

Conception de Sites Web Statiques : TD 2

HTML5 statique, feuille de style CSS, mise en page, site web adaptatif

<https://philippe-biasi.emi.u-bordeaux.fr/pai/web/>

Voici une petite synthèse du fonctionnement : En CSS (*cascading style sheets*), on définit un ensemble de règles, et les règles ont la syntaxe suivante :

```
sélecteur
{
  propriété : valeur;
  propriété : valeur;
  ...
}
```

L'idée des règles CSS est de sélectionner un ou plusieurs balises HTML avec des **sélecteurs** afin d'y appliquer un ou plusieurs styles, définies entre accolades, et caractérisé par des **propriétés** auxquelles on attribue des **valeurs**. Ceci permet de préciser le style à appliquer aux balises sélectionnées.

Les **sélecteurs** les plus courants sélectionnent soit une ou plusieurs balises, soit des classes, soit un élément avec un attribut `id` précis:

sélecteur	balises sélectionnées
p	Tous les éléments de la balise <code><p></code>
h1, h2	Tous les éléments des balises <code><h1></code> et <code><h2></code>
.menu	Tous les éléments de la classe "menu"
#contenu	L'unique élément avec un attribut <code>id "contenu"</code>
...	...

Les **propriétés** modifient les styles, comme par exemple :

propriété	en gras : valeurs
color	couleur du texte, p.ex. <code>#FF0000</code> ou <code>red</code> ou <code>rgb(255,0,0)</code>
font-weight	épaisseur du texte, p.ex. pour en gras: bold
...	...

Le nom "cascade" vient de la combinaison de différentes sources de styles appliqués à un même document, selon leur degré respectif de priorité. Si plus d'une règle s'applique à un élément et spécifie la même propriété, CSS donne la priorité à la règle ayant le sélecteur le plus spécifique. Par exemple, un sélecteur `id` est plus spécifique qu'un sélecteur de classe, qui lui-même est plus spécifique qu'un sélecteur de balise.

Notez que vous pouvez aussi mettre des commentaires comme suit :

```
/* commentaire */
```

Les possibilités avec les feuilles de styles en cascade CSS sont très nombreuses, et on ne vous demande pas de tous les connaître. **Par contre, il faut que vous sachiez vous en servir des références :**

- le pense-bête CSS (sur le site web du cours)
- la référence MDN : <https://developer.mozilla.org/fr/docs/Web/CSS>
- la référence W3C des sélecteurs : <https://www.w3.org/community/webed/wiki/CSS/Selectors>
- la référence W3C des propriétés : <https://www.w3.org/community/webed/wiki/CSS/Properties>
- le support de cours (sur le site web du cours)

PARTIE 1

1 Démarrage : "Hello World" avec CSS

Exercice 2.1 *Document HTML statique avec inclusion de feuille de style CSS*

Dans cet exercice, vous allez établir un document HTML qui utilise une feuille de style CSS. Si vous ne l'avez pas encore fait, dans vos documents personnels, créer un dossier **web**.

1. Dans ce dossier **web**, créer un sous-dossier **td2**. Dans ce sous-dossier, créez deux sous-dossiers **css** et **images**.
2. Avec votre éditeur de texte, créer un nouveau fichier dans votre dossier **web/td2** s'appelant **index.html**, et un nouveau fichier s'appelant **style.css** dans votre dossier **web/td2/css**.

Votre **arborescence de fichiers** sera donc :

```
web/
├── td1/
│   └── ...
├── td2/
│   ├── index.html
│   ├── css/
│   │   └── style.css
│   ├── images/
│   └── ressources/
└── ...
```

3. Dans votre fichier **index.html**, tapez le code suivant représentant un document HTML5 simpliste et enregistrez-le. Notez la séparation en en-tête et corps du document.

