

# SUPPORT DE COURS DE DEVELOPPEMENT WEB

**Fait par Khadim NDAO**

## ➤ MAÎTRISER LES FONDAMENTAUX

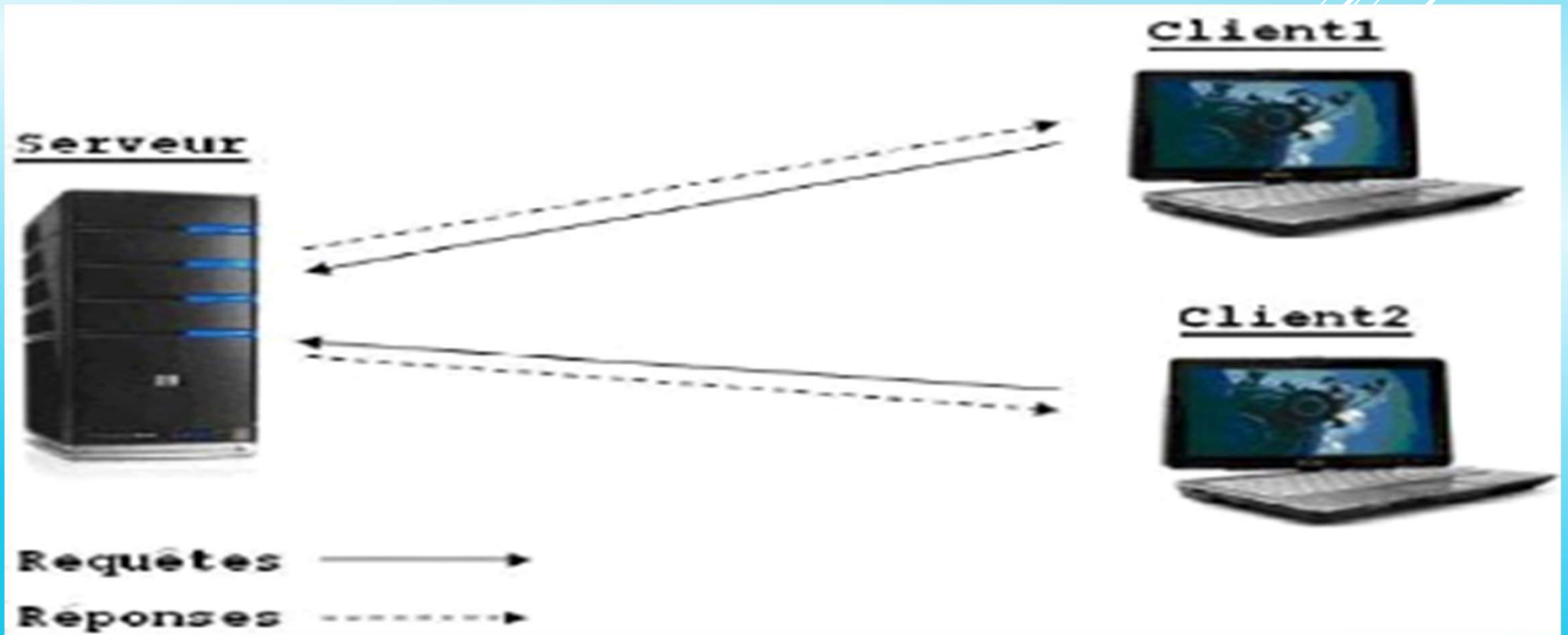
- ❖ Le web
- ❖ Démystifier l'environnement technique (client-serveur)
- ❖ Langages & outils internet (HTML, CSS, JS, PHP, CMS...)

# LE WEB

Le web ou WWW (pour *World Wide Web* ou *Toile mondiale* en français), permet de consulter via un navigateur des pages regroupées sur des sites via le réseau **internet**.

L'image de la toile d'araignée découle des liens hypertextes qui relient les pages entre elles et qui peuvent donc faire penser à une toile d'araignée.

