

## Historique des versions

Le HTML 5(octobre 2014) puis HTML 5.1 (octobre 2017) succède à HTML 4.1(décembre 1999), qui lui-même succède à la version 4(avril 1998).

L'objectif du cours n'est pas de faire un comparatif entre les différentes versions.

Le W3C (World Wide Web Consortium) est un organisme international qui développe des standards pour le Web. Actuellement, de nombreux standards sont élaborés par cet organisme : XML, CSS, SVG... et HTML.

Le W3C travaille sur le mandat de ses membres parmi lesquelles se trouvent les grands noms de l'informatique : Google, Mozilla, Microsoft, HP, Cisco ....

En clair : aucune technologie n'obtient le cachet de W3C sans être approuvée par la communauté du Web.

Les membres délèguent des ingénieurs au sein de W3C et participent ainsi à l'élaboration des spécifications techniques pour les technologies du Web.

Le HTML 5 et tous les autres standards sont issus de ce travail.

Ainsi une page WEB écrite en HTML 5, qui respecte les recommandations du W3C sera interprétée de la même façon, quel que soit le navigateur (chrome, IE, Edge, Firefox, Safari, Opera...)

Le W3C propose le service Markup Validation Service à l'adresse : <https://validator.w3.org>. Ce service permet de vérifier que le code HTML d'une page respecte le standard.

Le W3C recommande HTML5 depuis 2014.

## Qu'est-ce que le HTML ?

**HTML** signifie **H**yper **T**ext **M**arkup **L**anguage : Langage de balisage d'hypertexte

Permet de définir des documents Web, reliés entre eux par des liens, contenant :

- Du texte
- Des images
- Du son, des vidéos

C'est un langage de description de documents. Il n'a pas pour objectif la mise en forme des documents. Ce qui permet la mise en forme d'un document c'est le CSS que nous étudierons plus tard dans ce cours.

**Conclusion** : on utilise **HTML** pour décrire le fond, et **CSS** pour la forme.

## La notion de web sémantique

Les pages Web que nous rédigeons sont destinées à être lues par des humains, non par des machines. Néanmoins des ordinateurs spécialisés sont chargés d'analyser les pages Web dans l'objectif de les référencer. C'est le cas des moteurs de recherche comme Google. Pour analyser vos pages, ses derniers doivent comprendre le sens du contenu.

Définition Wikipédia : le Web sémantique est un système qui permet aux machines de « comprendre » et de répondre aux demandes complexes de l'homme en fonction du sens de ces demandes. Une telle « compréhension » exige que les sources d'information pertinentes aient été sémantiquement structurées au préalable.


Autre définition Wikipédia : L'optimisation pour les moteurs de recherche, où SEO (pour search engine optimization) est un ensemble de techniques pour optimiser la visibilité d'une page Web dans les pages de résultats de recherche. Ces techniques cherchent à améliorer la compréhension par les robots d'indexation de la thématique et du contenu d'une ou de l'ensemble

des pages d'un site Web et à augmenter le trafic naturel du site.

L'objectif est d'améliorer le positionnement d'une page Web dans les pages de résultats de recherche sur des mots-clés ciblés (selon les thèmes principaux du site). On considère que le positionnement d'un site est bon lorsqu'il est positionné (classé) dans la première page des résultats de recherche, dans l'une des dix premières réponses naturelles d'une recherche sur des mots-clés correspondant précisément à sa thématique.

Exemple avec les balises de h1 à h6 :

```
<h1> titre 1 </h1>
<h2> titre 2 </h2>
<h3> titre 3 </h3>
<h4> titre 4 </h4>
<h5> titre 5 </h5>
<h6> titre 6 </h6>
```

 Copier le code

Visuellement nous voyons que la balise h1 affiche du texte en "plus gros" que la balise h2. Mais attention le HTML n'est pas fait pour mettre en forme. Ce n'est pas ici le but. Ici l'objectif principal est de définir la sémantique (donner du sens) de vos pages pour les moteurs de recherche.

Avec la balise h1 nous indiquons juste au moteur de recherche que le texte entre balises h1 est un texte très important.

### Ce qu'il n'est pas

HTML n'est pas un langage de programmation. Il ne permet pas de faire des traitements, et n'offre pas les instructions classiques de déclaration, d'affectation de variables, de boucles, de tests SI...ALORS..SINON...etc.

