

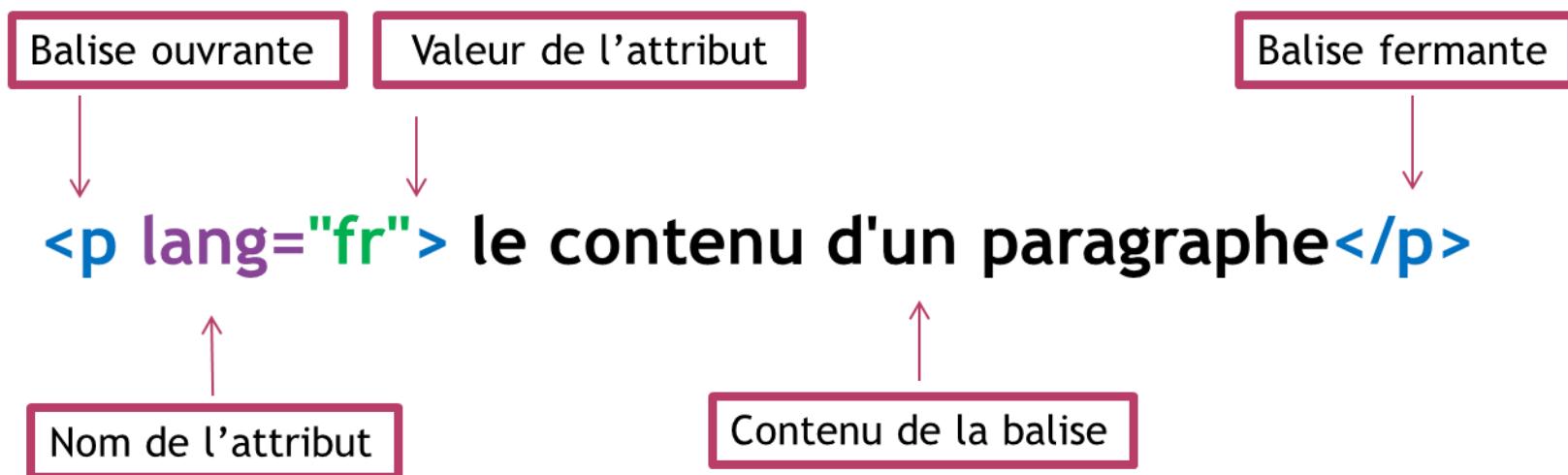
HTML

QU'EST CE QUE LE HTML ?

- HTML (HyperText Markup Language) est un langage de structuration de contenu.
- Langage **interprété et non compilé** fonctionnant généralement via des interpréteurs spéciaux appelés navigateurs.
- Langage basé sur le balisage de contenu.
- HTML contient plus de 100 balises qui fournissent au navigateur des indications sur le sens d'un élément, son interprétation ou son affichage.
- Le html est insensible à la casse.

BALISE & ÉLÉMENT HTML

- Les balises constituent l'élément de base du langage HTML, elles respectent une syntaxe **simple et stricte**.
- Un élément HTML se compose d'une balise ouvrante, d'un contenu textuel et d'une balise fermante



Note :

- Une balise fermante doit avoir **le même nom** que la balise ouvrante correspondante.
- De plus, une balise fermante doit contenir une barre oblique entre le chevron initial et le nom de la balise.

BALISE & ÉLÉMENT HTML

- La rigidité des conventions pour la fermeture des balises est utile afin de pouvoir imbriquer différents éléments les uns dans les autres. (première ouverte - dernière fermée)

```
<ol><li>Item</ol></li>
```



```
<ol><li>Item</li></ol>
```



- Le symbole '<' correspondant à l'ouverture d'une balise en HTML, il n'est pas possible d'utiliser ce symbole comme un caractère normal.
- Il est recommandé de remplacer les caractères spéciaux par leur code HTML.

> → >	& → &	&nbs; → espace
< → <	" → "	

Exemple: Pour écrire un paragraphe contenant "1 < 2" en HTML, je dois alors écrire <p> 1 < 2 </p>.

ATTRIBUTS

- Tous les éléments HTML peuvent avoir des attributs
- Les attributs fournissent des informations supplémentaires sur un élément
- Les attributs sont toujours spécifiés dans la balise de début, et se présentent généralement sous forme de paires nom / valeur comme: nom = "valeur "

Note:

- La norme HTML5 ne nécessite pas que les noms des attributs soit en minuscules,
-> cependant le W3C recommande l'utilisation des minuscules.
- La norme HTML5 ne nécessite pas de guillemets autour des valeurs d'attribut,
-> cependant le W3C recommande l'utilisation des guillemets.

ATTRIBUTS

Les guillemets simples ou doubles?

Les **guillemets doubles** autour des valeurs d'attribut sont les plus courants en HTML, mais des **guillemets simples** peuvent également être utilisés.

Note : Dans certaines situations, lorsque la valeur d'attribut elle-même contient des guillemets doubles, il est nécessaire d'utiliser des guillemets simples:

```
<p title='John "ShotGun" Nelson'>
```

Ou bien:

```
<p title="John 'ShotGun' Nelson">
```

LES BALISES VIDES

- Un élément HTML est composé de texte entre deux balises. Cependant, **certaines balises ne contiennent pas de texte.**

- Exemple

- la balise **
**

- la balise **<hr>**

- la balise ****.

- ``

LES ÉLÉMENTS SPÉCIAUX

En HTML, il existe **deux éléments spéciaux** qui n'ont pas de balise. Ces éléments sont essentiels pour tout document HTML.

Le *doctype*

C'est une déclaration formelle, pour «type de document», placée **au tout début d'un document HTML**.

La <!DOCTYPE> déclaration pour HTML5 est:

<!DOCTYPE html>

LES ÉLÉMENTS SPÉCIAUX

Les commentaires

Ils sont des éléments très particuliers. Ce sont des notes que vous pouvez utiliser pour annoter votre code HTML .

