



# Initiation **HTML** et **CSS**

Apprendre **les bases** du langage  
pour créer des sites web

Initiation HTML et CSS - @crédits Stéphanie Walter  
Licence CC-BY-NC-SA

# Description du cours

- Le but de ce cours de formation initiation au HTML et CSS est de vous permettre d'appréhender les bases du HTML et CSS afin de pouvoir ensuite créer de manière autonome des sites web.
- Le cours est une initiation, nous n'aurons pas le temps de voir en détail toutes les balises et positionnement mais des liens seront fournis afin de vous permettre d'approfondir.



A mettre en œuvre sur votre TAF

# Programme initiation au HTML

- **Les bases d'un site web**
- **Le HTML – généralités**
  - HTML et définition
  - W3C
  - Présentation rapide de quelques outils : éditeurs de texte, wysiwyg
- **La syntaxe HTML : balises, éléments et attributs**
  - Le principe de balise et d'élément
  - Attributs et valeurs
  - Quelques exemples d'attributs
- **Doctype, html, body : structure d'un document valide**
  - Le doctype
  - Les balises <html>, <head> et <body>
  - Les commentaires
- **Les éléments de structuration du contenu**
  - La balise *division* <div>
  - La balise paragraphe <p>
  - Espaces insécables Titres <h1> à <h6>
  - La balise <blockquote>
- **Les liens hypertexte**
  - Fonction d'un lien
  - La balise <a>
  - Liens externes
  - Liens internes et hiérarchisation du site
  - Ouvrir un lien dans un nouvel onglet ou laisser l'utilisateur choisir ?
  - Les liens *spéciaux* : mailto, ancrs, etc.

# Programme initiation au HTML

- **Les éléments de « mise en forme » de texte**
  - Les balises `<strong>` et `<em>`
  - La balise `<span>`
  - D'autres balises (`<sup>`, `<sub>`, etc.)
- **Les éléments de liste**
  - Listes ordonnées
  - Listes non ordonnées
- **Ajouter des images avec la balise `<img>`**
  - Guide des formats d'image pour le web et outils de compression
  - La balise `<img>` et ses attributs
- **Les tableaux HTML**
  - Créer un tableau : `<table>` `<tr>` `<td>`
  - Un exemple de tableau simple
  - Ajout d'en-têtes avec `<th>`
  - Légende de tableau
  - Attributs avancés, dont `colspan`, `rowspan`
- **Les formulaires**
  - Les formulaires sont partout autour de nous
  - La balise `<form>`
  - Le champ `<input>`
  - Ajout de libellés avec `<label>`
  - Les *placeholder* HTML5
  - Zone de texte multiligne `<textarea>`
  - Bouton de validation `<input type="submit">`
  - Présentation des autres éléments (checkbox, radio, `<select>`)

# Programme initiation au CSS

- **CSS : introduction et notions de base**
  - Ajoutez du style à vos pages !
  - CSS : Cascading Style Sheets, une définition
  - Les CSS, un peu d'histoire
  - Différents rendus entre les navigateurs : le *pixel perfect* n'existe pas
- **Appliquer un style, oui mais où ?**
  - Le CSS *inline* et inconvénients
  - Le CSS "interne" intégré au document et inconvénients
  - Feuille de style externe et balise <link>
- **Syntaxe CSS**
  - La syntaxe de base : selecteur, propriété, valeur
  - La déclaration CSS
  - Les commentaires CSS
- **Généalogie et différents types de sélecteurs**
  - Le sélecteur usuel d'élément HTML
  - Hiérarchie et généalogie : notion d'enfants, de parent, de descendance
  - Sélecteur de groupe, de classe
  - L'id ou identifiant
  - Espace entre sélecteurs, ou pas ?
  - Les pseudos-classes sur les liens
  - Les autres pseudo-classes
- **Couleurs et unités**
  - Notation des couleurs en CSS
  - Les unités *fixes* en CSS
  - Les unités *fluides* en CSS
  - En web, quelle unité utiliser ?

# Programme initiation au CSS

- **Propriétés de typographie, polices et puces**
  - Font-family et polices utilisables
  - Propriété typographiques (gras, italique, interlignage, etc.)
  - Changer la couleur du texte
  - Alignements de texte
  - Décoration de texte (soulignement, etc.)
  - Changer les icônes d'une liste à puce
- **Bordures et arrières plans**
  - Bordures et styles, bords arrondis CSS3
  - Arrière-plan et propriétés background
- **Dimensions, margin et padding**
  - Différences entre inline et block
  - Largeur et hauteur avec width et height
  - Dimensions maximum et minimum
  - Marge extérieure margin
  - Centrer horizontalement à l'aide de margin
  - Marge intérieure padding
  - Calcul des dimensions réelles d'un élément
  - S'affranchir du calcul avec box-sizing: border-box
  - Gestion des dépassements avec overflow
- **Positionnement CSS : les flottants pour placer des images**
  - Les *flottants* et le flux
  - Ferrer les images à gauche et à droite
  - Nettoyer les flottants avec clear
  - Notion de contexte de formatage de bloc et gestion des conflits de flottants

# Programme initiation au CSS

- **Propriété display**
  - Valeurs block, inline-block, inline et none
  - Créer une navigation à partir des propriétés de display
- **Positionnement CSS relative, absolute et fixed**
  - Position relative pour déplacer des éléments
  - Position fixe et barre de navigation fixée en haut
  - Position absolue et déplacement d'élément dans bloc
- **Aller plus loin**
  - Mises en page avec Flexbox et Grid Layout
  - Avant-goût du Responsive Web Design

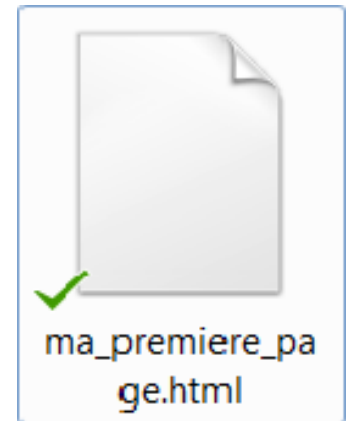
# Les bases d'un site web

---



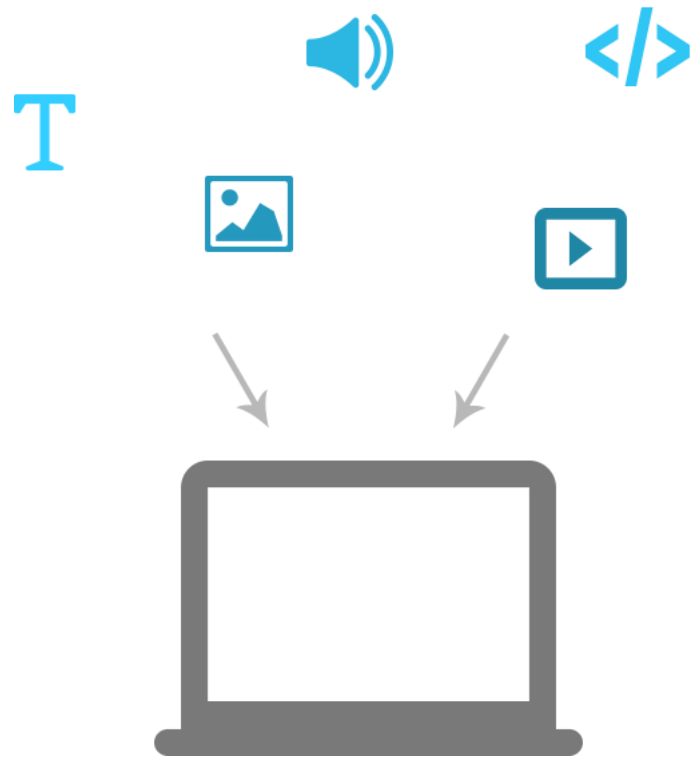
# Une page Web c'est ...

- Un fichier HTML est un format de fichier « texte » éditables dont les éléments ont du sens
- Avec l'extension **.html**
- Peut contenir du texte, des images, des liens, des médias, etc.
- Peut être liée à une autre page via des liens
- => ouvrez <https://creativecommons.org/> sur Firefox et faites CTRL + U au clavier



# Un site Web c'est ...

- Un ensemble de pages liées entre elles
- Accessible en ligne depuis n'importe où



On peut  
aussi dire  
Site Internet  
!

# Mais pas que ...

- Contenus textuels dans le fichier HTML, mais aussi images, sons, vidéos, CSS, JavaScript, polices d'écriture, etc. tous ces fichiers sont chargés dans le navigateur pour créer une page web.



# Le HTML - généralités

---

# HTML : définition

- **HyperText Markup Language (HTML)**
- **Langage de balises** qui permet de structurer des pages
- Différentes versions depuis 1989
- Aujourd'hui :
  - XHTML (2000 – 2006)
  - HTML5

```
1 <!DOCTYPE html>
2
3 <html lang="fr">
4 <head>
5
6   <meta charset="UTF-8" />
7
8   <title>Quelle différence y a-t-il entre HTML et XHTML ? - Alsacreations</title>
9
10  <link rel="shortcut icon" href="http://www.alsacreations.com/favicon.ico" />
11  <link rel="icon" type="image/x-icon" href="http://www.alsacreations.com/favicon.ico" />
12  <link rel="icon" type="image/png" href="http://www.alsacreations.com/favicon.png" />
13
14  <meta name="keywords" content="xhtml, validation, html, tableless, doctype, différence" />
15
16  <meta name="geo.placename" content="Strasbourg, Alsace, France" />
17
18  <link rel="apple-touch-icon" href="http://www.alsacreations.com/apple-touch-icon.png" />
19  <link rel="stylesheet" type="text/css" href="http://www.alsacreations.com/css/styles20120827.min.css" />
20  <link rel="search" type="application/opensearchdescription+xml" title="Alsacreations"
21    href="http://www.alsacreations.com/opensearch.xml" />
22
23  <meta name="viewport" content="width=device-width" />
24
25  <!--[if lte IE 6]><link type="text/css" rel="stylesheet" href="http://www.alsacreations.com/css/ie6fix.css"
26    media="all" /><![endif-->
27  <!--[if lte IE 7]><style type="text/css">#footer, #page { zoom: 1; }</style><![endif-->
28
29  <link rel="stylesheet" title="Zenburn" href="/js/highlight/styles/zenburn.css"><link rel="alternate"
30    type="application/rss+xml" href="/rss/actualites.xml" title="Actualités" />
31  <link rel="alternate" type="application/rss+xml" href="/rss/commentaires.xml" title="Commentaires" />
32  <link rel="alternate" type="application/rss+xml" href="/rss/apprendre.xml" title="Apprendre (dernières
33    publications)" />
34
35  <script type="text/javascript">
36
37    var _gaq = _gaq || [];
38    _gaq.push(['_setAccount', 'UA-1456483-4']);
39    _gaq.push(['_trackPageview']);
40
41    (function() {
42      var ga = document.createElement('script'); ga.type = 'text/javascript'; ga.async = true;
43      ga.src = ('https:' == document.location.protocol ? 'https://ssl' : 'http://www') + 'google-
```

# Le W3C

- W3C : World Wide Web Consortium (1994)
  - Tim Berners Lee, fondateur du W3C et inventeur du HTML
  - Chargé de promouvoir la compatibilité des technologies dans les navigateurs
  - 378 entreprises membres qui peuvent faire des propositions (Microsoft, Apple, Mozilla, Opera, Adobe, etc.)
  - Proposent un validateur <http://validator.w3.org/>  
Le validateur permet de contrôler votre page au sens de la norme HTML5.

