

# Développement Web - INF04

*Markup Languages HTML* et Formulaires

Mussab Zneika  
CY-Tech

# LANGAGES DE BALISAGE

## Le balisage de document

- Un langage de balisage est un langage informatique qui utilise des balises(tags) pour définir des éléments dans un document.
- Les langages de balisage sont des moyens d'annoter un document électronique.
- le balisage spécifie comment quelque chose doit être affiché ou ce que quelque chose signifie.
- Certains langages de balisage populaires sont
  - ▶ le langage de balisage hypertexte (HTML),
  - ▶ le langage de balisage extensible (XML) et
  - ▶ le langage de balisage hypertexte extensible (XHTML).

# LANGAGES DE BALISAGE

## HTML

- Hypertexte désigne un texte stocké sous forme électronique avec des liens de renvoi entre les pages.
- Le langage de balisage hypertexte est utilisé pour faciliter la publication de pages Web en fournissant une structure qui définit des éléments tels que des tableaux, des formulaires, des listes et des en-têtes, et identifie où commencent et se terminent les différentes parties de notre contenu.
- Il peut être utilisé pour intégrer d'autres formats de fichiers tels que des vidéos, des fichiers audio, des documents tels que des PDF et des feuilles de calcul, entre autres.
- HTML est le langage le plus utilisé dans la création de sites Web.

# LANGAGES DE BALISAGE

## HTML Exemple

```
<html>

  <head>

  </head>

  <body>
    <p> This is a paragraph.</p>
  </body>

</html>
```

## BALISES, ATTRIBUTS

- HTML repose sur la notion de **Balises** (tags).
- les **Balises** permettent de structurer le contenu d'un document HTML
- Les balises HTML peuvent être de deux types :
  - ✓ balises qui sont ouvertes puis fermées, et encadrent du contenu  
**Example: <b> gras texte </b>**
  - ✓ balises qui s'ouvrent et se ferment en même temps  
**Example: <br /> , <hr />, <meta>, <img>**

```
<h1> Exemple </h1>
<hr />
<p>
<b>gras</b> puis <em>italique</em>, puis <b><em>gras et italique</em></b>.
</p>
```

## BALISES, ATTRIBUTS

- Les balises HTML peuvent avoir des **attributs** qui configurent ces balises ou ajustent leurs comportements de différentes manières
- Les attributs sont déclarés au sein la balise ouvrante de l'élément

```
<p align = "right">C'est aligné à droite </p>
```

- En HTML, les noms de balise et d'attribut ne sont pas sensibles à la casse.

```
<B> Valid Text.</B>  
<b> valid Text.</b>
```

- Il est important de fermer les balises dans l'ordre

```
<b><em>gras et italique</em></b>
```

# HTML PAGE LAYOUT

## Structure globale du document



```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
<title> Web programming Class </title>
```

```
</head>
```

```
<body>
```

```
  Hello world
```

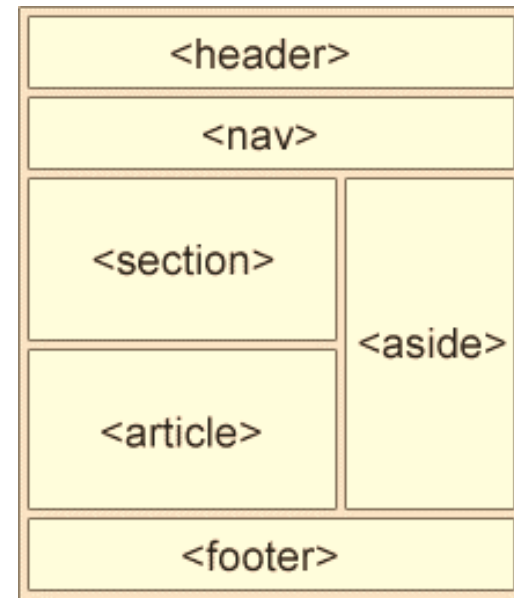
```
</body>
```

```
</html>
```

# BALISES EN HTML5

## Structuration du document HTML5

- Entête : `<header>` `</header>`
- Menu de navigation : `<nav>` `</nav>`
- Section : `<section>` `</section>`
- Article : `<article>` `</article>`
- Encadré : `<aside>` `</aside>`
- Pied de page : `<footer>` `</footer>`
- Boîte de dialogue : `<dialog>` `</dialog>`





# CRÉATION DE FICHIERS HTML

## Format de fichier

- Avant de pouvoir créer notre première page Web, nous devons créer un fichier que notre service reconnaitra comme une page Web.
- Pour ce faire, nous pouvons ouvrir notre éditeur de texte choisi (voir une courte liste de potentiels dans la section sur les environnements de développement intégrés), et créer un nouveau document s'il n'a pas été créé automatiquement.
- Nous sélectionnerons immédiatement « Enregistrer sous » dans le menu Fichier de votre éditeur et donnerons son nom à votre nouvelle page.
- Quelques éditeurs HTML gratuits :
  - ▶ Notepad++
  - ▶ Brackets.io
  - ▶ Atom
  - ▶ Visual Studio Code
  - ▶ Sublime Text 3

# PAGE LAYOUT

## Tête (Head)

- ❖ C'est là que nous mettons des informations sur la page elle-même (appelées métadonnées).
- ❖ Remarque : Il est important que vos balises `<head>` soient toujours le premier ensemble de balises après `<html>`.
- ❖ Il est très fortement conseillé de toujours spécifier :
  - ✓ Le titre
  - ✓ L'auteur
  - ✓ La description
  - ✓ Les mots-clés

```
<head>
  <title>Our First Page</title>
  <meta name="author" content="Your Name Here" />
  <meta name="description" content="A beginning web page for CSIT-107 SUNY Fredonia" />
  <meta name="keywords" content="beginning html, learning, CSIT-107" />
</head>
```

# PAGE LAYOUT

## > Tête (Head)

- > Étant donné que notre contenu ne changera pas, et que nous n'avons pas encore de feuille de style, nous allons simplement ajouter notre déclaration de type de contenu pour le moment.

```
<meta http-equiv="content-type" content="text/html" />
```

```
<META HTTP-EQUIV="EXPIRES" CONTENT="Mon, 22 Jul 2002 11:12:01 GMT">  
<META HTTP-EQUIV="CACHE-CONTROL" CONTENT="NO-CACHE">
```

## PAGE LAYOUT

### Corps (Body)

- Tout le contenu que nous souhaitons avoir à l'écran doit être englobé par un ensemble de balises corporelles.

```
<body>
  <a href="page2.html">Hello in Web progrming Class!</a>
  <p> Teacher: DR Mussab Zneika </p>

</body>
```

# PAGE LAYOUT

## Corps (Body) Attributs (avant HTML5)

- ✓ **BGColor**="..."

La couleur de l'arrière-plan de la page .

-<body BGColor="#9999ff">

- ✓ **Background**="..."

Le nom ou l'URL de l'image à afficher sur l'arrière-plan de la page.

- ✓ **Text**="..."

La couleur du texte de la page. (la valeur par défaut est le noir).

- ✓ **Link**="..."

La couleur des liens non suivis. (la valeur par défaut est le bleu).

- ✓ **ALink**="..."

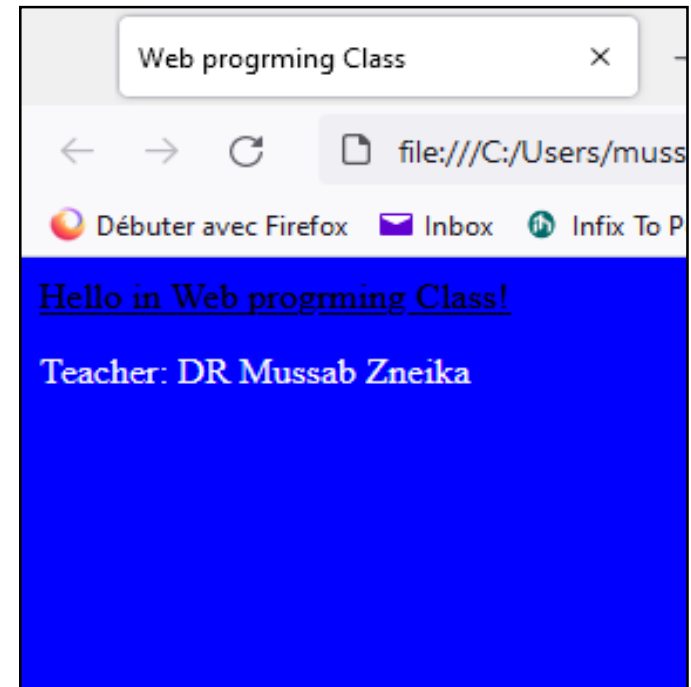
La couleur des liens activés. (la valeur par défaut est le rouge).

- ✓ **VLink**="..."

La couleur des liens visités. (la valeur par défaut est le violet)

## PAGE LAYOUT

### Corps (Body) attributs



```
<!DOCTYPE html>
<html lang="en">
  <head> <title> Web progrming Class </title>
  </head>

  <body bgcolor="blue" text="white" link="black">
    <a href="page2.html">Hello in Web progrming Class!</a>
    <p> Teacher: DR Mussab Zneika </p>
  </body>
</html>
```

# Attributs universels

- Les **attributs universels** sont des attributs qui peuvent être ajoutés sur tous les éléments.
- Certains des attributs couramment utilisés sont :
  - **class** Spécifie un ou plusieurs noms de classe pour un élément (fait référence à une classe dans une feuille de style CSS).
  - **Id** définit un identifiant, unique au sein de tout le document, pour un élément.
  - **style** Spécifie un style CSS en ligne pour un élément.
  - **lang** Spécifie la langue du contenu de l'élément.