# ENVIRONNEMENT CLIENT-SERVEUR)



# ENVIRONNEMENT CLIENT-SERVEUR)

L'environnement **client–serveur** désigne un mode de communication à travers un réseau entre plusieurs programmes : l'un, qualifié de client, envoie des requêtes ; l'autre ou les autres, qualifiés de serveurs, attendent les requêtes des clients et y répondent.

- Un serveur est un programme qui offre un service sur le réseau. Le serveur accepte des requêtes, les traite et renvoie le résultat au demandeur. | Une requête est un appel de fonction, la réponse éventuelle pouvant être synchrone ou asynchrone (le client peut émettre d'autres requêtes sans attendre) | Les arguments et les réponses sont énoncés dans un protocole
- Un logiciel client est un programme qui utilise le service offert par un serveur. Le client envoie une requête et reçoit la réponse.

# ENVIRONNEMENT CLIENT-SERVEUR)

Liens Utiles

[Formation comprendre le web](#)

# LES LANGAGES ET OUTILS

**Les pages Web statiques** offrent une information fixe et permanente. Le visiteur ne peut pas interagir avec le contenu de la page (commentaires, formulaire de contact et autres). Ce type de page Web (bien souvent en **HTML et CSS**) est majoritairement utilisé par des entreprises qui souhaitent établir un site « vitrine » divulguant les informations suivantes : « Qui sommes-nous ? », « Services », « Nous contacter » etc.

**Les pages Web dynamiques** offrent un contenu « en mouvement ». L'utilisateur peut interagir avec le contenu de la page. Ces pages Web peuvent être des blogs, des forums ou bien encore des sites de vente en ligne de produits ou de service. Le développement de ces pages Web est plus compliqué car elles requièrent la connaissance de langages de programmation plus avancés que le HTML et le CSS, comme le langage **PHP** pour ne citer que celui-ci. Ces pages Web requièrent la création et la gestion **d'une base de données** car l'information de ces sites doit être actualisée en permanence.

## LES LANGAGES ET OUTILS

**Une Base de données permet de stocker les informations de notre site. Elle est composée de Table. Ces dernières sont composées de ligne et de colonnes. Les colonnes sont appelées "champ" et les lignes "enregistrements« . Pour chaque table on doit avoir un champ identifiant qui permet de rendre unique chaque enregistrement. Comme base de données on peut citer MySQL, Oracle, PostgreSQL ...**



# LES LANGAGES ET OUTILS

CMS, **Content Management System**, en français **Système de Gestion de Contenu** est un terme vague, puisqu'en fait tout logiciel gère un contenu. Le gestionnaire de contenu se charge de la partie commune aux pages de votre site et génère les pages à partir du texte ou des données que vous lui fournissez. Il vous délivre donc de tâches répétitives. Il peut être statique et créer les pages avant qu'elles ne soient mises en lignes, ou dynamique et créer la page à la demande du visiteur.

Il n'apporte rien quand au contenu lui-même, mais permet de gérer la structure du site, ajouter et classer les pages... Les principaux CMS permettent de gérer un forum, un annuaire, les nouvelles, la plupart du temps par l'ajout d'extensions.

Quelquefois il permet d'adapter l'interface du site, de générer des pages imprimables plus dépouillées ou de faire participer des contributeurs au site.

On a plusieurs CMS dont les plus populaires sont : Wordpress (qu'on va utiliser dans notre cours), Drupal, Joomla, PrestaShop pour ne citer que cela.

