

ressource

Mémo HTML CSS

1 HTML

HTML (HyperText Markup Language) est le langage utilisé pour décrire le contenu d'une page web. Il est utilisé dans un fichier nommé avec l'extension « .html ».

1.1 Contenu minimale d'une page html

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta charset="utf-8" />
    <title>Texte de l'onglet </title>
  </head>
  <body>
    Texte de la page
  </body>
</html>
```

1.2 Valider le contenu d'une page

<https://validator.w3.org/>

1.3 Structurer le contenu

1.3.1 Ajout d'un commentaire

```
<!-- Ceci est un commentaire -->
```

1.3.2 Organiser la page

Utiliser de préférence les balises définissant les différentes zones standards de la page :

- l'entête dont le contenu est défini entre des balises `<header>`
- le contenu principal (spécifique à cette page, ne peut être incluse dans un autre élément) : `<main>`
- le pied de page : `<footer>`
- une barre (souvent latérale) pouvant donner des informations complémentaires sur le contenu : `<aside>`
- un menu de navigation : `<nav>`
- une zone de contenu indépendante du reste de la page : `<article>`
- une section : `<section>`
- une zone de bloc quelconque : `<div>`

1.3.3 Structurer le texte

- `<p>` pour définir un paragraphe `</p>`
- `
` : pour revenir à la ligne dans un paragraphe
- `<h1>` un titre principal `</h1>`
 - `<h2>` un sous titre `</h2>`
 - ... jusqu'à ...
 - `<h6>` un sous sous sous `</h6>`
- `` déclaration d'une « unordered list »
 - `` élément de la liste``
 - `` autre élément de la liste````
- `` déclaration d'une « ordered list »
 - `` élément de la liste``
 - `` autre élément de la liste````

1.3.4 Mettre en valeur des éléments

Des balises permettent d'indiquer au navigateur que certains mots ont une importance particulière.

`` marque la particularité d'un texte, une citation, un titre de livre : en italique par défaut ``

`` pour mettre en avant un texte, le faire ressortir : en gras par défaut ``

`<mark>` pour attirer l'attention sur un texte : surligne par défaut `</mark>`

1.4 Ajouter du contenu

1.4.1 Insérer une image

Pour insérer une image, on utilise la balise auto-fermante ``.

``

- L'attribut `src` (source), obligatoire, indique l'image à insérer. Le chemin doit être de préférence un chemin relatif (par rapport au dossier où se trouve le fichier html).
- L'attribut `alt` (alternatif), obligatoire, est le texte qui sera affiché si l'image ne peut être chargée ou qui sera lu si une personne malvoyante utilise un dispositif de lecture de site.
- L'attribut `title`, facultatif, contient le texte qui s'affiche en infobulle lorsque le curseur de la souris est au-dessus de l'image.

Pour les images à insérer, on choisira préférentiellement d'utiliser le format :

- jpeg, s'il s'agit de photo
- png, s'il s'agit d'autre chose (dessin, schéma, ...)



Pour les noms de fichiers html, de dossiers, d'images ..., il est :

- impératif de n'utiliser que les lettres minuscules et majuscules non accentuées, les tirets et les underscores
- conseillé de n'utiliser que les minuscules et les tirets

1.4.2 Insérer des liens

Les liens sont au cœur du web, ils permettent de se déplacer de pages en pages à l'intérieur d'un site ou entre différents sites. Ils utilisent les balises :

` texte du lien `

Un attribut facultatif supplémentaire `target="_blank"` permet de forcer l'ouverture du lien dans un nouvel onglet ou une nouvelle fenêtre (selon les réglages du navigateur).

1.4.3 Insérer une ancre

Une ancre est un lien vers n'importe quel point du site : un titre, une image, un paragraphe ...

Il faut tout d'abord utiliser l'attribut `id` pour identifier ce point (unique sur la page),

Exemple : `<h1 id="nom-ancree">`

Puis créer un lien vers cette ancre en utilisant le symbole `#` devant le nom de l'ancree

`` ou ``

2 CSS

CSS (Cascading Style Sheet) est le langage utilisé pour mettre en forme (esthétisme et positionnement) le contenu défini dans un fichier html.

2.1 Créer et lier un fichier CSS à un fichier HTML.

On met le CSS dans un fichier séparé nommé par exemple « styles.css », que l'on va lier au fichier html en ajoutant dans sa balise <head>