Pour 'programmer' une page Web, il faudra utiliser un langage comme JavaScript, PHP ou java par exemple.

## Généralités sur l'aspect du code

1. En HTML5, les balises ne sont pas sensibles à la casse. L'usage s'est généralisé d'écrire les balises en minuscules. On peut donc écrire les balises ainsi : <BALISE>, <Balise> ou <balise>.
2. Toute balise ouverte <balise> doit être fermée par </balise>, mis à part les balises dites orpheline
3. Les balises doivent être correctement imbriquées. Lorsqu'on affecte plusieurs balises à un élément, l'ordre de fermeture de celles-ci est essentiel. La première balise de fermeture doit correspondre à la dernière balise d'ouverture non fermée.

Exemple :

```
Est correct: <a><b><c>contenu</c></b></a>.
Est incorrect: <a><b><c>contenu</a></c></b>.
```

 Copier le code

4. Les valeurs des attributs doivent toujours figurer entre des "guillemets".

Exemple : ici **lang** est l'attribut, **"fr"** sa valeur.

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

 Copier le code

5. Un code doit être indenté pour être lisible. L'indentation consiste à ajouter les tabulations de façon à marquer qu'une balise HTML dépend de la balise précédente. Sauter des lignes dans le code favorise la lecture de celui-ci.

## Le Doctype

Le **doctype**, littéralement « type de document » est une instruction placée au début des pages HTML.


Cette instruction spécifie le DTD (**D**ocument **T**ype **D**efinition) utilisé. Le DTD définit comment les balises HTML sont formées. Par exemple le DTD définit si une balise peut inclure une autre balise ou non, si oui lesquelles, il définit aussi les attributs utilisables dans une balise, etc. ... Le DTD est indispensable aux validateurs comme celui du W3C, qui en ont besoin pour savoir à quelles règles le document il est supposé obéir. Le validateur est disponible à cette URL : <http://validator.w3.org/>

Le doctype indique au navigateur la version HTML utilisée, ce qui va permettre à ce dernier d'interpréter correctement le document, notamment au niveau du rendu.

## En HTML 5

Les choses ont été grandement simplifiées concernant le doctype. Il faut ajouter la ligne suivante en début de page :


```
<!DOCTYPE html>
```

 Copier le code

Dans les versions précédentes, il fallait ajouter la ligne


- HTML 4.01 strict :

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

 Copier le code

- HTML 4.01 Transitional :

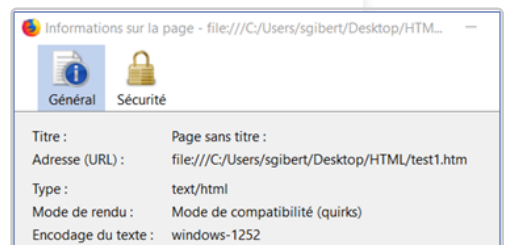
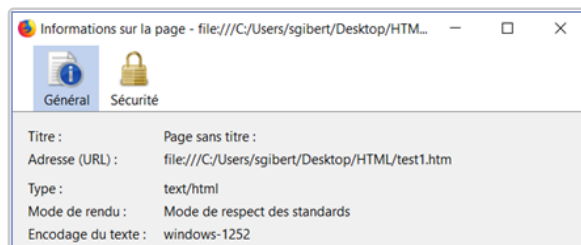
```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

 Copier le code

De quoi s'y perdre un peu.

## Le Doctype ? Est-ce vraiment important ?

Certains navigateurs (Mozilla Firefox par exemple) permettent d'obtenir les informations concernant une page Web. Dans l'écran affiché, une ligne nous intéresse : "le mode de rendu". Dans le cas où le doctype est renseigné, nous aurons la valeur **"Mode de respect des standards"**. Dans le cas contraire, nous aurons la valeur **"Mode de compatibilité (quirks)"**.




En l'absence d'un doctype, le navigateur ignore selon quelles règles il doit traiter la page. Il se rabat ainsi sur des procédures qui lui sont propres pour afficher toute que toute votre page HTML. Ce qui peut entraîner des différences de restitution sensibles entre les différents navigateurs. Ce mode bancal est appelé mode compatibilité ou **quirks mode**.

## La balise <html>

La balise **<html>** indique au navigateur qu'il s'agit d'un document HTML.

La balise **<html>** est l'élément le plus haut ou l'élément racine du document. Elle prend donc place juste après la déclaration

de doctype.

 Copier le code

