

Tutoriel

Créer et mettre en forme sa page personnelle

Lucile Laulin

2018

Table des matières

1	Introduction	2
2	Base du code	2
	2.1 Le code HTML	2
	2.2 Le code CSS	3
3	Comprendre et utiliser le HTML	4
	3.1 Structure	4
	3.2 Le texte	5
	3.3 Liens	5
	3.3.1 Liens usuels	5
	3.3.2 Un autre type de liens	6
	3.4 Listes	6
	3.5 Éléments de code supplémentaires.	7
4	Comprendre et utiliser le CSS	8
	4.1 Fonctionnement	8
	4.1.1 Structure	8
	4.1.2 Éléments modifiables.	8
	4.2 Positionnement	9
	4.2.1 Marges et espacement	9
	4.2.2 Autres techniques.	10
	4.3 Style du texte	10
	4.3.1 Cas général	10
	4.3.2 Cas des liens	11
	4.4 Style des boîtes	11
	4.5 Images et arrière plan	12
	4.6 Le menu de navigation.	12
5	Un exemple	14
	5.1 Mise en place	14
	5.2 Le contenu	15
	5.3 La mise en forme	16
	5.4 Conclusion	19

1 Introduction

Ce tutoriel a pour objectif de vous aider à créer une page personnelle originale. L'idée est de voir comment passer de la figure 1 à quelque chose d'un peu plus complet comme la figure 2 et enfin à une mise en forme comme la figure 3 si vous le souhaitez.



Figure 1

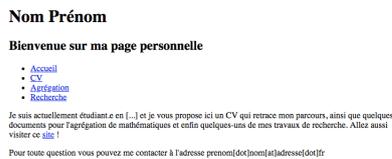


Figure 2



Figure 3

Pour cela nous allons d'abord voir les bases du code HTML et CSS. Pour coder dans ces deux langages il existe plusieurs possibilités :

- Utiliser un éditeur de texte, comme par exemple **Sublime Text 3**. Cela permet de coder simplement, l'éditeur connaît beaucoup de langages et mettra les mots clés en valeur. Pour l'aperçu vous n'aurez plus qu'à ouvrir vos fichiers `.html` dans votre navigateur internet préféré.
- Utiliser un logiciel "tout en un" comme **BlueGriffon**. C'est un logiciel de création de pages web dans lequel on a directement un aperçu de ce qu'on fait. On peut utiliser alternativement l'outil de code ou bien les éléments proposés par le logiciel. On modifie alors à la fois le contenu et l'apparence.

Personnellement j'ai commencé avec BlueGriffon pour me familiariser avec les langages, mais je l'ai vite trouvé limité si l'on fait de la mise en forme un peu poussée.

2 Base du code

2.1 Le code HTML

Le HTML, pour *HyperText Markup Language* est un langage permettant de définir les différents éléments d'une "page" qu'on observe avec un navigateur internet : les titres, le menu de navigation, les liens... C'est avec ce langage que l'on donne du contenu à une page web.

Code pour la figure 1

```
<!DOCTYPE html>
<html>
<head>
  <title>Page perso !</title>
</head>
<body>
<h1>Voici le titre de ma page !</h1>
<p>Bonjour et bienvenue sur ma page.</p>
</body>
</html>
```

Les éléments entre `<` et `>` s'appellent des balises : ce sont elles qui disent au navigateur ce qu'il doit faire (nous verrons ensuite que l'on peut lui dire comment effectuer l'action). Par exemple ici `<h1>...</h1>` correspond au titre de la page et `<p>...</p>` au contenu d'un paragraphe. Nous verrons cela plus en détails dans la partie 3.

Il faut savoir que malgré ce que vous pourriez donner comme indication de code, l'affichage peut différer d'un navigateur à l'autre !

2.2 Le code CSS

Le CSS, pour *Cascading Style Sheets* est un langage utilisé pour mettre en forme une page web. En bref, c'est ce qui va rendre votre page web plus "jolie", ce qui sert à la mettre en forme. Il y a trois manières d'ajouter du style à votre page :

- en ajoutant le code dans vos balises HTML à l'aide de `style="..."`,
- en ajoutant au début de votre fichier `.html` des balises `<style>...</style>`,
- en créant un fichier `.css` à part et en l'appelant dans votre fichier `.html`.

Style dans la balise paragraphe

```

<!DOCTYPE html>
<html>
<head></head>
<body>
<h1>Voici le titre de ma page !</h1>
<p style="background-color:blue;border:2px solid black;color:white;
font-size:1.3em;padding:0.5em;">
Ce paragraphe est style !</p>
</body>
</html>

```

La technique ci-dessus va modifier le paragraphe en cours, mais pas les suivants. Avec les deux techniques suivantes la modification de style s'appliquera à tous les éléments de la catégorie. On utilisera plutôt l'une ou l'autre des techniques selon la situation. Ainsi on choisira la deuxième si on a peu de style à modifier, et la troisième sinon.

Style en début de code

```

<!DOCTYPE html>
<html>
<style type="text/css">
  p {
    background-color:blue;
    border:2px solid black;
    color:white;
    font-size:1.3em;
    padding:0.5em;
  }
</style>
<head></head>
<body>
<p> Ce paragraphe est style !</p>
</body>
</html>

```

Avec la seconde technique, si l'on veut modifier la mise en forme sur toutes les pages, il faudrait modifier l'en-tête `<style>` pour chacune des pages. La troisième technique permet, en modifiant un seul fichier, de modifier le style sur l'ensemble des pages.

Style dans un autre fichier, le .css

```
p {
  background-color : blue;
  border : 2px solid black;
  color : white;
  font-size : 1.3em;
  padding : 0.5em;
}
```

Style dans un autre fichier, le .html

```
<!DOCTYPE html>
<html>
<link type="text/css" rel="stylesheet" href="
style.css"/>
<head></head>
<body>
<p> Ce paragraphe est style !</p>
</body>
</html>
```

Ces trois codes donnent le même résultat, présenté en figure 4.



