

FORMATION

HTML 5, CSS 3, Responsive

25/09/2023

Votre formateur : Quentin BIDOLET

0.1 - Présentations



Quentin BIDOLET

5 ans d'expérience en développement web

Un profil fullstack, je fais aussi bien du back que du front

Développement représente 70 % de mon activité

La formation représente 30 % de mon activité



Participants

Avoir des connaissances de base HTML / CSS.

Quelles sont vos attentes par rapport à cette formation ?

Avez-vous déjà eu une expérience professionnelle dans l'informatique ?

0.2 - Slack



Accès à **Slack**

Merci de bien vouloir me rejoindre sur Slack via le lien que je vais vous fournir.

0.3 - Déroulement de la formation



Horaires, temps de pause et **dossier pédagogique**

De 9h à 17h30.

Une pause de 15mn le matin et une pause de 15mn l'après midi.

Midi, une pause déjeuner de 1h15.

1 - Initiation à Visual Studio Code

Raccourcis

- Ouvrir fermer la fenêtre latérale
Mac – Command + B
Windows – Ctrl + B
Ubuntu – Ctrl + B
 - Sélectionner un mot
Mac – Command + D
Windows – Ctrl + D
Ubuntu – Ctrl + D
 - Sélectionner plusieurs lignes
Mac – Shift + Option + Haut/Bas
Windows – Ctrl + Shift + Haut/Bas
Ubuntu – Ctrl + Shift + Alt + Haut/Bas
 - Formatter le code
Windows – Shift + Alt + F
Mac – Shift + Option + F
Ubuntu – Ctrl + Shift + I
 - Copier / Coller
Windows – Ctrl + C / Ctrl + V
Mac – Command + C / Command + V
Ubuntu – Ctrl + C / Ctrl + V
-
- Supprimer une ligne
Mac – Command + Shift + K
Windows/Ubuntu – Ctrl + Shift + K
 - Trouver un mot/phrased
Mac – Command + F
Windows/Ubuntu – Ctrl + F

Sur un projet entier
Mac – Command + Shift + F
Windows/Ubuntu – Ctrl; + Shift + F
 - Renommer
Mac/Windows/Ubuntu – F2
 - Dupliquer une ligne
Mac - Shift + Option + Haut/Bas
Windows - Shift + Alt + Haut/Bas
 - Annuler/Rétablir
Mac – Command + Z / Command + Y
Windows/Ubuntu – Ctrl + Z / Ctrl + Y

2 - La structuration d'une page HTML

Qu'est ce qu'une balise HTML ?

Un élément HTML est défini par une balise d'ouverture, du contenu, puis une balise de fermeture.

```
<nombalise> Le contenu de la balise </ nombalise>
```

Le contenu d'un élément est tout ce qui se trouve entre la balise d'ouverture et la balise de fermeture, par exemple :

```
<h1>Mon premier titre</ h1>
```

La structure d'une page HTML

```
<html>
```

```
<head>
```

```
<title>Titre de la page</title>
```

```
</head>
```

```
<body>
```

```
<h1>Un titre</h1>
```

```
<p>Un paragraphe.</p>
```

```
<p>Un autre paragraphe.</p>
```

```
</body>
```

```
</html>
```

Une page **HTML** simple

```
<html>
<head>
  <title>Titre de la page</title>
  <meta charset="utf-8">
</head>
<body>
  <h1>Mon premier titre</h1>
  <p>
    Mon premier paragraphe.
  </p>
</body>
</html>
```

Explication

`<html>` : racine de votre page HTML

`<head>` : contient les métadonnées (titre du document, l'encodage, la déclaration des styles et scripts ...) de votre page HTML. Elles ne sont pas affichées dans la page.

`<title>` : le titre de votre page HTML (celui qui est affiché comme nom d'onglet). Vous ne pouvez en définir qu'un seul.

`<body>` : le corps de votre page HTML, il contient tous les éléments visibles comme les titres, paragraphes, images, liens, tableaux, listes ...

`<h1>` : un titre sur la page, celui-ci est affiché. Vous pouvez en avoir plusieurs.

`<p>` : un paragraphe.

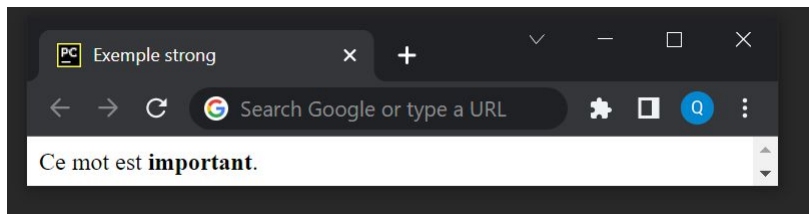
Les éléments **peuvent être imbriqués**

Un élément HTML peut être placé à l'intérieur d'un autre élément.

On appelle cela l'imbrication.

Par exemple, si nous voulons écrire un paragraphe et mettre en valeur un mot, nous pouvons utiliser la balise `` à l'intérieur du paragraphe.

```
<p>Ce mot est <strong>important</strong></p>
```





Notre première **page HTML**

Ecrivons ensemble notre première page HTML : **salle-classe.html**



Hypertext Markup Language (HTML)

En français, on peut le traduire par langage de balisage hypertexte.

Nous venons de voir l'aspect balisage du HTML.

Nous allons voir l'aspect hypertexte, c'est-à-dire créer des liens entre les différentes pages.

La balise **<a>**

Elle sert à créer un lien vers une autre page.

Le texte à l'intérieur de la balise sera le texte affiché.

L'attribut href définit la destination du lien.

```
<a href="description.html">la description</a>
```

Les attributs permettent de spécifier des informations supplémentaires à propos d'un élément.

Par exemple

```
<a href="description.html">la description</a>
```

href spécifie la destination du lien

On écrit toujours les attributs de la même façon, le nom de l'attribut, le signe égal et la valeur de l'attribut, entouré de double guillemets.

Vous pouvez trouver des codes HTML où les valeurs des attributs n'ont pas de double guillemets, cela fonctionne, mais ce n'est pas une bonne pratique.

Les chemins en HTML

Les chemins indiquent au navigateur la localisation d'un élément sur le serveur.

Un site web à une structure de dossier.

Les chemins sont utilisés lorsqu'on appelle des fichiers externes comme lorsqu'on appelle :

- un lien vers une autre page
- une image
- etc.



Chemin **absolu**

Un chemin absolu est une URL complète, par exemple :

```
<a src="https://google.com" alt="Mountain" />Google</a>
```

Chemin **relatif**

Un chemin relatif pointe sur un fichier relatif à la page actuelle.

Dans l'exemple suivant, le fichier se trouve dans un dossier image qui est à la racine de notre serveur web.

```

```

Dans l'exemple suivant, le fichier se trouve dans un dossier image le même dossier de notre serveur web.

```

```

Dans l'exemple suivant, le fichier se trouve dans un dossier image le un niveau au dessus de notre serveur web.

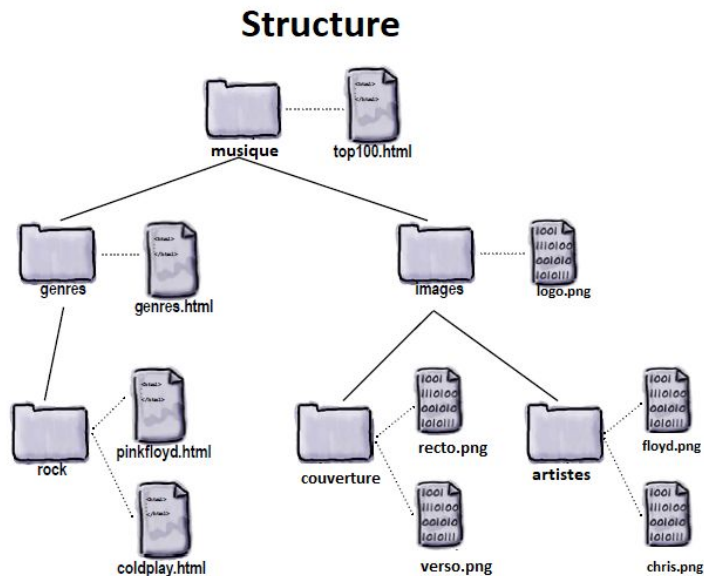
```

```

Exercice sur les chemins

Quel chemin pour aller :

- top100.html à genre.html
- top100.html à logo.png
- genres.html à logo.png
- top100.html à pinkfloyd.html
- coldplay.html à chris.png





Améliorons **notre salle de classe**

Créons une nouvelle salle de classe que nous allons améliorer dans un nouveau fichier : **nouvelle-salle-classe.html**

Pour créer des listes en HTML, nous avons besoin de deux balises. La première va entourer chaque élément de notre liste, le second est utilisé pour déterminer le type de liste que nous utilisons : ordonnée ou désordonnée

Chaque élément de la liste doit être entouré avec ``.

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

La liste entière doit se trouver entre les balise `` ou ``.

**** affiche une liste ordonnée, c'est-à-dire que l'ordre est important.

```
<ol>  
  <li>Paris</li>  
  <li>Berlin</li>  
  <li>Londres</li>  
</ol>
```

Affiche :

1. Paris
2. Berlin
3. Londres

**** est une liste sans ordre particulier, par exemple une liste à puce.

```
<ul>
  <li>Paris</li>
  <li>Berlin</li>
  <li>Londres</li>
</ul>
```

Affiche :

- Paris
- Berlin
- Londres

Dans une liste, vous devez toujours utiliser la balise ``.

Vous ne devez pas mettre une autre balise que `` à l'intérieur de `` ou ``.

