



# Développement Web

---

**Pr. Hajar LAZAR**

**[hajar.lazar@ucd.ac.ma](mailto:hajar.lazar@ucd.ac.ma)**

# Plan

---

- ❑ Introduction & langage HTML5
- ❑ Langage CSS3 et Bootstrap
- ❑ Langage Javascript & JQuery
- ❑ Langage PHP
- ❑ Framework Laravel
- ❑ Responsive design
- ❑ NodeJS

# Chapitre1

---

## Introduction & langage HTML5



# Introduction

---

- C'est quoi l'internet et le web?
- Comment le Web fonctionne ?
- Comment créer des sites web ?

# Internet

---

## Définition (Internet)

**Internet** est un ensemble de réseaux interconnectés, utilisant un ensemble de protocoles de communication et d'échanges de données standardisés.

## Définition (World Wide Web W3C)

Le **World Wide Web** créé en 1994 par Tim Berners-Lee.

Un organisme de normalisation qui développe des standards pour le Web.

## Définition WHATWG (Web Hypertext Application Technology Working Group)

est un groupe de travail fondé en **2004** par **Apple, Mozilla et Opera** (rejoins plus tard par Google) pour développer et maintenir les standards du web, notamment le **HTML** et les technologies associées.

# Web

---

## Le Web (Tim Berners-Lee, 1989)

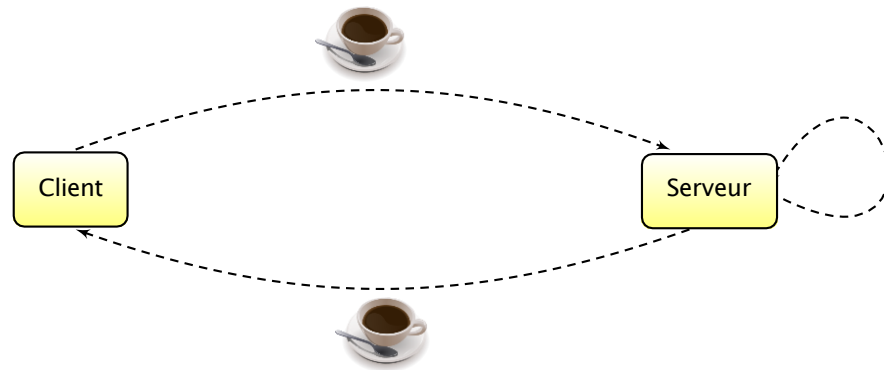
- repose sur internet
- système d'information réparti en « pages web » = documents web
- basé sur la notion **d'hypertexte** et la notion **d'hyperliens** permettant de naviguer entre les documents web
- des protocoles de communication HTTP, HTTPS
- des adresses pour nommer les documents : URL
- des langages pour créer les documents : HTML, CSS, javascript
- des navigateurs qui interprètent les documents

# Architecture Client et Serveur

Client : « Bonjour, je voudrais un café. »

(préparation du café)

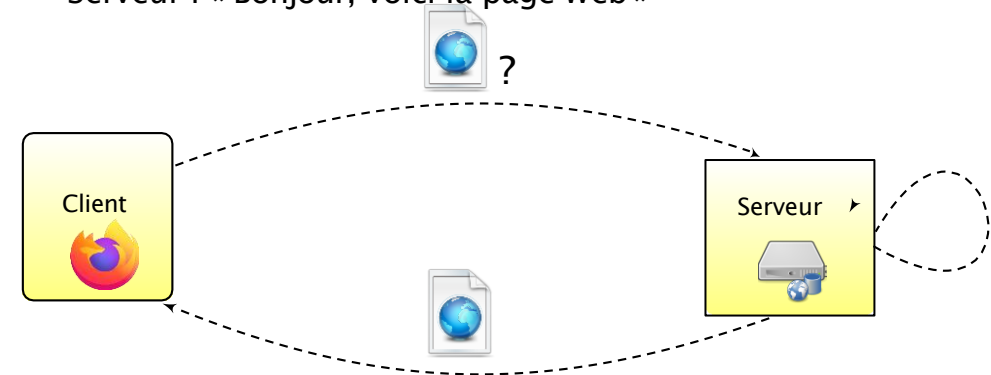
Serveur : « Bonjour, voici votre café. »



Client : « Bonjour, je voudrais <https://www.google.com>

» (préparation de la page Web)

Serveur : « Bonjour, voici la page Web »



- **Le client** envoie une requête et reçoit la réponse.
- Le **serveur** accepte des requêtes, les traite et renvoie le résultat au demandeur.
- Le terme serveur s'applique à la machine sur lequel s'exécute le logiciel serveur.
- Pour pouvoir offrir ces services en permanence, le serveur doit être sur un site avec accès permanent
- Il doit s'exécuter en permanence.

# Architecture Client / Serveur

## Définition (Serveur)

Un **serveur** est un ordinateur (et/ou un programme informatique) offrant un **service** ou une **ressource** sur un réseau.

## Définition (Client)

Un **client** est un programme informatique contactant un serveur via un réseau, afin de bénéficier d'un **service** ou d'une **ressource**.

## Architecture Client et Serveur

L'architecture client serveur s'appuie sur **le serveur**, qui **envoie des données** aux machines clientes.



Le premier serveur Web (1990)



Salle de serveurs (2009)

## Clients les plus courants aujourd'hui

| Nom                         | Depuis | Licence      |
|-----------------------------|--------|--------------|
| Android (navigateur)        | 2008   | GNU GPL      |
| Chromium                    | 2008   | BSD (etc.)   |
| Google Chrome               | 2008   | propriétaire |
| Internet Explorer (et Edge) | 1995   | propriétaire |
| Mozilla Firefox             | 2002   | MPL          |
| Opera                       | 1995   | propriétaire |
| Safari                      | 2003   | propriétaire |
| Samsung Internet            | 2015   | BSD (?)      |



# Architecture Client / Serveur

---

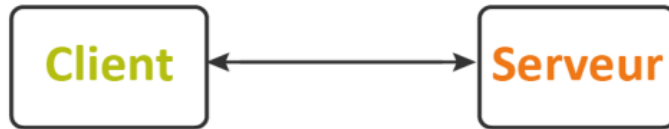
## Avantages

- **Unicité de l'information** : toutes les données sont stockées sur un même serveur
- **Meilleure sécurité** : puisque le nombre de points d'entrée permettant l'accès aux données est connu
- **Mise à jour** : mise à jour centralisé aussi bien des données et logiciels
- **Meilleure fiabilité** : En cas de panne, seul le serveur fait l'objet d'une réparation
- **Facilité d'évolution** : architecture évolutive, il est très facile de rajouter ou d'enlever des clients ou des serveurs.

