURE 3.32 – Le texte en dessous de l'image

Il est possible d'utiliser des flottants pour créer une mise en page en colonnes.

2. La propriété **display :valeur** : En html existe deux types catégories

de balises :

• **block** : ce type de balise crée automatiquement un retour à la ligne avant et après, les blocks se positionne les uns en-dessous des autres peuvent être redimensionnés à laide de la propriété *width* exp de balise type block : **div, h1, p...**

• **inline** : ce type de balise ne crée pas de retour à la ligne, le texte qui se trouve à l'intérieur s'écrit donc à la suite du texte précédent, sur la même ligne exp : **a, span ,em ....**

La propriété display permet de transformer n'importe quel élément de votre page d'un type à l'autre.

Elle peut prendre les valeurs suivantes :

• *inline :* les eléments Se placent les uns à côté des autres.

• *block* : les Eléments Se placent les uns en-dessous des autres et peuvent être redimensionnés.

• *inline-block* :une propriété hybride qui permet à un élément d'avoir les propriétés d'un élément en ligne (pas de retour à la ligne après l'élément), mais avec les propriétés d'un bloc (possibilité d'avoir une dimension et des marges)

• *None* : l'élément sera masqué.

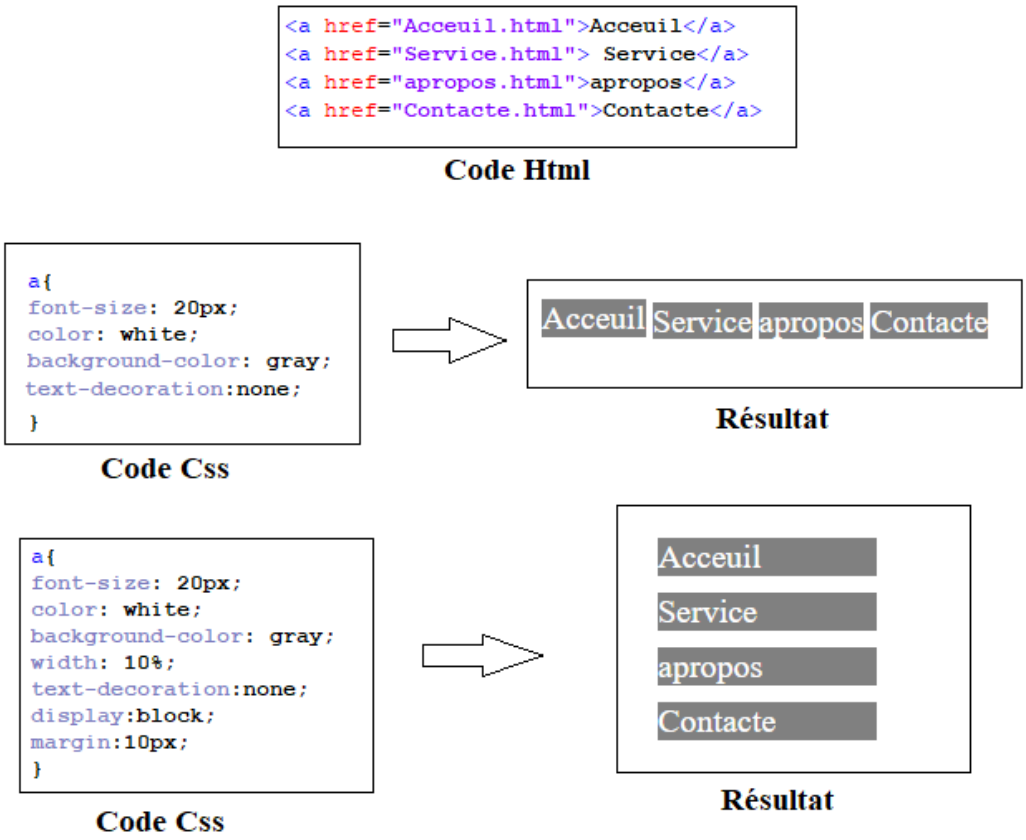


FIGURE 3.33 – Rendre la balise **a** sous forme de block

-Avec la **propriété display** nous avons pu rendre les liens sous forme de blocks avec une largeur flexible grâce à la **propriété width**, come le montre la figure (3.33).

-Dans la figure (3.34) nous avons utilisé la valeur **inline** pour rendre les balises h1 et p aligné dans la même ligne.

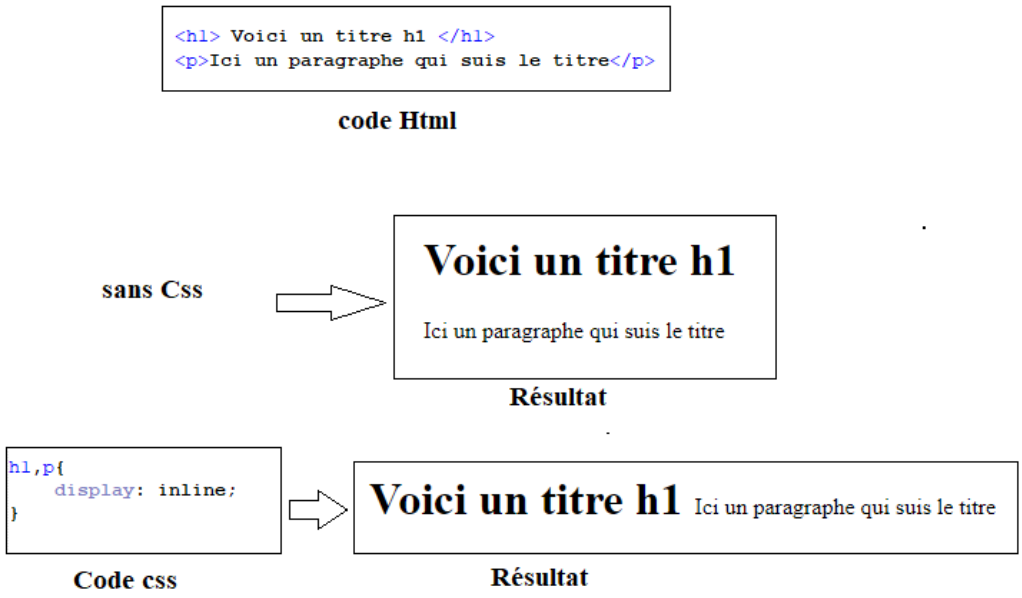


FIGURE 3.34 – Rendre les balises **h1** et **p** sous forme inline

- Pour la valeur les eléments positionnés les uns à côté des autres (comme les inlines) mais qui peuvent être redimensionnés (comme les blocs).Voir figure(3.35)

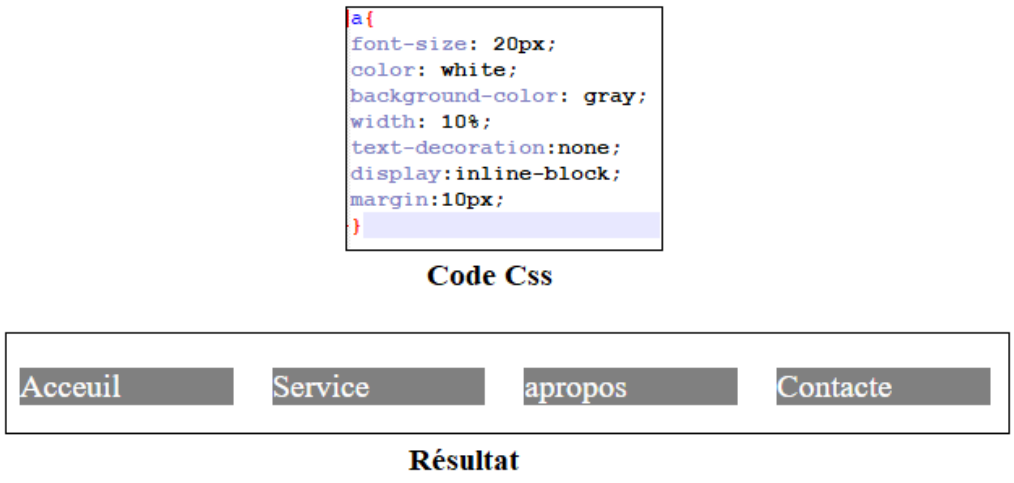


FIGURE 3.35 – La balise **a** sous forme de block et en inline en même temps

3. La propriété **position :valeur** : Cette propriété permet de positionner avec précision des éléments sur la page web , elle peut prendre les valeurs suivante :

   • ***static*** : positionnement par defaut

   • ***relative*** : positionnement relatif. permet de placer l'élément par rapport à sa position normale. Voir la figure (3.36)

   • ***fixed*** : positionnement fixe, l'élément restera visible si on descend dans la page comme background-attachment : fixed ; Voire la figure (3.37).

   • ***absolute*** : positionnement absolu n'importe où sur la page calculée par rapport au coin supérieur gauche du contenant (page ou l'élément parent positionné en absolu, fixe ou relatif). Voir figure(3.39).

   Ces trois dernières propriétés ont besoin de quatre autre propriétés pour préciser le positionnement :

   • *left* : valeur (en px ou %) position par rapport à la gauche.

   • *right* : position par rapport à la droite de la page ;

   • *top* : position par rapport au haut de la page ;

   • *bottom* : position par rapport au bas de la page.

   La figure(3.38) montre l'utilisation de top, left, right.