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

  <head>
    <meta charset="utf-8">
    <title>
      Matchball
    </title>
  </head>

  <body>
    <h1>Bienvenue sur le site Matchball</h1>
  </body>

</html>
```

4. Pour utiliser une feuille de style externe stockée dans le fichier **css/style.css**, il faut ajouter la ligne suivante dans l'en-tête de la page **index.html** (entre le **<head>** et **</head>**, par exemple **après** la balise **<title>...</title>**). **Attention** : après la balise signifie après la fermeture de la balise !!

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

2 Utilisation de sélecteurs et propriétés en CSS

Exercice 2.2 Feuille de style CSS : Sélecteur pour des balises

Dans cet exercice, vous allez éditer votre première feuille de style.

1. En utilisant votre éditeur de texte, ajoutez la règle suivante dans votre première feuille de style stockée dans le fichier `css/style.css`. Notez que cette règle utilise un **sélecteur simple** pour la balise `h1` : les propriétés dans le bloc de règles s'appliquent à toutes les balises du sélecteur. Pour l'instant, votre fichier contiendra donc seulement les lignes suivantes :

```
h1
{
    color: #000000;
    text-align: center;
    background-color: #CCCCCC;
}
```

2. Ouvrir un navigateur, par exemple Firefox, et avec CTRL-O ou dans le Menu *Fichier - Ouvrir un fichier*, ouvrez votre fichier `index.html` du dossier `web/td2`.
3. Les couleurs sont spécifiées par des triplets *rouge vert bleu* (RGB) en codage hexadécimal. Ici, pour générer une couleur de fond gris, chaque élément de couleur a la même valeur `CC`, soit $12 \times 16 + 12 = 204$ en décimal. Essayez d'autres valeurs pour chacun des éléments RGB, p.ex. `88`, `AA`, ou bien `FF`. Notez qu'au lieu des nombres dans le système hexadécimal, vous pouvez aussi mettre les nombres dans le système décimal qui nous est plus habituel `background-color: rgb(204,204,204);`
4. Dans le fichier `index.html`, après votre titre `h1`, rajoutez quatre nouveaux titres avec la balise `h2` contenant les textes "Introduction", "Classements", "Historique", et "Sujets liés".
5. Dans le fichier `css/style.css`, modifiez le sélecteur `h1` en `h1,h2` pour que les propriétés du bloc suivant s'appliquent aussi aux titres `h2`. Vous voyez donc que vous pouvez appliquer les mêmes propriétés à plusieurs sélecteurs, séparés par des virgules.

Exercice 2.3 Sélecteurs pour les classes

1. Dans votre fichier `index.html`, après le titre "Introduction", ajoutez trois paragraphes contenant du faux texte "Lorem ipsum" depuis Wikipedia (n'utilisez pas la balise `<em>...</em>`). Les trois paragraphes appliqueront une classe s'appelant `loremipsum` chacune comme suit :

```
<p class="loremipsum">...</p>
```

2. Dans votre feuille de style `css/style.css`, ajoutez la règle ci-dessous. Notez que cette règle utilise un **sélecteur de classe** pour la classe `loremipsum`, spécifié par un point `(.)` en préfixe.

