

# Conception de sites web dynamiques

***HTML – CSS – JAVASCRIPT – PHP  
BASE DE DONNÉES***

IF04U050

David Bouchet

*david.bouchet.paris5@gmail.com*

## ***HTML 5*** ***1<sup>re</sup> partie***



# Déroulement

- **Cours (12 séances)**

- Vendredi de 9 h 30 à 11 h

- **TP (12 séances)**

- Vendredi de 11 h 15 à 14 h 15
- Vendredi de 12 h 15 à 15 h 15
- Vendredi de 14 h 45 à 17 h 45

- **Supports :**

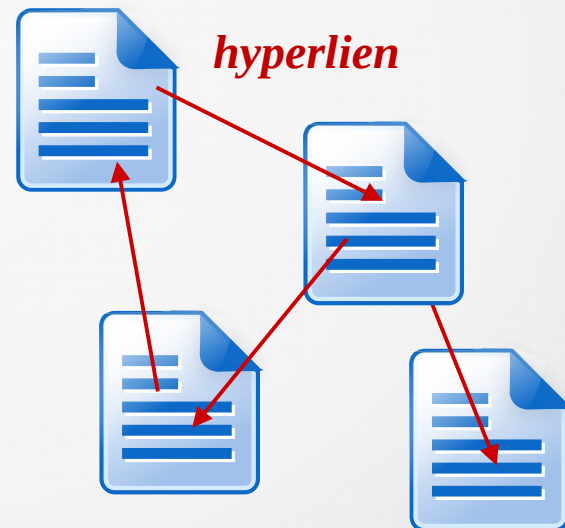
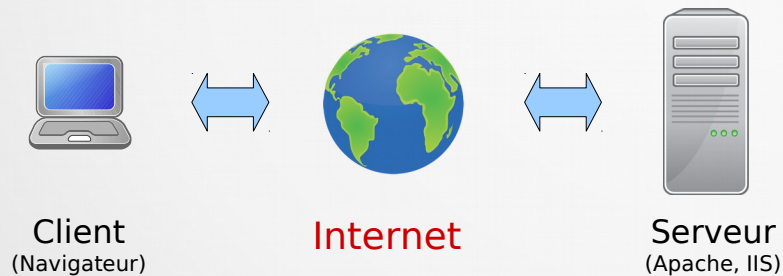
<http://www.debug-pro.com/paris5/web/index.html>

# Évaluation

- Un examen de contrôle continu
- Un examen terminal
- $\text{Note finale} = \text{MAX}(\text{Ex. Term}, \text{Moy}(\text{Ex.Term}, \text{CC}))$

# Qu'est-ce qu'un site Web ?

- Un site web est un ensemble de documents *hypertexte*
  - Hébergé sur un serveur (accessible via à une adresse web)
  - Consulté par des utilisateurs via un navigateur (Firefox, Chrome, etc.)



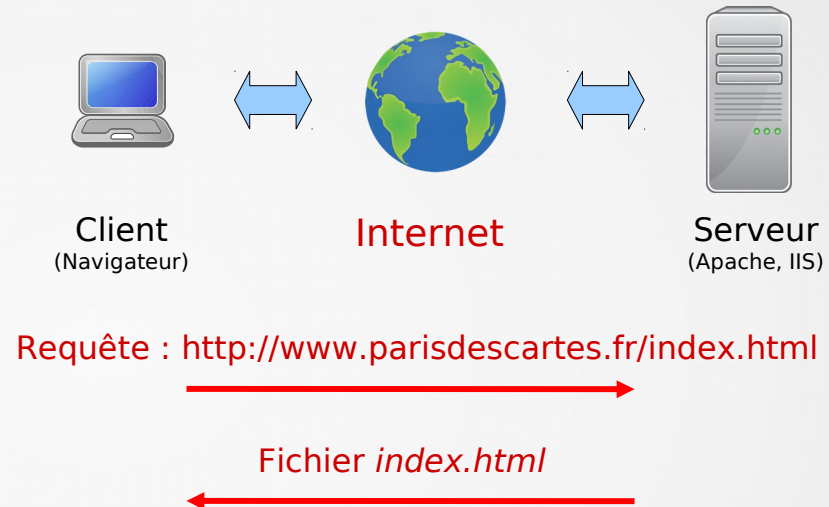
*Pages au format HTML*



# Communication Client-Serveur

## ■ Navigateurs

- Chrome
- Internet Explorer
- Mozilla / Firefox
- Safari
- Opera
- NetScape



- 1) Le **navigateur = client** qui demande une page web au serveur
- 2) Le **serveur** envoie un fichier HTML = fichier texte
- 3) Le **navigateur** interprète le fichier HTML

# Statique vs dynamique

- ◆ Un site web statique est constitué de pages Web (généralement **HTML** et **CSS**) déjà présentes sur le serveur web. Elles sont prêtes à être envoyées vers le client sans aucune modification.
- ◆ Un site web dynamique (côté serveur) génère des pages Web (généralement **HTML** et **CSS**) à la demande. Ces pages n'existaient pas avant la connexion du client.
- ◆ Une page Web peut également avoir un aspect dynamique côté client (même si le site est statique). Le langage **JavaScript** permet au navigateur de modifier une page Web directement sur le client (sans passer par le serveur).