Les commentaires , en HTML, sont écrits avec du texte contenu entre `<!-- et -->`

`<!-- les commentaires ne sont pas affichés par le navigateur, mais ils peuvent aider à documenter votre code source HTML -->`

Note :

- Les commentaires courts doivent être écrits sur une seule ligne, comme ceci : `<!-- This is a comment -->`
- Les commentaires qui s'étendent sur plus d'une ligne doivent être écrits comme ceci:

`<!--`

`This is a long comment example. This is a long comment example.`
`This is a long comment example. This is a long comment example.`

`-->`

STRUCTURE DE LA PAGE HTML

```
<html>  
  
    <head>  
  
        <title> Titre de la page </title>  
  
    </head>  
  
    <body>  
  
        <h1> Ceci est un titre </h1>  
  
        <p> Ceci est un paragraphe. </p>  
  
        <p> Ceci est un autre paragraphe. </p>  
  
    </body>  
  
</html>
```

- L'élément `<html>` ou `<html></html>` : Cet élément est le contenant de tout le code de la page et est parfois connu comme l'élément racine.
- La racine contient obligatoirement deux fils **head** et **body**

STRUCTURE DE LA PAGE HTML

```
<html>  
  
<head>  
  
    <title> Titre de la page </title>  
  
</head>  
  
<body>  
  
    <h1> Ceci est un titre </h1>  
  
    <p> Ceci est un paragraphe. </p>  
  
    <p> Ceci est un autre paragraphe. </p>  
  
</body>  
  
</html>
```

- **<head> fournit des renseignements sur la page**
- Le contenu de l'élément **<title>** sera affiché dans la barre de titre du navigateur ou dans l'onglet de la page.
- Le contenu de l'élément **<body>** sera affiché dans un navigateur.

VALIDATEUR W3C

- Une bonne pratique consiste à **valider son code pour s'assurer qu'il est correct.**
- Le meilleur outil pour cela est **le validateur HTML du W3C.**

<http://validator.w3.org>

- C'est un outil en ligne qui analyse votre document HTML et vous indique si celui-ci respecte les spécifications XHTML.

Note :

une seule erreur peut en déclencher d'autres en cascades. Si le résultat de l'analyse comporte une myriade d'erreurs, commencez par résoudre la première, cela permettra éventuellement de résoudre des erreurs suivantes.

VALIDATEUR W3C

The screenshot shows the W3C Markup Validation Service interface. At the top, there's a blue header bar with the W3C logo and the text "Markup Validation Service" and "Check the markup (HTML, XHTML, ...) of Web documents". Below the header, there are three tabs: "Validate by URI", "Validate by File Upload", and "Validate by Direct Input", with "Validate by Direct Input" being the active tab. The main area is titled "Validate by direct input" and contains a text area for pasting markup. The markup shown is:

```
<!DOCTYPE html>
<html lang="fr">
<head>
<title> TITRE OBLIGATOIRE </title>
<meta charset="utf-8" />
</head>
<body>
</body>
</html>
```

Below the text area, there's a link "More Options" and a "Check" button.

STRUCTURE D'UN DOCUMENT HTML5 MINIMALE ET VALIDE

Une fichier HTML est un fichier texte avec l'extension .html ou .htm (Il n'y a aucune différence entre les deux extensions.)

Pour définir un document HTML valide, il faut impératif utiliser au moins une structure minimale qui a la forme suivante :

```
<!DOCTYPE html >

<html lang= "fr">

<head >
<title > TITRE OBLIGATOIRE </title >
</head >

<body >
</body >

</html >
```

L'ÉLÉMENT <HEAD>

Cet élément contient des informations relatives au document.

Les balises pour définir ces informations sont:

- ❖ **<title>** : titre de la page
- ❖ **<meta>**: encodage, type du document, . . . etc
- ❖ **<link>**: relier la page à d'autres ressources
- ❖ **<script>**: insérer une portion de script (langage de script à l'exemple du javascript)
- ❖ **<style>**: les informations sur le style du document

L'ÉLÉMENT <HEAD>

Balise <meta>

```
<head>
  <meta charset="UTF-8">
  <!-- Encodage des caractères -->
  <meta name="description" content="Free Web tutorials">
  <!-- Description de la page affichée par les moteurs de recherche -->
  <meta name="keywords" content="HTML, CSS, JavaScript">
  <!-- Des mots clés pour l'indexation de la page -->
  <meta name="author" content="John Doe">
  <!-- l'auteur du document -->
  <meta http-equiv="refresh" content="30">
  <!-- rafraîchissement de la page toutes les 30 secondes -->
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <!-- HTML5 a introduit une méthode permettant aux concepteurs Web de
prendre le contrôle de la fenêtre d'affichage, cette dernière varie en
fonction de l'appareil et sera plus petite sur un téléphone mobile que sur
un écran d'ordinateur. -->
</head>
```

L'ÉLÉMENT <HEAD>

Balise <link>

La balise **<link>** définit la **relation** entre le document actuel et une ressource externe.

Elle est souvent utilisée pour **créer un lien vers des feuilles de style externes**.

