

## Module 2

# Structure HTML

La base de toute page Web



## Table des matières

<b>STRUCTURE HTML.....</b>	<b>4</b>
<b>HTML, XHTML, HTML5 ?.....</b>	<b>4</b>
Une base commune.....	4
Différences et évolutions.....	4
<b>Balises et attributs.....</b>	<b>6</b>
<b>Document HTML minimal.....</b>	<b>7</b>
Balise <!doctype>.....	7
Balise <html>.....	7
Balise <head>.....	8
Balise <body>.....	8
Structure minimale XHTML1.0.....	9
Structure minimale HTML5.....	9
Balises inline et balise block.....	11
block.....	11
inline.....	12
Imbrication des balises.....	13
Principe d'imbrication.....	13
Principe inline et block en XHTML (assouplissement en HTML5).....	14
Exceptions inline et block.....	14
Liste des balises HTML et leurs définitions en ligne.....	16
<b>Formatage du texte.....</b>	<b>17</b>
Titres et paragraphes.....	17
Titres.....	17
Paragraphes et retour à la ligne.....	17
Texte.....	17
Les listes.....	18
3 types de listes.....	18
Exemples.....	18
Les liens.....	20
Balise et attributs d'un lien Hypertexte.....	20
Lien absolu :.....	20
Lien relatif :.....	20
Lien vers une ancre.....	21

Lien vers une adresse mail.....	21
Lien un numéro de téléphone.....	21
Lien pour télécharger un fichier (.doc, .pdf, .zip).....	21
Les images.....	22
Balise image (auto-fermante).....	22
Principaux attributs.....	22
Formats d'images.....	22
Droits des images.....	23
Zoom.....	23
Les tableaux.....	24
Corps du tableau.....	24
Titre du tableau.....	24
Ligne d'un tableau.....	24
Cellules d'un tableau.....	24
Exemple.....	25
<b>Structure d'un site Web.....</b>	<b>26</b>
L'élément <div> en HTML4 et HTML5.....	26
Exemple.....	26

# STRUCTURE HTML

## HTML, XHTML, HTML5 ?

### Une base commune

HTML (Hypertext Markup Language), langage de balisage permettant les liens hypertextes constitue la structure de base d'une page Web.

D'autres types de fichiers sont liés à cette structure de base : CSS (Cascading Style Sheets) pour personnaliser l'apparence, JavaScript pour l'interactivité, PHP permettant les relations à la base de données, des animations sous flash, flux xml...

Le langage HTML 4 était mature à la fin des années 90. Une fois HTML 4.0.1 finalisé, l'équipe de [W3C](#) \* poursuit l'évolution du Web en lançant un langage appelé XHTML 1.0 (eXtensible HyperText Markup Language), le document pouvant être traité comme du HTML mais aussi comme du XML

Allez sur un site de votre choix, clic droit dans la page "code source de la page" permet de voir le code HTML.

### Différences et évolutions

Les différences entre HTML 4.0.1 et XHTML 1.0 sont mineures essentiellement d'ordre syntaxiques :

- Toute balise ouvrante doit être fermée, et les balises dites «vides» sont écrites avec une barre oblique finale (exemple: `<br />`).
- Les noms des balises et des attributs sont écrits en minuscules.
- Les valeurs des attributs sont encadrés par des 'quotes' (apostrophes droites) ou des "double quotes" (guillemets droits).
- Chaque attribut doit avoir une valeur (pas d'attribut vide comme checked, qui doit être écrit `checked="checked"`).

Par contre, de grandes différences entre HTML4 et HTML5 avec l'apparition de **nouvelles balises**.

## HTML5

Le HTML5 a officiellement été lancé par le W3C\* en 2014 et la majorité des sites sont aujourd'hui créés en HTML5

Navigateurs supportant HTML5 :

- Firefox 4.0 et +
- Chrome 11.0 et +
- Safari 5.0 et +
- Internet explorer 9 et +
- Opéra 11.0

Cependant, à l'heure actuelle (2017) la plupart des sites sont encore construits en XHTML 1.0. Il est donc important de voir l'évolution de ce langage.