# Programme du cours

- Structure d'une page web statique : **HTML 5**
- Mise en forme avec feuilles de styles : **CSS 3**
  - **CSS = Cascading Style Sheets**
- Programmation côté client : **JavaScript**
  - Interaction avec l'utilisateur côté client
- Programmation côté serveur : **PHP**
  - Sites web dynamiques côté serveur
  - Connexion à une base de données : **MySQL ou PostGresql**



# Adresse web – URL

**URL** : *Uniform Resource Locator*

**http://serveur.domaine.fr/chemin/page.extension**

↑  
Protocole de communication

↑  
adresse du serveur

↑  
Chemin du fichier  
sur le serveur

↑  
Ressource

## Exemples :

- <http://www.ratp.fr/informer/trafic/trafic.php>
- [file:///users/lipade2/mahe/tmp/test\\_html5.html](file:///users/lipade2/mahe/tmp/test_html5.html)
- <http://123.87.54.251/index.html>
- <https://www-projet.cea.fr/doku.php?id=wp2&do=media>

# Concepts fondamentaux du HTML

- **HTML** : *HyperText Markup Language*
  - Langage de description de pages web
  - Hypertexte : façon de relier des documents multimédias
  - Mise en forme décrite par des fichiers de style **CSS**
- Un document **HTML** est:
  - Un fichier texte, construit suivant une syntaxe précise
  - Structuré par des balises
  - Interprété (visualisé) par un **navigateur**

# Éléments, balises et attributs

**Trois termes à connaître en HTML :**

Élément  
Balise  
Attribut

# Éléments

Un élément permet de définir un type d'objet dans une page.

Quelques exemples :

- Élément *title* = Titre de la page
- Élément *p* = Paragraphe
- Élément *h1* = Titre de niveau 1
- Élément *h2* = Titre de niveau 2
- Élément *a* = Lien vers un autre emplacement (interne ou externe)
- Élément *img* = Image

# Balises (1)

- Les balises permettent de placer les éléments dans une page.  
`<p>`Ici se trouve un paragraphe`</p>`
- Les balises sont définies entre deux chevrons `<>`
- Le contenu d'un élément se place entre une **balise ouvrante** et une **balise fermante** :  
`<balise>`Contenu d'un élément`</balise>`
- Il existe également des balises orphelines :  
Par exemple :  
`<br>` Élément *br* (*break*) : retour à la ligne.  
`<input>` Élément *input* (*break*) : contrôle interactif.



## Balises (2)

- La plupart des éléments peuvent être imbriqués selon des règles bien définies :
  - `<h1><strong>Mon titre</strong></h1>`
  - ~~`<h1><strong>Mon titre</h1></strong>`~~
- Les minuscules/majuscules n'interviennent pas dans la définition des balises
  - Pas sensible à la casse

# Attributs

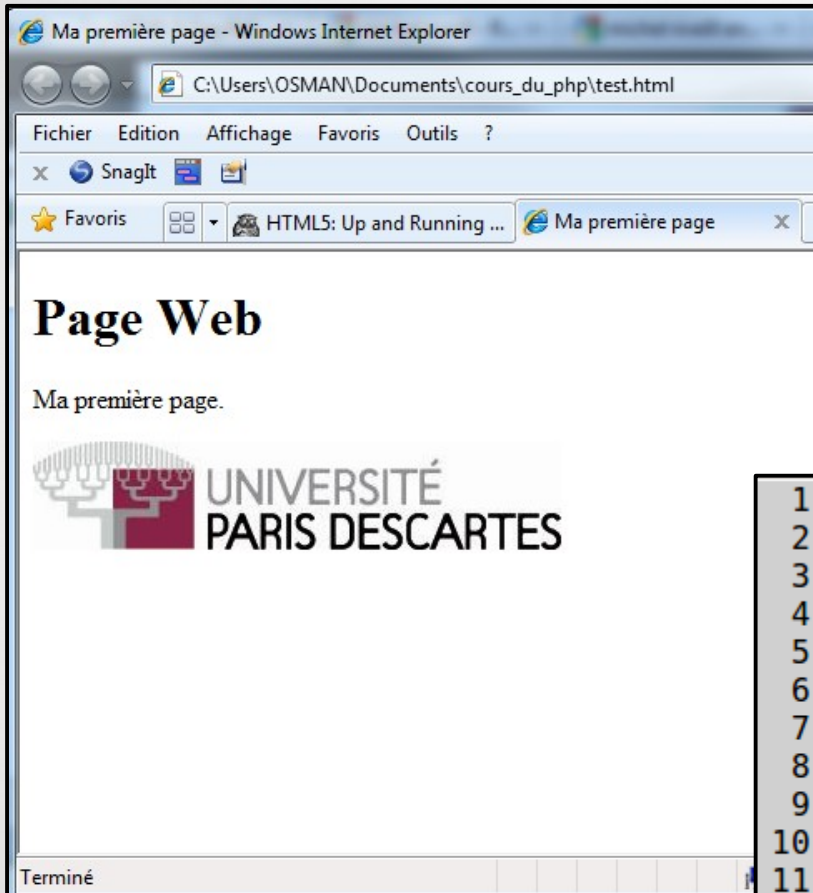
- Un attribut permet de donner des paramètres ou des indications à un élément. Un attribut se situe toujours dans une balise ouvrante :

```
<balise attribut="paramètre">Contenu de l'élément</balise>
```