```
<link rel="stylesheet" href="styles.css">
```

Le fichier CSS va lui-même être découpé en 2 parties (on peut même faire 2 fichiers distincts) :

- une partie « chartre graphique » dans laquelle on va définir les tailles et polices d'écriture, les couleurs, appliquer différents effets, ...
- une partie « structure » dans laquelle on gère le positionnement des blocs.

2.2 Syntaxe du CSS

Le principe de base du css est de définir des règles dans lesquelles **un sélecteur** indique les éléments de la page pour lesquels on affecte **des valeurs** à **des propriétés**.

La syntaxe d'une règle est :

```
sélecteur {  
    propriété1: valeur1;  
    propriété2: valeur2;  
    propriété3: valeur3;  
}
```

2.2.1 Les sélecteurs

Les sélecteurs peuvent être tout type de balise : p, h1, body, ...

Il est possible de combiner les sélecteurs pour cibler certains éléments :

- **h3, em** : sélectionne tous les éléments <h3> et tous les éléments
- **h3 em** : sélectionne tous les éléments situés à l'intérieur d'un élément <h3>
- **h3 + p** : sélectionne le premier élément <p> situé après un titre <h3>.
- **a[title]** : sélectionne tous les liens <a> qui possèdent un attribut title
- **a[title="Cliquez ici"]** : sélectionne les liens <a> qui possèdent un attribut title valant "cliquez ici"
- **a[title*="ici"]** : sélectionne les liens <a> qui possèdent un attribut title qui contient "ici"

On peut créer ses propres sélecteurs :

- Dans HTML, identifier un bloc avec l'un des 2 attributs ci-dessous :
 - **<p class="un-nom-qui-peut-être-commun-a-plusieurs-éléments">**
 - **<p id="un-nom-unique-sur-la-page">**
- Dans CSS, utiliser le sélecteur approprié :
 - **.nom-de-class**
 - **#nom-de-id**

2.2.2 Les commentaires

```
/* des commentaires en css */
```

2.3 Esthétique

2.3.1 Modification des couleurs

On utilise la propriété `color` avec les valeurs possibles ci-dessous :

- 16 couleurs nommées standards présent en charge par tous les navigateurs (white, silver, gray, black, red, maroon, lime, green, yellow, olive, blue, navy, fuchsia, purple, aqua, teal)
- 140 couleurs nommées présent en charge par les navigateurs récents



<https://htmlcolorcodes.com/fr/noms-de-couleur/>

- code RGB de la couleur, exemple : `rgb(12, 200, 0)`
- code hexadécimal de la couleur, exemple : `#45A1BB`

Exemple :

```
p {  
  color: blue;  
}
```

2.3.2 Taille d'écriture

La propriété `font-size` permet de choisir la taille du texte.

Elle est utilisable de plusieurs façons :

- en valeur absolue, en précisant le nombre de pixels de haut → `font-size: 12px`
- en valeur relative, en donnant comme valeur :
 - `xx-small, x-small, small, medium, large, x-large, xx-large`
 - `1em, 0.85em, 1.3em` (relatif à la taille de l'élément parent)
 - `80%, 110 %`, ...

Remarque : afin d'améliorer l'adaptabilité de votre site à la résolution et aux paramètres des utilisateurs, il est conseillé d'utiliser de préférence les tailles en valeurs relatives.

2.3.3 Effets d'écriture

```
font-style : italic ;  
font-style : oblique;  
font-style : normal;
```

```
font-weight : bold ;  
font-weight : normal
```

```
text-decoration: underline;  
text-decoration: line-through;  
text-decoration: overline;  
text-decoration: blink;  
text-decoration: none;
```

```
text-align: left;  
text-align: right;  
text-align: center;  
text-align: justify;
```

2.3.4 Police d'écriture

Pour choisir la police d'écriture, on utilise la propriété `font-family` en précisant par ordre de préférence les polices standards que le navigateur essaiera d'appliquer (il est conseillé d'en mettre systématiquement plusieurs, au cas où ...).

Exemple :

```
body {  
    font-family: Calibri, Verdana, Arial, sans-serif;  
}
```

Si les polices standards ne conviennent pas, on peut faire télécharger au navigateur une police mise à disposition par Google sur <https://fonts.google.com> (choisir une police en cliquant sur son bouton +, puis suivre la procédure en cliquant sur le bandeau « families selected »).

2.3.5 Fond d'écran, fond de bloc, fond ...

La propriété `background-color` permet d'appliquer une couleur au fond de la page, ou au fond d'un bloc paragraphe, ou d'un bloc titre, ou d'un bloc ...

Il est également possible d'appliquer une image en fond du bloc que l'on souhaite.

