

420-1W1-DM Web 1

CSS et la mise en page

Au dernier cours, nous avons vu comment lier un fichier CSS avec un document HTML avec la balise `<link>` qu'on ajoute dans le `<head>`. Nous avons appris comment appliquer du CSS sur des éléments HTML avec les sélecteurs de classes `.ma-classe`, d'identifiants `#mon-id` et d'éléments `p`.

```
/*  
Ce code change la couleur du texte en bleu pour toutes les classes 'ma-classe' ainsi que po  
*****/  
.ma-classe,  
#mon-id,  
.p {  
    color: blue;  
}
```

Nous avons appris que l'**ordre** de définition du CSS est important pour déterminer la **priorité d'application** de celle-ci sur l'élément.

Nous avons vu les propriétés CSS suivantes : - **color**: pour modifier la couleur du texte. - **background**: pour modifier la couleur de fond. - **font-size** : pour modifier la grosseur des caractères du texte. - **font-weight** : pour modifier l'épaisseur des lettres (**gras** ou non) - **font-style** : pour modifier le texte en italique ou non. - **font-family** : pour modifier la police de caractères. - **text-decoration** : pour ajouter une décoration avec une couleur et une position. Généralement, c'est pour ajouter une ligne sous ou sur le texte. - **text-align** : pour aligner le texte horizontalement à gauche, à droite ou au centre. - **text-shadow** : pour ajouter une ombre au texte.

Nous avons appris comment modifier les hyperliens `<a>` pour avoir un style différent avec les pseudo-classes (`a:link`, `a:visited`, `a:hover`, ...). Ceux-ci permettent de changer entre autre la couleur du lien quand il est cliqué.

Nous avons vu comment modifier des listes pour avoir un style différent avec la propriété **line-style** et modifier la hauteur entre chaque élément avec **line-height**.

Le CSS sert aussi à positioner les éléments à l'intérieur de la page pour faciliter

la lecture. C'est ce que l'on va explorer dans ce cours.

Le modèle de boîtes du CSS

- Toutes les balises HTML peuvent être considérées comme des boîtes. Dans le CSS, l'expression *"modèle de boîtes"* (*"box model"* en anglais) est utilisée en parlant de design et d'agencement
- Le modèle de boîtes du CSS est essentiellement une boîte qui entoure chaque balise HTML

```
<div class="border w3-theme">
  <div class="padding">
    <div class="content"></div>
  </div>
</div>
```

- **Content** (Contenu) : le contenu de la boîte, dans laquelle le texte et les images apparaissent
- **Padding** (Rembourrage) : espace transparent autour du contenu
- **Border** (Bordure) : une bordure qui entoure le contenu et le rembourrage
- **Margin** (Marge) : espace transparent qui entoure la bordure

Donc, si on n'utilise pas de bordure, l'espacement sera similaire qu'on définisse un `padding` ou un `margin`.

On modifie ces propriétés en précisant une valeur (`em`, `px`, ...) ou un pourcentage. Il est possible de modifier en même temps tous les côtés ou de spécifier certains côtés de la boîte.

```
/* On applique la même valeur aux quatre côtés */
padding: 1em;
```

```
/* vertical (haut et bas) / horizontal (droite et gauche) */
padding: 5% 10%;
```

```
/* haut / horizontal / bas */
padding: 1em 2em 2em;
```

```
/* haut / droit / bas / gauche */
padding: 5px 1em 0 2em;
```

Exemple 6: ex6.html

- La classe `hidden` peut être utilisée pour cacher certaines balises
 - elle va être expliquée plus en détails un peu plus loin

```
<!DOCTYPE html>
<html lang="en">
<head>
```

```

    <meta charset="UTF-8">
    <title>CSS Exemple 6</title>
    <link rel="stylesheet" type="text/css" href="ex6.css">
</head>
<body>
<h1>CSS Exemple 6</h1>
<dl class="special_box">
    <dt>Content</dt>
    <dd>The content of the box, where text and images appear</dd>
    <dt>Padding</dt>
    <dd>Clears an area around the content. The padding is transparent</dd>
    <dt class="hidden">Border</dt>
    <dd class="hidden">A border that goes around the padding and content</dd>
    <dt>Margin</dt>
    <dd>Clears an area outside the border. The margin is transparent</dd>
</dl>
</body>
</html>

```

Fichier ex6.css

```

.special_box {
    background-color: lightgoldenrodyellow;
    padding: 25px;
    border: 5px solid red;
    margin: 100px;
}

dt {
    font-weight: bolder;
}

/* uncomment either the display or visibility declaration to enable the hidden class */
.hidden {
    /*display: none;*/
    /*visibility: hidden;*/
}

<!DOCTYPE html>

<dl class="special_box">
    <dt>Content</dt>
    <dd>The content of the box, where text and images appear</dd>
    <dt>Padding</dt>
    <dd>Clears an area around the content. The padding is transparent</dd>
    <dt class="hidden">Border</dt>
    <dd class="hidden">A border that goes around the padding and content</dd>

```

```

    <dt>Margin</dt>
    <dd>Clears an area outside the border. The margin is transparent</dd>
</dl>

```

La propriété border (Référence)