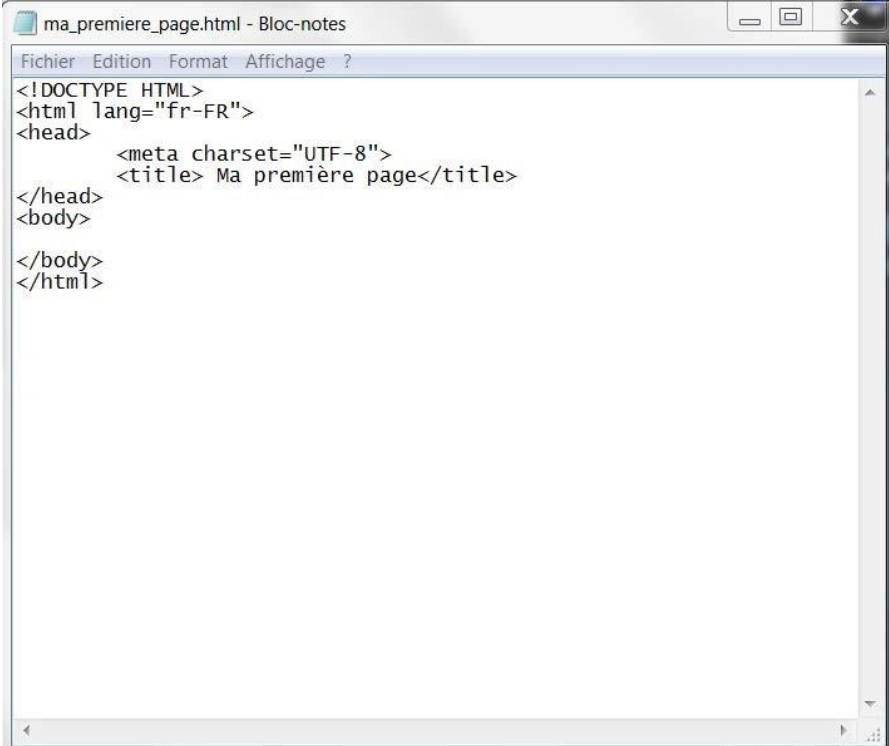


A utiliser impérativement  
avant de rendre votre TAF

# Générer du HTML avec un éditeur

Blocnote ou TextEdit, si vous n'avez rien d'autre sous la main :

- Pas de coloration syntaxique
- Nécessite de connaître tout le langage

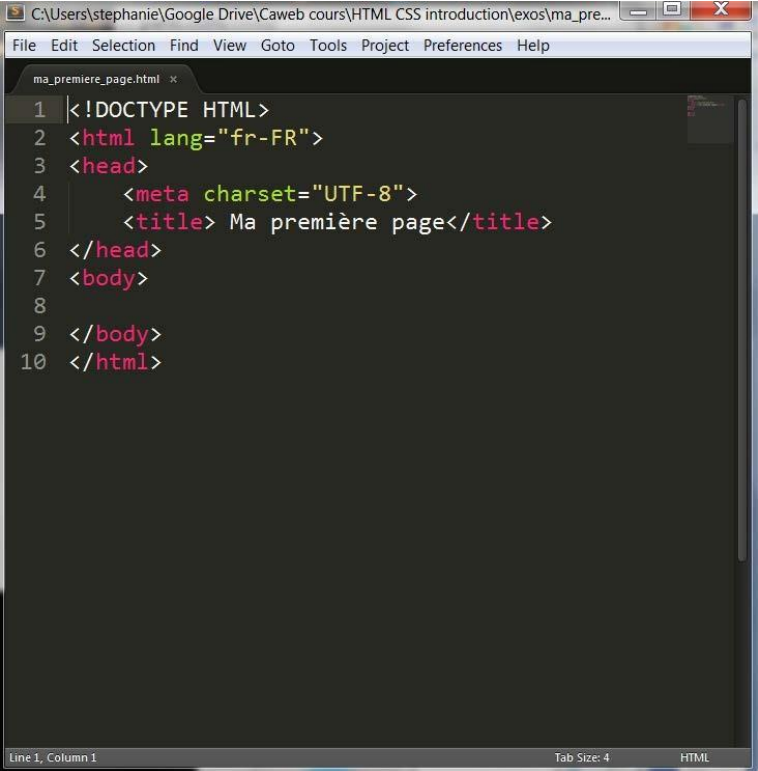


```
<!DOCTYPE HTML>
<html lang="fr-FR">
<head>
  <meta charset="UTF-8">
  <title> Ma première page</title>
</head>
<body>

</body>
</html>
```

# Les éditeurs de texte plus avancés et spécialisés

- <http://www.sublimetext.com/>
- Éditeur spécialisé
- Auto-complétion, retour à la ligne, colorisation syntaxique
- Plugins, snippets, mode « sans distraction »
- 30 jours essai Windows/Mac/ Linux puis payant



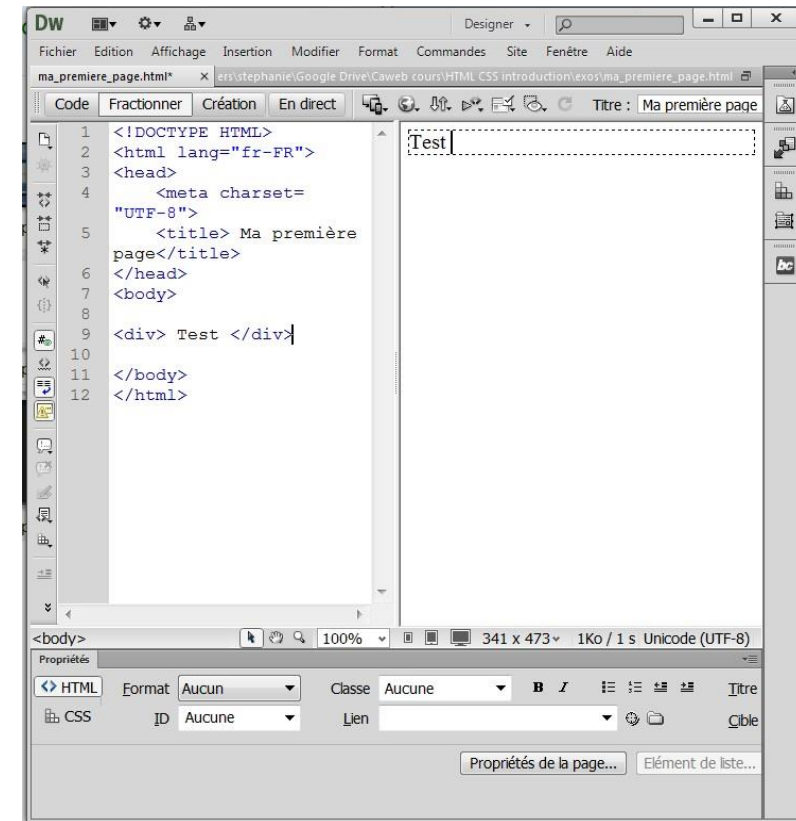
```
1 <!DOCTYPE HTML>
2 <html lang="fr-FR">
3 <head>
4   <meta charset="UTF-8">
5   <title> Ma première page</title>
6 </head>
7 <body>
8
9 </body>
10 </html>
```

Et bien d'autres encore ....



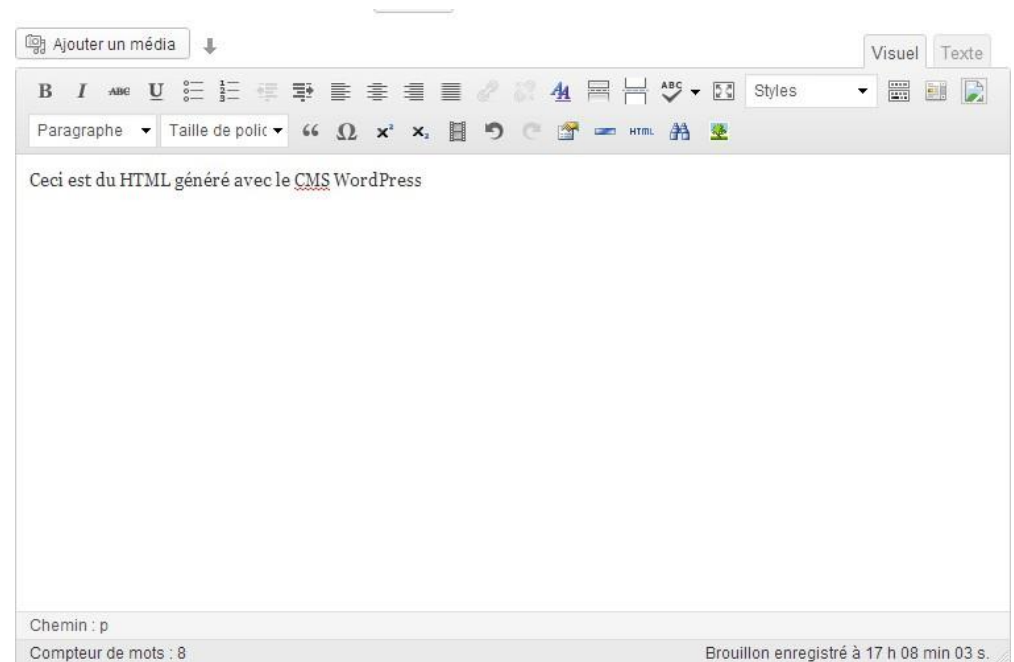
# Les éditeurs « WYSIWYG »