# PAGE LAYOUT

## Attributs universels

- **title** Contient une présentation textuelle de l'information auquel il est lié.
- **dir** Spécifie la direction du texte pour le contenu d'un élément. Les valeurs autorisées pour cet attribut sont :
  - ✓ ltr (Left To Right) indique que le contenu est écrit de gauche à droite.
  - ✓ rtl (Right To Left) indique que le contenu est écrit de droite à gauche.
  - ✓ auto : c'est l'utilisateur qui décide.
- **contenteditable** Spécifie si le contenu d'un élément est modifiable ou non.

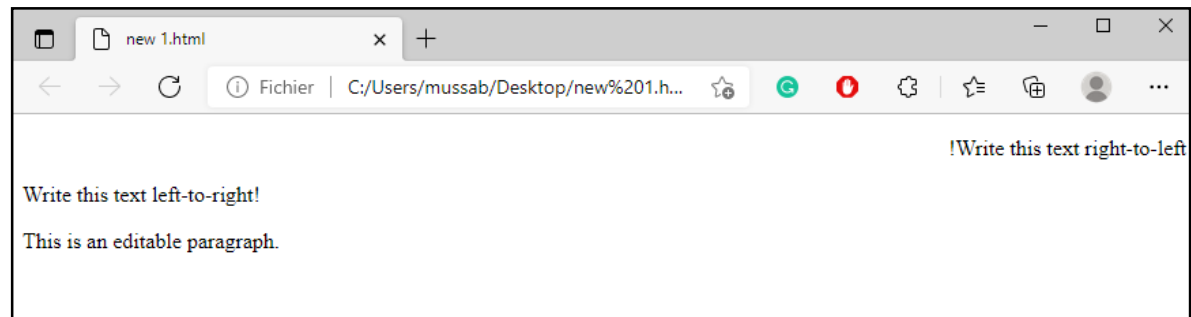


# PAGE LAYOUT

## Attributs universels : Exemple

```
<html>
<body>
<p id="p1" dir="rtl">Write this text right-to-left!</p>
<p dir="ltr">Write this text left-to-right!</p>
<p contenteditable="true">This is an editable paragraph.</p>
</body>
</html>
```

Dans cet exemple nous montrons comment utiliser l'attribut **dir** pour le sens des paragraphes de texte. Le premier de droite à gauche et le second de gauche à droite.



## MISE EN FORME DE TEXTE

### > Balises de formatage des caractères

✓ **<B>...</B>** Texte en gras.

✓ **<STRONG>...</STRONG>** Emphase plus forte (généralement en gras).

✓ **<I>...</I>** Texte en italique.

✓ **<EM>...</EM>** Emphase (généralement en italique).

✓ **<SMALL>...</SMALL>** Le texte est légèrement plus petit que la normale.

## PAGE LAYOUT

### Balises de formatage des caractères

```
<B>Texte en gras.</B> <br>  
<STRONG>Emphase plus forte (généralement en gras).</STRONG> <br>  
<I>Texte en italique.</I> <br>  
<EM>Emphase (généralement en italique)</EM> <br>  
<SMALL>le texte est légèrement plus petit que la normale </SMALL>
```

**Texte en gras.**  
**Emphase plus forte (généralement en gras).**  
*Texte en italique.*  
*Emphase (généralement en italique)*  
le texte est légèrement plus petit que la normale

## MISE EN FORME DE TEXTE

### Balises de formatage des caractères

- ✓ **<SUB>...</SUB>** Subscript.
- ✓ **<SUP>...</SUP>** Superscript.
- ✓ **<U>...</U>** Underlined text.
- ✓ **<TT>...</TT>** Chaque lettre utilise la même quantité d'espace horizontal.

# MISE EN FORME DE TEXTE

## Balises de formatage des caractères

```
Test <SUB>Subscript.</SUB> <br>  
Test <SUP>Superscript.</SUP> <br>  
<U>underlined text.</U> <br>  
<TT> Chaque lettre utilise la même quantité d'espace horizontal</TT>
```

```
TestSubscript.  
Test Superscript.  
underlined text.  
Chaque lettre utilise la même quantité d'espace horizontal
```

## MISE EN FORME DE TEXTE (PLUS)

### Texte marqué

- L'élément **<mark>** est nouveau dans HTML5 et est utilisé pour marquer ou surligner une section de texte dans un document.

```
<p>This is <mark>marked text</mark> yellow</p>
```

Couleur jaune par default.

This is **marked text** yellow

```
<p>This is <mark style="background-color:red;"> marked  
text</mark> red</p>
```

This is **marked text** red

## MISE EN FORME DE TEXTE (PLUS)

### Abréviation

- Nous utilisons la balise **<abbr>** pour marquer une expression comme abréviation(forme abrégée d'un mot ou d'une phrase).
- L'attribut **title** est utilisé pour spécifier le terme complet.

```
<p>in our class we study the  
<abbr title="Hypertext Markup Language">HTML</abbr>  
</p>
```

in our class we study the HTML

Hypertext Markup Language

## MISE EN FORME DE TEXTE (PLUS)

### Modifications apportées au contenu

- La balise **<ins>** est utilisé pour afficher le contenu qui a été inséré dans un document.
- La balise **<del>** est utilisé pour afficher du texte qui a été supprimé d'un document.
- Le contenu d'un élément **<ins>** est généralement souligné, alors que le contenu d'un élément **<del>** est généralement barré.

```
<p><ins>Inserted text</ins></p>  
<p><del>Deleted text</del></p>
```

Inserted text

~~Deleted text~~



# MISE EN PAGE DU TEXTE (TEXT LAYOUT)

## Paragraphes

- ❖ Pour développer un peu notre structure de base, nous pouvons diviser une longue section de texte en paragraphes.
  - Nous pouvons le faire en ajoutant des pauses (`<br/>`) dans notre code.
- ❖ Si nous allons le faire plusieurs fois, et si nous voulons styliser nos paragraphes sur toute la ligne, nous devrions plutôt envelopper chacun d'eux dans un ensemble de balises de paragraphe, `<p></p>`.
  - L'utilisation des balises de paragraphe nous permet d'ajouter automatiquement un espacement autour de notre contenu pour le séparer du reste de la page.

Avant	Après
<pre>&lt;body&gt;   This is some text. It is really long. We want to break this   into paragraphs so it looks more like a document. This is some   text. It is really long. We want to break this into paragraphs so   it looks more like a document. This is some text. It is really   long. This is some text. It is really long. We want to break this   into paragraphs so it looks more like a document. This is some   text. It is really long. We want to break this into paragraphs so   it looks more like a document. This is some text. It is really   long. We want to break this into paragraphs so it looks more like a   document. &lt;/body&gt;</pre>	<pre>&lt;body&gt;   &lt;p&gt; This is some text. It is really long. We want to break   this into paragraphs so it looks more like a document. This is some   text. It is really long. We want to break this into paragraphs so   it looks more like a document. This is some text. It is really   long. &lt;/p&gt;   &lt;p&gt; This is some text. It is really long. We want to break   this into paragraphs so it looks more like a document. This is some   text. It is really long. We want to break this into paragraphs so   it looks more like a document. This is some text. &lt;/p&gt;   &lt;p&gt; This is some text. It is really long. We want to break   this into paragraphs so it looks more like a document. This is some   text. It is really long. We want to break this into paragraphs so   it looks more like a document. &lt;/p&gt; &lt;/body&gt;</pre>

# PAGE LAYOUT

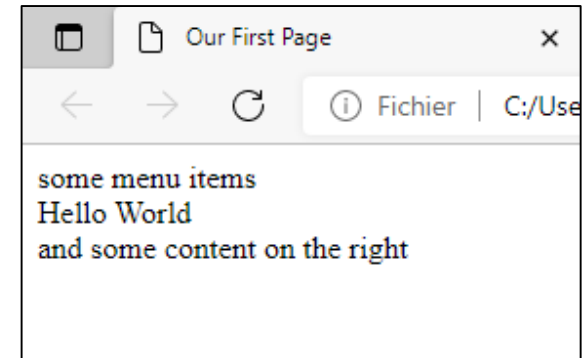
## Div/Span

- Alors que `<p>` nous aide à diviser notre texte, nous avons également besoin d'un mécanisme pour séparer différents éléments de contenu, comme nous le faisons lorsque nous utilisons l'en-tête et le pied de page.
  - Cela nous permettra de définir plus qu'un simple haut, milieu et bas de notre page. Pour ce faire, nous pouvons envelopper ces sections dans des balises `<div>`.
- L'élément HTML `<div>` permet de regrouper un ensemble d'éléments en un seul bloc.
- L'élément HTML `<span>` est très similaire à `div`, sauf qu'il doit identifier le contenu en ligne, c'est-à-dire le matériel qui se trouve dans un bloc de texte.

# PAGE LAYOUT

## Div/Span : Exemple

```
<body>
  <header>
    <h1>This is our first page! </h1>
  </header>
  <div id="left">
    some menu items
  </div>
  <div id="content">
    Hello World
  </div>
  <div id="right">
    and some content on the right
  </div>
</body>
```



- L'attribut **align** est désormais **obsolète** et ne doit plus être appliqué pour un **<div>**
- Vous remarquerez que la mise en page ne va pas à gauche ou à droite lorsque nous étiquetons nos « **divs** » (tout s'organisera de haut en bas).
  - C'est parce que nous devons ajouter CSS pour le plein effet. Nous reprendrons cet exemple plus tard pour ajouter le CSS nécessaire à la création de la mise en page souhaitée.