Exemple : **<link rel="stylesheet" href="mystyle.css">**

Balise <style>

La balise **<style>** est utilisé pour définir les informations de style pour **une seule page HTML**

QUELQUES BALISES HTML

LES TITRES

- Les titres sont définis par la balise **<hn>**
- Dans HTML, il y a six niveaux de titre : de h1 à h6
 - <h1> définit le titre le plus important.
 - <h6> définit le titre le moins important.

```
<h1>Titre1</h1>
<h2>Titre2</h2>
<h3>Titre3</h3>
<h4>Titre4</h4>
<h5>Titre5</h5>
<h6>Titre6</h6>
```



Titre1

Titre2

Titre3

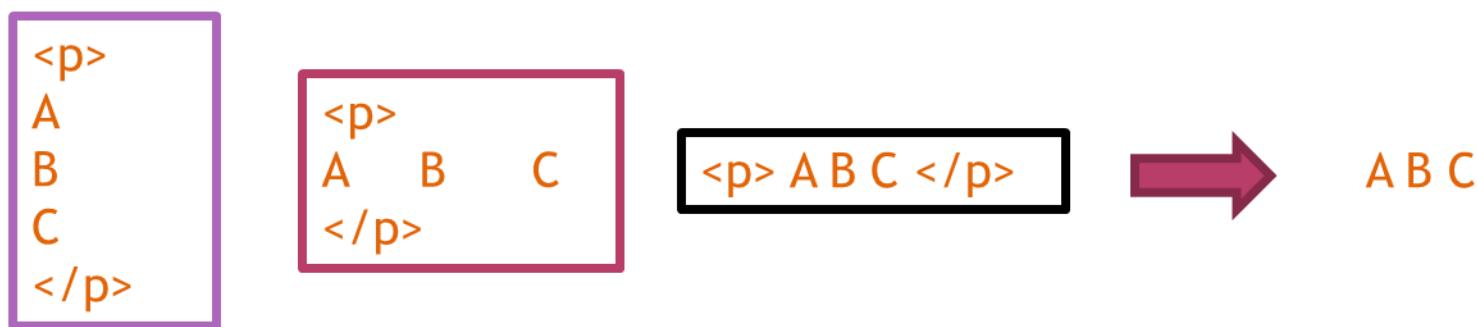
Titre4

Titre5

Titre6

PARAGRAPHES

- Les paragraphes HTML sont définis avec la balise `<p>`
- Un paragraphe commence toujours sur une nouvelle ligne et les navigateurs ajoutent automatiquement un espace blanc (une marge) avant et après un paragraphe.
- Avec HTML, vous ne pouvez pas modifier l'affichage en ajoutant des espaces ou des lignes supplémentaires dans votre code HTML.



RÈGLES HORIZONTALES HTML

L'élément `<hr>` définit une coupure thématique dans une page HTML et est le plus souvent affichée sous forme de règle horizontale.

SAUTS DE LIGNE HTML

L'élément `
` définit un saut de ligne
À utiliser `
` si vous souhaitez un saut de ligne (une nouvelle ligne) sans commencer un nouveau paragraphe.

LES LISTES EN HTML

Les listes HTML permettent aux développeurs Web de regrouper un ensemble d'éléments associés dans des listes.

- Une **liste non ordonnée** commence par la balise ****.
- Une **liste ordonnée** commence par la balise ****.
- **Chaque élément de la liste** commence par la balise ****.

```
<ul>
<li>Coffee</li>
<li>Tea</li>
<li>Milk</li>
</ul>
```



- Coffee
- Tea
- Milk

```
<ol>
<li>Coffee</li>
<li>Tea</li>
<li>Milk</li>
</ol>
```



1. Coffee
2. Tea
3. Milk

LES LISTES EN HTML

Les listes peuvent être imbriquées (liste dans la liste):

```
<ul>
  <li>Coffee</li>
  <li>Tea
    <ul>
      <li>Black tea</li>
      <li>Green tea</li>
    </ul>
  </li>
  <li>Milk</li>
</ul>
```



- Coffee
- Tea
 - Black tea
 - Green tea
- Milk

```
<ol>
  <li>Coffee</li>
  <li>Tea
    <ol>
      <li>Black tea</li>
      <li>Green tea</li>
    </ol>
  </li>
  <li>Milk</li>
</ol>
```



1. Coffee
2. Tea
 1. Black tea
 2. Green tea
3. Milk

Note : un élément de liste (``) peut contenir une nouvelle liste et d'autres éléments HTML, tels que des images et des liens, etc.

LES LISTES EN HTML

HTML prend également en charge **les listes de description** (une liste de termes, avec une description de chaque terme).

La balise **<dl>** définit la liste de description,
la balise **<dt>** définit le terme (nom)
et la balise **<dd>** décrit chaque terme:

```
<dl>
<dt>Coffee</dt>
<dd> - black hot drink</dd>
<dt>Milk</dt>
<dd> - white cold drink</dd>
</dl>
```



Coffee	- black hot drink
Milk	- white cold drink

LES LIENS EN HTML

- Les liens HTML sont définis avec la balise `<a>`
- La destination du lien est spécifiée dans l'attribut `href` :

```
<a href="https://www.google.com">Google</a>
```

Par défaut, un lien apparaîtra dans les navigateurs comme ceci:

- Un lien non visité est souligné et bleu
- Un lien visité est souligné et violet
- Un lien actif est souligné et rouge

Note:

Pour ouvrir un lien dans une nouvelle fenêtre, utiliser l'attribut «`target`» avec la valeur «`_blank`».

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

LES LIENS EN HTML

URL absolue \ relative

- Chaque ressource dans le web a un identifiant unique qui s'appelle URL
- Une ***URL absolue*** commence par le nom de domaine du site contenant la ressource
- Quand les pages appartiennent au même site, on peut faire usage des ***URLs relatives*** (sans la partie "https:// www")

```
<h2>Absolute URLs</h2>
<p><a href="https://www.w3.org/">W3C</a></p>
<p><a href="https://www.google.com/">Google</a></p>

<h2>Relative URLs</h2>
<p><a href="html_images.asp">HTML Images</a></p>
<p><a href="/css/default.asp">CSS Tutorial</a></p>
```



Absolute URLs

[W3C](#)

[Google](#)

Relative URLs

[HTML Images](#)

[CSS Tutorial](#)

Note: Les URLs relatives ne commencent pas par le nom de domaine mais indiquent une position par rapport à la page en cours.