# HTML ET CSS

## I. HTML

- ✓ **Structure d'une page web**
- ✓ **Les balises de mise en page de texte**
- ✓ **Les balises de médias**
- ✓ **Les tableaux**
- ✓ **Les formulaires**
- ✓ **Les balises sémantiques**

# HTML ET CSS

## Structure d'une page web

```
<!doctype html>
<html lang="fr">
  <head>
    <meta charset="utf-8">
    <title>Titre de la page</title>
    <link rel="stylesheet" href="style.css">
    <script src="script.js"></script>
  </head>
  <body>
    ... <!-- Le reste du contenu --> ...
  </body>
</html>
```

# HTML ET CSS

## Les balises de mise en page de texte

➤ Les titres :

En HTML on a 6 niveau de titre allant de h1 à h6 du plus au moins important

Syntaxe :

```
<h1> Titre </h1>
```

Exemple:

```
<h1>Mon super Titre</h1>
```

```
<h2>Mon super Titre</h2>
```

# HTML ET CSS

## Les balises de mise en page de texte

### ➤ Les paragraphes

En HTML on a 2 manière de gérer les paragraphes avec la balise <p></p> et la balise <pre></pre>.

La balise p est aider par la la balise <br> pour aller à la ligne.

Quant a la balise pre il affiche textuellement son contenue

# HTML ET CSS

## Les balises de mise en page de texte

### ➤ La mise en valeur de texte

- les balises `<b></b>` et `<strong></strong>` pour mettre en gras
- La balise `<mark></mark>` pour marquer (surligner)
- La balise `<em></em>`, `<i></i>` pour mettre en italique
- La balise `<u></u>` pour souligner

**NB : Tout cela se fait avec du CSS, mais c'est toujours bon de le savoir quand même.**

# HTML ET CSS

## Les balises de mise en page de texte

### ➤ Les listes

Elles nous permet de structurer nos informations dans une page.

On a 2 types de listes :

- Les listes ordonnées ou numérotées : `<ol></ol>`
- Les listes non-ordonnées ou listes à puces : `<ul></ul>`

Les listes sont composées d'éléments qui sont dans la balise : `<li></li>`

**NB:** Selon les informations à structures on peut avoir des listes dans les éléments `<li>`, c'est ce qu'on appelle l'imbrication de liste.

# HTML ET CSS

## Les balises de mise en page de texte

### ➤ Les liens

Ils nous permettent de naviguer d'une page à une autre. Ils sont écrits d'une façon différente (par défaut, en bleu et soulignés). Pour créer un lien on utilise la balise `<a></a>` avec l'attribut `href` qui permet d'indiquer le chemin de la page vers laquelle le lien doit rediriger. On a plusieurs types de liens :

- Lien vers un autre site
- Lien vers une autre page de notre site
- Lien vers une ancre
- Lien de téléchargement



# HTML ET CSS

## Les balises de mise en page de texte

### ➤ Les liens

#### Lien vers un autre site

Pour faire un lien vers un autre site, il faut copier son adresse (on parle d'URL) en `http://`. Notez que certains liens commencent parfois par `https://` (sites sécurisés). Dans ce cas de figure on parle de lien absolue, car on indique l'adresse complète.

Exemple : `<a href="https://www.google.com"></a>` vas rediriger vers le site **Google.com**

# HTML ET CSS

## Les balises de mise en page de texte

### ➤ Les liens

**Lien vers une autre page de notre site**

**Pour faire un lien vers une page de notre site on a 3 niveau**

- **Soit la page est dans le même dossier que la page appelant**
- **Soit la page est dans un sous dossier**
- **Soit la page est dans un dossier de niveau supérieur au dossier courant**

# HTML ET CSS

## Les balises de mise en page de texte

### ➤ Les liens

**Lien vers un autre page de notre site**

**Pour faire un lien vers une page de notre site on a 3 niveau**

**Soit la page est dans le même dossier que la page appelant**

**Exemple : `<a href='page.html'>lien vers la page</a>`**

# HTML ET CSS

## Les balises de mise en page de texte

### ➤ Les liens

**Lien vers un autre page de notre site**

**Pour faire un lien vers une page de notre site on a 3 niveau**

**Soit la page est dans un sous dossier du dossier contenant la page appelante**

**Exemple : `<a href='nomsousdossier/page.html'>lien vers la page dans le sous dossier</a>`**

# HTML ET CSS

## Les balises de mise en page de texte

### ➤ Les liens

**Lien vers un autre page de notre site**

**Pour faire un lien vers une page de notre site on a 3 niveau**

**Soit la page est dans un dossier de niveau supérieur au dossier courant**

**Exemple : `<a href=“../nomdossier/page.html”>lien vers la page dans le dossier</a>`**

# HTML ET CSS

## Les balises de média

- **Les images**
- **Les vidéos**
- **Les audios**

# HTML ET CSS

## Les balises de média

### ➤ Les images

Syntaxe :

```

```

L'attribut `src` permet d'indiquer le chemin du dossier où se trouve de l'image.