- Un élément peut contenir plusieurs attributs.

```
<a href="http://www.xyz.fr" target="_blank">Video Index</a>
```

# Interprétation et affichage du HTML



interprété

Fichier HTML

brut

```
1  <!DOCTYPE html>
2  <html lang=fr>
3  <head>
4      <meta charset="utf-8">
5      <title>Ma première page</title>
6  </head>
7  <body>
8      <h1>Page Web</h1>
9      <p>Ma première page.</p>
10     
11 </body>
12 </html>
13
```

# Format d'un document HTML – *DOCTYPE*

## ■ **<!DOCTYPE html>**

- N'est pas une balise html
- *Document Type Declaration*
- Indication au navigateur sur la version de la page html
- Indique les règles d'écritures utilisées dans le fichier

## ■ En HTML 4.01

- `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">`

## ■ En XHTML 1.0

- `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">`



# Format d'un document HTML – *html, head, body*

- Tout document HTML commence par la balise `<html>` et finit par la balise `</html>`
- Tout document HTML contient
  1. Un en-tête: `<head>` et `</head>`
  2. Un corps: `<body>` et `</body>`



1	<code>&lt;!DOCTYPE html&gt;</code>	<code>&lt;!-- DOCTYPE --&gt;</code>
2	<code>&lt;html lang=fr&gt;</code>	<code>&lt;!-- Début du document HTML --&gt;</code>
3	<code>&lt;head&gt;</code>	<code>&lt;!-- Début de l'entête --&gt;</code>
4	<code>&lt;meta charset="utf-8"&gt;</code>	
5	<code>&lt;title&gt;Ma première page&lt;/title&gt;</code>	
6	<code>&lt;/head&gt;</code>	<code>&lt;!-- Fin de l'entête --&gt;</code>
7	<code>&lt;body&gt;</code>	<code>&lt;!-- Début du corps --&gt;</code>
8	<code>&lt;h1&gt;Page Web&lt;/h1&gt;</code>	
9	<code>&lt;p&gt;Ma première page.&lt;/p&gt;</code>	
10	<code>&lt;/body&gt;</code>	<code>&lt;!-- Fin du corps --&gt;</code>
11	<code>&lt;/html&gt;</code>	<code>&lt;!-- Fin du document HTML --&gt;</code>
12		



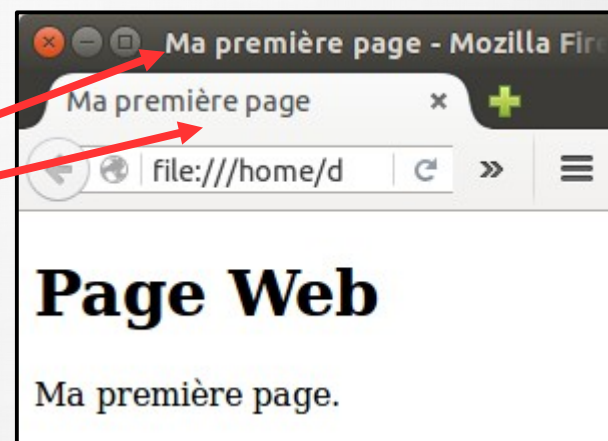
# En-tête <head>

- Délimité par les balises **<head>** et **</head>**
- Pas d'affichage dans le contenu du navigateur
- Contient des informations générales sur le document, toujours chargées avant le corps
  - Titre du document **<title>**
  - Informations sur le contenu du document **<meta>**
    - `<meta charset="UTF-8">`
    - `<meta name="keywords" content="html, elements, web, page, structure, code">`
  - Variables et fonctions des scripts JavaScript **<script>**
  - Les références aux feuilles de style **<link>**
  - Etc.

# Titre de la page web <title>

- Contenue dans l'en-tête du document
- Définit le titre du document, terminé par la balise **</title>**
- Le titre doit être court et explicite car :
  - Il apparaît en tant que titre de la fenêtre du navigateur ;
  - Il apparaît dans la liste des *bookmarks* ;
  - Il est utilisé par les moteurs de recherche.

```
1  <!DOCTYPE html>
2  <html lang=fr>
3    <head>
4      <meta charset="utf-8">
5      <title>Ma première page</title>
6    </head>
7    <body>
8      <h1>Page Web</h1>
9      <p>Ma première page.</p>
10   </body>
11 </html>
12
```



# Corps de la page web <body>

- Délimité par les balises **<body>** et **</body>**
- Contient les informations affichables du document
  - Texte du document
  - Image
  - Vidéo
  - Etc...