LES LIENS EN HTML

Liens vers une partie de la page

- ◉ Les liens HTML peuvent être utilisés pour créer des signets afin que les lecteurs puissent accéder à des parties spécifiques d'une page Web.
 - ◉ Les signets peuvent être utiles si une page Web est très longue.
- Tout d'abord utilisez l'attribut **id** pour créer un signet:
- ```
<h2 id="C4">Chapitre 4</h2>
```
- Ensuite, ajoutez un lien vers le signet ("Aller au chapitre 4"), depuis la même page:
- ```
<a href="#C4"> Aller au chapitre 4 </a>
```
- Vous pouvez également ajouter un lien vers un signet sur une autre page (avant de mettre l'id de l'élément, on met l'adresse de la page) :
- ```
Jump to Chapter 4
```

# LES LIENS EN HTML

## *Liens vers des mails*

Un lien peut déclencher le programme client mail au lieu de pointer vers une ressource, il suffit d'utiliser «**mailto:**» dans la cible du lien.

```
<body>
Contactez-nous en cliquant ici
</body>
```



Contactez-nous en cliquant ici

## *Liens sous forme d'image*

Pour utiliser une image comme lien, placez simplement la balise **<img>** à l'intérieur de la balise **<a>**.

# LES IMAGES EN HTML

- Les images en HTML sont définies avec la balise **<img>**.

- L'attribut **src** indique le chemin vers l'image (ça peut être en local ou sur un autre site)
- L'attribut **alt** Spécifie un texte alternatif pour l'image si l'image ne se charge pas
- L'attribut **title** donne un intitulé à l'image

```

```

## Note :

- Il est préférable d'utiliser des URL **relatives**. si vous changez de domaine ils resteront toujours valable .
- Une bonne organisation consiste à mettre toutes les images dans un dossier séparé (appelé Images ou Img par exemple) .
- Les formats les plus utilisés du web sont JPEG, GIF et PNG

# LES TABLEAUX EN HTML

Les tableaux HTML permettent aux développeurs Web d'organiser les données en lignes et en colonnes.

- La balise **<table>** définit un tableau HTML.
- Chaque **ligne du tableau** est définie avec une balise **<tr>**,
- Chaque **en-tête de tableau** est défini avec une balise **<th>**,
- Chaque **donnée / cellule du tableau** est définie avec une balise **<td>**,
- Pour ajouter une légende à un tableau, utilisez la balise **<caption>**.

## Note:

- les éléments **<td>** sont les conteneurs de données de la table. Ils **peuvent contenir toutes sortes d'éléments HTML**: texte, images, listes, autres tableaux, etc.
- La balise **<caption>** doit être insérée immédiatement après la balise **<table>**.

# LES TABLEAUX EN HTML

```
<table style="width:40%">
 <caption>Mon premier tableau</caption>
 <tr>
 <th>A</th> <th>B</th> <th>C</th>
 </tr>
 <tr>
 <td>a1</td> <td>b1</td> <td>c1</td>
 </tr>
 <tr>
 <td>a2</td> <td>b2</td> <td>c2</td>
 </tr>
</table>
```



Mon premier tableau		
A	B	C
a1	b1	c1
a2	b2	c2

# LES TABLEAUX EN HTML

Pour qu'une cellule s'étende sur plusieurs colonnes,  
utilisez l'attribut **colspan**:

```
<table style="width:100%">
 <tr>
 <th>Name</th>
 <th colspan="2">Telephone</th>
 </tr>
 <tr>
 <td>Bill Gates</td>
 <td>55555555</td>
 <td>66666666</td>
 </tr>
</table>
```



Name	Telephone	
Bill Gates	55577854	55577855

# LES TABLEAUX EN HTML

Pour qu'une cellule s'étende sur plusieurs lignes,  
utilisez l' attribut **rowspan**:

```
<table style="width:100%">
<tr>
 <th>Name:</th>
 <td>Bill Gates</td>
</tr>
<tr>
 <th rowspan="2">Telephone:</th>
 <td>55577854</td>
</tr>
<tr>
 <td>55577855</td>
</tr>
</table>
```



Name:	Bill Gates
Telephone:	55577854
	55577855

# LES ÉLÉMENTS DE FORMATAGE HTML

Les éléments de mise en forme ont été conçus pour afficher des types de texte spéciaux:

**<b>** - définit le texte en gras, sans aucune importance supplémentaire.

```
<p>texte normal.</p>
```

```
<p>texte en gras.</p>
```



texte normal.

**texte en gras.**

**<i>** - définit le texte en italique. utilisée pour indiquer un terme technique, une phrase d'une autre langue, une pensée, un nom de navire, etc.

```
<p>Ce texte est normal.</p>
```

```
<p><i>Ce texte est en italique.</i></p>
```



Ce texte est normal.

*Ce texte est en italique.*

# LES ÉLÉMENTS DE FORMATAGE HTML

**<mark>** - définit le texte qui doit être marqué ou mis en évidence

```
<p>This is a <mark>new</mark> element in HTML 5.</p>
```



This is a **new** element in HTML 5.

**<ins>** - définit un texte qui a été inséré dans un document. Les navigateurs soulignent généralement le texte inséré

```
<p>Ma couleur préférée est le <ins>rouge</ins>.</p>
```



Ma couleur préférée est le rouge.