```
<! DOCTYPE html>
<html lang="fr">
...
</html>
```

La balise `<html>` peut prendre comme attribut `lang="fr"` qui spécifie que le document est en langue française. Cette information est utilisée par les moteurs de recherche comme Google.

À l'adresse [www.w3schools.com](http://www.w3schools.com), vous pouvez trouver une liste complète des autres langues référencées. Ce site doit être un site de référence pour l'apprentissage du html.

## L'élément `<head>` : l'En-tête

La balise `<head>... </head>` qui se positionne juste après le doctype et la balise `<html>` contient ce qu'on appelle l'en-tête du document. Dans cet en-tête du document se situe toute une série d'informations relatives au document lui-même, indépendamment du contenu de celui-ci.

Ces informations peuvent être de natures très diverses :

- Le titre du document. Soit la balise `<title>... </title>` reprise au point suivant.
- Des déclarations ou des appels de feuilles de style CSS.
- Des fonctions ou des liens vers des fichiers JavaScript.
- Des informations à l'intention des moteurs de recherche pour le référencement de la page.
- Des informations à l'intention des navigateurs.
- Des balises méta qui reprendront quant à elles l'encodage, la description de la page, les mots-clés associés à la page, le nom de l'auteur, des mentions de copyright, etc.

Dans un souci de simplification, le HTML5 spécifie des valeurs par défaut pour l'attribut `type` des scripts, des styles et les éléments `link`. Sauf si vous avez expressément besoin d'une valeur différente que celles définies par défaut, vous pouvez omettre en HTML5 cet attribut de type.


Le JavaScript est défini comme langage de script par défaut.

Ainsi, en HTML 4.0, on devait écrire :

 Copier le code

```
<script type="text/javascript"> ... </script>
<script type="text/javascript" src="fichier.js"></script>
```

En HTML5, on peut se contenter de : (javascript est ici le langage par défaut)

 Copier le code

```
<script> ... </script>
<script src="fichier"></script>
```

De même pour le CSS : c'est le langage de feuilles de style par défaut.


Ainsi, en HTML 4.0, on devait écrire :

 Copier le code

```
<style type="text/css"> ... </style>
<link rel="stylesheet" type="text/css" href="fichier.css">
```


En HTML5, on peut se contenter de :

```
<style>...</style>
<link rel="stylesheet" href="fichier.css">
```

 Copier le code

Notre document minimal HTML5 devient :

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    ...
  </head>
  ...
</html>
```

 Copier le code

Mis à part le titre de la page (voir ci-après), aucun élément compris entre les balises <head>... </head> n'apparaît pas dans le navigateur.

## La balise <title>

<https://codepen.io/samuel-gibert/pen/oNGoYwG>

La balise **<title>**, incluse dans l'en-tête du document, est la seule balise obligatoire de celui-ci.

La balise **<title>**... **</title>** donne un titre à votre page.

Ce titre est affiché dans la barre de titre, située en haut à gauche de tout navigateur et/ou dans les onglets ouverts par celui-ci.

La balise **<title>** a une place importante dans l'algorithme de référencement de Google. On veillera à avoir pour les documents Web, un titre accrocheur et synthétique.

Attention s'il y a une erreur dans cette balise par exemple : **<title>**... **<title>** une simple page blanche est affichée. Ici la balise est mal refermée : **<title>**... **</title>**

```
<title> Exemple Titre hx </title>

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

 Copier le code

 Run

## L'encodage des caractères

<https://codepen.io/samuel-gibert/pen/oNGobqx>

Si les caractères accentués ou autres caractères spéciaux de votre page sont remplacés par des carrés, des points

d'interrogation, ou des séries de lettres comme « Ä© » ou « Ä§ », vous avez probablement un problème d'encodage des caractères. Le problème le plus courant vient du fait que le navigateur ne sait pas quel encodage utiliser pour afficher la page, soit parce qu'aucune information ne lui est transmise, soit parce que l'information est erronée.

Exemple : Sans précision d'encodage, ce code sera affiché de la façon suivante :

**Je suis content d'Äatre en BTS SIO**

```
<h1>Je suis content d'Ätre en BTS SIO</h1>
```

 Copier le code

Depuis fin 2011, le WC3, recommande le charset UTF-8 pour l'édition de pages Web en grande partie pour son universalité.

En 2017, 90.4 % des sites utilisent ce jeu de caractères. Pour indiquer au navigateur comment interpréter les caractères :

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

 Copier le code

Pour information, cette déclaration d'encodage prenait en HTML 4.0 la forme :