```
.loremipsum
{
    font-style: italic;
    text-align: left;
}
```

3. Testez en ouvrant votre fichier `index.html` avec votre navigateur.
4. Modifiez la propriété *text-align* en mettant successivement des valeurs `left`, `right`, `center`, et `justify`, et testez avec Firefox à chaque fois.

Exercice 2.4 *Sélecteur pour un identifiant*

1. Dans la partie "Classement", ajoutez deux listes ordonnées `ol` avec le Top10 des classements des joueurs de tennis masculin et féminin, respectivement. Avant chaque liste, ajoutez un titre `h3` précisant le sexe.
2. Pour un seul joueur (ou joueuse) de votre choix, ajoutez un identifiant `id="prefere"`, et définissez un **sélecteur pour cet identifiant** (précédé par un `#`) ayant une couleur de fond légèrement rouge ou bleu, selon votre goût. *Notez qu'un identifiant est unique, il ne peut en avoir qu'un dans tout votre document !*

Exercice 2.5 *La propriété CSS "display"*

1. Encapsulez chacune de vos listes de classements (incluant le titre `h3`) en une balise `div` qui applique une classe que vous appelez `classement`. *Notez : encapsulez signifie placer la balise ouvrante avant, et la balise fermante après.* Vous devriez donc avoir une succession des balises comme suit : `<div class="classement"><h3>...</h3><ol>...</ol></div><div class="classement"><h3>...</h3><ol>...</ol></div>`
2. Dans votre feuille de style CSS, appliquez la règle suivante aux classes `classement` :

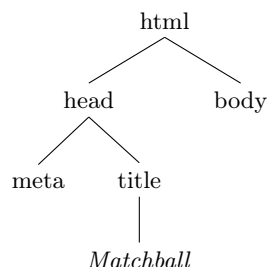
```
.classement
{
    display: block;
}
```

3. Modifiez la propriété *display* en mettant successivement des valeurs `inline-block`, `none`, et `inline`, et testez avec Firefox à chaque fois. N'hésitez pas à faire varier la largeur de la fenêtre de votre navigateur afin de voir comment le navigateur réagit.

Exercice 2.6 *Le Document Object Model (DOM)*

Le DOM est une interface de programmation (API) créé par le consortium W3C. Avec le DOM, un document HTML5 peut être représenté sous la forme d'un arbre, et permet à des scripts d'examiner et de modifier le contenu du navigateur web.

1. Complétez l'arbre DOM de votre page web (à l'ancienne, sur une feuille papier p.ex.). Vous pouvez raccourcir le contenu des feuilles.



2. Vérifiez votre solution en ouvrant les outils de développement web de votre navigateur (clic droit sur un élément à explorer, et ensuite "examiner l'élément", ou alors "inspecter l'élément", selon votre navigateur. Vous voyez aussi que les éléments sont représentés sous forme d'un arbre (horizontal ici - et vous pouvez cliquer sur les petits triangles gris pour faire apparaître les balises enfants).
3. Toujours avec les outils de développement, supprimez un élément de liste `li`.
4. Toujours avec les outils de développement, supprimez la liste `ol`. Avez-vous vu que toute la branche de l'arbre disparaît ?

Exercice 2.7 Les pseudo-classes CSS

Voici une petite synthèse du fonctionnement des pseudo-classes :

Une pseudo-classe est un mot-clé qui peut être ajouté à un sélecteur afin d'indiquer l'état spécifique dans lequel un élément doit être pour être ciblé par la déclaration.

```
sélecteur:pseudo-classe
{
    propriété : valeur;
    propriété : valeur;
    ...
}
```

1. Mettez en forme vos listes des classement Top10 pour que le premier entrée de chaque classement apparait avec une couleur de fond légèrement rouge, en utilisant la pseudo-class `:first-child`.
2. Mettez en forme vos listes des classement Top10 pour que les rangs pairs apparaissent avec une couleur de fond légèrement grisée en utilisant la pseudo-class `nth-child` (il faut l'appliquer à l'élément de liste !). **Utilisez la référence MDN pour trouver le fonctionnement, notamment les exemples :**
<https://developer.mozilla.org/fr/docs/Web/CSS/:nth-child>.
3. En dessous du titre "Sujets liés", ajouter des liens hypertextes vers des sites web intéressants pour le tennis (p.ex. WTA, ATP, et FFT).
4. Les pseudo-classes permettent d'appliquer un style aussi en fonction de facteurs externes (l'historique de navigation par exemple avec `:visited`. Définissez les *pseudo-classes* pour les liens hypertextes dans votre fichier CSS :