- What You See Is What You Get
- Permet de visualiser le rendu directement
- Code parfois pas toujours propre ou optimisé
- Exemple : Dreamweaver (payant), Blue Griffon (gratuit)



# Générer du HTML depuis un CMS

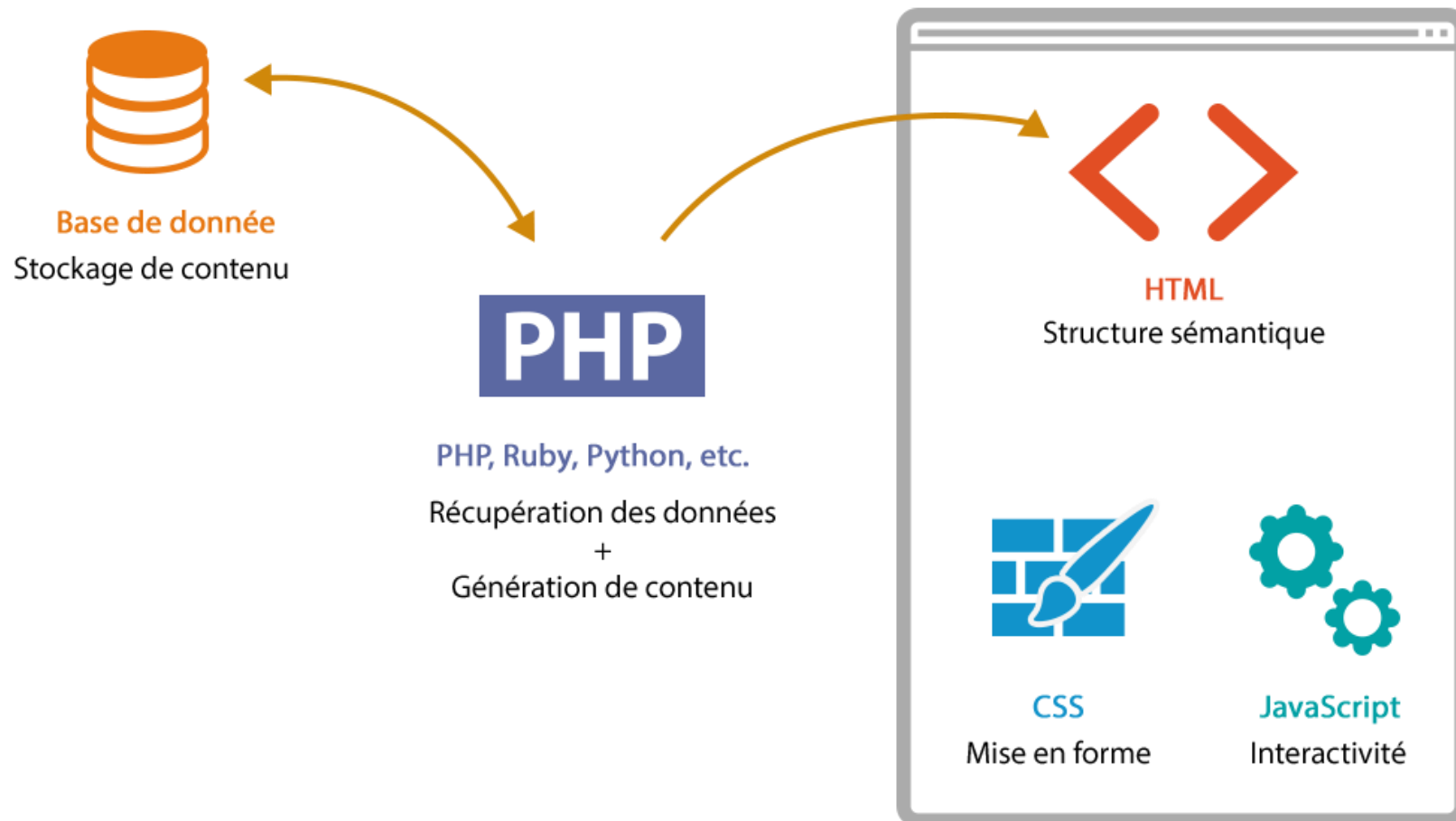
- CMS : Content Management System ou systèmes de gestion de contenu
- Développé dans un langage de programmation web (ex : PHP) et fait appel à une base de données (ex : SQL)
- Éditeurs « visuels » sans avoir besoin de notions de code
- Ex : WordPress, Drupal, Joomla, Prestashop, etc.



# Générer du HTML depuis un langage côté serveur

- Possibilité de générer « automatiquement » du HTML depuis un langage de programmation côté serveur
- PHP, Python, Ruby, etc.

```
559 function twentyeleven_comment( $comment, $args, $depth ) {
560     $GLOBALS['comment'] = $comment;
561     switch ( $comment->comment_type ) :
562         case 'pingback' :
563             case 'trackback' :
564                 ?>
565                 <li class="post pingback">
566                     <p><?php _e( 'Pingback:', 'twentyeleven' ); ?> <?php
comment_author_link(); ?><?php edit_comment_link( __( 'Edit',
'twentyeleven' ), '<span class="edit-link">', '</span>' ); ?></p>
567                     <?php
568                         break;
569                     default :
570                         ?>
571                 <li <?php comment_class(); ?> id="li-comment-<?php comment_ID(); ?
>">
572                     <article id="comment-<?php comment_ID(); ?>" class="comment">
573                         <footer class="comment-meta">
574                             <div class="comment-author vcard">
575                                 <?php
576                                     $avatar_size = 68;
577                                     if ( '0' != $comment->comment_parent )
578                                         $avatar_size = 39;
579
580                                     echo get_avatar( $comment, $avatar_size );
581
582                                     /* translators: 1: comment author, 2: date and time */
583                                     printf( __( '%1$s on %2$s <span class="says">said:
</span>', 'twentyeleven' ),
584                                         sprintf( '<span class="fn">%s</span>',
get_comment_author_link() ),
585                                         sprintf( '<a href="%1$s"><time pubdate
```



# La syntaxe HTML : balises, éléments et attributs

---

# Le principe de balise

- Les balises structurent le contenu de la page (texte, images, etc.)
- Chaque balise a un rôle et donne du sens au contenu présenté





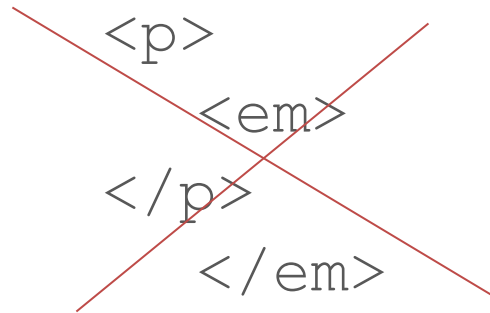
# Le principe de balise

- Touche < du clavier pour ouvrir : <
- Touche maj + < pour fermer : >
- Le nom des balises est prédéfini dans les spécifications HTML, on ne peut donc PAS en inventer !
- Quelques exemples : <html>, <body>, <img>, <p>, <div>, <a>, etc.



# Le principe de balise

- Par convention et pour faciliter la lecture du code, toute balise ouverte doit être fermée (sauf exception).
- Interdiction de les faire se chevaucher :



```
<p>  
  <em>  
    </p>  
    </em>
```

- Certaines balises bien particulières n'ont pas besoin d'être fermées, on les dit « auto-fermantes », elles n'ont ni contenu ni balise fermante. Dans ce cas, je mets le / final par convention, mais pas obligatoire.

# Attributs et valeurs

- Certaines balises peuvent être complétées par des attributs précisant certains paramètres (l'adresse des liens, source d'une image à afficher, etc.)
- Une balise peut contenir plusieurs attributs

`<p id="section1"> Bonjour ! </p>`

↑   ↑   ↑

balise   attribut   valeur

# Attributs et valeurs

- ! Les attributs sont des éléments prédéfinis dans les spécifications HTML. On ne peut pas en « inventer ».
- ! On peut les mettre entre simple quote, double quote ou rien. Par convention nous choisissons les doubles quotes " "

`<div id="kittens"> // <div id=kittens>`

- ! Les attributs sont ***toujours dans la balise ouvrante***

# Exemples d'attributs

- ! **src=" "** : donner la source (d'une image par exemple)
  - ``

Note : pour la balise img, la source et le alt sont OBLIGATOIRE !

- ! **href=" "** : donner la destination d'un lien
  - `<a href="http://www.google.fr"> Texte du lien </a>`

# Exemples d'attributs

- ! **id=" "** : donner un identifiant à l'élément que l'on pourra utiliser en CSS.

Les Ids doivent être uniques par page

- `<p id="monid"> contenu </p>`

- ! **class=" "** : donner une classe à l'élément que l'on pourra utiliser en CSS.

Les classes peuvent être dupliquées sur la page

- `<p class="maclasse"> contenu </p>`

# Doctype, html, body : structure d'un document valide

---

# Structure minimale d'un document HTML

Décomposons balise par balise !

```
1 <!doctype html>
2 <html lang="fr">
3 <head>
4     <meta charset="UTF-8">
5     <title> Titre de ma page </title>
6 </head>
7 <body>
8
9     Contenu de la page
10
11 </body>
12 </html>
```

# La notion de doctype

- Permet au navigateur de savoir quelle version de HTML (ou XHTML) est utilisée sur la page
- Obligatoire pour valider une page (validateur W3C)
- La doctype HTML5 est simple et concise ce qui n'est pas le cas des normes précédentes.

`<!doctype html>`



# La balise <html>

- Tout document HTML commence par la balise <html> qui se ferme en fin de document : </html>
- La balise <html> contient nécessairement deux balises définissant l'entête (<head>) et le corps (<body>) du document.

# L'attribut lang

- Permet de spécifier la langue de traitement du document global
  - Indexation dans la bonne langue par les moteurs de recherche
  - Synthèse vocale (Navigateur dédié aux handicaps)
  - Vérification orthographique des formulaires

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

# La balise <head>

- Détermine l'entête du document et contient les méta-informations

Les métadonnées ne servent pas seulement aux moteurs de recherche. Si Google capture la page entière, un navigateur ouvre l'en-tête (head) avant le contenu (body). Par exemple, l'ensemble des données de l'en-tête sont accessible avant le chargement du contenu.

- L'ordre des méta-informations importe peu.
- Il peut y en avoir beaucoup (exemple wikipédia), mais il faut savoir rester lisible.