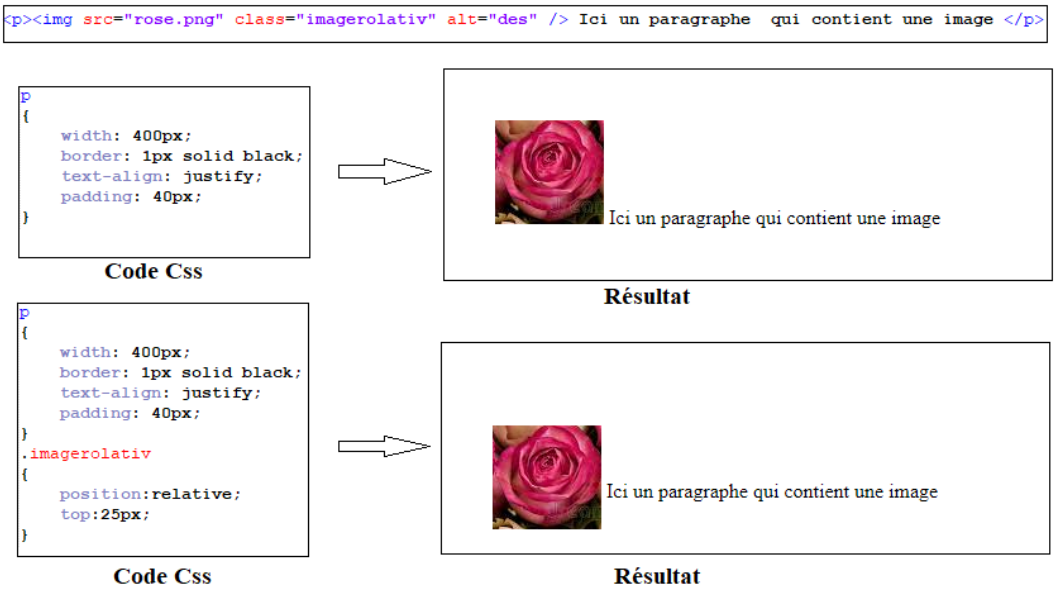
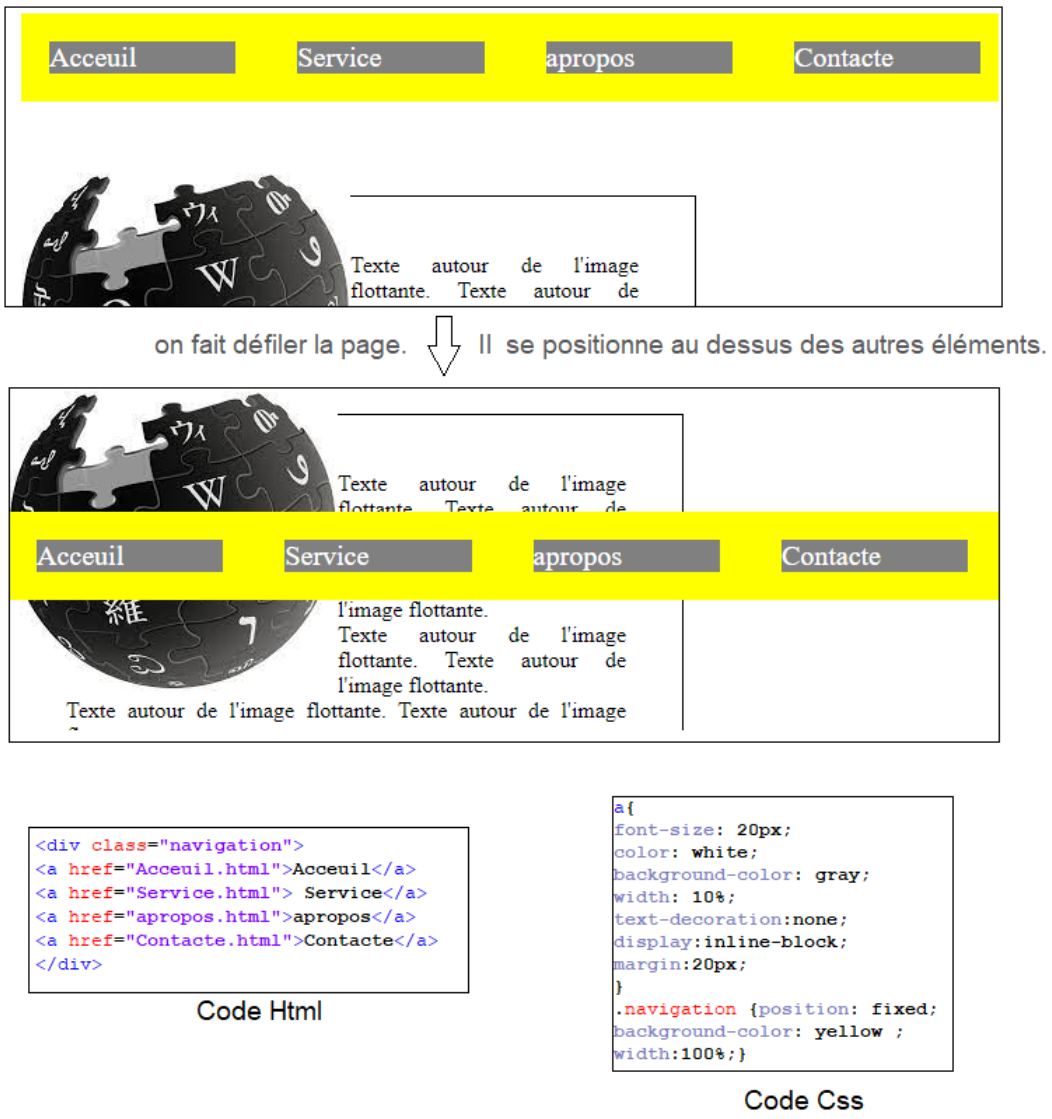


FIGURE 3.36 – La valeur **relative**




```html
<div class="navigation">
<a href="Acceuil.html">Acceuil</a>
<a href="Service.html"> Service</a>
<a href="apropos.html">apropos</a>
<a href="Contacte.html">Contacte</a>
</div>
```

Code Html

```css
a{
font-size: 20px;
color: white;
background-color: gray;
width: 10%;
text-decoration:none;
display:inline-block;
margin:20px;
}
.navigation {position: fixed;
background-color: yellow ;
width:100%;}
```

Code Css

FIGURE 3.37 – La barre de navigation reste toujours fixé en haut de page.

FIGURE 3.38 – La valeur **fixed**

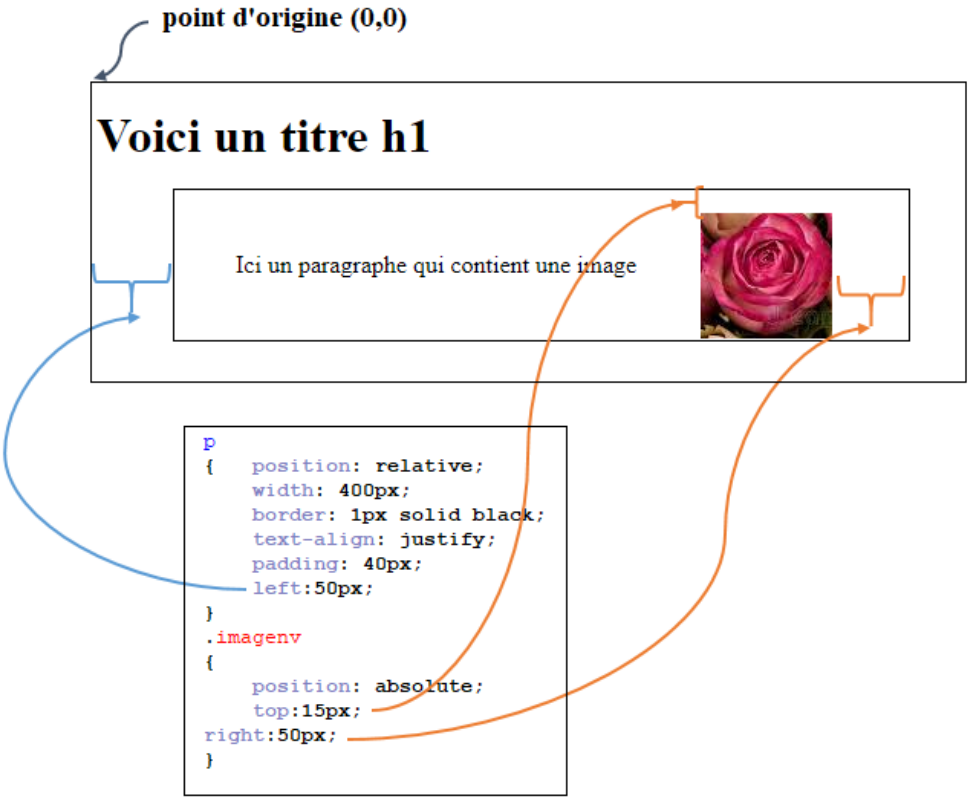


FIGURE 3.39 – La valeur **absolate**

3.8 Conclusion

Dans ce chapitre, nous avons établi une brève présentation des différentes propriétés CSS, qui permet de donner une vue colorer, arrangé et organisé des pages web HTML.

Deuxième partie : Web dynamique

Chapitre 4

PHP : Balises, Variables et Structure de contrôle

Sommaire

4.1 Introduction

Un site web dynamique c'st site web qui se change sans l'intervention du webmaster (le développeur du web).

HTML c'est le langage de base pour créer des sites web. Il été créé pour produire des sites web statiques. Cependant, il ne suffit pas pour réaliser des sites web dynamiques. Il faut le compléter avec d'autres langages tels que le langage php.[4].

Comme le montre la figure (4.1) le site web développé en **php** sera stocker dans des machines de haute puissance appelé **Serveur**, cette machine joue le rôle de point de diffusion du site web aux clients.(*Cette opération appelé hébergement du site web*)

Seul le serveur est capable de lire du PHP. à l'exécution du langage php le serveur produit du code HTML, car les ordinateurs des visiteurs sont incapables de comprendre le code PHP, ils ne connaissent que le langage HTML. Enfin le navigateur du client (*Chrome par exemple*) traduit ce code HTML(*diffusé par le serveur*) en un contenu lisible pour le visiteur (*du texte, des images et des liens*).

FIGURE 4.1 – Requête Client Serveur

Pour créer cette structure client\serveur nous allons utiliser le logiciel **Wampserver** après l'installation l'écone du logiciel doit être verte comme le montre la figure 4.2 (Dans la science de TP nous allons voir plus de détail sur ce logiciel).

