

HTML



Remerciements

- Le matériel des diapositives est pris de différentes sources comprenant :



Le W3C

- Le **W3C** est un organisme **indépendant** qui émet des **standards** pour le **WEB** et notamment le **HTML**
 - Ce dernier est administré par trois entités :

Le **MIT** (États-Unis)

L'université **Keio** (Japon)

L'**Ercim** (Europe)



- Le **W3C** standardise également de nombreuses autres technologies comme **PNG** et autres variations autour de **XML**

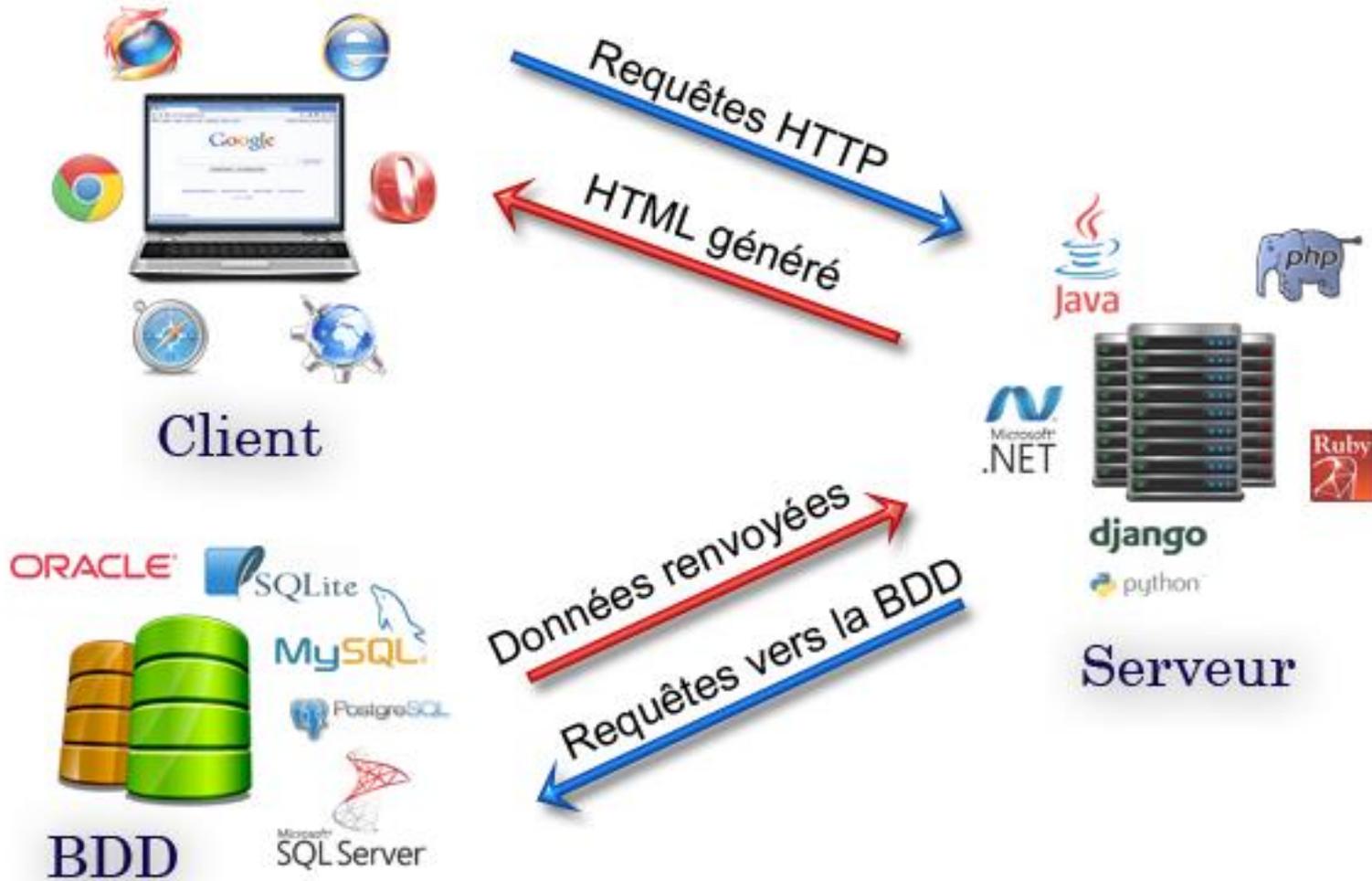


Le WhatWG

- Le **W3C** tente de persévérer vers la voie du **XML** (création du XHTML)
- Le **WhatWG** souhaite améliorer HTML et débute son travail en 2004 avec **Ian Hickson** qui sera l'éditeur officiel du HTML 5
- Le **HTML5** est soutenu devant le **W3C** en 2007. Il sera **retenu** et son premier brouillon sera publié l'année suivante.



Le fonctionnement du WEB



Les familles des métiers du web





HTML

- Qu'est-ce ?
 - Langage de définition de documents sur le Web
 - Utilise le principe de balisage introduit dans SGML :
 - `<balise [liste d'attributs]>...contenu...</balise>`
 - pour décrire des documents :
 - textuels
 - hypertextes
 - multimédias
 - ...
 - Le balisage indique comment interpréter le contenu d'un élément



Les navigateurs et le support de l'HTML 5

- Distinguons les **navigateurs** et les **moteurs de rendu**.
- Chaque navigateur intègre **un** moteur de rendu qui peut être commun à **plusieurs** autres navigateurs.
- Un petit tableau des correspondances :

Moteur	Navigateurs	Développement
Trident	Internet Explorer	Microsoft
Webkit	Google chrome , apple safari , android, epiphany	Apple, Google, KDE, Nokia, RIM, Palm et autres
Gecko	Mozilla firefox , Camino, K-Meleon, Galeon, Flock	Fondation Mozilla
Presto	Opera , opera mobile, opera Wii	Opera Software
KHTML	Konqueror	KDE



Les navigateurs et le support de l'HTML 5

- La fréquence des mises à jours des navigateurs est telle qu'il devient difficile de lister toutes les fonctionnalités **HTML 5** prises en charges.
- Cependant, un site tel que <http://caniuse.com> ou encore <http://fmbip.com> mise à jour régulièrement permet d'avoir des informations assez précises sur la prise en charge du HTML 5 de votre site ou de votre navigateur.

Navigateur	Version	points
Google Chrome	44.0	526
Mozilla Firefox	40.0	467
Opera	31.0	525
Apple Safari	8.0	396
Microsoft Internet Explorer (edge)	11	402



Introduction au langage HTML

- HTML (**H**ypertext **M**arkup **L**anguage) est le format de données structurant une page web.
- HTML est initialement dérivé du **SGML** (**S**tandard **G**eneralized **M**arkup **L**anguage) dont la première publication date de 1986.
- C'est un langage de **balisage** : `<balise>Contenu</balise>`

```
<html>
  <head>
    Les en-têtes du document
  </head>
  <body>
    Le corps du document
  </body>
</html>
```



Le Doctype avant tout

- Le **doctype** est le préambule d'une page HTML, il spécifie le type de document.

- Doctype HTML 4.01

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

- Doctype XHTML 1.1

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"  
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
```

- Doctype HTML 5

```
<!DOCTYPE html>
```



Codes HTML valides W3C

➤ Code HTML 5 valide W3C

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML5</title>
    <meta charset="UTF-8">
  </head>
  <body>
    Vive HTML 5 ! :D
  </body>
</html>
```

➤ Code XHTML 1.1 valide W3C

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML Basic 1.1//EN"
  "http://www.w3.org/TR/xhtml-basic/xhtml-basic11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>XHTML</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  </head>
  <body>
    <p><span>Vive pas XHTML5 ! :D</span></p>
  </body>
</html>
```



<html></html>

- L'élément `<html>` constitue la racine de tout document HTML. Il suit en général le doctype et clôt l'ensemble en fin de page par une balise fermante `</html>`. Il ne peut contenir qu'un seul élément `<head>` suivi immédiatement par un `<body>`.
- Parmi les attributs globaux qui sont applicables à l'ensemble des éléments HTML, il faut noter que `lang` est très fortement recommandé pour l'élément racine `<html>`. Un attribut `lang` vide correspond à un langage inconnu.
- Dans un écosystème XHTML, il est préconisé de l'épauler par un attribut `xml:lang` possédant la même valeur.
- En cas d'absence totale de `lang`, le navigateur se réfère à l'en-tête HTTP `Content-Language`. Si ce dernier est également absent, alors le document est de langue inconnue.



Exemple

```
<!DOCTYPE html>
```

```
<html lang="fr">
```

```
<head>
```

```
<!-- L'en-tête du document -->
```

```
</head>
```

```
<body>
```

```
<!-- Le corps du document -->
```

```
</body>
```

```
</html>
```



<head></head>

- L'en-tête du document exerce un rôle crucial, car il fournit de multiples renseignements sur le document lui-même, que le contenu n'indique pas.
- Dans cette optique, il recueille plusieurs autres balises aux rôles précis. Parmi celles-ci, seule `<title>` est obligatoire.

Élément	Occurrences	Rôle
<code><title></code>	1	Titre du document
<code><meta></code>	0 ou +	Méta-informations
<code><link></code>	0 ou +	Relations vers des ressources externes (par exemple des feuilles de style CSS)
<code><style></code>	0 ou +	Styles CSS embarqués dans le document
<code><script></code>	0 ou +	Scripts embarqués dans le document
<code><base></code>	0 ou 1	Base d'adresses par défaut et de cible pour tous les liens du document

Tableau Imbrications pour `<head>`



Exemple d'utilisation de <head>

```
<!DOCTYPE html>
```

```
<html lang="fr">
```

```
<head>
```

```
<title>Titre du document</title>
```

```
</head>
```

```
<body>
```

```
<!-- Le corps du document -->
```

```
</body>
```

```
</html>
```



<title></title>

- Le titre du document détenu par `<title>` est affiché par le navigateur dans le titre de la fenêtre et/ou dans l'onglet actif. Il est aussi choisi, à juste titre, comme intitulé du lien dans les résultats des moteurs de recherche.



Exemple d'utilisation de <title>

```
<!DOCTYPE html>
```

```
<html lang="fr">
```

```
<head>
```

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

```
<title>Ma page </title>
```

```
</head>
```

```
<body>
```

```
<p>Ceci est ma première page HTML5.</p>
```

```
</body>
```

```
</html>
```



<meta>

- En complément du titre, les balises `<meta>` comblent le besoin de véhiculer plus de méta-informations au sujet du document.
- Il s'agit là aussi de renseignements qui ne sont pas affichés dans le corps de la page, mais qui sont importants pour son interprétation ou son référencement par les moteurs de recherche.

Attribut	Valeurs	Rôle
<code>name</code>	<code>application-name</code> <code>author</code> <code>description</code> <code>generator</code> <code>keywords</code> ou autres valeurs enregistrées	Méta-informations relatives à la page : nom de l'application web, auteur, description du contenu, programme de conception utilisé, mots-clés relatifs au contenu.
<code>http-equiv</code>	<code>refresh</code> <code>default-style</code> <code>content-type</code>	Rafraîchissement de la page. Feuille de style préférée. Déclaration de la page de code et du type MIME.
<code>content</code>	texte	En combinaison avec <code>name</code> et <code>http-equiv</code> , confère sa valeur à la balise.
<code>charset</code>	encodage des caractères	Déclaration de la page de code qui doit être utilisée pour l'interprétation, si celle-ci est différente de l'ASCII.

Tableau Attributs spécifiques à `<meta>`



<link>

- L'élément `<link>` représente une méta-information qui exprime des relations interdocuments. Son rôle est exprimé grâce à la valeur de l'attribut `rel`.

Attribut	Valeurs	Rôle
<code>href</code>	URL	Adresse de la cible.
<code>hreflang</code>	code langue	Langue de base du document cible.
<code>rel</code>	<code>alternate</code> , <code>archives</code> , <code>author</code> , <code>first</code> , <code>help</code> , <code>icon</code> , <code>index</code> , <code>last</code> , <code>license</code> , <code>next</code> , <code>pingback</code> , <code>prefetch</code> , <code>prev</code> , <code>search</code> , <code>sidebar</code> , <code>stylesheet</code> , <code>tag</code> , <code>up</code>	Indique la relation existant entre le document courant et celui de la cible (valable uniquement si l'attribut <code>href</code> est précisé).
<code>media</code>	media query	Spécifie une requête de média pour laquelle la cible est optimisée
<code>type</code>	type MIME	Spécifie le type MIME de la cible, par exemple : <code>text/javascript</code> pour du code JavaScript ; <code>text/css</code> pour une feuille de style CSS.
<code>sizes</code> <nouveau>	<code>any</code> ou <code>hauteur x largeur</code>	Dans le cas d'une relation icône (<code>rel="icon"</code>), indique les dimensions de l'image en pixels. Plusieurs valeurs peuvent être indiquées, séparées par des espaces. Le terme <code>any</code> indique que l'image convient à plusieurs résolutions, par exemple lorsqu'elle est au format vectoriel SVG.



Exemple d'utilisation de `<link>`

```
<!-- Icône associée au document -->
```

```
<link rel="icon" href="icone.gif" type="image/gif" sizes="32x32">
```

```
<!-- Feuille de style pour l'affichage -->
```

```
<link rel="stylesheet" type="text/css" href="css/styles.css"
media="screen">
```

```
<!-- Feuille de style pour l'impression -->
```

```
<link rel="stylesheet" type="text/css" href="css/print.css"
media="print">
```



<base>

- L'unique élément `<base>` situé dans la section `<head>` représente une base commune pour tous les éléments du document. Il est bien entendu facultatif.
- On précise soit l'un ou l'autre de ses deux attributs, soit les deux en même temps.
- Afin de le rendre actif pour toutes les ressources, il est conseillé de le placer en tant que tout premier élément dans `<head>`, car les suivants utiliseront son information.

Attribut	Valeurs	Rôle
<code>href</code>	URL	Indique l'adresse à utiliser comme base de référence (préfixe) pour tous les hyperliens du document.
<code>target</code> <code><nouveau></code>	<code>_blank</code> <code>_parent</code> <code>_self</code> <code>_top</code> <i>nom d'un élément de type <code>iframe</code> (attribut <code>name</code>)</i>	Cible par défaut pour l'ouverture de tous les liens de la page. Cette indication générale peut être contournée individuellement en utilisant un attribut <code>target</code> dans tout lien.

Tableau Attributs spécifiques à `<base>`



<body></body>

- Après tous les éléments susceptibles d'être contenus dans l'entête du document, ce dernier prend fin avec une balise fermante `</head>`, et l'on passe au corps proprement dit de la page HTML.
- Exemple d'utilisation:
 - `<!DOCTYPE html>`
 - `<html lang="fr">`
 - `<head>`
 - `<meta charset="utf-8">`
 - `<title>Ma page </title>`
 - `</head>`
 - `<body>`
 - `<p>Ceci est ma première page HTML5.</p>`
 - `</body>`
 - `</html>`