# La balise <title>

- On y trouve la balise <title> qui sera le titre du document
  - Sert en référencement (repris dans les résultats de recherche)

Stéphanie Walter - Web & UX Designer spécialisée en design d ...

<https://www.stephaniewalter.fr/> ▼

**Stéphanie Walter** : Web et UX design spécialisée dans le design de sites webs, d'application mobiles et d'interfaces webs.

[Blog](#) · [Réalisations](#) · [A propos](#) · [Content is like Water](#)

- Affichée dans l'onglet du navigateur



# La balise <head> : les balises meta

- On y ajoutera également les appels pour les feuilles de style CSS et certains JavaScripts
- On peut y trouver des balises « meta » parmi lesquelles
  - `<meta charset="UTF-8"/>` : permet de définir l'encodage de caractère de la page (à mettre directement sous le <head>)



[A quoi servent les balises META](#)

# La balise <head> : les balises meta

- `<meta name="description" content="description pour le référencement"/>` affiché dans les résultats de recherche

**Alsacréations.** Agence web, création de site et formations web

[www.alsacreations.fr/](http://www.alsacreations.fr/) ▼

Agence web spécialisée dans la conception de sites conformes aux standards W3C et formations HTML 5, CSS 3, Accessibilité, jQuery, Wordpress et PHP.

...

# La balise <body>

- ! Délimite le contenu de la page
- ! Le contenu est constitué de texte, images et autres éléments qui seront affichés

<body>

*Contenu de la page*

</body>

# En résumé

```
<!doctype html>  
<html lang="fr">  
<head>
```

```
  <meta charset="UTF-8">  
  <title> Titre de ma page </title>
```

```
</head>
```

```
<body>
```

Contenu de la page

```
</body>  
</html>
```



# C'est valide !

- ! La validation reste un outil et non une fin en soi, mais un code non valide a des chances de poser des soucis plus tard
- Et de ne pas être lisible pour les navigateurs spécialisés

## Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

### Showing results for contents of text-input area

Checker Input

Show ☒ source ☐ outline ☐ image report [Options...](#)

Check by text input

```
</p>
</div>
<div id="kittens">
  
</div>

<div>
  <p>
    Dragée jelly jelly-o chocolate cake icing gingerbread soufflé fruitcake. Dragée biscuit cookie wafer faworki pie fruitcake candy. Halvah
    chocolate tootsie roll jelly-o soufflé pastry jelly beans.
  </p>
</div>
<p id="footer">
  Jelly beans gummies lemon drops brownie halvah cookie. Icing topping caramels.
</p>
```

[Check](#)

Document checking completed. No errors or warnings to show.

# Les commentaires dans le HTML

```
<p> Du texte ici <!-- avec un commentaire invisible --></p>
```

Du texte ici

Il est possible d'écrire du texte qui ne sera visible que dans le code source et non sur le site : un commentaire

# Les éléments de structuration du contenu

---

# La balise « division » <div>

```
<div>Un bloc ici</div>  
<div>un autre bloc ici !</div>
```

```
<div>  
  Encore du texte !  
  Toffee cake faworki topping dessert lemon drops lollipop dessert. Sweet  
  croissant halvah croissant macaroon tiramisu cookie. Cotton candy  
  chocolate cake ice cream faworki icing liquorice pastry marzipan chocolate  
  . Cookie wafer tiramisu tart icing cheesecake jelly beans pastry chocolate  
  . Pastry muffin dessert tart cupcake. Faworki brownie cookie dessert  
  gingerbread bonbon gummies ice cream dragée. Wafer tiramisu wypas  
  croissant danish chocolate bar dragée dessert pudding. Marshmallow sweet  
  lollipop powder. Pie brownie cotton candy sesame snaps.  
</div>
```

Un bloc ici

un autre bloc ici !

Encore du texte ! Toffee cake faworki topping dessert lemon drops lollipop dessert. Sweet croissant halvah croissant macaroon tiramisu cookie. Cotton candy chocolate cake ice cream faworki icing liquorice pastry marzipan chocolate. Cookie wafer tiramisu tart icing cheesecake jelly beans pastry chocolate. Pastry muffin dessert tart cupcake. Faworki brownie cookie dessert gingerbread bonbon gummies ice cream dragée. Wafer tiramisu wypas croissant danish chocolate bar dragée dessert pudding. Marshmallow sweet lollipop powder. Pie brownie cotton candy sesame snaps.

# La balise « division » <div>

- <div> est un **conteneur**
- Cette balise n'a aucune valeur sémantique (aucun « sens »)
- Élément de type **bloc** qui peut inclure n'importe quel autre balise HTML
- Suivi d'un passage à la ligne
- Permet de regrouper d'autres blocs de HTML (paragrapes, tableaux, etc.) et de leur donner un style CSS commun

# Grille en div

div header

div content

div sidebar

div footer

# La balise paragraphe <p>

```
<p> Voici un joli paragraphe</p>
<p> Un autre Pastry chocolate bar fruitcake pudding. Topping
tootsie roll chocolate cake powder jujubes caramels wafer gummies
topping. Tootsie roll candy jujubes lollipop unerdwear.com
gingerbread. </p>
<p> Un paragraphe
    sur plusieurs
    lignes dans le code source
</p>
```

Voici un joli paragraphe

Un autre Pastry chocolate bar fruitcake pudding. Topping tootsie roll chocolate cake powder jujubes caramels wafer gummies topping. Tootsie roll candy jujubes lollipop unerdwear.com gingerbread.

Un paragraphe sur plusieurs lignes dans le code source

# La balise paragraphe <p>

- Les paragraphes sont des éléments de type « bloc », suivis d'un passage à la ligne et d'une marge par défaut
- On peut imbriquer un paragraphe dans un div
- On ne doit PAS mettre un paragraphe dans un paragraphe



# Espaces insécables

<p>

Un extrait de text un peu plus long ici : pour éviter que les deux points passent à la ligne.

</p>

Un extrait de text un peu plus long  
ici : pour éviter que les deux points  
passent à la ligne.

# Espaces insécables

- Il existe des entités HTML pour encoder certains caractères spéciaux comme les espaces insécables : &nbsp;



[Caractères spéciaux et entités HTML](#)

# D'autres entités HTML

Coö(y) p@STē ©ñΔRΔC(t)eℜ

Click to copy — press down ⌘alt for multiple

iPhone Twitter About Sign in to create sets ▼

Our favorite set ▼

Share this set ▼



[copypastetecharacter.com](http://copypastetecharacter.com)

# Les titres h1 – h6

```
<h1> Titre de niveau 1 H1 </h1>  
<h2> Titre de niveau 2 H2 </h2>  
<h3> Titre de niveau 3 H3 </h3>  
<h4> Titre de niveau 4 H4 </h4>  
<h5> Titre de niveau 5 H5 </h5>  
<h6> Titre de niveau 6 H6 </h6>
```

..

**Titre de niveau 1 H1**

**Titre de niveau 2 H2**

**Titre de niveau 3 H3**

**Titre de niveau 4 H4**

**Titre de niveau 5 H5**

**Titre de niveau 6 H6**

# Les titres h1 – h6

- Il existe six niveaux de titre `<h1>` `<h2>` `<h3>` `<h4>``<h5>` et `<h6>` (pas un de plus ! )
- Un nouveau niveau provoque un passage à la ligne, les éléments de titre étant des éléments de bloc.
- La hiérarchisation visuelle est automatique (peut être modifiée via CSS)
- Il est possible d'avoir plusieurs titres de même niveau
- On ne peut pas mettre un Hn dans un p, ou un p dans un Hn

# Blockquote, une citation

```
<p>Une citation ci dessous :</p>
<blockquote>
    Mon médecin m'a recommandé d'arrêter les petits
    dîners pour quatre. A moins qu'il n'y ait trois
    autres personnes
</blockquote>
<p>Orson Welles</p>
```

Une citation ci dessous :

Mon médecin m'a recommandé d'arrêter les petits dîners pour quatre. A moins qu'il n'y ait trois autres personnes

Orson Welles

# Blockquote, une citation

- La balise `<blockquote>` marque une citation
- Une citation est suivie d'un retour à la ligne

“ Pellentesque habitant morbi tristique senectus et netus et malesuada  
fames ac turpis egestas. Vestibulum tortor quam, feugiat vitae, ultricies eget,  
tempor sit amet, ante. Donec eu libero sit amet quam egestas semper.  
Aenean ultricies mi vitae est. Mauris placerat eleifend leo.

<https://css-tricks.com/snippets/css/simple-and-nice-blockquote-styling/>

# Les éléments de « mise en forme » de texte

---



# Éléments en ligne « inline »

- Ces éléments sont des éléments dit « **inline** »
- On les place autour du contenu (et du texte)
- Ne crée pas de retour à la ligne
- Servent à donner du **sens** aux parties du texte/contenu

# L' élément <strong>

```
<p> Du texte normal et <strong>du texte fortement mis en valeur |</strong> </p>
```

Du texte normal et **du texte fortement mis en valeur**

- **<strong>** élément sémantique qui marque du texte comme fortement mis en valeur
- « en gras » par défaut dans les navigateurs

## L' élément <em>

```
<p> Du texte normal et <em>du texte mis en emphatique</em> </p>
```

Du texte normal et *du texte mis en emphatique*

- **<em>** balise sémantique qui marque du texte comme en emphase (mis en évidence) sert à « insister » sur une partie du texte
- en « italique » par défaut

# L' élément <span>

```
<h2> Des livres <span class="colored"> à la une </span>
```

Les livres à la une

- Un élément « générique » structurant de type « inline »
- Permet de styler facilement des morceaux de texte en CSS
- Par défaut affiche son contenu sans le modifier
- Aucune valeur sémantique → (l'équivalent d'un div, en ligne)

# D'autres éléments

- L'élément **<sup>** permet d'afficher un texte exposant

`E=mc<sup>2</sup>`     $E=mc^2$

L'élément **<sub>** permet d'afficher un texte en indice

`H<sub>2</sub>O`     $H_2O$

- L'élément **<small>** représente des annotations et des petits caractères comme les informations légales ou liées au copyright

`des petits caractères comme <small>les informations légales ou liées au copyright</small>`

des petits caractères comme les informations légales ou liées au copyright

# Les liens dans les documents HTML

---

# Fonction d'un lien

- Le lien est un élément cliquable qui renvoie l'utilisateur sur une nouvelle page (ou lui permet de télécharger un fichier)
- Il est souvent stylé différemment (bleu souligné par défaut)
- Il peut renvoyer :
  - vers un autre site (lien externe)
  - vers une autre page du site (lien interne)
  - vers des fichiers (image, .PDF, etc.)