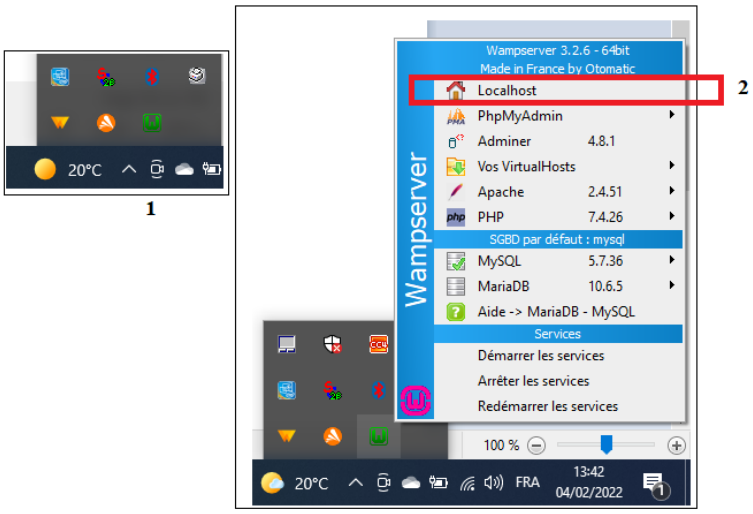


FIGURE 4.2 – Le logiciel Wampserver : 1 l'écone est verte, 2 Localhost pour exécuter le code php

Le code php doit être enregistrer dans le dossier **www** du logiciel **wampserver**, puis l'ouverture de la page php sera depuis **le Localhost** du logiciel.

4.2 Les balises PHP

1. Pour écrire du code php utiliser une nouvelle balise qui commence par **<?php** et se termine par **?>** au sein du code HTML, **Mais** dans l'enregistrement du fichier utiliser l'extension **.php** à la place du **.html** ou dés le début, choisir le langage php dans **Notepad++** et c'est celui qui génère l'extension.

   La balise <?php ?> peut être placée n'importe où dans le code html.

   Code PHP :

   <body>

   <h2>Teste</h2>

   <?php /* Insérer du code PHP ici */ ?>

** Bla bla bla **

2. **L'instruction de base echo :**

   Nous avons introduis le mot instruction car le PHP c'est un vrai langage de programmation ; tandis qu'à html qui est un langage de description.

   Toute instruction en PHP se termine par un point-virgule.

   L'instruction d'affichage du texte en php est l'instruction **echo** appelée aussi **print**. Cependant, le mot echo le plus couramment utilisée.

   Le message à afficher sera entre deux guillemets **""**.

   Code php :

   <?php

   echo "Bla Bla Bla " ;

   ?>

   Pour sauter la ligne utiliser la balise

FIGURE 4.3 – Saut de ligne

Pour écrire des guillemets utiliser un backslash \ juste avant les écrire.

Code php :

< ?php

echo " Voici un guillemet \" " ;

?>

```php
<!doctype html>
<html lang="fr">
    <head>
    <title>  Page PHP</title>
    <meta charset="UTF-8"/>
    <link rel="stylesheet" href="fccs.css" />

    </head>
    <body>
            <h1> Premier code PHP</h1>
            <br/>
            <br/>
            <?php
            echo "voici un guillemet \".";
            ?>
    </body>
</html>
```

Le code pagephp.php

Le résultat

FIGURE 4.4 – Affichage des guillemets

3. **Écrire un commentaire :**

   • Les commentaires monolignes : taper 2 slashs : //. ensuite votre commentaire.

   • Les commentaires multilignes :taper /* votre commentaire en plu-

sieurs ligne puis */

4.3 Les variables

4.3.1 Définition :

Une variable c'est une information stockée temporairement dans un espace mémoire .
En effet, elles n'existent que durant l'exécution du code PHP ; elles ne persistent pas dans le temps. Si on souhaite sauvegarder durablement des informations, il faut se tourner vers une autre méthode : base de données, fichiers, cookies par exemple.
En PHP, les variables sont représentées par nom sous forme d'une chaîne de caractères, ayant toujours comme premier caractère, le caractère dollar ($).
Dans le nom d'une variable, ne jamais utiliser d'accent ni d'espace.
Une variable est typée et elle peut prendre 4 types présentés ci-dessous :

- **String** : une chaîne de caractères écrite entre guillemets ou entre apostrophe, exp : « voici du texte »
- **Integer** : nombre entier exp :1
- **bool** (booléen qui signifie une logique binaire genre vrai ou faux).
- **float** : nombre réel, ce sont les nombres à virgule, ils doivent être écrits avec un point au lieu de la virgule exp : 1.87.

4.3.2 Affecter une valeur à une variable :

Pour assigner une valeur à une variable, on utilisera l'opérateur = comme le montre a figure (4.5).

Code PHP :

< ?php

$nom = "ALGERIE" ;

$chiffre = 52 ;

$test = true ;

$nbreel=112.6 ;

?>

FIGURE 4.5 – Affichage des valeurs des variables

4.3.3 Afficher la valeur d'une variable

Pour afficher une variable il suffit d'utiliser l'instruction echo.

Code php :

< ?php

$chiffre = 20 ;

echo $chiffre ;

?>

Et pour afficher une variable au sein du texte utiliser la façon suivante :

Code php :

```
<?php
$nombre= 400;
echo " j'ai gagné $nombre DA ";
?>
```

Vous pouvez aussi utiliser les apostrophes (les guillemets simples), mais cette fois-ci il faut écrire les variables en dehors de ces guillemets et séparer entre les différentes parties de la phrase par un point.Comme le montre la figure (4.5).

Les variables sont sensibles à la casse(minuscule ou majuscule) des caractères. Une différence dans les noms de variables s'avère être une erreur courante en développement.

```php
<?php
    $color = 'vert';
    echo 'Mon stylo  est '.$color;
    echo 'Mon stylo est '.$COLOR;
    echo 'Mon stylo est '.$Color;
?>
```

Code PHP

Mon stylo est vert

Notice: Undefined variable: COLOR in C:\wamp64\www\teste\codephp.php on line *43*

Call Stack				
#	Time	Memory	Function	Location
1	0.0189	361976	{main}( )	...\codephp.php:0

Mon stylo est

Notice: Undefined variable: Color in C:\wamp64\www\teste\codephp.php on line *44*

Call Stack				
#	Time	Memory	Function	Location
1	0.0189	361976	{main}( )	...\codephp.php:0

Mon stylo est

Résultat

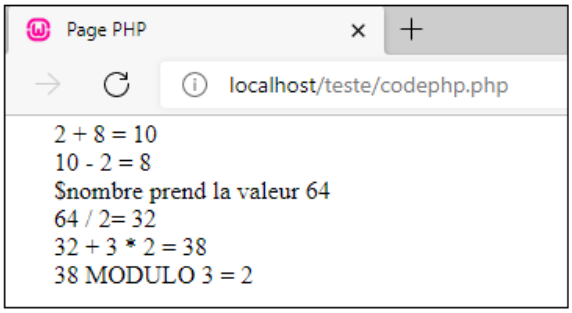
FIGURE 4.6 – Sensibilité à la Casse : le résultat de la première ligne est correcte, les deux autre lignes génère une erreur car les deux variable $COLOR et $Color sont considérées non déclarés

4.3.4 Opération arithmétique sur les variables

Sur les variables qui contiennent des chiffres, nous pouvons effectuer les différentes opérations arithmétique d'addition de soustraction et de multiplication.

```php
<?php
$nombre = 2 + 8; echo "2 + 8 = $nombre <br/>"; // $nombre prend la valeur 10
$nombre = $nombre - 2; echo "10 - 2 = $nombre <br/>"; // $nombre prend la valeur 8
$nombre = $nombre*$nombre;echo  '$nombre'." prend la valeur $nombre <br/>"; .
// $nombre prend la valeur64
$nombre = $nombre/ 2; echo "64 / 2= $nombre <br/>";// $nombre prend la valeur 32
$nombre = $nombre + 3 * 2; echo " 32 + 3 * 2 = $nombre <br/>";// $nombre prend la valeur 38
$nombre = $nombre % 3; echo " 38 MODULO 3 = $nombre <br/>  ";// $nombre prend le reste du 38/3
?>
```

Code Php



Résultat

FIGURE 4.7 – Les opérateurs sur les variables

4.4 Les structures de contrôles

Dans les structures de contrôle nous utilisons des opérateurs de comparaisons et des opérateurs logiques. Il existe plusieurs opérateurs ceux que nous utiliserons dans ce cours sont listés dans les tableaux ci-après.

==	True si la valeur est égale à
!=	True si la valeur est différente de
<	True si la valeur est strictement inférieur à
>	True si la valeur est strictement supérieure à
<=	True si la valeur est inférieur ou égale à
>=	True si la valeur est supérieure ou égale à

TABLE 4.1 – Les opérateurs de comparaison

&&	true seulement si tous les tests de condition sont évalués à true
||	true seulement si l'un des tests de condition est évalué à true
!	Inverse la logique d'un test de condition

TABLE 4.2 – Les opérateurs Logiques

4.4.1 les Conditions :

Ls conditions permettent d'exécuter une partie de code selon si une condition est vérifiée ou non. Elle permette de vérifier par exemple si le nom d'utilisateur est correcte ou pour pouvoir accéder ou non son compte.

4.4.1.1 L'instruction if

— C'est le type de condition le plus simple :

Code PHP :

```
<? php
$nombre = 5;
if ($nombre > 0)
{
echo $nombre.' le nombre est positif';
}
?>
```

— Pour traiter le cas contraire de la condition **if** avec une autre condition, on utilise l'instruction **elseif** comme le montre l'exemple ci-après :

code PhP

```
<? php
```

```
$nombre = 5;
if ($nombre > 0)
{
echo $nombre.' un nombre positif';
}
elseif ($nombre < 0)
{
echo $nombre.' le nombre est négatif';
}
?>
```

Après un premier **if**, il est possible d'avoir plusieurs **elseif** les uns à la suite des autres. Le premier **elseif** qui sera vérifié et qui sera évalué à *true* sera exécuté.