Et l'attribut `alt` permet d'afficher un texte en alternative si l'image ne s'affiche pas.

# HTML ET CSS

## Les balises de média

### ➤ Les vidéos

Syntaxe :

```
<video src="source_de_limage">
```

L'attribut `src` permet d'indiquer le chemin du dossier où se trouve la vidéo.



# HTML ET CSS

## Les balises de média

### ➤ Les vidéos

Syntaxe :

```
<video src="source_de_limage">
```

L'attribut `src` permet d'indiquer le chemin du dossier où se trouve de la vidéo.

# HTML ET CSS

## Les balises de média

### ➤ Les audios

Syntaxe :

```
<audio src="source_de_limage">
```

L'attribut `src` permet d'indiquer le chemin du dossier où se trouve de l'audio.

# HTML ET CSS

## Les tableaux

**Les tableaux nous permettent de représenter nos informations de manière structurée. Pour les tableaux on a 4 balises**

**La balise `<table></table>`**

**La balise `<tr></tr>`**

**La balise `<td></td>`**

**La balise `<caption></caption>`**

**La balise `<th></th>`**

# HTML ET CSS

## Les tableaux

**La balise `<table></table>`**

**C'est dans cette balise qui permet de définir un tableau. Il peut avoir les attribut suivants :**

- **Border** : qui permet de définir les bordure des cellules du tableau (par défaut les tableau en html n'ont pas de bordure.
- **Cellpadding** : permet de définir la distance entre la marge et le contenu d'une cellule.
- **Cellspacing** : permet de définir la distance entre les

# HTML ET CSS

## Les tableaux

**La balise `<tr></tr>`**

**Cette balise permet de définir une ligne en dans un tableau. Chaque ligne est composé de plusieurs cellules.**

# HTML ET CSS

## Les tableaux

**La balise `<td></td>`**

**Cette balise permet de définir une cellule,  
il est toujours inclus dans un tr.**

# HTML ET CSS

## Les tableaux

**La balise `<th></th>`**

**C'est comme une balise `td` mais c'est en gras et center dans la cellule qui le contient. Elle permet de définir un entête de tableau.**

# HTML ET CSS

## Les tableaux

**La balise `<caption></caption>`**

**Elle permet de définir le titre d'un tableau.**

**Syntaxe:**



# HTML ET CSS

## Les tableaux

Syntaxe :

```
<table>  
  <caption>titre</titre>  
  <tr>  
    <th>Entete</th>  
  </tr>  
  <tr>  
    <td>contenue</td>  
  </tr>  
</table>
```

# HTML ET CSS

## Les Formulaires

Pour la manipulation des formulaires on utilise en général les balises :

`<fieldset><legend></legend></fieldset>`

`<label></label>`

`<input>`

`<select></select>`

`<textarea></textarea>`

`<label></label>`

# HTML ET CSS

## Les Formulaires

La balise `fieldset` permet de regrouper des informations d'une formulaire et de l'associer à une légende avec la balise `legend`.

**Syntaxe:**

```
<fieldset>
```

```
  <legend>Groupe d'information</legend>
```

```
  <!-- ensemble de input et de label -->
```

```
</fieldset>
```

# HTML ET CSS

## Les Formulaires

La balise `label` permet d'indiquer à l'utilisateur l'information qu'il doit saisir.