Vous pouvez imbriquer les listes entre elles.



Essayons **les listes**

Nous allons mettre en pratique les listes dans le fichier : **blog.html**

Pour créer un tableau en HTML, il doit être composé de :

- Une balise `<table></table>` qui entoure le tableau
- Des cellules définies par les balises `<td></td>`
- Des lignes, qui commencent par `<tr>` et se terminent par `</tr>`
- Une entête (facultative), dans ce cas on remplace `<td>` par `<th>`

Exemple d'un tableau

```
<table>
  <tr>
    <th>Personne 1</th>
    <th>Personne 2</th>
    <th>Personne 3</th>
  </tr>
  <tr>
    <td>Quentin</td>
    <td>Pierre</td>
    <td>Fabien</td>
  </tr>
  <tr>
    <td>28</td>
    <td>24</td>
    <td>10</td>
  </tr>
</table>
```

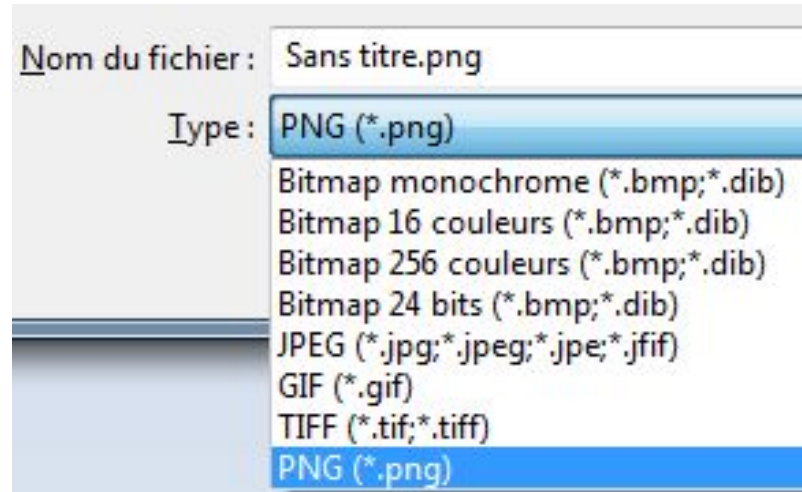



Ajoutons un tableau

Ajoutons un tableau à notre blog : **blog.html**

Les formats d'images

Il existe une multitude de format d'image, par exemple :



Les images au format JPEG (Joint Photographic Expert Group) sont très répandues sur le Web. Ce format est conçu pour réduire le poids des photos (c'est-à-dire la taille du fichier associé), qui peuvent comporter plus de 16 millions de couleurs différentes. La figure suivante est une photo enregistrée au format JPEG.

Les images JPEG sont enregistrées avec l'extension .jpg ou .jpeg.

Notez que le JPEG détériore un peu la qualité de l'image, d'une façon généralement imperceptible. C'est ce qui le rend si efficace pour réduire le poids des photos. Quand il s'agit d'une photo, on ne peut généralement pas détecter la perte de qualité. Par contre, si ce n'est pas une photo, vous risquez de voir l'image un peu « baver ». Dans ce cas, il vaut mieux utiliser le format PNG.

Le format PNG (Portable Network Graphics) est le plus récent de tous. Ce format est adapté à la plupart des graphiques (je serais tenté de dire « à tout ce qui n'est pas une photo »). Le PNG a deux gros avantages : il peut être rendu transparent et il n'altère pas la qualité de l'image.

Le PNG a été inventé pour concurrencer un autre format, le GIF, à l'époque où il fallait payer des royalties pour pouvoir utiliser des GIF. Depuis, le PNG a bien évolué et c'est devenu le format le plus puissant pour enregistrer la plupart des images.

Le PNG existe en deux versions, en fonction du nombre de couleurs que doit comporter l'image :

PNG 8 bits : 256 couleurs ;

PNG 24 bits : 16 millions de couleurs (autant qu'une image JPEG).

C'est un format assez vieux, qui a été néanmoins très utilisé (et qui reste très utilisé par habitude). Aujourd'hui, le PNG est globalement bien meilleur que le GIF : les images sont généralement plus légères et la transparence est de meilleure qualité. Je vous recommande donc d'utiliser le PNG autant que possible.

Le format GIF est limité à 256 couleurs (alors que le PNG peut aller jusqu'à plusieurs millions de couleurs).

Néanmoins, le GIF conserve un certain avantage que le PNG n'a pas : il peut être animé.

Quel format choisir ?

Si on résume, voici quel format adopter en fonction de l'image que vous avez :

Une photo : utilisez un JPEG.

N'importe quel graphique avec peu de couleurs (moins de 256) : utilisez un PNG 8 bits ou éventuellement un GIF.

N'importe quel graphique avec beaucoup de couleurs : utilisez un PNG 24 bits.

Une image animée : utilisez un GIF animé.

Créer votre propre site CV !

Il doit comporter plusieurs pages :

- accueil avec une description de votre parcours en informatique
- expériences, qui liste vos expériences professionnelles
- compétences, avec un tableau qui doit comporter le nom de votre compétence et votre niveau dans celle-ci. Lister toutes les technologies que vous connaissez (HTML, CSS, Javascript, Excel, Word, Windows ...)

Ajoutez quelques images, faites attention au format.

3 - Ajouter des styles CSS

Premiers pas **en CSS**

CSS est un langage différent de HTML, il permet de mettre en forme les pages web.

Vous devez sélectionner l'élément HTML que vous voulez mettre en forme puis lui appliquer des styles.

```
p {  
  background-color: yellow;  
}
```

Puis vous devez définir la **propriété** que vous voulez mettre en forme, sous la forme **propriété: valeur**;

Toutes vos propriétés doivent être placées entre des **accolades**, elles s'appliqueront à tous les éléments sélectionnés.

Une ou **plusieurs propriétés**

Vous pouvez ajouter autant de propriétés que vous voulez pour chaque **règle** CSS. Par exemple, vous voulez ajouter une bordure à vos paragraphes.

```
p {  
    background-color: yellow;  
    border 1px solid gray;  
}
```

Ajoutons du style à une page HTML

Pour ajouter du style CSS dans une page HTML, nous pouvons utiliser la balise `<style>`, dans le header de notre page.

```
<head>  
  <style>  
  
  </style>  
</head>
```

Le fonctionnement des sélecteurs

Lorsque nous écrivons :

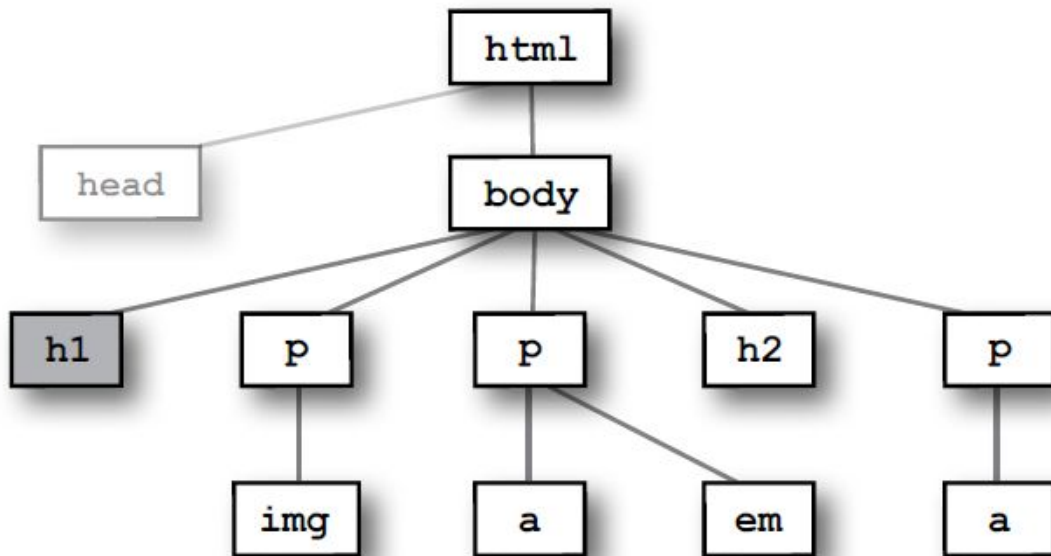
```
h1 {  
  color: gray;  
}
```

h1 est un sélecteur et la propriété sera appliquée uniquement aux éléments décrits par le sélecteur.