*\* W3C est un sigle utilisé pour définir le World Wide Web Consortium qui est une organisation non lucrative dont la mission est de définir les standards pour les technologies liées au Web.*

# Balises et attributs

**Balise** : caractère ou une série de caractères. Utilisé pour la structuration d'un document et qui sera invisible par le lecteur final.

`<p></p>` : balise ouvrante et balise fermante d'un paragraphe.

La plupart des balises doivent être fermées en HTML mais certaines sont **auto-fermante**.

Dans ce cas ne pas oublier de laisser un espace avant le trait oblique :

`<input />` : boutons et champs de saisie des formulaires)

`<br />` : retour à la ligne.

De préférence écrites en lettres minuscules.

**Attribut** : propriété de la balise à laquelle il est rattaché. Doit être écrit en minuscule. La valeur de l'attribut est écrite entre guillemets. Une même balise peut avoir plusieurs attributs.

Exemple d'attribut associé à la balise p :

```
<p class="intro">Ceci est un texte d'introduction</p>
```

- Nom de la balise : p
- Nom de l'attribut : class
- Valeur de l'attribut : `intro`

Exemple d'attribut associé à la balise input :

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

- Nom de la balise (auto fermante) : input
- Nom de l'attribut : type
- Valeur de l'attribut : text

Exemple d'attributs associés à la balise input :

```
<input type="button" value="OK" />
```

- Nom de la balise : input
- Avec 2 attributs :
  - type > button
  - value > OK

# Document HTML minimal

## Balise <!doctype>

La balise doctype (= Déclaration de type de document) est la première d'une page HTML. Elle sert à indiquer au navigateur la façon dont il doit interpréter la suite du code contenu dans la page. Également, elle sert d'indicateur pour une validation par le w3c ([validator.w3.org](http://validator.w3.org)).

Voici les principales DTD :

### HTML4.01 transitional

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

### HTML4.01 strict

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

### XHTML1.0 transitional

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

### XHTML1.0 strict

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

### HTML5

```
<!DOCTYPE html>
```

Remarque : Le mode strict en XHTML considère l'utilisation de certaines balises issues de HTML4 comme mal venues : dir, menu, center, isindex, applet, font, basefont, s, strike.

## Balise <html>

Cet élément délimite tous ceux qui suivent et spécifie au navigateur qu'il s'agit d'un document HTML (ou XHTML)

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

## Balise <head>

Cette section comprend des informations sur la page mais son contenu n'apparaît pas dans la page elle-même.

Compris dans la partie head :

a) La balise `<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />` qui permet de fournir des informations sur l'encodage de la page.

Pour une structure pouvant s'adapter à différentes langues (cas d'un site multilingue) nous choisissons le charset UFT-8. Il existe d'autres types d'encodage (iso-8859-1).

En savoir plus : <http://www.alsacreations.com/astuce/lire/34-charset-iso-8859-1-iso-8859-15-utf-8-lequel-choisir.html>

b) La balise `<title></title>` doit être renseignée avec soin car elle apparaît :

- en intitulé de l'onglet de votre page
- mais également dans les moteurs de recherches.

C'est donc une balise clé pour le référencement d'un site.



## Balise <body>

C'est le corps du document. Tout ce qui se trouve dans cette section peut être visible dans le navigateur.



## Structure minimale XHTML1.0

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

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" lang="fr">

<head>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
  <title>Titre du document</title>
</head>

<body>
  <!-- Partie Visible -->
</body>

</html>
```

## Structure minimale HTML5

```
<!doctype html>
<html lang="fr">
<head>
  <meta charset="UTF-8" />
  <title>Titre du document</title>
</head>

<body>
  <!-- Partie Visible -->
</body>
</html>
```

### Commentaires

Tout le texte compris entre les symboles `<!--` et `-->` n'est pas visible à l'écran, il vous permettra de commenter votre code pour vous y retrouver, par exemple : `<!-- menu principal -->` Cela aidera aussi d'autres personnes (graphistes, développeurs) qui devront travailler sur votre code.

## TP guidé

1- Avec Notepad++ créez un nouveau fichier html5 : "tests.html" enregistré dans [page].

2- Entre <body> et </body> copiez collez le code suivant :

```
<input type="button" value="OK" />
```

3- Changez "OK" en "Valider"

4- Visualiser les changements dans votre navigateur

5- Essayez d'autres balises plus bas dans le cours

## **TP4 structure html**

## Balises inline et balise block

### block

Les éléments de rendu CSS **block** se placent toujours l'un en dessous de l'autre par défaut (comme un retour chariot). Par exemple: une suite de paragraphes (balises `<p></p>`).

Par ailleurs, un élément "bloc" occupe automatiquement, par défaut, toute la largeur disponible dans son conteneur.