```
<BODY BGCOLOR="#RRVVBB" BACKGROUND= "AdresseImage"  
LINK="#RRVVBB" VLINK="#RRVVBB"  
TEXT="#RRVVBB" ALINK="#RRVVBB">
```

- BGCOLOR : couleur de fond de la fenêtre
- BACKGROUND : image de fond (répétée sous forme de mosaïque)
- TEXT : couleur du texte ordinaire *
- LINK : couleur du texte/de la bordure d'une image qui est un lien *
- VLINK : couleur du texte/de la bordure d'une image qui est un lien amenant sur une page déjà vue *
- ALINK : couleur prise par un lien tant qu'on clique dessus (peu utilisé)

- * = si omis, utilisation des couleurs standards (définies dans les préférences utilisateur)



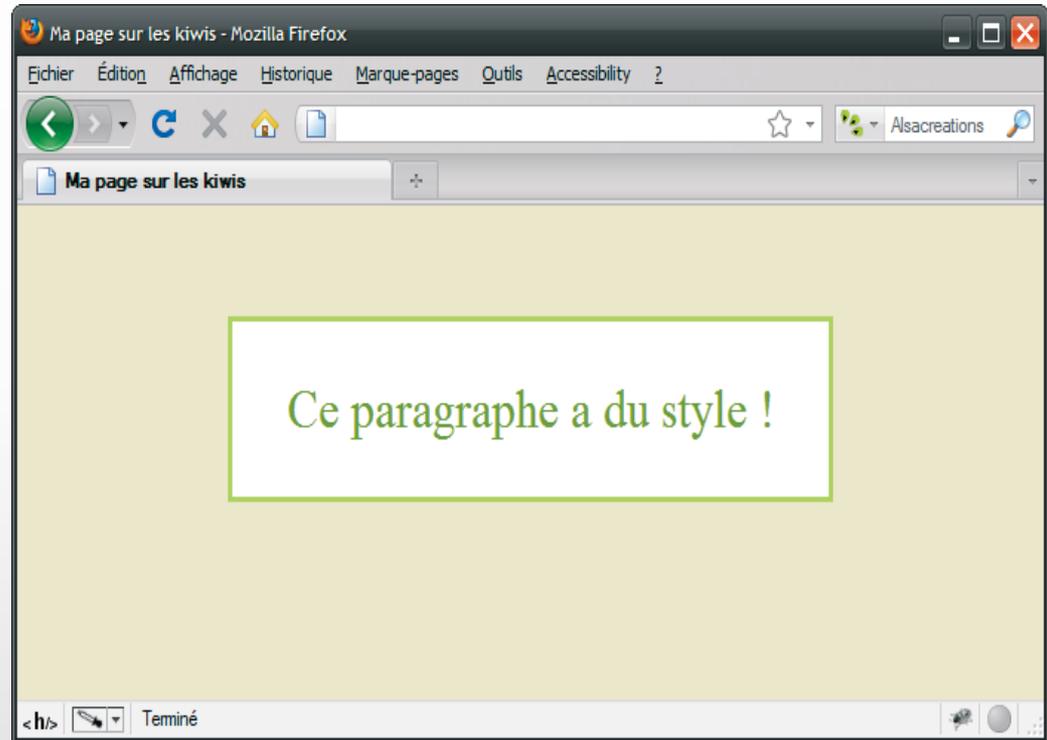
<style>

- En complément d'une feuille de style externe – qui reste la solution la plus performante et la plus modulaire pour un site classique – l'élément `<style>` permet d'introduire des propriétés spécifiques au document courant. Son contenu doit exclusivement relever de la syntaxe des feuilles de style.
- Les propriétés seront appliquées au corps de la page et à ses descendants si ceux-ci satisfont aux sélecteurs CSS indiqués.

Attribut	Valeurs	Rôle
<code>type</code>	type MIME	Type MIME d'un langage de style. Dans l'extrême majorité des cas, il s'agira de <code>text/css</code> .
<code>media</code>	media query	Spécifie une ou plusieurs requête(s) de média à laquelle l'instruction de style sera appliquée.
<code>scoped</code> <code><nouveau></code>	scoped	Indique que les instructions de style s'appliquent uniquement à l'élément parent qui contient la balise <code><style></code> ainsi qu'à ses descendants. Si cet attribut est absent, les propriétés CSS s'appliqueront à l'ensemble du document.

Exemple d'utilisation de `<style>`

```
<!doctype html>
<html lang="fr">
<head>
<meta charset="utf-8">
<title>Ma page sur les kiwis</title>
<style type="text/css" media="screen">
  body {
    background:#EFECCA;
  }
  p {
    color:#78A419;
    font-size:2em;
    text-align:center;
    border:3px solid #B5E655;
    background:white;
    padding:1em;
    width:50%;
    margin:2em auto;
  }
</style>
</head>
<body>
<p>Ce paragraphe a du style !</p>
</body>
</html>
```





Texte

- Affichage écran
 - Tout texte tapé hors d'une balise est considéré comme du texte et apparaîtra de ce fait à l'écran
- Mise en forme à 2 niveaux
 - Caractères
 - Paragraphes
- Codage de caractères
 - Utilisation d'un système de codage qui permette un affichage universel quelque soit la langue de l'ordinateur sur lequel les pages sont lues.
 - Spécifiques à certaines langues (et notamment au français)

Le codage des caractères spéciaux

 	Espace insécable	¦		±	±
£	£	¥	¥	Ø	Ø
©	©	®	®	°	°
²	²	³	³	&	&
¼	¼	½	½	¾	¾
à	à	À	À	œ	œ
é	é	É	É	ë	ë
î	î	Î	Î	ç	ç
"	"	ß	ß	ñ	ñ
<	<	>	>	µ	μ



Mise en forme des caractères

- Les styles
 - **Gras** : ` ... `
 - *Italique* : `<I> ... </I>`
 - Souligné : `<U> ... </U>`
- Taille de la police par défaut
 - Possibilité de fixer la taille de police d'une page web `<BASEFONT SIZE="taille">` (avec $1 \leq \text{taille} \leq 7$)
 - Si cette balise n'est pas utilisée, la taille par défaut des caractères est fixée à 3.



Mise en forme des caractères

■ Spécification d'une police locale

- `<FONT FACE="police1[,police2,...]
SIZE="s|+s|-s" COLOR="#RRVVBB"> ...`

- ``

- **FACE** : la première police installée est utilisée. Ne proposer que des polices dont on est sûr qu'elles sont disponibles sur l'ordinateur de consultation :

- Arial/Helvetica
- Times New Roman/Times

- **SIZE** : taille de la police

- 1 ... 7 (taille absolue)
- +1, +2... (taille relative par rapport à la valeur par défaut)



Mise en forme des caractères

- Remarque sur les balises

- Elles peuvent être imbriquées :

```
<FONT SIZE="..."> ...  
  <FONT SIZE="... ">  
  ...  
  </FONT>  
  ...  
</FONT>
```

- Permet de changer des paramètres ponctuellement (couleur ou taille)



Mise en forme des paragraphes

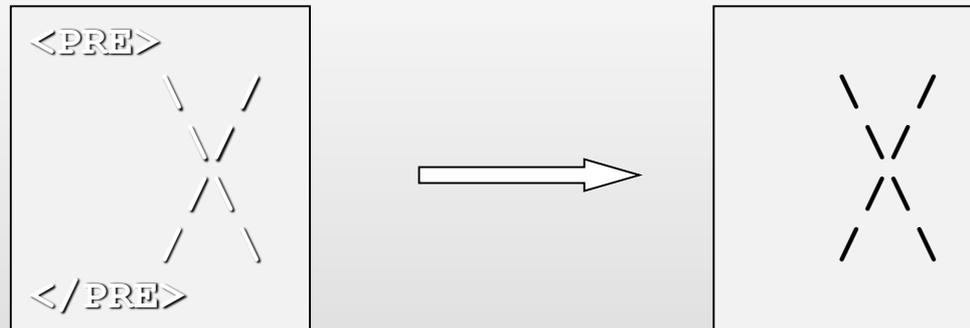
- `<CENTER> ... </CENTER>`
 - Centre les éléments inclus entre la balise ouvrante et la balise fermante
- `<JUSTIFY> ... </JUSTIFY>`
 - Justifie les éléments inclus entre la balise ouvrante et la balise fermante
 - À utiliser avec prudence !
- `
`
 - Pas de balise fermante !
 - Retour à la ligne
 - `

` : 2 retours à la ligne = 1 ligne d'espacement

Mise en forme des paragraphes

- `<PRE> ... </PRE>`

- Chaque espace, chaque retour à la ligne est pris en compte. C'était l'ancienne méthode pour faire les tableaux
- Affichage sans mise en page
 - Saut de ligne = saut de ligne
 - Espace = espace insécable





Groupement: div

- Conteneur générique par excellence, `<div>` représente une division de document.
- D'un point de vue sémantique, il n'a aucune valeur et n'est voué qu'à regrouper des éléments ayant des propriétés communes.
- La balise `<div>` définit une division ou une section, appelée aussi calque, dans un document HTML.
- La balise `<div>` est utilisée pour regrouper des éléments sous forme de bloc et ainsi pouvoir les formater en CSS.
- **Remarque** : Les navigateurs placent toujours un saut de ligne avant et après un élément `<div>`.



Exemple d'utilisation de <div>

□ Exemple 1:

```
<div>
```

```
<p>Un paragraphe</p>
```

```

```

```
<p>Un autre paragraphe</p>
```

```
</div>
```

□ Exemple 2:

```
<div lang="fr" class="introduction">
```

```
<p>Paragraphe d'introduction</p>
```

```
</div>
```

```
<div class="suite">
```

```
<p>Un autre paragraphe</p>
```

```
</div>
```



- À la différence de `<div>`, un conteneur `` est spécifiquement dédié au contenu phrasé.
- En revanche, tout comme lui, il ne possède pas de sens particulier d'un point de vue sémantique.
- Il peut être utilisé également pour grouper des éléments ou une portion de texte à des fins d'affectation de styles (par exemple en utilisant les attributs `id` ou `class`) ou de partage d'attributs.



Listes

- Les listes HTML sont prévues pour dresser des énumérations possédant une valeur sémantique.
- Les structures disponibles pour une liste ordonnée (liste numérotée) et non ordonnée (liste à puces) sont constituées par les éléments `` et `` au sein desquels chaque élément individuel est un ``.
- Elles sont couramment employées pour la conception de menus de navigation, qui ne sont en réalité que des énumérations de liens.
- Lorsqu'il s'agit de rédiger une liste de définitions, l'élément `<dl>` peut être utilisé en conjonction avec `<dt>` et `<dd>`.



- Une liste ordonnée (ordered list en anglais) est équipée de puces numérotées automatiquement par le navigateur.
- Le conteneur `` est toujours parent de zéro ou plusieurs éléments de liste ``.
- Exemple de liste ordonnée ``:

```
<p>Les modules de cette semestre:</p>
```

```
<ol>
```

```
<li>Module 1</li>
```

```
<li>Module 2</li>
```

```
<li>Module 3</li>
```

```
</ol>
```



Attributs spécifiques à

- Exemples de listes ordonnées avec attributs:

```
<ol start="2" type="A">
```

```
<li>Premier élément</li>
```

```
<li>Deuxième élément</li>
```

```
<li>Troisième élément</li>
```

```
</ol>
```

Figure
Liste ordonnée

B. Premier élément
C. Deuxième élément
D. Troisième élément

Tableau Attributs spécifiques à

Attribut	Valeurs	Rôle
start	nombre entier	Valeur de départ pour la numérotation. Peut être négatif.
reversed	reversed ou "" ou sans valeur	Indique que la numérotation est descendante.
type	1 ou a ou A ou i ou I	Type de numérotation (numérique classique, lettrée, chiffres romains).



- La liste non ordonnée (unordered list en anglais) `` partage le podium avec `` bien qu'elle soit plus fréquemment utilisée.
- Comme son nom l'indique, son contenu n'est pas spécialement trié et son ordre peut être changé sans altérer significativement son sens.
- Exemple de liste non ordonnée ``:

`<p>`Plan B :`</p>`

``

``Stocker tout le parmesan de l'univers``

``Lancer une nouvelle pizza``

``Conquérir le monde``

``

Figure
Aperçu de liste à puces,
non ordonnée

Plan B :

- Stocker tout le parmesan de l'univers
- Lancer une nouvelle pizza
- Conquérir le monde



- Si vous avez bien suivi les précédentes explications, vous savez déjà que `` est un élément (list item en anglais) de liste ordonnée `` ou non ordonnée ``. Il remplit cette même fonction pour l'élément `<menu>` (voir du côté des éléments interactifs).
- Il comprend un attribut facultatif nommé `value`, qui fut déclaré obsolète dans les précédentes versions de HTML, mais réintroduit pour des raisons pratiques en HTML 5. Cet attribut n'est autorisé que dans le cadre d'une liste ordonnée `` puisqu'il correspond à la numérotation de cette liste pour contrecarrer les effets de l'incrément automatique.