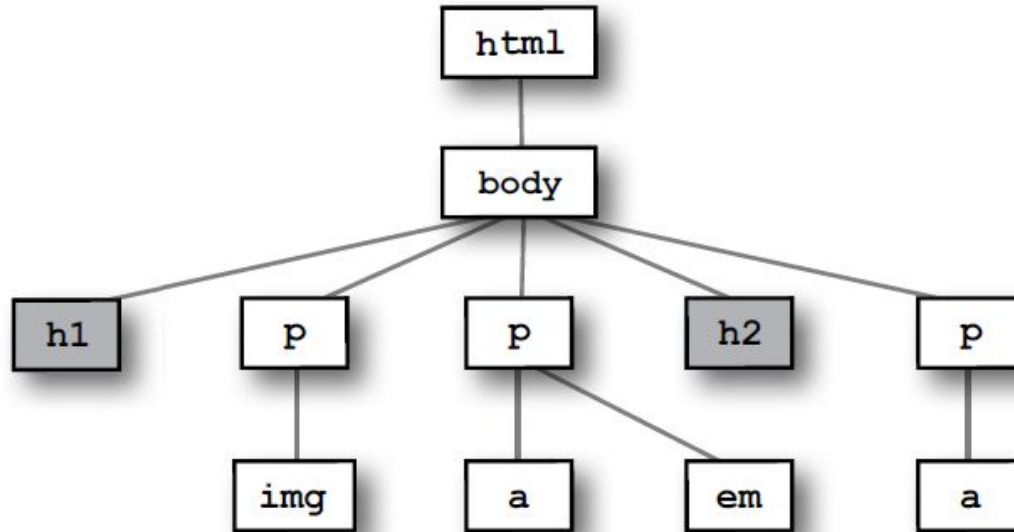
Vous pouvez sélectionner plusieurs éléments en les séparant par des virgules.

```
h1, h2 {  
  color: gray;  
}
```

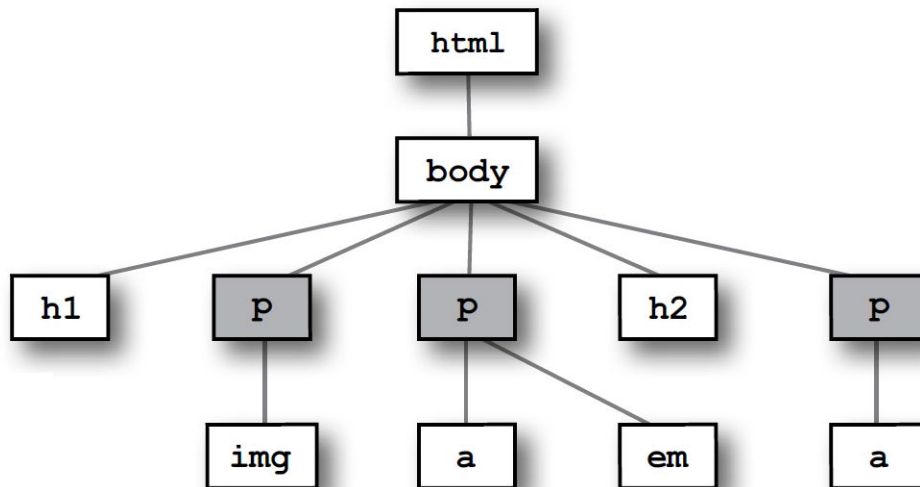
Visuellement, voici ce qu'il se passe lorsque nous sélectionnons l'élément h1.



En sélectionnant h1 et h2.



Et en sélectionnant une balise p.



La balise **<link>**

Nous préférierions ne pas mélanger le HTML et le CSS dans un seul fichier. Pour cela on peut utiliser la balise `<link>` qui va permettre de “linker” un fichier externe.

Dans notre cas, un fichier de type **text/CSS**, comme étant un **stylesheet**. Il faut aussi préciser le chemin du fichier, par exemple.

```
<link type="text/css" rel="stylesheet" href="nouvelle-salle-classe.css" >
```

Préciser le type (=text/css) est optionnel dans les navigateurs modernes (en HTML5). `<link>` est une balise “vide”, elle n’a pas besoin d’être fermée.

Notion d'héritage

Les éléments HTML peuvent **hériter** des attributs de leurs parents.

En changeant la police d'un `<p>`, les `<a>` et `` vont hériter de cette propriété.

Nous pouvons aller plus loin, en modifiant directement la police depuis le `<body>` !

Surcharger un héritage

En ajoutant la propriété au `<body>`, cela modifiera la police de la page entière.

Que faire en cas d'exception ?

Nous pouvons **surcharger** la police d'un élément enfant en modifiant la même propriété, cela **surchargera** la propriété.

En CSS, la **règle** la plus spécifique sera toujours utilisée.

Ajouter une classe

Nous pouvons ajouter une classe à un élément HTML avec l'attribut **class**.

```
<p class="machine-cafe">
```

Une classe est souvent utilisée pour sélectionner des éléments HTML depuis le CSS.

Pour la sélectionner depuis le CSS, on utilise le **nom balise.nom classe**.

```
p.machine-cafe {  
    background-color: yellow;  
}
```

Vous pouvez appliquer à un attribut **class** à toute balise HTML.

Ajouter **plusieurs classes**

Une balise peut avoir plusieurs classes.

```
<p class="machine-cafe description copyright">
```



En pratique

Appliquons ces concepts à **salle-classe.html**

Polices **par défaut**

CSS a 5 noms génériques pour les polices : serif, sans-serif, monospace, cursive et fantasy.

La police de caractères exacte utilisée dépend de chaque navigateur et peut varier pour chaque système d'exploitation sur lequel ils s'exécutent.

Dans le pire des cas, le navigateur essaiera de trouver une police appropriée.



Les balises **font**

Il existe beaucoup de propriétés CSS pour customiser vos textes.

Changer la police avec **font-family**

font-family: Verdana, Geneva, Arial, sans-serif;

Contrôler la taille avec **font-size**

font-size: 14px;

La couleur avec **color**

color: silver;

Le “poids” avec **font-weight**

font-weight: bold;

Soulignement ou barrer le texte avec **text-decoration**

text-decoration: underline;

Plusieurs **fonts**

Vous pouvez attribuer plusieurs polices à un élément.

`font-family: Verdana, Geneva, Arial, sans-serif;`

Le navigateur va regarder s'il a la police Verdana de disponible, si ce n'est pas le cas il va essayer sur Geneva, sinon Arial etc.

Il est conseillé de toujours inclure une police générique à la fin de votre propriété `font-family` par sécurité.

Documentation CSS : **developer.mozilla.org**

Quelques exemples :

- <https://developer.mozilla.org/fr/docs/Web/CSS/font-style>
- <https://developer.mozilla.org/fr/docs/Web/CSS/text-transform>
- <https://developer.mozilla.org/fr/docs/Web/CSS/text-decoration>

Comment les **fonts** fonctionnent

Au chargement de la page le navigateur charge vos fichiers HTML et CSS puis il va chercher les fichiers de police (.woff)

Une fois le fichier retrouvé, le navigateur peut afficher la police.

Vous pouvez ajouter votre propre fonts, on appelle cela les **Web fonts**.

Il faut avant tout déclarer notre Web Font, en précisant un nom et une source.

```
@font-face {  
  font-family: "Emblema One";  
  src: url("https://fonts.gstatic.com/s/emblemaone/v21/nKKT-G00F5dSY8vzG0rOELRIFF9LGw.woff")  
}
```

La police est ensuite utilisable comme toutes les autres.

```
h1 {  
  font-family: "Emblema One", sans-serif;  
}
```

Les **fichiers** **fonts**

Il existe type de fichier fonts :

- TrueTypeFonts : **.ttf**, format standard le plus ancien
- OpenTypeFonts : **.otf**, Open Type est une amélioration de TrueTypeFonts
- EmbeddedOpenType fonts : **.eot**, une forme compact de OpenType. Son propriétaire est Microsoft et il est seulement compatible IE et Edge.
- SVG fonts : **.svg**, Scalable Vector Graphics, est un format d'image standard pour le Web Fonts. Très bien supporté par les navigateurs modernes.
- Web open font format : **.woff**, basé sur TrueType, est devenu le standard pour les Web Fonts.

Il faut l'ajouter en haut de nos fichiers CSS.

Nous devons définir les propriétés :

- font-family, qui sera le nom d'utilisation de la police dans le fichier CSS. Généralement on met simplement le nom de la police.
- src, qui indique la source de la police. On peut indiquer plusieurs sources, séparées par des virgules, comme des fichiers .ttf et woff.

```
@font-face {  
    font-family: "Emblema One";  
    src:  
url("https://fonts.gstatic.com/s/emblemaone/v21/nKKT-GO0F5dSY8vzG0rOELRIFF9LGw.woff2"),  
url("https://fonts.gstatic.com/s/emblemaone/v21/nKKT-GQ0F5dSY8vzG0rOELRIFF9LGw.ttf" );  
}
```

Ajuster la **taille des polices**

On utilise la propriété font-size, on peut la définir avec plusieurs unités de mesure :

- En pixel, on indique au navigateur la hauteur d'une lettre en pixel.

```
font-size: 14px;
```

- En pourcentage, c'est un pourcentage relatif par rapport à l'élément parent.

```
font-size: 150%;
```

- Relatif à la taille courante de l'élément avec em. 2em veut dire fois plus grand que la taille actuelle.

```
font-size: 1.2em;
```

- Par mot clé, chaque palier est environ 20% plus grand que le précédent tel que : xx-small, x-small, small, medium, large, x-large

```
font-size: small;
```

- Et bien d'autres que vous pouvez retrouver sur https://www.w3schools.com/cssref/css_units.php



Quelle unité de mesure choisir ?

Je vous conseille de définir small ou medium comme taille par défaut dans votre body, cela servira de taille par défaut et de référence pour les autres tailles.

Puis définissez vos tailles relativement à body utilisant des pourcentages ou em.

Cela permet l'ajustement et la modification des polices plus facilement, en changeant la taille du body vous pouvez ainsi ajuster toutes vos polices.



Modifier le **font-weight**

La propriété font-weight permet de modifier l'épaisseur du trait de votre police.

Vous pouvez utiliser des mots-clés comme **light**, **bold** ou utiliser un nombre entre **100** et **900** ou, dernière possibilité, utiliser les deux mots-clés relatifs **lighter** et **bolder**.

Changer le **style d'une police**

font-style met votre texte en italique et permet d'en modifier l'inclinaison.

```
font-style: italic;
```

```
font-style: oblique;
```

```
font-style: oblique 40deg;
```


Les couleurs

Les couleurs sont un mélange de rouge, bleu et vert.



RGB

En CSS, il y a environ 150 couleurs prédéfinies.