```
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
```

 Copier le code

## Autres types de balise

Nous verrons plus tard comment intégrer des scripts (javascript), des feuilles de styles CSS, autres balises <meta> seront étudiés plus tard dans le module.

## La balise <body>

Les balises <body>... </body> déterminent ce que l'on appelle le **corps** du document. C'est entre ces balises que prendra place le code HTML5 qui sera utilisé pour élaborer le contenu de la page.

C'est cette partie du document HTML qui sera affiché dans la fenêtre du navigateur et donc visible par l'internaute.

En HTML5, les attributs dédiés à la présentation du corps du document de la balise <body> sont maintenant abandonnés pour laisser place à l'utilisation des feuilles de style.

Ces attributs qui déterminaient la couleur de l'arrière-plan (bgcolor), l'image d'arrière-plan (background) ou la couleur du texte (text) n'ont plus leur place en HTML5 (obsolète), même si cela reste toujours possible.

Au final, notre document HTML5 minimal se présente ainsi :

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <title>Html5</title>
    <meta charset="UTF-8">
  </head>
  <body>
  </body>
</html>
```

 Copier le code

## Les commentaires

Il est recommandé de commenter son code HTML (comme pour tout code de programmation), pour en faciliter la compréhension lors d'une mise à jour.


Les commentaires doivent être précédés de la balise `<!--` et être fermés par la balise `-->`.

Tout ce qui sera écrit entre ces balises ne sera pas affiché à l'écran par le navigateur. Notons cependant que ces commentaires restent visibles lorsqu'on consulte le code source de la page.

Ajoutons que les commentaires peuvent occuper plusieurs lignes et peuvent prendre place à n'importe quel endroit de la page.

Exemple :

```
<h1>Je suis content d'être en BTS SIO</h1>
<!-- commentaire
    particulier sur plusieurs
    lignes-->
```

 Copier le code

 Run

## Les balises Conteneurs textes

Lorsque vous allez incorporer du texte dans vos pages Web, vous devrez placer celui-ci dans des conteneurs. Même s'il est possible de ne pas le faire, il vaut mieux éviter cette pratique. Dans ce cas-là, il ne sera pas possible de mettre en forme le texte à l'aide du CSS.

Vous avez à votre disposition tous les conteneurs nécessaires : pour les titres, pour le texte courant, pour créer des listes...

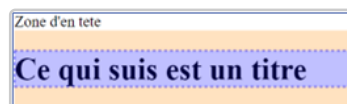
### Les Balises titre <H1> à <H6>

<https://codepen.io/samuel-gibert/pen/oNGoYwG>

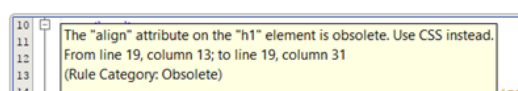
Le langage HTML propose (depuis le début de HTML 1 jusqu'à aujourd'hui) 6 grandeurs de caractères pour les titres. Quand on sait que titre se dit en anglais "heading", l'élaboration de la balise de titre devient évidente ; `<h>` pour heading suivi du numéro 1 (pour le plus grand), 6 (pour le plus petit).

Quelques remarques :

- Par défaut, avec la balise `<h>... </h>`, le texte est mis en gras, avec une certaine police de caractère, une taille. La raison première de cette balise n'est pas la mise en forme, mais la structuration de la page. Tout ce qui concerne la mise en forme doit être fait en CSS.
- La balise `<h>` est un élément de bloc, et ajoute une ligne vide avant et après le bloc comme illustré. Ces lignes vides s'appellent des marges (margin) qui permettent de définir l'espacement entre les différents éléments HTML.