— Pour un cas contraire sans nouvelle condition, on utilise l'instruction **else** :

code PhP

```
<? php
$nombre = 5;
if ($nombre > 0)
{
echo $nombre.' un nombre positif';
}
elseif ($nombre < 0)
{
echo $nombre.' un nombre négatif';
}
else
```

```
{
echo $nombre.' égale à zéro';
}
```
?> Le résultat qui sera affiché est : *'5 un nombre est positif'* car la valeur de la variable $nompre est égale à 5.

— Nous pouvons aussi utiliser les conditions multiples enexploittant les opérateurs logique vu dans le tableau (4.1).

code PhP

```
<? php
$nombre = 5;
if ($nombre >= 0 && $nombre < 10)
{
echo $nombre.' est compris entre 0 et 9';
}
elseif($nombre >= 10 && $nombre < 20)
{
echo $nombre.' est compris entre 10 et 19';
}
else
{
echo $nombre.' est plus grand que 19';
}
?>
```

4.4.1.2 Switch :

Le switch représente exactement la même chose qu'une succession d'un if et de plusieurs elseif.

La structure de switch est la suivant :

Code PHP :

```
<?php
$ouiounon = "oui";
switch ($ouiounon) { case 'oui' :
echo 'Votre réponse est oui.';
break;
case 'non' :
echo 'Votre réponse est non.';
break;
default :
echo 'ni oui ni non, il faut entrer une des deux!!!';
}
?>
```

Le résultat de ce code est *'Votre réponse est oui.'* car la variable $ouiounon = "oui".

Attention ! Notez bien l'utilisation de **break** dans chaque cas de votre switch. Si celui-ci est omis, tous les messages s'afficheront.

4.4.2 Les boucles

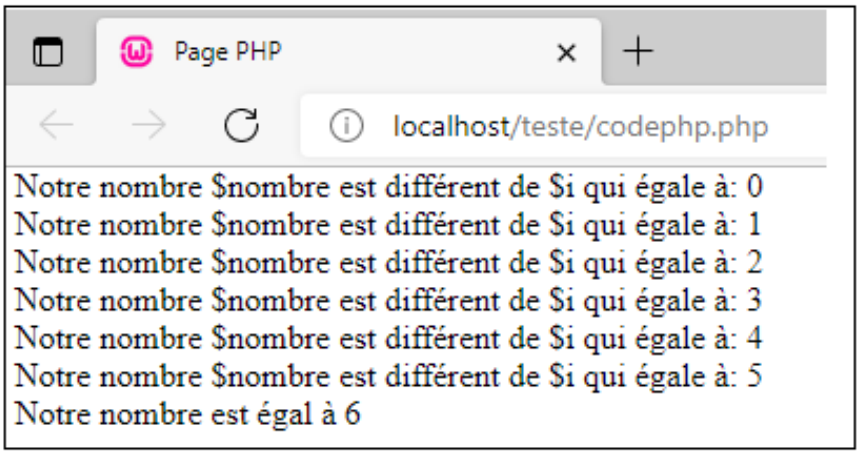
Les boucles permettent de répéter l'exécution d'une portion de code un certain nombre de fois en fonction d'un *test de condition*. Tant que ce dernier est vérifié, alors la portion de code s'exécute.

4.4.2.1 While

L'instruction while permet d'exécuter un block de code (une série d'instructions) **tant que** une condition est vérifiée.

Code PHP :

```
<?php
$nombre = 6;
$i=0;
while ( $i < $nombre ) { echo 'Notre nombre $nombre est différent de $i qui égale à : '. $i . '<br />';
$i++;
} echo 'Notre nombre est égal à '.$i;
?>}
```



Résultat

FIGURE 4.8 – Le résultat de la boucle **while**

4.4.3 Do..while

La boucle **do...while** commence par exécuter son block de code avant de vérifier sa condition donc le block sera exécute au moins une fois.

Code PHP :

```
<?php
$b = 6;
do {
echo 'La valeur de la variable $b est : '. $b .'<br/>';
$b++;
} while($b <= 10);
?>
```

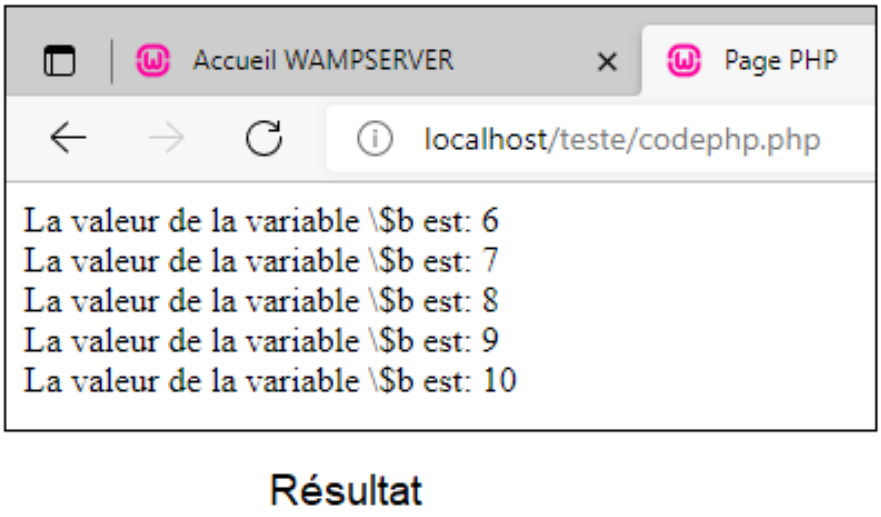


FIGURE 4.9 – Le résultat de la boucle **do..while**

4.4.4 For :

La boucle **for** est un autre type de boucle. avec une syntaxe assez particulière :

Code PHP :

```
<?php
for($b = 1 ;$b <= 10 ;$b++)
{
echo 'La valeur de la variable $b est : '. $b .'<br/>' ;
}
?>
```

4.4.5 Foreach

Foreach est un autre type de boucle. Elle est utile pour les variables de type array (tableaux). Nous aborderons cette boucle dans le prochain chapitre.

4.5 Conclusion

Dans ce chapitre, nous avons vu les bases sur le fonctionnement des variables, les conditions et es boucles en PHP. Ces thématiques sont essentielles et incontournables pour développer en PHP ou dans un autre langage de programmation [4]

Chapitre 5

les tableaux, les formulaires, et les fichiers

Sommaire

5.1 Introduction

Dans ce chapitre, nous allons voir les tableaux et fichiers qui permettent de stoker une grande masse de données. Comment récupérer et manipuler (afficher, stocker, etc.) les données récoltées via les formulaires.

5.2 Les tableaux

Les tableaux sont un type de variable particulier. Ils permettent de stocker plusieurs valeurs dans une seule variable.
Pour déclarer un tableau on utilise la fonction array() qui permet de créer une variable de type array.

Code PHP :

```
< ?php
$nom = array() ;
?>
```

Dans un tableau, chaque valeur est reliée par défaut à une clé numérique. On parle alors de **tableau numéroté ou indexé**.En PHP, un tableau numéroté commence toujours par la clé 0.

5.2.1 Initialiser un tableau :

Il existe trois méthodes pour initialiser les valeurs dans un tableau.

1. Utiliser la clé numérique :

   Code PHP :

   ```
   < ?php
   $nom = array() ;
   $nom[0] = "boulam" ;
   $nom[1] = "bousad" ;
   $nom[2] = "benkaf" ;
   ?>
   ```

2. Sans écrire les numéros des clés, le php va sélectionner un numéro à

chaque élément par ordre de création. Cette méthode moins utilisé car elle est moins lisible.

Code php :

```
< ?php
$nom = array() ;
$nom[] = "boulam" ;
$nom[] = "bousad" ;
$nom[] = "benkaf" ;
?>
```

3. L'ors de la création du tableau, on initialise les valeurs.

   Code php :
   ```
   < ?php
   $nom = array ('boulem', 'bousad', 'boukaf') ;
   ?>
   ```

Au lieu d'utiliser des chiffres pour les indices nous pouvons utiliser des chaînes de caractères. On parle alors de **tableau associatif**.

Code php :

```
< ?php
$nom = array() ;
$nom[mohamed] = "boulam" ;
$nom[ahmed] = "bousad" ;
$nom[mostafa] = "benkaf" ;
?>
```

Et pour utiliser la troisième méthode, on utilise un => pour associer chaque étiquette (la clé)avec sa valeur.

Code php :

```
< ?php
```

$nom = array ('mohamed'=>'boulem','ahmed'=> 'bousad','mostafa'=>
'boukaf') ;
?>

5.2.2 Affichage du tableau :

1. Pour afficher un élément d'un tableau on utilise l'instruction echo.

   Code PHP :

   echo $nom [0] ;

   echo $noms['mohamed'] ;

2. Pour un Affichage complet du tableau on utilise la boucle for :

   Code PHP :

   < ?php

   $nom = Array() ;

   $nom[0] = "boulam" ;

   $nom[1] = "bousad" ;

   $nom[2] = "benkaf" ;

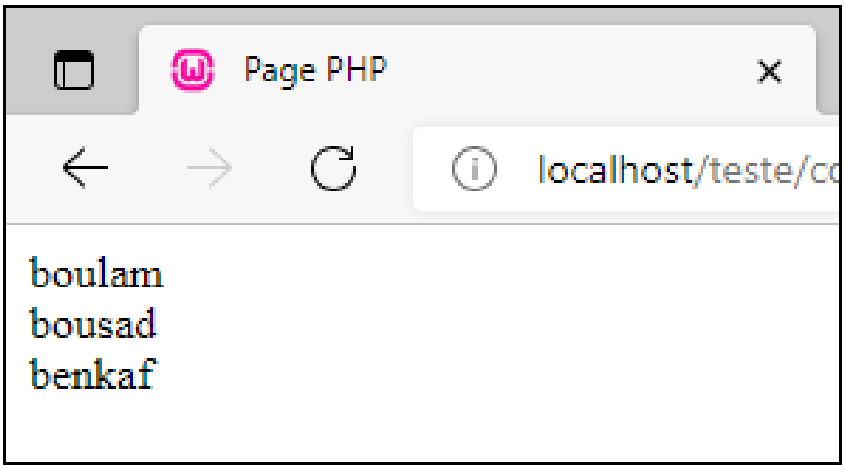
   for ($n=0 ; $n< 3 ; $n++)

   {

   **echo $nom[$n] .'
' ;**

   }

   ?>



Résultat

FIGURE 5.1 – Affichage du tableau avec la boucle For

La boucle for est faisable uniquement pour les tableaux numérotés mais pour les tableaux associatifs ca ne marche, on utilise la boucle **foreach**.