Si vous voulez une couleur personnalisée vous pouvez :

- `rgb(80%, 40%, 0%)`
- `rgb(204, 102, 0)`
- Avec la valeur hexadécimale `#cc6600`

Les 16 couleurs de base

Bleu clair	aqua	#00FFFF	
Noir	black	#000000	
Bleu	blue	#0000FF	
Mauve	fuchsia	#FF00FF	
Gris	gray	#5D5D5D	
Vert	green	#005D00	
Vert clair	lime	#00FF00	
Marron	maroon	#5D0000	

Les 16 couleurs de base

Bleu marine	navy	#00005D	
Vert olive	olive	#5D5D00	
Pourpre	purple	#5D005D	
Rouge	red	#FF0000	
Argent	silver	#A8A8A8	
Cyan foncé	teal	#005D5D	
Blanc	white	#FFFFFF	
Jaune	yellow	#FFFF00	



Essayons

Appliquons ces concepts à **cafe.html**

4 - HTML, balises, blocs et tableaux

4.1 - Les blocs

Les éléments de type **bloc**

- Les balises de type bloc HTML sont des boîtes d'éléments qui peuvent contenir des éléments orphelins (comme du texte, simples caractères...), des balises de type en ligne, des balises de type auto fermante et même d'autres balises de type bloc.
- Les balises de type bloc se positionnent toujours les unes en dessous des autres.
- Leur largeur native (par défaut) est celle de l'élément qui les contient.

Les blocs : type block

Un élément de type **block** occupe tout l'espace horizontal de son élément conteneur (parent) et la hauteur de son contenu.

Cet espace est appelé un bloc.

Les navigateurs affichent les blocs avec une nouvelle ligne avant et une nouvelle ligne après l'élément.

Horizontalement, le bloc s'étend toujours autant qu'il le peut.

Exemples :

- `<p>`
- `<h1>`, `<h2>` ...
- ``

Les blocs : type inline

Un élément de type **inline** est un élément qui est placé à l'intérieur d'une ligne avec le texte plutôt que sur une nouvelle ligne.

Il ne crée pas de nouvelle ligne sur la page.

Il va seulement prendre le plus petit espace possible pour afficher son contenu.

Exemples :

- `<a>`
- ``
- ``
- `
`
- ``
- `<i>`

4.2 - Balises et attributs supplémentaires

La balise `<div>` 1/2

En HTML, la balise `<div>` est une balise utilisée pour regrouper d'autres éléments HTML en une seule unité.

La balise `<div>` crée un nouveau bloc de contenu, qui peut être utilisé pour organiser et formater le contenu d'une page web.

La balise `<div>` est souvent utilisée comme conteneur générique pour d'autres éléments HTML, comme des paragraphes de texte, des images, des tableaux, ou des formulaires.

Elle peut également être utilisée pour appliquer des styles, comme les couleurs, les polices, ou les marges, à l'ensemble des éléments qu'elle contient.

La balise `<div>` 2/2

Par exemple, si vous voulez organiser le contenu de votre page en deux colonnes, vous pouvez utiliser deux balises `<div>` pour créer deux blocs de contenu, et utiliser des styles CSS pour leur donner la bonne largeur et la bonne position.

Vous pouvez également utiliser la balise `<div>` pour créer des menus, des en-têtes, ou des pieds de page, en lui donnant un identifiant ou une classe unique pour pouvoir y appliquer des styles spécifiques.

Il est important de noter que la balise `<div>` n'a pas de signification spécifique en soi, et qu'elle ne doit être utilisée que pour regrouper d'autres éléments HTML.

Pour des éléments de contenu spécifiques, il est recommandé d'utiliser des balises plus spécialisées, comme `<p>` pour les paragraphes, `` pour les images, ou `<form>` pour les formulaires.

L'attribut **id** 1/2

L'attribut id en HTML est utilisé pour donner un nom unique à un élément d'une page HTML. Cet identifiant peut être utilisé pour ajouter des styles à cet élément en utilisant CSS, ou pour faire référence à cet élément dans du code JavaScript.

Par exemple, si vous voulez donner un nom unique à un paragraphe de votre page, vous pouvez utiliser l'attribut id comme ceci :

```
<p id="intro">
```

```
  Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
```

```
</p>
```

L'attribut **id** 2/2

Le résultat de ce code sera que le paragraphe aura un nom unique, qui pourra être utilisé pour ajouter des styles à ce paragraphe en CSS ou pour faire référence à ce paragraphe dans du code JavaScript.

Il est important de noter que l'attribut id doit être unique sur la page HTML, c'est-à-dire qu'il ne doit pas y avoir deux éléments avec le même identifiant. Cela permet d'éviter les ambiguïtés lorsqu'on fait référence à un élément en utilisant son identifiant.

En CSS, pour sélectionner un id vous pouvez utiliser **#**

```
#intro {  
  /* styles à ajouter à l'élément */  
}
```

Il est important de noter que l'identifiant d'un élément en CSS doit être écrit exactement comme dans le code HTML, y compris les majuscules et les minuscules.

L'attribut **alt** 1/2

L'attribut alt en HTML est utilisé pour ajouter une description alternative à une image.

Cette description est importante pour les utilisateurs qui ne peuvent pas voir l'image, soit parce qu'ils utilisent un navigateur web qui ne supporte pas les images, soit parce qu'ils ont des difficultés pour voir les images (par exemple en raison d'une déficience visuelle).

La description alternative est affichée à la place de l'image, soit dans le texte alternatif (si le navigateur web supporte ce type de texte), soit dans un outil d'aide pour les utilisateurs ayant des difficultés pour voir les images.

L'attribut **alt** 2/2

Pour utiliser l'attribut alt en HTML, il suffit de l'ajouter à la balise de l'image, comme dans l'exemple suivant :

```

```

Dans cet exemple, la description de l'image est "Description de l'image". Cette description doit être courte et concise, et doit refléter le contenu de l'image de manière précise et claire.

En résumé, l'attribut alt en HTML est très important pour la qualité de l'accessibilité des pages web. Il permet d'ajouter une description alternative à une image, qui est utile pour les utilisateurs qui ne peuvent pas voir l'image.

Balise : DOCTYPE 1/2

Le mot "doctype" en HTML désigne la déclaration de type de document. Cette déclaration se trouve en tête de tous les documents HTML, et indique au navigateur web le type de document qu'il va afficher.

La déclaration de type de document est importante car elle permet au navigateur web de savoir comment interpréter le code HTML du document. Si la déclaration de type de document est incorrecte ou manquante, le navigateur web peut ne pas afficher le document de la manière attendue.

La syntaxe de la déclaration de type de document en HTML est la suivante :

```
<!DOCTYPE html>
```

Balise : DOCTYPE 2/2

Cette déclaration indique au navigateur web que le document est un document HTML5, qui est la dernière version du HTML. La déclaration de type de document doit être la première ligne de code dans le document HTML, avant toutes les autres balises HTML.

CSS : Margin et padding 1/2

En CSS, margin et padding sont des propriétés qui permettent de définir des espaces autour d'un élément. La propriété margin permet de définir des marges autour de l'élément, tandis que la propriété padding permet de définir des espaces à l'intérieur de l'élément, entre le contenu de l'élément et ses bords.

Les marges et les espacements sont généralement utilisés pour créer des espaces entre les éléments d'une page HTML, pour les mettre en relief ou pour les séparer visuellement les uns des autres.

Voici un exemple de comment utiliser margin et padding en CSS pour ajouter des espaces autour d'un élément :

```
#mon-element {  
  margin: 20px;  
  padding: 10px;  
}
```

CSS : Margin et padding 2/2

Dans cet exemple, nous définissons une marge de 20 pixels autour de l'élément avec l'identifiant `mon-element`, ainsi qu'un espace de 10 pixels à l'intérieur de l'élément, entre son contenu et ses bords.

Il est important de noter que vous pouvez définir des valeurs différentes pour chaque côté des marges et des espacements en utilisant les propriétés `margin-top`, `margin-right`, `margin-bottom` et `margin-left` pour les marges, et `padding-top`, `padding-right`, `padding-bottom` et `padding-left` pour les espacements. Par exemple, pour ajouter une marge de 20 pixels en haut et en bas d'un élément, mais pas sur les côtés, vous pouvez utiliser la syntaxe suivante :

```
#mon-element {  
  margin-top: 20px;  
  margin-bottom: 20px;  
}
```

Le résultat de ce code sera un élément avec des marges de 20 pixels en haut et en bas, mais pas sur les côtés.

La propriété `border` en CSS est utilisée pour définir une bordure autour d'un élément HTML. Elle peut être configurée avec différentes valeurs pour personnaliser la taille, le style et la couleur de la bordure.

La propriété **`border`** est une propriété raccourcie qui peut être décomposée en trois propriétés distinctes :

`border-width` : détermine l'épaisseur de la bordure. Elle peut être spécifiée en pixels (par exemple, 2px) ou en utilisant un des mots-clés suivants : `thin`, `medium`, `thick`.

`border-style`: détermine le style de la bordure. Quelques valeurs courantes comprennent `solid`, `dashed`, `dotted`, `double`, `groove`, `ridge`, `inset`, `outset`, et `none` (aucune bordure).

`border-color`: détermine la couleur de la bordure. Elle peut être spécifiée en utilisant des noms de couleur (`red`, `blue`, `green`, etc.), des valeurs hexadécimales (`#FF0000` pour rouge, par exemple), des valeurs RGB (`rgb(255,0,0)` pour rouge, par exemple) ou des valeurs RGBA pour ajouter de la transparence.