: [un lien externe vers google !](#)

# La balise <a>

- On appelle « ancre de lien » le texte à l'intérieur de la balise
- On ajoute l'attribut href="cible\_du\_lien" pour indiquer vers quelle page on veut envoyer l'utilisateur
- On peut ajouter l'attribut title="titre du lien" pour ajouter une infobulle au survol et indiquer à l'utilisateur où il va arriver

**<a href="index.html" title="titre">** Ancre du lien**</a>**



# La balise <a> pour les liens

```
<p> Bonjour, je suis <a href="http://www.google.fr" title="Moteur de  
recherche">un lien externe vers google !</a></p>
```

Bonjour, je suis [un lien externe vers google !](http://www.google.fr)

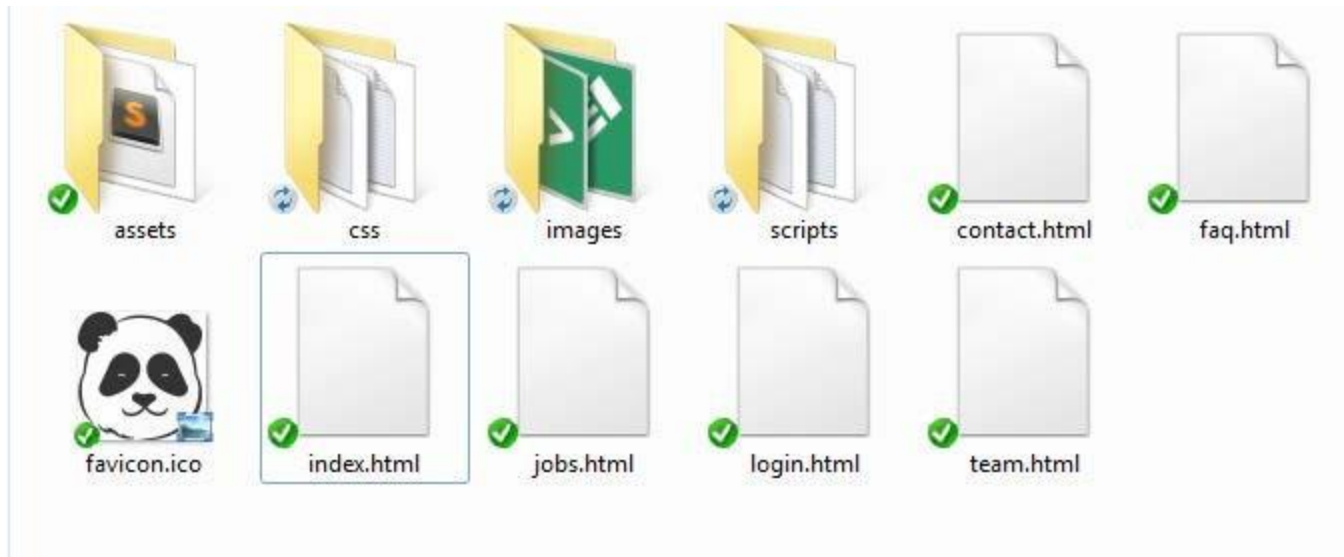
# Lien externe vers un autre site

- La cible sera l'url (ou adresse) complète du site, http:// compris
- Il s'agit d'un lien absolu car on indique l'adresse complète

```
<a href="http://www.google.fr" title="Moteur de recherche  
google"> lien externe vers google </a>
```

# Liens internes et hiérarchisation du site

- Un site est constitué d'un ensemble de pages au sein d'un dossier (appelé racine du site).
- Ces pages peuvent être organisées dans une hiérarchie de sous-dossiers.



# Liens internes et hiérarchisation du site

- Par convention, le fichier « racine » sera appelé index.html.
- C'est lui qui est affiché par défaut s'il existe quand l'utilisateur demande le dossier.

## Index of /sites

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 <a href="#">Parent Directory</a>		-	
 <a href="#">Responsive-Navigatio..&gt;</a>	2013-07-21 12:54	-	
 <a href="#">caweb_wp_demo/</a>	2012-06-04 21:50	-	
 <a href="#">geeksandthecity/</a>	2012-09-18 20:41	-	
 <a href="#">inpixelitrust/</a>	2012-01-08 18:01	-	
 <a href="#">integrationWEB/</a>	2013-07-23 21:04	-	
 <a href="#">opera-fonticon/</a>	2013-04-20 12:25	-	
 <a href="#">restaurant_picker-demo/</a>	2013-06-25 19:31	-	
 <a href="#">wdstr/</a>	2012-01-08 18:01	-	
 <a href="#">web5-conf/</a>	2013-06-12 23:27	-	
 <a href="#">wordpress-test-inpix..&gt;</a>	2012-01-08 18:01	-	

# Lien vers un fichier du même répertoire

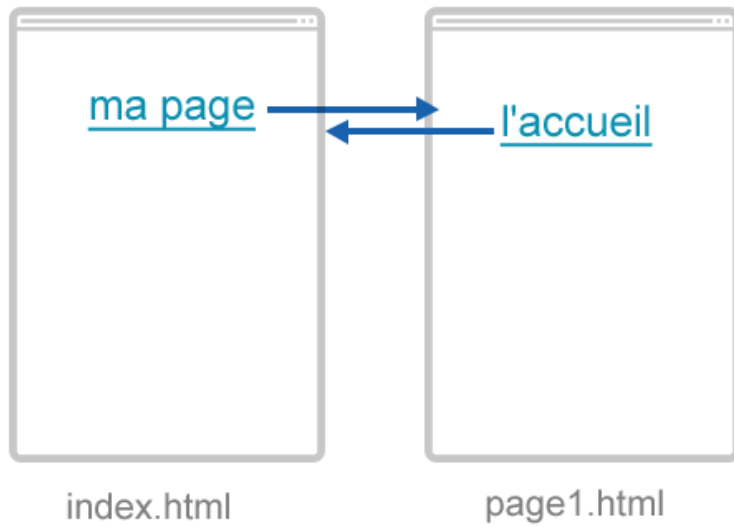
- Lorsque deux fichiers sont dans le même répertoire, il suffit d'appeler le nom du fichier comme cible.
- Ce sont des **liens relatifs**, car relatif au site et dossier dans lequel il se trouve

```
<a href="page1.html" title="ma 1ere page"> ma page</a>
```

```
<a href="index.html" title="accueil du site">l'accueil</a>
```

# Mon premier site

mon\_premier\_site



# Liens vers un autre répertoire

- On peut avoir des répertoires et sous-répertoires
- On ajoute le nom du répertoire pour descendre : monrepertoire/  
mapage.html
- On ajoute **../** pour remonter si on se trouve dans un sous répertoire :  
**../index.html**

# Descendre dans un sous répertoire

- ! Depuis la racine index vers page2 :

```
<a href="page_interne/page2.html">page 2</a>
```

- ! Si on avait d'autres sous-répertoires :

```
<a href="page_interne/autre_sous_rep/page2.html">page 2</a>
```

---

```
/exercice03-mon_premier_site/page_interne/page2.html
```

---

---

```
\exercice03-mon_premier_site\page_interne
```

---



# Remonter d'un répertoire

- ! Depuis la page2 vers la racine :

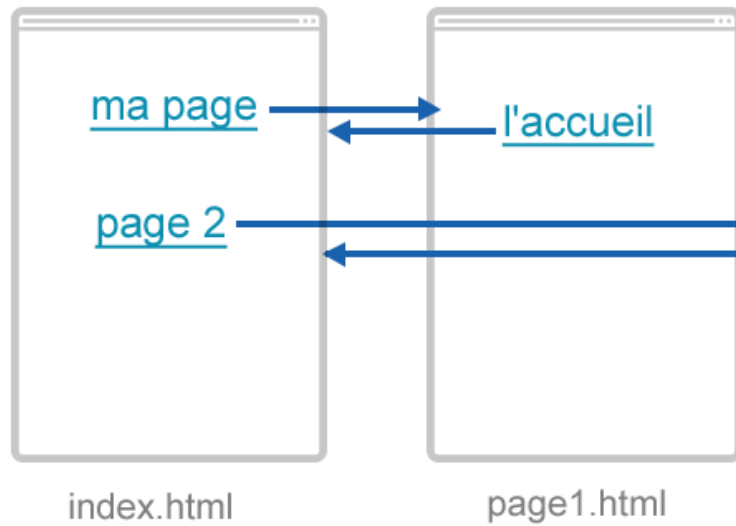
```
<a href=" ../index.html"> l'accueil</a>
```

- ! Si on veut remonter plusieurs répertoires (autant de ../ que de répertoires:

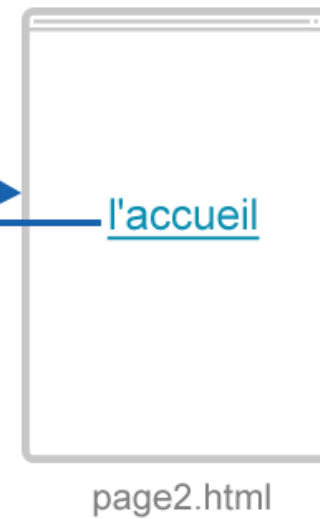
```
<a href=" ../../index.html"> l'accueil</a>
```

# Mon premier site

mon\_premier\_site



page\_interne



# Ouvrir un lien dans un nouvel onglet

- ! L'attribut `target="_blank"` permet d'ouvrir un lien dans une nouvelle fenêtre/onglet en fonction du navigateur

```
<a href="index.html" target="_blank" >
```

!