```
<ol>
```

```
<li value="1">Premier élément</li>
```

```
<li value="3">Deuxième élément</li>
```

```
<li>Troisième élément</li>
```

```
<li value="1">Quatrième élément</li>
```

```
<li>Cinquième élément</li>
```

```
</ol>
```



<dl>

- La liste de définitions `<dl>` (description list en anglais) répond au besoin de lister des paires de termes et de descriptions qui leur sont liées.
 - On peut la comparer à un dictionnaire associant la définition `<dd>` au mot `<dt>`.
 - Aucun tri alphabétique n'est cependant à l'œuvre, il appartient à l'auteur de la liste de l'ordonner dans le code source.

Exemple de liste de définitions

■ Code

```
<p>Mots-clés pour <span lang="en">media queries</span> :</p>
```

```
<dl>
```

```
<dt>screen</dt>
```

```
<dd>Écrans</dd>
```

```
<dt>handheld</dt>
```

```
<dd>Périphériques mobiles ou de petite taille</dd>
```

```
<dt>print</dt>
```

```
<dd>Impression</dd>
```

```
<dt>projection</dt>
```

```
<dd>Projecteurs (ou présentations avec slides)</dd>
```

```
<dt>tv</dt>
```

```
<dd>Téléviseur</dd>
```

```
<dt>all</dt>
```

```
<dd>Tous les précédents</dd>
```

```
</dl>
```

Liste de définitions

Mots-clés pour media queries :

screen

Écrans

handheld

Périphériques mobiles ou de petite taille

print

Impression

projection

Projecteurs (ou présentations avec slides)

tv

Téléviseur

all

Tous les précédents



Création de lignes horizontales

- `<HR SIZE="s" WIDTH="w|w%" ALIGN="LEFT|CENTER|RIGHT" COLOR="#RRVVBB" NOSHADE >`
 - **SIZE** : épaisseur en pixel de la ligne
 - **WIDTH** : largeur de la ligne.
Peut s'exprimer en valeur absolue (pixels) ou en valeur relative (pourcentage de la largeur de la fenêtre)
 - **ALIGN** : alignement à gauche au centre à droite dans la fenêtre (à utiliser si la ligne ne fait pas toute la largeur de la fenêtre)



Création de lignes horizontales

■ `<HR>`

- Utilisée sans option, cette balise produit une ligne grise ombrée de 1 pixel d'épaisseur faisant toute la largeur de la fenêtre

- Attention, options particulières à certains navigateurs !

■ `COLOR="#RRVVBB"`

- Option valable avec Internet Explorer
- Permet de spécifier la couleur de la ligne

■ `NOSHADE`

- Option valable avec Netscape
- Permet de désactiver l'ombrage de la ligne



Sections et titres: Les sections

- Les sections délimitent les zones du document possédant une valeur sémantique particulière.
- Contrairement à `<div>` et `` qui n'ont aucun rôle particulier, hormis celui de regrouper des éléments, chaque section possède un rôle précis, et peut être utilisée une à plusieurs fois dans le document.
- À l'exception de `<address>`, il s'agit de nouveautés HTML 5.



Nouveaux éléments de section HTML 5

Tableau

Nouveaux éléments de section HTML 5

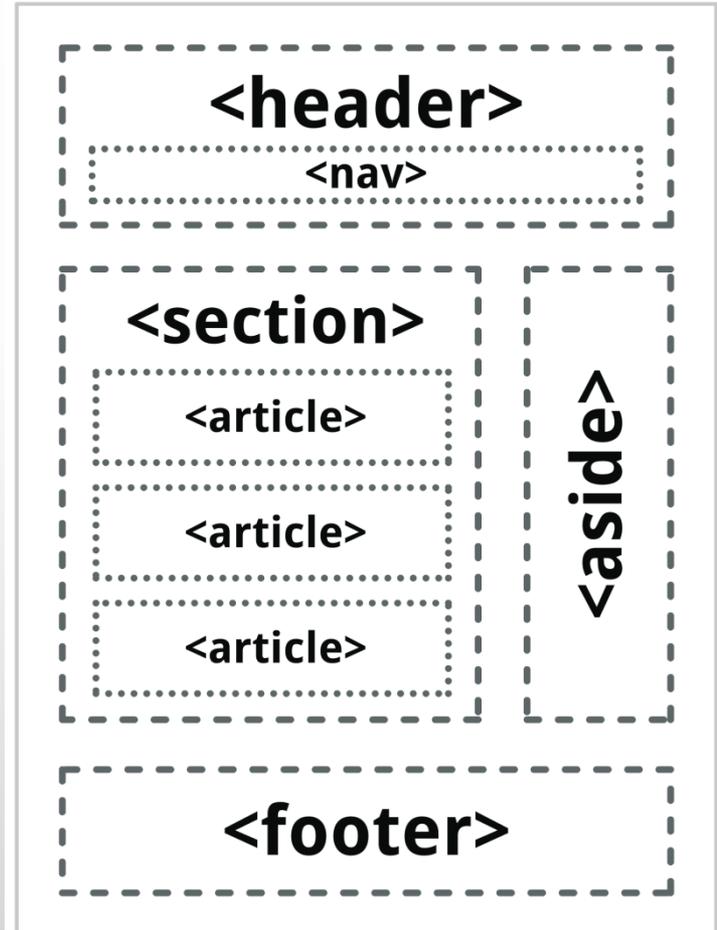
Nom	Détails
<code>section</code>	Section générique, regroupant une même thématique de contenu, de préférence avec un en-tête.
<code>article</code>	Section de contenu dans un document ou une application web, dont la composition peut être indépendante du reste de la page et extraite individuellement.
<code>nav</code>	Section abritant des liens de navigation majeurs, permettant de naviguer au sein du document ou vers d'autres pages.
<code>aside</code>	Section dont le contenu est tangentiellement lié à ce qui l'entoure, et qui peut être considérée comme séparé de ce contenu.
<code>header</code>	Section d'introduction (d'un document, d'une autre section, d'un article) pouvant contenir – selon le contexte – titres, navigation, formulaire de recherche, table des matières, logo, etc.
<code>footer</code>	Pied de page, de section ou d'article – selon son plus proche ancêtre – pouvant contenir des informations connexes et une navigation annexe.



Exemple

■ Exemple de code en HTML 5

```
<!doctype html>
<html >
  <head>
    <title>Ma structure complète en HTML5</title>
    <meta charset="utf-8">
  </head>
  <body>
    <!-- Début -->
    <header>
      <nav></nav>
    </header>
    <section>
      <article></article>
      <article></article>
      <article></article>
    </section>
    <aside></aside>
    <footer></footer>
    <!-- Fin -->
  </body>
</html>
```





<section>

- Élément fondamental de la nouvelle approche de découpe sémantique des documents, `<section>` regroupe un contenu relatif à une même thématique.
- Il peut être introduit par des éléments de titraile (`<header>`, `<hgroup>`, `<h1>` à `<h6>`, etc.).
- Ses cas d'utilisation sont multiples, cependant il ne faut pas le surexploiter : ce n'est pas un remplaçant de `<div>` en HTML 5, mais bien un élément spécialisé. On peut l'envisager pour scinder un document en plusieurs chapitres, pour découper un contenu présenté dans différents onglets ou différentes vues, voire pour délimiter les thématiques d'un élément `<article>`.
- Il faut l'éviter si l'objectif est uniquement de l'utiliser pour affecter un style ou appliquer un script ; si d'autres éléments sont plus appropriés (particulièrement `<article>`, `<aside>`, `<nav>`, `<header>`, `<footer>` et `<div>` en dernier recours).



Exemple d'utilisation de `<section>` englobant des articles

```
<section>
```

```
<h1>Articles</h1>
```

```
<article>
```

```
<h2>Titre de l'article</h2>
```

```
<p>Contenu de l'article</p>
```

```
</article>
```

```
<article>
```

```
<h2>Titre de l'article</h2>
```

```
<p>Contenu de l'article</p>
```

```
</article>
```

```
<article>
```

```
<h2>Titre de l'article</h2>
```

```
<p>Contenu de l'article</p>
```

```
</article>
```

```
</section>
```



Exemple d'usage de `<section>` dans un article

```
<article>
```

```
  <h1>Les couleurs</h1>
```

```
  <p>Ce sont des perceptions visuelles subjectives dépendant de la  
    fréquence des ondes lumineuses.</p>
```

```
  <section>
```

```
    <h2>Le rouge</h2>
```

```
    <p>Couleur primaire, le rouge excite le plus l'œil humain après le jaune.</p>
```

```
  </section>
```

```
  <section>
```

```
    <h2>Le vert</h2>
```

```
    <p>Menthe à l'eau ou grenouille, le vert est souvent associé à la nature.</p>
```

```
  </section>
```

```
</article>
```



<article>

- L'élément `<article>` est une spécialisation de `<section>`.
- Il possède une plus forte valeur sémantique, et pour cela il faut le considérer comme une portion de document qui se suffit à elle-même, par exemple une entrée distincte parmi d'autres. On peut extraire son contenu, éventuellement le syndiquer, sans que celui-ci perde son sens.
- Étant donné le caractère souhaité autonome de cet élément vis-à-vis de ce qui l'entoure, il est tout à fait possible d'utiliser une balise `<header>` pour l'introduire et `<footer>` pour lui affecter un « pied d'article », voire de le découper en plusieurs parties via `<section>`.



Exemple d'utilisation de <article>

```
<article>
```

```
<header>
```

```
<h1>Titre de l'article</h1>
```

```
<p>Publication le <time datetime="2011-02-03" pubdate>Jeudi 3  
février 2011 par George Abitbol</p>
```

```
</header>
```

```
<p>Contenu de l'article, premier paragraphe.</p>
```

```
<p>Contenu de l'article, deuxième paragraphe.</p>
```

```
<footer><!-- Pied de l'article --></footer>
```

```
</article>
```



<header>

- Comme son nom le suggère, <header> correspond à un en-tête de section. Il peut s'agir de l'en-tête général du document <body>, s'il est placé de telle sorte dans la hiérarchie qu'il ne dispose pas d'ancêtre de section. Il peut alternativement être utilisé en début de <section> ou <article> le cas échéant, et donc contenir :
 - des informations d'introduction (titres <h1> à <h6>, paragraphes, méta-informations, etc.) ;
 - une navigation (<nav>, <form>, <a>, etc.) pour cette sous-partie de document ;
 - une table des matières pour l'ancêtre <section> ou <article> qui le contient.
- Son usage n'est donc pas limité à une seule occurrence par page.



Exemple d'utilisation de <header>

```
<body>
```

```
<header>
```

```

```

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

```
<nav>
```

```
<ul>
```

```
<li><a href="/">Accueil</a></li>
```

```
<li><a href="/contact">Contact</a></li>
```

```
<li><a href="/a-propos">À propos</a></li>
```

```
</ul>
```

```
</nav>
```

```
</header>
```

```
<!-- Contenu du document... -->
```

```
</body>
```



<footer>

- Le pendant de `<header>` pour le pied de page est `<footer>`.
- Il s'agit également d'un élément spécialisé possédant une valeur sémantique, visant à contenir les informations que l'on peut habituellement placer en fin de section, telles que des mentions légales, des informations de contact ou les sources d'une actualité, et éventuellement une navigation.
- Ce contenu se rapporte au plus proche ancêtre de section, c'est-à-dire le document dans son ensemble s'il n'en possède pas, ou bien son parent de type `<article>` ou `<section>`.
- De par sa nature, un élément `<footer>` ne peut être contenu dans `<header>`, `<address>` ou `<footer>` lui-même.
- Son usage n'implique pas obligatoirement la présence d'un élément `<header>`.



Exemple d'utilisation de <footer>

```
<body>
```

```
<!-- Contenu du document... -->
```

```
<footer>
```

```
<p>Tous droits réservés</p>
```

```
<nav>
```

```
<ul>
```

```
<li><a href="/">Retour à l'accueil</a></li>
```

```
<li><a href="/contact">Contact</a></li>
```

```
<li><a href="/plan">Plan du site</a></li>
```

```
</ul>
```

```
</nav>
```

```
</footer>
```

```
</body>
```



<nav>

- Les niveaux de navigation peuvent être multiples sur un site web ou une application.
- Il peut s'agir de liens portant sur l'intégralité du site, sur une sous-partie de l'arborescence en particulier, vers des pages communes et transversales, ou bien encore vers des sections du même document.
- L'élément `<nav>` est un choix de prédilection pour la navigation principale (souvent dans `<header>`) et éventuellement pour les navigations annexes de moindre valeur (souvent dans `<footer>`).



Exemple d'utilisation de `<nav>`

```
<nav>
```

```
<ul>
```

```
<li><a href="/accueil">Accueil</a></li>
```

```
<li><a href="/articles">Articles</a></li>
```

```
<li><a href="/archives">Archives</a></li>
```

```
<li><a href="/contact">Contact</a></li>
```

```
</ul>
```

```
</nav>
```



<aside>

- Une page web est bien souvent dotée d'un contenu principal et d'informations connexes qui ne sont pas essentielles à sa compréhension.
- On parle alors de contenu tangentiel, et l'élément `<aside>` est destiné à l'abriter. Il ne s'agit pas simplement de le considérer comme une section vouée à être rendue graphiquement à gauche ou à droite du contenu principal, mais bien d'un point de vue sémantique comme une section possédant une substance périphérique ou ampliative.
- On peut considérer comme répondant à ces critères : une définition d'un des termes utilisés dans le contenu, une biographie, un glossaire, une chronologie apportant un fond historique, voire d'autres éclaircissements, astuces et notes qui n'ont de limite que l'imagination.



Exemple d'utilisation de <aside>

```
<body>
```

```
<!-- Contenu du document... -->
```

```
<aside>
```

```
<h2>Gestion</h2>
```

```
<ul>
```

```
<li><a href="/panier">Panier</a></li>
```

```
<li><a href="/commandes">Commandes</a></li>
```

```
<li><a href="/suivi">Suivi</a></li>
```

```
</ul>
```

```
<h2>Glossaire</h2>
```

```
<dl>
```

```
<dt>CSS</dt>
```

```
<dd>Cascading Style Sheets</dd>
```

```
<dt>HTML</dt>
```

```
<dd>HyperText Markup Language</dd>
```

```
</dl>
```

```
</aside>
```

```
</body>
```



<address>

- Un bloc contenant des informations de contact peut être réalisé grâce à l'élément `address`. Celui-ci s'applique à son plus proche ancêtre de type `body` ou `article`.
- Si l'élément `address` est unique pour le document (dans `body`), alors il contient les informations de contact pour cet ensemble.
- Si l'on en retrouve plusieurs au sein d'éléments de type `article`, alors ils contiennent les informations de contact relatives à l'auteur de l'article uniquement.



Exemple d'utilisation de `<address>`

```
<article>
```

```
<p>Auteurs de cet article :</p>
```

```
<address>
```

```
<a href="http://www.xxxx.fr/">Raphaël Goetter</a>,&br/><a href="http://www.yyyy.ma/">Moi-même</a>
```

