

INITIATION HTML - CSS

CRÉER UNE PAGE HTML

CRÉER UNE PAGE HTML

1 - Création d'une page HTML 5

1. Créez un dossier sur votre ordinateur et nommez le “Mon premier site” par exemple.
2. Dans ce dossier nous y mettrons les pages HTML, les images ainsi que le CSS.
3. Ouvrez votre éditeur de texte et créez un nouveau fichier et enregistrez le sous : index.html dans votre dossier nouvellement créé.

2 - Présentation

HTML 5 est le successeur de HTML 4.

Le travail sur HTML 5 a commencé fin 2003 grâce au WhatWG.

Ce n'est qu'en 2007 que le W3C accepte la vision HTML5 en intégrant en son sein le WhatWG.

Les principes de conception sont de simplifier l'usage du HTML, officialiser les pratiques courantes, et assurer un maximum de rétro-compatibilité avec les anciens navigateurs.

HTML5 offre de nouvelles avancées :

- une grammaire simplifiée ;
- de nouveaux éléments sémantiques et de nouveaux attributs ;
- de la vidéo et du son sans plug-ins ;
- plus d'options sur les formulaires et de sécurisation ;
- la localisation ;
- la création de dessins à l'aide de l'élément `<canvas>` ;
- etc.

3 - Syntaxe

Le HTML5 est donc l'héritier du HTML4 ce qui le rend permissif.

Les majuscules sont autorisées pour les noms d'éléments, les attributs, les apostrophes simples ou doubles sont non obligatoires.

Les éléments `<p>`, `<dd>`, `<dt>`, `<li>`, `<optgroup>`, `<option>`, `<td>`, `<th>`, `<tr>`, `<thead>` et `<tfoot>` ne nécessitent pas de balise fermante pour être valides. Seule la version XHTML 5 oblige à fermer ces éléments.

Les éléments suivants ne nécessitent pas de balises fermante et ouvrantes : `<html>`, `<head>`, `<body>`, `<thead>`, `<tfoot>` et `<tbody>`. Ils deviennent donc implicite.

Mais attention conservez une syntaxe XHTML propre au seins de vos documents. Cela le rend maintenable.

4 - Doctype

Le **doctype** définit donc la version du HTML qui doit être lu par le navigateur pour être en conformité.

La doctype ici présent nous indique que cette page est codée en HTML 5.

```
<!DOCTYPE html>
```

5 - Balise html

La balise `<html>` est une balise ouvrante et fermante. Elle pourrait s'écrire simplement `<html>`, mais il est préférable d'y apposer l'attribut langue.

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

6 - Balise head

La balise `<head>` est une balise ouvrante et fermante.

La balise `<head>` est un élément structurel pour la page html (au même titre que les balises `<html>` ou `<body>`).

Elle contient les informations essentielles à la description du document qui sera affiché dans le corps de la page.

Tout ce qui est inclu dans la balise `<head>` est non visible pour l'internaute dans la fenêtre du navigateur (mis à part le `<title>`).

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

7 - Balises metas

Les balises `<meta>` servent à spécifier l'encodage des caractères le titre du document, les références aux feuilles de styles utilisées dans le document, les références aux fichiers javascripts, etc.

Elles se trouvent toutes entre ces deux balises et ne sont pas affichées sur la page du site. Ces informations sont utiles pour les navigateurs et les moteurs de recherche, bien évidemment si les moteurs de recherche les exploitent toutes. Nous listerons donc les incontournables.

7.1 - Encodage des caractères

La première ligne du `<head>` est l'encodage des caractères :

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

C'est une balise auto fermante et elle indique le type de codage utilisé, ici utf-8. Il faut donc paramétrer votre éditeur de texte en utf-8 et il vous sera alors possible de taper les accents, les ç, les symboles \$, €, sans avoir à les coder avec des entités HTML, comme € pour le caractère €.

Les principaux encodages sont les suivants :

- utf-8 (charset=utf-8) : codage universel Unicode (sur un ou deux octets), à privilégier car c'est la solution d'avenir.
- iso-8859-1 ou Latin-1 (charset=iso-8859-1) : codage occidental classique, souvent utilisé.
- iso-8859-15 ou Latin-9 (charset=iso-8859-15) : codage occidental iso-8859-1 plus quelques caractères, dont €, OE, oe, et Ÿ.
- Windows-1252 ou ANSI (charset=windows-1252) : codage provenant du iso-8859-1, comprenant également €, OE, oe, et Ÿ.