Figure 4

Nous verrons ceci plus en détails dans la partie 4.

3 Comprendre et utiliser le HTML

3.1 Structure

Le langage HTML donne au navigateur la structure de la page web. Dans les exemples précédents les balises sont en **bleu**, nous allons voir les principales balises et leur signification.

- `<html>...</html>` : ce qui se trouve entre les balises est en html,
- `<head>...</head>` : contient les informations données au navigateur (ce n'est pas montré sur la page), souvent accompagné de `<title>...</title>` qui correspond au titre (ce qu'on voit dans l'onglet),
- `<body>...</body>` : ce qui devrait être montré dans le navigateur,
- `<h1>...</h1>` : le titre principal, il y a aussi les sous-titres `<h2>` jusqu'à `<h6>`,
- `<p>...</p>` : délimite un paragraphe, deux paragraphes sont présentés sur des lignes différentes,
- `<!--...-->` : délimite une zone de commentaire.

Ci-dessus sont présentés les éléments qui permettent de structurer votre page. Voici un exemple de structure en HTML.

```
<!DOCTYPE html>
<html>
<head>
  <title>Ma page personnelle</title>
</head>
<body>
  <h1> Un premier titre</h1>
  <h2> Un sous titre</h2>
  <p>Voici un paragraphe.</p> <!--un commentaire-->
  <h2>Un autre sous-titre</h2>
  <p>Et un autre paragraphe.</p>
</body>
</html>
```



Un premier titre

Un sous titre

Voici un paragraphe.

Un autre sous-titre

Et un autre paragraphe.

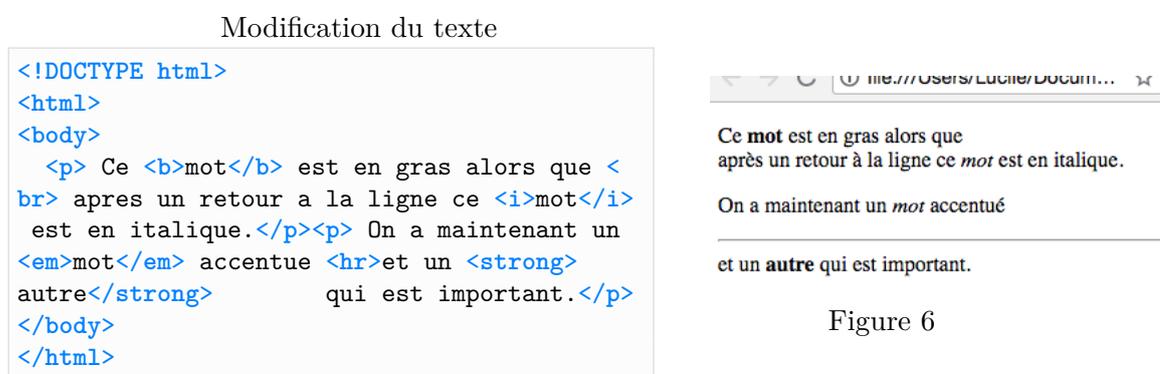
Figure 5

3.2 Le texte

La plupart de votre texte se trouvera dans des paragraphes ou des titres. Il est possible de structurer et personnaliser votre texte grâce aux éléments suivants :

- `...` fera apparaître le texte en gras,
- `<i>...</i>` fera apparaître le texte en italique,
- `...` indique au navigateur que le texte est important, par défaut cela le fera souvent apparaître en gras,
- `...` va accentuer le mot, par défaut cela le fera souvent apparaître en italique,
- `
` effectue un retour à la ligne,
- `<hr>` effectue un retour à la ligne et ajoute une ligne horizontale.

De même que pour les paragraphes, des titres différents seront sur des lignes différentes. De plus, comme en \LaTeX , seul un espace entre les mots sera pris en compte. Voici un exemple de ces différentes modifications.



3.3 Liens

3.3.1 Liens usuels

Les liens sont définis à l'aide de `...`. C'est grâce au `href=` que l'on indique ce vers quoi on veut envoyer. Vous allez surtout utiliser deux types de liens différents :

- les liens **internes**, qui envoient vers une autre page de votre site. Il vous suffit d'indiquer le nom de la page (et le dossier dans lequel elle se trouve si elle n'est pas dans le même que votre page actuelle), par exemple `Page` pour envoyer vers une page ou `Mon fichier` pour envoyer vers le *fichier* qui se trouve dans le *dossier*.
- les liens **externes**, qui envoient vers un autre site. Dans ce cas il vous suffit d'indiquer l'adresse de la page du site vers laquelle vous souhaitez renvoyer, par exemple `Site de l'ENS Rennes`.

Lorsque vous envoyez vers une de vos pages ou l'un de vos fichiers, il faut donc faire attention à sa position par rapport à la page dont vous partez. Par exemple avec l'organisation de la figure 7



Figure 7

- si vous êtes sur la page "enfant.html" du dossier Enfant et que vous voulez envoyer vers la page grand-parent.html du dossier Grand-parent alors le début du lien sera ``, le ../../ indique que vous souhaitez remonter de deux dossiers.
- si vous êtes sur la page "grand-parent.html" du dossier Grand-parent et que vous voulez envoyer vers la page parent.html du dossier Parent alors le début du lien sera ``, le Parent/ indique que vous souhaitez descendre d'un dossier dans le dossier Parent.
- enfin si vous êtes sur la page "grand-parent.html" du dossier Grand-parent et que vous voulez envoyer vers le fichier Fichier.pdf alors le début du lien sera `` car les deux sont dans le même dossier.