3. **La boucle foreach :**

La boucle foreach permet de parcourir un tableau avec un indice chaine de caractères. Elle passe le tableau ligne par ligne, puis elle prend la valeur de chaque élément et la mettre dans une variable temporelle à votre choix.

La structure de foreach est la suivante :

Code PHP :

```
< ?php
$noms = array ('mohamed'=>'boulem','ahmed'=> 'bousad','mostafa'=>
'boukaf') ;
foreach($noms as $element) { echo $element.'<br/>' ;
```

```
}
?>
```

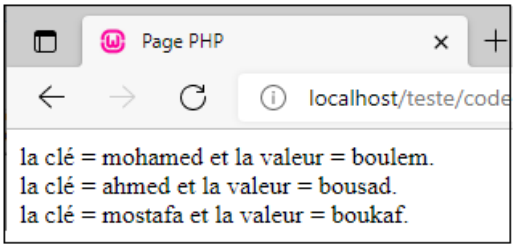
On peut aussi récupérer la clé de l'élément. On doit dans ce cas écrire foreach comme ceci :

Code php :

```
< ?php
$noms = array ('mohamed'=>'boulem','ahmed'=> 'bousad','mostafa'=>
'boukaf') ;
foreach($noms as $cle=>$element) {
echo 'la clé = '.$cle. ' et la valeur = '.$element.'. <br/>';
}
?>
```

```php
<?php
    $noms = array ('mohamed'=>'boulem','ahmed'=> 'bousad','mostafa'=> 'boukaf');
    foreach($noms as $cle=>$element)
        { echo 'la clé = '.$cle. ' et la valeur = '.$element.'. <br/>';
        }
?>
```

Code PHP



Résultat

FIGURE 5.2 – Affichage du tableau clé et valeur avec la boucle foreach

5.2.3 Fonction spécifique pour les tableaux

1. La fonction : **Print_r :** Elle permet d'afficher le tableau en entier sans boucle.

   Code php :

   < ?php

   $noms = array ('mohamed'=>'boulem','ahmed'=> 'bousad','mostafa'=> 'boukaf') ;

   echo '<pre>';

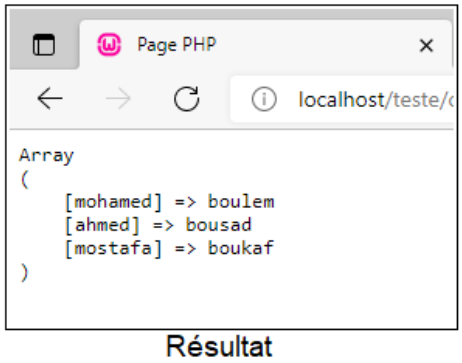
   print_r($noms) ;

   echo '</pre>';

   ?>

```php
<?php
    $noms = array ('mohamed'=>'boulem','ahmed'=> 'bousad','mostafa'=> 'boukaf');
    echo '<pre>';
    print_r($noms);
    echo '</pre>';
?>
```

Code PHP



Résultat

FIGURE 5.3 – Affichage du tableau avec print_r

2. La fonction **array_key_exists** : elle permet de rechercher si une clé existe dans le tableau ou non, elle renvoie la valeur true ou false.

Code php :

```
<?php
$noms = array ('mohamed'=>'boulem','ahmed'=> 'bousad','mostafa'=>
'boukaf') ;
if (array_key_exists('mohamed', $noms))
{
echo 'La clé "mohamed" se trouve dans le tableau noms !';
}
if (array_key_exists('khaled', $noms))
{
echo 'La clé "khaled" se trouve dans le tableau noms !';
}
?>
```

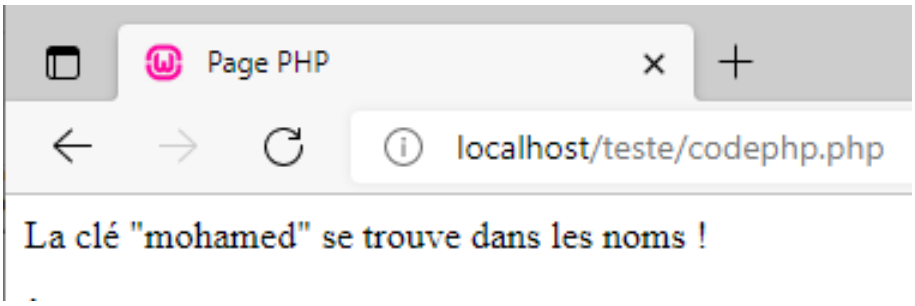


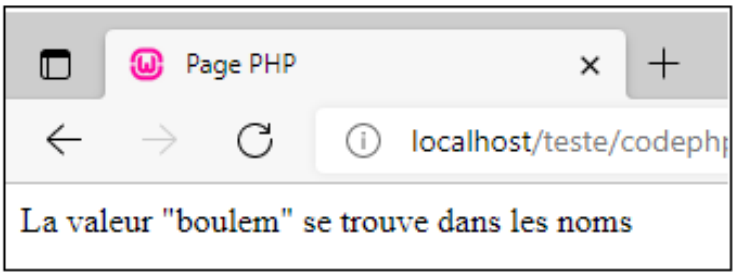
FIGURE 5.4 – La recherche de la clé avec la fonction **array_key_exists**

3. La fonction **in_array** : cherche une valeur dans le tableau ou non.

Code php :

```
<?php
$noms = array ('mohamed'=>'boulem','ahmed'=> 'bousad','mostafa'=>
'boukaf') ;
if (in_array('boulem', $noms))
{
```

```
echo 'La valeur "boulem" se trouve dans les noms !';
}
if (in_array('boutek', $noms))
{
echo 'La valeur "boutek" se trouve dans les noms !';
}
?>
```



Résultat

FIGURE 5.5 – La recherche d'une valeur avec la fonction **in_array**

5.3 les fichiers :

Les fichiers permettent de sauvegarder des données sur disque et les rendent en quelque sorte permanente à la différence des variables.
– On peut les utiliser plusieurs fois dans des sessions ultérieures.
En php, il y a une multitudes de fonctions pour manipuler les fichers à savoir : la lecture, l'écriture.
Tout d'abord on doit créer le fichier, de préférence dans le même dossier contenant la page web, sinon on doit respecter les noms des sous dossiers ou les dossiers parents dans son chemin.

- La création initiale du fichier se fait par un éditeur de texte tel que :bloc-notes et on doit l'enregistrer dans le même répertoire que la page PHP sous extension **.txt**.

5.3.1 Les fonctions PHP sur les fichiers

5.3.1.1 Ouvrir

Pour ouvrir un fichier on utilise la fonction fopen(chaine nomdufichier, chaine mode) Le mode indique le type d'opération qu'il sera possible d'effectuer sur le fichier après ouverture, qui peut prendre les valeurs suivante :

- **r** : ouvre le fichier en **lecture seule**, et place le pointeur au **début** du fichier.
- **w** : ouvre le fichier en **écriture seule** ; réduit la taille du fichier à 0 (c-à-d elle efface l'ancien contenu du fichier avant d'écrire) ; place le pointeur au début du fichier. Si le fichier n'existe pas, on tente de le créer.
- **a** : ouvre en **écriture seule** ; place le pointeur de fichier **à la fin** du fichier file. Si le fichier n'existe pas, on tente de le créer.
- **r+** : ouvre le fichier en **lecture et écriture**, et place le pointeur au **début** du fichier.
- **w+** : ouvre en **lecture et écriture** ; réduit la taille du fichier à 0 ; place le pointeur de fichier au début du fichier. Si le fichier n'existe pas, on tente de le créer.
- **a+** : ouvre en **lecture et écriture ;** place le pointeur de fichier à la fin du fichier. Si le fichier n'existe pas, on tente de le créer.