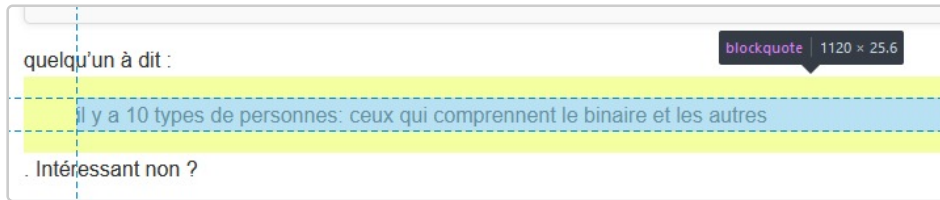


- Comme déjà évoqué ces balises sont très importantes pour le référencement de vos pages. Il est donc important que ces balises contiennent des mots-clés pertinents.
- L'attribut `align`, disponible dans le HTML 4.0, n'est pas repris dans le HTML 5. C'est le cas pour toutes les balises du type conteneur de texte. L'alignement du paragraphe se réalise ainsi par une propriété de style CSS. Attention comme déjà indiqué même si l'usage de certaines balises deviennent obsolète il reste toujours interprété par les navigateurs. Nous verrons plus loin dans ce cours l'usage des différents attributs les plus communs.









Comme illustré elle s'affiche également avec un léger retrait par rapport à la marge gauche et un retrait en marge droite. C'est une balise du type élément bloc.

La balise <q> permet également une citation. Elle ajoute automatiquement des guillemets à la citation. C'est une balise du type en ligne. Il est donc possible de l'incorporer dans une balise <p>

```
<p> quelqu'un à  
dit : <q>il y a 10 types de personnes: ceux qui comprennent le  
binaire et les autres</q>. Intéressant non ?</p>
```

 Copier le code

 Run

Quelqu'un à dit : «il y a 10 types de personnes: ceux qui comprennent le binaire et les autres». Intéressant non ?

La balise <cite> fait à peu près la même chose que <q>. C'est une balise de type en ligne. Elle formate la citation en italique.

Exemple :

```
<p> quelqu'un a dit : <cite>il y a 10 types de  
personnes: ceux qui comprennent le binaire et les autres</cite>.  
Intéressant non ?</p>
```

 Copier le code

 Run

Quelqu'un à dit : *il y a 10 types de personnes: ceux qui comprennent le binaire et les autres.* Intéressant non ?

## Les listes : <OL> <UL> <LI> <DL> <DT> <DD>

Les listes, bien connues des utilisateurs de traitement de texte, sont indispensables pour structurer votre texte.

### Les listes ordonnées

Une liste ordonnée est mise en place par la balise <ol>... </ol>. On détermine ensuite, à l'intérieur de ces balises, les éléments ou les items de la liste, soit les balises <li>... </li>. La structure générale d'une liste ordonnée ou numérotée est donc :

```
<ol>  
  <li>Chapitre 1</li>  
  <li>Chapitre 2</li>  
  <li>Chapitre 3</li>  
  <li>Chapitre 4</li>  
</ol>
```


 Copier le code

 Run

- L'attribut **start** permet de commencer la numérotation à un autre chiffre que 1.
- Le nouvel attribut HTML5 **reversed** permet une numérotation descendante (5, 4, 3, 2, 1) au lieu de la numérotation ascendante habituelle (1, 2, 3, 4, 5). À ce jour, ce nouvel attribut est repris par les versions récentes de Firefox, Chrome et Opera.
- Bien que déprécié, l'attribut **type** permet de définir le type de numérotation : type="1|a|A|i|I"

### Exemple :

```
<ol start="4" type="I" reversed>
  <li>Chapitre 1</li>
  <li>Chapitre 2</li>
  <li>Chapitre 3</li>
  <li>Chapitre 4</li>
</ol>
```

 Copier le code

 Run


IV. Chapitre 1  
III. Chapitre 2  
II. Chapitre 3  
I. Chapitre 4

- L'attribut **value** permet de modifier la valeur pour un item : `<li value="8">Chapitre 4</li>`

### Les listes non ordonnées

Les listes non ordonnées, aussi appelées listes à puces, sont introduites par la balise `<ul>... </ul>`. On détermine ensuite, à l'intérieur de cette balise, les éléments ou les items de celle-ci, soit les balises `<li>... </li>`. La structure générale d'une liste à puces est donc :

```
<ul>
  <li>Chapitre 1</li>
  <li>Chapitre 2</li>
  <li>Chapitre 3</li>
  <li>Chapitre 4</li>
</ul>
```


 Copier le code

 Run

- L'attribut **type** permet de définir la forme de la puce : "disc, square, circle".