```
</address>
```

```
</article>
```



<h1> à <h6>

- Les éléments de type `<hX>` où `X` représente un chiffre entre 1 et 6 correspondent à des titres (heading en anglais) de niveaux différents.
- `<h1>` est un titre de premier niveau, `<h2>` un titre de deuxième niveau, et ainsi de suite jusqu'au petit dernier `<h6>`.
- Ils possèdent une forte valeur sémantique et doivent être utilisés là où un titre de document ou de section est nécessaire, car il ne suffit pas d'affecter des propriétés de style telles qu'une police énorme à un paragraphe `<p>` pour lui conférer la vocation d'un titre.

Exemple d'utilisation de `<hx>`

■ Code

```
<body>
```

```
<h1>Titre de niveau 1</h1>
```

```
<h2>Titre de niveau 2</h2>
```

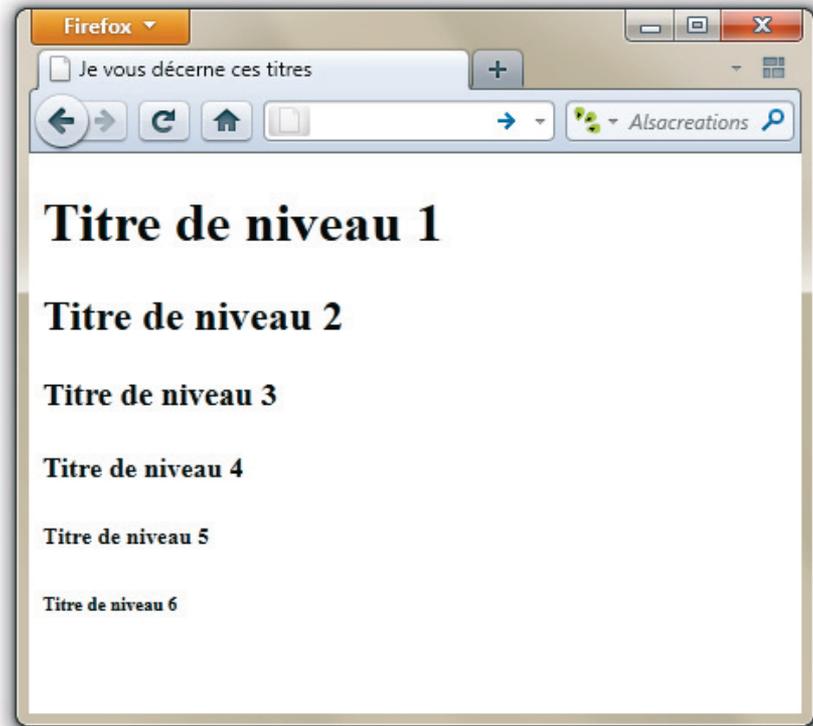
```
<h3>Titre de niveau 3</h3>
```

```
<h4>Titre de niveau 4</h4>
```

```
<h5>Titre de niveau 5</h5>
```

```
<h6>Titre de niveau 6</h6>
```

```
</body>
```





<hgroup>

- L'élément `<hgroup>` regroupe un ou plusieurs titres `<h1>` à `<h6>`. Idéalement, il n'a de vocation à être utilisé qu'à partir de deux titres, car il représente l'en-tête d'une section, lorsque celle-ci dispose de plusieurs titres, sous-titres, ou titres alternatifs.

```
<article>
```

```
<hgroup>
```

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

```
<h2>Sous-titre</h2>
```

```
</hgroup>
```

```
<p>Contenu de l'article</p>
```

```
</article>
```



<p>

- Certainement une des balises les plus rencontrées en HTML, `<p>` marque le début un nouveau paragraphe qui se voit fermé par `</p>`.
- Son contenu est de type phrasé, c'est-à-dire qu'il contient effectivement du texte, potentiellement des sauts de ligne `
`, mais aussi d'autres éléments de phrasé pouvant contribuer à la sémantique de ce texte, par exemple ``, ``, `<abbr>`.....
- C'est un élément de type bloc qui offre une information sémantique au texte qu'il contient (un autre paragraphe le suivant pourra correspondre à une nouvelle idée).
- Un paragraphe ne peut contenir un autre paragraphe.
- Le style par défaut des navigateurs introduit un espace avant et après chacun des paragraphes.



<blockquote>

- Un bloc de citation `<blockquote>` représente une section du document qui est extraite d'une autre source. Il peut contenir des éléments de flux, c'est à-dire principalement un ou plusieurs paragraphes, mais aussi des titres et des images.

Exemple de citation à l'aide de `<blockquote>`

```
<p>Un vieux sage disait :</p>  
<blockquote>  
  <p>Un homme azerty en vaut deux.</p>  
</blockquote>
```



<blockquote> (cite)

- L'attribut `cite` fait référence à l'adresse de cette source. Il n'est pas rendu visuellement dans les navigateurs traditionnels, mais peut être exploité par des extensions ou des robots d'indexation.

Exemple de citation à l'aide de <blockquote>

```
<p>Tim Berners-Lee a écrit :</p>
<blockquote cite="http://www.w3.org/People/Berners-Lee/Kids">
  <p lang="en">I just had to take the hypertext idea and connect it to
the TCP and DNS ideas and -- ta-da! -- the World Wide Web.</p>
</blockquote>
```



- L'élément `` confère au texte une forte importance. Visuellement, il est représenté par un corps de police plus gras, mais contrairement à ``, il possède une réelle valeur sémantique. Du contenu placé entre balises `` doubles aura deux fois plus d'importance.

Exemple d'usage de ``

```
<p>Il est important de noter que <strong>nous ne remboursons pas</strong> les chaussettes usagées.</p>
```



- Une emphase peut être appliquée à une portion de texte avec ``. Il s'agit d'un procédé linguistique qui donne de l'importance ou une affectation pompeuse au discours.
- Il ne doit pas être confondu avec `<i>`, même si son apparence par défaut est un style de texte italique.



- Elle représente désormais une mise en valeur via un style différent, sans lui conférer d'importance particulière dans son contexte.
- Il peut s'agir de mots-clés dans un document, de noms de produits dans un article.

Exemple d'usage de

```
<p>Le <b>patator</b> (aussi appelé lance-patate) est une arme artisanale, projetant à moyenne distance un projectile, le plus souvent une patate.</p>
```



- la balise `<i>` tient une bonne place sur le podium des balises historiques, pour le rôle de la mise en forme en italique.

Exemple d'usage de `<i>`

```
<p>La lettre <i>A</i> est la première de l'alphabet.</p>
```



<small>

- La balise small est utilisée comme contenu relatif mais non essentiel. On utilisera pour déclarer un copyright, des mises en garde..
- Exemple

```
<small>Les éléments de cette page sont sous licence <a rel="license" href="http://creativecommons.org/licenses/by-nc-sa/2.0/fr/">Licence Creative Commons</a></small>
```



<abbr>

- Les abréviations sont de plus en plus courantes. Cette balise permet de créer les abréviations
- Exemple
- `<abbr title="Hypertext Markup Language">HTML</abbr>`



<code>

- Un fragment de code informatique – quel que soit le langage – peut être contenu dans un élément `<code>`.
- En général, une police à pas fixe lui est appliquée, ce qui le distingue d'un texte conventionnel et évoque bien un morceau d'instruction qui pourrait être lue dans un éditeur de texte.
- Exemple:

`<p>` Ce livre contient des allusions à la fonction `<code>document.getElementById()</code>` et à la balise `<code><script></code>`, sans oublier la propriété CSS `<code>display</code>`.`</p>`



Liens

- Les liens sont l'essence même du web car ils permettent de relier des pages entre elle
- Les liens entre les pages sont la source de l'analogie entre internet et la d'araignée (web)
- Le bon choix du contenu et de la disposition des liens est primordial pour la **navigabilité** du site



<a>

- L'élément `<a>` (anchor en anglais ou ancre) est un hyperlien.
- L'activer: cliquer dessus dans la plupart des cas, signifie au navigateur que l'on souhaite changer d'adresse de navigation et consulter une autre ressource.
- Ce fondement même des principes du Web n'est pas très complexe à mettre en œuvre. Il convient en général de savoir écrire l'adresse de destination dans l'attribut `href`, éventuellement complété par des renseignements précisés dans les autres attributs.



Attributs spécifiques à <a>

Tableau Attributs spécifiques à <a>

Attribut	Valeurs	Rôle
href	URL	Cible du lien.
hreflang	code de langue	Langage de base de la cible du lien.
media <nouveau>	media query	Spécifie une requête de média pour laquelle la cible est optimisée
rel	alternate, archives, author, bookmark, external, first, help, index, last, license, nextnofollow, norereferrer, prefetch prev, search, sidebar, tag, up	Spécifie la relation établie par le lien, entre le document courant et la cible. Plusieurs valeurs peuvent être combinées, séparées par des espaces. (Ne sont pas autorisés contrairement à <link> : icon, pingback, stylesheet)
target	_blank, _parent, _self, _top <i>nom d'un élément de type iframe (attribut name)</i>	Attribue un contexte de navigation dans lequel suivre le lien.
type <nouveau>	type MIME	Le type MIME de la destination du lien.
download <nouveau>	nom du fichier après téléchargement ou (vide)	Indique que la ressource liée est prévue pour être téléchargée. Si une valeur est donnée à l'attribut, elle représente le nom du fichier après téléchargement.

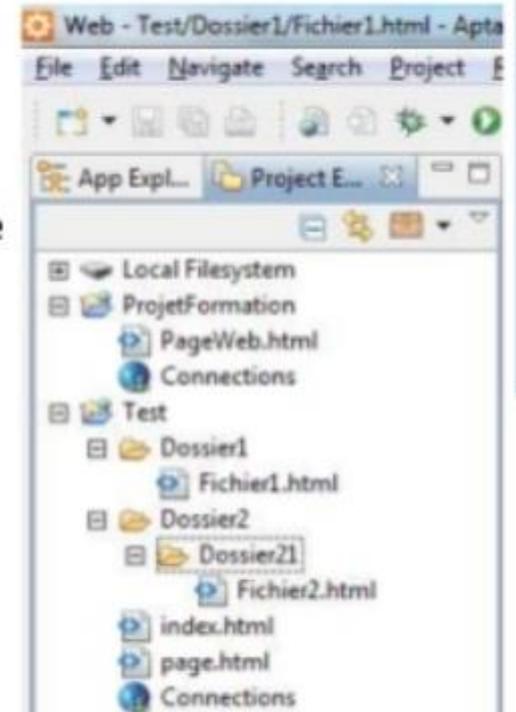


URLs absolues et URLs relatives

- Chaque ressource dans le web a un identifiant unique qui s'appelle URL (Unified Resource Locator)
- Une **URL absolue** commence par le nom du domaine du site contenant la ressource
- Quand les pages appartiennent au même site, on peut faire usage des **URLs relatives**
- Les **URLs relatives** ne commencent pas par le nom du domaine mais indiquent une position par rapport à la page en cours

URLs absolues et URLs relatives- Exemple

```
<a href="page.html">Une page dans le même dossier</a> <!-- index.html -->  
<a href="Dossier1/Fichier1.html">Une page dans un sous-dossier</a> <!-- index.html -->  
<a href="Dossier2/Dossier21/Fichier2.html">Une page dans un autre sous-dossier</a>  
<a href="Dossier1/Fichier1.html">Une page dans un sous-dossier</a> <!-- index.html -->  
<a href="../index.html">Lien vers un dossier parent</a><!-- Fichier1.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.
- Exemple

```
<body>  
Contactez_moi en cliquant <a href: "mailto: othdil@yahoo.fr ">ici </a>  
</body>
```



Contenu embarqué

- On désigne par contenu embarqué tout élément faisant référence à des données stockées dans un fichier ou une ressource externe au document HTML.
- Il s'agit d'images, d'autres documents HTML, de médias vidéo et d'audio.
- Les nouveaux éléments `<audio>`, `<video>`, `<source>`, méritent une attention particulière.
- Il en va de même pour `<canvas>` qui nécessite un développement poussé de ses fonctionnalités ainsi que des instructions JavaScript.



- `` embarque une image stockée dans un fichier externe à l'intérieur d'un document web.
- Les images ne sont pas intégrées dans ce même fichier HTML, mais stockées dans des fichiers voisins (dans le même répertoire ou non) avec différents formats de compression, possédant chacun des avantages et inconvénients.
- Actuellement, trois formats graphiques sont utilisés sur le Web : GIF, PNG et JPEG, qu'on voit plus souvent sous la forme abrégée JPG.
 - GIF signifie Graphics Interchange Format.
 - PNG se prononce « ping » et signifie Portable Network Graphics.
 - Quant à JPEG, il signifie Joint Photographic Experts Group.
- Grosso modo, GIF et PNG conviennent mieux aux images de type logo ou bande dessinée, alors que JPG sera à privilégier pour des images de type photo.



- Quelques cas concrets dans lesquels il est recommandé d'utiliser `` :
 - un logo cliquable, associé au titre principal ;
 - une photo d'illustration pour un article ou dans un album photo ;
 - des icônes cliquables pour des commandes ou des liens.

- Quelques cas concrets dans lesquels il est recommandé d'utiliser les CSS :
 - images de fond en général (zone de contenu, de navigation, de formulaires) ;
 - images et icônes décoratives, séparations verticales ou horizontales.



Attributs spécifiques à

Attribut	Valeurs	Rôle
<code>src</code>	URL	Adresse du fichier image.
<code>alt</code>	texte	Texte alternatif à utiliser en cas d'absence de la ressource, d'incapacité de l'agent utilisateur à rendre l'image, ou de synthèse vocale.
<code>width</code>	nombre entier positif	Largeur de l'image en pixels.
<code>height</code>	nombre entier positif	Hauteur de l'image en pixels.
<code>ismap</code>	<code>ismap</code> ou "" ou (vide)	Indique que l'élément image donne accès à une <i>image map</i> côté serveur.
<code>usemap</code>	chaîne de texte	Nom de l'élément <code><map></code> à utiliser pour la définition des zones cliquables de l' <i>image map</i> côté client.



src

- L'attribut `src` est la pierre angulaire de l'image, car il indique l'adresse à laquelle le navigateur doit la télécharger avant de l'afficher.
- Exemple

```
<!-- Image dans le même répertoire que la page HTML -->  
  
<!-- Image stockée à la racine du site web -->  
  
<!-- Image située sur un autre site -->  

```



Alt, width, height

- Le texte alternatif d'une image est spécifié par l'attribut alt. Il faut le considérer comme un contenu de repli lorsque l'image ne peut être visualisée.
- Soit parce qu'elle n'a pas été trouvée et téléchargée, soit parce que le navigateur ne peut comprendre son format, soit parce que l'utilisateur n'est pas en mesure de la voir.
- Les attributs **width** et **height** définissent respectivement l'espace d'affichage alloué pour l'image. Idéalement, ceux-ci reflètent les dimensions réelles du fichier image, en pixels. Si tel n'est pas le cas, l'image est redimensionnée par le navigateur.
- Si aucune dimension n'est attribuée grâce à **width** ou **height**, le navigateur alloue automatiquement l'espace requis d'après la résolution native du fichier image.
- Il est souvent recommandé de préciser ces valeurs de largeur et hauteur afin d'indiquer dès le chargement du code HTML les dimensions de l'espace à réserver.
- Le navigateur n'aura pas besoin d'attendre le chargement ultérieur de la ressource image pour en déduire ses propriétés.