## Inconvénients

- Un **coût d'exploitation élevé dû à la technicité du serveur** (bande passante, câbles, ordinateurs surpuissants)
- En **cas de panne** du serveur, aucun client n'a accès aux informations
- Si trop de clients veulent communiquer avec le serveur ce dernier risque de ne pas supporter **la charge**

# Architecture Client / Serveur



Requête/Réponse



Le serveur contacté peut  
faire appel à un service  
sur un autre serveur (ex.SGBD)

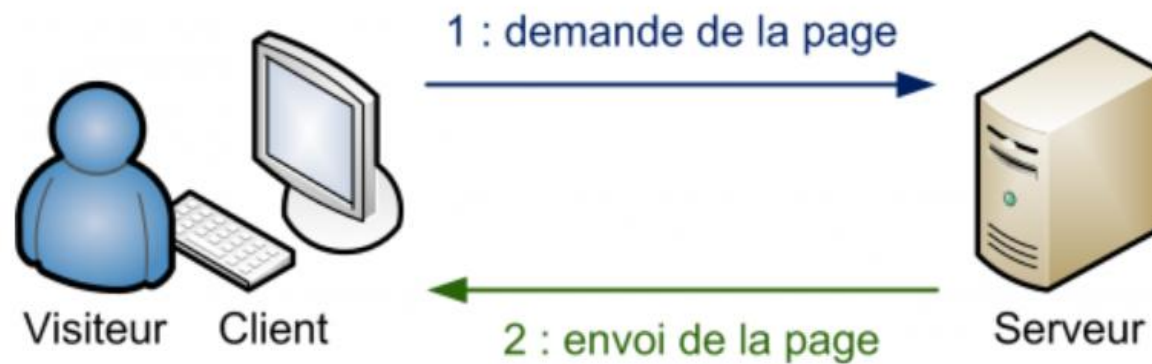
- Un **tier** correspond à un processus (programme en cours d'exécution) impliqué dans l'utilisation d'une application informatique.
- **L'architecture 2 tiers**, caractérise les systèmes clients/serveurs pour lesquels le client à un besoin et le serveur le lui fournit directement en utilisant ses propres ressources.
- **L'architecture 3 tiers** est l'extension du modèle client/serveur. Ce type d'architecture est le plus courant des architectures multi-tiers Il est également appelé client serveur de deuxième génération ou client serveur distribué

# Architecture Client et Serveur

## Site Web Statique

Un site Web statique se compose essentiellement de fichiers sources **HTML** (pour le contenu des pages) et **CSS** (pour la mise en forme).

Lorsqu'un client envoie une requête au serveur Web hébergeant un site statique, le serveur se contente de la renvoyer la ressource demandée. Les pages HTML affichées ne peuvent pas évoluer automatiquement, d'où le terme "**statique**".

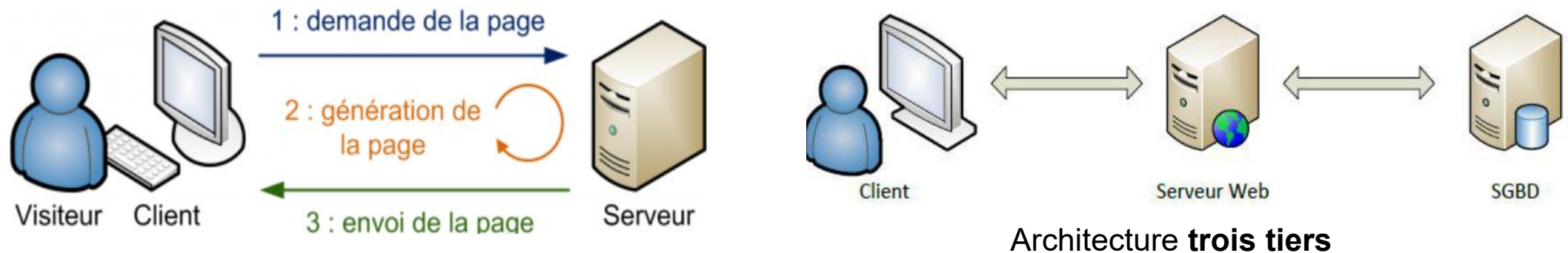


# Architecture Client et Serveur

## Site Web Dynamique

Un site **dynamique** est un type de site Web dans lequel le contenu des pages peut évoluer automatiquement.

Lorsqu'un client envoie une requête au serveur Web hébergeant un site dynamique, **le serveur** prépare la page HTML correspondant à cette requête, puis la renvoie **au client** pour affichage. Les pages Web affichées aux clients ne sont pas codées "en dur" comme dans le cas d'un site statique, mais générées au moment de leur consultation.



# Développement Back end et Front end

---

## Font end (côté client)

- Partie visible par **l'utilisateur**, c'est **l'interface** avec laquelle il interagit
- C'est la partie **du code** qui est reçue par le **client**.
- **le client** c'est notre **navigateur web**, l'outil qui va nous permettre de voir le Web " Chrome Firefox Safari Internet Explorer etc ".
- **Technologies principales** :HTML, CSS ,JavaScript , React, Angular, Vue.js, etc
- **Exemple** : Boutons, formulaires, menus, animations, affichage des données.

# Développement Back end et Front end

---

## Back end (côté serveur)

- Partie **cachée** qui gère la logique, les données et la communication entre l'**interface** et la **base de données**.
- **Technologies principales :**
  - Langages : PHP, Java, Python, Node.js, etc
  - Frameworks : Spring Boot (Java), Django (Python), Laravel (PHP), Express (Node.js)
  - Bases de données : SQL (MySQL, PostgreSQL), NoSQL (MongoDB, Redis).
- **Exemple :** Envoi d'un formulaire → traitement sur le serveur → sauvegarde dans la base de données → retour d'une réponse.

# Comment fonctionne les sites web

## Langages HTML et CSS



Traduction par l'ordinateur



## Résultat visible à l'écran



- Il s'agit de langages informatiques permettent de créer des sites web
- Les langages **HTML** et **CSS** sont à la base du fonctionnement de tous les sites web.
- L'ordinateur se base sur ce qu'on a écrit en HTML et CSS pour savoir ce qu'il doit afficher et comment.

HTML  
(pas de CSS)



HTML + CSS



# HTML 5 et CC3

**HTML5** « HyperText Markup Language »

HTML5 sert à structurer votre contenu.

Il est le fruit des travaux du W3C (World Wide Web Consortium) et du WHATWG (Web Hypertext Application Technology Working Group).