**Syntaxe:**

```
<label>votre information </label>
```

**Exemple:**

```
<label>donez votre nom</label>
```

# HTML ET CSS

## Les Formulaires

**La balise input (champ) permet à l'utilisateur d'entrée des information. Ces derniers peuvent être de plusieurs types : text, number, password, tel, email, date, checkbox, radio, submit, reset**

# HTML ET CSS

## Les Formulaires

- **Text** : permet à l'utilisateur de donner des informations de type text.

**Syntaxe :**

```
<input type="text" name="nom">
```

**Exemple :**

```
<input type="text" name="nom" id="nom">
```

# HTML ET CSS

## Les Formulaires

- **Password** : permet à l'utilisateur de donner des mots de passe, c'est information est caché par des pointillés (c'est-à-dire qu'on écrit pas les informations en clair)

**Syntaxe :**

```
<input type="password" name="pass">
```

**Exemple :**

```
<input type="password" name="pass" id="pass">
```

# HTML ET CSS

## Les Formulaires

- **Number** : permet à l'utilisateur de donner les nombres, dans un champ de type number on ne peut saisir que des caractères numériques.

**Syntaxe :**

```
<input type="number" name="nom">
```

**Exemple :**

```
<input type="number" name="age" id="age">
```



# HTML ET CSS

## Les Formulaires

- **Email** : permet à l'utilisateur de donner un email, dans ce type de champ l'utilisateur est tenu de mettre un @ dans le champ pour valider la saisie.

**Syntaxe :**

```
<input type="email" name="nom">
```

**Exemple :**

```
<input type="email" name="email" id="email">
```

# HTML ET CSS

## Les Formulaires

- **Tel** : permet à l'utilisateur de donner un numéro de téléphone. On utilise l'attribut **pattern** pour définir le format du numéro de téléphone

**Syntaxe :**

```
<input type="tel" name="nom">
```

**Exemple :**

```
<input type="tel" name="tel" id="tel" pattern="">
```

# HTML ET CSS

## Les Formulaires

- **Date : permet à l'utilisateur de donner une date.**

**Syntaxe :**

```
<input type="date" name="nom">
```

**Exemple :**

```
<input type="date" name="date" id="date">
```

# HTML ET CSS

## Les Formulaires

- **Radio** : permet à l'utilisateur de faire un choix en sélectionnant un champs.

**Syntaxe :**

```
<input type="radio" name="nom">
```

**Exemple :**

```
<input type="radio" name="genre" id="genre" value="m">M
```

# HTML ET CSS

## Les Formulaires

- **Radio** : permet à l'utilisateur de faire plusieurs choix en cochant des cases.

**Syntaxe :**

```
<input type="checkbox" name="nom">
```

**Exemple :**

```
<input type="checkbox" name="sport"  
value="foot">Football
```

```
<input type="checkbox" name="sport"
```

# HTML ET CSS

## Les Formulaires

- **Submit : permet à l'utilisateur de valider un formulaire.**

### **Syntaxe :**

```
<input type="submit">
```

Exemple :

```
<input type="submit" value="enregistrer">
```

# HTML ET CSS

## Les Formulaires

- **Reset** : permet à l'utilisateur de réinitialiser un formulaire.

### **Syntaxe :**

```
<input type="reset">
```

Exemple :

```
<input type="reset">
```

# HTML ET CSS

## Les Formulaires

**La balise `<select></select>` est un type de champ qui permet à l'utilisateur de choisir parmi une liste d'option. Il est accompagné de la balise `option` qui gère les éléments de la liste déroulante.**

### **Syntaxe:**

```
<select name="nom">  
  <option value="valeur1">valeur 1</option>  
  <option value="valeur2">valeur 2</option>  
</select>
```



# HTML ET CSS

## Les Formulaires

La balise `<textarea></texarea>` est un type de champ qui permet à l'utilisateur d'éditer du texte sur plusieurs lignes. Les attributs `cols` et `rows` permettent de définir la taille du `textarea`

### Syntaxe:

```
<textarea name="nom" cols="_" rows="_">
```

```
</textarea>
```

# HTML ET CSS

## Les balises sémantiques

Les balises sémantiques HTML5 ont un rôle précis à jouer et nous disent quel type de contenu elles contiennent. Parmi les balises sémantiques que nous utilisons depuis longtemps, il y a `<head>` et `<body>`.