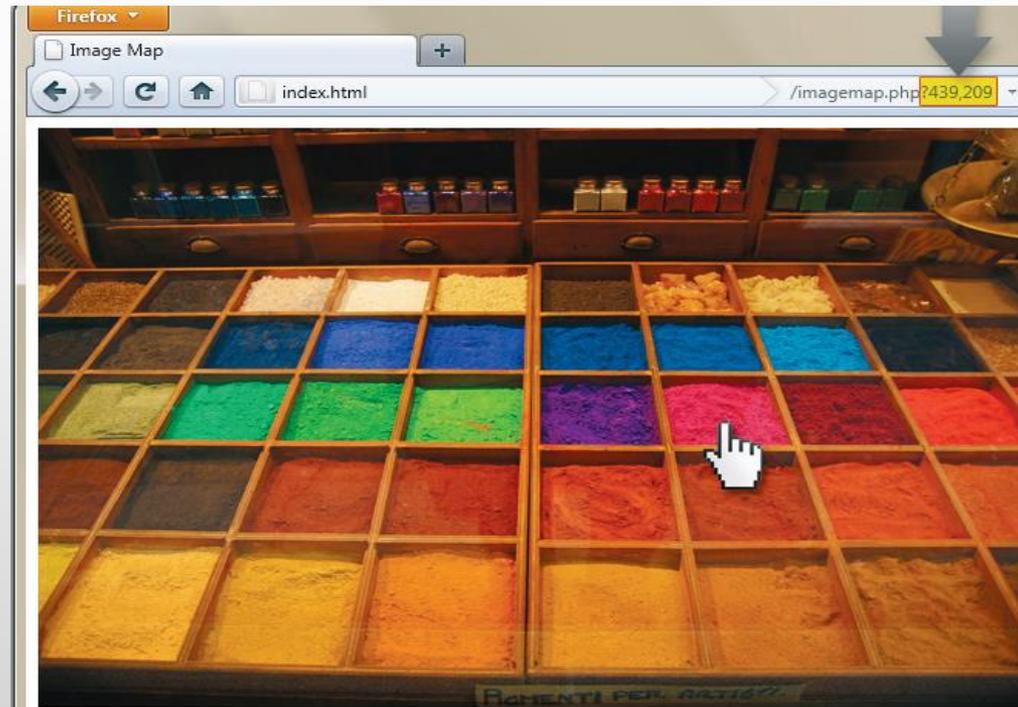


Usemap et ismap

- Une image peut être associée à la définition d'une zone de liens cliquables (ou image map), telle que décrite avec l'élément `<map>`. Cet attribut contient alors l'identifiant unique de ce dernier pour y faire appel.
- La valeur booléenne (`true` ou `false`) de l'attribut `ismap` indique que le navigateur doit transmettre les coordonnées du point cliqué sur l'image au serveur, lorsque celle-ci est bien entendu cliquable (par exemple dans un lien `<a>`). Ces coordonnées sont envoyées sous la forme `?x,y` et peuvent être analysées par langage interprété côté serveur.



- ``





Liens sur images

- Les liens hypertextes appliqués aux images revêtent les mêmes caractéristiques que les liens sur le contenu texte. L'intégralité de la surface de l'image est alors cliquable.
- Pour des raisons d'accessibilité, il est très fortement recommandé de toujours préciser un texte alternatif comme valeur de l'attribut alt. Exemple:

```
<p>  
<a href="hd/chat1.jpg" title="Voir la photo HD"></a>  
<a href="hd/chat2.jpg" title="Voir la photo HD"></a>  
<a href="hd/chat3.jpg" title="Voir la photo HD"></a>  
<a href="hd/chat4.jpg" title="Voir la photo HD"></a>  
</p>
```



<map>

- Un élément `<map>` est utilisé en conjonction avec une image `` pour concevoir une carte cliquable appliquée sur celle-ci.
- Le résultat consiste en des zones distinctes et transparentes, de formes variées, équipées de liens.
- L'attribut `name` est requis pour cet élément, car il permet de l'associer à l'élément image `` via son attribut `usemap`.
- Une image map contient un ou plusieurs éléments `<area>` qui définissent chacun une zone cliquable d'après une forme (`shape`), des coordonnées pour cette forme (`coords`) et une adresse (`href`).



■ Principes

- Association d'une carte à une image
``

■ Image-map

- Carte qui référence des zones d'une image, ainsi que les liens déclenchés sur chacune de ces parties d'images.

- Définition d'une carte

- `<MAP NAME="nom de la carte">`
 `<AREA ...>`
 `[<AREA ...>]`
`</MAP>`



< AREA ... >

```
■ <AREA  SHAPE=" rect | circle | poly | default "  
    COORDS="x1,y1,x2,y2  
            | x,y,r  
            | x1,y1,x2,y2,x3,y3 ..."  
    HREF="Url cible" >
```

■ SHAPE=" ... "

- rect → zone rectangulaire
- circle → disque
- poly → zone polygonale
- default → zone par défaut,
= toute la partie de l'image non recouverte par
une zone définie.



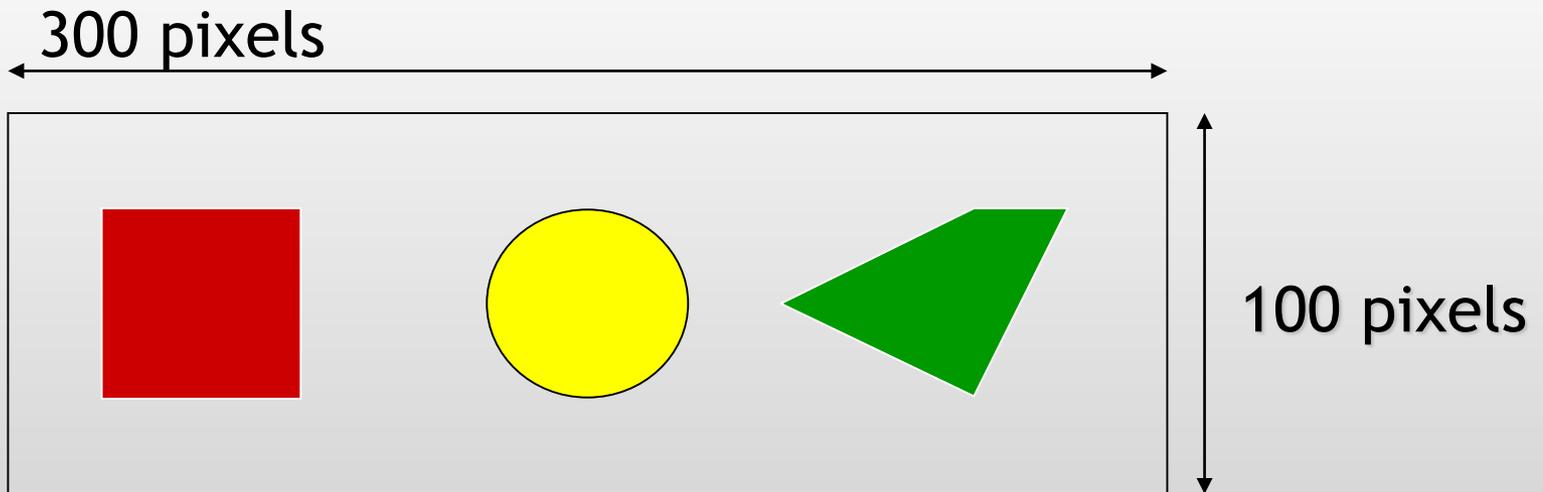
< AREA ... >

- **COORDS="x₁,y₁,x₂,y₂"**
 - Utilisé si SHAPE="rect"
 - (x₁, y₁) défini le coin supérieur gauche de la zone rectangulaire,
(x₂, y₂) défini le coin inférieur droit.
- **COORDS="x,y,r"**
 - Utilisé si SHAPE="circle"
 - (x, y) désigne le centre du cercle, r désigne le rayon.
- **COORDS="x₁,y₁,x₂,y₂, ... x_n,y_n "**
 - Utilisé si SHAPE="poly"
 - (x_i, y_i) définissent les point successifs du polygone.
 - Le polygone est fermé :
(x₁, y₁) est automatiquement relié à (x_n, y_n)



< AREA ... >

```
<MAP NAME="carte_image">  
  <AREA      SHAPE="rect"    COORDS="25,25,75,75"  
             HREF="rectangle.html">  
  <AREA      SHAPE="circle"  COORDS="150,50,25"  
             HREF="cercle.html">  
  <AREA      SHAPE="poly"    COORDS="200,50,250,75,275,25,250,25"  
             HREF="polygone.html">  
</MAP>
```





<map>

- Exemple d'usage <map> pour :

```
  
  <map name="nerdmap">  
    <area shape="rect" coords="264,233,404,423" alt="N" href="n.html">  
    <area shape="rect" coords="410,232,540,412" alt="E" href="e.html">  
    <area shape="rect" coords="545,229,657,398" alt="R" href="r.html">  
    <area shape="rect" coords="663,218,780,390" alt="D" href="d.html">  
  </map>
```



Figure: Mise en évidence des zones cliquables



<figure>

- Il permet d'associer une légende optionnelle `<figcaption>` à du contenu regroupé en un bloc, qui peut être indépendant du contenu principal du document. Cela signifie que l'ensemble peut être placé à part, sur une page dédiée ou une annexe sans modifier de façon majeure la signification principale du texte.
- En d'autres termes, il permet d'enrichir une image, un graphique, une capture d'écran, par une légende.
- Des contenus plus variés peuvent être concernés, avec les éléments `<video>`, `<audio>`, `<canvas>`, des blocs de code `<code>` ou des tableaux `<table>` complets.



<figure>

- Exemple d'usage de l'élément <figure>:

```
<!-- Une image sans légende -->  
<figure>  
    
</figure>
```

- Autre exemple d'usage de <figure>

```
<!-- Plusieurs images avec une légende -->  
<figure>  
    
    
    
  <figcaption>De gauche à droite plusieurs fruits : une fraise, une  
goyave et un kiwi</figcaption>  
</figure>
```



<figcaption>

- Le rôle de `<figcaption>` est de conférer une légende à l'élément `<figure>`.
- Son contenu autorisé est un contenu de type flux, c'est-à-dire du texte, mais aussi d'autres éléments de flux tels que des liens pouvant cibler une page avec des explications supplémentaires ou une référence à l'auteur de l'illustration.
- Une légende peut figurer dans le code avant ou après le contenu dans `<figure>`.

Exemple d'usage de l'élément `<figcaption>`

```
<figure>  
  <figcaption>Une photo du New York Water Taxi près de Brooklyn Bridge  
</figcaption>  
    
</figure>
```



Données tabulaires: table

- Un élément `<table>` est destiné à contenir des données tabulaires. Par cela, on désigne un ensemble d'informations qui peuvent ou qui doivent être présentées dans une structure formatée en colonnes et en lignes.
- Différents éléments enfants peuvent apporter une valeur sémantique au contenu : en-tête et titre de tableau, intitulés de colonnes ou de lignes, etc.
- Entre la balise ouvrante `<table>` et la balise fermante `</table>`, peuvent être placés :



table

Tableau Imbrications pour <table>

Élément	Occurrences	Rôle
<caption>	0 ou 1	Titre du tableau.
<colgroup>	0 ou +	Propriétés pour un groupe de colonnes.
<thead>	0 ou 1	En-tête du tableau.
<tfoot>	0 ou 1	Pied du tableau.
<tbody>	0 ou +	Corps du tableau (contenant <code>tr</code> , <code>th</code> , <code>td</code>).
ou <tr>	1 ou +	Lignes du tableau (contenant <code>th</code> , <code>td</code>).



Exemple d'utilisation

```
<table>
  <tr>
    .....
    <td>Adil</td>
    <td>33 ans</td>
    <td>Maroc</td>
  </tr>
  <tr>
    .....
    <td>Michelle</td>
    <td>26 ans</td>
    <td>États-Unis</td>
  </tr>
</table>
```



<TABLE> ... </TABLE>

- `<TABLE WIDTH="w|w%"
BORDER="b"
CELLPADDING="cp"
CELLSPACING="cs">`
- `WIDTH="w|w%"`
 - Largeur du tableau exprimée en pixel ou en pourcentage de la largeur de la fenêtre (à utiliser avec prudence).
- `BORDER="b"`
 - Largeur de l'encadrement du tableau et des cellules exprimée en pixels.
 - Si une cellule est vide, il n'apparaît pas.
- `CELLPADDING="cp"`
 - Marge intérieure de chaque cellule (en pixels).
- `CELLSPACING="cs"`
 - Espacements horizontal et vertical entre les cellules du tableau



<thead>

- Un élément `<thead>` représente un en-tête de tableau, utilisé en conjonction avec `<tbody>` et `<tfoot>`.
- Il doit être placé après `<caption>` et `<colgroup>` s'il y a lieu, et avant `<tbody>` ou `<tfoot>`.
- Son contenu est constitué d'enfants `<tr>`, eux-mêmes constitués d'un ou plusieurs `<th>`.
- La spécification HTML n'autorise pas d'y placer des cellules classiques `<td>`, car il ne s'agit pas de cellules pouvant représenter une tête de tableau.