Enfin en ajoutant `target="_blank"`, vous enverrez vers le lien souhaitez en ouvrant une nouvelle fenêtre. Ainsi à l'aide du code

```
<a href="www.ens-rennes.fr" target="_blank">Site de l'ENS Rennes</a>
```

on ouvrira le site de l'ENS Rennes dans une autre fenêtre.

3.3.2 Un autre type de liens

Il existe un troisième type de liens qui renvoie vers une certaine partie d'une même page. Pour cela on va donner un nom à la partie (généralement un titre) vers laquelle on veut renvoyer à l'aide de `id="identifiant"`. L'exemple ci-dessous montre comment cela fonctionne pour une page divisée en deux parties.

```
<h1 id="debut">Bienvenue sur ma page</h1>
<a href="#parcours">Parcours</a>
<a href="#cv">CV</a>
<h2 id="parcours">Parcours</h2>
<p>Voici mon parcours, avec beaucoup de texte [...]</p>
<h2 id="cv">Curriculum Vitae</h2>
<p>Je vous donne ici avec beaucoup de details mon CV [...]</p>
<p><a href="#debut">Retour en haut de la page</a></p>
```

Ces liens sont intéressants à condition que la page soit très longue, sinon on a déjà un aperçu direct de tout ce qu'il y a sur la page.

3.4 Listes

Il y a plusieurs choses à dire sur les listes, mais nous allons ici nous concentrer sur ce qui sera utile pour créer un menu de navigation. Un élément d'une liste est placé entre les balises `...` (li pour list item). Nous allons nous intéresser aux listes non-ordonnées, qui sont entourées par les balises `...` (ul pour unordered list). Ce type de liste permet de créer un menu, avec des liens distincts les uns des autres. Avec le code précédent les deux liens du début étaient collés, grâce aux listes ils sont distincts et il sera possible de leur donner du style.

```
<h1 id="debut">Bienvenue sur ma page</h1>
<a href="#parcours">Parcours</a>
<a href="#cv">CV</a>
<h2 id="parcours">Parcours</h2>
<p>Voici mon parcours, avec beaucoup de texte [...]</p>
<h2 id="cv">Curriculum Vitae</h2>
<p>Je vous donne ici avec beaucoup de details mon CV [...]</p>
```

Bienvenue sur ma page

[Parcours CV](#)

Parcours

Voici mon parcours, avec beaucoup de texte [...]

Curriculum Vitae

Je vous donne ici avec beaucoup de details mon CV [...]

Figure 8

```

<h1 id="debut">Bienvenue sur ma page</h1>
<ul>
  <li><a href="#parcours">Parcours</a></li>
  <li><a href="#cv">CV</a>Parcours</li>
</ul>
<h2 id="parcours"></h2>
<p>Voici mon parcours, avec beaucoup de texte [...]</p>
<h2 id="cv">Curriculum Vitae</h2>
<p>Je vous donne ici avec beaucoup de details mon CV [...]</p>

```

Bienvenue sur ma page

- [Parcours](#)
- [CV](#)

Pacours

Voici mon parcours, avec beaucoup de texte [...]

Curriculum Vitae

Je vous donne ici avec beaucoup de details mon CV [...]

Figure 9

Nous verrons dans la partie 4 comment mettre en forme les listes afin d’avoir un beau menu.

3.5 Éléments de code supplémentaires

Il nous reste encore éléments de code utiles à voir.

L’attribut `id` qu’on a déjà vu en 3.3, cela permet d’identifier n’importe quel élément, ce qui peut être utile lorsqu’on utilise du CSS pour mettre en forme uniquement cet élément. L’intérêt est qu’il soit donné à un unique élément.

L’attribut `class` qu’on peut donner aussi à n’importe quel élément, mais cette fois plusieurs éléments peuvent avoir la même `class`. C’est encore intéressant avec du CSS afin de modifier plusieurs éléments en même temps. Il est possible d’avoir plusieurs `class`.

La balise `<div>...</div>` qui permet de regrouper plusieurs éléments dans une même boîte. Le contenu de différentes `div` commencera sur une nouvelle ligne. Là encore l’intérêt vient lorsqu’on utilise du CSS, cela permet de modifier par exemple la taille occupée par le groupe d’éléments. C’est aussi utile pour s’y retrouver dans le code.

La balise `...` qui permet, là encore, de contrôler l’apparence d’un ou plusieurs éléments au milieu d’autres, mais sans revenir à la ligne.

La balise `<nav>...</nav>` dans laquelle on trouvera des liens permettant d’accéder à d’autres pages du site ou à différents contenus de la page.

Code avec différents éléments

```

<!DOCTYPE html>
<html>
<body>
  <div id="haut">
    <h1>Bienvenue sur ma page</h1> <!-- Titre -->
    <nav> <!-- Menu -->
    <ul>
      <li><a href="#parcours">Parcours</a></li>
      <li><a href="cv.html">CV</a></li>
      <li><a href="fichier.pdf">Un fichier</a>Parcours</li>
    </ul>
    </nav>
  </div>
  <div id="contenu">
    <p id="intro">Ici je fais une introduction.</p> <!-- Paragraphe -->
    <h2 id="parcours">Parcours</h2>
    <p id="para1" class="normal">Voici mon parcours, avec beaucoup de texte !
  </p>
  </div>
</body>
</html>

```

4 Comprendre et utiliser le CSS

4.1 Fonctionnement

4.1.1 Structure

Le CSS fonctionne comme si chaque élément était dans une boîte, on va alors modifier les paramètres de cette boîte. La balise `div` permet d'avoir plusieurs éléments dans une même boîte, on peut alors modifier chaque élément comme `p` ou `h1` seul ou tous en même temps. La présentation est celle des éléments suivants, par exemple 1 modifie la couleur et la taille du texte des paragraphes. C'est la dernière modification qui est prise en compte, ainsi écrire 2 revient à écrire 3.