```
a:link    { color: red }    /* lien non-visité */
a:visited { color: blue }   /* lien visité    */
a:hover   { color: yellow } /* lien survolé   */
a:active  { color: lime }   /* lien activé    */
```

PARTIE 2

3 Structuration sémantique et mise en page

Exercice 2.8 Les balises HTML5 structurantes : *article*, *header*, *footer*

1. Dans votre fichier `index.html`, encapsulez votre titre `h2` "Introduction" et les trois paragraphes de Lorem Ipsum avec une balise `<article>...</article>` (encapsulez signifie placer la balise ouvrante avant, et la balise fermante après). Notez que que l'utilisation de la balise `article` n'a qu'une signification sémantique - par défaut elle réagit de la même manière que les balises `div`.
2. Encapsulez également le titre `h2` "Classement" avec les classements dans une balise `<article>...</article>`.
3. Encapsulez également le titre `h2` "Historique" dans une balise `<article>...</article>`.
4. Toujours dans votre fichier `index.html`, encapsulez votre titre `h1` dans une balise `<header>...</header>`.
5. Encapsulez également le titre `h2` "Sujets liés" avec la liste des liens dans une balise `<aside>...</aside>`.
6. A la fin de votre fichier `index.html`, juste avant la fermeture de la balise `body`, ajoutez une balise `<footer>...</footer>` dans laquelle vous écrivez un paragraphe "Ce site a été créé par ..." complété par votre nom.

Exercice 2.9 Le modèle de boîte CSS (*CSS Box Model*)

1. Dans votre feuille de style `css/style.css`, ajoutez la règle ci-dessous en utilisant de nouveau un sélecteur de classe.

```
article
{
    margin: 10px;
    padding: 5px;
    border: 1px solid #AAAAAA;
    background-color: #CCCCCC;
}
```

2. Testez. Avez-vous compris la différence entre les propriétés `margin` et `padding` ?
3. Ajoutez également des marges à votre en-tête (`header`) et votre pied de page (`footer`). Expérimentez.

Propriétés raccourcie: La propriété CSS `border` est une *propriété raccourcie* qui permet de définir trois propriétés liées à la bordure à la fois, à savoir les valeurs de `border-width`, `border-style` et `border-color`.

Cela signifie que la ligne

```
border: 1px solid #AAAAAA;
```

est équivalente aux trois lignes suivantes :

```
border-width: 1px;
border-style: solid;
border-color: #AAAAAA;
```

Remarques : Dans cet exemple, la propriété CSS `border-width` est d'ailleurs elle même une propriété raccourcie de `border-top-width`, `border-right-width`, `border-bottom-width`, `border-left-width`. En utilisant ces dernières propriétés à la place de la propriété raccourcie, il est donc possible d'avoir des épaisseur de bordures différents pour les quatre côtés !

D'ailleurs, les propriétés des marges extérieurs (`margin`) et intérieur (`padding`) sont également des propriétés raccourcie : ces marges peuvent être réglés séparément pour les quatres côtés : `margin-top`, `margin-right`, `margin-bottom` et `margin-left`, et, respectivement, `padding-top`, `padding-right`, `padding-bottom` et `padding-left`.

Exercice 2.10 Outils de développement Web

Dans cet exercice, pour votre navigateur, nous allons découvrir deux outils de développement Web très utiles pour vos futurs développements. Il s'agit d'une part d'une extension Chrome ou pour Firefox, et d'autre part d'un outil dit *natif*, c'est à dire disponible par défaut dans ces navigateurs.

1. Inspecteur : Passez votre souris sur le titre "Introduction" et sélectionnez ce texte. Cliquez sur le bouton droit de la souris et sélectionnez "Examiner l'élément" (ou "Inspecter l'élément") : une interface s'affiche vous permettant de facilement modifier des propriétés de style. Cliquez sur la valeur **10px** de la propriété **margin**, et modifiez cette valeur avec les flèches de votre clavier. Notez que cette modification sera perdue lors du rechargement de votre page web dans le navigateur - reportez donc les valeurs qui vous plaisent dans le fichier contenant votre feuille de style CSS que vous éditez avec VSCode.

2. Dans votre feuille de style `css/style.css`, pour la propriété **margin**, définissez deux valeurs séparées par un espace (voir ci-dessous). Avec l'examineur, modifiez les deux valeurs. Avez-vous compris l'effet ?

```
margin: 10px 5px;
```

3. Maintenant, pour la propriété **margin**, définissez quatre valeurs séparées par un espace. Expérimentez.