<tfoot>

- Un élément `<tfoot>` représente un pied de tableau, utilisé en conjonction avec `<thead>` et `<tbody>`. On peut y placer zéro ou plusieurs `<tr>`.
- Attention : `<tfoot>` doit apparaître avant `<tbody>` dans le code afin de fournir au navigateur les renseignements qu'il détient, avant de charger le reste du contenu.



<tbody>

- Un élément `<tbody>` représente un corps de tableau, utilisé en conjonction avec `<thead>` et `<tfoot>`.
- Il regroupe les lignes `<tr>` qui contiennent les données du tableau. Dans ce cas de figure, aucun élément `<tr>` ne peut être directement enfant de `<table>`.
- Les navigateurs peuvent exploiter cette structuration pour agrémenter l'affichage de tableaux, particulièrement s'ils possèdent des dimensions qui excèdent la surface de visualisation, en maintenant ces éléments visibles à l'écran indépendamment des lignes, durant le défilement du document.



Les lignes : `<TR> ... </TR>`

- Un élément `<tr>` (table row) représente une ligne de tableau. Il contient un ou plusieurs éléments `<td>` et/ou `<th>`.
- Un élément `<tr>` peut être enfant de :
 - `<thead>`, et dans ce cas, il ne doit contenir que des `<th>` ;
 - `<tfoot>` ;
 - `<tbody>` ;
 - `<table>` lorsqu'il n'y a pas d'élément `<tbody>`, après d'éventuels frères `<caption>`, `<colgroup>` et `<thead>`.



Les lignes : `<TR> ... </TR>`

- `<TR ALIGN="LEFT|CENTER|RIGHT|JUSTIFY" VALIGN="TOP|MIDDLE|BOTTOM">`
- `ALIGN="LEFT|CENTER|RIGHT|JUSTIFY"`
 - Alignement horizontal du contenu de toutes les cellules de la ligne :
 - **LEFT** → à gauche (attribut par défaut)
 - **CENTER** → au centre
 - **RIGHT** → à droite
 - **JUSTIFY** → justifié (à utiliser avec prudence !)
- `VALIGN="TOP|MIDDLE|BOTTOM">`
 - Alignement vertical du contenu de toutes les cellules de la ligne :
 - **TOP** → haut de la cellule
 - **MIDDLE** → centre verticalement (attribut par défaut)
 - **BOTTOM** → bas de la cellule



Les cellules : `<TD>` ... `</TD>`

- Une cellule `<td>` (table data) peut contenir du texte et beaucoup d'autres éléments HTML tels que des liens, des images, des listes, etc.
- Elle est toujours imbriquée dans un élément `<tr>`.

```
<table>
  <tr>
    <td>Kangourou</td>
    <td>marsupial</td>
    <td>mammifère</td>
  </tr>
</table>
```



Les cellules : `<TD>` ... `</TD>`

- `<TD` `ALIGN="LEFT|CENTER|RIGHT|JUSTIFY"`
`VALIGN="TOP|MIDDLE|BOTTOM"`
`COLSPAN="c"`
`ROWSPAN="r"`
`WIDTH="w|w%"``>`
- `ALIGN="LEFT|CENTER|RIGHT|JUSTIFY"`
 - Alignement horizontal cellule par cellule (cf. `<TR>` ... `</TR>`).
- `VALIGN="TOP|MIDDLE|BOTTOM"`
 - Alignement vertical cellule par cellule (cf. `<TR>` ... `</TR>`).
- `WIDTH="w|w%"`
 - Largeur de la cellule en pixels ou en pourcentage de la largeur du tableau.
 - A spécifier une seule fois dans le tableau (ie pour une seule ligne) car elle détermine la largeur de la colonne dont fait partie la cellule.
 - Quand cette option est omise, la largeur des cellules est calculée par rapport à celle du tableau, aux espacements et la largeur de la cellule contenant le plus de texte, colonne par colonne).



Les cellules : `<TD> ... </TD>`

■ `COLSPAN="c"`

- Nombre de cellules fusionnées à l'horizontal. Permet de disposer du texte au dessus de plusieurs colonnes.
- Pour que le tableau s'affiche correctement, il faut que ligne par ligne, le nombre de cellules soit le même. Dans ce cas où une cellule est une cellule fusionnée (avec un `COLSPAN`), elle compte pour autant de cellule que la valeur du `COLSPAN`.

■ `ROWSPAN="r"`

- idem pour
- les fusions à la verticale

<code><TD ROWSPAN=3></code>	
	<code><TD COLSPAN=2></code>

Un exemple de tableau complexe

TABLE WIDTH="450" CELLSPACING="0" CELLPADDING="6" BORDER="1"		
TD WIDTH="100"	TD WIDTH="100"	TD WIDTH="100"
TD COLSPAN="3" ALIGN="center" Cellule fusionnée (3 colonnes) Texte centré en hauteur et en largeur		
TD ROWSPAN="2" WIDTH="100" VALIGN="top" Cellule fusionnée (2 lignes). Texte aligné à gauche et en haut	TD WIDTH="100"	TD WIDTH="100"
	TD COLSPAN="2" ALIGN="right" VALIGN="bottom" Cellule fusionnée (2 colonnes) Texte aligné à droite et en bas.	



<th>

- Un élément `<th>` (table header) est une cellule spécialisée qui joue le rôle d'en-tête de colonne (ou de ligne) dans un tableau. Il doit être imbriqué dans une balise `<tr>`.

```
<table>
  <tr>
    <th scope="col">Animal</th>
    <th scope="col">Groupe</th>
    <th scope="col">Classe</th>
  </tr>
  <tr>
    <td>Kangourou</td>
    <td>marsupial</td>
    <td>mammifère</td>
  </tr>
  <tr>
    <td>Baleine</td>
    <td>cétacé</td>
    <td>mammifère</td>
  </tr>
</table>
```



Les frames (ou cadres)

- ❖ Objectif
 - Permettent d'obtenir une ou plusieurs divisions horizontales et/ou verticales de la fenêtre du navigateur, et ainsi de disposer "virtuellement" de plusieurs fenêtres.
- ❖ Utilisation
 - Au moins une des divisions est généralement employée pour afficher un menu de parcours du site.
 - Utilisables avec tous les navigateurs dont la version > 3.0
- ❖ Pour créer une page contenant des frames
 - Définir le découpage (FRAMESET).
 - Définir le contenu des cadres (pages HTML classiques).



<FRAMESET> ... </FRAMESET>

- `<FRAMESET ROWS|COLS="d1[,d2,d3...],*" BORDER="b" FRAMEBORDER="YES|NO" FRAMESPACING="fs">`

- `ROWS|COLS="d1[,d2,d3...],*"`

- ROWS → division en **lignes** (horizontale)
- COLS → pour une division en **colonnes** (verticale)
- d1[,d2, d3...] : indique la hauteur (largeur) en pixel de chaque ligne (colonne) ; le séparateur de valeurs est la virgule.
 - * permet d'attribuer à la dernière ligne (colonne) tout l'espace restant.
- Possibilité d'indiquer une valeur en pourcentage de la taille de la page. Attention : l'affichage dépendra de la taille de la fenêtre du navigateur !



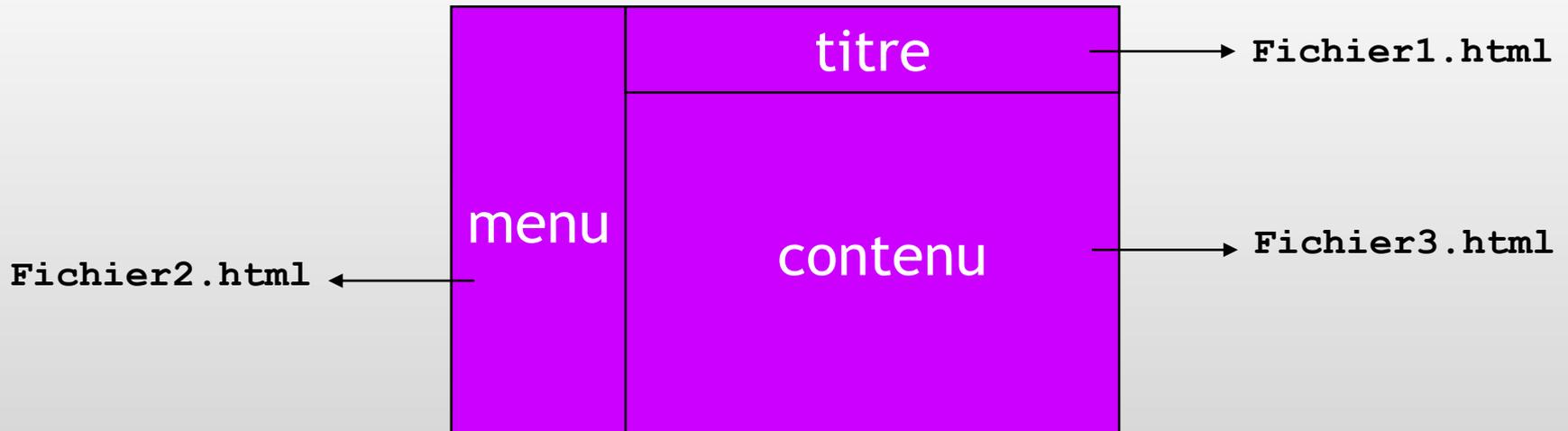
<FRAMESET> ... </FRAMESET>

- Désactiver l'affichage des bordures entre les frames
 - Nécessite d'utiliser les 3 options :
 - **BORDER=0**
 - **FRAMEBORDER=NO**
 - **FRAMESPACING=0**
 - en même temps de manière à ce que toutes les versions de Netscape Navigator et Internet Explorer se comportent de manière identique.
- Contenu de `<FRAMESET> ... </FRAMESET>`
 - Chaque FRAMESET contient autant d'éléments `<FRAME>` que d'arguments dans l'option ROWS ou COLS.
 - Chaque `<FRAME>` indique quelle page afficher.

Exemple de découpage en frame

Fichier frame.html :

```
<FRAMESET cols="70,424">  
  <FRAME name="menu" src="Fichier1.html">  
  <FRAMESET rows="69,487">  
    <FRAME name="titre" src="Fichier2.html">  
    <FRAME name="contenu" src="Fichier3.html">  
  </FRAMESET>  
</FRAMESET>
```





<FRAME ... >

- <FRAME NAME="nom de la frame" SRC="url"
SCROLLING="YES|NO|AUTO NORESIZE
FRAMEBORDER="YES|NO" BORDER="b"
FRAMESPACING="fs"
MARGINWIDTH="mw"
MARGINHEIGHT="mh">
- SRC= "url"
 - URL de la page chargée dans la fenêtre
- SCROLLING="YES|NO|AUTO"
 - Gère l'affichage des ascenseurs
 - YES : toujours présents
 - NO : jamais
 - AUTO : affichés si nécessaire

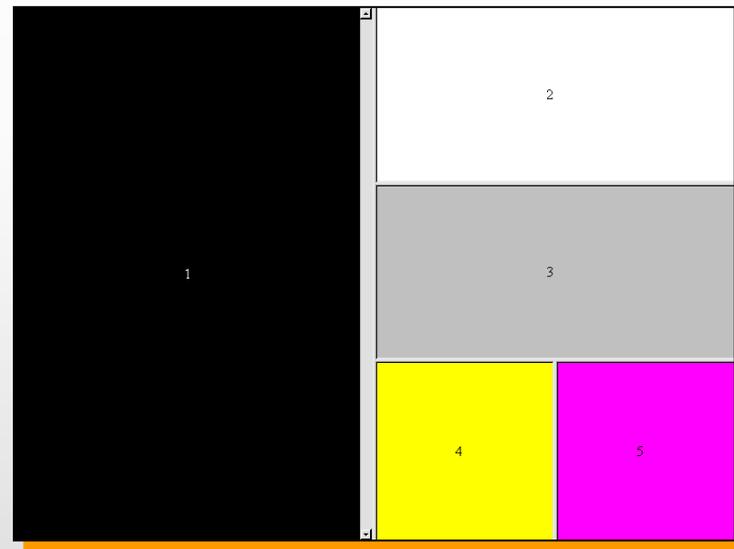


<FRAME ... >

- **NAME="nom de la frame"**
 - Désignation utilisée pour identifier chaque cadre.
 - Sert à désigner le cadre dans lequel va s'afficher un document pointé par un lien hypertexte.
 - Option **TARGET** (dans un élément **A**) utilisée avec **HREF**.
`< A TARGET="nom du cadre de destination" HREF="url" >`
 - L'absence de **TARGET** dans un élément **A** provoque l'affichage dans le cadre qui contient le lien.
 - Nom de cadre prédéfinis : `_parent`, `_top`, `_blank`, ...
- Exemple
 - `< A HREF="lien.html" TARGET="contenu">lien`

Exemple de découpages multiples

- Deux façons de le faire
 - Découpages imbriqués dans un seul fichier
 - Découpages dans plusieurs fichiers



Exemple de découpages multiples

■ Découpage unique imbriqué

```
<FRAMESET COLS="50%,*" BORDER="2" FRAMEBORDER="YES"
FRAMESPACING="2">
```

```
<FRAME NAME="1" SRC="1.html" SCROLLING="yes">
```

```
<FRAMESET ROWS="33%,33%,*">
```

```
<FRAME NAME="2" SRC="2.html" SCROLLING="auto">
```

```
<FRAME NAME="3" SRC="3.html" SCROLLING="auto">
```

```
<FRAMESET COLS="50%,*">
```

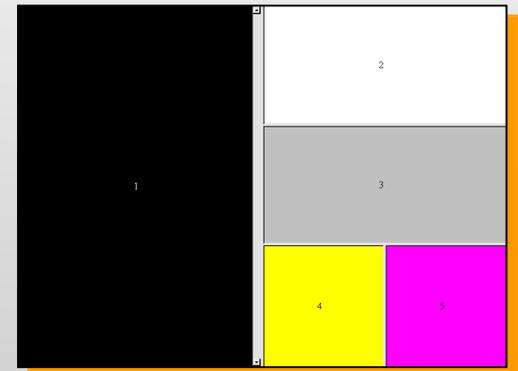
```
<FRAME NAME="4" SRC="4.html" SCROLLING="auto">
```

```
<FRAME NAME="5" SRC="5.html" SCROLLING="auto">
```

```
</FRAMESET>
```

```
<FRAMESET>
```

```
</FRAMESET>
```



Exemple de découpages multiples

■ Découpages en plusieurs fois

frameset.html

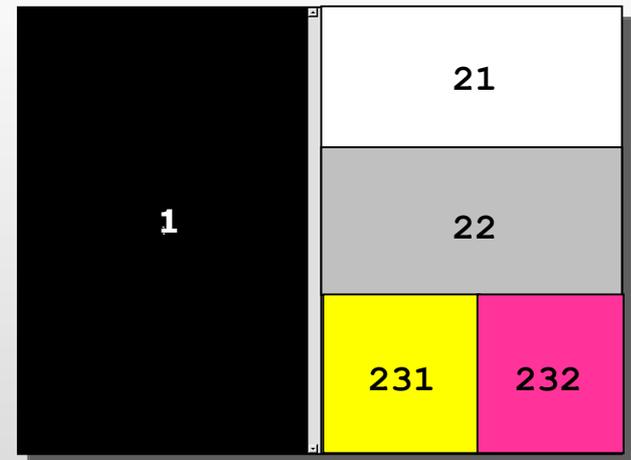
```
<FRAMESET COLS="50%,*" BORDER="2" FRAMEBORDER="YES" >
  <FRAME NAME="1" SRC="f1.html" SCROLLING="yes">
  <FRAME NAME="2" SRC="f2.html" >
</FRAMESET>
```

f2.html

```
<FRAMESET ROWS="33%,33%,*">
  <FRAME NAME="21" SRC="f21.html">
  <FRAME NAME="22" SRC="f22.html">
  <FRAME NAME="23" SRC="f23.html">
</FRAMESET>
```

f23.html

```
<FRAMESET COLS="50%,*">
  <FRAME NAME="231" SRC="4.html" >
  <FRAME NAME="232" SRC="5.html" >
  </FRAMESET>
```





Découpage unique vs. découpages en plusieurs fois

■ Découpage unique

- **Avantage** : facilite la maintenance car il n'y a qu'un seul fichier FRAMESET.
- **Inconvénient** : impossibilité de remettre simultanément à jour plusieurs cadres.