# Titres hiérarchiques

- Les balises **h1**, **h2**, ...**h6** permettent de baliser un paragraphe comme étant un titre d'un certain niveau
  - affectent la taille des caractères (taille décroissante)
  - Titre suivi d'un saut de ligne
  - **<hn>** ne doit pas être placé dans **<p>...</p>**

```
1  <!DOCTYPE html>
2  <html lang="fr">
3  <head>
4      <meta charset="utf-8">
5      <title>Ma premiere page</title>
6  </head>
7  <body>
8      <h1>Titre 1</h1>
9      <h2>Titre 2</h2>
10     <h3>Titre 3</h3>
11     <p>Du texte normal</p>
12 </body>
13 </html>
14
```





# Paragraphe

- Un paragraphe est délimité par les balises `<p></p>`
- Un début de paragraphe provoque :
  - Un passage à la ligne
  - Un décalage d'une hauteur d'environ une ligne

```
1  <!DOCTYPE html>
2  <html lang="fr">
3  <head>
4      <meta charset="utf-8">
5      <title>Ma première page</title>
6  </head>
7  <body>
8      <h1>Titre "h1"</h1>
9      <p>Texte sous le titre.</p>
10     <h2>Titre "h2"</h2>
11     <p>Texte sous le titre.</p>
12     <h3>Titre "h3"</h3>
13     <p>Texte sous le titre.</p>
14 </body>
15 </html>
16
```

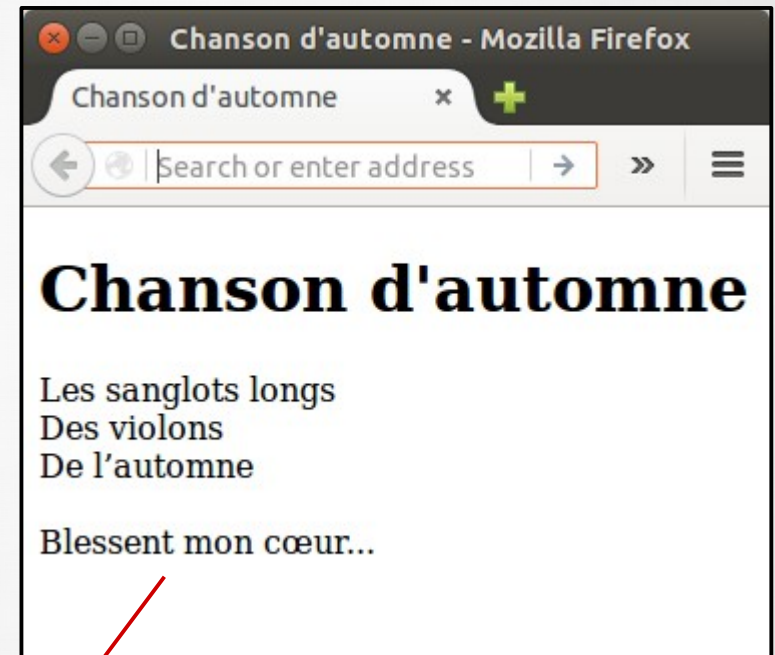




# Balise <br>

## ■ Retour à la ligne

```
1  <!DOCTYPE html>
2  <html lang="fr">
3  <head>
4      <meta charset="utf-8">
5      <title>Chanson d'automne</title>
6  </head>
7  <body>
8      <h1>Chanson d'automne</h1>
9      Les sanglots longs<br>
10     Des violons<br>
11     De l'automne<br>
12     <br>
13     Blessent
14     mon cœur...
15 </body>
16 </html>
17
```



Sans la balise <br> le retour à la ligne n'est pas pris en compte.  
Il est simplement remplacé par un espace.

# Commentaires

- Tout texte commençant par « `<!--` » et se terminant par « `-->` » est un commentaire pour l'interpréteur HTML.

```
1  <!DOCTYPE html>
2  <html lang="fr">
3  <head>
4      <meta charset="utf-8">
5      <title>Chanson d'automne</title>
6  </head>
7  <body>
8      <!-- Ceci est un commentaire -->
9  </body>
10 </html>
11
```

# Structurer / Baliser un texte (1)

HTML permet de structurer sémantiquement le contenu d'une page web à l'aide de balises.

Chacune de ces balises possède une mise en forme par défaut qui peut être modifiée par les styles CSS.

L'ajout d'une balise dans un texte ne doit pas se faire en fonction de la mise en forme, mais en fonction de la sémantique du texte.

HTML seul permet la mise en forme d'un texte, mais cela doit être évité à tout prix. La mise en forme doit se faire à l'aide de styles CSS.

## Structurer / Baliser un texte (2)

**Les principales balises de structuration sémantique sont données dans les *slides* suivantes.**

Ces balises seront accompagnées d'un texte d'exemple.

Dans un souci pédagogique, les balises seront affichées en rouge, mais il est évident que dans un fonctionnement normal du navigateur les balises n'apparaissent pas.

La mise en forme sera celle d'un navigateur par défaut, elle peut varier légèrement d'un navigateur à l'autre.



## Structurer / Baliser un texte (3)

**<b>** Texte en gras par défaut (aucun sens sémantique) **</b>**

**<i>** Texte en italique par défaut (aucun sens sémantique). **</i>**

**<strong>** Renforcement sémantique (gras par défaut). **</strong>**

**<em>** Emphase sémantique (italique par défaut). **</em>**

**<mark>** Repérer un passage particulier (surligné par défaut). **</mark>**

**<u>** Annotation non textuelle (souligné par défaut). **</u>**

**<small>** Afficher le texte plus petit. **</small>**



## Structurer / Baliser un texte (4)

Afficher un exposant : X **<sup>** 10 **</sup>**

Afficher un indice : X **<sub>** 10 **</sub>**

**<s>** ~~Partie du texte qui n'est plus pertinente (barré par défaut).~~ **</s>**

Utiliser les éléments *del* et *ins* pour remplacer explicitement du texte. :  
Mon alcool préféré est le **<del>**whisky**</del>** **<ins>**rhum**</ins>**

Utiliser l'élément *kbd* pour afficher la saisie d'un utilisateur :  
Appuyez sur **<kbd>**Ctrl + c**</kbd>** pour copier un texte.