# PAGE LAYOUT

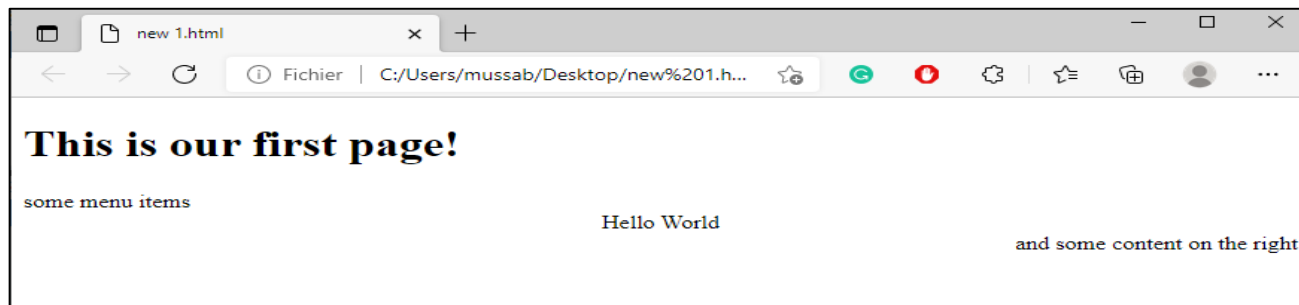
## Div/Span : Exemple

```
<html>
  <head>
    <style>
      .lefttext {text-align: left;}
      .righttext{text-align: right;}
      .centertext{text-align: center;}
    </style>
  </head>

  <body>
    <h1>This is our first page! </h1>
    <div id="left" class="lefttext"> some menu items </div>
    <div id="content" class="centertext"> Hello World </div>
    <div id="right" class="righttext"> and some content on the right </div>
  </body>
</html>
```



Dans cet exemple, nous utilisons trois classes CSS (righttext, lefttext, centertext) pour définir l'alignement de trois éléments div (gauche, centre, droite )



## MISE EN PAGE DU TEXTE (TEXT LAYOUT)

### Éléments de titres HTML

- Les titres HTML sont des titres ou des sous-titres que vous souhaitez afficher sur une page Web :
  - `<H1>...</H1>` Un titre de premier niveau.
  - `<H2>...</H2>` Un titre de deuxième niveau.
  - `<H3>...</H3>` Un titre de troisième niveau .
  - `<H4>...</H4>` Un titre de quatrième niveau.
  - `<H5>...</H5>` Un titre de cinquième niveau.
  - `<H6>...</H6>` Un titre de sixième niveau.

## MISE EN PAGE DU TEXTE (TEXT LAYOUT)

### Éléments de titres HTML: Exemple

```
<body>  
<H1>A first-level heading.</H1>  
<H2>A second-level heading.</H2>  
<H3>A third-level heading.</H3>  
<H4>A fourth-level heading.</H4>  
<H5>A fifth-level heading.</H5>  
<H6>A sixth-level heading</H6>  
</body>
```

**A first-level heading.**

**A second-level heading.**

**A third-level heading.**

**A fourth-level heading.**

**A fifth-level heading.**

**A sixth-level heading**

## MISE EN PAGE DU TEXTE (TEXT LAYOUT)

### Listes ordonnées et non ordonnées

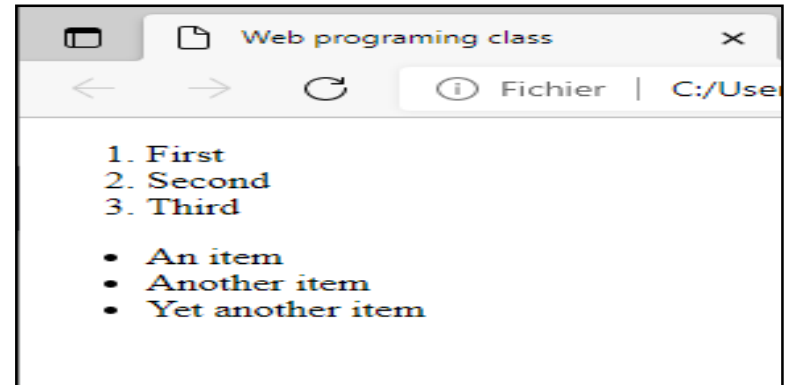
- Nous utilisons les balises **<ol>** (listes ordonnées) ou **<ul>** (listes non ordonnées).
- Les listes ordonnées, en revanche, concernent les articles qui doivent être commandés pour une raison, comme des instructions qui doivent être suivies dans le bon ordre.
- Les listes ordonnées et non ordonnées sont respectivement des listes d'éléments alphanumériques et non ordonnées.
- En les utilisant, nous pouvons créer des listes pour les afficher à l'écran car nous sommes habitués à voir une liste d'éléments.
- L'affichage des listes peut être personnalisé par les feuilles de style CSS.
- Nous pouvons placer des balises **<li>** imbriquées dans les balises **<ol>** ou **<ul>** pour représenter chaque élément de la liste.

# MISE EN PAGE DU TEXTE (TEXT LAYOUT)

## Listes ordonnées et non ordonnées : Example

```
<ol>
<li>First</li>
<li>Second</li>
<li>Third</li>
</ol>

<ul>
<li>An item</li>
<li>Another item</li>
<li>Yet another item</li>
</ul>
```

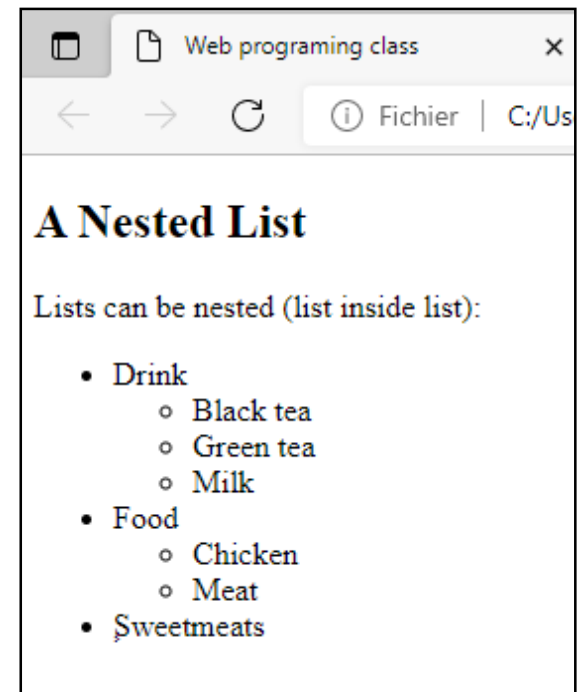




# MISE EN PAGE DU TEXTE (TEXT LAYOUT)

## Listes ordonnées et non ordonnées : Example

```
<body>
<h2>A Nested List</h2>
<p>Lists can be nested (list
inside list):</p>
<ul>
  <li>Drink
    <ul>
      <li>Black tea</li>
      <li>Green tea</li>
      <li>Milk</li>
    </ul>
  </li>
  <li>Food
    <ul>
      <li>Chicken</li>
      <li>Meat</li>
    </ul>
  </li>
  <li>Sweetmeats</li>
</ul>
```

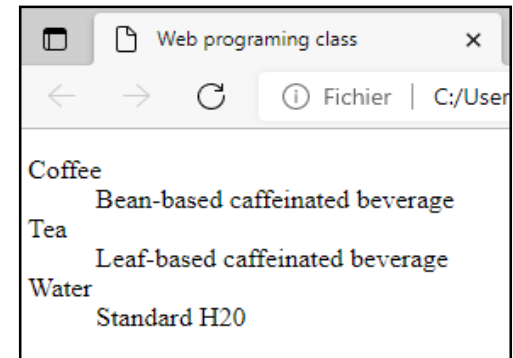


# MISE EN PAGE DU TEXTE (TEXT LAYOUT)

## Listes de définitions

- Un ensemble de balises associé peut être utilisé lorsque vous souhaitez répertorier des définitions.
- Ce sont **<dl>** pour la liste elle-même, avec **<dt>** imbriqué à l'intérieur pour les termes et **<dd>** également imbriqué, pour la définition, à la suite de son **<dt>** correspondant.

```
<dl>
<dt>Coffee</dt>
  <dd>Bean-based caffeinated beverage
</dd>
<dt>Tea</dt>
  <dd>Leaf-based caffeinated beverage</dd>
<dt>Water</dt>
  <dd>Standard H2O</dd>
</dl>
```



# MISE EN PAGE DU TEXTE (TEXT LAYOUT)

## Entités

- Dans les exemples de la section En-tête et pied de page, nous avons placé un symbole de droit d'auteur à l'écran à l'aide de « **&copy;** »
- « **&copy;** » a indiqué au navigateur quel symbole nous voulions utiliser. Il s'agit d'un symbole réservé, ou d'une entité, en HTML.
- Nous pouvons appeler des entités en utilisant **&[nom de l'entité ici];** ou **&[numéro d'entité ici]**
- Par exemple, **&nbsp;** signifie un espace standard. C'est une façon d'insérer des espaces supplémentaires dans notre sortie.