### Combines les types de listes

```
<ol>
  <li>Les titres</li>
  <li>Les listes en Html5
    <ul>
      <li>Listes ordonnées</li>
      <li>Listes non ordonnées</li>
      <li>Listes imbriquées</li>
      <li>Listes de définitions</li>
      <li>Le menu de listes</li>
    </ul>
  </li>
  <li>Les tableaux en Html5</li>
  <li>Les formulaires en Html5</li>
</ol>
```

 Copier le code

 Run

1. Les titres  
2. Les listes en Html5

- Listes ordonnées
- Listes non ordonnées
- Listes imbriquées
- Listes de définitions

### Les listes de définition

Cette liste, dite de définition, présente une liste de termes, chacun d'entre eux étant suivi de sa description. Elle utilise la balise `<dl>...</dl>`. Entre ces balises, on reprend ce qu'on a appelé le terme par la balise `<dt>... </dt>` suivi par la définition dans la balise `<dd>... </dd>`. Exemple :

ol	Liste ordonnée
ul	Liste non ordonnée
dl	Liste de définitions

```
</dl>
  <dt>ol</dt>
  <dd>Liste ordonnée</dd>
  <dt>ul</dt>
  <dd>Liste non ordonnée</dd>
  <dt>dl</dt>
  <dd>Liste de définitions</dd>
</dl>
```



Le texte préformaté : <PRE>

La balise `<pre>... </pre>` permet d'afficher le texte tel qu'il a été encodé dans l'éditeur de texte. Les espaces, les tabulations et les retours charriot sont respectés à l'écran. Si elle comporte du texte, celui-ci sera affiché avec une police à pas fixe (tous les caractères ont exactement la même largeur). C'est le cas des polices de caractères comme Courier New, Consolas, qui sont utilisés par les IDE.

```
<pre>
Le texte inclus dans une balise pre
est affiché dans une police à pas fixe
et préserve                les espaces
Ainsi que les passages à la ligne.
</pre>
```



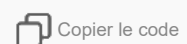
```
Le texte inclus dans une balise pre
est affiché dans une police à pas fixe
et préserve          les espaces
Ainsi que les passages à la ligne.
```

Ajout d'une ligne séparatrice : `<Hr>`

Le trait horizontal peut se révéler fort utile pour définir un changement dans le contenu. La balise `<hr>` réalise cette fonction. Comme pour la balise `br` celle-ci n'a pas de balise fermante. On pourra donc écrire `<hr>`. Avec le CSS il est alors possible d'en modifier l'aspect.



Chapitre1



```
<hr>
<p>Chapitre 2</p>
```



Chapitre 1

Chapitre 2

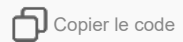
## La mise en forme du texte

Une fois que vous avez placé vos textes dans les conteneurs texte (par exemple une balise `<p>`) on peut appliquer des éléments HTML de mise en forme qui sont tout aussi sémantiques. C'est-à-dire que la mise en forme donne du sens au contenu concerné, ce n'est pas simplement une mise en forme graphique. Il ne s'agit que de quelques formatages de texte des plus basiques comme mettre du texte en gras, en italique, en exposant ou en indice. La plupart de ses balises sont de type en ligne

### Mettre en gras : `<B>` `<STRONG>`

B pour Bold.

```
<p>Texte: <b> Du texte en gras </b></p>
<p>Texte: <strong> Du texte en gras </strong></p>
```



Visuellement pas de différence, mais l'interprétation en est fort différente dans les synthèses vocales qui lisent les pages Web à l'intention des personnes visuellement déficientes. La balise `<strong>` renforcera le contenu par une intonation de voix différente ou un niveau sonore plus élevé

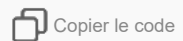
Texte: **Du texte en gras**

Texte: **Du texte en gras**

### La mise en italique `<i>` `<em>`

Exemple: il est possible de combiner texte en gras et en italique.

```
<p><i>Du texte en italique seul</i></p>
<p><b><i>Du texte en italique et en gras</i></b></p>
```



Il existe aussi la balise `<em>` (emphase) qui met également le texte en italique. La balise `<em>` a pour but de mettre en avant, de mettre en exergue le texte qu'elle contient de façon sémantique.