## Structurer / Baliser un texte (5)

Utiliser l'élément *code* pour afficher du code :

**<code>**

```
counter = 5 + 10;  
echo "Le résultat est ";  
echo $counter;  
return 0;
```

**</code>**

Utiliser l'élément *samp* pour afficher la sortie d'un programme :

**<samp>** Le résultat est 15 **</samp>**

Utiliser l'élément *var* pour afficher une variable :

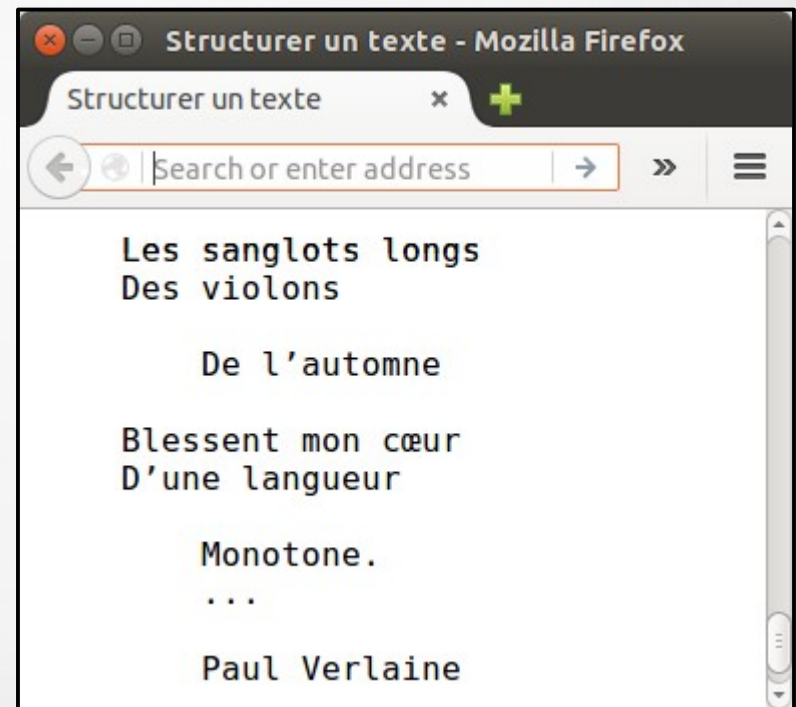
La variable **<var>** *counter* **</var>** est égale à 15.

## Structurer / Baliser un texte (6)

**Utiliser l'élément *pre* pour préserver les espaces et les retours à la ligne.**

### Fichier HTML

```
114 <pre>
115     Les sanglots longs
116     Des violons
117         De l'automne
118
119     Blessent mon cœur
120     D'une langueur
121
122         Monotone.
123         ...
124
125         Paul Verlaine
126 </pre>
127
128
```



# Jeux de caractères

## ■ Charset

- Jeux de caractères ;
- « Ã© » ou « Ã§ » : problème d'**encodage des caractères** ;
- le navigateur ne sait pas quel encodage utiliser pour afficher la page ;
- **ISO-8859-1** (parfois appelé latin1) : tous les caractères du français ;
- **ISO-8859-15** (parfois appelé latin9) : ajoute le symbole « euro » et le « o dans l'e » ;
- **UTF-8** : toutes les langues, du français au japonais en passant par l'arabe ;
- Vérifier dans les paramètres de configuration de votre éditeur (Dreamweaver, Notepad++, Bluefish, etc.).

## ■ HTML5

- `<meta charset="UTF-8">`



# Les caractères spéciaux

Result	Description	Entity Name	Entity Number
	non-breaking space	&nbsp;	&#160;
<	less than	&lt;	&#60;
>	greater than	&gt;	&#62;
&	ampersand	&amp;	&#38;
¢	cent	&cent;	&#162;
£	pound	&pound;	&#163;
¥	yen	&yen;	&#165;
€	euro	&euro;	&#8364;
§	section	&sect;	&#167;
©	copyright	&copy;	&#169;
®	registered trademark	&reg;	&#174;
™	trademark	&trade;	&#8482;

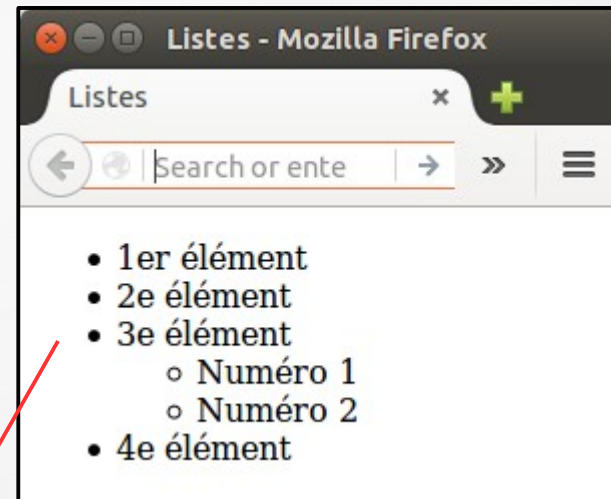
<http://www.w3.org/TR/html5/syntax.html#named-character-references>