La balise **<meta>**

`<meta>` en HTML est utilisé pour fournir des informations supplémentaires sur une page HTML, comme la description de la page, les mots-clés associés à la page, l'auteur de la page, etc.

Ces informations sont généralement utilisées par les moteurs de recherche pour mieux comprendre le contenu de la page et la classer dans les résultats de recherche.

L'élément `<meta>` est un élément vide, c'est-à-dire qu'il n'a pas de contenu et qu'il est utilisé uniquement pour définir des attributs. Par exemple, pour ajouter une description à une page, vous pouvez utiliser l'attribut `description` de l'élément `<meta>` comme ceci :

```
<meta name="description" content="Ceci est une page d'exemple qui montre comment utiliser l'élément <meta> en HTML.">
```

Le résultat de ce code sera que la description que vous avez définie sera associée à la page. Cette description ne sera pas visible sur la page elle-même, mais elle pourra être utilisée par les moteurs de recherche pour afficher des informations supplémentaires sur la page dans les résultats de recherche.

La balise **<blockquote>**

`<blockquote>` en HTML permet de représenter une citation longue ou un texte important qui est indiqué comme étant tiré d'une autre source. Le contenu de cet élément est généralement affiché en indenté et en italique pour le mettre en relief. Par exemple :

```
<blockquote>
```

```
<p>
```

```
    Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor  
    incididunt ut labore et dolore magna aliqua.
```

```
</p>
```

```
</blockquote>
```

Le résultat de ce code sera une citation qui sera mise en évidence sur la page HTML.

La balise ****

`` en HTML permet de grouper des parties d'un texte en un seul élément, ce qui peut être utile pour ajouter des styles à ces parties de texte. Par exemple, vous pouvez utiliser `` pour mettre en gras ou en italique une partie d'un paragraphe :

```
<p>
```

```
  Lorem ipsum dolor sit amet, consectetur adipiscing  
elit</span>, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
```

```
</p>
```

Le résultat de ce code sera un paragraphe où seule la partie qui est incluse dans `` sera en gras.

Définir le **titre d'un tableau**

<caption> en HTML permet de définir un titre pour une table. Ce titre est affiché au-dessus de la table et il peut être utilisé pour donner une brève description ou un résumé de la table.

```
<table>
  <caption>Résumé des ventes</caption>
  <tr>
    <th>Article</th>
    <th>Prix</th>
  </tr>
  <tr>
    <td>Livre</td>
    <td>10€</td>
  </tr>
  <tr>
    <td>Stylo</td>
    <td>2€</td>
  </tr>
</table>
```

Le résultat de ce code sera une table avec un titre qui dit "Résumé des ventes" au-dessus de la table.

Tableaux : Comment afficher **une cellule de plusieurs lignes ?**

L'attribut `rowspan` de l'élément `<td>` en HTML permet de faire fusionner plusieurs lignes d'une table. Par exemple, si vous avez une table à deux colonnes et que vous voulez que la première cellule de la première colonne fusionne les deux premières lignes, vous pouvez utiliser `rowspan="2"` comme ceci :

```
<table>
<tr>
  <td rowspan="2">Ceci est une cellule qui fusionne deux lignes</td>
  <td>Ceci est une cellule de la deuxième colonne, première ligne</td>
</tr>
<tr>
  <td>Ceci est une cellule de la deuxième colonne, deuxième ligne</td>
</tr>
</table>
```

Le résultat de ce code sera une table avec deux colonnes et deux lignes, où la première cellule de la première colonne fusionne les deux premières lignes.

Tableaux : Comment afficher **une cellule de plusieurs colonnes**?

L'attribut `colspan` de l'élément `<td>` en HTML permet de faire fusionner plusieurs colonnes d'une table. Par exemple, si vous avez une table à deux lignes et que vous voulez que la première cellule de la première ligne fusionne les deux premières colonnes, vous pouvez utiliser `colspan="2"` comme ceci :

```
<table>
<tr>
  <td colspan="2">Ceci est une cellule qui fusionne deux colonnes</td>
  <td>Ceci est une cellule de la troisième colonne, première ligne</td>
</tr>
<tr>
  <td>Ceci est une cellule de la première colonne, deuxième ligne</td>
  <td>Ceci est une cellule de la deuxième colonne, deuxième ligne</td>
  <td>Ceci est une cellule de la troisième colonne, deuxième ligne</td>
</tr>
</table>
```

Le résultat de ce code sera une table avec trois colonnes et deux lignes, où la première cellule de la première ligne fusionne les deux premières colonnes.

Propriétés CSS

	Properties
Colors	color
	background-color
Spacing	margin
	padding
	margin-left, margin-right, margin-top, margin-bottom
	padding-left, padding-right, padding-top, padding-bottom
Borders	border-width
	border-style
	border-color
	border (to set the width, style, and color in one step)
Text alignment	text-align
	text-indent
	word-spacing
	letter-spacing
	line-height
Fonts	white-space
	font-family
	font-size
	font-weight
	font-style
	font-variant
	text-decoration
Size	@font-face (for using fancy fonts; see page 244)
	width
	height
Layout	position
	left, right
	float, clear
Graphics	background-image
	background-repeat
	background-position

Exercice : Planning d'équipes

Reproduisez le tableau suivant en utilisant HTML et CSS.

Planning de l'équipe

Equipes	Lundi	Mardi	Mercredi	Jeudi	
Equipe1	Pierre			Quentin	Semaine 1
Equipe2	Julien		Frédéric	Luc	
Equipe3	Pierrick	Marie			Semaine 2
Equipe4	Pierre	Patrick	Isabelle	Simon	

4 - Création de formulaire

Une URL est l'adresse d'une ressource unique donnée à travers le Web.

Ces ressources peuvent être en HTML, un document CSS, une image ...

La ressource représentée par l'URL et l'URL elle-même sont gérées par le serveur Web, c'est au propriétaire du serveur de choisir quelle URL est associée à une ressource.

Anatomie d'une URL

Une URL contient plusieurs parties, certaines obligatoires d'autres facultatives.

Le protocole est la première partie d'une URL, cela indique le protocole utilisé par le navigateur pour récupérer le contenu. Nous reviendrons sur les protocoles dans le prochain chapitre.



The diagram shows the URL `http://www.example.com:80/path`. The `http://` portion is highlighted with a green square. A green arrow points from the bottom of this square to the word **Protocole**.

`http://www.example.com:80/path`

Protocole

URL : Le nom de domaine

Le nom de domaine est la seconde partie d'une URL. Il est séparé par du protocole par `://`

Le nom de domaine indique quel serveur web est requêté. Il peut être remplacé par une adresse IP mais c'est moins pratique.

Un port peut être ajouté. S'il est omis alors le navigateur utilise les ports par défaut (80 pour HTTP, 443 pour HTTPS). Le nom de domaine et le port sont séparés par :



URL : le chemin de la ressource

/path/to/myfile.html est le chemin d'accès à la ressource sur le serveur Web. Au début du Web, un chemin comme celui-ci représentait un emplacement de fichier physique sur le serveur Web. De nos jours, il s'agit surtout d'une abstraction gérée par des serveurs Web sans aucune réalité physique.

n:80 /path/to/myfile.html?key1=value1



Chemin de la ressource

URL : les paramètres

?key1=value1&key2=value2 sont des paramètres supplémentaires fournis au serveur Web. Ces paramètres sont une liste de paires clé/valeur séparées par le symbole &. Chaque serveur Web a ses propres règles concernant les paramètres, et le seul moyen fiable de savoir si un serveur Web spécifique gère des paramètres est de demander au propriétaire du serveur Web.

html?key1=value1&key2=value2



Paramètres

Ce qu'il **faut retenir**

- Les éléments HTML indiquent au navigateur comment afficher le contenu.
- CSS décrit comment les éléments HTML doivent être affichés.
- HTTP(S) est le protocole permettant d'envoyer des requêtes aux serveurs web par l'intermédiaire d'un navigateur.
- Une URL est l'adresse d'une ressource unique donnée à travers le Web.

A quoi servent **les formulaires** ?

Les formulaires HTML sont généralement utilisés pour collecter des informations telles que des noms, des adresses électroniques, des numéros de téléphone et des commentaires à partir des utilisateurs.

Les formulaires HTML sont composés de

- champs de saisie
- boutons d'envoi
- cases à cocher
- menus déroulants

Ils sont utilisés pour prendre en charge la saisie de données par les utilisateurs.

Comment fonctionnent les formulaires ?

Lorsqu'un utilisateur remplit un formulaire HTML, les données qu'il a saisies sont généralement envoyées à un serveur web pour traitement ultérieur.

Pour envoyer les données du formulaire à un serveur web, l'utilisateur doit généralement cliquer sur un bouton d'envoi situé à la fin du formulaire.

Les formulaires HTML sont créés en utilisant des balises HTML spéciales, par exemple :

- la balise `<form>` est utilisée pour créer un formulaire
- la balise `<input>` est utilisée pour créer un champ de saisie
- de nombreuses autres balises HTML sont également utilisées pour des boutons, des cases à cocher, des menus déroulants ...

Une fois les données transmises au serveur web, elles peuvent être enregistrées dans une base de données, utilisées pour générer un courriel ou pour effectuer un calcul.

Les possibilités de traitement des données sont pratiquement illimitées et dépendent de la façon dont le formulaire est configuré et utilisé.

Ce qu'il faut écrire **en HTML**

Ajoutez la balise `<form>` pour définir le début du formulaire.

```
<form action="http://monsite.com/form.php" method="POST">
```

Pour ajouter des champs de saisie utiliser des balises `<input>`.