```
<p>Paragraphe 1</p>  
<p>Paragraphe 2</p>
```

Ces deux paragraphes vont s'afficher sur deux lignes, car la balise `<p>` est par défaut un élément de rendu "bloc" : chaque paragraphe va occuper une ligne.

#### + Liste des éléments de type block :

- [ADDRESS](#) - Address
- [BLOCKQUOTE](#) - Block quotation
- [CENTER](#) - Centered block
- [DIR](#) - Directory list
- [DIV](#) - Generic block-level container
- [DL](#) - Definition list
- [FIELDSET](#) - Form control group
- [FORM](#) - Interactive form
- [H1](#) - Level-one heading
- [H2](#) - Level-two heading
- [H3](#) - Level-three heading
- [H4](#) - Level-four heading
- [H5](#) - Level-five heading
- [H6](#) - Level-six heading
- [HR](#) - Horizontal rule
- [ISINDEX](#) - Input prompt
- [MENU](#) - Menu list
- [NOFRAMES](#) - Frames alternate content
- [NOSCRIPT](#) - Alternate script content
- [OL](#) - Ordered list
- [P](#) - Paragraph

- [PRE](#) - Preformatted text
- [TABLE](#) - Table
- [UL](#) - Unordered list

## inline

Les éléments de rendu **inline** se placent toujours l'un à côté de l'autre afin de rester dans le texte (par exemple la balise `<strong>`).

Par défaut, il n'est pas prévu qu'ils puissent se positionner sur la page (même si cela est possible), ni de leur donner des dimensions (hauteur, largeur, profondeur) : leur taille va être déterminée par le texte ou l'élément qu'ils contiennent.

### + Liste des éléments de type inline

- [A](#) - Anchor
- [ABBR](#) - Abbreviation
- [ACRONYM](#) - Acronym
- [B](#) - Bold text
- [BASEFONT](#) - Base font change
- [BDO](#) - BiDi override
- [BIG](#) - Large text
- [BR](#) - Line break
- [CITE](#) - Citation
- [CODE](#) - Computer code
- [DFN](#) - Defined term
- [EM](#) - Emphasis
- [FONT](#) - Font change
- [I](#) - Italic text
- [IMG](#) - Inline image
- [INPUT](#) - Form input
- [KBD](#) - Text to be input
- [LABEL](#) - Form field label
- [Q](#) - Short quotation
- [S](#) - Strike-through text
- [SAMP](#) - Sample output
- [SELECT](#) - Option selector
- [SMALL](#) - Small text
- [SPAN](#) - Generic inline container

- [STRIKE](#) - Strike-through text
- [STRONG](#) - Strong emphasis
- [SUB](#) - Subscript
- [SUP](#) - Superscript
- [TEXTAREA](#) - Multi-line text input
- [TT](#) - Teletype text
- [U](#) - Underlined text
- [VAR](#) - Variable

Source : <http://htmlhelp.com/>

## Imbrication des balises

### Principe d'imbrication

Les balises doivent être imbriquées selon un ordre logique, les **balises parents contenant les balises enfants** (poupées Russes).

Bonne construction :

```
<div>
  <p>
    Ici mon <strong>texte</strong>
  </p>
  <p>
    Un autre texte
  </p>
</div>
```

Mauvais :

```
<div>
  <p>
    Ici mon <strong>texte</strong>
</div>
</p>
```

Pour y voir « plus clair » il est fortement conseillé d'**indenter son code** c'est à dire de décaler les balises enfants par rapport aux balises parents.

L'ouverture et la fermeture de balise se faisant sur la même ligne verticale.

Remarque : les éditeurs de codes proposent parfois un outil d'indentation automatique de code (plus ou moins efficace).

## Principe inline et block en XHTML (assouplissement en HTML5)

- Un élément de type block peut contenir un (ou plusieurs) autres éléments Block et/ou inline.
- Une élément du type inline ne peut contenir QUE un (ou plusieurs autres) éléments inline.

Bonne construction