Enfin en CSS les commentaires sont entre `/* ...*/`.

4.1.2 Éléments modifiables

On peut modifier énormément de choses grâce au CSS. La liste suivante est un échantillon assez large des éléments modifiables et explique rapidement comment procéder.

- Les éléments usuels du langage HTML, à savoir `p`, `h1...`, `h6`, `header`, `a`, `span`, `ul`, `li`, `html`, `em`, `nav`, sont modifiables selon les exemples ci-dessus. On les appelle **tag**.
- On peut aussi modifier les éléments qui ont un `id="identifiant"` avec `#identifiant` à la place du `p`.
- On peut aussi modifier les éléments qui sont dans un autre type d'éléments, par exemple les éléments d'une liste dans la navigation avec `nav>ul>li` ou `nav ul li`.
- Enfin on peut modifier les éléments qui ont une `class="classe"` soit avec `.classe` à la place du `p` pour tous les modifier, `p.classe` pour modifier les paragraphes qui ont la classe `"classe"` ou encore `.classe p` pour modifier les paragraphes qui sont dans un élément ayant la classe `"classe"`.

Voici quelques exemples de modification de style.



Figure 10

Figure 11

Figure 12

De même que c'est la dernière modification qui est prise en compte, dans le cas des `class` c'est la plus générale qui l'est. Il faut alors préciser que l'on veut modifier tel(s) élément(s) de la classe. En bref, le style d'un `#identifiant` est prioritaire sur celui d'une `.classe`

qui est lui même prioritaire sur celui d'un tag comme `p`. Il est possible d'ajouter un `!important` après une modification pour qu'elle soit considérée plus importante que les autres appliquées au même éléments.

4.2 Positionnement

4.2.1 Marges et espacement

On va ici revenir sur le fait que "chaque élément est dans une boîte". On s'intéresse aux marges intérieures, ou `padding`, et aux marges extérieures, ou `margin` d'un élément. Le plus simple pour comprendre comment cela fonctionne est de regarder plusieurs exemples. On considère le code HTML suivant et on observe en figure 13 les marges intérieures et extérieures entre les différents éléments.

```
<!DOCTYPE html>
<html>
<style>
  /* On va le completer de differentes manieres */
</style>
<body>
  <h1>Le titre</h1>
  <p>Un paragraphe</p>
  <p>Et un autre</p>
</body>
</html>
```

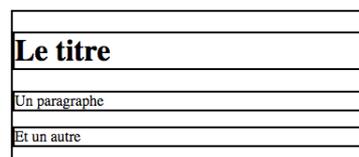
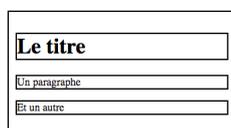


Figure 13

Nous allons maintenant modifier ces valeurs – nous verrons prochainement la signification de `px` et comment faire apparaître les bordures des boîtes. En figure 14 les marges extérieures sont trois fois plus importantes que les marges intérieures pour le contenu, alors qu'en figure 15 les marges intérieures pour le titre sont deux fois plus grandes que les extérieures.

```
body {
  margin : 30px;
  padding : 10px;
}
```



```
h1 {
  margin : 6px;
  padding : 12px;
}
```

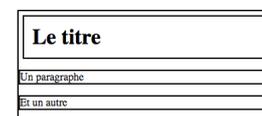


Figure 15

Figure 14

Il est possible de modifier plus précisément ces marges, soit avec `margin / padding - top / bottom / left / right` soit en utilisant la syntaxe dont le résultat est présenté en figure 16. On peut alors voir que le navigateur effectue un retour à la ligne dans les paragraphes pour respecter les marges intérieures. D'autres par les `margin` vont s'ajouter, ainsi l'écart entre le titre et le premier paragraphe est de $6\text{px} + 20\text{px} = 26\text{px}$ et celui entre les deux paragraphes de $20\text{px} + 0\text{px} = 20\text{px}$ car on n'a pas modifié `margin-bottom`. Enfin pour centrer un élément, il faut régler les marges droite et gauche sur `auto`.

```
h1 {
  margin : 6px 12px; /* haut/bas droite/gauche */
  padding-top : 20px;
  padding-bottom : 40px;
}
p {
  margin-left : 13px;
  margin-top : 20px;
  padding : 10px 280px 40px 60px;
  /* haut droite bas gauche */
}
```

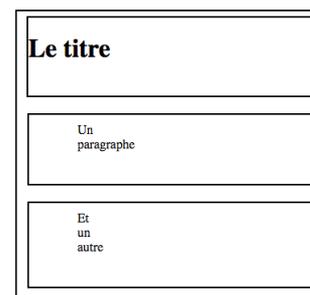


Figure 16

4.2.2 Autres techniques

Pour positionner des éléments, on utilisera majoritairement les `margin` et `padding` vu à en 4.2.1. C'est souvent grâce à la balise `div` qu'on positionnera les éléments, en leur donnant une `class` ou un `id`.

Il existe cependant d'autres techniques de positionnement dont on peut faire usage. Pour cela, il est important de faire la différence entre des éléments `inline` et les éléments `block`. Les premiers n'occupent que la largeur nécessaire et ne provoquent pas de retour à la ligne, comme par exemple les balises `a` ou `span`. Les seconds occupent toute la largeur disponible, deux éléments `block` sont par défaut disposés l'un sous l'autre, comme par exemple les balises `div`, `h1` ou `p`.

La commande `display: block / inline / inline-block` permet de changer le `style` d'un élément et donc de donner un rendu différent de celui de base. Elle sera notamment utile lorsqu'on mettra en forme le menu de navigation.