# LES ÉLÉMENTS DE FORMATAGE HTML

**<sub>** - définit le texte en indice. Le texte en indice peut être utilisé pour les formules chimiques, comme H<sub>2</sub>O.

```
<p>H₂O</p>
```



H<sub>2</sub>O

**<sup>** - définit le texte en exposant. Le texte en exposant peut être utilisé pour les notes de bas de page, comme HTML<sup>[1]</sup>.

```
<p>HTML^[1]</p>
```



HTML<sup>[1]</sup>

# L'ÉLÉMENT HTML <PRE>

- La balise `<pre>` définit le texte préformaté.
- Le texte à l'intérieur de la balise `<pre>` est affiché dans une police de largeur fixe (généralement Courier), et il préserve à la fois les espaces et les sauts de ligne:

```
<p>
A.
B.
C.
</p>
```



A. B. C.

```
<pre>
A.
B.
C.
</pre>
```



A.  
B.  
C.

# LES ABRÉVIATIONS EN HTML

- La balise `<abbr>` définit une abréviation ou un acronyme, comme "HTML", "CSS", "Mr.", "Dr.", etc.
- Le marquage des abréviations peut fournir des informations utiles aux navigateurs, aux systèmes de traduction et aux moteurs de recherche.

```
<p><abbr title="HyperText Markup Language">HTML</abbr> est le langage de balisage conçu pour représenter les pages web. </p>
```



HTML est le langage de balisage conçu pour représenter les pages web.

HyperText Markup Language

**Note:** utilisez l'attribut **title** pour afficher la description de l'abréviation / acronyme lorsque vous passez la souris sur l'élément.

# TYPES D'ÉLÉMENT EN HTML

Chaque élément HTML a une valeur d'affichage par défaut, en fonction du type d'élément dont il s'agit. Il existe deux valeurs d'affichage: **block et inline**.

éléments de type block	éléments de type inline
commence toujours sur une <b>nouvelle ligne</b>	ne commence pas sur une nouvelle ligne.
occupe toujours <b>toute la largeur disponible</b>	occupe que <b>la largeur nécessaire</b> .
<b>a une marge supérieure et inférieure</b>	<b>n'a pas une marge</b> supérieure et inférieure

**Note :**

un élément de type inline **ne peut pas contenir** un élément de type block.

# TYPES D'ÉLÉMENT EN HTML

## Les éléments de type block

<address>	<article>	<aside>	<blockquote>	<canvas>
<dd>	<div>	<dl>	<dt>	<fieldset>
<figcaption>	<figure>	<footer>	<form>	<h1>-<h6>
<header>	<hr>	<li>	<main>	<nav>
<noscript>	<ol>	<p>	<pre>	<section>
<table>	<tfoot>	<ul>	<video>	

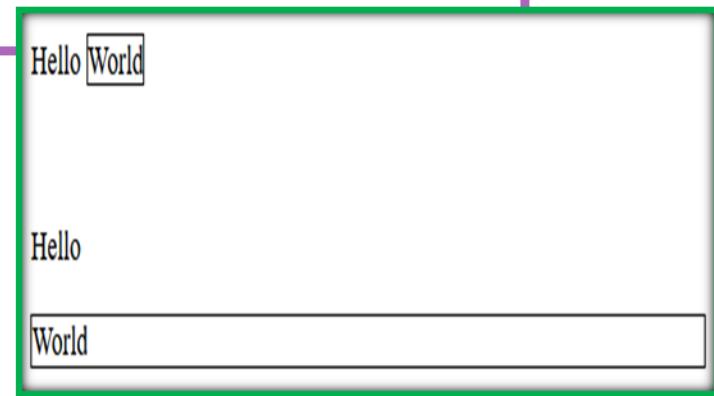
## Les éléments inline

<a>	<abbr>	<acronym>	<b>	<bdo>
<big>	 	<button>	<cite>	<code>
<dfn>	<em>	<i>	<img>	<input>
<kbd>	<label>	<map>	<object>	<output>
<q>	<samp>	<script>	<select>	<small>
<span>	<strong>	<sub>	<sup>	<textarea>
<time>	<tt>	<var>		

# DIV ET SPAN

```
<p> Hello
World </p>

<p> Hello <div style="border: 1px solid black">World</div>
</p>
```



- La balise **<div>** est souvent utilisé comme **conteneur pour d'autres éléments HTML**.
- La balise **<span>** est un conteneur inline utilisé pour **marquer une partie d'un texte ou une partie d'un document**.

# LES FORMULAIRES EN HTML

- Le web a très évolué de simple sites web vers de véritables applications web où les données sont le noyau de ces applications.
- Pour qu'une application soit utile, il ne suffit pas qu'elle affiche des données, il faut qu'elle donne à l'utilisateur de créer ou de mettre à jour ces données.
  - Les formulaires sont le moyen de collecte de données depuis les utilisateurs.
- En HTML, les formulaires apportent l'interactivité à une page web. Ils permettent aux utilisateurs d'introduire des données, de sélectionner et de valider des choix.

**Exemple :** connexion, inscription, création d'un produit, ajout d'un commentaire,...

# LES FORMULAIRES EN HTML

- La balise **<form>** est utilisé pour créer un formulaire HTML,
  - Tout ce qui est entre cette balise double fait partie du formulaire.
- Le langage HTML ne permet pas d'analyser ou de traiter ces informations. Ceci, nécessite l'utilisation d'un autre langage complémentaire tel que le langage PHP ou JAVASCRIPT.
- La balise **<form>** est un conteneur pour différents types d'éléments d'entrée, tels que: champs de texte, cases à cocher, boutons radio, boutons d'envoi, etc.
  - Un formulaire est composé de plusieurs *contrôles* et d'un *bouton d'envoi*.

# LES FORMULAIRES EN HTML

```
<form action="/action_page.php">
<label for="fname">Prénom:</label>

<input type="text" id="fname" name="fname"
value=Karim>

<label for="lname">Nom:</label>

<input type="text" id="lname" name="lname"
value=Mazi">

<input type="submit" value="Submit">
</form>
```

The diagram illustrates the process of submitting a form. On the left, a red-bordered box contains the HTML code for a form. An arrow points from this box to a blue-bordered box representing the form interface. This interface shows two text input fields: one for 'Prénom' containing 'Karim' and another for 'Nom' containing 'Mazi'. A 'Submit' button is also visible. A second arrow points from the form interface to a white box at the bottom containing the text 'Your input was received as:' followed by a text input field showing the submitted data: 'fname=Karim&lname=Mazi'.