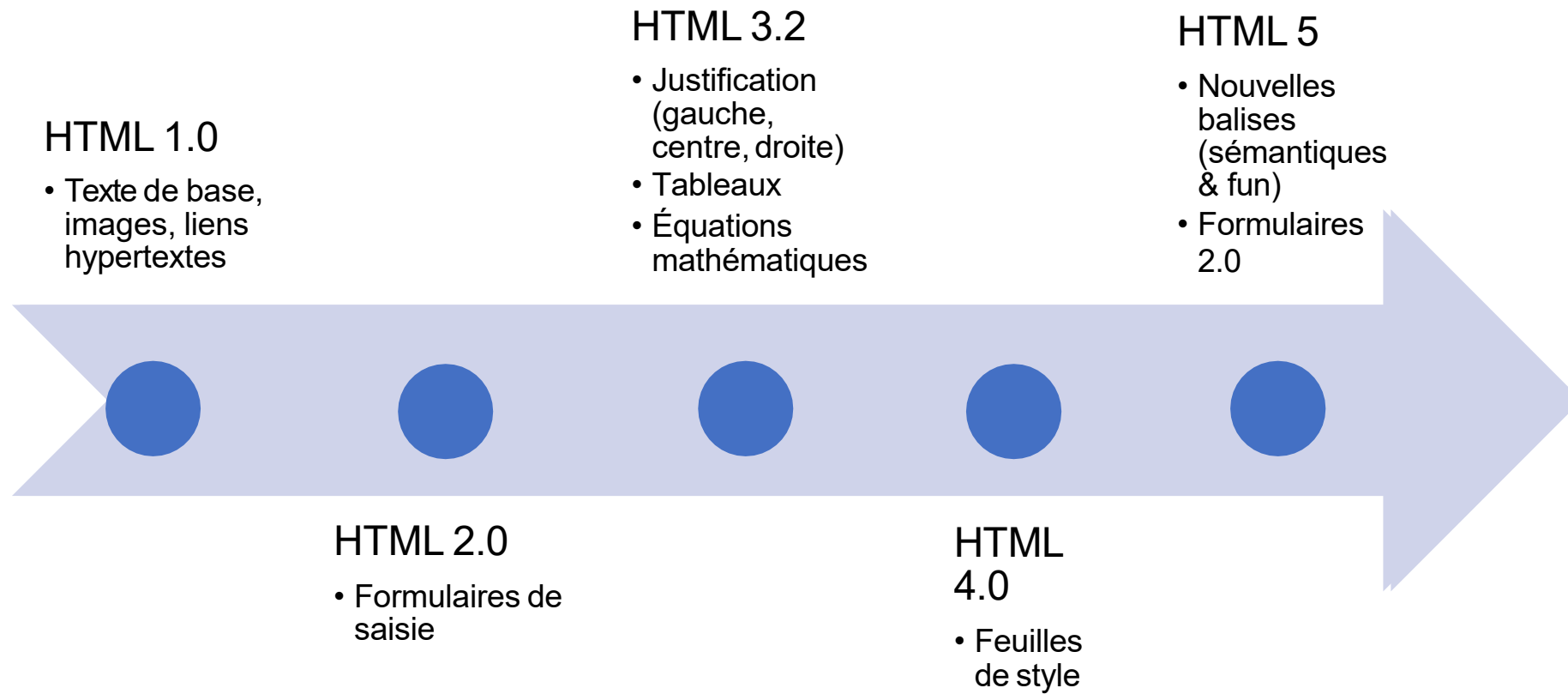
**CSS3**

Cascading Style Sheets, aussi appelées Feuilles de style) sert à gérer l'apparence de la page web (agencement, positionnement, décoration, couleurs, taille du texte...).



# HTML : Evolution

---



# HTML : Hyper Text Markup Language

---

- HTML utilise des balises (tags) pour :
  - Indiquer la façon dont le document doit être affiché
  - Spécifier la structure (organisation) d'un document
- Une Balise: <nom balise> Contenu </nom balise>
- HTML est un langage interprété par le navigateur pour la description et le formatage de la page
- L'extension (suffixe) d'un document HTML est : **.htm** ou **.html**

# Balise de formatage

---

**<balise attributs> Contenu </balise>**

**Balise** (tag en anglais):

Un mot clé du langage associant une propriété particulière au contenu de la balise

*Exemple* : mettre en gras le contenu après cette balise

**Attributs** (ils ne sont pas toujours obligatoires) :

Ils définissent les valeurs spécifiques à appliquer sur le contenu.

*Exemple* : taille 12 associée à la balise "police de caractère"

# Balise de formatage

---

## Contenu :

Il représente le contenu à mettre en forme (texte, image, ... etc.) ou d'autres balises

**</balise>** : C'est une étiquette précisant la fin de la balise

*Exemple* : arrêter de mettre en gras après cette balise

# Balise de formatage

---

**<balise attributs> Contenu </balise>**

- Les attributs définissent les valeurs spécifiques à appliquer sur le contenu.
- Les attributs se trouvent toujours dans **la balise d'ouverture** et ils ne sont pas toujours **obligatoires**.
- Les attributs sont séparés par des espaces.
- Chaque attribut a une valeur : **attribut= valeur**
- L'ordre des attributs n'a **aucune importance**

# Balise de formatage

---

Une balise possédant des attributs ressemble à ceci:

`<balise attribut1="valeur1" attribut2="valeur2" attribut3="valeur3"...>Contenu</balise>`

Exemple: `<font color="red" face="arial">Bonjour à tous</font>`

La balise ouvrante `<font>` applique la couleur rouge et la police Arial au contenu qui la suit qui n'est rien d'autre que le message "Bonjour à tous".

La balise fermante `</font>` cesse l'effet appliqué par la balise ouvrante du même nom.

# Structure d'une page web en HTML 5

```
<html>

  <head>
    <title>Titre de ma page </title>
  </head>

  <body>

    <!-- Ceci est un commentaire -->

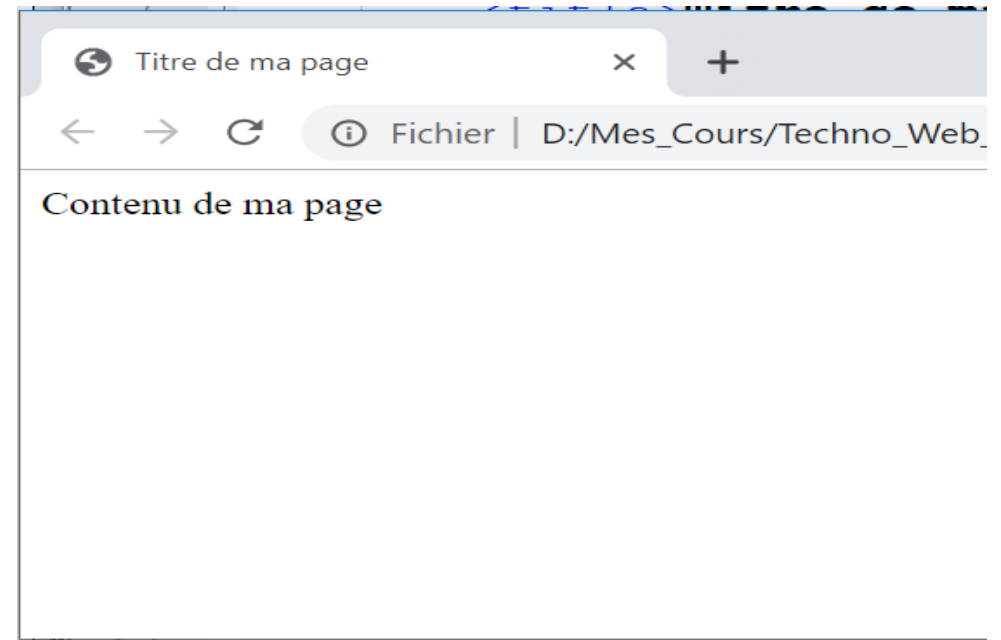
    Contenu de ma page

  </body>

</html>
```

Document HTML

Navigateur



Résultat de son interprétation par le navigateur

# Structure d'une page web en HTML 5

---

**<!DOCTYPE html>**: indique le type du document dans ce cas HTML

**<HTML> ... </HTML>** : Marqueurs de début et de fin du document HTML

**<HEAD> ... </HEAD>** : Ensemble des méta informations relatives au document

**<TITLE> Mon titre </TITLE>** : Titre du document qui apparaît sur la barre de titre du navigateur

```
<html>

  <head>
    <title>Titre de ma page </title>
  </head>

  <body>

    <!-- Ceci est un commentaire -->

    Contenu de ma page

  </body>

</html>
```



# Structure d'une page web en HTML 5

---

**<META>** : Paramètres utiles notamment pour l'indexation du document par les moteurs de recherche

**<BODY> ... </BODY>**: Corps du document

**<!-- Mon commentaire -->** : pour insérer des commentaires dans le document

```
<html>
  <head>
    <title>Titre de ma page </title>
  </head>
  <body>
    <!-- Ceci est un commentaire -->
    Contenu de ma page
  </body>
</html>
```

```
<html>
<head>
<title>Titre de ma page</title>
</head>
<body>
<!-- Ceci est un commentaire -->
Contenu de ma page
</body>
</html>
```

Version 1

```
<html>

  <head>
    <title>Titre de ma page </title>
  </head>

  <body>

    <!-- Ceci est un commentaire -->

    Contenu de ma page

  </body>

</html>
```