La commande `float: left / right / none` permet de positionner des éléments côte à côte, notamment pour créer des colonnes. Elle implique l'utilisation d'éléments en `block` et forcera cette disposition même si le `display` indique autre chose. Pour que des éléments ne soient pas concernés par le flottement on utilise `clear: left / right / both`.

Enfin la commande `position: absolute / relative / fixed` permet aussi de changer la position mais nous ne nous attarderons pas dessus car elle est plus compliquée et surtout risquée à utiliser puisqu'elle peut provoquer des chevauchements.

4.3 Style du texte

4.3.1 Cas général

Nous allons maintenant voir comment modifier le texte grâce au CSS.

Famille

Modification à l'aide de `font-family`, il faut alors spécifier la police – Times, Helvetica... – et le type – serif, sans-serif ou monospace si nécessaire.

Taille

Modification à l'aide de `font-size`, il faut donner une valeur numérique

- en pixel `px`, qui sera donc toujours la même peu importe sur quoi on affiche la page,
- en pourcentages `%`, qui va dépendre de la taille par défaut du navigateur (généralement 12 ou 16px),
- en ems `em`, qui est équivalent à la largeur de la lettre `m`.

On a l'équivalence `12(ou 16)px=100%=1em`.

Épaisseur

Modification à l'aide de `font-weight` prenant les valeurs `lighter / normal / bold / bolder` ou une valeur numérique entre 100, 200,..., 900.

Style

Modification à l'aide de `font-style` prenant les valeurs `normal / italic / oblique`.

Transformation

Modification à l'aide de `text-transform` prenant les valeurs `uppercase / lowercase`.

Décoration

Modification à l'aide de `text-decoration` prenant les valeurs `none / underline`.

Alignement

Modification à l'aide de `text-align` prenant les valeurs `left / right / center / justify`.

Couleur

Modification à l'aide de `color` prenant comme valeur une couleur au format HEX comme `#FF3333`, RGB comme `rgb(255, 51, 51)` ou simplement `red`, `blue`... pour les couleurs usuelles.

Espacement des lettres

Modification à l'aide de `letter-spacing` prenant une valeur numérique. C'est utile dans les menus ou les titres pour apporter une meilleure visibilité.

4.3.2 Cas des liens

Voici quelques commandes qui permettent de modifier spécifiquement les liens.

- La commande `a:link` permet de modifier l'apparence d'un lien qui n'a pas encore été visité.
- La commande `a:visited` permet de modifier l'apparence d'un lien qui a déjà été visité.
- La commande `a:hover` permet de modifier l'apparence d'un lien lorsqu'on place la souris dessus.

Nous verrons l'utilisation de ces commandes dans la partie 4.6 sur le menu de navigation.

4.4 Style des boîtes

Dimensions

Par défaut la taille d'une boîte est juste celle qu'il faut pour entourer son contenu. On peut néanmoins modifier manuellement ces dimensions à l'aide des commandes `width` et `height` auxquelles il faut donner une valeur numérique selon les mêmes modalités que pour la taille du texte, soit

- en pixel `px` la dimension sera fixée peu importe le navigateur,
- en pourcentages `%` elle sera relative à la largeur de la fenêtre qui l'affiche,
- en ems `em` elle dépendra de la taille du texte qu'elle contient.

Il faut faire attention aux dimensions des boîtes car en les modifiant le contenu peut dépasser de la boîte.

Couleur

Pour modifier la couleur d'une boîte (donc la couleur autour du contenu) il faut utiliser la commande `background` ou `background-color` qui prend en valeur une couleur.

Bordures

Nous avons vu ces bordures pour la première fois dans en 4.2.1 afin de comprendre les marges. Nous allons maintenant nous intéresser à leurs propriétés et leur mise en forme.

– Épaisseur

Modification à l'aide de `border-width` prenant une valeur numérique ou les valeurs `thin` / `medium` / `thick`. Elle fonctionne comme `margin` et `padding`.

– Style

Modification à l'aide de `border-style` prenant les valeurs `solid` / `dotted` / `dashed` / `double` / `groove` / `ridge` / `inset` / `outset`.

– Couleur

Modification à l'aide de `border-color` prenant une couleur.

– Coins

Modification de l'arrondi des coins à l'aide de `border-radius` et d'une valeur numérique, 0 étant la valeur pour un angle droit.

– Raccourcis

Enfin la commande `border` : épaisseur style couleur permet de modifier rapidement une bordure. Il faut toujours préciser l'épaisseur, le style et la couleur.

Le résultat du code avec différentes syntaxes ci-dessous, est présenté en figure 17.

```
h1 {
  border : 4px solid #FF6666;
}
h2 {
  border-color : #0080FF;
  border-width : 2px;
  border-style : solid dashed;
  /* haut/bas droite/gauche */
  border-radius : 5px 15px;
  /* haut-gauche bas-droite
    haut-droite bas-gauche */
  padding : 20px;
}
p {
  border-color : #CCC666;
  border-style : dotted;
  border-width : 10px 2px 5px 0px;
  /* haut droite bas gauche */
  border-radius : 3px;
  padding : 10px;
}
```

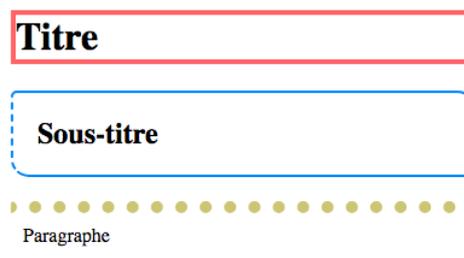


Figure 17

4.5 Images et arrière plan**Généralités**

Il est possible d'ajouter des images et de contrôler leur taille et leur position. Le code html correspondant à ajouter là où elle doit apparaître est le suivant.

```

```

Pour gérer les dimensions et la position on utilise ensuite les `classes` auxquelles on donne les paramètres vu précédemment comme `width`, `height`, `margin`...