Un navigateur sait avec certitude que tout ce qui se trouve dans la `<head>` concerne les métadonnées de la page et que tout ce qui est dans `<body>` concerne la partie visible de la page qui sera montrée à l'utilisateur.

# HTML ET CSS

## Les balises sémantiques

Les balises que nous allons utiliser spécifiquement dans ce cours seront `<header>`, `<footer>`, `<main>`, `<article>`, `<aside>`, `<span>`, `<div>`, `<section>` et `<nav>`.

Ces balises permettent montrer à l'algorithme des moteurs de recherche où se trouve le contenu quel type de contenu dans la page. Il permet aussi une meilleure lisibilité du code du point de vue développement.

# HTML ET CSS

## II. CSS

- ✓ **Ou insérer le CSS**
- ✓ **Les sélecteurs**
- ✓ **Mise en forme du texte**
- ✓ **L'arrière plan (couleur, images)**
- ✓ **Les blocs**
- ✓ **Positionnement**
- ✓ **Les listes**
- ✓ **Le Framework Bootstrap**

# HTML ET CSS

## CSS

**CSS signifie en anglais « Cascading Style Sheets » et se traduit « feuilles de style en cascade » en français.**

**On utilise le langage CSS pour décrire la présentation d'un document écrit en HTML.**

**Le CSS décrit les couleurs, la police, les couleurs et images de fond, le remplissage, les marges, les bordures, les emplacements des éléments etc;**

**Il permet de rendre une page plus agréable à voir et d'avoir une meilleur présentation.**

# HTML ET CSS

## **Où placer le CSS**

**On peut placer le code CSS dans 3 niveaux sur notre site :**

- 1) Dans une feuille de style externe**
- 2) Au niveau de notre balise head**
- 3) Intégré dans une balise**

# HTML ET CSS

## Ou placer le CSS

### Dans une feuille de style externe

**C'est le meilleur endroit ou on peut placer notre code css, il va nous permettre d'utiliser la feuille de style dans d'autres pages. Pour lier notre page au fichier de style on utilise la balise <link> dans le head**

### Syntaxe:

```
<head>  
  <link rel="stylesheet" type="text/css" href="ma_feuille_de_style.css" />  
</head>
```

# HTML ET CSS

## Où placer le CSS

Au niveau de la balise head

On utilise la balise `<style></style>` avec l'attribut `type="text/css"` pour dire qu'on va écrire du css.

**Syntaxe:**

```
<head>
<style type="text/css">
  .selecteur {
    /* règle css */
  }
</style>
</head>
```



# HTML ET CSS

## Où placer le CSS

### Directement dans une balise

On utilise l'attribut style pour écrire dedans des règles css, chaque instructions css est séparé de l'autre par un ;

### Syntaxe:

```
<balise style="regle_css1;regle_css2;">
```

```
</balise>
```

# HTML ET CSS

## Les sélecteurs

**Les sélecteurs permettent de cibler l'élément que vous voulez mettre en forme.**

**Pour sélectionner un élément dans notre code html on peut utiliser les balises, ou bien les attributs class et id (ou sélecteurs universelles)**

# HTML ET CSS

## Les sélecteurs

### Syntaxe:

#### Cas d'une balise

```
balise {  
    /* règles css*/  
}
```

#### Cas d'un attribut class

```
.class {  
    /* règles css*/  
}
```

#### Cas d'un attribut id

```
#id {  
    /* règle css*/  
}
```

# HTML ET CSS

## Mise en forme du texte

- ✓ **Police des caractères (font-family)**
- ✓ **Taille du texte (font-size)**
- ✓ **Couleur du texte (color)**
- ✓ **Texte en gras (font-weight)**
- ✓ **Texte en italic (font-style avec les propriété oblique, italic et normal)**
- ✓ **La casse (text-transform)**
- ✓ **Soulignement du texte (text-decoration)**
- ✓ **Hauteur de ligne (line-height)**