```
<h1>Il y a un <a href="#">lien</a> dans ce titre</h1>
```

Mauvais

```
<a href="#"><h1>Ce titre est contenu dans un lien</h1></a>
```

## Exceptions inline et block

### + Éléments de type bloc (block), exceptions :

ADDRESS

En HTML (et XHTML) Strict, la balise ADDRESS ne peut contenir que des éléments en-ligne. En transitionnel, elle peut contenir également la balise P. Parents possibles pour cette balise : BLOCKQUOTE, BODY, BUTTON, DD, DEL, DIV, FIELDSET, FORM, INS, LI, MAP, NOSCRIPT, OBJECT, TD, TH

**BODY**

Ne peut pas être parent direct de caractères ou d'éléments de type En-ligne.

BLOCKQUOTE

En HTML (et XHTML) Strict, la balise BLOCKQUOTE ne peut être parente que d'éléments de type Bloc. En transitionnel, elle peut également être parente d'éléments de type En-ligne.

DL

Ne peut être parent direct **que** des éléments DT et/ou DD

DT

Ne peut pas contenir d'éléments blocs

FIELDSET

Doit contenir en premier l'élément LEGEND.

FORM

Ne peut être parent direct que d'éléments blocs. Ne peut pas contenir d'autres éléments FORM

**H1, H2,... H6**

Ne peut être parent que d'éléments en-ligne.

HR

Ne peut pas contenir d'éléments.

NOSCRIPT

Ne peut contenir directement que des éléments de type bloc.

**OL et UL**

Ne peut contenir directement que des éléments de liste LI.

**P**

Ne peut être parent que d'éléments en-ligne.

PRE

Ne peut être parent que d'éléments en-ligne, sauf IMG, OBJECT, APPLET, SUB, SUP.

TABLE

Peut être parent direct des balises suivantes : TR, CAPTION, THEAD, TFOOT, TBODY, COL, COLGROUP.

### **+ Éléments en ligne (inline), exceptions :**

**A**

Ne peut pas contenir d'autres éléments A.

BR

Ne peut pas contenir d'éléments.

IMG

Ne peut pas contenir d'éléments. Ne peut pas être contenu dans un élément PRE.

INPUT

Ne peut pas contenir d'éléments. Ne peut pas être contenu dans un élément BUTTON.

LABEL

Ne peut pas contenir d'autres éléments LABEL. Ne peut pas être contenu dans un élément BUTTON.

SELECT

Peut être parent direct des éléments OPTGROUP ou OPTION . Ne peut pas être contenu dans un élément BUTTON.

TEXTAREA

Ne peut contenir que du texte simple et des entités. Ne peut pas être contenu dans un élément BUTTON.

Source : [alscreations](#)

## Liste des balises HTML et leurs définitions en ligne

- <http://www.vectorskin.com/referentiels-standards-w3c/balises-html5/>  
(bien présenté, clair)
- <http://www.w3schools.com/tags/>  
(plus complet mais en Anglais)
- <http://www.startyourdev.com/html/tag-html-index>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element>  
(exhaustif)

**TP5 debug**



# Formatage du texte

## Titres et paragraphes

### Titres

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

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

```
<h6>Dernier niveau</h6>
```

**Titre**

**Sous-titre**

**Dernier niveau**

### Paragraphes et retour à la ligne

```
<p>1er paragraphe.</p>
```

```
<p>Autre paragraphe.<br />
Retour à la ligne</p>
```

1er paragraphe.

Autre paragraphe.  
Retour à la ligne

### Texte

```
<strong></strong> Mise en exergue (gras)
```

```
<em></em> Emphase (italique)
```

```
<br /> Retour à la ligne
```

```
<hr /> Ligne horizontale
```

```
<blockquote></blockquote> Citation longue
```

```
<q></q> Citation courte
```

```
<abbr></abbr> Abréviation
```

Attribut: title - Message au survol

Exemple: `<abbr title="Hypertext Markup Language">HTML</abbr>`