<http://dev.w3.org/html5/html-author/charref>

# Listes non ordonnées

- Les balises
  - `<ul>` : liste à puces (*unordered list*)
  - `<li>` : élément de liste

```
9      <ul>
10      <li>1er élément</li>
11      <li>2e élément</li>
12      <li>3e élément
13          <ul>
14              <li>Numéro 1</li>
15              <li>Numéro 2</li>
16          </ul>
17      </li>
18      <li>4e élément</li>
19  </ul>
20
```

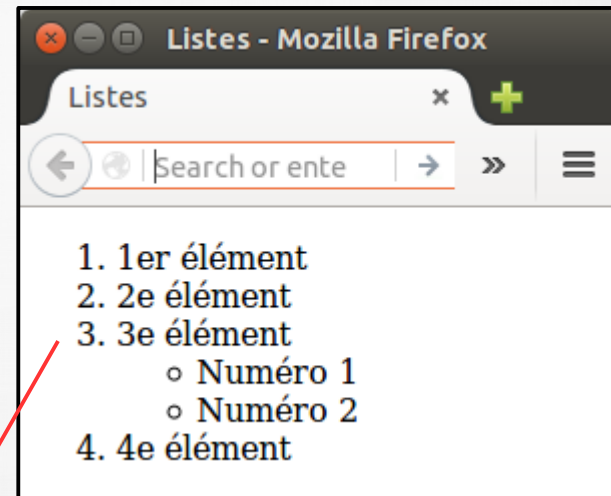


La forme des puces peut être modifiée à l'aide de styles CSS.  
(Cercles, carrés, couleurs, images, etc.)

# Listes ordonnées

- Les balises
  - `<ol>` : liste ordonnée (*ordered list*)
  - `<li>` : élément de liste

```
9      <ol>
10         <li>1er élément</li>
11         <li>2e élément</li>
12         <li>3e élément
13             <ul>
14                 <li>Numéro 1</li>
15                 <li>Numéro 2</li>
16             </ul>
17         </li>
18         <li>4e élément</li>
19     </ol>
20
```

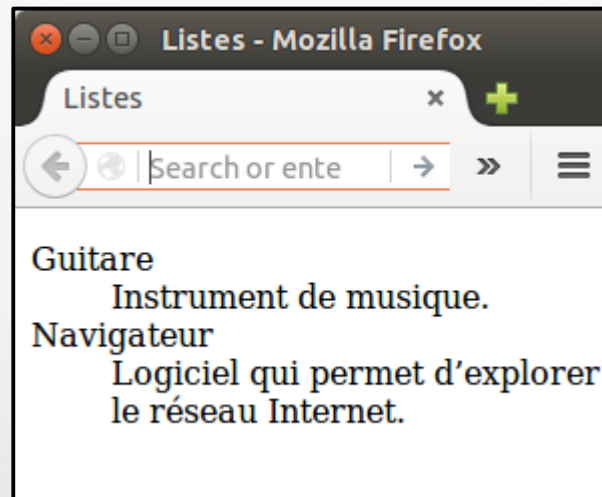


L'apparence des numéros peut être modifiée à l'aide de styles CSS.  
(Chiffres romains, lettres, majuscules, minuscules, etc.)

# Listes de définitions

- Les balises
  - `<dl>` : liste de définition
  - `<dt>` : terme à définir
  - `<dd>` : définition du terme

```
9      <dl>
10          <dt>Guitare</dt>
11          <dd>Instrument de musique.</dd>
12          <dt>Navigateur</dt>
13          <dd>Logiciel qui permet d'explorer le réseau Internet.</dd>
14      </dl>
```

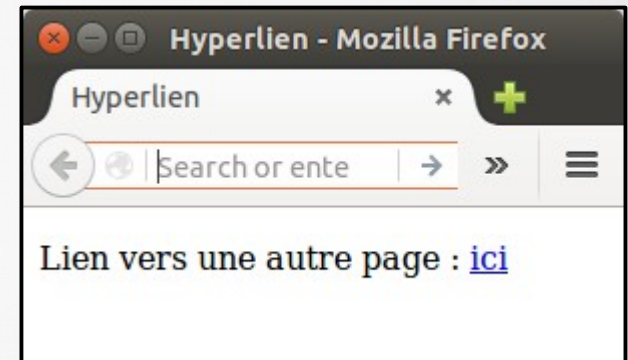




# Hyperliens (1)

- Tout l'intérêt du HTML
- Balise **<a></a>**
- **Attribut** principal : href

```
7 <body>
8   <p>
9     Lien vers une autre page :
10    <a href="autrepage.html">ici</a>
11  </p>
12 </body>
```



```
7 <body>
8   <p>
9     Lien vers une autre page :
10    <a href="http://www.google.com">google</a>
11  </p>
12 </body>
```

## Hyperliens (2)

- Ouverture du lien dans une nouvelle fenêtre :
- **Attribut** : target="\_blank"

```
7  <body>
8  <p>
9      Lien vers une autre page :
10     <a href="http://www.google.com" target="_blank">google</a>
11 </p>
12 </body>
```

- Affichage d'une infobulle lors du survol du lien :
- **Attribut** : title="Texte de l'infobulle."

```
7  <body>
8  <p>
9      Lien vers une autre page :
10     <a href="http://www.google.com" title="Site de google">google</a>
11 </p>
12 </body>
```