# HTML ET CSS

## Mise en forme du texte

### ✓ Soulignement du texte (`text-decoration`)

La propriété `text-decoration` peut prendre les valeurs : `underline`, `overline`, `line-through`, `none`

On peut aussi changer la couleur et le style du soulignement avec les propriétés :

- `Text-decoration-color` : couleur
- `Text-decoration-style` : avec les valeurs `dashed`, `dotted`, `solid`, `double`, `all`.

Exemple :

```
text-decoration: underline;  
text-decoration-color: aqua;  
text-decoration-style: double;
```

# HTML ET CSS

## Mise en forme du texte

- ✓ **Texte en gras (font-weight)**

La propriété **font-weight** peut prendre les valeurs: **bold**, **bolder**, **light**, **lighter** ou des valeur numérique allant de **100** à **900**.

**Exemple :**

**Font-weight : bold;**

# HTML ET CSS

## Mise en forme du texte

- ✓ **La casse du texte (text-transform)**

**La propriété text-transform peut prendre les valeurs : uppercase, lowercase, inherit, capitalize, none.**

**Exemple :**

**Text-transform : capitalize;**

# HTML ET CSS

## Mise en forme du texte

- ✓ Texte en italic (font-style)

La propriété font-style peut prendre les valeurs: italic, oblique ou normal.

Exemple :

Font-style : italic;



# HTML ET CSS

## Mise en forme du texte

### ✓ **Soulignement du texte (text-decoration)**

#### **valeurs possibles :**

underline (souligné)

overline (ligne au dessus)

line-through (barré)

blink (clignotant)

none (aucun)

# HTML ET CSS

## Mise en forme du texte

- ✓ **La casse (text-transform)**
  - **capitalize**, première lettre du texte en majuscule,
  - **uppercase**, texte en majuscule
  - **lowercase**, texte en minuscule,
  - **none**, texte normal.
  - **inherit** hérite de son parent

# HTML ET CSS

## **Couleur et image d'arrière plan**

- ✓ **Couleur d'arrière plan (background-color)**
- ✓ **Les images d'arrière plan (background-image)**

# HTML ET CSS

## Couleur et image d'arrière plan

### Les autres style lier aux images d'arrière plan

- ✓ **Background-repeat** (repeat, no-repeat, repeat-x, repeat-y, inhérit)
- ✓ **Background-attachement** (fixed, scroll, local, inhérit) css3
- ✓ **Background-position** (left, center, right, ou des valeurs en px)
- ✓ **Background-size** (cover, 0%)

# HTML ET CSS

## Couleur et image d'arrière plan

Les autres style lier aux images d'arrière plan

**Redimensionner l'image (background-size:cover).**

**Ceci redimensionne l'image de sorte que la zone d'arrière-plan soit complètement recouverte par l'image d'arrière-plan tout en conservant les proportions. La zone entière sera couverte. Cependant, une partie de l'image peut ne pas être visible si la largeur/hauteur de l'image redimensionnée est trop grande**

# HTML ET CSS

## Couleur et image d'arrière plan

### Exemple :

```
.image-de-fond {  
    background-color: #FFC;  
    background-image: url(image.png);  
    background-repeat: no-repeat;  
    background-position: right center;  
}
```

# HTML ET CSS

## Les blocs

- ✓ Espace entre les mots (**word-spacing**)
- ✓ Espace entre les lettres (**letter-spacing**)
- ✓ Alignement du texte (**text-align**)
- ✓ Indentation du texte (**text-indent**)
- ✓ L'affichage d'un élément (**display**)