```
margin: 10px 5px 15px 20px;
```

4. ColorZilla : Installer l'extension "ColorZilla". Pour ce faire, utilisez votre navigateur *Chrome* ou *Mozilla Firefox*, et rendez vous à la page suivante : <https://www.colorzilla.com/>. Expérimentez : sélectionnez des couleurs avec "ColorZilla", et modifiez les valeurs dans votre feuille de style CSS.

Exercice 2.11 Flexbox : une méthode de mise en page performante

Dans l'exercice 2.5, nous avons vu comment utiliser la propriété CSS `display` pour un exemple d'agencer des balises verticalement ou horizontalement en utilisant les valeurs `block` et `inline-block`, respectivement.

Dans cette exercice, nous allons utiliser une autre méthode, qui se veut plus performante, et qui s'appelle **Flexbox**. C'est une méthode de mise en page selon un **axe principal** et un **axe secondaire**, permettant de disposer plusieurs éléments en ligne ou en colonne. Flexbox est connu pour faciliter considérablement les tâches de mise en page.

Préparation

1. Nous allons commencer par créer trois balises `div` qui contiennent un début d'historique de joueurs de tennis qui étaient le numéro 1 mondial à la fin de la saison. Pour cela, directement après le titre `h2 "Historique"` (et donc dans votre article), insérer les balises HTML suivantes :

```
<div class="historique">
  <div class="no1"><strong>2020</strong><br>Novak Djokovic</div>
  <div class="no1"><strong>2019</strong><br>Rafael Nadal</div>
  <div class="no1"><strong>2018</strong><br>Novak Djokovic</div>
</div>
```

Il s'agit donc simplement de trois balises `div` de la classe `no1` qui contiennent l'année et les joueurs, et qui sont encapsulés dans une balise `div` de la classe `historique`. Regardez la mise en page dans votre navigateur.

2. Maintenant, nous allons mettre en forme chaque balise `div` de la classe `no1` selon le modèle de boîte CSS. Pour cela, dans votre feuille de style CSS, ajoutez les règles suivantes :

```
.no1
{
  border: 1px solid black;
  width: 200px;
  height: 40px;
  text-align: center;
  margin: 10px;
  background-color: #ffffff;
}
```

Premier exemple Flexbox

3. Maintenant, nous allons définir que la balise `div` de l'historique doit utiliser Flexbox, et nous allons définir la mise en forme. Pour cela, dans votre feuille de style CSS, ajoutez les règles suivantes :

```
.historique
{
  display: flex;

  flex-direction: column; /* column par défaut, ou bien column-reverse, row, ou row-reverse */
  justify-content: flex-start; /* flex-start par défaut, ou bien flex-end, center, space-between, space-around */
  flex-wrap: nowrap; /* nowrap par défaut, ou bien wrap */

  height: 200px;
}
```

Notez que pour utiliser Flexbox, il faut mettre la valeur `flex` à la propriété `display`. Les autres propriétés `flex-direction`, `justify-content`, et `flex-wrap`, par exemple, permettent de spécifier des paramètres à Flexbox.

4. Regardez le résultat.

Axe principal

5. La propriété **flex-direction** définit l'axe principal du flux : sa valeur par défaut, donc lorsque vous ne spécifiez pas cette propriété, est la valeur **column**. Pour comprendre le fonctionnement, changez la valeur à **column-reverse**, **row**, ou **row-reverse**, et regardez à chaque fois le résultat ! Pour la suite, revenez vers un alignement en lignes avec **row**.
6. La propriété **justify-content** définit l'alignement de l'axe principal du flux : sa valeur par défaut, donc lorsque vous ne spécifiez pas cette propriété, c'est la valeur **flex-start**. Pour comprendre le fonctionnement, changez la valeur à **flex-end**, **center**, **space-between** ou **space-around**. Regardez à chaque fois le résultat ! Pour la suite, revenez vers un alignement **flex-start**.
7. Ajoutez 4 années de plus dans l'historique des joueurs, en créant donc 4 balises **div** de la classe **no1** supplémentaires, et en mettant les années 2017 à 2014 et les joueurs. Regardez le résultat.
8. La propriété **flex-wrap** définit l'alignement de l'axe principal du flux : sa valeur par défaut, donc lorsque vous ne spécifiez pas cette propriété, c'est la valeur **nowrap**. Pour comprendre le fonctionnement, changez la valeur à **wrap**. Essayez ces deux valeurs chacune pour une mise en forme par colonne et par ligne (donc 4 possibilités). Avez-vous compris le fonctionnement ?

Remarque : Propriété raccourcie pour Flexbox :

La propriété CSS **flex-flow** est une propriété raccourcie pour les propriétés **flex-direction** et **flex-wrap**. Cela signifie que la ligne