Arrière plan

A l'aide de la commande `background` ou `background-image` on peut ajouter une image d'arrière plan avec le code suivant. Il y a plus d'options disponibles mais généralement celles-là suffisent.

```
background-image : url("images/arriere-plan.jpg") /* votre image est
dans le dossier images */
background-image-repeat : repeat / no-repeat
/* OU BIEN - le raccourcis*/
background : #FFF999 url("images/arriere-plan.jpg") repeat /* couleur
si l'image n'est pas disponible + image + repetition */
```

4.6 Le menu de navigation

Ici nous allons utiliser plusieurs choses vues précédemment afin de créer un menu de navigation. Nous partons du code suivant, illustré par la figure 18.

```

<!DOCTYPE html>
<html>
<style>/* rempli ci-dessous */</style>
<body>
  <nav>
    <ul>
      <li><a href="#">Accueil</a></li>
      <li><a href="#">Parcours</a></li>
      <li><a href="#">Enseignement</a></li>
      <li><a href="#">Agregation</a></li>
      <li><a href="#">English</a></li>
    </ul>
  </nav>
</body>
</html>

```

- [Accueil](#)
- [Parcours](#)
- [Enseignement](#)
- [Agrégation](#)
- [English](#)

Figure 18

On utilise le code ci-dessous dont le résultat est présenté en figure 19. Il permet d'avoir un menu en ligne.

```

nav {
  background-color : #FFE5CC;
  text-align : center;
}
nav ul { /* liste dans la navigation */
  margin : 0 auto;
  padding : 0; /* enleve l'indentation de default */
}
nav ul li {
  display : inline-block; /* les elements de la liste sont presentes en ligne
  mais se comportent comme un block */
}
li a {
  display : block; /* les liens se comportent comme des block individuels,
  cela evite le chevauchement en cas de petite largeur d'ecran */
  padding : 1em;
  margin : 1em;
  text-decoration : none; /* enleve le soulignement habituel des liens */
  letter-spacing : 1px; /* pour plus de lisibilite */
  font-size : 0.8em;
  font-weight : 500;
  font-family : 'Arial';
  color : #FFF;
  background-color : orange;
  border-radius : 2px;
}

```



Figure 19

Pour un menu en colonne on utilisera plutôt le code ci-dessous, dans lequel seules les modifications apparaissent et dont le résultat est la figure 20.

```
html, body {
  height: 100%; /* utilise toute la hauteur de page
disponible */
}
nav {
  float : left;
  height : 100%;
}
nav ul { /* liste dans la navigation */
  margin : 0;
}
nav ul li {
  display : block; /* les elements de la liste sont un
block */
}
li a {
  margin : 1em;
  color : orange;
  border-width : 0px 0px 2px 0px;
  border-style : solid;
  border-color : orange;
  /*background-color : orange;*/
  border-radius : 0px;
}
```



Figure 20

Enfin il est possible de faire ressortir la page sur laquelle on se trouve actuellement pour cela il faut créer une classe ou un identifiant, par exemple `active` et modifier les paramètres. Dans le cas du menu en colonne on peut alors obtenir avec le code suivant les figures 21 ou 22.

Code CSS

```
#active { /* element ayant l'id "active" */
  background-color : orange;
  color : #FFE5CC;
}
/* OU BIEN */
.aktif a{ /* lien qui est dans un element ayant la classe "
aktif" */
  background-color : orange;
  color : #FFE5CC;
}
```

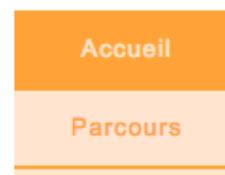


Figure 21

Code HTML

```
<li><a href="#" id="active">Accueil</a></li>
<!-- OU BIEN -->
<li class="aktif"><a href="#">Enseignement</a></li>
```



Figure 22

5 Un exemple

5.1 Mise en place

Commençons par créer deux fichiers : un fichier `index.html` qui sera la page d'accueil de votre page – il faut absolument le nommer comme ceci, et un fichier `style.css` qui contiendra la mise en forme et qui permettra de l'appliquer à toutes vos pages. Placez

les dans un même dossier `page personnelle`. Ensuite écrivons le code suivant dans le fichier `index.html`.

`index.html`

```

<!DOCTYPE html>
<html>
  <head>
    <title></title>
    <link href="style.css" rel="stylesheet" type="text/css"/>
  </head>
  <body>
</body>
</html>

```

5.2 Le contenu

Il faut maintenant réfléchir au contenu de votre page personnelle, comme le nombre de pages différentes, les informations que vous voulez faire apparaître, les fichiers que vous voulez proposer. On va utiliser les balises usuelles et en créer de nouvelles pour structurer la page. Par exemple supposons que vous vouliez quatre pages différentes : accueil, CV, agrégation, recherche. Vous mettez de plus un paragraphe introductif ainsi que quelques informations complémentaires. Votre code pourrait ressembler à ceci, et vous auriez dans un navigateur quelque chose comme la figure 23.