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

Bouton d'envoi, `<button>` ou `<input>` :

```
<button type="submit">Envoyer</button>
```

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


Le **text input**

Le **text input** est utilisé pour entrer une ligne de texte.

```
<input type="text" name="nomcomplet">
```

Des attributs optionnels permettent de rentrer un nombre maximum de caractères et la largeur du champs.

Le **submit input**

Le **submit input** crée un bouton qui permet de soumettre le formulaire.

```
<input type="submit">
```

Quand vous appuyez sur ce bouton, le navigateur envoie le formulaire au serveur.

Le **radio input**

Le radio input est un type de champ de saisie qui permet à l'utilisateur de sélectionner une seule option parmi plusieurs choix.

Les radio inputs sont souvent utilisés dans les formulaires HTML pour collecter des informations telles que des réponses à des questions ou des préférences d'utilisateur.

```
<input type="radio" name="chaudoufroid" value="chaud">  
<input type="radio" name="chaudoufroid" value="froid">
```

Le **checkbox input**

Le **checkbox input** permet de sélectionner plusieurs options sous forme de case à cocher. Chaque case peut être sélectionnée indépendamment.

```
<input type="checkbox" name="epice" value="Sel">  
<input type="checkbox" name="epice" value="Poivre">
```

Le **text area**

Le **text area** crée une zone de texte multiligne. Si le texte ne tient pas dans la zone de texte, une barre de scroll apparaîtra sur le côté.

```
<textarea name="commentaire" rows="10" cols="48"></textarea>
```

Le **select** est un menu qui permet la sélection d'une option parmi plusieurs choix.

```
<select name="personnage">
  <option value="superman">superman</option>
  <option value="batman">batman</option>
  <option value="ironman">ironman</option>
</select>
```



En pratique : Boutique M2i

Nous allons créer un formulaire pour commander du café en ligne.

Exercice : Ajouter du style

Essayons d'ajouter du CSS à notre formulaire :

- Ajoutez une marge aux éléments
- Mettez une couleur de fond pour toute la page
- Agrandissez le champ commentaire pour qu'on puisse écrire un paragraphe.

Mettons un peu de style à notre CV !

En utilisant les notions que nous avons vues aujourd'hui, essayons d'améliorer l'allure de notre site.

- définissez des tailles relatives à tous vos textes
- mettez un peu de couleur
- n'hésitez pas à souligner, mettre en gras, en italique ...
- ... ou surcharger des propriétés, comme celles des liens
- essayez d'avoir une architecture propre
- une page contact, avec : Nom, prénom, e-mail, société, téléphone, votre message, une case à cocher pour l'autorisation de marketings - conformité RGPD, un bouton envoyer

5 - Les ajouts de HTML5

HTML 5 : définition

HTML5 est la dernière version de HTML. Elle offre de nouvelles fonctionnalités et améliorations :

- Amélioration de la sémantique des éléments
- Elle prend en charge les éléments multimédias, comme les vidéos et les animations, de manière native et efficace.
- Plus responsive et portable.



Ce qu'on a déjà vu

`<!DOCTYPE html>`, `<meta charset="utf-8">`, `<html lang=fr>`, `<link>` sont toutes des syntaxes HTML 5.

Les règles **assouplies**

HTML 5 a assouplie plusieurs règles :

- <html> <head> et <body> sont devenus optionnels
- Ignore la casse <p> est équivalent à <P>
- Les attributs n'ont plus besoin de double quote

```
<img alt="Un cheval" src=cheval.jpg>
```

Attention, même si HTML5 le permet, ce sont de mauvaises pratiques.

Balises ajoutées : aperçu

Nous allons commencer par voir les nouvelles balises HTML5.

Sémantique pour structurer une page	<code><article></code> , <code><aside></code> , <code><figcaption></code> , <code><figure></code> , <code><footer></code> , <code><header></code> , <code><nav></code> , <code><section></code> , <code><details></code> , <code><summary></code>
Sémantique pour du texte	<code><mark></code> , <code><time></code> , <code><wbr></code>
Formulaires	<code><input></code> (tous les nouveaux sous-types), <code><datalist></code> , <code><keygen></code> , <code><meter></code> , <code><progress></code> , <code><command></code> , <code><menu></code> , <code><output></code>
Multimédias	<code><audio></code> , <code><video></code> , <code><source></code>
Canvas	<code><canvas></code>

Ancienne façon de structurer les pages

```
<div class="Header">
  <h1>Titre</h1>
  <p class="description">Paragraphe.</p>
</div>
<div class="Content">
  <p>...</p>
  <p>...</p>
  <h2>...</h2>
  <p>...</p>
  ...
</div>
<div class="Footer">
  <p class="Disclaimer">...</p>
  <p>
    <a href="AboutUs.html">About Us</a>
    ...
  </p>
  <p>Copyright ©2022</p>
</div>
```

Une nouvelle façon de structurer les pages

```
<header class="header">
  <h1>Titre</h1>
  <p class="description">....</p>
</header>
<div class="content">
  <p>...</p>
  <p>...</p>
  <h2>...</h2>
  <p> ...</p>
  ...
</div>
<footer class="footer">
  <p class="disclaimer">...</p>
  <p>
    <a href="aboutUs.html">About Us</a>
    ...
  </p>
  <p>Copyright ©2022</p>
</footer>
```


Avantage de cette **nouvelle structure**

Parce que la page contient seulement un `<header>`, vous pouvez sélectionner plus facilement son contenu en CSS.

```
header h1 {  
  ...  
}  
  
/* La description dans le <header>. */  
header.description {  
  ...  
}
```

<article>

<article> est utilisé pour représenter un contenu autonome, indépendant et réutilisable, comme un article de blog, une publication dans un journal ou un forum, un commentaire d'utilisateur, etc.

```
<article>  
  <h2>Article</h2>  
  <p>Contenu ...</p>  
</article>
```

<figure>

<figure> est utilisé pour représenter du contenu autonome, comme des images, des schémas, des graphiques, des diagrammes, des tableaux, etc.

Éventuellement accompagné d'une légende facultative, qui est spécifiée à l'aide de l'élément <figcaption>.

La figure, sa légende et son contenu sont référencés comme une seule unité.

```
<figure>  
    
  <figcaption>Un tigre</figcaption>  
</figure>
```

<aside> (aparté en français) est utilisé pour représenter du contenu qui est en marge du contenu principal d'une page Web.

L'élément <aside> est généralement utilisé pour inclure du contenu supplémentaire ou annexe, comme des notes de bas de page, des citations, des exemples, des anecdotes, etc.

<header> est utilisé pour représenter une en-tête de section ou de page.

L'élément <header> est généralement utilisé pour inclure du contenu qui donne une introduction ou un sommaire à la section ou à la page, comme un titre, un logo, un menu de navigation, etc.

Il peut être utilisé à différents niveaux de structure dans une page Web, il peut définir :

- l'en-tête d'une page entière
- d'une section de contenu
- d'un article
- d'un formulaire
- etc.

L'élément <header> permet aux utilisateurs et aux logiciels de détecter facilement le contenu de l'en-tête d'une page et de le traiter de manière appropriée.

<nav> est une section de la page Web qui contient des liens de navigation vers d'autres pages ou des parties de la page actuelle.

L'élément <nav> est généralement utilisé pour inclure un menu de navigation, un plan de site, un index, etc

```
<nav class="menu">
  <ul>
    <li><a href="#">Accueil</a></li>
    <li><a href="#">A propos</a></li>
    <li><a href="#">Contact</a></li>
  </ul>
</nav>
```

<footer> est utilisé pour représenter un pied de section ou de page.

L'élément <footer> est généralement utilisé pour inclure du contenu qui se trouve en fin de section ou de page, comme des informations de copyright, des liens vers des mentions légales ou des conditions d'utilisation, des coordonnées, etc.

```
<footer>  
  <p>Auteur: Quentin BIDOLET</ p>  
  <p><a href="mailto:quentin.bidolet@gmail.com" >quentin.bidolet@gmail.com</ a></p>  
</footer>
```

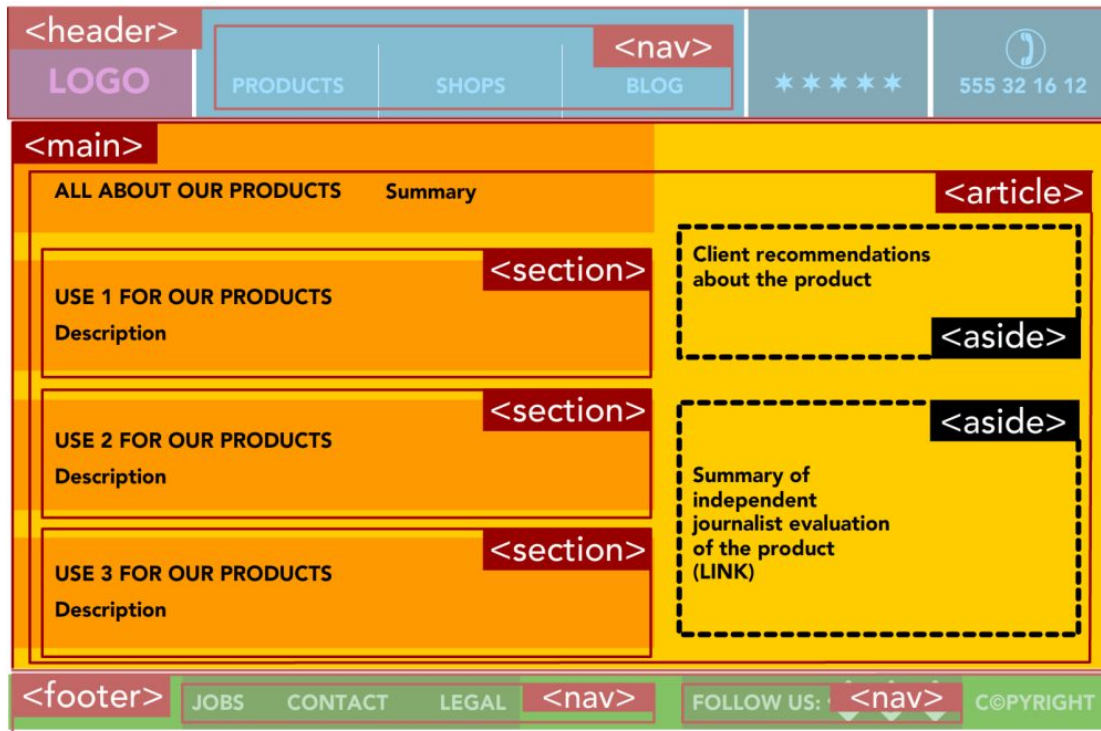
<section>

L'élément HTML <section> représente une section autonome générique d'un document, qui n'a pas d'élément sémantique plus spécifique pour la représenter.