Version 2



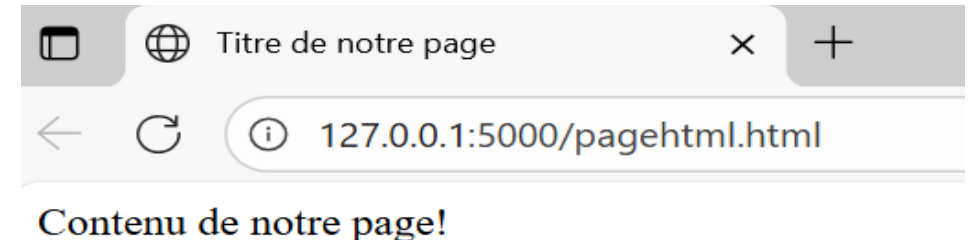
# L'indentation du code

# Le Paragraphe <HEAD> ... </HEAD>

- Contient des informations relatives au document, notamment sur les scripts, les feuilles de style, les méta-balises, des scripts, etc.
- **<TITLE>** C'est le titre du document affiché dans la barre de titre du navigateur ainsi que dans les signets et les favoris.

```
<> pagehtml.html > ...  
1  <!DOCTYPE html>  
2  <html lang="fr">  
3    <head>  
4      <title>Titre de notre page</title>  
5    </head>  
6  
7    <body>  
8      Contenu de notre page!  
9    </body>  
10  </html>
```

Navigateur



# Balises de mise en forme

---

- La balise `<p> </p>` : définit un paragraphe.
- La balise orpheline `<br />` : crée un retour à la ligne.
- Six niveaux de titre, de `<h1> </h1>` à `<h6> </h6>` définissent des titres (par ordre d'importance).
- La balise `<strong> </strong>` est utilisé pour définir un contenu comme important
- la balise `<em> </em>` est utilisé pour définir un contenu comme assez important ( mettre le texte en *italique*)
- La balise `<mark> </mark>` est utilisé pour faire ressortir une portion de texte.

# Balises de mise en forme : les titres

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <title>Titre de notre page</title>
  </head>
  <body>
    <!--Titres-->
    <h1>Le titre général</h1>
    <h2>Une première grande section</h2>
    <p>Un texte d'introduction</p>
    <h3>Un sous-titre dans la première section</h3>
    <h4>Un autre sous-titre du sous dans la première section</h3>
    <h2>Une seconde grande section</h2>
  </body>
</html>
```

*Navigateur*



## **Le titre général**

### **Une première grande section**

Un texte d'introduction

#### **Un sous-titre dans la première section**

**Un autre sous-titre du sous titre dans la première section**

### **Une seconde grande section**

# Balises de mise en forme : Styles

`<B>texte gras</B> <br>`

`<I>texte italique</I><br>`

`<U>texte souligné</U> <br>`

`<S>texte barré</S> <br>`

`texte en<SUB>indice</SUB><br>`

`texte en<SUP>exposant</SUP><br>`

`<BLINK>texte clignotant</BLINK> <br>`



# Balises de mise en forme : LISTES ORDONNEES et NON-ORDONNEES

---

➤ Pour créer une liste non-ordonnée:

- Utiliser la balise `<ul>` `</ul>`
- Pour créer chaque élément de la liste `<li>` `</li>`

➤ Pour créer une liste ordonnée:

- Utiliser la balise `<ol>` `</ol>`
- Pour créer chaque élément de la liste `<li>` `</li>`

# Balises de mise en forme : LISTES ORDONNEES et NON-ORDONNEES

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <title>Titre de notre page</title>
  </head>
  <body>
    <!--Liste non ordonnée-->
    <ul>
      <li>Fraises</li>
      <li>Framboises</li>
      <li>Cerises</li>
    </ul>
    <!--Liste ordonnée-->
    <ol>
      <li>Partie I</li>
      <li>Partie II</li>
      <li>Partie III</li>
    </ol>
  </body>
</html>
```

Navigation



- Fraises
- Framboises
- Cerises

1. Partie I
2. Partie II
3. Partie III



# Exercice

---

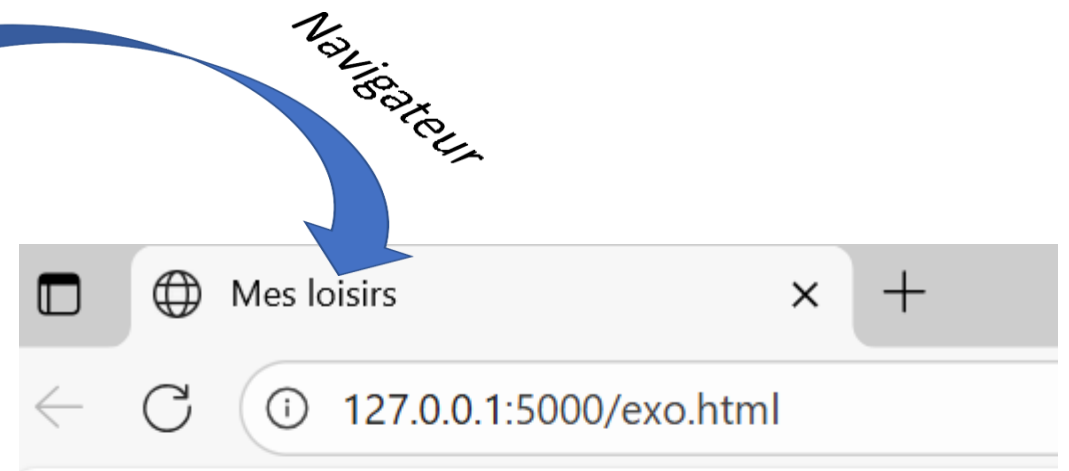
Ecrivez vos premières lignes de HTML en présentant vos loisirs

## Instructions

- Ajoutez un titre à votre page web
- Ajoutez un titre de niveau 1
- A la suite du titre, ajoutez un paragraphe indiquant "Voici mes loisirs:"
- Ajoutez une liste ordonnée à 3 éléments au moins de vos loisirs
- Mettez en valeur (emphase forte) le premier loisir de la liste

# Correction

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title> Mes loisirs </title>
  </head>
  <body>
    <h1> Mes loisirs </h1>
    <p> "Voici mes loisirs" </p>
    <ol>
      <li> <strong>Lecture </strong> </li>
      <li> Sport</li>
      <li> Voyage </li>
    </ol>
  </body>
</html>
```



## Mes loisirs

"Voici mes loisirs"

1. **Lecture**
2. Sport
3. Voyage

# LIENS INTERNES ET EXTERNES

---

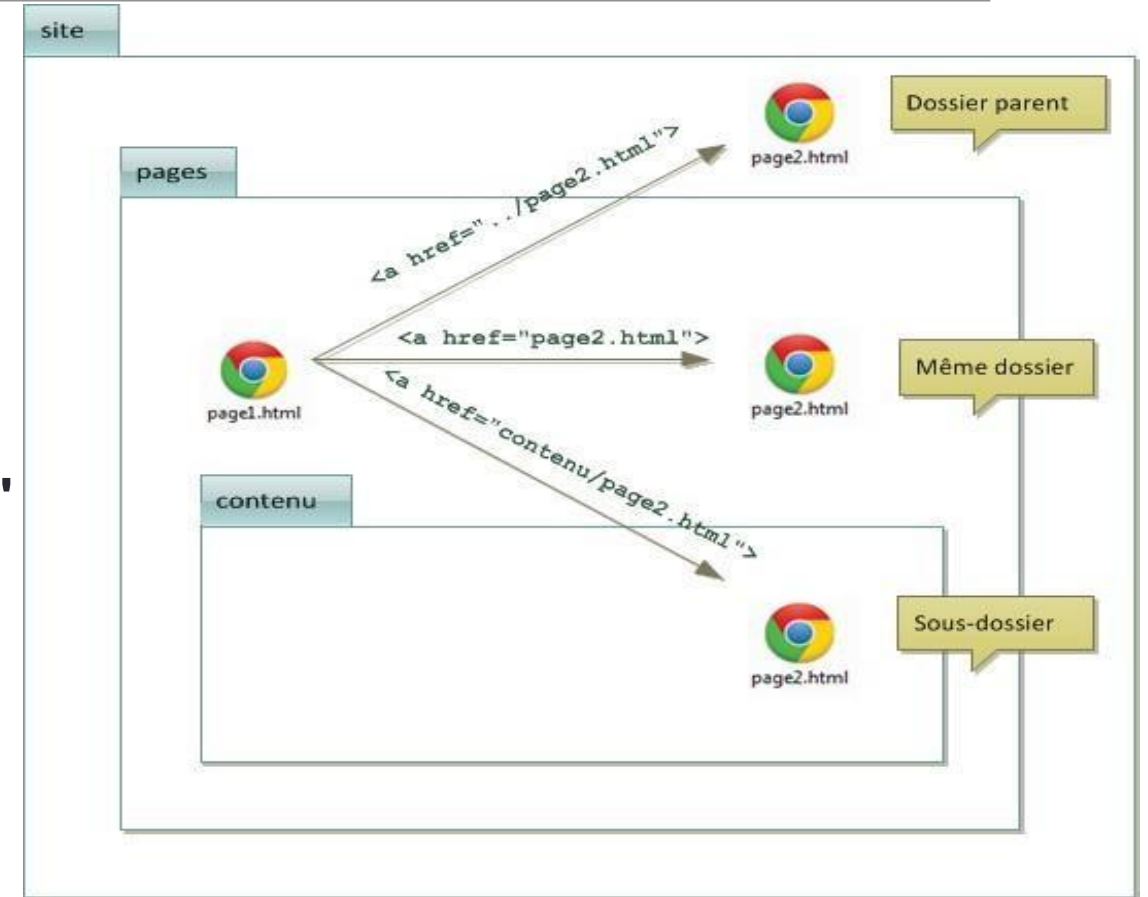
- **Liens internes** : entre deux pages d'un même site.
- **Liens externes** : d'un site vers un autre site.
- **Créer les liens avec:**
  - La balise **<a>** et son attribut **href** (Hypertext reference).

**Syntaxe :**