```
flex-flow: row wrap;
```

est équivalente aux deux lignes suivantes :

```
flex-direction: row;  
flex-wrap: wrap;
```

Axe secondaire

Avec la propriété **flex-direction**, vous avez défini l'axe principal du flux, par colonne ou par ligne. L'axe secondaire (aussi appelé axe croisé ou axe perpendiculaire) correspond simplement à l'autre axe, et vous pouvez définir, par exemple, l'alignement du flux pour cet axe secondaire en utilisant la propriété **align-items**, ayant pour valeurs soit **stretch**, **flex-start**, **flex-end**, **center**, ou **baseline**. Si vous le souhaitez, ajoutez la propriété **align-items** au sélecteur pour la classe **historique**, et expérimentez différentes valeurs.

Mise en page plus complexe

La clé pour faire de mises en pages plus complexes est d'imbriquer plusieurs balises de type Flexbox dans d'autres balises qui seront elles-mêmes des balises de type flexbox. Ainsi, vous créez une hiérarchie des différents alignements. Si vous le souhaitez, créez une nouvelle balise **div** de classe **historique** pour l'historique des numéros 1 féminin, et imbriquez cette balise dans une nouvelle balise **div** avec une nouvelle classe, et ajustez cette mise en page selon votre goût !

Exercice 2.12 Menu

1. Après votre **header**, créez une barre de navigation avec (notez qu'il arrive aussi que la navigation soit mis directement dans le **header**, mais ici, nous le mettons après) :

```
<nav>
  <ul class="menu">
    <li class="menuitem"><a href="#">Introduction</a></li>
    <li class="menuitem"><a href="#">Classements</a></li>
    <li class="menuitem"><a href="#">Historique</a></li>
  </ul>
</nav>
```

2. Ajouter des attributs pour des identifiants dans les titres **h2** s'appelant "introduction", "classements", et "historique" pour créer des ancres.

3. Créez des liens internes en modifiant la propriété **href** de chacun des liens, afin de positionner votre navigateur au titre correspondant.

4. Dans votre feuille de style CSS, ajoutez le code suivant et testez. Notez que ce code repose sur l'utilisation de Flex Box CSS qui est un standard CSS3 de disposition des éléments dans une page web permettant d'avoir un design adaptatif à la taille de l'écran.

```
ul.menu {
  list-style-type: none;
  margin: 0;
  padding: 0;

  display: flex;
  flex-direction: row; /* ou bien column pour vertical */
  flex-wrap: wrap;
  justify-content: flex-start; /* ou bien center ou bien flex-end */

  background-color: #CCCCCC;
  border: 4px solid #AAAAAA;
}
```

5. Changez la mise en forme des liens hypertextes du menu (sans l'avoir souligné avec **text-decoration**, avec des marges intérieures **padding**, pour avoir une zone cliquable sur toute la largeur **display**), et des couleurs de votre choix :

```
ul.menu li a
{
  text-decoration: none;
  padding: .75em 1.25em;

  display: block;

  color: black;
  background-color: #CCCCCC;
}
```

6. Au survol de votre lien, ajoutez le pseudo-sélecteur **:hover** suivant :

```
ul.menu li a:hover
{
  background-color: #999999;
}
```

7. Vous pouvez aussi mettre le menu en vertical. Modifiez la propriété **flex-direction** comme indiquée dans le commentaire en valeur **column**. Essayez ! Et remettez la valeur en **row** pour la suite.

8. Vous pouvez également changer l'alignement en modifiant la propriété **justify-content** en valeur **center** comme indiquée dans le commentaire. Essayez !

9. Pour rajouter un effet de transition, rajouter le code `transition: background .4s ease-in-out;` au sélecteur `ul.menu li a`.

10. Modifiez le menu selon votre goût.

Exercice 2.13 Les "Media queries" pour des sites web adaptatifs

Les "Media queries" permettent la mise en forme en fonction du média (ordinateur, tablette, téléphone) dans lequel la page web est affichée.

1. Rajoutez le code suivant à la fin de votre feuille de style. Expérimentez, en modifiant la largeur de la fenêtre de votre navigateur avec votre souris/trackpad.