Result	Description	Name	Number
	non-breaking space	&nbsp;	&#160;
<	less than	&lt;	&#60;
>	greater than	&gt;	&#62;
&	ampersand	&amp;	&#38;
©	copyright	&copy;	&#169;
®	registered trademark	&reg;	&#174;
™	trademark	&trade;	&#8482;

# NAVIGATION

## Enchaînement (Linking)

- Les liens nous permettent de passer d'une page Web à une autre.
- Il existe généralement plusieurs types de liens :
  - ✓ Liens d'une page à une autre sur le même site Web.
  - ✓ Liens d'une partie d'une page Web vers une autre partie du même page.
  - ✓ Liens qui s'ouvrent dans une nouvelle fenêtre de navigateur.
  - ✓ Liens qui démarrent votre programme de messagerie et adressent un nouvel e-mail à quelqu'un.
- Les liens en HTML peuvent prendre deux formes, toutes deux créées avec la balise d'ancrage (**<a>**).

# NAVIGATION

## Enchaînement (Linking)

- L'élément **<a>** permet de définir des liens hypertextes externes ou internes au document.
- La valeur de l'attribut **href** est la page que les utilisateurs retrouvent quand ils cliquent sur le lien.

```
<a href=https://www.w3.org/>W3C </a>
```

- Le texte entre l'ouverture Balise **<a>** et balise de fermeture **</a>** est connu sous le nom de texte de lien.
- Lorsque nous établissons un lien vers un site Web différent, la valeur de l'attribut href sera l'adresse Web complète du site, connue sous le nom d'URL absolue.

## NAVIGATION

### Enchaînement (Linking) :Exemple

```
<H3> Useful Sites </H3>
<ul>
  <li><a href="https://www.google.com/" >google</a></li>
  <li><a href="https://www.yahoo.com/" >yahoo</a></li>
  <li><a href="https://www.w3schools.com/" >W3C</a></li>
</ul>
```

#### Useful Sites

- [google](https://www.google.com/)
- [yahoo](https://www.yahoo.com/)
- [W3C](https://www.w3schools.com/)

# NAVIGATION

## Cible (Target) Attribut:

- Bien que la balise d'ancrage prenne en charge plusieurs attributs, l'un des plus importants d'entre eux est « cible ».
- Cet attribut décrit où les liens seront chargés, comme un nouvel onglet ou la même fenêtre d'onglet/navigateur que nous utilisons déjà.
- L'attribut peut prendre l'une des valeurs suivantes pour définir cet emplacement.
  - ✓ **\_blank** Affiche le nouveau document dans une nouvelle fenêtre non nommée.
  - ✓ **\_self** Le nouveau document s'affiche dans le même cadre que l'ancrage qui le charge ; c'est la valeur par défaut.
  - ✓ **\_parent** Affiche le nouveau document dans le cadre parent ; si pas de parent, comme **\_self**.
  - ✓ **\_top** Affiche le nouveau document dans toute la fenêtre ; s'il n'y a pas de cadres, comme **\_self**.
  - ✓ **name** Un cadre nommé dans l'attribut Name d'un élément FRAME.

## NAVIGATION

### Cible (Target)

- Exemple (Ouvrir le lien dans un nouvel onglet/une nouvelle fenêtre )
  - Dans cet exemple, nous utilisons l'attribut **target** sur la balise d'ouverture **<a>** avec la valeur **\_blank** pour dire au navigateur d'ouvrir le lien google dans une nouvelle fenêtre ou un nouvel onglet.

```
<body>  
  <a href="http://www.google.com" target="_blank"> Google</a>  
</body>
```



# NAVIGATION

## Lien vers un autre site

- Lorsque vous créez un lien vers un site Web différent, la valeur de l'attribut **href** sera l'adresse Web complète du site, qui est connue sous le nom d'URL absolue.

```
<H3> Useful Sites </H3>
<ul>
  <li><a href="https://www.google.com/" >google</a></li>
  <li><a href="https://www.yahoo.com/" >yahoo</a></li>
  <li><a href="https://www.w3schools.com/" >W3C</a></li>
</ul>
```

### Useful Sites

- [google](https://www.google.com/)
- [yahoo](https://www.yahoo.com/)
- [W3C](https://www.w3schools.com/)

# NAVIGATION

## Lien vers d'autres pages sur le même site

- Si les pages liées au site sont dans le même dossier, puis la valeur de l'attribut **href** est juste.
- S'ils sont dans différents dossiers, alors nous utilisons les chemins relatifs.

```
<H1 > Welcome to my home page </H1>
<a href="about.html" >About me</a> &nbsp; || &nbsp;
<a href="Teaching.html" >My Teaching</a> &nbsp; || &nbsp;
<a href="publications.html" >My Publications </a> &nbsp; || &nbsp;
<a href="CV.html" >My Short CV </a>
```

**Welcome to my home page**

[About me](#) || [My Teaching](#) || [My Publications](#) || [My Short CV](#)

# NAVIGATION

## Lien vers une spécificité partie de la même page

- La balise `<a>` peut pointer vers n'importe quel élément ayant un `id` attribut.
- La valeur de l'attribut `href` commence par le symbole `#`, suivi de la valeur de l'attribut `id` de l'élément auquel vous souhaitez établir un lien.

```
<H1 id="top" > Welcome to my home page </H1>
<a href="#down">Click here to go further down.</a>
Some more text.
Even more text!
<a href="#top"> go to top of the page </a>
<h1 id="down"> This is where we want to "jump" to.</h1>
```

## NAVIGATION

### Liens Adresse e-mail et numéro de téléphone

- Si la valeur de l'attribut **href** commence par **tel:**, votre appareil composera le numéro lorsque vous cliquerez dessus. Cela fonctionne sur appareils exécutant des logiciels permettant de passer des appels téléphoniques.

```
<a href="tel:0033xxxxxxx">Call us</a>
```

- Si la valeur de l'attribut **href** commence par **mailto:** il essaiera d'ouvrir le programme de messagerie de l'utilisateur et adressera un e-mail à une adresse e-mail spécifiée.

```
<a href="mailto:mussab.zneika@cyu.fr">Send email</a>
```

# NAVIGATION

## Nav

- HTML5 définit un élément **<nav>** qui peut être utilisé pour contenir la navigation principale d'un site Web, qu'il s'agisse d'une liste de liens ou d'un élément de formulaire tel qu'un champ de recherche.
- Ceci est particulièrement utile pour les logiciels de lecture d'écran, car les balises indiquent à quoi servent les liens.

```
<nav >
  <ul>
    <li><a href="about.html">About Us</a></li>
    <li><a href="clients.html">Our Clients</a></li>
    <li><a href="services.html">Our Services</a></li>
    <li><a href="contact.html">Contact Us</a></li>
  </ul>
</nav>
```

- [About Us](#)
- [Our Clients](#)
- [Our Services](#)
- [Contact Us](#)

# GRAPHIQUES

## Images

- Traditionnellement, nous avons fait face à ce compromis en utilisant différents formats d'image dans différentes zones de notre site.
- Tout en réservant JPG pour nos images ou photos plus grandes, nous pouvons utiliser GIF pour les icônes et indicateurs plus petits.
- Les GIF nous limitent à 256 couleurs, mais comme la plupart des icônes utilisent peu de couleurs, nous pouvons ici capitaliser sur les avantages de ce format.
- Il est important de noter que les images raster perdent rapidement en qualité lorsqu'elles sont rendues à des tailles supérieures à la largeur ou à la hauteur de l'image d'origine.

# GRAPHIQUES

## Ajout d'images

- La balise d'image **<img>** est utilisées pour ajouter une image à une page Web.
- La balises d'image **<img>** n'a pas de balises de fermeture.
- La balise d'image **<img>** doit comporter les deux attributs suivants :
  - Src : Spécifie l'URL de l'image
  - Alt : Fournit une description textuelle de l'image qui décrit le image si nous ne pouvons pas la voir.

```
  

```

# GRAPHIQUES

## Ajout d'images

- Outre les attributs universels, la balise <img> a les attributs suivants
  - **Src** : spécifie l'URL de l'image.
  - **Alt** : fournit une description textuelle de l'image qui décrit le image si nous ne pouvons pas la voir.
  - **Sizes** : tailles d'image entre les points d'arrêt.
  - **Usemap** : nom de l'image cliquable à utiliser.
  - **Ismap** : indique si l'image est une image cliquable côté serveur.
  - **Srcset** : images à utiliser dans différentes situations (par exemple, écrans haute résolution, petits moniteurs, etc.).
  - **Width** : Spécifie la largeur de l'image.
  - **Height** : Spécifie la hauteur de l'image .



## NAVIGATION

### Ajout d'images : Exemple

```
<center>

</center>
<p> The inventor of the World Wide Web and one of Time Magazine's
'100 Most Important People of the 20th Century', Sir Tim Berners-Lee
is a scientist and academic whose visionary and innovative work has
transformed almost every aspect of our lives..
</p>
```



The inventor of the World Wide Web and one of Time Magazine's '100 Most Important People of the 20th Century', Sir Tim Berners-Lee is a scientist and academic whose visionary and innovative work has transformed almost every aspect of our lives..

# TABLEAUX

## Définition d'un tableau

- Un tableau est défini en HTML comme le regroupement d'un ensemble de lignes.
- Chaque ligne est composée d'un ensemble de cellules.
- Le début d'un tableau est défini par la balise **<table>**, la balise **</table>** le termine. Alors pour créer un tableau, nous devons d'abord définir son début et sa fin avec des balises :

```
<table>.....</table>
```

- L'attribut déconseillé **"border"** définit la largeur de la bordure entourant le tableau.