```
<span></span> Élément générique de type inline
```

```
<small></small> Sert à déclarer un contenu non essentiel (copyright ...)
```

# Les listes

## 3 types de listes

Listes à puces, liste non triée

```
<ul>..</ul>
```

Liste triée, liste à numéro

```
<ol>..</ol>
```

Element de liste

```
<li></li>
```

## Exemples

```
<ul>
  <li>Un élément</li>
  <li>Un autre</li>
  <li>Encore un autre</li>
</ul>
```

- Un élément
- Un autre
- Encore un autre

```
<ol>
  <li>Premier élément</li>
  <li>Deuxième</li>
  <li>Troisième</li>
</ol>
```

1. Premier élément
2. Deuxième
3. Troisième

```
<ul>
  <li>Un élément de niveau 1
    <ul>
      <li>Un élément de niveau 2</li>
      <li>Un autre de niveau 2</li>
      <li>Encore un autre de niveau 2</li>
    </ul>
  </li>
  <li>Un autre de niveau 1</li>
  <li>Encore un autre de niveau 1</li>
</ul>
```

- Un élément de niveau 1
  - Un élément de niveau 2
  - Un autre de niveau 2
  - Encore un autre de niveau 2
- Un autre de niveau 1
- Encore un autre de niveau 1

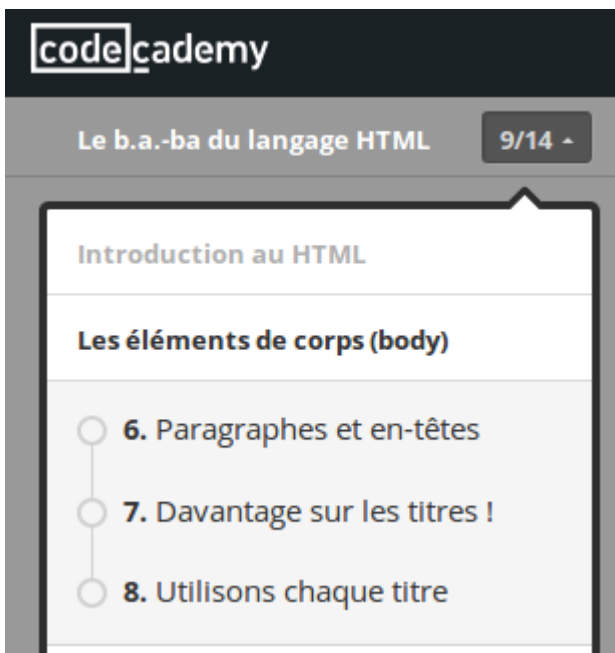
## TP Guidé

Entretenez vous à faire des listes imbriquées dans votre fichier "tests.html", vous pouvez jeter un œil à cet article :

<http://www.alsacreations.com/astuce/lire/66-mes-listes>

## TP Autonome

Créez un compte sur <https://www.codecademy.com/> et commencez les tutos HTML/CSS jusqu'au point 8.



# Les liens

## Balise et attributs d'un lien Hypertexte

### Balise

```
<a></a>
```

### Attributs

- `href` - La valeur est l'adresse de la page vers laquelle pointe ce lien (obligatoire, sinon pourquoi faire un lien ?)
- `title` - La valeur sera affichée en infobulle
- `target="_blank"` - Permettra d'ouvrir le lien dans un nouvelle onglet. Créer un site en « abusant » des `target_blank` est considéré comme une mauvaise pratique (« prolifération » de fenêtres ouvertes). Il est d'usage de laisser l'internaute décider : CTRL+clic pour ouvrir dans un nouvel onglet / SHIFT+clic pour ouvrir dans une nouvelle fenêtre.

### Lien absolu :

```
<a href="http://www.grafikart.fr/tutoriels" target="_blank">Grafikart</a>
```

### Lien relatif :

```
<a href="page2.html" title="Vers la page 2 de ce site">Page 2</a>
```

[Page 2](#)

Vers la page 2 de ce site

Si je veux faire un lien vers la page2 se trouvant dans un sous-dossier intitulé [page] mon lien sera de la forme :