Les sections doivent toujours avoir un titre, à de très rares exceptions près.

```
<section>  
  <h1>Titre de la section</h1>  
  <p>Contenu de la section</p>  
</section>
```


HTML5 : schéma sémantique



[source](#)

5.2 - CSS - Vue d'ensemble

Sélecteurs **simples**

- Sélecteur d'élément : cible tous les éléments de ce type (par exemple, **p** pour les paragraphes)
- Sélecteur de classe : cible tous les éléments ayant une classe spécifique (par exemple, **.ma-classe**)
- Sélecteur d'ID : cible un élément ayant un ID spécifique (par exemple, **#mon-id**)
- Sélecteur descendant : cible les éléments descendants d'un autre élément (par exemple, **ul li**)
- Sélecteur enfant : cible les éléments qui sont des enfants directs d'un autre élément (par exemple, **ul > li**)
- Sélecteur universel : cible tous les éléments de la page (*)

Sélecteurs **avancés**

- Sélecteur d'attribut : cible les éléments ayant un attribut spécifique (par exemple, `[type="checkbox"]`)
- Sélecteur de pseudo-classe : cible des états spécifiques des éléments (par exemple, `:hover`)
- Sélecteur de pseudo-élément : cible des parties spécifiques des éléments (par exemple, `::before`)
- Sélecteur de négation : cible tous les éléments qui ne correspondent pas à un autre sélecteur (par exemple, `:not(p)`)

Propriétés des boîtes

- En CSS, les boîtes (ou "boxes" en anglais) sont des éléments de base pour la mise en page et le positionnement des éléments sur une page web.
- Elle est composée de quatre parties principales :
 - le contenu (content) : texte, images etc.
 - la bordure (border) : ligne qui entoure le contenu
 - le rembourrage (padding) : espace entre le contenu et la bordure
 - la marge (margin) : espace entre la bordure et les éléments adjacents



5.3 - Formulaire en HTML5

Fieldset

```
<fieldset>
  <legend>Contact</legend>
  <label for="name">Nom<em>*</em></label>
  <input id="name"><br>
  <label for="telephone">Telephone</label>
  <input id="telephone"><br>
  <label for="email">Email <em>*</em></label>
  <input id="email"><br>
</fieldset>
```



Formulaire HTML5 : **vue d'ensemble**

Texte mono-ligne	<pre><input type="text"> <input type="password"></pre>	Une ligne de texte. Si type="password" alors le texte n'est pas montré en clair.
Texte multiligne	<pre><textarea>...</textarea></pre>	Une large zone de texte.
Case à cocher	<pre><input type="checkbox"></pre>	Une case à cocher, deux valeurs possibles.
Bouton Radio	<pre><input type="radio"></pre>	Bouton radio. Habituellement un groupe radio partage le même name, dans ce cas on peut sélectionner qu'une option.
Bouton	<pre><input type="submit"> <input type="image"> <input type="reset"> <input type="button"></pre>	Bouton cliquable standard. Submit/image envoient le formulaire au serveur. Reset "clear" les champs du formulaire. Type button ne fait rien à part executer un code Javascript.
Liste	<pre><select>...</select></pre>	Une liste où l'utilisateur peut sélectionner une option.

Formulaire : **placeholders**

Par défaut les formulaires sont vides.

Vous pouvez ajouter un **placeholder** pour aider vos utilisateurs à remplir les champs.

```
<label for="nom">Nom<em>*</em></label>  
<input id="nom" placeholder="Jean DUPONT"><br>  
<label for="telephone">Telephone</label>  
<input id="telephone" placeholder="06 xx xx xx xx"><br>
```

Formulaire : autofocus

Vous pouvez choisir la sélection par défaut de l'utilisateur, au chargement de la page, avec l'attribut **autofocus**.

```
<label for="nom">Nom<em>*</em></label>
<input id="nom" placeholder="Jean DUPONT" autofocus><br>
<label for="telephone">Telephone</label>
<input id="telephone" placeholder="06 xx xx xx xx"><br>
```

Formulaire : **required**

Vous pouvez choisir plusieurs champs qui doivent être obligatoirement remplis pour pouvoir soumettre le formulaire. Dans ce cas, vous devez ajouter l'attribut **required** aux champs.

```
<label for="nom">Nom<em>*</em></label>  
<input id="nom" placeholder="Jean DUPONT" autofocus required><br>  
<label for="telephone">Telephone</label>  
<input id="telephone" placeholder="06 xx xx xx xx" required><br>
```

Formulaire : **style CSS**

Vous pouvez styliser l'apparence des champs basés sur leurs **états**. Vous pouvez utiliser les **pseudo class** pour cela.

```
input:required {  
    background-color: lightyellow;  
}
```

```
input:required:invalid {  
    background-color: lightyellow;  
}
```

6 - Les écrans

Taille d'écran

- La taille de l'écran influence directement le rendu visuel et la mise en page des éléments sur la page web.
- Lorsqu'un site web est affiché sur un écran de petite taille, comme un smartphone, les éléments peuvent être difficiles à lire et à manipuler si la mise en page n'est pas adaptée.
- Le responsive design consiste à créer des mises en page qui s'adaptent à différentes tailles d'écran.

Résolution d'écran

- La résolution d'écran peut avoir un impact sur la qualité de l'affichage des images et des éléments graphiques sur la page web.
- La résolution d'écran fait référence au nombre de pixels affichés sur l'écran.
- Plus la résolution d'écran est élevée, plus le nombre de pixels affichés est important, ce qui améliore la netteté et la précision de l'affichage des images et des éléments graphiques.
- On peut utiliser des images haute résolution, avec une densité de pixels plus élevée (appelée "retina" ou "hi-dpi") pour garantir une qualité d'affichage optimale sur les écrans haute résolution.

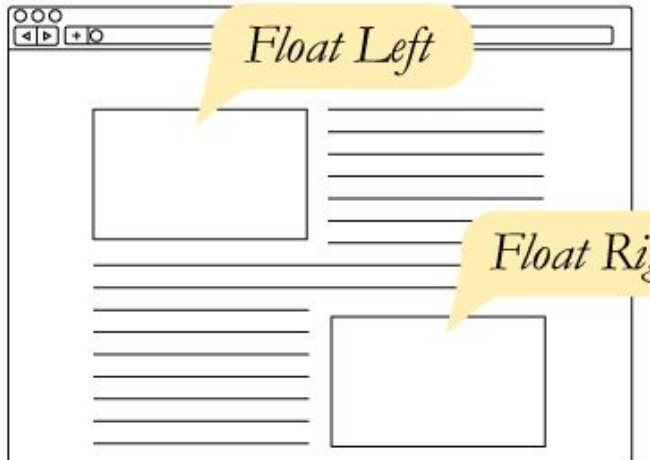
CSS pixel / Device pixel

- Le CSS pixel est une unité de mesure virtuelle qui est indépendante de la densité de pixels physique de l'écran.
- Le device pixel fait référence aux pixels physiques sur l'écran.
- Les pixels physiques peuvent être plus petits ou plus grands selon la densité de l'écran.
- Si la densité de pixels de l'écran est élevée, il peut être nécessaire d'utiliser des images haute résolution et des dimensions d'éléments plus grands pour garantir un affichage net et précis.

7 - Les modes de positionnement

Flottement

- Le flottement (ou "float" en anglais) est une propriété qui permet de positionner un élément d'une page web à gauche ou à droite d'un conteneur en faisant en sorte que les autres éléments de la page s'alignent autour de lui.



Position **absolue**

- La position absolue permet de positionner un élément HTML à une position spécifique sur une page web, indépendamment de la position de tous les autres éléments de la page.
- Elle est souvent utilisée pour créer des éléments de superposition comme des menus déroulants, des boîtes de dialogue ou des publicités, qui doivent être placés à des positions spécifiques et ne pas affecter le reste de la mise en page.
- Attention, lorsque vous utilisez la position absolue les éléments peuvent facilement se chevaucher ou sortir de la zone visible du navigateur.

Boîtes flexibles

- Les boîtes flexibles (ou "Flexbox" en anglais) permettant de disposer des éléments en ligne ou en colonne.
- Les éléments se réduisent ou s'agrandissent pour occuper l'espace disponible.
- Ils permettent de créer des mises en page responsives qui s'adaptent automatiquement à la taille de l'écran.
- Ils offrent également une grande flexibilité pour aligner les éléments, réorganiser l'ordre des éléments et répartir l'espace disponible de manière équitable entre les éléments.