# TABLEAUX

## Définition d'un tableau

- La balise **<tr> (Table Row)** introduit une nouvelle ligne, elle est suivi d'un ou plusieurs éléments **<td>** (un pour chaque cellule sur cette ligne). A la fin de la ligne, vous utilisez une balise de fermeture **</tr>**.
- Chaque ligne du tableau aura exactement la largeur du tableau.
- Toutes les lignes d'un tableau possèdent exactement le même nombre de cellules que la ligne qui en comporte le plus.
- Une cellule est définie par les balises **<td> (Table Data)** et **</td>**.
- Une cellule peut contenir tout type d'information (image, lien, liste,...).
- La balise **<th> (Table Heading)** est utilisée juste comme l'élément **<td>**, mais son but est de représenter le titre pour une colonne ou une ligne.

# TABLEAUX

## Définition d'un tableau (Exemple)

- Dans cet exemple, nous créons un tableau avec une bordure de 5 pixels.

```
<table border="5">
  <tr>
    <th>Column 1</th>
    <th>Column 2</th>
    <th>Column 2</th>
  </tr>
  <tr>
    <td>Data 1</td>
    <td>Data 2</td>
    <td>Data 3</td>
  </tr>
</table>
```

Column 1	Column 2	Column 2
Data 1	Data 2	Data 3

# TABLEAUX

## **<caption> : légende d'un tableau**

- L'élément **<caption>** définit la légende (ou le titre) d'un tableau.
- L'élément **<caption>** devrait être le premier élément enfant de l'élément parent **<table>**.
- On peut utiliser les attributs universels sur cet élément.
- Ne pas utiliser **align** attribut. Il a été déprécié. L'élément **<caption>** devrait être mis en forme grâce au CSS. Afin d'obtenir des effets similaires à cet attribut, on utilisera les propriétés CSS **caption-side** et **text-align** (on vais voir ça plus tard en CSS).

## TABLEAUX

### <caption> : légende d'un tableau (Exemple)

```
<table border="2">
  <caption>Web programming: Grades</caption>
  <tr>
    <th>First name</th>
    <th>Last name</th>
    <th>Grade</th>
  </tr>
  <tr>
    <td>Mussab</td>
    <td>Zneika</td>
    <td>15/20</td>
  </tr>
</table>
```

First name	Last name	Grade
Mussab	Zneika	15/20

## TABLEAUX

**<table> Attributs** : Cet élément inclut les attributs universels. Les attributs listés ci-après sont désormais tous dépréciés.

- **align** : est utilisé pour spécifier l'alignement du tableau et de son contenu.
  - trois valeurs : “left”, “right” ou “center”. A éviter.
- **width** : la largeur du tableau, en pixels ou bien en pourcentage de la largeur de son environnement.
- **cellspacing** : espace (en pixels) qui existe entre chaque cellule du tableau.
- **cellpadding** : espace entre le bord de la cellule et son contenu en pixel.
- **border** : contrôle l’affichage des bordures du tableau. L’affichage de ces bordures est très différent d’un navigateur à l’autre. Pour désactiver l’affichage des bordures : **<table border="0">**
- **bgcolor** : la couleur de fond du tableau.

Nous le faisons généralement avec CSS.

# TABLEAUX

## <table> Attributs (Example)

```
<table border="2"
cellspacing="0"
cellpadding="10"
width="600 "
bgcolor="lightyellow">
```

```
<table border="2"
cellspacing="10"
cellpadding="0"
width="600 "
bgcolor="lightyellow">
```

```
<table border="2"
cellspacing="10"
cellpadding="10"
width="600 "
bgcolor="lightyellow">
```

cellspacing="0 " cellpadding="10"

First name	Last name	Grade
Mussab	Zneika	15/20

cellspacing="10 " cellpadding="0"

First name	Last name	Grade
Mussab	Zneika	15/20

cellspacing="10 " cellpadding="10"

First name	Last name	Grade
Mussab	Zneika	15/20



# TABLEAUX

## Fusion de cellules (Cell Spanning)

- Le navigateur fait l'hypothèse que :
  - toutes les lignes ont le même nombre de cellules.
  - toutes les colonnes ont le même nombre de lignes.
- Certains tableaux ont des cellules qui doivent s'étendre sur plusieurs lignes et colonnes.
- Nous pouvons créer des cellules « **extra-larges** » en ajoutant :
  - L'attribut **rowSpan** est utilisé pour créer une cellule qui s'étend sur plusieurs lignes.
  - l'attribut **colSpan** est utilisé pour créer une cellule qui s'étend sur plusieurs colonnes.

# TABLEAUX

## Fusion de cellules (Cell Spanning) : Exemple

```
<table border="2" cellspacing="2" bgcolor="lightyellow">

  <tr> <th>ID</th> <th>First Name</th> <th>Monday</th> <th>Tuesday</th>
  <th>Thursday</th> <th>Friday</th> </tr>
  <tr>
    <td>1</td>
    <td>John</td>
    <td colspan="2">work</td>
    <td>not work</td>
    <td rowspan="2">closed</td> </tr>
  <tr>
    <td>2</td>
    <td>Jane</td>
    <td>not work</td>
    <td colspan="2">work</td>
  </tr>

</table>
```

ID	First Name	Monday	Tuesday	Thursday	Friday
1	John	work		not work	closed
2	Jane	not work	work		

# TABLEAUX

## Utilisation des parties

- Les balises **<thead>**, **<tfoot>** et **<tbody>** sont utilisées pour regrouper les parties du tableau (En-tête, pied de page, corps).
- Notez que **<tfoot>** DOIT apparaître après **<tbody>**.
- Ce sont aussi des éléments logiques (sémantiques) plutôt que physiques.

# TABLEAUX

## Utilisation des parties (Exemple)

```
<table>
  <thead>
    <tr> <th>First name</th> <th>Last name</th> </tr>
  </thead>
  <tbody>
    <tr> <td>Mussab</td> <td>Zneika</td> </tr>
    <tr> <td>Alex</td> <td>Zeno</td> </tr>
  </tbody>
  <tfoot>
    <tr> <td colspan="2" align="center"> Footer</td> </tr>
  </tfoot>
</table>
```

First name	Last name
Mussab	Zneika
Alex	Zeno
Footer	

# TABLEAUX

## Regroupement de colonnes

- La balise **<col>** apparaît en tant qu'enfant de la balise **<table>** ou **<colgroup>** et décrit le formatage de la colonne entière.
- La balise **<colgroup>** permet d'appliquer une mise en forme à plusieurs colonnes.
- Par exemple, nous pouvons utiliser ces balises pour définir la mise en forme des cellules entières d'une ou plusieurs colonnes, en une étape au lieu de définir la mise en forme de chaque cellule séparément.

# TABLEAUX

## Regroupement de colonnes

Dans cet exemple, nous définissons la couleur d'arrière-plan de la première colonne en bleu clair, et la couleur d'arrière-plan des deux dernières colonnes en jaune.

```
<table border="1" cellspacing="0" cellpadding="0" width="600">  
  <colgroup>  
    <col style="background-color:lightblue">  
    <col span="2" style="background-color:yellow">  
  </colgroup>
```

id	First name	Last name
1	Mussab	Zneika
2	Alex	Zeno

## SUPPORT MÉDIA

### Vidéo

- L'élément **<video>** intègre un contenu vidéo dans un document.
- Les fichiers vidéo que nous souhaitons utiliser doivent être au format OGG, qui est un format vidéo open source.
- Nous définissons notre vidéo de manière très similaire aux autres éléments que nous prenons dans notre page.
- Une vidéo initiale qui est simplement lue lorsque la page est chargée peut être complétée en ajoutant les éléments suivants :

```
<video src="video.ogg" width="400" height="222">Ici la  
description alternative</video>
```

# SUPPORT MÉDIA

## Vidéo Attributs

- L'attribut **controls** pour ajouter les boutons « Lecture », « Pause » et « la barre de défilement ».
- L'attribut **preload="auto"** permet de spécifier au navigateur de débiter le téléchargement de la vidéo tout de suite.
- L'attribut **autoplay="true"** permet de lancer la lecture automatiquement.
- L'attribut **poster="image.jpg"** permet d'indiquer une image à afficher par défaut dans l'espace réservé par la vidéo, avant que la lecture de celle-ci ne soit lancée.

```
<video controls width="400" height="222"  
src="20141001_184503.mp4" preload="auto" poster="index.jpg">
```



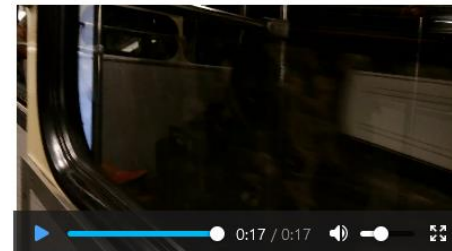


# SUPPORT MÉDIA

## Vidéo

- Les navigateurs ne prennent pas en charge l'ensemble des formats vidéo, et il est possible de fournir plusieurs sources grâce à des éléments **<source>**. Le navigateur utilisera la première ressource dont il connaît le format.

```
<video controls>  
<source src="maVideo.mp4" type="video/mp4">  
<source src="maVideo.webm" type="video/webm">  
<p>Votre navigateur ne prend pas en charge les  
vidéos HTML5</p>  
</video>
```



# SUPPORT MÉDIA

## Audio

- Les fichiers audio sont en fait identiques à l'approche vidéo (fichiers .ogg). A l'exception de la possibilité de définir la largeur et la hauteur.
- Tout ce que nous devons changer est notre vocabulaire, en utilisant le mot audio au lieu de vidéo dans notre tag.
- Il n'y a pas de codecs qui fonctionnent dans tous les navigateurs.
- Pour faire fonctionner votre document audio ou vidéo sur toutes les plateformes, vous devrez encoder le même vidéo en de multiples codecs.

```
<audio controls="controls">  
<source src="song.mp3" type="audio/mpeg" />  
<source src="song.ogg" type="audio/ogg" />  
</audio>
```

# LES FORMULAIRES

## Introduction aux formulaires

- Traditionnellement, "**formulaire**" fait référence à un document imprimé qui contient des espaces pour remplir certaines informations.
- HTML emprunte le concept de formulaire pour faire référence à différents éléments qui vous permettent de collecter des informations auprès des visiteurs de nos sites.
- Les formulaires sont partout autour de nous (login, formulaire de contact, sign up, etc...).



amazon.fr

Ouvrir une session

Quelle est votre adresse e-mail ?