7.2 - Balise meta description

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <meta name="description" content="Description de la page" />
</head>
</html>
```

Cette balise `<meta>` est auto-fermante et elle fournit une description courte de la page web. 156 caractères maximum.

Cette description est utilisée par les principaux moteurs de recherche pour fournir le court texte d’aperçu des pages de résultats.

On conseille de ne pas utiliser la même description pour toutes les pages de votre site. Dans l’idéal, chaque page aura sa propre description. Voici quelques astuces pour la rédaction de la description :

- elle doit être unique, et pas dupliquée sur d’autres pages du site ;
- elle doit refléter fidèlement le contenu de la page ;
- elle doit être attractive pour l’utilisateur dont le regard «scanne» une page de résultats de moteur de recherche ;
- elle doit reprendre les principaux mots-clé de la page et être plutôt brève (on recommande de ne pas dépasser 200 caractères, bien qu’il ne s’agisse pas d’une règle stricte).

Ressource détaillée sur les meta :

[Alsacrédation – À quoi servent les balises META ?](#)

7.3 - Balise title

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <title>Titre du site</title>
  <meta name="description" content="Description de la page" />
</head>
</html>
```

La balise `<title>` est ouvrante et fermante, Elle permet de d'écrire le titre de la page. Ce dernier apparaîtra dans la barre de titre du navigateur.

Les moteurs de recherche et divers outils, se servent du `<title>` pour afficher le titre de la page dans les résultats de recherche.

Elle est obligatoire en HTML.

Nous parlons pas de balise `<title>` car c'est un cas particulier mais bien d'un élément html.

Elle sert également au référencement. Un élément `<title>` bien rédigé doit :

- être lisible, informatif et compréhensible par un humain,
- refléter le contenu de la page,
- contenir des mots-clé,
- faire moins de 70 caractères (pas une règle absolue, mais Google coupe les titres trop longs entre 65 et 70 caractères dans les pages de résultat, et les titres trop longs diluent le poids des différents mots-clé).

8 - Balise body

La balise `<body>` est ouvrante et fermante.

Elle spécifie le contenu en tant que tel de la page qui sera donc affiché à l'écran.

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <title>Titre du site</title>
  <meta name="description" content="Description de la page" />
</head>
<body>

</body>
</html>
```

9 - Commentaires

Les commentaires dans le code HTML peuvent se révéler très utiles afin de pouvoir donner une information sur les éléments de structuration de la page et pour vous y retrouver au fur et mesure que vous développez vos pages.

Cependant attention aux commentaires mal écrits qui saboteront vos pages tout simplement. Ceci aura pour effet de bloquer l’affichage de votre page.

Le commentaire ci-dessous est correct donc valide :

```
<!-- commentaire HTML valide -->
```

Ce commentaire n’est pas valide :

```
<!-- commentaire HTML -- invalide -->
```


Exemple :

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <!-- Encodage des caractères -->
  <meta charset="UTF-8">

  <!-- Titre du document -->
  <title>Titre du site</title>

  <!-- Metas informations -->
  <meta name="description" content="Description de la page" />
</head>
<body>

</body>
</html>
```

10 - Espaces et sauts de lignes

Dans votre fichier html, n'hésitez pas à sauter des lignes pour aérer le code, et indentez votre code.

Quel que soit le nombre d'espaces se succédant, le navigateur saura n'en afficher qu'un seul. Quant aux sauts de ligne, il les ignorera.

Pour créer un saut de ligne à afficher dans le navigateur, il vous faut utiliser la balise `<br />` (mais l'utiliser n'est pas recommandé, vous ajoutez alors du code pour rien, alors que vous pourrez dans les 3/4 des cas résoudre le problème en CSS).

Pour forcer l'affichage de plusieurs espaces successifs sur une ligne, il faut utiliser le caractère spécial ` ` pour espace insécable en français.

L'indentation quand à elle, permet de comprendre les niveaux d'imbrication de chaque éléments html et de permettre d'en comprendre plus facilement la structure hiérarchique.

Vous pouvez par la suite à la fin de vos développements utiliser des outils en ligne qui vous compacteront votre code pour en assurer un chargement plus optimal et donc gagner en performance web.