Prénom:	Karim
Nom:	Mazi

Submit

Your input was received as:

fname=Karim&lname=Mazi

## Note :

- L'attribut **action** donne l'url de la page qui va recevoir les informations
- Notez que **chaque champ de saisie** doit avoir un attribut **name** à soumettre. Si l'attribut **name** est omis, la valeur du champ de saisie ne sera pas du tout envoyée.
- La balise **<input>** est l'élément de formulaire le plus utilisé.

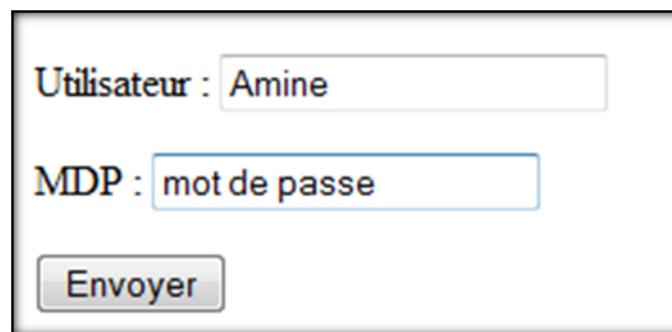
# LES FORMULAIRES EN HTML

- Une fois le bouton envoi appuyé, le navigateur formate les données du formulaire sous forme de paires *nom / valeur* puis les envoie au serveur -> Cette opération est appelée **sérialisation**.
- La sérialisation GET et POST est parmi les méthodes les plus populaires et les plus simples.
- La méthode GET:
  - Les données sont ajoutées à la fin de l'URL définie dans l'attribut action -> données envoyées clairement dans l'URL.
  - La méthode est idéale pour les très petits formulaires contenant des données pas sensibles.-> La longueur de l'URL ne doit pas dépasser 2048 caractères.
  - Les requêtes GET restent dans l'historique du navigateur
- La méthode POST:
  - les données sont transmises dans le corps de la requête HTTP -> données envoyées d'une façon transparente.
  - POST n'a pas de **limites de taille** et peut être utilisé pour envoyer de grandes quantités de données.
  - permet de : uploader un fichier, encapsuler des données sensibles, gérer les connexions cryptées.

# LES FORMULAIRES EN HTML

Soit le formulaire suivant :

```
<form action="url" method="get" >
<p>Utilisateur: <input type="text" name="login" /></p>
<p>MDP: <input type="password" name="mdp" /></p>
<input type="submit" value="Envoyer"/>
</form>
```



Utilisateur : Amine

MDP : mot de passe

Envoyer

L'appui sur « Envoyer » redirige vers l'URL suivante :

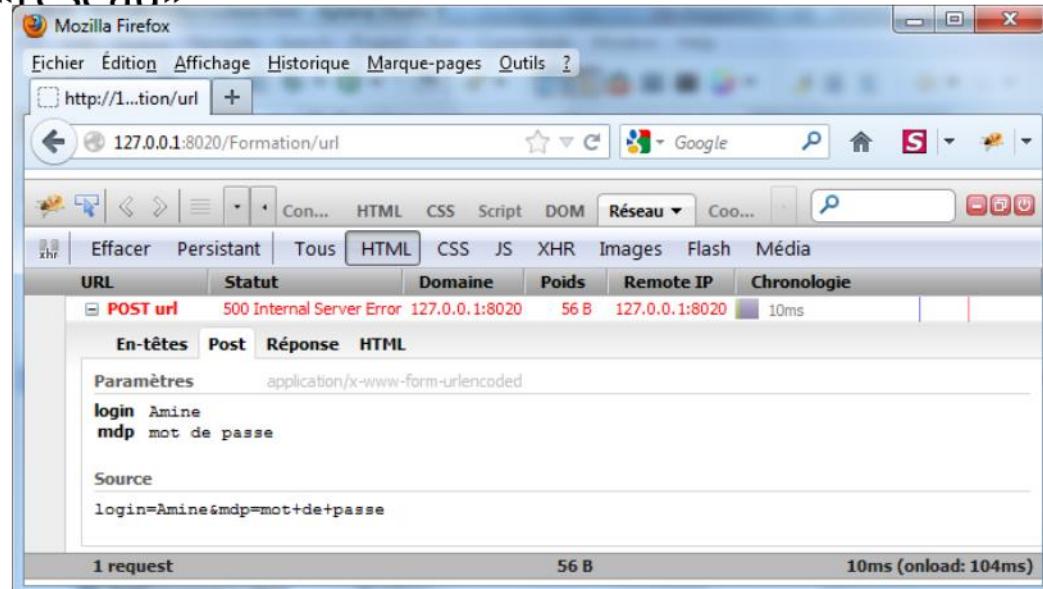
<http://127.0.0.1:8020/Formation/url?login=Amine&mdp=mot+de+passe>

# LES FORMULAIRES EN HTML

Soit le formulaire suivant :

```
<form action="url" method="post">
<p>Utilisateur : <input type="text" name="login" /></p>
<p>MDP : <input type="text" name="mdp" /></p>
<input type="submit" value="Envoyer"/>
</form>
```

- Ouvrir la fenêtre «inspecter l'élément» puis cliquez sur envoyer
- Examiner l'onglet «réseau»



# LES FORMULAIRES EN HTML

## La saisie de texte

La saisie de texte est effectuée avec la balise *input* avec l'attribut **type** ayant la valeur «**text**».