# Des liens « internes » ou ancres internes

- ! Comme dans Word, permet de renvoyer l'utilisateur à l'intérieur d'une page
- ! La cible est définie par un `id="ancre_du_lien"`
- ! On utilise `<a href="#ancre_du_lien">` pour créer le lien
- ! Utilisé pour les liens « retour en haut de page », wikipedia, etc.
- ! L'ancre apparaît dans l'url :

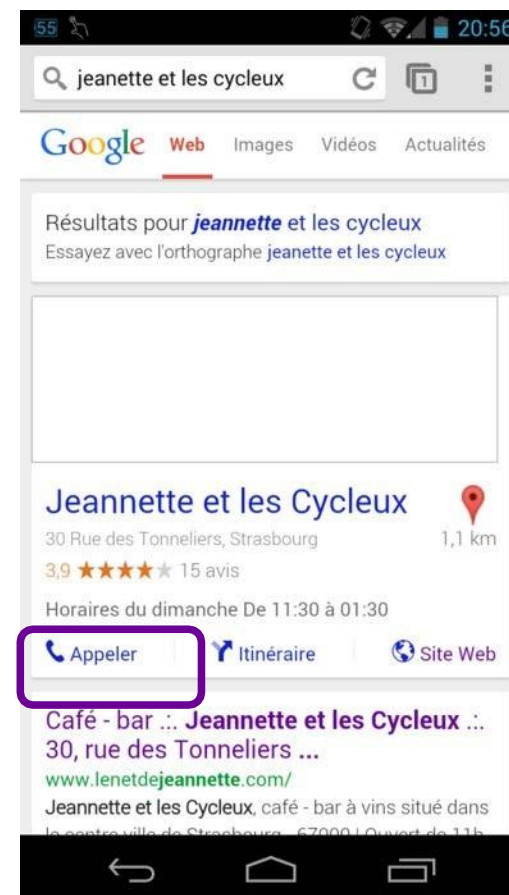
<https://fr.wikipedia.org/wiki/>

[Hypertext Markup Language#Description de HTML](https://fr.wikipedia.org/wiki/Hypertext_Markup_Language#Description_de_HTML)

# Un lien vers une adresse mail

! `<a href="mailto:monmail@gmail.com">` permet de créer un lien qui ouvrira automatiquement le client mail de la personne (à éviter car mail en dur donc récupérable pour spam => formulaire de contact )

! `<a href="tel:13371337">` : permet de définir un numéro de téléphone à appeler



# Les éléments de liste

---

## Les listes non ordonnées <ul>

```
<ul>  
  <li> élément de liste 1 </li>  
  <li> élément de liste 2 </li>  
  <li> élément de liste 3 </li>  
</ul>
```

- élément de liste 1
- élément de liste 2
- élément de liste 3

- La balise **<ul>** permet de créer une liste non ordonnée (unordered list). Il faut l'ouvrir en début de liste et la fermer après le dernier élément de la liste
- La balise **<li>** permet de créer UN élément de la liste

# Les listes ordonnées <ol>

```
<ol>  
  <li> élément numéroté de liste 1 </li>  
  <li> élément numéroté de liste 2 </li>  
  <li> élément numéroté de liste 3 </li>  
</ol>
```

```
1. élément numéroté de liste 1  
2. élément numéroté de liste 2  
3. élément numéroté de liste 3
```

- La balise **<ol>** permet de créer une liste ordonnée et numérotée (par défaut dans le navigateur). Il faut l'ouvrir en début de liste et la fermer après le dernier élément de la liste.
- La balise **<li>** permet de créer UN élément de la liste



# Contenu d'une liste

- Les éléments `<ul>` et `<ol>` ne peuvent contenir que des éléments `<li>`
- Chaque élément `<li>` peut contenir des éléments de bloc ou des éléments incorporés (p, div, blockquote, h1, etc.).
- Les listes peuvent s'imbriquer
- On peut mélanger `<ul>` et `<ol>` à condition de les fermer correctement

# Exemple d'imbrication

```
<ul>
  <li>élément 1 de liste</li>
  <li>élément 2 de liste
    <ul>
      <li>élément 1 de sous liste</li>
      <li>élément 2 de sous liste</li>
      <li>élément 3 de sous liste</li>
    </ul>
  </li>
  <li>élément 3 de liste</li>
</ul>
```

- élément 1 de liste
- élément 2 de liste
  - élément 1 de sous liste
  - élément 2 de sous liste
  - élément 3 de sous liste
- élément 3 de liste

- élément 1 de liste
- **Mes films préférés**
  - 1. Pulp Fiction
  - 2. Fight Club
  - 3. Reservoir Dogs
- Un autre object de liste
- Ma musique préférée
  - The prodigy
  - Asian Dub Foundation
  - Nero
- D'autres trucs



```
<ul>
  <li>élément 1 de liste</li>
  <li> <h1>Mes films préférés</h1>
    <ol>
      <li>Pulp Fiction</li>
      <li>Fight Club</li>
      <li>Reservoir Dogs</li>
    </ol>
  </li>
  <li>Un autre object de liste</li>
  <li>Ma musique préférée
    <ul>
      <li>The prodigy</li>
      <li>Asian Dub Foundation</li>
      <li>Nero</li>
    </ul>
  </li>
  <li>D'autres trucs</li>
</ul>
```

# Les listes en pratique – les onglets

```
<ul>
  <li><a href="#home">Home</a></li>
  <li><a href="#profile">Profile</a></li>

  <li>
    <a href="#">Dropdown</a>
    <ul>
      <li><a href="#dropdown1">@fat</a></li>
      <li><a href="#dropdown2">@mdo</a></li>
    </ul>
  </li>
</ul>
```



Raw denim you probably haven't heard of them jean shorts Austin. Nesciunt tofu stumptown aliqua, retro synth master cleanse. Mustache cliche tempor, williamsburg carles vegan helvetica. Reprehenderit butcher retro keffiyeh dreamcatcher synth. Cosby sweater eu banh mi, qui irure terry richardson ex squid. Aliquip placeat salvia cillum iphone. Seitan aliquip quis cardigan american apparel, butcher voluptate nisi qui.

<http://getbootstrap.com/javascript/#tabs>

# Les listes en pratique – les navigations



```

<ul class="primary-nav__list">
  ::before
  <li class="primary-nav__item jsDropDownMenu">
    <a href="/platform/better-mobile-shopping/" class="primary-nav__link">
      Platform
    </a>
    <span class="primary-nav__mobile-icon jsDropDownMenuButton" aria-hidden=
      "true">...</span>
    <nav class="primary-nav__sub jsDropDownMenuItems">
      <ul class="primary-nav__sub-list">
        <li class="primary-nav__sub-item">...</li>
        <li class="primary-nav__sub-item">...</li>
        <li class="primary-nav__sub-item">...</li>
        <li class="primary-nav__sub-item">...</li>
        <li class="primary-nav__sub-item">...</li>
        <!-- <hr> -->
        <li class="primary-nav__sub-item">...</li>
        <li class="primary-nav__sub-item">...</li>
      </ul>
    </nav>
  </li>
  <li class="primary-nav__item jsDropDownMenu">...</li>
  <li class="primary-nav__item jsDropDownMenu">...</li>
  <li class="primary-nav__item jsDropDownMenu">...</li>
  ::after
</ul>

```

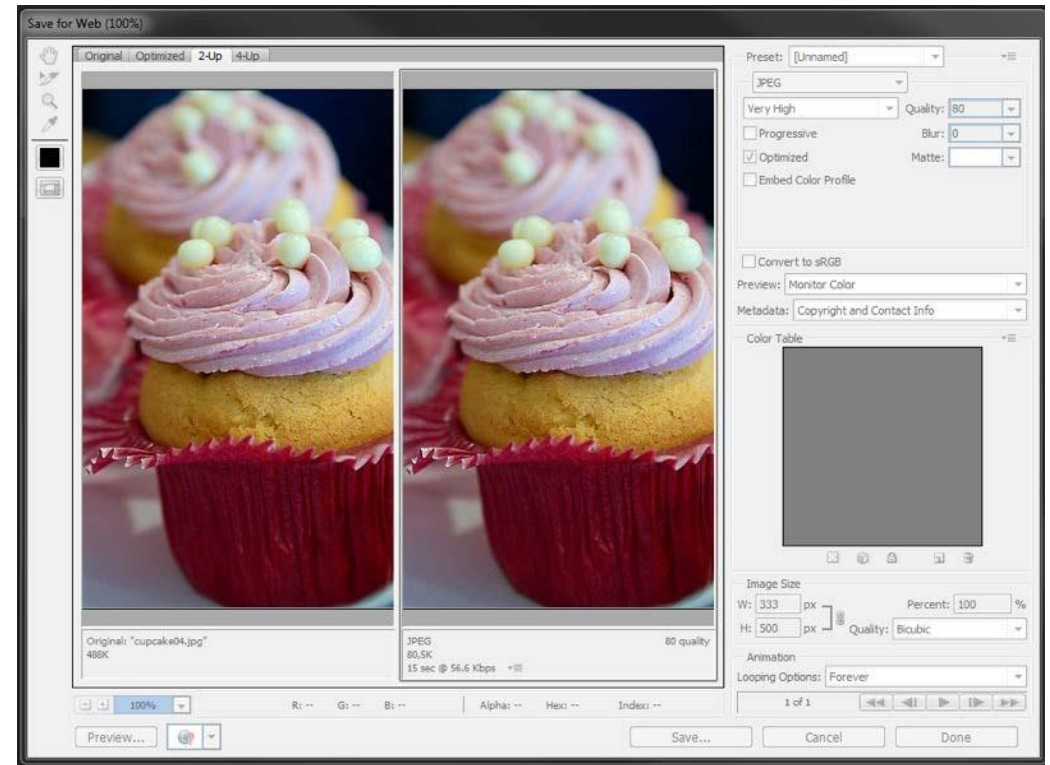
<http://www.mobify.com/>

# Ajouter des images avec la balise `<img>`

---

# Le format JPEG- Joint Photographic Experts Group

- Le JPEG (.jpg, .jpeg) :  
format compressé qui réduit  
le poids des images pour le  
web (donc temps de  
chargement)



=> Pour la compression avec pertes très efficace pour des images photographiques

# Le format GIF - Graphics Interchange Format

- Utilise 256 couleurs
- Compression sans perte basée sur la succession de pixels de même couleur
- Couleur de transparence sans couche alpha
- animation
- extension .gif
- => Utilisé pour les logos



<http://lesjoiesducode.tumblr.com/>

# Le format PNG - Portable Network Graphics



- PNG8 : 256 couleurs, utilisé pour les images nécessitant peu de couleurs et motifs répétitifs
- PNG24 : 16 millions de couleurs, transparence extension .png
- => Utilisé pour les icônes, logos et fichiers qui ont besoin d'un fond transparent



# En résumé

- Une photo : un JPEG
- Une illustration avec peu de couleurs sans transparence importante : PNG8
- Une illustration avec beaucoup de couleurs : PNG24
- Une image animée : GIF

# Compresser une image JPEG outils

- En ligne :
  - <https://imagify.io/> outil en ligne avec traitement par lots
  - <https://tinypng.com/> jusqu'à 20 images en même temps
- En Local :
  - <https://imageoptim.com/fr.html> sur mac PNG et JPG
  - <http://luci.criosweb.ro/riot/> sur windows, PNG et JPG
  - <http://pnggauntlet.com/> sur windows, PNG et JPG
- Photoshop, Fireworks, etc.