# Cibles d'un hyperlien

## Il existe plusieurs cibles possibles pour un hyperlien :

- Lien absolu vers un autre site ;
- Lien relatif vers une page du même site ;
- Lien vers une adresse e-mail ;
- Lien vers un fichier à télécharger ;
- Lien vers une ancre.

```
10      <!-- Lien absolu vers un autre site -->
11      <a href="http://www.google.com">google</a>
12
13      <!-- Lien relatif vers une page du même site -->
14      <a href=" ../cours_02/index.html">Cours 2</a>
15
16      <!-- Lien vers une adresse e-mail -->
17      <a href="mailto:contact@paris5.fr">Nous contacter</a>
18
19      <!-- Lien vers un fichier à télécharger -->
20      <a href="http://www.paris5.com/Cours.pdf">Cours.pdf</a>
21
22      <!-- Lien vers une ancre -->
23      <a href="#top">Haut de la page</a>
```

# Ancre

Une ancre s'utilise en plaçant un attribut *id* (identifiant) sur une balise.

```
9  <p>
10    <a href="#partie_1">Partie 1</a>
11    <a href="page_02.html#partie_2">Partie 2</a>
12    ...
13    ...
14    <h1 id="partie_1">Partie 1</h1>
15  </p>
```

Page\_01.html

```
9  <p>
10    ...
11    ...
12    <h1 id="partie_2">Partie 2</h1>
13  </p>
```

Page\_02.html

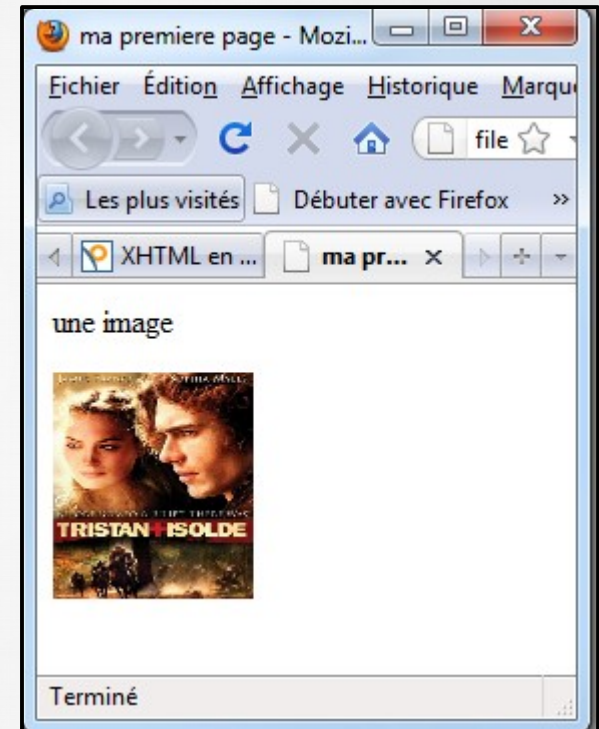
Toute balise peut posséder un identifiant (attribut global).

Il peut exister plusieurs identifiants au sein d'une même page HTML, mais chaque identifiant doit être unique.



# Images (1)

- La balise `` permet de placer une image sur le document.
- Attributs :
  - `src` : URL de l'image (relative ou absolue)
  - `width` : largeur (facultatif)
  - `height` : hauteur (facultatif)
  - `title` : infobulle (facultatif)
  - `alt` : texte alternatif (obligatoire)



9

```

```

## Images (2)

Une image peut également être transformée en lien.  
Il faut pour cela l'insérer dans un élément a.

```
11 <a href="http://www.google.fr">  
12     
13 </a>
```

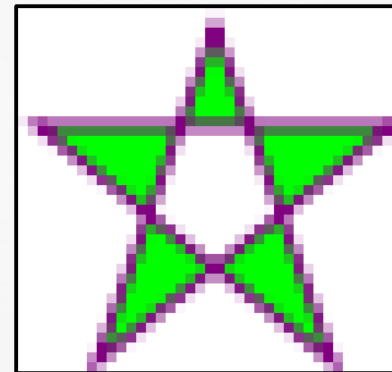
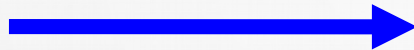
# Images (3)

**SVG** = *Scalable Vector Graphics* (recommandation du W3C)

- SVG décrit un graphique dans un **format XML**
- Chaque élément d'un graphique SVG **peut être animé**
- **Pas de perte de qualité** si changement de taille



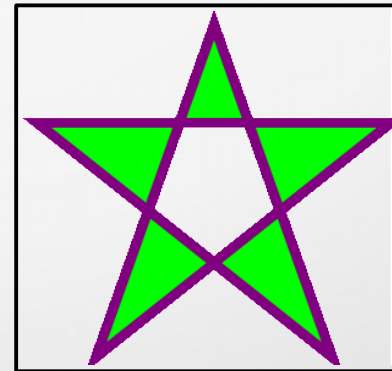
Format bitmap  
(jpg, png, etc.)



**Zoom 8x**



Format vectoriel  
(SVG, PS, etc.)



# Programmer proprement

## **HTML est très tolérant :**

- Il est possible d'écrire une page HTML avec des erreurs
  - Balise ouverte sans fermeture, etc.