L'attribut **maxlength** définit le nombre maximum de caractères qu'un utilisateur peut entrer dans la zone de texte

```
<p>Nom : <input name="nomText" maxlength="30" type="text" /></p>
```



# LES FORMULAIRES EN HTML

## La saisie de mots de passe

La saisie de mots de passe est effectuée avec la balise *input* avec l'attribut **type** ayant la valeur «**password**»

L'attribut **maxlength** définit le nombre maximum de caractères qu'un utilisateur peut entrer dans la zone de mdp

```
<p>Login : <input name="loginText" maxlength="30" type="text" /></p>
<p>MDP : <input name="passText" maxlength="8" type="password" /></p>
```



Login : Amine

MDP :

# LES FORMULAIRES EN HTML

## La saisie de long texte

La saisie de long texte est effectuée avec la balise **textarea**

L'attribut **rows** définit le nombre de lignes visibles et l'attribut **cols**, le nombre de colonnes.

```
<p>Vos commentaires :</p>
<textarea name="comments" rows="5"></textarea>
```



Vos commentaires :

Entrez les commentaires ici

# LES FORMULAIRES EN HTML

## Boutons radio

Le bouton radio est effectué avec la balise *input* avec l'attribut *type* ayant la valeur «*radio*»

L'attribut *value* indique la valeur qui va être envoyée au serveur

```
<p>
Votre langage de programmation préféré :

<input name="langage" value="c#" type="radio" />C#
<input name="langage" value="java" type="radio"/>Java
<input name="langage" value="php" type="radio"/>PHP
</p>
```



Vos langage de programmation préféré :  
 C#  Java  PHP

# LES FORMULAIRES EN HTML

## Les cases à cocher

La case à cocher est effectuée avec la balise *input* avec l'attribut *type* ayant la valeur «**checkbox**»

L'attribut *value* indique la valeur qui va être envoyée au serveur

```
<p>
Vos marques préférées :

<input name="marque" value="microsoft" type="checkbox" />Microsoft
<input name="marque" value="apple" type="checkbox"/>Apple
<input name="marque" value="google" type="checkbox"/>Google
</p>
```



Vos marques préférées :

Microsoft  Apple  Google

# LES FORMULAIRES EN HTML

## Les listes déroulantes

- La liste déroulante est définie par la balise **select**,
- la balise **option** quant-à-elle définit les éléments de cette liste
- L'attribut **value** indique la valeur qui va être envoyée au serveur
- L'attribut **selected** indique l'élément sélectionné après le chargement de la page

```
<p> Votre wilaya d'origine:

<select name="wilaya">
<option value="16">Alger</option>
<option value="17">Djelfa</option>
<option value="18"
selected="selected">Jijel</option>
</select>
</p>
```



Votre wilaya d'origine:

Jijel ▾

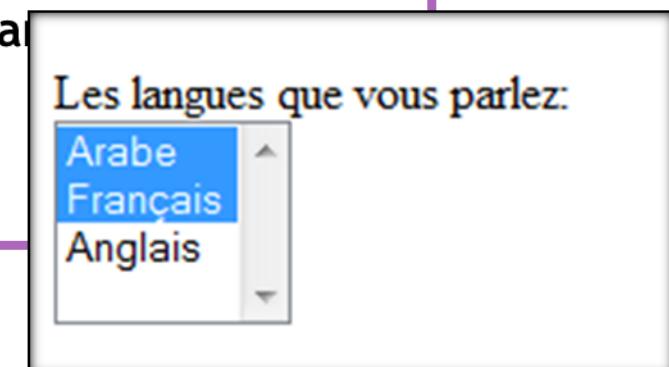
# LES FORMULAIRES EN HTML

## Les listes à sélection multiples

- La liste à sélection multiple est définie par la balise ***select***,
- la balise ***option*** quant-à-elle définit les éléments de cette liste.
- L'attribut ***size*** indique le nombre d'éléments visibles simultanément
- L'attribut ***multiple*** indique que l'utilisateur peut sélectionner plusieurs éléments.

```
<p> Les langues que vous parlez:

<select name="langue" multiple="multiple">
<option value="ar" selected="selected">Arabe</option>
<option value="fr" selected="selected">Français</option>
<option value="en" >Anglais</option>
</select>
</p>
```



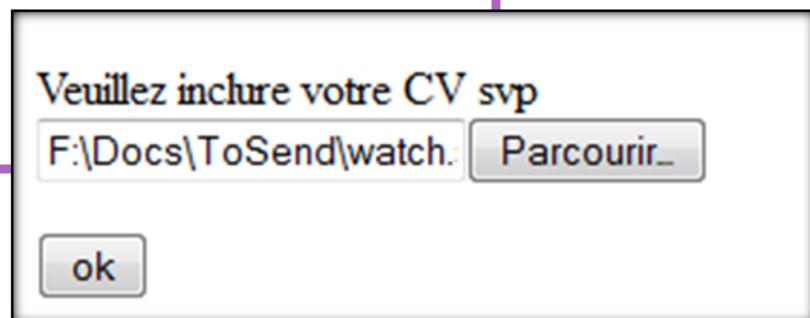
# LES FORMULAIRES EN HTML

## Uploader des fichiers

- L'upload de fichiers est effectué avec la balise *input* avec l'attribut **type** ayant la valeur «**file**»
- Le navigateur ajoute automatiquement le bouton «parcourir».
- Pour pouvoir faire des uploads, la méthode du formulaire doit être obligatoirement **POST** et enctype à **multipart/form-data**

```
<form method="post" enctype="multipart/form-data">
<p>
Veuillez inclure votre CV svp

<input type="file" name="cv" />
</p>
<input type="submit" value="ok" />
</form>
```



A screenshot of a web browser showing a file input field. The placeholder text "Veuillez inclure votre CV svp" is visible above the input field. The input field contains the text "F:\Docs\ToSend\watch.". To the right of the input field is a "Parcourir..." button. Below the input field is a large "ok" button.