`index.html`

```

<!DOCTYPE html>
<html>
  <head>
    <title>Page personnelle</title>
    <link href="style.css" rel="stylesheet" type="text/css"/>
  </head>
  <body>
    <div id="logo">
      <h1>Nom Prenom</h1>
      <h2>Bienvenue sur ma page personnelle</h2>
    </div>
    <nav>
      <ul>
        <li><a href="index1.html">Accueil</a></li>
        <li><a href="CV.pdf">CV</a></li>
        <a href="agregation.html">Agregation</a></li>
        <a href="recherche.html">Recherche</a></li>
      </ul>
    </nav>
    <div id="contenu">
      <p>Je suis actuellement etudiant.e en [...] et je vous propose ici un CV qui retrace mon parcours, ainsi que quelques documents pour l'agregation de mathematiques et enfin quelques-uns de mes travaux de recherche. Allez aussi visiter ce site !</p>
      <div id="contact">
        <p>Pour toute question vous pouvez me contacter a l'adresse
          <span>prenom[dot]nom[at]adresse[dot]fr.</span></p>
      </div>
    </div>
  </body>
</html>

```

Nom Prénom

Bienvenue sur ma page personnelle

- [Accueil](#)
- [CV](#)
- [Agrégation](#)
- [Recherche](#)

Je suis actuellement étudiant.e en [...] et je vous propose ici un CV qui retrace mon parcours, ainsi que quelques documents pour l'agrégation de mathématiques et enfin quelques-uns de mes travaux de recherche. Allez aussi visiter ce [site](#) !

Pour toute question vous pouvez me contacter à l'adresse prenom[dot]nom[at]adresse[dot]fr

Figure 23

5.3 La mise en forme

Généralités

On commence par quelques modifications générales de l'apparence.

style.css 1

```
body {
  font-family : Arial; /* police de
  caracteres */
  background-color : #F5F5F5; /* pas
  tout a fait blanc */
  min-width : 520px; /* si l'affichage
  est moins large que 520px alors celui du
  corps sera de 520px */
  margin : 0; /* on enleve les marges
  par default */
  padding : 0; /* de meme */
}
```

Nom Prénom

Bienvenue sur ma page personnelle

- [Accueil](#)
- [CV](#)
- [Agrégation](#)
- [Recherche](#)

Je suis actuellement étudiant.e en [...] et je vous propose ici un CV qui retrace mon parcours, ainsi que quelques documents pour l'agrégation de mathématiques et enfin quelques-uns de mes travaux de recherche. Allez aussi visiter ce [site](#) !

Pour toute question vous pouvez me contacter à l'adresse prenom[dot]nom[at]adresse[dot]fr.

Figure 24

Logo

On ajoute à présent du style sur le logo.

style.css 2

```
#logo {
  float : left;
  margin : 1% 4%; /* marges adaptatives
  pour meilleure lisibilite */
  letter-spacing : 1.5px;
}
#logo h1 {
  color : #004C99;
}
#logo h2 {
  color : #0066CC;
  font-size : 1.5em;
}
```

Nom Prénom

Bienvenue sur ma page personnelle

- [Accueil](#)
- [CV](#)
- [Agrégation](#)
- [Recherche](#)

Je suis actuellement étudiant.e en [...] et je vous propose ici un CV qui retrace mon parcours, ainsi que quelques documents pour l'agrégation de mathématiques et enfin quelques-uns de mes travaux de recherche. Allez aussi visiter ce [site](#) !

Pour toute question vous pouvez me contacter à l'adresse prenom[dot]nom[at]adresse[dot]fr.

Figure 25

Contenu

Avec les modifications suivantes on fait ressortir le contenu. Le menu reste encore à droite du logo/titre.

style.css 3

```
#contenu
{
  background-color : #FFF; /* font different */
  color : #22303D;
  padding : 5% 8%; /* plus agreable a lire et au visuel */
  clear : both; /* pas concerne par les float donc retour a la ligne */
  border-top : 2px solid #DBE3EB; /* faire ressortir */
  border-bottom : 2px solid #DBE3EB;
}
span {
  font-weight : bold; /* faire ressortir */
  letter-spacing: 1px; /* plus de lisibilite */
  color : #003366;
}
a { /* style des liens */
  text-decoration : none; /* pas de soulignement */
  color : #0066CC;
}
a:hover {
  font-weight : bold; /* pour ressortir quand survole */
}
p {
  text-align : justify; /* plus beau car bien aligne */
}
```

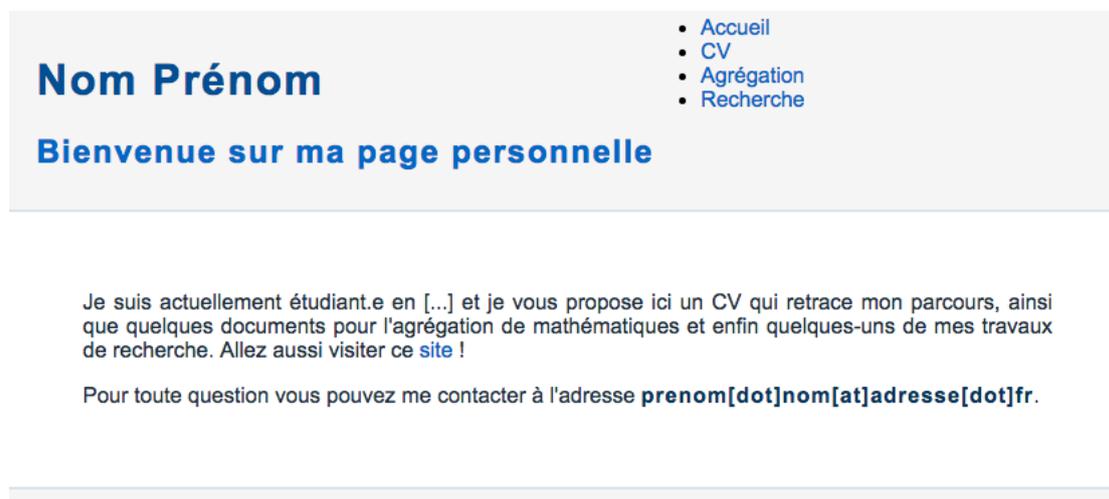


Figure 26

Menu de navigation

On positionne le menu à droite du logo. Ce positionnement va dépendre de la taille de la fenêtre, vous pouvez l'observer en diminuant la largeur.