Votre adresse e-mail est:

Disposez-vous d'un mot de passe Amazon.fr ?

☐ Vous êtes un nouveau client.  
(vous créerez un mot de passe plus tard)

☒ Vous êtes déjà client Amazon ?  
Votre mot de passe est :

☐ Maintenir ma session ouverte. [Détails](#)

[Continuer \(serveur sécurisé\)](#)

[Vous avez oublié votre mot de passe ?](#)

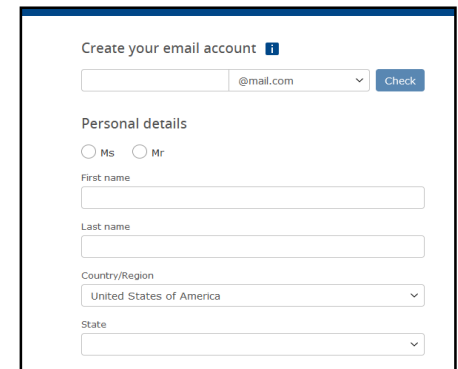
[Vous avez changé d'adresse e-mail depuis votre dernière commande ?](#)



Illustration of a person sitting at a desk with a computer, with the Google logo and binary code in the background.

Recherche Google

J'ai de la chance



Create your email account

@mail.com

Personal details

☐ Ms ☐ Mr

First name

Last name

Country/Region

State

Date of birth

# LES FORMULAIRES

## Introduction aux formulaires

- Le formulaire est utilisé pour envoyer les données que l'utilisateur saisira au serveur qui les traitera.
- Comment fonctionnent les formulaires :
  1. Un utilisateur remplit un formulaire puis il appuie sur un bouton de soumission pour soumettre les informations au serveur.
  2. Le nom de chaque champ de formulaire est envoyé au serveur avec la valeur saisie ou sélectionnée par l'utilisateur.
  3. Le serveur traite les informations à l'aide d'un langage de programmation tel que **PHP, ASP.net, JSP (Java Server Pages)** ou **Python**.
- Les actions et les champs que vous autorisez dans votre formulaire déterminent ce que l'utilisateur est autorisé à faire et quelles informations il est autorisé à voir.

# LES FORMULAIRES

## La balise Formulaire <Form>

- Pour créer une section de formulaire, nous fournissons au formulaire un **nom**, un **identifiant**, une **action** et une **méthode**. Un exemple avec des attributs vides ressemble à ceci :

```
<form action="xxxx" method="xxxx"> </form>
```

- Le **nom** et l'**identifiant** de notre formulaire sont la façon dont nous nous y référerons dans notre code lors de l'interaction avec lui en utilisant CSS, JavaScript ou d'autres langages.
- L'**action** est l'endroit où la page doit envoyer les informations (et où le navigateur ira lorsque nous appuierons sur **envoyer**).
- Notre **méthode** est la façon dont nous enverrons nos informations, en utilisant **GET** ou **POST**.

## LES FORMULAIRES

### Les attributs de formulaire :

Attribut	Description
<b>action</b>	Sa valeur est l'URL spécifiant où envoyer les données lorsque le bouton Soumettre est cliqué.
<b>method</b>	Les formulaires peuvent être envoyés en utilisant l'un des deux méthodes : <ul style="list-style-type: none"><li>→ "get" : informations passées dans la barre d'adresse</li><li>→ "post" : envoie les données dans le corps de la requête</li></ul>
<b>target</b>	Indique où ouvrir la page envoyée à la suite de la demande <ul style="list-style-type: none"><li>→ target =_blank signifie ouvrir dans une nouvelle fenêtre</li><li>→ target =_top signifie utiliser la même fenêtre</li></ul>
<b>id</b>	la valeur est utilisée pour identifier la forme distinctement de autres éléments de la page
<b>name</b>	Le nom du formulaire

# LES FORMULAIRES

## Method (GET)

- L'envoi des données à l'aide de la méthode **GET**, place tous les champs du formulaire par nom et valeur (appelée paire clé et valeur) dans la barre d'adresse. Ceci allonge notre URL en ajoutant chaque élément à l'adresse de la page de réception.

<https://www.google.com/search?q=cergy+université>

- L'avantage d'utiliser la méthode **GET** est que les paramètres de l'URL peuvent être enregistrés avec l'adresse du site Web. Cela permet de mettre une requête de recherche en marque-page et de la récupérer plus tard.
- Bien que bénéfique, il existe deux cas dans lesquels **nous ne voulons PAS** utiliser **GET** :
  - nous ne voulons pas que l'utilisateur voit ce qui a été envoyé (comme des mots de passe ou des informations confidentielles) ;
  - nous voulons envoyer beaucoup d'informations.

# LES FORMULAIRES

## Method (POST)

- Les données publiées sont envoyées du navigateur au serveur en arrière-plan, lorsque le client et le serveur commencent à communiquer.
- Les données sont envoyées dans les en-têtes de la communication, et ne sont pas visibles pour l'utilisateur final.
- Les pages mises en signet avec la méthode de publication n'auront plus accès aux informations par la suite, et ces informations sont perdues si l'utilisateur quitte la page.
- La manière dont les données sont utilisées et les valeurs renvoyés nous amènent aux langages de script.



# LES FORMULAIRES

## GET VS POST

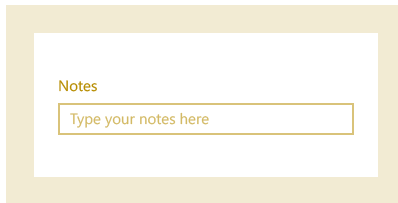
	GET	POST
Visibilité	Visible pour l'utilisateur dans le champ d'adresse	Invisible pour l'utilisateur
Marque-page et historique de navigation	Les paramètres de l'URL sont stockés en même temps que l'URL	L'URL est enregistrée sans paramètres URL
Cache et fichier log du serveur	Les paramètres de l'URL sont stockés sans chiffrement	Les paramètres de l'URL ne sont pas enregistrés automatiquement
Comportement lors de l'actualisation du navigateur / Bouton « précédent »	Les paramètres de l'URL ne sont pas envoyés à nouveau	Le navigateur avertit que les données du formulaire doivent être renvoyées
Type de données	Caractères ASCII uniquement.	Caractères ASCII mais également données binaires
Longueur des données	Limitée - longueur maximale de l'URL à 2 048 caractères	Illimitée

# LES FORMULAIRES

## Types de contrôles de formulaire

- Il existe plusieurs types de contrôles de formulaire.

### Text Box



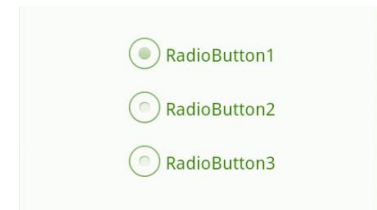
A screenshot of a web form titled "Notes". It features a text input field with the placeholder text "Type your notes here". The form is enclosed in a light yellow border.

### File upload



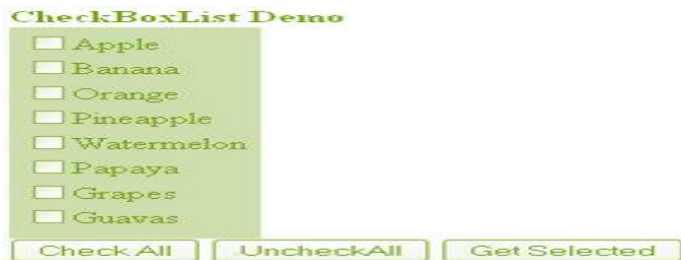
A screenshot of a file upload control. It includes an "Upload" button and a "Browse..." button.

### Radio Button



A screenshot of three radio buttons arranged vertically. The first button is selected and is labeled "RadioButton1". The second button is labeled "RadioButton2" and the third is labeled "RadioButton3".

### Check Box



A screenshot of a "CheckBoxList Demo". It shows a list of fruits with checkboxes: Apple, Banana, Orange, Pineapple, Watermelon, Papaya, Grapes, and Guavas. Below the list are three buttons: "Check All", "Uncheck All", and "Get Selected".

### Password



A screenshot of a password input field, represented by a series of dots.

### Submit buttons



A screenshot of a green "SUBSCRIBE" button.

# LES FORMULAIRES

## La balise <input>

- La balise **<input>** est utilisée pour créer plusieurs formulaires différents.
- Le **type** d'attribut indique de quel type d'élément il s'agit.
- la **valeur** de l'attribut type peut être :
  - text
  - checkbox
  - radio
  - password
  - hidden
  - submit
  - reset
  - button
  - file
  - image

# LES FORMULAIRES

## La balise <input>

- La balise **<input>** est utilisée pour créer plusieurs formulaires différents.
- Le **type** d'attribut indique de quel type d'élément il s'agit.
- La valeur de l'attribut **type** peut être :
  - text, checkbox, radio, password, hidden, submit, reset, button, file, image
- Attributs communs de la balise **<input>** :
  - **name** : le nom de l'élément, chaque **<input>** élément doit avoir un nom pour que l'on peut récupérer sa valeur. Ce nom n'est pas visible sur la page.
  - **value** : la valeur de l'élément ; utilisé de différentes manières pour différents valeurs de type.
  - **readonly** : la valeur ne peut pas être modifiée.
  - **disabled** : l'utilisateur ne peut rien faire avec cet élément.