# LES FORMULAIRES EN HTML

## Regrouper des éléments

- La balise **fieldset** groupe les éléments ayant un certain lien dans un formulaire
- La balise **legend** donne un intitulé au groupe (fieldset)

```
<form action="url" method="post" enctype="text/plain" >
<fieldset> <legend>Identité</legend>
<p>Nom : <input type="text" name="nom" /> </p>
</fieldset>
<fieldset> <legend>Marques</legend>
Vos marques préférées :
<input name="marque" value="microsoft" type="checkbox" id="cbMicrosoft" />
<label for="cbMicrosoft">Microsoft</label>
<input name="marque" value="apple" type="checkbox" id="cbApple"/>
<label for="cbApple">Apple</label>
<input name="marque" value="google" type="checkbox" id="cbGoogle"/>
<label for="cbGoogle">Google</label>
</fieldset>
</form>
```

Identité

Nom :

Marques

Vos marques préférées :  Microsoft  Apple  Google

# LES FORMULAIRES EN HTML

## Les boutons

- Le boutons est défini par la balise **button**
- Un bouton transforme son contenu en bouton

```
<form method="post">
<p> Nom : <input type="text" name="nom" /> </p>
<button value="send" >Envoyer</button>
</form>
```

Nom :

Envoyer

# LES FORMULAIRES EN HTML

## Les champs cachés

- Un champ caché est défini par la balise **input** avec l'attribut **type** ayant la valeur **hidden**
- Les champs cachés sont très importants car ils intègrent des informations qui ne doivent pas être visible à l'utilisateur mais qui doivent être remontées au serveur
- L'attribut **value** donne la valeur du champ caché. La seule façon de la voir est de voir le code source.

```
<form method="post">
<p> Nom : <input type="text" name="nom" />
<input type="hidden" name="id" value="55"/>
</p>
<button value="send" >Envoyer</button>
</form>
```

Nom :

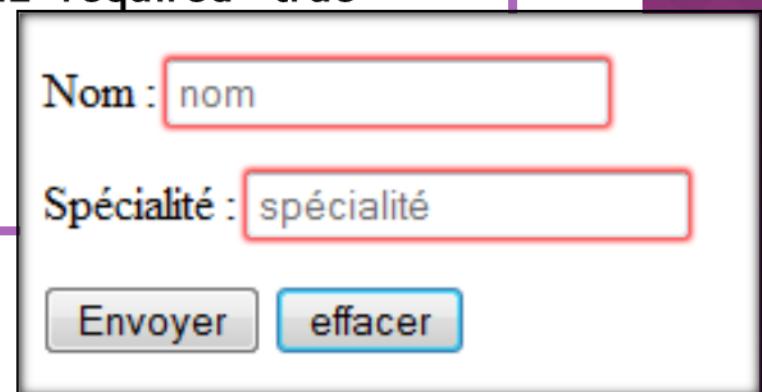
Envoyer

# LES FORMULAIRES EN HTML

## Effacer un formulaire

- On efface un formulaire en ajoutant un contrôle *input* avec le type égal à *reset*.

```
<form method="post" >
<p> Nom : <input type="text" name="login" required="required"
placeholder="nom"/></p>
<p> Spécialité : <input type="text" pattern="SI | ISIL" required="true"
placeholder="spécialité" /></p>
<input type="submit" value="Envoyer" />
<input type="reset" value="effacer" />
</form>
```



The screenshot shows a web page with a form. It contains two text input fields: one for 'Nom' with placeholder 'nom' and another for 'Spécialité' with placeholder 'spécialité'. Below the inputs are two buttons: a grey 'Envoyer' button and a blue 'effacer' button.

### Note : (Validation HTML5 des champs texte )

- L'attribut *required* indique que le champ est obligatoire
- L'attribut *pattern* définit une expression régulière à laquelle doit obéir la valeur du champ
- L'attribut *placeholder* donne une indication de la valeur à entrer

# LES FORMULAIRES EN HTML

## Entrée des champs numériques

- Utiliser la balise ***input*** avec l'attribut type à ***number***
- Les attributs ***min*** et ***max*** définissent l'intervalle de la valeur

```
<p> Nom :
<input type="text" name="login"
required="required"/>
</p>
<p> Age :
<input type="number" name="age"
required="true" min="18" max="35" />
</p>
```

The screenshot shows a web form with two fields. The first field is a text input labeled "Nom" containing the value "sfgf". The second field is a number input labeled "Age" containing the value "99". The "Age" input has a yellow border, indicating it is the active or selected field. Below the inputs is a button labeled "Envoyer".

# LES FORMULAIRES EN HTML

## Entrée des champs date

- Utiliser la balise ***input*** avec l'attribut type à ***date***
- Les attributs ***min*** et ***max*** définissent l'intervalle de la valeur

Nom :

D.Naissance :  ▼

```
<p> Nom :
<input type="text" name="login" required="required"/>
</p>
<p> D.Naissance:
<input type="date" name="dnaiss" required="true" />
</p>
```

# LES FORMULAIRES EN HTML

## Entrée de champs mail et URL

- Utiliser la balise ***input*** avec l'attribut type à ***date***
- Les attributs ***min*** et ***max*** définissent l'intervalle de la valeur

```
<p> Nom :
<input type="text" name="login" required="required"/></p>
<p> Mail :
<input type="email" name="mail" required="true" /></p>
<p> Site perso :
<input type="url" name="url" required="true" />
```

Nom : amine

Mail : amine

Site perso : mo

Envoyer

# IDENTIFICATION ET REGROUPEMENT DES ÉLÉMENTS

- L'attribut ***id*** est utilisé pour identifier un élément unique
- L'attribut ***class*** permet de désigner des éléments ayant des points en commun
- Les attributs ***id*** et ***class*** sont très utiles pour les utiliser dans CSS et JS
- Un élément peut appartenir à plusieurs classes (séparer les noms de classe par des espaces)
- Le nom de la classe est **sensible à la casse**.
- **Différents éléments HTML** peuvent pointer vers le **même nom de classe**.

```
<p id="p1" class="parAvecIndent"> 1er par </p>
<p id="p2"> 2ème par </p>
<p id="p3" class="parAvecIndent parEnGras"> 3eme par</p>
```