En générale, il est utile de tester si l'ouverture de fichier s'est bien déroulée ou non et stopper le script PHP si y aura un échoue d'ouverture :

Code PHP :

```
<?
if (!$fiche = fopen(""monfichier.txt","r"))
{
echo "Echec de l'ouverture du fichier";
exit;// pour stopper le script
}
else
{
echo 'Le reste du code si lóuverture est bien faite';
}
?>
```

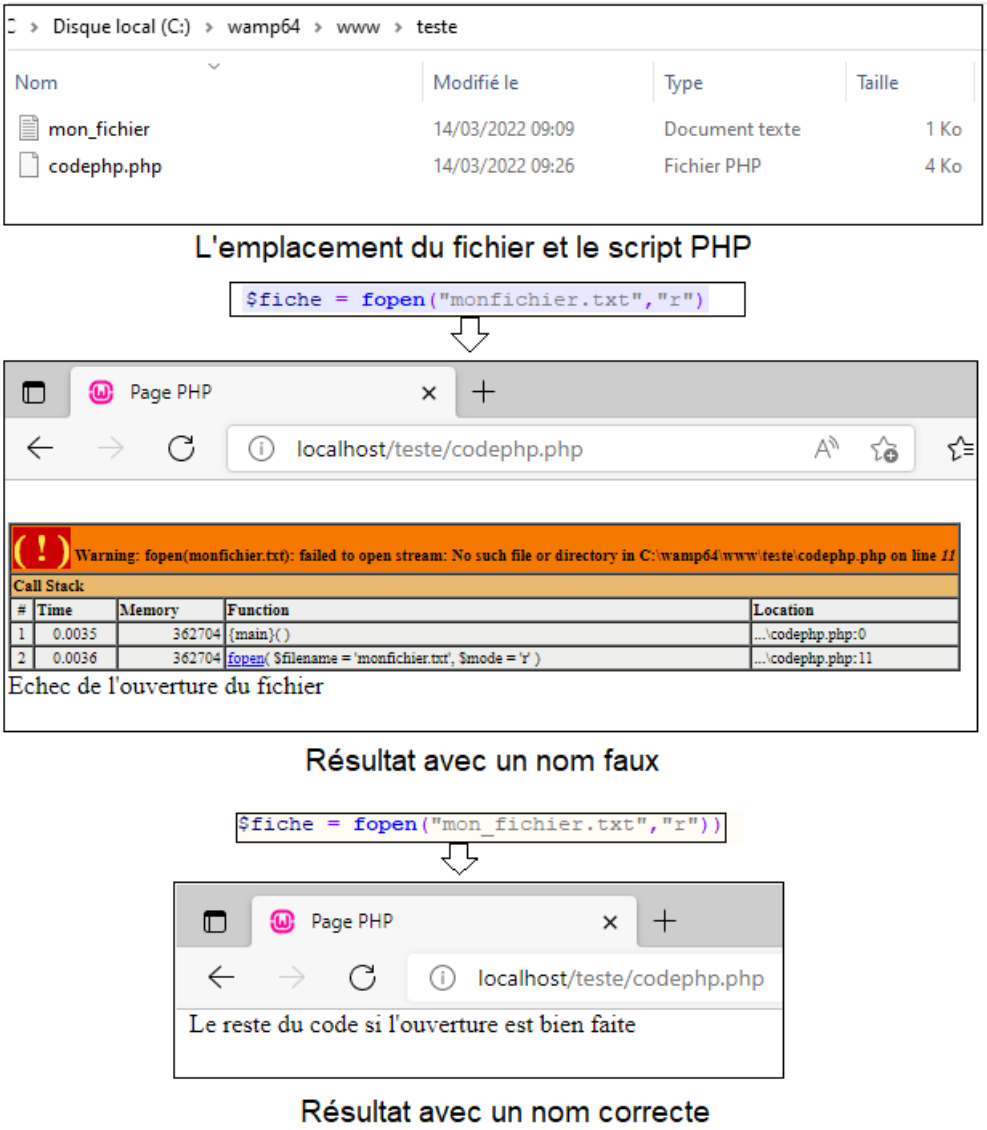


FIGURE 5.6 – Etat initial du fichier Compteur.txt

Pour fermer un ficher ouvert on utilise la fonction **fclose()**.

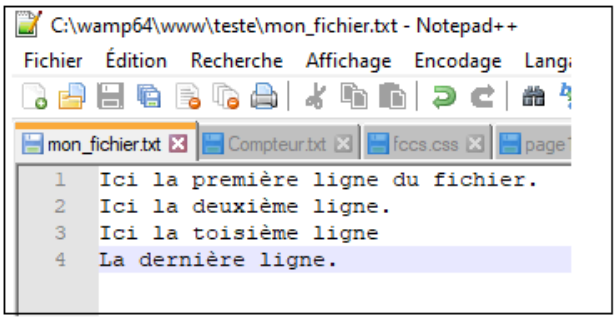
5.3.1.2 Lire et Ecrire

Une fois que le fichier a été ouvert avec le mode désiré, il est possible de lire son contenu et d'y écrire des informations grâce aux fonctions :
-**fputs()** (aussi parfois appelée **fwrite()**, les deux noms sont équivalents, elles permettent d'écrire une chaîne de caractères dans le fichier.
-**fgets()** permettant de récupérer une ligne du fichier. La fonction accepte un paramètre pour fixer la longueur de la ligne.
La fonction **fgets()** récupère à chaque appel une nouvelle ligne du fichier, donc , pour récupérer l'intégralité du contenu d'un fichier on doit l'insérer dans une boucle **while**.
La boucle while travail tant-que la fin du fichier n'a pas été atteinte pour vérifier cela on utilise la fonction **feof()**.

Code PHP :

```
< ?php
if ( !$fiche = fopen("mon_fichier.txt","r"))
// on ouvre le fichier en mode lecture avec le pointeur au debut du fichier
{
echo "Echec de l'ouverture du fichier" ;
exit ;
}
else {
while( !feof($fiche))
{
// On récupère une ligne
$Ligne = fgets($fiche,255) ;
// On affiche la ligne
```

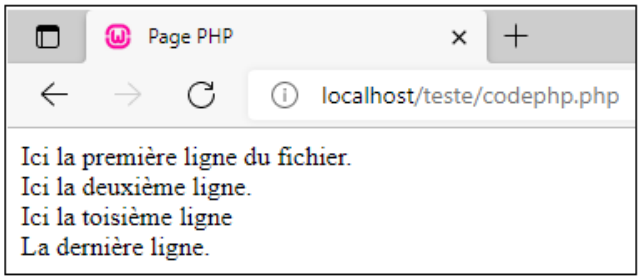
```
echo $Ligne.'<br/>';
}
fclose($fiche) ; // On ferme le fichier
}
?>
```



Contenu du fichier mon_fichier.txt

```php
<?php
if (!$fiche = fopen("mon_fichier.txt","r"))
// on ouvre le fichier en mode lecture avec le pointeur au debut du fichier
{
echo "Echec de l'ouverture du fichier";

exit;

}
else {
    while(!feof($fiche))
    {
    // On récupère une ligne
        $Ligne = fgets($fiche,255);

    // On affiche la ligne
        echo $Ligne.'<br/>';

    }
    fclose($fiche); // On ferme le fichier
}

?>
```

Code PHP



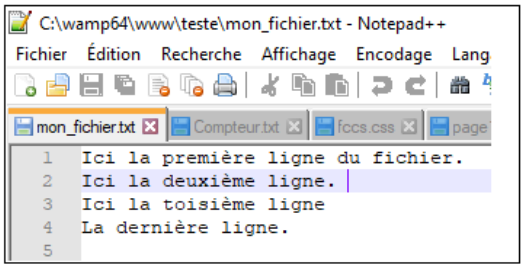
Résultat

FIGURE 5.7 – Récupérer l'intégralité du contenu d'un fichier avec la boucle **while** et la fonction **fgets()**

Maintenant pour écrire dans un ficier on doit l'ouvrire en mode écriture **"w"** pour une écriture au début du ficchier et en mode **"a"** pour une écritre à la fin du fichier.

Code PHP :

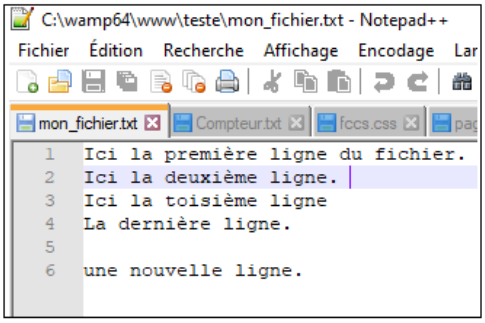
```
<?php
$fiche = fopen("mon_fichier.txt","a");
// on ouvre le fichier en mode écriture avec curseur à la fin du fichier.
$nom="une nouvelle ligne."; // la vaiable qu'on va rajouter dans la fin du fichier
fputs($fiche, "\ n"); // on va a la ligne
fputs($fiche,$nom); // on écrit la valeur de la variable.
fclose($fiche); // On ferme le fichier
?>
```



Le contenu du fichier "mon_fichier.txt" avant l'écriture

```php
<?php
$fiche = fopen("mon_fichier.txt","a");
// on ouvre le fichier en mode écriture avec curseur à la fin du fichier.
    $nom="une nouvelle ligne."; // la vaiable qu'on va rajouter dans la fin du fichier
    fputs($fiche, "\n"); // on va a la ligne
    fputs($fiche,$nom); // on écrit la valeur de la variable.
    fclose($fiche); // On ferme le fichier


?>
```

Code PHP



Le fichier "mon_fichier après l'execution du scripte PHP.

FIGURE 5.8 – Ecriture dans un fichier

5.3.1.3 Exemple

-Afin de mettre les fichiers en pratique, supposons que nous avons créé un fichier **Compteur.txt** avec Notepad++. Ce fichier va représenter un mini compteur du nombre de visites de la page web. Ce fichier nous allons le placer dans le même répertoire que le script PHP.