Les grilles (grid en anglais) est une mise en page qui permet de créer des mises en page en divisant l'espace d'affichage en régions.

La grille permet d'aligner des éléments sous forme de colonnes et de lignes mais à la différence des tableaux, la grille n'a pas de structure de contenu.

8 - Les Media Queries

Types de médias

- Écran : cible les écrans de bureau, ordinateurs portables, tablettes et smartphones.
- Impression : cible les impressions papier, les aperçus d'impression et les pages PDF.
- Projection : cible les présentations projetées à partir d'un ordinateur.
- Handheld : cible les appareils portables avec un petit écran, comme les smartphones et les lecteurs de livres électroniques.
- TV : cible les téléviseurs et les moniteurs de télévision.
- All : applique les styles à tous les types de médias.

Breakpoints 1/2

- Les breakpoints sont des points de rupture dans la mise en page d'une page web qui permettent de modifier le style en fonction de la taille de l'écran du dispositif sur lequel la page est affichée.
- Ils sont définis en utilisant des valeurs de taille d'écran spécifiques, telles que la largeur de l'écran en pixels ou en pourcentage.

Breakpoints 2/2

Desktop



@media screen and
(min-width: 1024px)
{...}

Tablet



@media screen and
(min-width: 768px) and
(max-width: 1023px)
{...}

Smartphone



@media screen and
(max-width: 767px)
{...}

Syntaxe des requêtes

```
@media type and (caractéristique: valeur) {  
  /* styles à appliquer */  
}
```

@media : indique qu'une requête de média est en cours.

type : identifiant pour le type de média que la requête doit cibler, tel que screen, print, speech, etc.

caractéristique : largeur de l'écran (width), la résolution (résolution), l'orientation (orientation), etc.

valeur : C'est la valeur à laquelle la caractéristique doit être comparée.

Exemple : cible les écrans dont la largeur est supérieure à 768 pixels :

```
@media screen and (min-width: 768px) {  
  /* styles à appliquer */  
}
```

9 - Responsive Design

Qu'est-ce que **le Responsive Design** ?

- Le Responsive Design est une approche de conception de sites web qui vise à fournir une expérience utilisateur optimale sur tous les appareils.
- On adapte le contenu et la disposition en fonction de la taille de l'écran.
- Il permet de s'adapter à une large gamme d'appareils.
- Il améliore l'expérience utilisateur en rendant le contenu plus facile à lire et à naviguer, en optimisant les temps de chargement et en réduisant le taux de rebond.

Adaptation des contenus

- Grilles CSS : vous pouvez utiliser des grilles CSS pour définir des mises en page flexibles qui s'adaptent automatiquement à la taille de l'écran.
- Médias queries : vous pouvez utiliser les médias queries pour cibler des dispositifs spécifiques et appliquer des styles différents en fonction de la taille de l'écran.
- Images adaptatives : les images adaptatives sont des images qui s'adaptent à la taille de l'écran, ce qui permet d'optimiser les temps de chargement et d'améliorer l'expérience utilisateur.
- Polices adaptatives : les polices adaptatives sont des polices qui s'adaptent à la taille de l'écran, ce qui permet d'améliorer la lisibilité du texte sur les petits écrans.

Adaptation du positionnement

- Positionnement relatif : pour positionner un élément par rapport à sa position normale dans le flux de la page.
- Positionnement absolu : pour positionner un élément par rapport à son conteneur.
- Positionnement fixe : pour positionner un élément par rapport à la fenêtre du navigateur. Cet élément restera positionné même si l'utilisateur fait défiler la page.
- Grilles CSS : pour définir des mises en page flexibles qui s'adaptent automatiquement à la taille de l'écran.
- Médias queries : pour cibler des dispositifs spécifiques.

Adaptation de la navigation

- Menu déroulant : pour les écrans plus petits où l'espace est limité.
- Menu en accordéon : permet aux utilisateurs de voir rapidement les options de navigation disponibles tout en conservant l'espace disponible sur l'écran. Particulièrement utile sur les appareils mobiles où l'espace est limité.
- Menu latéral : pour les écrans plus larges, s'affiche à gauche ou à droite de l'écran.
- Menu fixe : vous pouvez fixer la position de votre menu de navigation en haut de la page pour qu'il reste visible même lors du défilement de la page.
- Utilisation des icônes : en remplaçant les options de navigation par des icônes, vous pouvez économiser de l'espace sur l'écran.
- Utilisation des médias queries : vous pouvez utiliser les médias queries pour masquer certaines options de navigation sur des écrans plus petits ou pour réorganiser l'ordre des éléments de navigation.

Qu'est-ce que **la stratégie du Mobile First** ?

- La stratégie du "Mobile First" consiste à commencer par la création d'une version optimisée pour les appareils mobiles.
- La conception est ensuite étendue pour les appareils de bureau et les tablettes.
- La plupart des utilisateurs naviguent sur le web à partir de leurs appareils mobiles.
- Google privilégie et met en avant les sites mobile-friendly dans les résultats de recherche.

10 - Frameworks et librairies

Modernizr

- Modernizr est une bibliothèque JavaScript qui vous permet de détecter les fonctionnalités HTML5 et CSS3 supportées par le navigateur de l'utilisateur.
- C'est utile lorsque vous voulez fournir des alternatives gracieuses pour les navigateurs qui ne supportent pas certaines fonctionnalités.

```
<script src="modernizr.js"></script>
```

Librairies de substitution

- C'est un ensemble de code qui est utilisé pour fournir des fonctionnalités modernes dans les navigateurs plus anciens qui ne les prennent pas en charge nativement.
- Elles sont souvent utilisées pour combler le fossé entre les différents navigateurs et leurs niveaux de compatibilité avec les normes web
- **Less** : Un préprocesseur CSS qui étend le CSS en ajoutant des fonctionnalités comme les variables, les mixins et les fonctions pour faciliter le développement de styles plus complexes et maintenables.
- **Sass** : Un préprocesseur CSS robuste qui offre des fonctionnalités avancées telles que les variables, les fonctions, les mixins et l'héritage, permettant aux développeurs d'écrire des styles plus réutilisables et modulaires.
- **CSS3 PIE** (Progressive Internet Explorer) : Une petite bibliothèque JavaScript qui permet aux versions plus anciennes d'Internet Explorer (6-9) de supporter certaines des fonctionnalités les plus courantes de CSS3 qui ne sont pas nativement prises en charge par ces navigateurs.

Frameworks

- **Bootstrap** offre une grille réactive, des composants pré-stylisés comme les modales et les carrousels, et des plugins JavaScript pour ajouter des fonctionnalités. Bootstrap vise à être mobile-first et est connu pour aider à créer rapidement des prototypes.
- **HTML5 Boilerplate** : C'est une base de départ professionnelle pour la construction de sites web robustes et adaptables. Il comprend une base solide de HTML5, CSS et JavaScript, avec des meilleures pratiques intégrées et une prise en charge du navigateur utile.

Outils de vérification HTML/CSS

- Valideur HTML du W3C (<https://validator.w3.org/>) : Il vérifie la conformité de votre code HTML aux normes actuelles du W3C.
- Valideur CSS du W3C (<https://jigsaw.w3.org/css-validator/>) : Il fait la même chose que le vérificateur HTML, mais pour le CSS.

11 - Bonnes pratiques

Bonnes pratiques 1/2

- Utilisez toujours des noms de classe et d'ID sémantiques et descriptifs.
- Organisez votre CSS en sections logiques et commentez-le pour faciliter la compréhension.
- Utilisez les balises HTML appropriées pour leur signification sémantique, pas seulement pour leur apparence.
- Utilisez les media queries pour rendre votre site web responsive.
- Utilisez toujours alt texte pour les images pour l'accessibilité.

Bonnes pratiques 2/2

- Essayez de maintenir votre code DRY (Don't Repeat Yourself) : si vous vous trouvez en train de répéter le même CSS plusieurs fois, pensez à utiliser une classe.
- Utilisez des feuilles de style pour réinitialiser/normaliser pour résoudre les incohérences de style entre les navigateurs.

FORMATION

HTML 5, CSS 3, Responsive

25/09/2023

Votre formateur : Quentin BIDOLET

font-family

- - Famille de la police de caractères
 - Valeurs possibles
 - nom_de_police : Arial, Times New Roman, Comic Sans Ms...
 - serif : police avec empattement (Times New Roman, Garamond...)
 - sans-serif : police sans empattement (Arial, Helvetica, Gill Sans MT)
 - monospace : police non proportionnelle (Courier New, Consolas...)
 - cursive : police de type cursive
 - fantasy : police de type fantaisie (Comic Sans MS)
 - inherit : hérite de son parent
- Exemple

font-family: arial, helvetica, sans-serif;

font-size

- Taille de la police de caractère
- Valeurs possibles :

Valeurs numériques : 10px, 12pt...

Valeurs prédéfinies : xx-small, x-small, small, medium, x-large, xx-large

Valeurs relatives : dépend de l'élément courant : smaller, larger, 50%...

inherit : hérite de son parent

- Exemples :

font-size: smaller;

font-size: 12px;

font-style

- Style de la police de caractère
- Valeurs possibles :
 - normal
 - italic : caractères en italique
 - inherit : hérite de son parent
- Exemple :
`font-style: italic`

font-weight

- Mise en gras du texte
- Valeurs possibles :
 - normal
 - bold : caractère gras
 - bolder : la graisse augmente
 - lighter : la graisse diminue
 - Valeur_numérique : de 100 à 900 par pas de 100. 400 = normal, 700 = bold
 - inherit : hérite de son parent
- Exemple
 - font-weight: bolder;