```
background-image: url("chemin-image.jpg");
```

Puis y ajouter les propriétés suivantes si besoin :

```
background-attachment: fixed; (ou scroll pour que l'image défile avec le texte)  
background-repeat: repeat; (ou repeat-x, repeat-y, no-repeat)  
background-position: 30px 50px;
```

2.4 Structuration de la page

2.4.1 Format des blocs

Parmi les balises vues jusqu'à présent, il existe :

- des balises bloc : elles définissent une zone sur la page et provoque un retour à la ligne (paragraphe, titre, section,)
- des balises inline, qui sont incluses dans une balise bloc, et qui ne provoque pas de retour à la ligne (strong, lien, ...).

Il existe des propriétés permettant d'imposer les dimensions des balises de type bloc, soit de façon fixe, soit relativement aux dimensions de l'écran :

```
height : 20 px;  
width : 50 %;
```

On peut spécifier des dimensions minimales et maximales pour la largeur et la hauteur :

```
min-width : 500px  
max-height : 300px
```

Les blocs ont une bordure ainsi que des marges intérieurs (padding) et extérieurs (margin) que l'on définit en pixel :

```
p {  
    border: 2px solid black ; (ou dotted, dashed, double, groove, ridge, inset, outset, none)  
    padding: 5px;  
    padding-top : 10px;
```

```
margin: 10px;  
margin-left : 30px;  
}
```

Pour centrer un bloc sur la page, celui-ci doit avoir une largeur indiquée, et on spécifie `margin : auto`.

2.4.2 Mise en place des blocs

La mise en place des blocs sur la page s'effectue de façon moderne avec les techniques Flexbox ou Grid.

Flexbox est à privilégier quand on veut placer des blocs en ligne les uns par rapport aux autres.

Grid est efficace dans des structures plus complexes avec un positionnement en 2 dimensions des blocs.

Le principe consiste en la définition d'un conteneur dans lequel sont placés des blocs.

Les différentes balises `body`, `header`, `section`, `main` ... peuvent être le conteneur d'autres balises bloc.

Il est également possible d'utiliser la balise bloc `<div>` qui crée un bloc mais sans signification particulière.

Le bloc servant de conteneur doit être défini comme flexbox :

```
display: flex ;
```



La propriété `display`, comme toutes les autres, s'applique en fonction d'un sélecteur. N'oubliez pas que si besoin vous pouvez créer des sélecteurs ciblant précisément certains éléments en utilisant des attributs `class` ou `id`. Cela permet par exemple de définir une certaine section comme étant un conteneur, sans que toutes les sections soient des conteneurs.

- Choix de la direction et l'ordre des blocs (à mettre pour le bloc conteneur)
 - `flex-direction: column;`
 - `flex-direction: row-reverse;`
 - `flex-direction: column-reverse;`

- Choix de l'alignement des blocs (à mettre pour le bloc conteneur)

Que les blocs soient en ligne ou en colonne, on va placer les éléments selon l'axe principal.

- `justify-content : flex-start;`
- `justify-content : flex-end;`
- `justify-content : center;`
- `justify-content : space-between;`
- `justify-content : space-around;`

On peut les placer également suivant l'axe secondaire, perpendiculaire à l'axe principal des blocs.

- `align-items: stretch;`
- `align-items: flex-start;`
- `align-items: flex-end;`
- `align-items: center;`

- Grossissement des blocs (à mettre pour le bloc contenu)

La propriété `flex` permet de faire grossir un élément afin qu'il occupe tout l'espace disponible.

Par exemple si un bloc a la propriété `flex: 1`; il prend tout l'espace disponible.

Si un bloc est `flex: 1` et qu'un autre est `flex: 2`, alors le second peut grossir 2 fois plus que le premier.