## **Ce n'est pas une raison pour en profiter !**

Écrire des pages conformes aux règles du W3C permet :

- de retravailler ses programmes ultérieurement ;
- de faciliter l'écriture collaborative des programmes ;
- d'améliorer le recensement de son site.



# HTML5 vs XHTML5

**XHTML5 est du HTML5 respectant la syntaxe XML.**

**La syntaxe XML est beaucoup plus stricte :**

Balises toujours en minuscule.

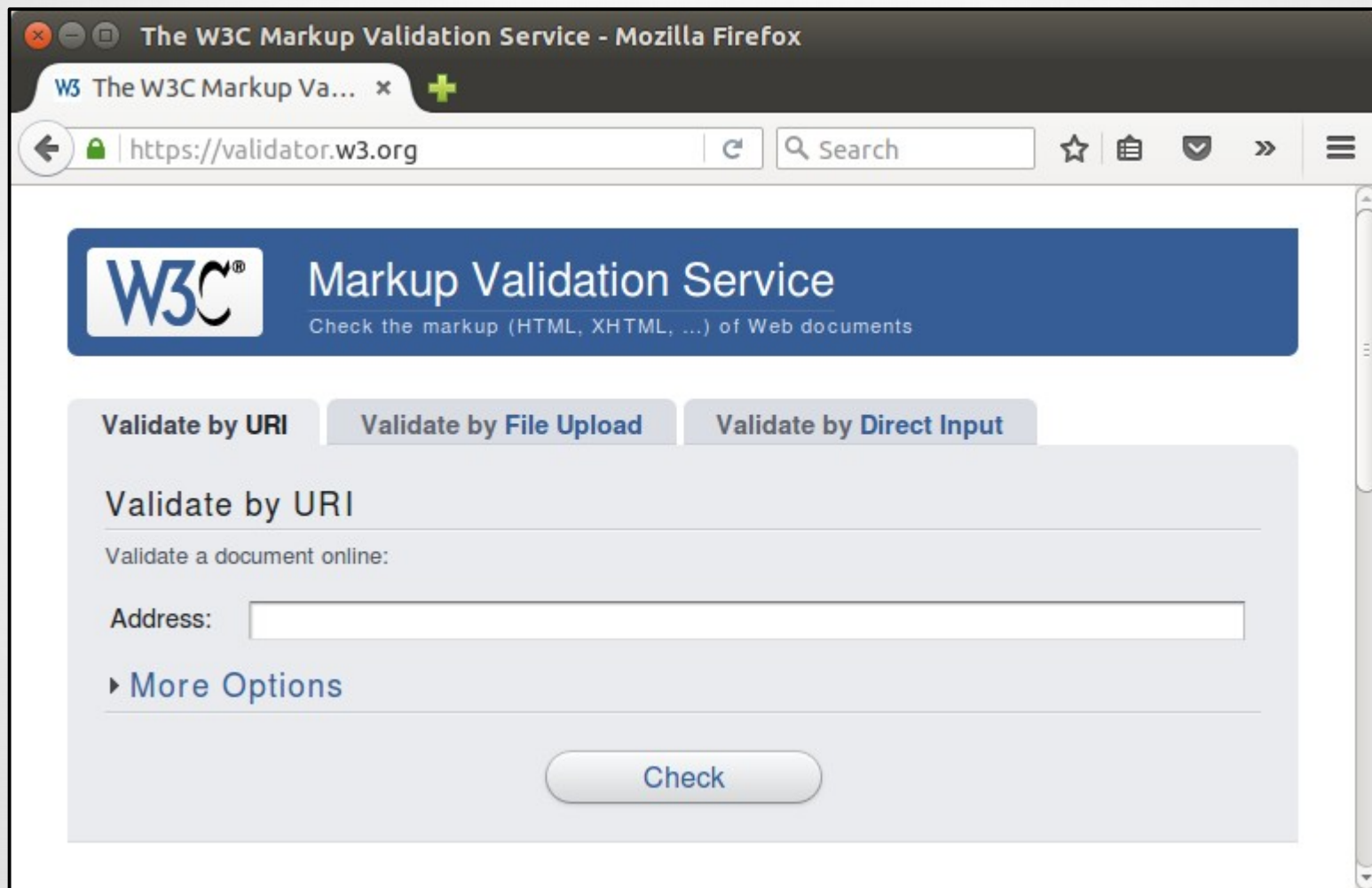
Les balises orphelines se terminent par un *slash* (**<br />**).

Etc.

***Utile si l'on veut traiter son fichier XML.***

***(Par exemple pour le transformer en fichier PDF, etc.)***

# Validation d'une page HTML



The screenshot shows a Mozilla Firefox browser window titled "The W3C Markup Validation Service - Mozilla Firefox". The address bar displays "https://validator.w3.org". The page header features the W3C logo and the text "Markup Validation Service" with the subtitle "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". The "Validate by URI" tab is selected. Under this tab, the text "Validate by URI" is followed by "Validate a document online:". Below this, there is a label "Address:" followed by a text input field. A link "More Options" is visible below the input field. At the bottom of the form, there is a "Check" button.

The W3C Markup Validation Service - Mozilla Firefox

W3 The W3C Markup Va... x +

https://validator.w3.org

W3C® Markup Validation Service

Check the markup (HTML, XHTML, ...) of Web documents

Validate by URI Validate by File Upload Validate by Direct Input

Validate by URI

Validate a document online:

Address:

► More Options

Check