### La mise en indice et en exposant : `<SUP>` `<SUB>`

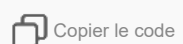
Il est parfois nécessaire de mettre des caractères en exposant ou en indice dans, par exemple, des formules mathématiques ou chimiques. La balise `<sup>...</sup>` mettra le contenu en exposant et la balise `<sub>...</sub>` en indice.

$$(a + b)^2 = a^2 + 2ab + b^2$$

H<sub>2</sub>O

Exemple :

```
<p>(a + b)<sup>2</sup> = a<sup>2</sup> + 2ab + b<sup>2</sup>
```



<br> H<sub>2</sub></p>



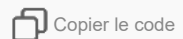
## Le souligné <u> et <INS>, le barré <S> et <del>

L'élément <u> permet de faire ressortir une portion de texte sans importance sémantique. Le fait de souligner un texte peut engendrer une confusion avec les liens hypertextes, qui sont aussi soulignés par défaut. De ce fait, le W3C suggère d'en faire un usage judicieux. <ins> a le même effet, mais indique une sémantique de contenu nouveau dans la page.

<s> permet de barrer un texte sans importance sémantique. <del> a le même effet, mais indique une sémantique de contenu supprimé dans la page.

Exemple :

De ce fait, le W3C suggère d'en faire un usage ~~systematique~~ judicieux.



```
<p>de ce fait, le W3C suggère d'en faire un usage <s>systematique</s><u>judicieux</u>.</p>
<p>de ce fait, le W3C suggère d'en faire un usage <del>systematique</del> <ins>judicieux</ins>.</p>
```



## Réduire la taille d'un texte<small>

La balise <small> permet d'écrire avec une taille de caractère plus petite que le contexte courant donnant une sémantique d'un texte de moindre importance. L'usage possible est fait pour les clauses des contrats, les mentions spécifiques et légales.

Exemple :

Voici le contrat: c'est pas parce que c'est écrit petit que je ne dois pas être lu...



```
<p>voici le contrat: <small> ce n'est pas parce que c'est écrit petit que je ne dois pas être lu... </small></p>
```



## Sens de lecture :<BDO>

Certains systèmes d'écriture, tels que l'alphabet arabe et hébreu, s'écrivent de droite à gauche. Le HTML, langage universel, se devait de tenir compte de cette spécificité. L'attribut dir permet de définir le sens de lecture : "ltr" (*left to right*) , dir="rtl" (*right to left*). Cet attribut est également utilisable au niveau des balises <p>, mais aussi au niveau <html>.

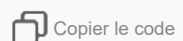
Placé dans une balise <bdo> cet attribut inverse l'ordre d'apparition des lettres.

Exemple :

L'attribut dir permet de définir le sens de lecture

erutcel ed snes el riniféd ed temrep rid tubirtta'L

erutcel ed snes el riniféd ed temrep rid tubirtta'L



```
<p dir="rtl">L'attribut dir permet de définir le sens de lecture</p>
<p><bdo dir="rtl" >L'attribut dir permet de définir le sens de lecture</bdo></p>
<p dir="rtl"><bdo dir="rtl" >L'attribut dir permet de définir le sens de lecture</bdo></p>
```



## Autres balises

## [Test dans codePen](#)

**<dfn>** : Permet de mettre en évidence un terme défini dans le contexte courant. Il n'y a pas de mise en forme par défaut, chaque navigateur est libre. Mais la plupart du temps, c'est de l'italique qui est appliqué.

**<abbr>** : permet de mettre en évidence une abréviation. Vous pouvez utiliser l'attribut title pour indiquer le libellé complet de l'abréviation.

**<code>** : permet de mettre en évidence un code informatique. La mise en forme se fait généralement avec une police de caractères à largeur fixe de type Courier.

**<var>** : permet de mettre en évidence une variable mathématique ou informatique. La mise en forme se fait avec de l'italique.

**<samp>** : permet de mettre en évidence un exemple ou un échantillon (*sample* en anglais). La mise en forme se fait généralement avec une police de caractères à largeur fixe de type Courier.

**<kbd>** : permet de mettre en évidence un texte à saisir au clavier, une touche ou une combinaison de touches. La mise en forme se fait généralement avec une police de caractères à largeur fixe de type Courier.

**<time>** : permet de mettre en évidence les indications de date et d'heure.

**<mark>** : permet de mettre en évidence une portion de texte importante, avec une mise en forme surlignée en fond jaune.

**<meter>** : permet d'insérer une jauge affichant un état numérique défini sur une échelle spécifiée.

**<progress>** : permet d'insérer une barre de progression, affichant un pourcentage de réalisation. Nous verrons comment faire évoluer cette barre par la suite.

**<summary>/<details>** : permettent d'afficher les détails (avec l'élément <details>), masqués par défaut, d'un élément affiché dans <summary>.

Le tout en image :