style.css 4

```
nav {
  text-align : right; /* tout doit etre a droite */
}
```

```

nav ul { /* liste dans la navigation */
  margin : 1% 3% 0 0;
  padding : 0; /* enleve l'indentation de defaut */
  float : right; /* colonne a droite par rapport au logo */
}
nav ul li {
  display : inline-block; /* les elements de la liste sont presentes en ligne
mais se comportent comme un block */
}
nav li a {
  display : block; /* les liens se comportent comme des block individuels,
cela evite le chevauchement en cas de petite largeur d'ecran */
  padding : 1em;
  margin : 1em 1.1em;
  text-decoration : none; /* enleve le soulignement habituel des liens */
  letter-spacing : 1.3px; /* pour plus de lisibilite */
  font-size : 0.9em;
  color : #F5F5F5;
  background-color : #003366;
}
nav li a:hover { /* style des boutons de liens qu'on survole */
  background-color : #0066CC;
  color : #F5F5F5;
  font-weight : normal; /* car on a regle le cas general sur bold */
}
#active { /* element ayant l'id "active" */
  background-color : #0066CC;
  color : #F5F5F5;
}

```

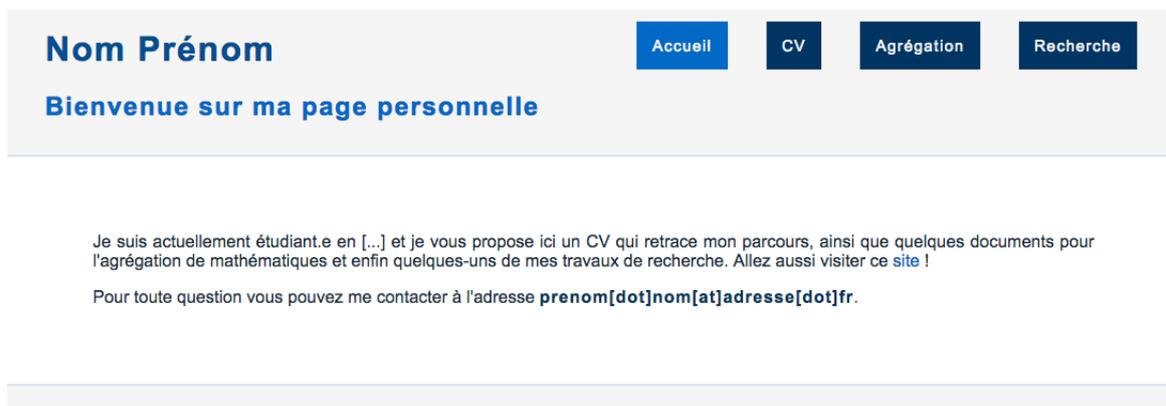


Figure 27

Décoration

Enfin on ajoute une image afin d'habiller un peu plus la page.

style.css 5

```

#banner { /* une banniere pour la decoration */
  background : #003366 url(geom.jpg) repeat center; /* couleur image
repetition */
  height : 100px; /* hauteur de la banniere */
  background-size : cover; /* prend la largeur de page */
  clear : both;
}

```

Le résultat final est donné en figure 28.

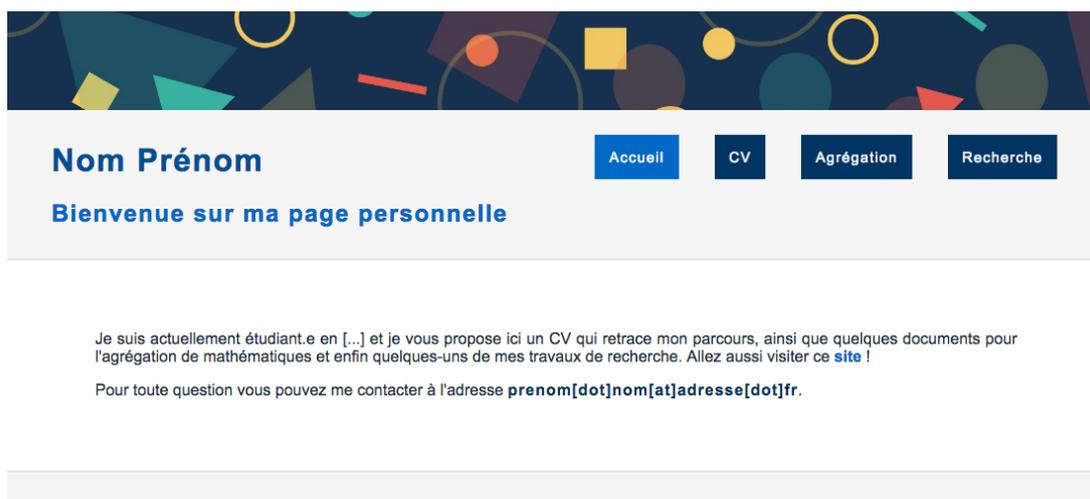


Figure 28

Pages supplémentaires

Pour ajouter une nouvelle page tout en gardant la même mise en forme, il suffit de copier le même en-tête de code HTML. Il ne faut pas oublier de changer l'identifiant du lien actif.

5.4 Conclusion

A travers cet exemple, nous avons vu une manière de créer et mettre en forme une page. Il existe énormément d'autres possibilités que vous pourrez découvrir en faisant varier tous les paramètres dont nous avons parlé : mettre le menu en colonne, en dessous ou au dessus du titre, une bannière ou pas, une image de fond... Bref à vous d'essayer et de trouver ce qui vous convient le mieux. Notez bien que ce tutoriel n'a pas vocation à être exhaustif, vous pourrez découvrir encore d'autres subtilités à l'aide de nombreux cours sur internet. L'idée ici était de présenter les principaux outils pour faire une page qui vous ressemble en peu de temps. J'espère qu'il vous aura été utile!