```
<a href="LA_CIBLE_DU_LIEN">Le texte à afficher</a>
```

# LIENS INTERNES

- Pour créer des liens internes, on utilise un **chemin relatif**. 3 cas :
- Même dossier : href = "page2.html "
  - Sous-dossier : href = "contenu/page2.html "
  - Dossier parent : href=" ../page2.html "



# LIENS EXTERNES

---

- Pour des liens externes, on spécifie **un chemin absolu**.
- La valeur de l'attribut href est l'adresse du site.  
EX: `<a href="https://google.com">vers le site google</a>`
- Pour ouvrir un lien dans une nouvelle fenêtre « onglet », on utilise l'attribut **target** et sa valeur « **\_blank** ».  
EX: `<a href ="https://google.com" target = “_blank “> google</a>`

# AUTRES TYPES DE LIENS

---

- **Un lien vers une ancre:** effectuer un lien vers une autre partie du **même document**
- Syntaxe pour définir l'endroit dans le document (destination) à l'aide de **id**

Exemple: `<h2 id="macible">Mes films préférés</h2>`

- Syntaxe pour définir le lien  
`<a href="#macible">Remonter aux films préférés</a>`

Affiche

**Mes films préférés**

...

[Remonter aux films préférés](#)

# AUTRES TYPES DE LIENS

---

- **Un lien pour envoyer un mail avec mailto.**

Ex: `<a href="mailto:h@gmail.com">Envoyez-moi un e-mail !</a>`

- **Lien pour télécharger un fichier en précisant un chemin relatif.**

Ex: `<a href="monfichier.zip">Télécharger le fichier</a>`

# Les images

---

- Différents formats d'images: JPEG, PNG, GIF...
- Insérer une image avec la balise orpheline **<img/>**
- Utiliser deux attributs obligatoires:
  - L'attribut **src** : indique le chemin de l'image à insérer
  - L'attribut **alt** : décrit le contenu de l'image

**Exemple:**

**<p>**

****

**</p>**



# Attribut style

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <title>Titre de notre page</title>
  </head>
  <body>
    <!--Appliquer des styles-->
    <h1 style="color: red"> premier cours programmation web</h1>
    <p style="font-size: 30px;color: blue;background-color: yellow;text-align: center;">
      autre cours</p>
  </body>
</html>
```



**premier cours programmation web**

autre cours

# Exemples d'attributs du Corps du document

---

```
<body
  bgcolor="#00CCCC"   Couleur de fond d'écran de la page

  background=" ../image.png"
                        Image comme fond d'écran

  text="#330066"      Couleur du texte de la page

  link="#CC3300"       Couleur de liens non encore visités

  vlink="#FF9900"

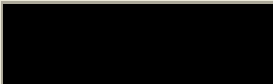

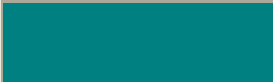

>
</body>
```

# Codes des Couleurs

---

HTML utilise des valeurs hexadécimales pour exprimer les couleurs fondamentales que sont le **Rouge**, le **Vert** et le **Bleu** (RVB)

Exemple: 16 couleurs de base

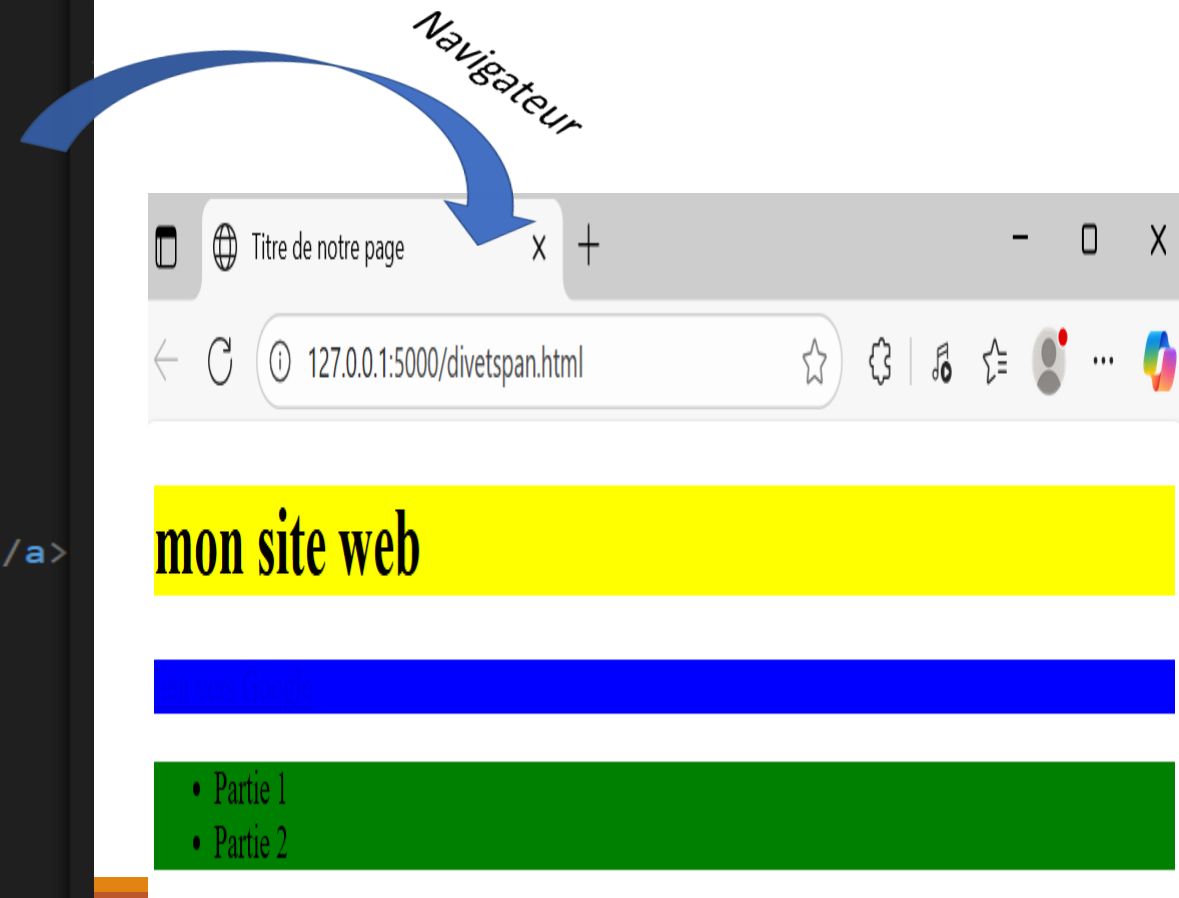
| Nom    | Code    | Résultat   |
|--------|---------|--|
| Black  | #000000 |    |
| Maroon | #800000 |    |
| Green  | #008000 |    |
| Olive  | #808000 |   |
| Navy   | #000080 |  |
| Purple | #800080 |  |
| Teal   | #008080 |  |
| Gray   | #808080 |  |