# La balise <img>

- Balise de type **inline** : pas de retour à la ligne, peut-être insérée dans le corps du texte. On la place généralement dans un <p> ou un <div> mais peut se placer n'importe où dans le body
- Balise auto-fermante

**<img** src="ref\_image" alt="description de l'image" **/>**

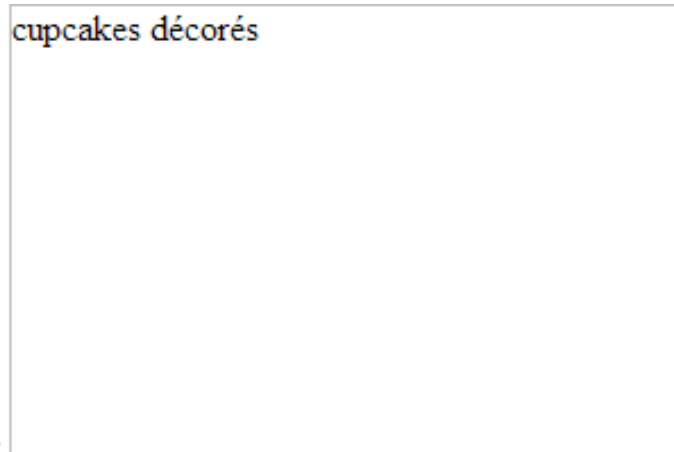
# La balise <img> et source de l'image

- Un attribut **src="image.jpg"** pour donner le chemin vers la source de l'image
  - chemin sur le serveur
  - url de stockage
  - même principe que pour les liens

# La balise <img> et attribut « alt »

- Un attribut **alt="description du contenu de l'image"** texte alternatif qui décrit le contenu de l'image et la remplace si l'image n'est pas téléchargée

Une image en jpeg de cupcake :



# La balise `<img>` et attribut « alt »

- L'attribut est obligatoire, mais il peut être vide pour les images décoratives
  - Lu vocalement par les lecteurs d'écran
  - Aide les personnes en situation de déficience visuelle à comprendre

# Attributs width et height

- Les attributs width et height peuvent être ajoutés pour donner une taille à l'image
- Sans width/height le navigateur « devine » la taille ce qui crée des sauts de page quand l'image se charge
- En général, il vaut mieux la redimensionner avant dans un logiciel, plutôt que charger une grande image et la rétrécir (performance)
- Attention à garder les proportions !

```

```

# Exemple

```
<p> Jelly beans cookie marshmallow cheesecake oat cake chupa chups pie sweet wypas.  
    
  Wafer pie macaroon lemon drops danish sugar plum pastry ice cream. Bear claw wafer  
  tiramisu chocolate. Marzipan croissant cake fruitcake.  
</p>
```



ake chupa chups pie sweet wypas.  
issant cake fruitcake.

Wafer pie macaroon lemon drops



# L'attribut title

```

```



Une image en jpeg de cupcake :

Attribut facultatif qui permet d'afficher une info-bulle au survol de l'image

# Bonus : intégrer une vidéo

- La plupart des sites de vidéo proposent une intégration via iframe qui se présente sous cette forme :

```
<iframe width= ...> ... </  
iframe>
```

- Il faut donc retrouver l'endroit où le site propose l'iframe





## Bonus : intégrer du son

- Beaucoup de sites de partage audio proposent également une intégration via iframe.
- Même principe que pour les vidéos : on récupère le code de l'iframe et on le colle sur notre site

Share


Embed





Code & preview

<iframe width="100%" height="166" scrolling="no" frameborder="0" src="https://w.soundcloud.com/player/?url=https%3A://api.soundcloud.com/tracks/229228277&visualizer=true" data-bbox="125 525 745 565">



☐ Wordpress code






Zeds Dead

Zeds Dead - Lost You...



3:38

5,631,226

Cookie policy

More options ▼

# D'autres types d'iframe



- Généralement les sites qui vous proposent d'intégrer leur contenu vont vous proposer une iframe. C'est le cas par exemple de :
  - [Google maps](#) pour des cartes interactives
  - [Slideshare](#) pour partager des diaporamas et présentations
  - [Vimeo](#), un autre service de partage de vidéos
  - Les boîtes de partages de réseaux sociaux comme [Facebook](#)

# Les tableaux HTML

---

# Les tableaux, introduction

- Un tableau sert à organiser des informations structurées sous forme tabulaire
- Il se compose de lignes organisées elles-mêmes en cellules

	COULEUR	TAILLE	PRIX		QTÉ		TOTAL	
	<u>De Ville - Rose framboise</u>	taille : 42/46	6,00 €	x	<div><div>+</div><div>-</div></div> <div>1</div>	=	6,00 €	<div>SUPPR.</div>
	<u>Violet électrique</u>	taille : 42/46	9,00 €	x	<div><div>+</div><div>-</div></div> <div>1</div>	=	9,00 €	<div>SUPPR.</div>
Port :							6,00 €	
Total :							21,00 €	
Pour obtenir la gratuité des frais de port, vous devez encore commander pour 21,00 €								

# Créer un tableau : <table>

- ! La balise **<table>** **</table>** permet d'indiquer le début et la fin du tableau.
- ! La balise **<tr>** caractérise une nouvelle ligne
- ! La balise **<td>** marque le contenu d'une cellule dans une ligne

# Exemple de tableau simple

```
<table>
  <tr>
    <td>ligne 1 colonne 1</td>
    <td>ligne 1 colonne 2</td>
    <td>ligne 1 colonne 3</td>
  </tr>
  <tr>
    <td>ligne 2 colonne 1</td>
    <td>ligne 2 colonne 2</td>
    <td>ligne 2 colonne 3</td>
  </tr>
</table>
```

ligne 1 colonne 1	ligne 1 colonne 2	ligne 1 colonne 3
ligne 2 colonne 1	ligne 2 colonne 2	ligne 2 colonne 3

Le nombre d'éléments **<td>** au sein des éléments **<tr>** doit rester le même dans la mesure où chaque ligne possède un même nombre de cellules, nombre qui correspond au nombre de colonnes du tableau.



# Ajouter une entête avec `<th>`

```
<table>
  <tr>
    <th>Fruits</th>
    <th>Poids</th>
    <th>Prix</th>
  </tr>
  <tr>
    <td>Kiwi</td>
    <td>3kg</td>
    <td>2€</td>
  </tr>
  <tr>
    <td>Banane</td>
    <td>2kg</td>
    <td>1€</td>
  </tr>
</table>
```

- ! `<th>` peut remplacer `<td>` pour créer une cellule d'entête au tableau
- ! Par défaut affiché en gras et centré dans les navigateurs

Fruits	Poids	Prix
Kiwi	3kg	2€
Banane	2kg	1€

# Ajouter un titre au tableau

```
<table>
  <caption>Prix du marché</caption>
  <tr>
    <th>Fruits</th>
    <th>Poids</th>
    <th>Prix</th>
  </tr>
  <tr>
    <td>Kiwi</td>
    <td>3kg</td>
    <td>2€</td>
  </tr>
  <tr>
    <td>Banane</td>
    <td>2kg</td>
    <td>1€</td>
  </tr>
</table>
```

- ! On utilise **<caption>** pour donner un titre au tableau
- ! La balise se place au début du tableau

Fruits	Poids	Prix
Kiwi	3kg	2€
Banane	2kg	1€

# Bonus - Fusion de cellules

```
<table>
  <tr>
    <th colspan="3">titre</th>
  </tr>
  <tr>
    <td>ligne 2 colonne 1</td>
    <td>ligne 2 colonne 2</td>
    <td>ligne 2 colonne 3</td>
  </tr>
</table>
```

- ! Il est possible de fusionner certaines cellules entre elles avec l'attribut **colspan="nombredecellules"** qui se place sur la cellule (th ou td)
- ! La valeur numérique de cet attribut précise le nombre de colonnes du tableau que couvre la cellule.

titre		
ligne 2 colonne 1	ligne 2 colonne 2	ligne 2 colonne 3

# Bonus - Fusion de lignes

```
<table>
  <tr>
    <td rowspan="2">ligne 1 et 2 colonne 1</td>
    <td>ligne 1 colonne 2</td>
    <td rowspan="3">ligne 1,2,3 colonne 3</td>
  </tr>
  <tr>
    <td>ligne 2 colonne 2</td>
  </tr>
  <tr>
    <td>ligne 3 colonne 1</td>
    <td>ligne 3 colonne 2</td>
  </tr>
</table>
```

- Il est possible de fusionner des lignes avec l'attribut **rowspan="nombre de lignes"** qui se place sur la cellule (th ou td)

ligne 1 et 2 colonne 1	ligne 1 colonne 2	ligne 1,2,3 colonne 3
	ligne 2 colonne 2	
ligne 3 colonne 1	ligne 3 colonne 2	

# Les formulaires

---

# Les formulaires sont partout autour de nous

- Le formulaire sert à récupérer des données que l'utilisateur va entrer et les envoyer au serveur pour un traitement (en PHP par exemple) : login, formulaire de contact et même le chat de Facebook

amazon.fr

Ouvrir une session

Quelle est votre adresse e-mail ?

Votre adresse e-mail est:

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

- ☐ Vous êtes un nouveau client.  
(vous créerez un mot de passe plus tard)
- ☒ Vous êtes déjà client Amazon ?

Votre mot de passe est :

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

[Continuer \(serveur sécurisé\)](#)

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

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

Statut Photo/vidéo

un formulaire pour m'exprimer, chouette ! |



Public

Publier

dribbble  
Sign in

Username or Email

Password [Forgot?](#)

Sign in

Sign in with Twitter

One-click sign in to Dribbble if your account is connected to Twitter. We'll walk you through connecting it if it isn't.