```
<a href="page/page2.html" title="Vers la page 2 de ce site">Page 2</a>
```

Au contraire, si je suis sur la page 2 et veux ressortir du dossier pour atteindre mon index par exemple, la syntaxe sera :

```
<a href="../index.html" title="Retour à l'accueil">Accueil</a>
```

## Lien vers une ancre

Exemple, pour revenir en haut de la page :

```
<body>
  <h1 id="titre-principal">Ici mon titre</h1>
  <p>
    Un paragraphe long
  </p>
  <p>
    <a href="#titre-principal" title="Aller en haut de la page">
      Haut de page
    </a>
  </p>
</body>
```

## Lien vers une adresse mail

```
<a href="mailto:damien@gmail.com">Envoyer un mail</a>
```

Déclenchera l'ouverture d'un message vide avec adresse pré remplie sur votre logiciel de messagerie. Attention, beaucoup d'Internautes n'ont pas de logiciel (Thunderbird, Outlook) correctement configuré, ce procédé risque de les bloquer.

## Lien un numéro de téléphone

```
Me contacter au :<a href="tel:0562894578">05.62.89.45.78</a>
```

Appel au clic (smartphone)

## Lien pour télécharger un fichier (.doc, .pdf, .zip)

```
<a href="cours.zip">Télécharger les cours</a>
```

Le navigateur reconnaît automatiquement le type de fichier et vous propose la procédure appropriée pour le télécharger ou l'afficher.

## TP Guidé

Faites un lien depuis la page tests.html vers la page définitions.html au clic sur la phrase : "Voir les mots clés du Web"

## Les images

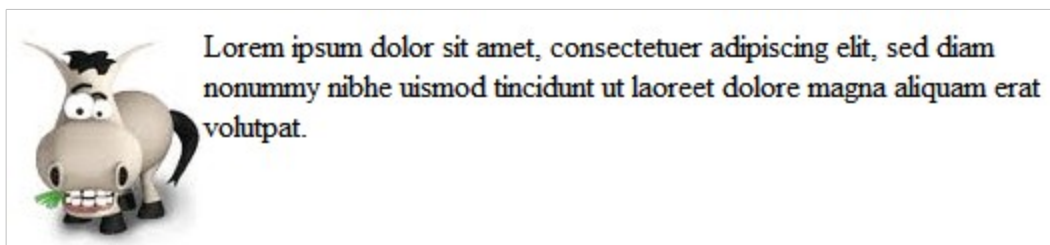
### Balise image (auto-fermante)

```
<img />
```

### Principaux attributs

- **src** - Valeur : URL vers le fichier image (obligatoire, sinon l'image ne s'affiche pas !)
- **alt** - Valeur : texte descriptif de l'image (important pour l'accessibilité et s'affiche en remplacement de l'image si le lien est brisé)
- **width** et **height** - Valeurs : Largeur et hauteur de l'image en pixels (important)
- **align** (déprécié en HTML5) - Valeurs : left, bottom, middle, top, right