| Nom     | Code    | Résultat  |
|---------|---------|---|
| White   | #FFFFFF |    |
| Red     | #FF0000 |    |
| Lime    | #00FF00 |    |
| Yellow  | #FFFF00 |   |
| Blue    | #0000FF |  |
| Fuchsia | #FF00FF |  |
| Aqua    | #00FFFF |  |
| Silver  | #C0C0C0 |  |

# Section Div

- Balise **<div>** :élément structurel (ou section)

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <title>Titre de notre page</title>
  </head>
  <body>
    <!--sections-->
    <div style="background-color: yellow ;">
      <h1> mon site web</h1>
    </div>
    <div style="background-color: blue ;">
      <a href="http://google.com" > lien vers Google</a>
    </div>
    <div style="background-color: green ;">
      <ul>
        <li>Partie 1</li>
        <li>Partie 2</li>
      </ul>
    </div>
  </body>
</html>
```



# Les tableaux

---

Un tableau est composé de lignes et de colonnes qui forment les cellules du tableau.

- Définition du tableau [Table] :

**<TABLE>..... </TABLE>** → Début et fin de tableau

- Définition d'une ligne [Table Row] :

**<TR> ..... </TR>** → Début et fin de ligne


- Définition d'une cellule [Table Data] :

**<TD></TD>** → Début et fin de cellule

# Balises de mise en forme : Les tableaux

---

```
<TABLE>
  <TR>
    <TD>1</TD>
    <TD>2</TD>
  </TR>
  <TR>
    <TD>3</TD>
    <TD>4</TD>
  </TR>
</TABLE>
```



---

|   |   |
|---|---|
| 1 | 2 |
| 3 | 4 |

# Les tableaux

---

Bordure de cadre [Border]

**<TABLE border=?> ..... </TABLE>**

L'espace entre les cellules ou l'épaisseur des lignes du quadrillage

**<TABLE cellspacing=?>**

L'enrobage des cellules ou l'espace entre le bord et le contenu

**<TABLE cellpadding=?>**

La largeur de la table

**<TABLE width=?>**

**<TABLE width=%>**

# Balises de mise en forme : Les tableaux

```
<TABLE border=2>
  <TR>
    <TD>1</TD>
    <TD>2</TD>
  </TR>
  <TR>
    <TD>3</TD>
    <TD>4</TD>
  </TR>
</TABLE>
```

|   |   |
|---|---|
| 1 | 2 |
| 3 | 4 |

```
<TABLE border=2 cellpadding=10>
  <TR>
    <TD>1</TD>
    <TD>2</TD>
  </TR>
  <TR>
    <TD>3</TD>
    <TD>4</TD>
  </TR>
</TABLE>
```

|   |   |
|---|---|
| 1 | 2 |
| 3 | 4 |

```
<TABLE border=2 cellspacing=10>
  <TR>
    <TD>1</TD>
    <TD>2</TD>
  </TR>
  <TR>
    <TD>3</TD>
    <TD>4</TD>
  </TR>
</TABLE>
```

|   |   |
|---|---|
| 1 | 2 |
| 3 | 4 |



# Balises de mise en forme : Les tableaux

---

Les cellules peuvent contenir tous les éléments Html :

- du texte
- des images
- des liens
- des arrière-plans
- et même des tableaux

## Largeur d'une cellule

- En pixels **<TD width=?>**
- En pourcentage **<TD width=%>**
- Fusion de lignes **<TD rowspan=?>**
- Fusion de colonnes **<TD colspan=?>**

# Balises de mise en forme : Les tableaux

Exemple 1 :

|           |        |   |
|-----------|--------|---|
| cellule 1 |        |   |
| cellule1  | cel. 2 | 3 |

```
<CENTER>
  <TABLE width=60% border=1>
    <TR>
      <TD colspan=3>cellule 1</TD>
    </TR>

    <TR>
      <TD width=33%>cellule1</TD>
      <TD width=33%>cel. 2</TD>
      <TD width=33%>3</TD>
    </TR>
  </TABLE>
</CENTER>
```

# Balises de mise en forme : Les tableaux

Exemple 2 :

|           |       |   |
|-----------|-------|---|
| cellule 1 | cel 2 | 3 |
|           | cel 2 | 3 |

```
<TABLE width=60% border=1>
  <TR>
    <TD width=33% rowspan=2>cellule 1</TD>
    <TD width=33%>cel 2</TD>
    <TD width=34%>3</TD>
  </TR>
  <TR>
    <TD width=33%>cel 2</TD>
    <TD width=34%>3</TD>
  </TR>
</TABLE>
```

# Les nouvelles balises sémantiques

En-tête

<HEADER>

Qui indique que l'élément est une en-tête

Menu <NAV>

Qui indique un élément de navigation tel qu'un menu

Contenu de la page

<ARTICLE>

Qui correspond à une zone secondaire non liée au contenu principal de la page

Colonne Latérale

<ASIDE>

Qui représente une portion de la page qui garde un sens même séparée de l'ensemble de la page (comme un article de blog par exemple)

<FOOTER>

Bas de page

Qui indique que l'élément est un pied-de-page

# Les nouvelles balises sémantiques

## Mon Site Web

Un exemple sémantique

Accueil

Articles

À propos

Contact

**<main>**

### Premier article

Ceci est le contenu de mon premier article. On utilise la balise `<article>` pour représenter un bloc autonome de contenu.

**<article>**

### Deuxième article

Voici un autre article, séparé du premier. Chaque `<article>` peut être lu indépendamment.

**<aside>**

### Informations complémentaires

Ceci est une zone d'`<aside>`, qui contient des infos connexes (par ex. pub, liens, actu, etc).

**<footer>**

© 2025 - Mon Site Web

# Les nouveautés dans le code HTML

Syntaxe :