Supposons que ce fichier texte contient initialement le chiffre 0. Figure (5.9)

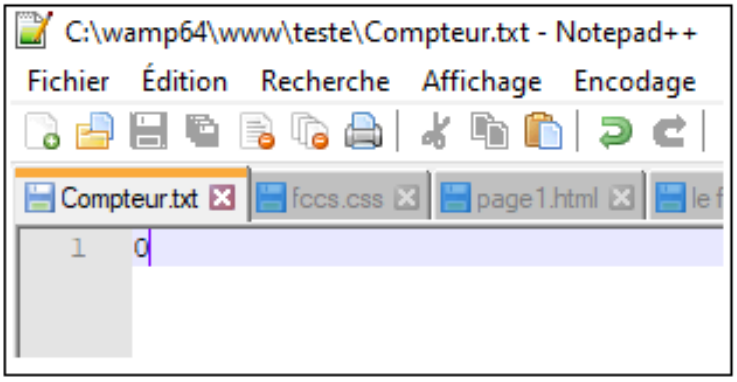


FIGURE 5.9 – Etat initial du fichier Compteur.txt

Code php :

```
<?php
// on ouvre le fichier avec fopen
$fichier = fopen ("compteur.txt", "r+");
// Instruction 2
$compteurs = fgets ($fichier);
// Instruction 3
$compteurs = $compteurs + 1;
// Instructions 4
fseek ($fichier, 0);
fclose ($fichier);
$fichier = fopen ("Compteur.txt", "r+");
fputs ($fichier, $compteurs);
// Instrcution 5
fclose ($fichier);
// Instrcution 6
echo 'Ce site est visité '.$compteurs.' fois !';
?>
```

Explications

- **Instruction 1** : on ouvre le fichier compteur.txt en lecture et en écriture.
- **Instruction 2** : on lit le contenu du fichier et on place ce contenu (qui est donc le nombre de visiteurs de notre page) dans la variable $compteur.
- **Instruction 3** : on augmente le nombre de visites de 1.
- **Instruction 4** : on doit placer le pointeur au début du fichier pour remplacer la valeur ancienne par la nouvelle valeur de $compteur pour cela on doit fermer le fichier et le réouvrir en mode **r+** qui est un mode qui assure l'emplacement du pointeur au début du fichier l'ors de l'ouverture.

  La première ouverture en mode **r+** ne suffit pas puisque on est dans la même session, le pointeur se met en première position uniquement l'ors de l'ouverture du fichier avec **fopen()** en suite il passe au fur et à mesure à la position suivante.
- **Instruction 5** : grâce à l'instruction fputs(), on écrit dans notre fichier la nouvelle valeur correspondant au nombre de visites.
- **Instruction 6** : on ferme le fichier.
- **Instruction 7** : on affiche le nombre de visites de notre page ! ! !

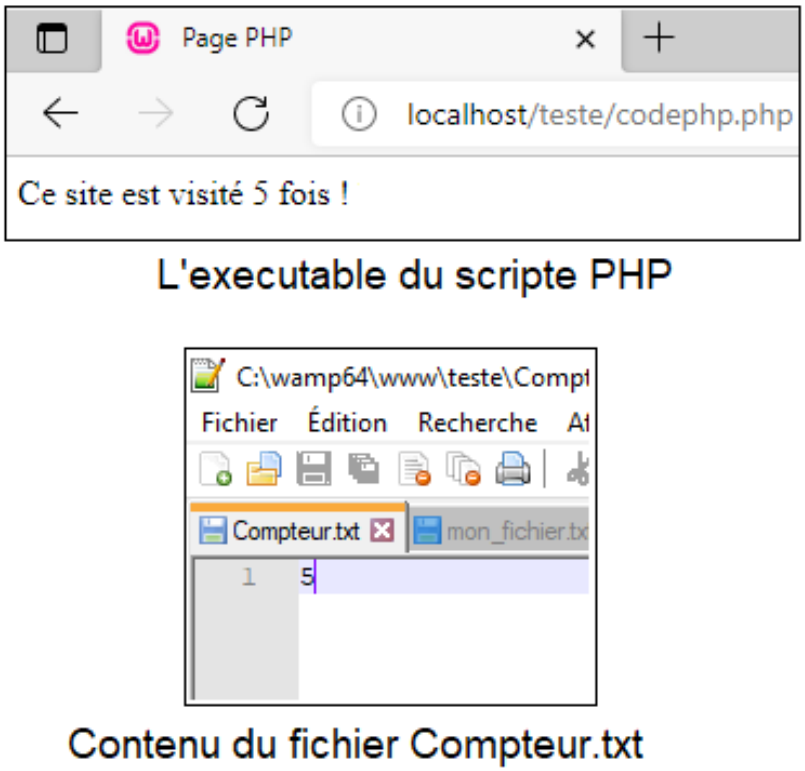


FIGURE 5.10 – Contenu du fichier Compteur.txt qui se change automatiquement après chaque actualisation de la page web (l'exécutable du scripte PHP)

En fin, on peut remplacer les instructions 4 par la fonction **fseek ();** qui permet de faire le retour au debut du fichier sans fermer et réouvrir le fichier, commme le montre le code suivant :

Code php :

```
<?php
// on ouvre le fichier avec fopen
$fichier = fopen ("compteur.txt", "r+");
// Instruction 2
$compteurs = fgets ($fichier);
```

```
// Instruction 3
$compteurs = $compteurs + 1 ;
// Instructions 4
fseek ($fichier, 0) ;
// Instructions 5
fputs ($fichier, $compteurs) ;
// Instrcution 5
fclose ($fichier) ;
// Instrcution 6
echo 'Ce site est visité '.$compteurs.' fois !' ;
?>
```

5.4 Récupérer les données d'un formulaire

Le HTML nous permet de créer nos formulaires mais pour récupérer et manipuler les données envoyées, nous allons besoin d'utiliser du PHP.
Soit le formulaire suivant :

Code php :

```
<form method="post" action="traitement.php">
<p>Entrez votre pseudo :</p> <input type="text" name="pseudo"/>
<br/>
<p>Entrez votre :</p> <input type="text" name="ville"/><br/>
<input type="submit" name="valider" value="OK"/>
</form>
```

Comme rappel l'attribut **method** indique la page PHP qui va traité les données envoyées et l'attribut **action** indique la façon d'envoie.

La page traitement.php doit récupérer les données saisis dans ce formulaire Cette page va automatiquement créer un **array** au nom un peu spécial : $_POST. Il s'agit d'un **array associatif** dont les clés correspondent aux noms des paramètres envoyés en formulaire.

Dans cet exemple nous avons trois variables :

$_POST [pseudo] : cette variable va contenir la valeur entrée dans pseudo.

$_POST [ville] : cette variable va contenir la valeur entrée dans ville.

$_POST [valider] : si elle existe ca veut dire qu'il y a eu clic sur la validation, et si elle n'existe pas, qu'il n'y a pas eu clic.

En php pour vérifier qu'une variable existe ou non on fait appel à une fonction un peu spéciale : **isset()**. Cette fonction nous allons nous en servir pour afficher un message spécifique si la validation est faite ou non.

Code php :

```
<?php
if(isset($_POST['valider']))
{ echo 'Salut !, je sais ton pseudo tu est : '. $_POST['pseudo'].' de
'. $_POST['ville']; echo'<br/>Bienvenue sur mon site !';
} else {
echo ' tu doit renseigner votre pseudo et votre ville dans la page
précédente ';
}?>
```

Si on exécute la page de traitement avant la page du formulaire le **else** va être déclencher.

FIGURE 5.11 – Page de Traitement PHP pour récupérer les données d'un formulaire

La figure (5.11)présente les codes Html et Php nécéssaire pour faire un formulaire, il est à noter que l'execution des codes est faite au niveau du **Localhost** du logiciel **Wampserver**.

Pour les boutons radio, case à coché et les listes déroulante voici un exemple de code qui les englobes tous. Voir les figures : (5.12)(5.13)(5.14) :

```
    <form name="inscription" method="post" action="traitement.php" >
    <label for="ps"><b>Entrez votre pseudo : </b></label> <br/>
    <input type="text" name="pseudo" id="ps"/>
    <br/>
    <br/>
    <label for="vil"><b>Entrez votre ville :</b></label> <br/>
    <input type="text" name="ville" id="vil"/><br/>
    <h3>Votre civilité: </h3>
    <input type="radio" name="civilite" id="made" value="mademoiselle">
    <label for="made">Mademoiselle</label><br/>
    <input type="radio" name="civilite" id="mad" value="madame" checked>
    <label for="mad">Madame</label><br/>
    <input type="radio" name="civilite" id="mon" value="monsieur">
    <label for="mon">Monsieur</label><br/><br/>
    <label for="pays"><b> Votre pays:</b></label>
                <select name="pays" id="pays">
                    <option value="france">France
                    <option value="italie">Italie
                    <option value="algerie" selected>Algerie
                </select>
    <br/>
    <h3>Plateforme(s)</h3></td>
    <input type="checkbox" name="materiel" value="windows" id="windows" checked>
    <label for="windows">Windows </label>    <br/>
    <input type="checkbox" name="materiel" value="mac" id="mac">
    <label for="mac">Macintosh</label>     <br/>
    <input type="checkbox" name="materiel" value="unix" id="unix">
    <label for="unix">Unix</label>
          <br/>
    <input type="submit" name="valider" value="OK"/>
</form>
```

FIGURE 5.12 – Code HTML pour les boutons radio, case à coché et les liste déroulante

```
<?php
if(isset($_POST['valider'])){
    echo 'Salut!, je sais ton pseudo tu est: '
    . $_POST['pseudo'].' de '. $_POST['ville'].'</br>';
    if(isset($_POST['civilite'])){
        echo 'Tu est '.$_POST['civilite'].'</br>';}
    if(isset($_POST['pays'])){
    echo 'Votre ville est en '.$_POST['pays'].'</br>';}
    if(isset($_POST['materiel'])){
        echo 'Et tu as comme plateforme '.$_POST['materiel'].'</br>';}
}
else {
echo ' Tu doit renseigner votre pseudo et votre ville dans la page précédente';}
?>
```

FIGURE 5.13 – Code PHP pour les boutons radio, case à coché et les liste déroulante