# LES FORMULAIRES

## Zone de texte (Textbox)

- Lorsque l'attribut **type** a du **text** comme valeur, le navigateur crée une zone de texte sur une seule ligne.

```
<input type="text" name="text1" value="enter your name"/>
```

- La balise input est auto-fermante.
- L'attribut **maxlength** définit le nombre de caractères maximum acceptés dans le champ. Par exemple, si nous demandons un code postal en France, l'attribut maxlength pourrait avoir une valeur de 5.

# LES FORMULAIRES

## Le Label

- Chaque contrôle de formulaire doit avoir son propre élément **<label>** qui décrit aux utilisateurs la fonctionnalité du champ.
- Il peut être utilisé de deux manières :
  - Enroulé autour de la description textuelle et de l'élément de contrôle de formulaire.

```
<label>First name:  
<input type="text" name="fname" />  
</label>
```

- Séparé du contrôle de formulaire. Dans ce cas nous utilisons le **for** pour indiquer au contrôle de formulaire qu'il s'agit d'une étiquette.

```
<label for="lname">Last name:</label>  
  
<input type="text" id="lname" />
```

# LES FORMULAIRES

## ► Premier exemple :

```
<form action= "proce.php" method= "post">  
  <label for="fname">First name:</label>  
  <input type="text" id= "fname" name="fname"/>  
  <label for="lname">Last name:</label>  
  <input type="text" id="lname" name="lname"/>  
  <label for="age">Age:</label>  
  <input type="text" id="age" name="age"/>  
</form>
```

First name:

Last name:

Age:

# LES FORMULAIRES

## Zone de texte multiligne <textarea>

- > L'élément **<textarea>** est utilisé pour créer une zone de texte multiligne.
- > Une balise qu'il faut ouvrir et fermer.
- > Pas de value= " ", le texte par défaut se met entre les deux balises ouvrante et fermante.
- > L'attribut **cols** indique la largeur de la zone de texte.
- > L'attribut **rows** indique combien de lignes la zone de texte doit occuper verticalement.

```
<textarea name="comments" cols="50" rows="5">  
Enter your comments...  
</textarea>
```



# LES FORMULAIRES

## Champ de mot de passe (Password input)

- La balise `<input>` avec une valeur de type `"password"` est utilisée pour créer une zone de texte dont ses caractères sont masqués.
- Même si le mot de passe est masqué à l'écran, les données d'un champ `"mot de passe"` ne sont pas nécessairement envoyées de manière sécurisée au serveur.
- Il peut également avoir les attributs `size` et `maxlength` comme la zone de texte d'une seule ligne.

```
<input type="password" name="password" size="20" maxlength="15"/>
```

# LES FORMULAIRES

## Les boutons radio (input type="radio")

- Les boutons radio permettent aux utilisateurs de sélectionner **une seule option** parmi plusieurs.
- La valeur est importante : c'est elle qui est renvoyée au serveur.
- La valeur de chacun des boutons dans un groupe devrait être différente.
- Les boutons radio d'un groupe doivent avoir le même nom.
- L'attribut "**checked**" est utilisé pour indiquer quelle valeur doit être sélectionnée lors du chargement de la page.
- Un seul bouton radio dans un groupe doit utiliser l'attribut "**checked**".

```
<p>Please select your favorite Drink: <br />
  <input type="radio" name="drink" value="tea" checked="checked" /> Tea
  <input type="radio" name="drink" value="coffee" /> coffee
  <input type="radio" name="drink" value="milk" /> Milk
</p>
```

Please select your favorite Drink:  
☒ Tea ☐ coffee ☐ Milk

# LES FORMULAIRES

## Les cases à cocher (input type="checkbox")

- Les cases à cocher permettent aux utilisateurs de sélectionner(et désélectionnez) **un ou plusieurs options**.
- L'attribut value indique la valeur envoyée au serveur si cela est la case cochée.
- L'attribut "**checked**" est utilisé pour indiquer quelle valeur doit être sélectionnée lors du chargement de la page.

```
<p>Please select your favorite social media applications: <br/>
<input type="checkbox" name="media" value="facebook"/>Facebook
<input type="checkbox" name="media" value="twitter" checked="checked"/>twitter
<input type="checkbox" name="media" value="whatsapp"/>whatsapp
</p>
```

Please select your favorite social media applications:  
☐ Facebook ☒ twitter ☐ whatsapp

## LES FORMULAIRES

### La liste déroulante (Drop Down List)

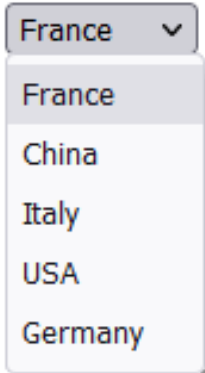
- La balise **<select>** est utilisée pour créer une liste déroulante.
- La balise **<select>** contient au moins deux **<option>** éléments.
- La balise **<option>** est utilisée pour spécifier les options que le l'utilisateur peut choisir.
- L'attribut **value** de l'élément **<option>** indique le valeur qui est envoyée au serveur.
- L'attribut **"selected"** est utilisé pour indiquer l'option qui doit être sélectionné lors du chargement de la page.

# LES FORMULAIRES

## La liste déroulante (Drop Down List) : Exemple

```
<form action="page1.php">
  <p>Select you country?</p>
  <select name="country">
    <option value="France">France</option>
    <option value="China">China</option>
    <option value="Italy">Italy</option>
    <option value="USA">USA</option>
    <option value="Germany">Germany</option>
  </select>
</form>
```

Select your country



France ▼

France

China

Italy

USA

Germany

Si l'utilisateur ne sélectionne pas d'option, alors la première option (France dans notre exemple) sera envoyée au serveur comme valeur de contrôle.

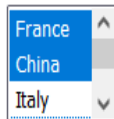
# LES FORMULAIRES

## La liste déroulante (Drop Down List)

- Les attributs «**size**» et «**multiple**» de la balise **<select>** peut transformer une liste déroulante, en une liste d'options multiples qui affiche plusieurs options à la fois.

```
<form action="page1.php">
  <p> You can select more than one option by holding down control.)</p>
  <select name="vistedcountries" size="3" multiple="multiple">
    <option value="France">France</option>
    <option value="China">China</option>
    <option value="Italy">Italy</option>
    <option value="USA">USA</option>
    <option value="Germany">Germany</option>
  </select>
</form>
```

You can select more than one option by holding down control on a PC or command key on a Mac while selecting different options.)



# LES FORMULAIRES

## La liste déroulante (Drop Down List)

- La balise **<optgroup>**, utilisé dans un formulaire, permet de créer un groupe d'options parmi lesquels on peut choisir dans un élément **<select>**.

```
<select name="city" id="city">
  <optgroup label="France">
    <option value="Paris">Paris</option>
    <option value="Lyon">Lyon</option>
    <option value="Marseille">Marseille</option>
  </optgroup>
  <optgroup label="German">
    <option value="Berlin">Berlin</option>
    <option value="Hamburg">Hamburg</option>
  </optgroup>
</select>
```

Choose city:

- France**
  - Paris
  - Lyon
  - Marseille
- German**
  - Berlin
  - Hamburg

## LES FORMULAIRES

### Zone d'entrée de fichier (<input type="file">)

- La zone de saisie de fichier aide les utilisateurs à **uploader leurs fichiers** vers le serveur.
- Utiliser les attributs suivants :

Attribut	Description
accept	Nous permet de définir les types de fichier qui doivent être acceptés.
files	Un objet Filelist qui liste les fichiers choisis.
capture	La source à utiliser pour capturer des images ou des vidéos.
multiple	Un attribut booléen qui, lorsqu'il est présent, indique que plusieurs fichiers peuvent être sélectionnés.



# LES FORMULAIRES

## Zone d'entrée de fichier (<input type="file">) : Exemples

```
<!--we can select only one file with all extensions -->  
<input type="file" id="myfile" name="myfile"/>
```

```
<!--we can select multiple files with all extensions -->  
<input type="file" id="myfile" name="myfile" multiple />
```

```
<!--we can select multiple files with word and pdf  
formats --> <input type="file" id="myfile" name="myfile"  
multiple accept=".doc,.docx,.pdf"/>
```

```
<!--we can select multiple files of all the formats of  
image, audio and video -->  
<input type="file" id="myfile" name="myfile" multiple  
accept="image/*, audio/*, video/*"/>
```

# LES FORMULAIRES

## Bouton d'envoi (Submit Button)

- Le bouton d'envoi `<input type="submit">` est utilisé pour envoyer les informations saisies dans les éléments de contrôle du formulaire au serveur.
- L'attribut **value** est obligatoire et permet de donner le «titre» du bouton.
- Les boutons de soumission peuvent être affichés de différents manières selon les navigateurs.
- CSS peut être utilisé pour contrôler l'apparence des boutons de soumission.

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

# LES FORMULAIRES

## Bouton de reset (Reset Button)

- Le bouton de reset `<input type= "reset">` est utilisé pour remettre à zéro toutes les informations saisies dans les éléments de contrôle du formulaire.
- L'attribut **value** permet de donner le «titre» du bouton.
- Les boutons de soumission peuvent être affichés de différents manières selon les navigateurs.
- CSS peut être utilisé pour contrôler l'apparence des boutons de reset.