```
@media screen and (max-width: 640px) {  
  .loremipsum {  
    background-color : #FFFFFF;  
  }  
}  
  
@media screen and (min-width: 641px) {  
  .loremipsum {  
    background-color : #CCCCCC;  
  }  
}
```

2. Testez en réduisant drastiquement la largeur de la fenêtre de votre navigateur.

Exercice 2.14 Les balises HTML5 structurantes : *section*

1. Encapsulez tous vos articles dans une section (`<section>...</section>`).
2. Encapsulez votre `section` (`<section>...</section>`) et votre partie `aside` avec une balise `<div id="conteneur">...</div>` en mettant un identifiant s'appelant `conteneur`).
3. Encapsulez tout le contenu de votre page (tout ce qui est entre `<body>...</body>`) dans une balise `div` en mettant un identifiant s'appelant `tout`.

Après ces opérations, la structure de votre document devrait maintenant être comme suit :

```
<!DOCTYPE html>  
<html lang="fr">  
  <head>  
    ...  
  </head>  
  <body>  
    <div id="tout">  
      <header> ... </header>  
  
      <nav> .... </nav>  
  
      <div id="conteneur">  
        <section>  
          <article><h2 id="introduction">Introduction</h2> ... </article>  
          <article><h2 id="classements">Classements</h2> ... </article>  
          <article><h2 id="historique">Historique</h2> ... </article>  
        </section>  
  
        <aside><h2>Sujets liés</h2> ... </aside>  
      </div>  
      <footer> ... </footer>  
    </div>  
  </body>  
</html>
```

Exercice 2.15 Mise en page

1. Dans votre feuille de style, ajoutez les instructions suivantes afin de bien séparer les différentes parties dans la structuration sémantique :

```
body
{
    background-color: #666666;
}
section, aside, header, footer
{
    border: 4px solid #AAAAAA;
    padding: 10px;
}
header, footer
{
    background-color: #888888;
}
```

2. Afin de mettre la barre latérale à coté de votre section, ajoutez dans votre feuille de style les instructions suivantes. De nouveau, nous utilisons les *Flex Box CSS*.

```
#conteneur
{
    display: flex;
    flex-direction: row;
}
section
{
    width: 70%;
    background-color: white;
}
aside
{
    width: 30%;
    background-color: beige;
}
```

3. Vous pouvez également fixer la largeur. Pour essayer, pour la propriété **width** du sélecteur **aside**, mettez une valeur de 200px au lieu des 30%.
4. Vous pouvez également fixer la largeur de tout le contenu de votre page, et, si vous le souhaitez, centrer ce contenu par rapport à la fenêtre du navigateur. Pour cela, ajoutez les instructions suivantes :

```
#tout
{
    width: 100%;
    margin: 0 auto;
}
```

Exercice 2.16 *Media queries pour la mise en forme sur écran mobile (site web adaptatif)*

Dans cet exercice, vous allez faire en sorte que le block **aside** revienne à la ligne lorsque vous êtes sur un petit écran.

1. Ajoutez à la fin de votre feuille de style :

```
/* max-width pour les petits écrans */
@media screen and (max-width: 640px) {
  #conteneur
  {
    flex-direction: column;
  }
  section, aside
  {
    width: auto;
  }
}
```

2. Faites en sorte que votre menu se présente horizontalement sur les plus grands écrans, et verticalement sur les petits écrans, selon les deux options que vous avez expérimentées dans l'exercice 2.12 sur les menus.

- Vérifier tous vos fichiers HTML5 avec le validateur W3C(<http://validator.w3.org/>)
- Vérifier tous vos fichiers CSS avec le validateur CSS (<http://jigsaw.w3.org/css-validator/>).
- Pour mettre en ligne vos fichiers, utilisez le canal de stockage sur l'ENT.

(Version 04/02/2025, 16:14:51)