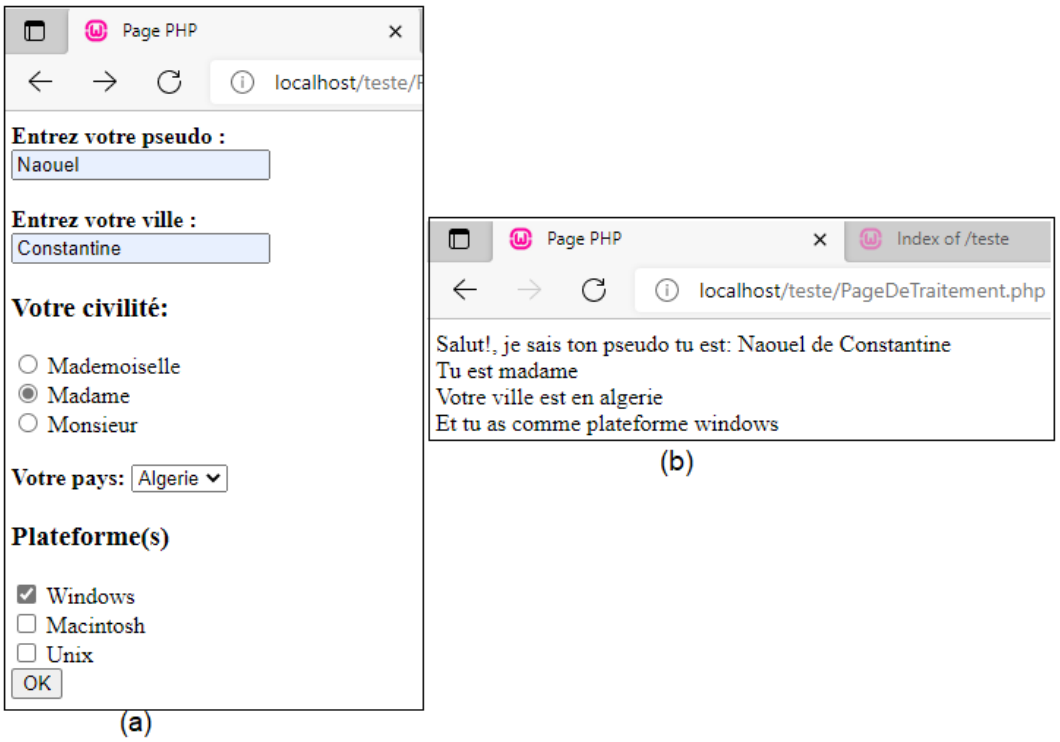


FIGURE 5.14 – (a) Résultat HTML, (b) Résultat de l'envoie du fichier.

Explication :

$_POST['civilite'] le code PHP retourne la valeur correspondante au bouton sélectionné. Il ne faut pas oublier de spécifier **une valeur** aux champs radio.

Ainsi **$_POST['pays']**, le code PHP retourne la valeur correspondante à l'option sélectionnée. IL ne faut pas oublier de spécifier **une valeur** aux options du menu de choix.

5.4.1 Uploader un fichier en HTTP sur un serveur avec PHP :

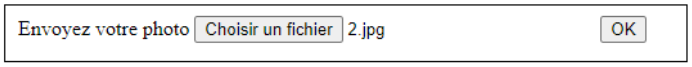
La création du formulaire : C'est un formulaire simple à une exception qu'il faut ajouter l'attribut **enctype="multipart/form-data"** à la balise

<**form**> pour envoyer le fichier, sinon il ne sera jamais envoyer.

```
<form name="inscription" method="post" action="PageDeTraitement.php" enctype="multipart/form-data">
<label for="fichier">Envoyez votre photo</label>
<input type="file" id="fichier" name="fichier">
<input type="hidden" name="taille_max" value="10000">
<input type="submit" name="valider" value="OK"/>
```

Code HTML

Résultat



Résultat après séléction d'une image

FIGURE 5.15 – La balise **form** pour aploader un fichier

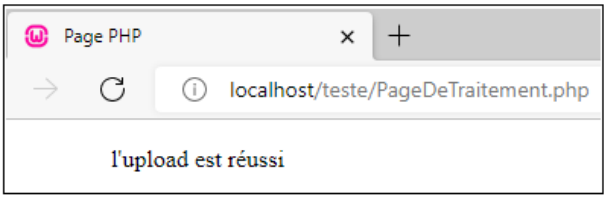
La figure (5.15) montre le code html nécessaire pour faire uplaoder un fichier.

Maintenant pour récupérer le fichier, le php crée une variable **$_FILES['nom_du_fichier']** Dans notre cas **$_FILES['fichier']**, cette variable peut prendre plusieurs valeurs :

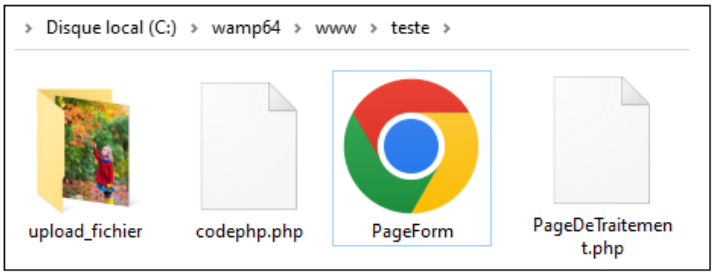
- Le nom du fichier choisi : $_FILES['nom_du_fichier']["name"] ;
- Le nom temporaire sur le serveur : $_FILES['nom_du_fichier']["tmp_name"] ;
- Le type du fichier choisi : $_FILES['nom_du_fichier']["type"] ;
- Le poids en octets du fichier choisit : $_FILES['nom_du_fichier']["size"] ;
- Un code de l'erreur si jamais il y en a une : $_FILES['nom_du_fichier']["error"] ;

```php
<?php
// récuperer un fichier ---->

//on vérifie que le champ est bien rempli:
if(!empty($_FILES["fichier"]["name"]) AND $_FILES['fichier']['error'] == 0)
{    // Testons si l'extension est autorisée
    $infosfichier = pathinfo($_FILES['fichier']['name']);
    $extension_upload = $infosfichier['extension'];
    $extensions_autorisees = array('JPG', 'JPEG', 'gif', 'png','jpg');
    if (in_array($extension_upload, $extensions_autorisees))
    {
        //chemin qui mène au dossier qui va contenir les fichiers uplaoded
        // vous devez le créer avant d'executer la page de traitement
        $chemin = "./upload_fichier/" ;
        if(copy($_FILES["fichier"]["tmp_name"], $chemin.$_FILES["fichier"]["name"]))
        echo("<br>l'upload a réussi") ;
        else
        echo("<br>l'upload a échoué") ;
    }//FIN IF
    else
    {   echo("Vous n'avez pas choisit de fichier!!<br>") ;
        echo("<a href=\"./PageForm.html\">Retour</a>") ;
    }//fin else
}
?>
```

Code de la page de Traitement



Résultat



L'image séléctionée sera stocké dans le dossier **upload_fichier**

FIGURE 5.16 – Le code PHP nécessaire pour récupérer un fichier

Explication :

1. On teste si la variable **$_FILES['monfichier']** n'est pas vide avec la fonction **!empty()** et on vérifie en même temps s'il n'y a pas d'erreur d'envoi.
2. Puis, on vérifie l'extension du fichier à l'aide de la fonction **pathinfo** qui renvoie un array contenant entre autres l'extension du fichier dans **$infos-fichier['extension']**. On la stocke dans une variable **$extension_upload**. Une fois l'extension récupérée, on peut la comparer à un tableau d'extensions autorisées (un array) et vérifier si l'extension récupérée fait bien partie des extensions autorisées à l'aide de la fonction **in_array()**.
3. On valide l'upload du fichier à l'aide de la fonction **move_uploaded_file()** qui va envoyer le fichier sélectionné sur votre serveur , elle prend deux paramètres : Le nom temporaire du fichier (on l'obtient avec **$_FILES['fichier']['tmp_name']**) et le chemin sous lequel sera stocké le fichier de façon définitive.
La figure (5.16) le résultat obtenu.
Pour afficher les détails du fichier on applique le code présenté dans la figure

```
echo 'Voilà les données de votre image: <br>\n nom => <b>'.$_FILES['fichier']['name'].
     '</b><br>\n taille => <b>'.$_FILES['fichier']['size'].
     '</b> octets<br>\n nom sur le serveur => <b>'.$_FILES['fichier']['tmp_name'].
     '</b><br>\n type de l\'image => <b>'.$_FILES['fichier']['type'];
```

code PHP

Salut!, je sais votre pseudo tu est: Naouel de Constantine
Vous etes madame
Votre ville est en algerie
Et vous avez comme plateforme windows

l'upload est réussiVoilà les données de votre image:
\n nom => **1.jpg**
\n taille => **12883** octets
\n nom sur le serveur => **C:\wamp64\tmp\phpD36D.tmp**
\n type de l'image => **image/jpeg**

Résultat

FIGURE 5.17 – Le code PHP nécessaire pour récupérer un fichier

5.5 Conclusion

Dans ce chapitre, nous avons présenté : les tableaux qui sont essentiels pour le stockage, la gestion et les opérations sur des jeux de variables. Les formulaires pour permetre à l'utilisateur de renseigner des donnée , et enfin nous avons vu les fichiers de leurs créations par une page Php à leurs récupération depuis l'ordinateur de l'utilisateur.

Conclusion générale

Dans ce cours nous avons comencé par une première parite qui présente les balises HTML les plus utilisées pour construire un contenu web, suivi par une étude sur le CSS pour donner un style aux pages web HTML. Dans la deuxième partie de ce cous, nous avons appris à manipuler les éléments de syntaxe de base du PHP et notamment des variables, des fonctions, des boucles et des conditions. Nous avons également découvert les tableaux , les fichiers et les formulaires du PHP qui rend ce langage d'autant plus puissant et permet une communication dynamique avec l'utilisateur.

5.6 Perspective

— Étudier la partie orientée objet du PHP et comprendre les concepts d'objet, de classe et d'instance.

— Utiliser l'ensemble de ces notions PHP pour créer et se connecter à des bases de données MySQL. Les bases de données MySQL nous permettent de stocker de grandes quantités de données.

— Étudier les différentes étape de la phase d'hébergement d'un site web.

Bibliographie

[1] *Documentation officielle : https ://www.w3.org/2010/04/xhtml10-strict.html*. 2021.

[2] *Documentation officielle : https ://www.w3.org/Consortium/Translation/*. 2021.

[3] *Documentation officielle : https ://www.w3.org/standards/webdesign/htmlcss*. 2021.

[4] *Documentation officielle : PHP :https ://www.php.net/manual/fr/*. 2021.