■ Intérêt du découpage en plusieurs fois

- **Avantage** : souplesse de mise à jour des cadres car il est possible de désigner des cadres isolés ou des ensembles de cadres.
- **Inconvénient** : lourd à maintenir.



<iframe>

- L'élément `<iframe>` embarque dans la page un autre document HTML.
- Il s'agit d'un contexte de navigation imbriqué dans un «cadre».

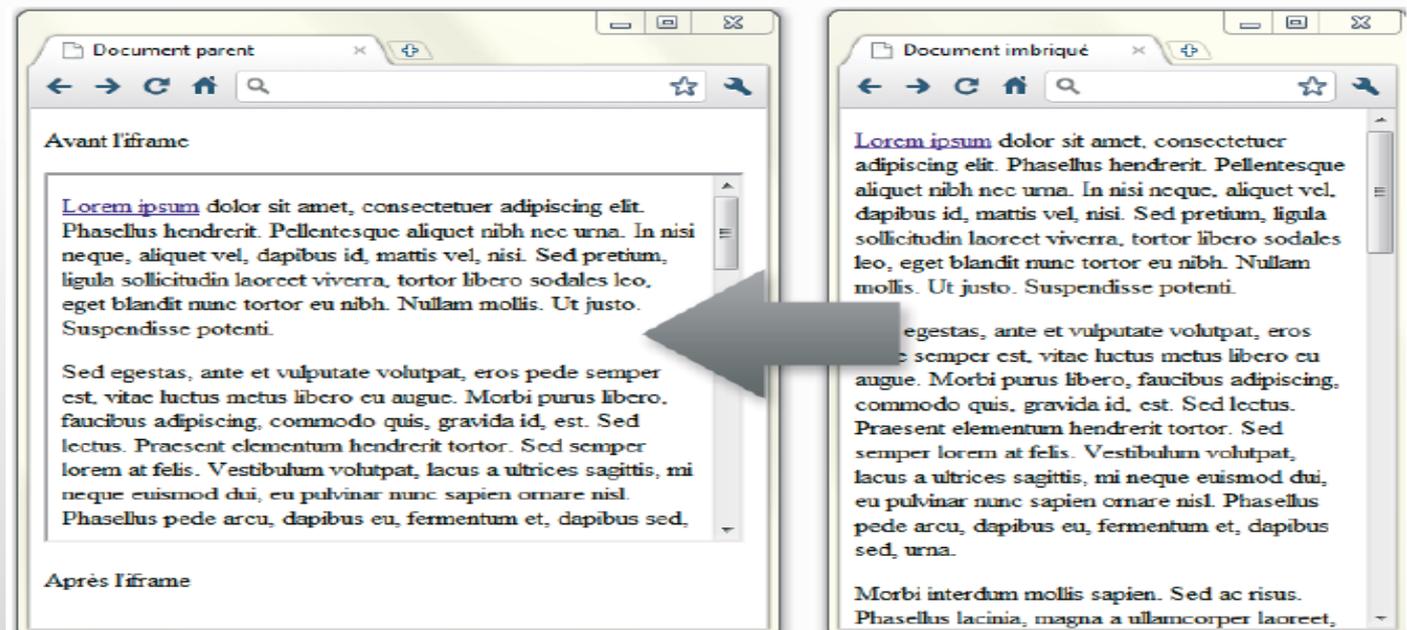


Figure Affichage d'un élément `<iframe>`



- `<IFRAME NAME="nom de la frame" SRC="url" HEIGHT="h" WIDTH="w" ALIGN="left|center|right|justify" SCROLLING="YES|NO|AUTO" FRAMEBORDER="YES|NO" MARGINWIDTH="mw" MARGINHEIGHT="mh">`
- `NAME="nom de la frame"`
 - Identifiant de la fenêtre pour l'utiliser dans les TARGET
- `SRC="url"`
 - URL de la page chargée dans la fenêtre



Définition d'une fenêtre dans une page

- **HEIGHT="h"**

- Hauteur de la fenêtre en pixel

- **WIDTH="w"**

- Largeur de la fenêtre en pixel

- **ALIGN="left|center|right|justify"**

- Alignement de la fenêtre dans le document qui la contient

- **SCROLLING, FRAMEBORDER, MARGINWIDTH, MARGINHEIGHT**

- Idem que dans FRAME



Formulaires

- Objectif
 - Interagir avec l'utilisateur en permettant d'obtenir des réponses de l'utilisateur
- Balise

```
<FORM          METHOD="GET|POST"  
    ACTION="Programme" >  
    ...  
</FORM>
```

- **METHOD / ACTION**

- Programme = méthode de traitement des informations recueillies dans le questionnaire.
- Dépendant des possibilités offertes par l'hébergeur des pages.
- Les valeurs à indiquer pour ces options sont à remplir suivant les indications de l'hébergeur.



<INPUT ... >

■ Rôle

- Permet de définir les différents types de champ d'un formulaire :
 - zone de saisie,
 - case à cocher,
 - bouton radio,
 - bouton de confirmation,
 - bouton de remise à zéro des champs.

■ Différentes syntaxes

- Le contenu des attributs est dépendant du type de champ.



<input> et ses variantes

- L'élément `<input>` est particulier, car il revêt de multiples apparences et fonctionnalités, uniquement caractérisées par l'attribut `type`.
- Son but est de recueillir les informations d'une entrée effectuée par l'utilisateur, éventuellement en lui fournissant toutes les indications nécessaires pour structurer son entrée ou son choix.
- Afin de pouvoir enrichir les éléments de formulaire en HTML, des nouveaux types ont été ajoutés pour `<input>` en HTML 5. Voici un récapitulatif des types pratiqués depuis les précédentes versions de HTML ainsi que ceux inaugurés récemment.

Les valeurs possibles pour l'attribut type de la balise <input>

Tableau Récapitulatif des valeurs possibles pour l'attribut type de la balise <input>

Type	Rôle	HTML 5
text	Champ de texte	
password	Champ mot de passe	
hidden	Champ caché (invisible)	
radio	Bouton radio (un seul choix)	
checkbox	Case à cocher (choix multiples)	
button	Bouton	
reset	Remise à zéro du formulaire	
submit	Bouton de validation du formulaire	
image	Image cliquable	
file	Fichier	

Les valeurs possibles pour l'attribut type de la balise <input> (suite)

Type	Rôle	HTML 5
tel	Numéro de téléphone	Nouveau
url	Adresse URL/URI	Nouveau
email	Adresse e-mail	Nouveau
search	Champ de recherche	Nouveau
date	Date	Nouveau
time	Heure	Nouveau
datetime	Date et heure	Nouveau
datetime-local	Date et heure (locales)	Nouveau
month	Mois	Nouveau
week	Semaine	Nouveau
number	Valeur numérique	Nouveau
range	Valeur numérique d'un intervalle, sans grande précision	Nouveau
color	Une couleur RVB (3 composantes de 8 bits)	Nouveau



<input> et ses variantes

- Un élément sans attribut **type** est considéré par défaut comme `<input type="text">`.
- Dans tous les cas de figure, l'attribut **value** détermine la valeur du champ transmise à la validation du formulaire, et l'attribut **name** son nom afin de retrouver quel champ correspond à quelle valeur lors du traitement des données après leur envoi via HTTP.
- Il est absolument obligatoire de vérifier et de valider les données au préalable du côté du serveur avant de les utiliser.



<INPUT ... >

- <INPUT NAME="nom de la zone"
 TYPE="TEXT
 | CHECKBOX
 | RADIO
 | SUBMIT
 | IMAGE
 | RESET"
 VALUE="valeur"
 CHECKED
 SRC="url"
 SIZE="taille"
 MAXLENGTH="nb" >



<INPUT ... >

- **VALUE="valeur"**
 - valeur correspond au :
 - au contenu initial de la zone de saisie (*si TYPE="TEXT"*)
 - au libellé du bouton (*si TYPE="SUBMIT" ou "RESET"*)
 - à la valeur retournée (*si TYPE="CHECKBOX" ou "RADIO"*)
- **CHECKED**
 - Dans le cas d'un bouton à cocher, indique qu'il apparaît coché par défaut
- **SRC="url"**
 - Utilisé avec *TYPE="IMAGE"*
 - Adresse d'une image à afficher, celle-ci jouant le rôle de bouton de validation
- **SIZE="taille"**
 - Utilisé avec *TYPE="TEXT"*
 - Correspond à la taille de la zone de saisie.



<INPUT TYPE="TEXT" ... >

■ Exemple

- ```
<INPUT TYPE="TEXT"
NAME="case_texte_1"
VALUE="Texte par défaut"
SIZE="20"
MAXLENGTH="80" >
```

## ■ Principes

- La case fait 20 caractères de large.
- Il est possible d'y saisir 80 caractères au maximum.

Texte par défaut



# Les attributs des champs de formulaire

- **value** : valeur par défaut du champ
- **placeholder** : indication grisée dans le champ (disparaît au clic)
- **readonly** : en lecture seule, ne peut être modifié par l'utilisateur
- **disabled** : champ non pris en compte, ne peut être modifié par l'utilisateur
- **maxlength** : nombre de caractères autorisé
- **autofocus** : champ sélectionné au chargement de la page
- **required** : champ obligatoire
- **min** et **max** : minimum et maximum pour les champs de type number



# Exemples de types input

```
<HTML>
```

```
 <HEAD><TITLE>Exemple de fichier HTML</TITLE></HEAD>
```

```
<SCRIPT>
```

```
</SCRIPT>
```

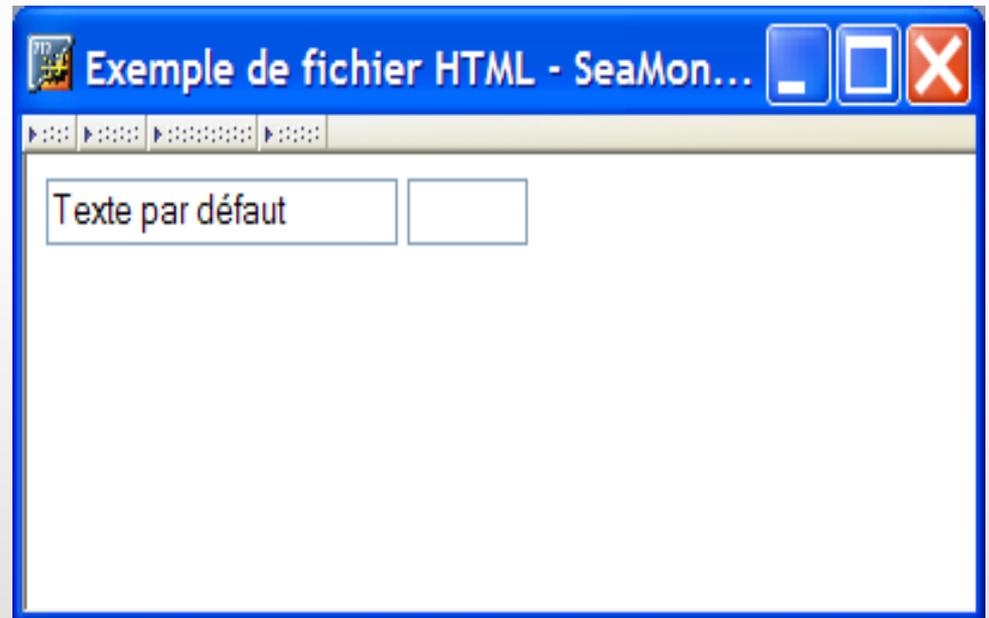
```
<BODY>
```

```
<INPUT TYPE="TEXT"
 NAME="case_texte_1"
 VALUE="Texte par défaut"
 SIZE="20"
 MAXLENGTH="80" >
```

```
<INPUT TYPE="TEXT"
 NAME="case_texte_2"
 VALUE=""
 SIZE="4"
 MAXLENGTH="4" >
```

```
</BODY>
```

```
</HTML>
```





# Exemples de types input

Choix 1    <INPUT TYPE="CHECKBOX" NAME="case\_1"  
VALUE="1" CHECKED><BR>

Choix 2    <INPUT TYPE="CHECKBOX" NAME="case\_2"  
VALUE="1"><BR>

Choix 3    <INPUT TYPE="CHECKBOX" NAME="case\_3"  
VALUE="1"><BR>

Choix 4    <INPUT TYPE="CHECKBOX" NAME="case\_4"  
VALUE="1">

Choix 1	<input checked="" type="checkbox"/>
Choix 2	<input type="checkbox"/>
Choix 3	<input type="checkbox"/>
Choix 4	<input type="checkbox"/>

## ■ Principes

- Plusieurs cases peuvent avoir l'option CHECKED
- Les 4 cases n'ont pas le même nom
- VALUE indique la valeur qui sera retournée en fonction de la(les) cases cochées.