Not a member? [Sign up as a spectator](#)

CONTACT US

Name

Email Address

Website (optional)

Message

SUBMIT

across and/or are work

- Resizing/zooming in
- Flash of unstyled co
- JavaScript replaces
- a1t text is not presen
- HTML right-click op
- Incompatible with o
- or MooTools
- Doesn't work with h
- security restrictions

Find a problem on your

Support th

swfIR is a labor of love,

# La balise <form>

- Balise principale du formulaire
- Deux méthodes :
  - method="get" : limitée à 255 caractères, informations passées dans la barre d'adresse
  - method="post" : envoie les données dans le corps de la requête sans passer par la barre d'adresse, c'est la méthode la plus utilisée
- Action : l'adresse du fichier ou programme qui va traiter les données

`<form method="post" action="traitement.php">`

# Exemple de la balise form en action

```
<body>  
  
<form method="post" action="traitement.php">  
    <!-- contenu du formulaire -->  
  
</form>  
  
</body>
```

Par défaut si elle est vide, c'est comme n'importe quel élément, rien n'est affiché



# La zone de texte monoligne <input>

```
<input type="text" name="nom_de_la_balise" />
```

- ! La balise input est auto-fermante
- ! Ne génère pas de retour à la ligne.
- ! Il faut lui donner un nom avec l'attribut « **name** » pour que l'on puisse récupérer sa valeur (éviter les espaces dans le name). Le nom n'est PAS visible sur la page

# Le label

```
<label for="nom">Votre nom</label>  
<input type="text" id="nom" name="nom" />
```

Votre nom

- ! Son rôle est de décrire à un être humain la fonction du champ (puisque name est invisible)
- ! Ajouter un **for="nom"** au label
- ! Ajouter **id="nom"** au champ

# Associer le label au champ

```
<label for="nom">Votre nom</label>  
<input type="text" id="nom" name="nom" />
```

Votre nom

- Permet à l'utilisateur de cliquer sur le label pour sélectionner le champ (le for et l'id doivent être identiques mais peuvent être différents du name)
- Balise inline, pas de retour à la ligne

## L'attribut « value »

```
<label for="nom">Votre nom</label>  
<input value="Stéphanie" type="text" id="nom"  
name="nom" />
```

Votre nom

- ! **value=" "** : donne une valeur par défaut au champ
- ! Vide (ou inexistant) si on n'a pas de données récupérées dans la base de donnée du site

# En résumé pour n'importe quel champ il faut toujours au moins

- Un label (avec un for=" ")
- Un champ avec :
  - Un id (qui a la même valeur que le for)
  - Un name pour entrer la valeur dans la base de donnée
- Facultatif : un attribut value (qui peut être vide) pour récupérer la valeur dans la base de donnée

# Attributs supplémentaires

- ! **maxlength=" "** : le nombre de caractères maximum acceptés dans le champ

# Les placeholder (ne sont pas des labels !)

```
<p>
  <label for="nom">Votre nom</label>
  <input value="Stéphanie" type="text" id="nom" name="nom"/>
</p>
<p>
  <label for="prenom">Votre Prénom</label>
  <input placeholder="ex : stéphanie" type="text" id="prenom" name="prenom" />
</p>
```

Votre nom

Votre Prénom

**placeholder=" "** : attribut HTML5, donne une indication de ce que devrait contenir le champ.

Attention, ne remplace PAS le label

## Zone de texte multiligne <textarea>

```
<label for="montext">Que souhaitez-vous me dire ?</label>  
<textarea name="montext" id="montext"></textarea>
```

Que souhaitez-vous me dire ?

- Une balise qu'il faut ouvrir et fermer
- Pas de retour à la ligne automatique



# Les cases à cocher <input type="checkbox">

```
<input type="checkbox" id="jazz" name="jazz" >
```

Quel type de musique aimez vous ?

- ☐ Le Jazz
- ☒ La techno
- ☐ Le rock

- Une case = une option = un input
- Plusieurs cases, toutes peuvent être cochées
- Éléments inline
- On peut mettre le label à gauche ou à droite

# Les cases à cocher <input type="checkbox">

```
<p>
  <input type="checkbox" name="jazz" id="jazz">
  <label for="jazz">Le Jazz</label>
</p>
<p>
  <input type="checkbox" name="techno" id="techno" checked="checked">
  <label for="techno">La techno</label>
</p>
<p>
  <input type="checkbox" name="rock" id="rock">
  <label for="rock">Le rock</label>
</p>
```

Quel type de musique aimez vous ?

- ☐ Le Jazz
- ☒ La techno
- ☐ Le rock

Les cases à cocher peuvent avoir un attribut name distincts, ou identique

Si la case est cochée, il renvoie « on » lors du traitement, ou peuple le tableau PHP de sa valeur (value)

# Les cases à cocher <input type="checkbox">

```
<p>  
  <input type="checkbox" name="jazz" id="jazz">  
  <label for="jazz">Le Jazz</label>  
</p>  
<p>  
  <input type="checkbox" name="techno" id="techno" checked="checked">  
  <label for="techno">La techno</label>  
</p>  
<p>  
  <input type="checkbox" name="rock" id="rock">  
  <label for="rock">Le rock</label>  
</p>
```

Quel type de musique aimez vous ?

- ☐ Le Jazz
- ☒ La techno
- ☐ Le rock

On peut pré-cocher une checkbox avec **checked="checked"** ou just checked

# Les boutons radio <input type="radio">

```
<input type="radio" id="sexe" name="femme" >
```

Vous êtes : ☐ une femme ☒ un homme ☐ je préfère ne pas le dire

- Une case = une option = un input
- Un seul choix possible parmi tous les éléments
- Éléments inline

## Les boutons radio <input type="radio">

```
<p> Vous êtes :  
    <input type="radio" name="sexe" value="femme" id="femme"/><  
    label for="femme"> une femme </label>  
</p>
```

- La value est importante : c'est elle qui est renvoyée au serveur
- **value="femme"** : on renvoie donc « femme » au serveur

# Groupe de boutons radio

```
<p> Vous êtes :  
  <input type="radio" name="sexe" value="femme" id="femme"/><  
  label for="femme"> une femme </label>  
  <input type="radio" name="sexe" value="homme" id="homme"  
  checked="checked"/><label for="homme"> un homme </label>  
  <input type="radio" name="sexe" value="nsp" id="nsp"/><label  
  for="nsp"> je préfère ne pas le dire </label>  
</p>
```

Vous êtes : ☐ une femme ☒ un homme ☐ je préfère ne pas le dire

Pour que le navigateur comprenne que des éléments forment un groupe, il faut leur **donner le même attribut name**

On peut pré-cocher un bouton radio avec **checked="checked"**

# Les listes déroulantes <select>

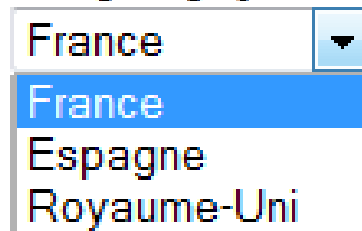
```
<p>
  <label for="pays">De quel pays venez vous ?</label>
  <select name="pays" id="pays">
    <option value="france">France</option>
    <option value="espagne">Espagne</option>
    <option value="royaume-uni">Royaume-Uni</option>
  </select>
</p>
```

De quel pays venez vous ?

France	▼
France	
Espagne	
Royaume-Uni	

# Les listes déroulantes <select>

De quel pays venez vous ?



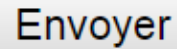
France	▼
France	
Espagne	
Royaume-Uni	

- ! La liste se trouve entre les éléments **<select id="nomdeliste">** **</select>**
- ! La balise **<label for="nomdeliste">** permet de donner le nom de la liste et est liée à <select>
- ! On crée un objet de la liste avec la balise **<option>** Nom **</option>**
- ! L'attribut value permet de donner la valeur de l'option de la liste



# Bouton d'envoi input type="submit"

```
<input type="submit" value="Envoyer" />
```



- ! Pour envoyer le formulaire on utilise **input type="submit"**
- ! L'attribut value=" " est obligatoire et permet de donner le « titre » du bouton
- ! Le formulaire est envoyé pour traitement au fichier que l'on a défini dans action=" "

# Petit résumé des champs

input type = text

input type = checkbox

input type = radio

input type= submit

select + option

textarea

Votre nom

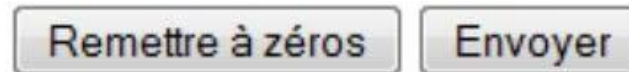
☐ Le Jazz ☒ La techno

☐ une femme ☒ un homme

Que souhaitez-vous me dire ?

## Bouton de reset, input type="reset"

```
<p> <input type="reset" value="Remettre à zéros" /> <input type="submit" value="Envoyer" /> </p>
```



- Pour remettre à zéro toutes les données du formulaire on utilise **input type="reset"**
- Merci de ne pas en abuser.

# Bouton <button type="button">

```
<p>  
  <button type="button">Cliquez moi</button>  
  <button type="submit">Envoyer</button>  
  <button type="reset">Reset</button>  
</p>
```

- ! Syntaxe différente de la balise input
- ! Possibilité d'ajouter des images à l'intérieur (balise non auto-fermante)
- ! Trois types :
  - **button** : bouton cliquable (utilisable en JavaScript)
  - reset : remettre à zéro le formulaire
  - submit : envoyer le formulaire

# Désactiver un champ : disabled="disabled"

```
<input type="submit" value="Envoyer" disabled="disabled"/>
```



- ! L'attribut **disabled="disabled"** peut-être utilisé pour désactiver un champ.
- ! Exemple : désactiver le champ d'envoi jusqu'à ce que toutes les données soient remplies
- ! Note : on peut aussi juste mettre **disabled** en attribut

# Les champs de mot de passe <input type="password">

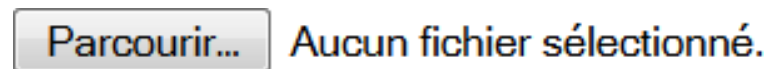
```
|  
<input type="password" id="password" name="password" />
```

Mon mot de passe type="password"

- ! **input type="password"** : permet de masquer un mot de passe

## Uploader un fichier : type = file

```
<input name= "myfile" type="file" />
```



- ! Pour télécharger vers le serveur un fichier on utilise **input type="file"**
- ! Si on envoie un fichier, il faut permettre au formulaire d'envoyer des données avec l'attribut **enctype="multipart/form-data"** sur l'élément form.

# Cacher un champ type="hidden"

```
<h2> Cachez moi ce champ ! </h2>  
<input type="hidden" name="key" value="masupercleeseçrète" />  
<label for="nom">Nom</label><input type="text" name="nom" id="nom"/>
```

**Cachez moi ce champ !**

Nom

- ! On peut cacher un champ avec **input type="hidden"**
- ! Permet d'envoyer des données sans que l'utilisateur ne doive les remplir ou ne les voir.



# Liste non exhaustive de quelques balises

- ! body
- ! div
- ! p
- ! h1 - h6
- ! ul
- ! ol
- ! li
- ! blockquote
- ! form
- ! table
- ! iframe
- ! etc.

Memento des balises HTML