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

## LES FORMULAIRES

### Bouton image (Image Button)

- Le `<input type="image">` est utilisé pour ajouter une image pour le bouton d'envoi.
- Nous pouvons utiliser les attribues **src**, **width**, **height**, et **alt** tels que nous les utilisons avec les balise `<img>`.

```
First name:<input type="text" id= "fname" name="fname"/><br>
Last name:<input type="text" id= "lname" name="lname"/><br>
<input type="image" src="submitb.jpg" width="100" height="50">
```

First name:

Last name:



## LES FORMULAIRES

### Bouton

- Un bouton est similaire au bouton d'envoi, mais contrairement aux autres styles de saisie, il peut inclure du texte ou une image.
- Sa mise en forme par défaut lui donne un aspect de bouton biseauté.
- Trois types : **button**, **reset**, **submit**

```
<button type="button">Click Here!</button><br>
```

```
<button type="submit"> </button> <br>
```

```
<button type="reset">Reset all!</button><br>
```



# LES FORMULAIRES

## Champ caché (Hidden input)

- On peut cacher un champ avec `<input type="hidden">`.
- Les données dans les contrôles cachés ne sont pas affichées sur le page, mais nous pouvons les voir si nous utilisons l'option «Afficher la source».
- Les données sont dans l'attribut value.

```
<input type="hidden" name="currentpage" value="orderpage">
```

# LES FORMULAIRES

## Champs de forme HTML5

- Bon nombre des nouveaux éléments HTML5 que nous examinons nous aideront également dans nos tâches de validation lorsque les utilisateurs remplissent des formulaires.
- Ces entrées tenteront de valider et/ou de limiter l'entrée de l'utilisateur aux seules données valides. En faisant cela immédiatement, nous créons une meilleure expérience à la fois pour l'utilisateur et le programmeur.
- Traditionnellement, la validation devait être effectuée lors de l'envoi des données au serveur, ce qui entraînait le rechargement de la page en cas d'erreurs.
- L'autre approche populaire consistait à effectuer une validation à l'aide de JavaScript côté client (en évitant le rechargement), mais la validation devrait toujours être répétée sur le serveur au cas où l'utilisateur final aurait désactivé JavaScript.
- Certains des types d'entrée les plus utiles sont les suivants :
- **<input type="url">** Tentera de formater le texte de l'utilisateur dans un lien approprié, ou affichera une erreur.
- **<input type="email">** S'assurera qu'un e-mail saisi est dans le bon format ou affichera une erreur.

# LES FORMULAIRES

## Champs de forme HTML5

- **<input type="email">** S'assurera qu'un e-mail saisi est dans le bon format ou affichera une erreur.

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

- Nous pouvons également créer une entrée qui limite les valeurs à une plage fixe et incrémente les limites, que nous devons auparavant afficher à l'utilisateur sur la page, puis valider après la saisie.

```
age: <input type="range" name="age"  
min="18" max="62" value="30" step="1">
```

- Ces limites sur une plage (affichées sous forme de curseur) sont également valables sur un champ numérique (affichées sous forme de flèches) :

```
age: <input type="number" name="age"  
min="18" max="62" value="30"  
step="1">
```

First name:   
Last name:   
email:   
**Submit**

Veuillez saisir une adresse électronique valide.

age:

age:



# LES FORMULAIRES

## Champs de forme HTML5

- HTML5 introduit également une multitude de contrôles de calendrier et d'heure.
- Nous pouvons spécifier une date, une semaine ou un mois ainsi qu'une heure, un jour et une heure et un jour et une heure locaux.
- Chacun de ces champs limitera l'entrée de l'utilisateur aux champs valides pour ce type.

**Birthday:** `<input type="date" name="date" value="2021-11-08">`

Birthday:

<

novembre 2021

>

lun.	mar.	mer.	jeu.	ven.	sam.	dim.
25	26	27	28	29	30	31
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5

On peut utiliser les attributs **min** et **max** afin de restreindre les dates qui peuvent être saisies par l'utilisateur. La date que nous pouvons sélectionner doit être comprise entre la date en valeur min et la date en valeur max.

# LES FORMULAIRES

## Champs de forme HTML5

### > Options de calendrier :

- > Le `<input type="month">` crée un contrôle de mois et d'année où l'utilisateur peut entrer le format de date comme "AAAA-MM".
- > On peut utiliser les attributs `min` et `max` afin de restreindre les dates qui peuvent être saisies par l'utilisateur.

```
<input type="month" name="month" value="2021-07"
min="2010-08" max="2021-08">
```



Ce champ n'est implémenté que sur Chrome, Opera et Edge sur les ordinateurs. Pour les autres navigateurs qui ne supportent pas ce champ, le champ sera transformé en une simple Zone de Texte.

# LES FORMULAIRES

## Champs de forme HTML5

- Options de calendrier :
  - Le `<input type="week">` définit un contrôle de semaine et d'année.
  - L'apparence de ce champ est différente selon les navigateurs.

```
Start Week: <input type="week" name="week" value="2021-W12" min="2010-W11" max="2021-W44">
```

Start Week: Semaine 12, 2021 

mars 2021  

Semaine	lu	ma	me	je	ve	sa	di
9	1	2	3	4	5	6	7
10	8	9	10	11	12	13	14
11	15	16	17	18	19	20	21
12	22	23	24	25	26	27	28
13	29	30	31	1	2	3	4
14	5	6	7	8	9	10	11

[Effacer](#) [Cette semaine](#)

# LES FORMULAIRES

## Champs de forme HTML5

- Options de temps :
  - L'apparence de ces champs est différente selon les navigateurs.

```
Time: <input type="time" name="time" value="14:30">
```

```
Time <input type="datetime-local" name="time2">
```

Time:

Submi

14	30
15	31
16	32
17	33
18	34
19	35
20	36

Time:

Submi

novembre 2021

lu	ma	me	je	ve	sa	di	22	26
1	2	3	4	5	6	7	23	27
8	9	10	11	12	13	14	00	28
15	16	17	18	19	20	21	01	29
22	23	24	25	26	27	28	02	30
29	30	1	2	3	4	5	03	31
6	7	8	9	10	11	12	04	32

Effacer Aujourd'hui

# LES FORMULAIRES

## Champs de forme HTML5

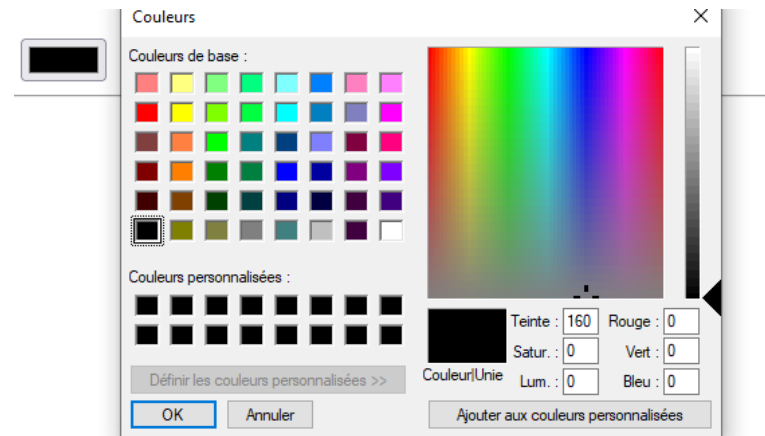
### > Plus des Champs

> `<input type="color">`

> `<input type="tel">`

> `<input type="search">`

> `<input type="url">`



# LES FORMULAIRES

## Balises de jeu de champs et de légende

- La balise **<fieldset>** est utilisée pour regrouper les contrôles liés dans un formulaire qui dessine un cadre autour des contrôles du formulaire lié.
- L'attribut **"form\_id"** spécifie à quelle forme le "fieldset" appartient.
- La balise **<legend>** représente une légende pour le contenu de son élément parent **<fieldset>**.
- L'attribut **"disabled"** spécifie qu'un groupe d'éléments de formulaire liés doit être désactivé.

# LES FORMULAIRES

## Balises de jeu de champs et de légende : Example:

```
<form action="page1.php">
  <fieldset>
    <legend>Your information:</legend>
    <label for="fname">First name:</label>
    <input type="text" id="fname" name="fname"><br><br>
    <label for="lname">Last name:</label>
    <input type="text" id="lname" name="lname"><br><br>
    <input type="image" src="submitb.jpg" width="50" height="25" >
  </fieldset>
</form>
```

Your information:

First name:

Last name:

## SOURCES D'INFORMATIONS

- <https://developer.mozilla.org/fr/docs/Learn/HTML/Tables>
- <https://developer.mozilla.org/en-US/docs/Learn/Forms>
- [https://www.w3schools.com/html/html\\_forms.asp](https://www.w3schools.com/html/html_forms.asp)
- [https://www.w3schools.com/html/html\\_tables.asp](https://www.w3schools.com/html/html_tables.asp)
- MENDEZ, MICHAEL. *THE MISSING LINK: AN INTRODUCTION TO WEB DEVELOPMENT AND PROGRAMMING*. 2014, CH14.
- MENDEZ, MICHAEL. *THE MISSING LINK: AN INTRODUCTION TO WEB DEVELOPMENT AND PROGRAMMING*. 2014, CH13
- MENDEZ, MICHAEL. *THE MISSING LINK: AN INTRODUCTION TO WEB DEVELOPMENT AND PROGRAMMING*. 2014, CH11,12
- MENDEZ, MICHAEL. *THE MISSING LINK: AN INTRODUCTION TO WEB DEVELOPMENT AND PROGRAMMING*. 2014, CH9,10