```
1 <audio controls>
2   <source src="yumeji_theme.ogg" type="audio/ogg">
3   <source src="yumeji_theme.mp3" type="audio/mpeg">
4   Votre navigateur ne prend pas en charge les formats audio
   proposés.
5 </audio>
```

**Google Chrome**



**Mobile Safari**



**Firefox**



**Desktop Safari**



# Les nouveautés dans le code HTML

---

## *Les balises multimédia*

### – **<video>**

Cette balise intègre directement un lecteur vidéo dans la page, avec des boutons Lecture, Pause, une barre de progression, du volume...



# Les formulaires HTML

Transmettre des informations saisies sur la page Web courante à une autre page Web

|                        |   |
|------------------------|---|
| Etes-vous marié(e)     | <input type="radio"/> Oui <input checked="" type="radio"/> Non                              |
| Cases à cocher         | <input type="checkbox"/> 1 <input checked="" type="checkbox"/> 2 <input type="checkbox"/> 3 |
| Champ de saisie        | <input type="text" value="qqs mots"/>   |
| Mot de passe           | <input type="password" value="*****"/>  |
| Boîte de saisie        | <div>ligne1</div> <div>ligne2</div> <div>ligne3</div>                                       |
| combo                  | <div>choix2</div>   |
| liste à choix simple   | <div>liste1</div> <div>liste2</div> <div>liste3</div>                                       |
| liste à choix multiple | <div>liste1</div> <div>liste2</div> <div>liste3</div>                                       |
| bouton                 | <div>Effacer</div>  |
| envoyer                | <div>Envoyer</div>  |
| rétablir               | <div>Rétablir</div>   |



# Les formulaires HTML

---

Les zones de saisie sont définies par la balise **<INPUT>** qui possède les attributs suivants :

**NAME** : nom du champ de saisie ;

**TYPE** : type de champ ;

- text : texte ;
- password : mot de passe
- checkbox : boîte à cocher
- radio : boîte d'option

**SIZE** : taille du champ

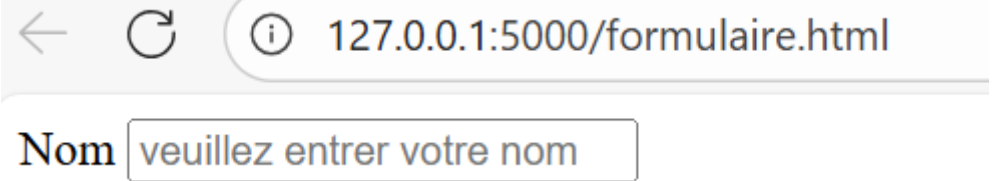
**MAXLENGHT** : nombre maximum de caractères à saisir

# Les formulaires HTML

---

- Il est conseillé d'ajouter à chaque champ un `label` (étiquette décrivant le champ)
- L'identifiant `for` du `label` doit se référer à l'identifiant `id` du champ correspondant
- Ces identifiants ne doivent pas contenir d'espace, et ne seront jamais affichés
- L'attribut `placeholder` contient le texte affiché par défaut (et peut, à l'inverse, contenir tous types de caractères)

```
<body>  
  <form>  
    <label for="nom">Nom</label>  
    <input id="nom" type="text" name="nom" placeholder="veuillez entrer votre nom">  
  
  </form>  
</body>
```



A screenshot of a web browser window. The address bar shows the URL '127.0.0.1:5000/formulaire.html'. Below the address bar, the word 'Nom' is displayed next to a text input field. The input field contains the placeholder text 'veuillez entrer votre nom'.

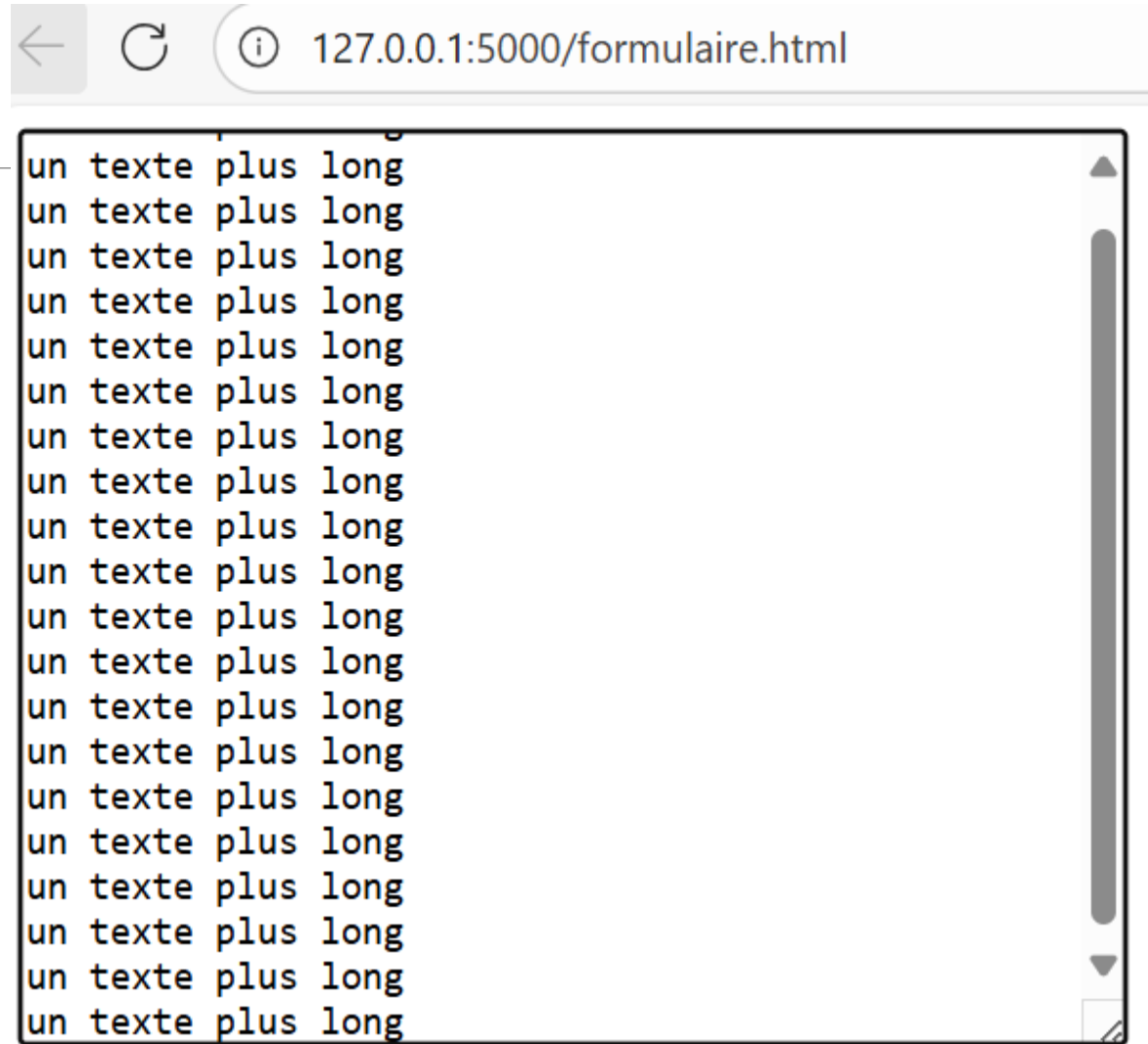
# Les formulaires HTML

## Balise <textarea>

La balise <textarea> permet de définir un espace de texte. Un espace de texte est une grande zone de texte qui permet d'écrire des paragraphes entiers.