```
<p>  
  
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam  
nonummy nibhe uismod tincidunt ut laoreet dolore magna aliquam  
</p>
```



### Formats d'images

Attention au **poids des images**, elles doivent être optimisées pour le Web. C'est à dire avoir un poids ne dépassant pas 500 ko tout en gardant une bonne qualité. Résolution libre (pas fixée à 72 dpi) : <https://www.lesintegrates.net/2011/05/06/web-resolution-72dpi/>

Dans photoshop : Fichier > **enregistrer pour le Web** et les périphériques.

Choisissez le format d'enregistrement qui allège le plus l'image entre jpg - gif - png - svg, redimensionnez-là au besoin et jouez sur la qualité du jpg.

## Droits des images

Enfin, attention aux droits des images. Seules les photos prises avec votre propre matériel vous appartiennent. Il vous faut aussi l'autorisation des personnes figurant sur la photo. Certains sites proposent des images libres de droits : <https://unsplash.com/> - <https://pixabay.com/>

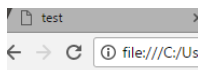
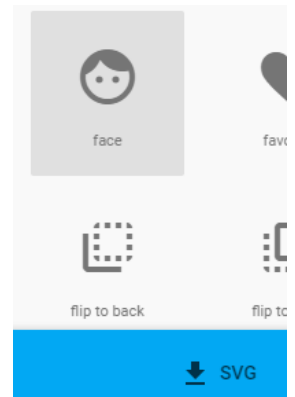
## Zoom

Format **SVG**, s'insère dans la page grâce à la balise <img> mais également en code (syntaxe xml). Voir l'article d'Alsacreations : <https://www.alsacreations.com/tuto/lire/1421-svg-initiation-syntaxe-outils.html>

## TP guidés

### 1- SVG

- Allez sur le site : <https://material.io/icons/>
- Cliquez sur une icône de votre choix et téléchargez-là au format SVG
- Ouvrez le fichier téléchargé avec votre éditeur (brackets)
- Copiez-collez le code dans votre fichier *tests.html*. Entre les balises <body>
- Essayez de changer la couleur, agrandir l'icône en modifiant ce code.



### 2- Photoshop

Alléger avec photoshop «image-lourde» qui se trouve dans [documents] en «enregistrant pour le web» et en choisissant le format le plus performant (qualité/poids).

## TP6 lien img

# Les tableaux

## Corps du tableau

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

Principaux attributs :

- **align** - position (left, right, center)
- **bgcolor** - couleur de fond du tableau
- **border** - bordure en pixel
- **cellpadding** - marge intérieure des cellules en pixel
- **cellspacing** - espacement des cellules en pixel
- **width** - largeur en pixel ou %

## Titre du tableau

```
<caption></caption>
```

## Ligne d'un tableau

```
<tr></tr>
```

Principaux attributs :

- **align** - alignement horizontal du texte (left, center, right)
- **valign** - alignement vertical du texte (top, middle, bottom)
- **bgcolor** - couleur de la ligne
- **border** - bordure en pixel

## Cellules d'un tableau

```
<th></th> Cellule de titre  
<td></td> Cellule de données
```

Principaux attributs

- **align** - alignement horizontal du texte (left, center, right)
- **valign** - alignement vertical du texte (top, middle, bottom)
- **bgcolor** - couleur de la cellule
- **colspan**, **rowspan** - fusionner colonnes et lignes



## Exemple

```
<table width="400" border="1" cellspacing="5" cellpadding="10">
  <caption>Mon beau tableau</caption>
  <tr>
    <th>Titre 1</th>
    <th>Titre 2</th>
  </tr>
  <tr>
    <td>Texte 1</td>
    <td>Texte 2</td>
  </tr>
  <tr>
    <td colspan="2" align="center">Fusion</td>
  </tr>
</table>
```

<b>Titre 1</b>	<b>Titre 2</b>
Texte 1	Texte 2
Fusion	

## TP7 balises html

# Structure d'un site Web

## L'élément <div> en HTML4 et HTML5

C'est donc en HTML que vous ordonnerez le **contenu d'une page Web** (texte, liens, images, listes...)

C'est ce que nous avons vu précédemment avec l'utilisation des balises <p>, <a>, <img />, <h2>

Le langage HTML va également permettre de créer **la structure visible d'un site Web**, c'est à dire entre les balises <body></body>.

Pour cela, il va falloir définir des "blocs" HTML qui contiendront les différents éléments du site : entête, page, pied de page...

La balise permettant cela en (x)html (d'autres balises existent en HTML5) :

```
<div> </div>
```

Une <div> seule ne représente rien, d'où la nécessité de lui donner un **identifiant** qui permettra de l'isoler afin de lui attribuer un style particulier (cf - module suivant), par exemple :

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

La valeur de l'attribut `id` est libre.

## Exemple

Exemple de structure de site Web :

```
<body>
  <div id="conteneur">
    <div id="entete"></div>
    <div id="menu"></div>
    <div id="page"></div>
    <div id="colonne"></div>
    <div id="pied"></div>
  </div>
</body>
```

*Note : par soucis de lisibilité du code, les autres balises nécessaires (<head>, <html>...) n'apparaissent pas dans cet exemple mais elles sont bien sûr obligatoires.*