# <INPUT TYPE="RADIO" ... >

## ■ Exemple

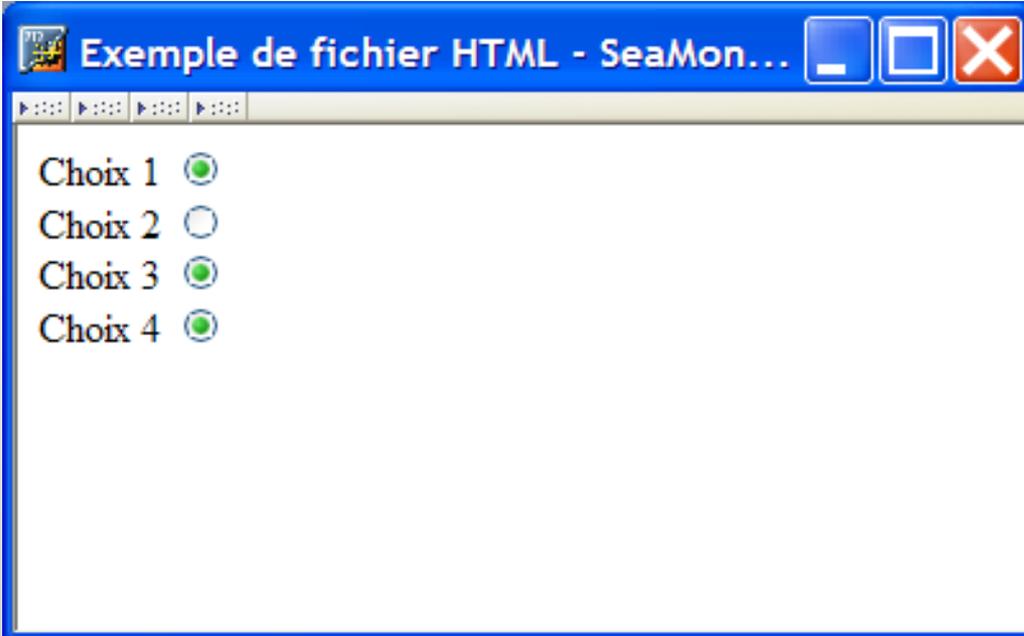
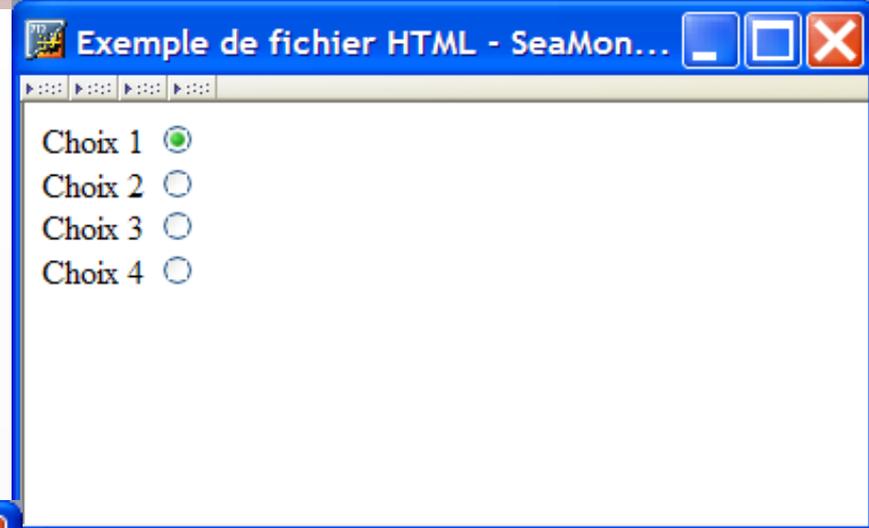
- Choix 1 &nbsp;<INPUT TYPE="RADIO" NAME="BoutonRadio" VALUE="1" CHECKED><BR>
- Choix 2 &nbsp;<INPUT TYPE="RADIO" NAME="BoutonRadio" VALUE="2"><BR>
- Choix 3 &nbsp;<INPUT TYPE="RADIO" NAME="BoutonRadio" VALUE="3"><BR>
- Choix 4 &nbsp;<INPUT TYPE="RADIO" NAME="BoutonRadio" VALUE="4">

## ■ Principes

- Une seule case peut avoir l'option CHECKED
- Les 4 cases ont le même nom
- **VALUE** indique la valeur qui sera retournée en fonction de la case cochée.



Ne peut être décoché



# Bouton de validation et de mise à zéro

- `<INPUT NAME="validation" TYPE="SUBMIT" VALUE="Valider">` 
- **Bouton de validation avec image :**
  - `<INPUT TYPE="IMAGE" NAME="validation" SRC="fluide.jpg">`  
(si l'image *fluide.jpg* est dans le même dossier que la page).
- `<INPUT NAME="Effacement" TYPE="RESET" VALUE="Effacer">` 



# Liste de choix

```
<SELECT NAME="nom de la liste" SIZE="1|s">
 <OPTION ... >texte de la ligne 1</OPTION>
 <OPTION ... >texte de la ligne 2</OPTION>
 ...
</SELECT>
```

SIZE="1|s"

- SIZE="1" → liste déroulante
- SIZE≠"1" → liste défilante.

```
<OPTION ... >texte de la ligne</OPTION>
```

- 3 attributs possibles : DISABLED, SELECTED et VALUE (même rôle que pour les boutons radios)



# Liste de choix

## ■ Exemple de liste déroulante

```
<SELECT NAME="liste1" SIZE="3">
 <OPTION>Ligne 1</OPTION>
 <OPTION>Ligne 2</OPTION>
 <OPTION>Ligne 3</OPTION>
 <OPTION>Ligne 4</OPTION>
 <OPTION>Ligne 5</OPTION>
 <OPTION>Ligne 6</OPTION>
</SELECT>
```



## ■ Exemple de liste déroulante

```
<SELECT NAME="liste1" SIZE="1">
 <OPTION>Ligne 1</OPTION>
 <OPTION>Ligne 2</OPTION>
 <OPTION>Ligne 3</OPTION>
</SELECT>
```





# Liste de choix

```
<HTML>
```

```
 <HEAD><TITLE>Exemple de fichier HTML</TITLE></HEAD>
```

```
<SCRIPT>
```

```
</SCRIPT>
```

```
<BODY>
```

```
 <SELECT NAME="liste1" SIZE="1">
```

```
 <OPTION>Ligne 1</OPTION>
```

```
 <OPTION>Ligne 2</OPTION>
```

```
 <OPTION>Ligne 3</OPTION>
```

```
</SELECT>
```

```
<SELECT NAME="liste2" SIZE="4">
```

```
 <OPTION>Ligne 1</OPTION>
```

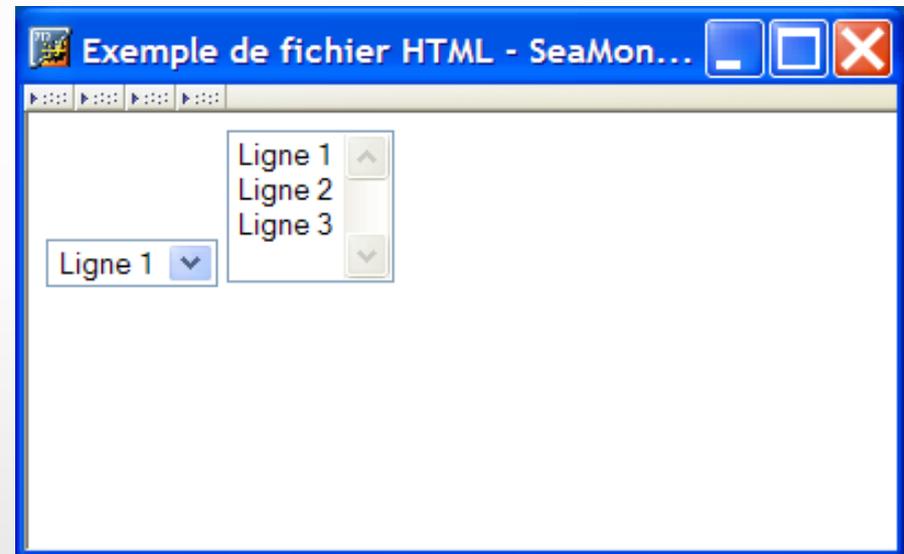
```
 <OPTION>Ligne 2</OPTION>
```

```
 <OPTION>Ligne 3</OPTION>
```

```
</SELECT>
```

```
</BODY>
```

```
</HTML>
```





```
< TEXTAREA ... > ... </ TEXTAREA ...>
```

```
<TEXTAREA NAME="nom de la zone"
 ROWS="r"
 COLS="c">
```



## Rôle

- Permet de définir une zone de texte pour saisir des données de taille plus importante que dans une case de texte.

**ROWS="r"**

- Taille de la zone de saisie en nombre de lignes

**COLS="c"**

- Taille de la zone de saisie en nombre de colonnes.



# Mailto

```

```

...

```

```

## Rôle

- Permet de déclencher l'ouverture d'une nouvelle fenêtre pour l'envoi d'un courrier électronique.
- L'ouverture se réalise lorsqu'on clique sur le lien compris dans l'élément A.
- Le destinataire est *toto@toto.com*
- Le sujet est *bla-bla-bla...*



# Équations mathématiques

$y = a x^y + b \sin x$	<code>&lt;math&gt;y = a x^y + b \sin x&lt;/math&gt;</code>
$y = a X^y + b \sin x$	<code>&lt;math&gt;y = a X^y + b \sin x&lt;/math&gt;</code>
$\frac{\partial p}{\partial x} = f(x)$	<code>&lt;math&gt;{\&amp;pd;p&lt;over&gt;&amp;pd;x}&amp;sp;=f(x)&lt;/math&gt;</code>
$y = \int_0^{\infty} \frac{\sin x}{1+x} dx$	<code>&lt;math&gt;y={\int_0^{\infty}^{\left&lt;sin x&lt;/math&gt;&lt;/math&gt;</code>



# Les balises media: <audio> et <video>

- Concrètement, <audio> et <video> en HTML se comportent globalement de la même façon et partagent beaucoup d'attributs. Leur seule différence réside dans leur apparence, induite par la nature du contenu : visible pour une vidéo et invisible pour de l'audio seul.
- Leur usage est extrêmement simple, contrairement au choix du format, et ressemble fortement à celui des images. Dans le meilleur des cas, il suffit de préciser un attribut `src`.
- Exemple

## Exemples de codes HTML variés

```
<!-- Une image -->

<!-- Une vidéo -->
<video src="mes_vacances.webm"></video>

<!-- Un son -->
<audio src="anton_aus_tiro1.mp3"></audio>
```

# Interface de contrôle selon le navigateur

**Figure**

Audio sous Google Chrome



**Figure**

Audio sous Internet Explorer 9



**Figure**

Audio sous Safari



**Figure**

Audio sous Opera



**Figure**

Audio sous Firefox





# <audio>

- L'intégration d'un fichier audio avec la balise `<audio>` dans le document HTML est extrêmement simple, il suffit de préciser l'attribut `src` pour faire référence au fichier concerné.

## Exemple minimaliste `<audio>`

```
<audio src="mamusique.oga">
 Télécharger le fichier son
</audio>
```

- *Remarque*: L'élément `<audio>` ne possède pas d'attributs `width` (largeur), `height` (hauteur).



# <video>

- L'intégration d'un fichier vidéo avec `<video>` est aussi simple. L'attribut `src` fait toujours référence au fichier à lire dans la page.

## Exemple minimaliste `<video>`

```
<video src="video.ogv" width="640" height="480">
 Télécharger la vidéo
</video>
```

- *Remarque*: il est fortement conseillé d'attribuer des dimensions, dans l'exemple ci-dessus grâce aux attributs `width` (largeur) et `height` (hauteur).



## <source>

- Ces deux nouveaux éléments média disposent, outre l'attribut `src`, d'un moyen alternatif pour indiquer des sources multiples grâce à `<source>`, éléments enfants de `<video>` et `<audio>`.
- Le navigateur décide alors d'utiliser la source qui lui semble la plus appropriée en fonction de ces critères.
- L'attribut `src` doit être omis sur la balise parent, car il obtient la priorité et oblige le navigateur à ignorer les sources alternatives indiquées.



## <source>

- Dans l'exemple ci-dessous, quatre sources sont proposées :
  - une vidéo à destination des mobiles (3GP avec media **handheld**) ;
  - une vidéo Ogg Theora à destination des navigateurs supportant le type MIME **video/ogg** et équipés des codecs theora (vidéo) et vorbis (audio) ;
  - une vidéo WebM à destination des navigateurs supportant le type MIME **video/webm** et équipés des codecs vp8 (vidéo) et vorbis (audio) ;
  - une vidéo MP4 pour les navigateurs connaissant ce format.

```
<video>
 <source src="video.mp4" type="video/mp4">
 <source src="video.3gp" type="video/3gpp" media="handheld">
 <source src="video.ogv" type="video/ogg; codecs=theora, vorbis">
 <source src="video.webm" type="video/webm; codecs=vp8, vorbis">
 Ce navigateur ne supporte pas l'élément <code>video</code>
</video>
```



# Attributs pour <audio> et <video>

- **autoplay** : Active la lecture dès le chargement de la page.
- **preload** : Précharge le fichier média, dès le chargement de la page.
- **controls** : La présence de l'attribut **controls** affiche les contrôles natifs du navigateur (lancement de la lecture, arrêt, progression, volume, etc.).
- **loop** : Active la lecture en boucle.
- **poster**: Indique l'adresse de l'image d'aperçu à charger pour occuper la zone d'affichage lorsque la vidéo n'est pas jouée.
- **width, height**: Spécifie les dimensions d'affichage de la zone vidéo.
- **mediagroup** : Permet de regrouper différents éléments média (audio, vidéo), par exemple dans une liste de lecture gérée par JavaScript.
- **muted** : Active le mode silencieux (volume nul).
- **src** : Indique l'adresse de la source du média (si **<source>** n'est pas utilisé).