```
<body>  
  <form>  
    <textarea cols="50" rows="20"></textarea>  
  </form>  
</body>
```

- **cols**: définit la largeur en caractères de l'espace de texte.
- **rows**: définit la hauteur de l'espace de texte.



The screenshot shows a web browser window with the address bar displaying "127.0.0.1:5000/formulaire.html". The main content area contains a large text area (textarea) with a vertical scrollbar on the right. The text area is filled with 20 lines of the text "un texte plus long", demonstrating the effect of the "rows" attribute. The text is displayed in a monospaced font.

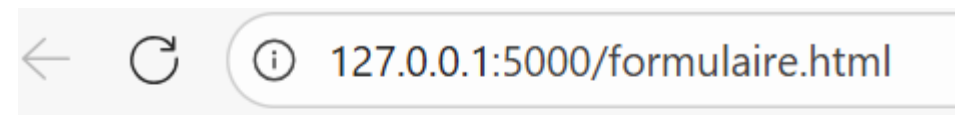
# Les formulaires HTML

---

## Balise <input>

checkbox: cases à cocher (plusieurs choix possibles)

```
<body>
  <form>
    <input type="checkbox" value="velo"> vélo
    <br>
    <input type="checkbox" value="voiture" checked> voiture
    <br>
    <input type="checkbox" value="avion"> avion
  </body>
```



- ☐ vélo
- ☒ voiture
- ☐ avion

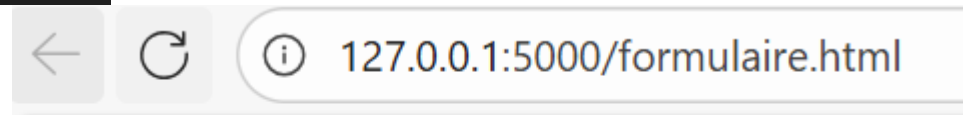
# Les formulaires HTML

---

## Balise <input>

radio: boutons radio (un seul choix possible)

```
<body>
  <form>
    <input type="radio" value="monsieur" name="sexe"> Monsieur
    <br>
    <input type="radio" value="madame" name="sexe"> Madame
  </body>
```



- ☐ Monsieur
- ☒ Madame

# Autres nouveaux types de champs dans les formulaires HTML5

---

## Balise `<select>`

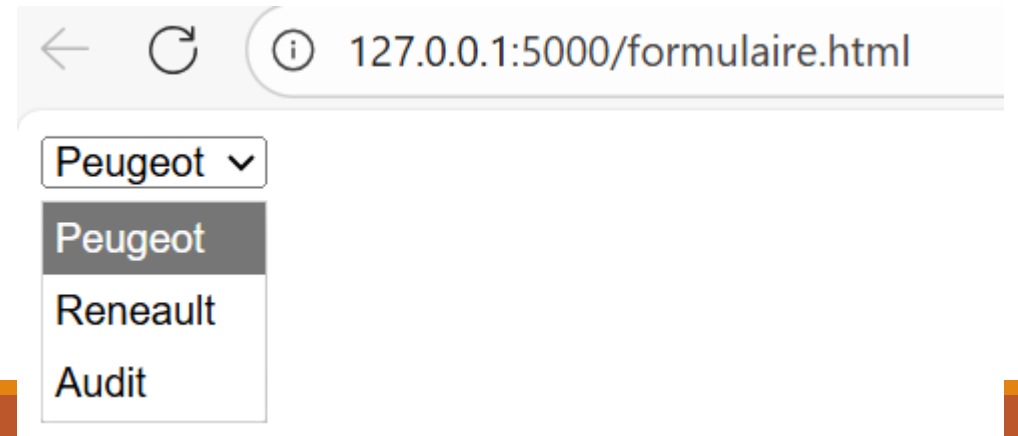
- La balise `<select>` permet de déclarer une liste de sélection (ou liste déroulante)
- Pour remplir cette liste avec les valeurs (appelées options) on doit déclarer une autre balise à l'intérieur, c'est la balise **`<option>`**.
- Dans la balise `<option>` on déclare le mot à afficher (en tant qu'élément de la liste).

**La balise `<option>`** dispose d'un attribut booléen du nom de **`selected`**. Si cet attribut est déclaré, c'est l'option qui le contient qui sera automatiquement sélectionnée au chargement de la page.

# Autres nouveaux types de champs dans les formulaires HTML5

## Balise <select>

```
<body>
  <form>
    <select name="voiture">
      <option value="Peugeot">Peugeot</option>
      <option value="Renault">Renault</option>
      <option value="Audit">Audit</option>
    </select>
  </form>
</body>
```



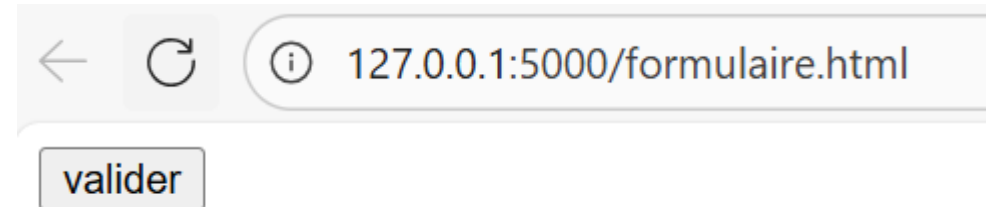
A screenshot of a web browser window. The address bar shows the URL "127.0.0.1:5000/formulaire.html". Below the address bar, there is a dropdown menu with the text "Peugeot" and a downward arrow. The dropdown menu is open, showing three options: "Peugeot", "Renault", and "Audit". The "Peugeot" option is highlighted.

# Autres nouveaux types de champs dans les formulaires HTML5

---

## Balise `<input>` de type bouton

```
<body>  
  <form>  
    <input type="submit" value="valider">  
  </form>  
</body>
```





# Les formulaires HTML

---

## *Les formulaires classiques:*

```
<FORM METHOD="POST" ou "GET" ACTION="url">  
...  
</FORM>
```

**Balise <form>** : délimite les champs du formulaire (zones de texte, boutons...). Elle peut avoir plusieurs attributs dont voici les plus fréquents:

**L'attribut name** permet d'identifier le formulaire par un nom.

- On peut mettre n'importe quel nom (à conditions qu'il ne contienne que les caractères conventionnels comme les lettres et les chiffres).
- Le nom du formulaire est unique.
- On peut déclarer plusieurs formulaires dans la même page, pour les distinguer les uns des autres il suffit de leur donner des noms différents.

# Les formulaires HTML

---

## *Les formulaires classiques:*

<FORM **METHOD="POST"** ou **"GET"** ACTION="url">

L'**attribut method** permet de spécifier la méthode à utiliser pour envoyer le formulaire au serveur.

- l'objectif d'un formulaire c'est de transférer les données saisies par le client au serveur (on dit généralement **poster** ou **soumettre** le formulaire). En HTML, il existe deux méthodes pour envoyer un formulaire: **GET** et **POST**.

L'**attribut action** : c'est l'adresse de la page ou du programme qui va traiter les informations

# Les formulaires HTML

---

## *Les formulaires classiques:*

<FORM **METHOD="POST"** ou **"GET"** ACTION="url">

- ❑ La méthode **GET** (method="get") est la valeur par défaut si on ne précise aucune valeur pour l'attribut method. Cette méthode permet d'envoyer les données du formulaires à travers l'URL en utilisant les symboles **?** pour préciser la suite de données et **&** pour séparer les données entre elles.
- ❑ La méthode **POST** (method="post") permet d'envoyer les données du formulaire au serveur à travers une entête. Les informations ne seront donc pas visibles sur l'URL. La méthode **POST** permet aussi de poster un volume de données plus important que celui transmis par **GET**.

# Validation et compatibilité HTML

---

- L'un des problèmes majeurs que rencontrent les intégrateurs HTML est la compatibilité de leur pages Web.
- Une page Web ne s'affiche pas toujours de la même manière d'un navigateur à un autre.
- Pour cela, ils se servent généralement du validateur de W3C dont le lien est [validator.w3.org](https://validator.w3.org) afin de rendre le code conforme aux spécifications du Consortium

Ce qui donne dans le navigateur :

☐ Madame ☐ Mademoiselle ☐ Monsieur

Votre prénom :

Votre nom :

Votre mot de passe :

Vous êtes

Un professionnel ▼

Quel type de prestation recherchez vous ?

☐ Location de mobilier ☐ Achat de mobilier

Votre message :

Vous pouvez saisir ici un message.

Envoyer

Annuler

# Exercice: créer un formulaire d'inscription

---