# HTML ET CSS

## Les blocs

**L'affichage d'un élément (display)**

**La propriété de feuille de style display css peut prendre la valeur de :**

- none, le bloc ne sera pas affiché.**
- inline, le bloc sera considéré comme étant un seule ligne**
- block, spécifie un bloc.**
- inline-block correspond à un élément inline avec un comportement bloc**



# HTML ET CSS

## Les blocs

### L'affichage d'un élément (display)

#### Exemple :

```
.bloc-inline{  
  display:inline;  
  background-color: #FCF  
}
```

# HTML ET CSS

## Les boîtes

- ✓ La largeur d'un élément (**width**)
- ✓ La hauteur d'un élément (**height**)
- ✓ Flottement de la boîte (**float**)
- ✓ L'affichage de l'élément flottant par rapport à l'élément précédent (**clear**)
- ✓ L'écart à l'intérieur de la boîte (**padding**)
- ✓ L'écart à l'extérieur de la boîte (**margin**)
- ✓ Les bordures (**border**)

# HTML ET CSS

## Les boites

**On distingue deux principaux types de balises en HTML :**

**Le type block (<p>, <h1>...) :** ces balises créent un retour à la ligne et occupent par défaut toute la largeur disponible. Elles se suivent de haut en bas.

**Le type inline (<a>, <strong>...) :** ces balises délimitent du texte au milieu d'une ligne. Elles se suivent de gauche à droite.

**On peut modifier la taille d'une balise de type block avec les propriétés CSS width (largeur) et height (hauteur).**

# HTML ET CSS

## Les boites

- ✓ **L'affichage de l'élément flottant par rapport à l'élément précédent (clear)**

### Valeurs possibles :

- **left (l'élément sera affiché au dessous de l'élément précédent si ce dernier est à float:left)**
- **right (l'élément sera affiché au dessous de l'élément précédent si ce dernier est à float:right)**
- **both (l'élément sera affiché au dessous de l'élément précédent si ce dernier est à float:left ou float:right)**
- **none (l'élément sera affiché à côté de l'élément précédent.)**

# HTML ET CSS

## Les boites

### ✓ Flottement de la boite (float)

On utilise la propriété css float pour gérer des colonnes en CSS.

Valeurs possibles :

- left (aligné sur la gauche)
- right (aligné sur la droite)
- none ( l'élément ne flotte pas

# HTML ET CSS

## Les boites

**On peut définir des minima et maxima autorisés pour la largeur et la hauteur : min-width, max-width, min-height, max-height. Les éléments de la page disposent chacun de marges intérieures (padding) et extérieures (margin). S'il y a trop de texte à l'intérieur d'un bloc de dimensions xes, il y a un risque de débordement. Dans ce cas, il peut être judicieux de rajouter des barres de délement avec la propriété overflow ou de forcer la césure avec word-wrap**

# HTML ET CSS

## Les boîtes

### En résumé

La mise en page d'un site web s'effectue en CSS. Plusieurs techniques sont à notre disposition. Le positionnement flottant (avec la propriété `float`) est l'un des plus utilisés à l'heure actuelle. Il permet par exemple de placer un menu à gauche ou à droite de la page. Néanmoins, cette propriété n'a pas été initialement conçue pour cela et il est préférable, si possible, d'éviter cette technique. Le positionnement `inline-block` consiste à affecter un type `inline-block` à nos éléments grâce à la propriété `display`. Ils se comporteront comme des inlines (placement de gauche à droite) mais pourront être redimensionnés comme des blocs (avec `width` et `height`). Cette technique est à préférer au positionnement flottant. Le positionnement absolu permet de placer un élément où l'on souhaite sur la page, au pixel près. Le positionnement `fixed` est identique au positionnement absolu mais l'élément restera toujours visible même si on descend plus bas dans la page. Le positionnement relatif permet de décaler un bloc par rapport à sa position normale. Un élément A positionné en absolu à l'intérieur d'un autre élément B (lui-même positionné en absolu, `fixed` ou relatif) se positionnera par rapport à l'élément B, et non par rapport au coin en haut à gauche de la page.