La propriété `border` spécifie le style, la largeur et la couleur d'une bordure d'un élément HTML. On peut aussi définir tous les côtés (gauche, droite, haut et bas) ou seulement une partie. Par exemple, pour définir la bordure du bas seulement on va utiliser `border-bottom`. La syntaxe est similaire pour toutes les propriétés `border-*`.

```

/** largeur est une unité de mesure comme px **/
border: largeur style couleur;

<!DOCTYPE html>

<div class="bordure-simple">
    Je suis du texte avec une bordure noire simple
</div>
<div class="bordure-bas">
    Je suis du texte avec une bordure rouge dans le bas
</div>
<div class="bordure-ronde">
    Je suis du texte avec une bordure ronde grise
</div>
<div class="bordure-gauche">
    Je suis du texte avec une bordure à gauche en bleu
</div>

.ex1-4 div {
    margin-top: 20px;
}

.ex1-4 .bordure-simple {
    border: 2px solid black;
}

.ex1-4 .bordure-bas {
    border-bottom: 2px solid red;
}

.ex1-4 .bordure-ronde {
    border: 2px solid gray;
    border-radius: 4px;
}

.ex1-4 .bordure-gauche {

```

```

border-left: 6px solid #2196F3;
background: #ddffff
}

<body>
  <div class="ex1-4">
    <div class="bordure-simple">
      Je suis du texte avec une bordure noire simple
    </div>
    <div class="bordure-bas">
      Je suis du texte avec une bordure rouge dans le bas
    </div>
    <div class="bordure-ronde">
      Je suis du texte avec une bordure ronde grise
    </div>
    <div class="bordure-gauche">
      Je suis du texte avec une bordure à gauche en bleu
    </div>
  </div>
</body>

```

La propriété display

- La propriété `display` spécifie comment la balise est affichée ou non.
- Chaque balise HTML a une valeur de `display` par défaut selon son type de balise
- La valeur par défaut de la plupart des balises est `block` ou `inline`

Balises de bloc (block)

- Une balise de bloc est une balise qui commence toujours sur une nouvelle ligne et qui utilise toute la largeur disponible
- Exemples de balise de bloc :
 - `<div>`
 - `<h1>` à `<h6>`
 - `<p>`
 - `<form>`

Balises en-ligne (inline)

- Une balise en-ligne ne commence pas toujours sur une nouvelle ligne et utilise seulement la largeur nécessaire
- Exemples de balise en-ligne :
 - ``

- <a>
-

Cacher une balise

display: none ou **visibility: hidden** ?

- Cacher une balise peut être fait en donnant la valeur **none** à la propriété **display** (*affichage = rien*)
 - La balise va être cachée et la page va être affichée comme si la balise n'existait pas
 - modifiez le fichier **ex6.css** en *décommentant* **display: none** à l'intérieur de la déclaration de classe **hidden** ; ensuite sauvegardez les changements et rechargez le fichier **ex6.html**
- **visibility: hidden** cache aussi une balise
 - Cependant, la balise va utiliser le même espace qu'avant
 - la balise sera cachée, mais va continuer d'affecter l'agencement et l'affichage de la page
 - modifiez le fichier **ex6.css** en commentant **display: none** et en décommentant **visibility: hidden** à l'intérieur de la déclaration de classe **hidden** ; ensuite sauvegardez les changements et rechargez le fichier **ex6.html**
 - qu'est-ce qui a changé ?

Exemple avec display: none <!DOCTYPE html>

```
<dl class="special_box">
  <dt>Content</dt>
  <dd>The content of the box, where text and images appear</dd>
  <dt>Padding</dt>
  <dd>Clears an area around the content. The padding is transparent</dd>
  <dt class="hidden">Border</dt>
  <dd class="hidden">A border that goes around the padding and content</dd>
  <dt>Margin</dt>
  <dd>Clears an area outside the border. The margin is transparent</dd>
</dl>
```

Exemple avec visibility: hidden <!DOCTYPE html>

```
<dl class="special_box">
  <dt>Content</dt>
  <dd>The content of the box, where text and images appear</dd>
  <dt>Padding</dt>
  <dd>Clears an area around the content. The padding is transparent</dd>
  <dt class="hidden">Border</dt>
  <dd class="hidden">A border that goes around the padding and content</dd>
```

```

        <dt>Margin</dt>
        <dd>Clears an area outside the border. The margin is transparent</dd>
    </dl>

```

La propriété `height` et `width`

Ces deux propriétés modifient la hauteur (`height`) et la largeur (`width`) du contenu. C'est une bonne façon de forcer l'élément HTML à prendre une dimension. À noter que les propriétés `height` et `width` **ne touchent pas** au `padding`, aux bordures et aux marges. Seulement la partie `Content` est touchée.

```

<div class="padding">
  <div class="content"></div>
</div>

```

Les propriétés `height` et `width` prennent les valeurs suivantes:

- `auto` : c'est la valeur par défaut. C'est le navigateur qui calcule la hauteur ou la largeur.
- `%` : La hauteur ou la largeur est calculée selon un pourcentage du block entourant l'élément.
- `mesure` : La hauteur ou la largeur est définie selon une mesure comme `px`, `cm`, ...
- `inherit` : La hauteur ou la largeur est héritée de la valeur de l'élément parent.

```
<!DOCTYPE html>
```

```

<div class="div-50">
  Je suis un élément avec une hauteur de 200 pixels et une largeur de 50% maximum de la page
</div>
<div class="div-500">
  Je suis un élément avec une hauteur de 100 pixels et une largeur de 500 pixels pour toute la page
</div>
<div class="div-auto">
  Je suis un élément avec une hauteur et une largeur automatique selon mon contenu! Ce qui est le mieux
</div>
<div class="div-max">
  Je suis un élément avec une largeur de 600 pixels au maximum. Alors je peux être plus large que la page
</div>
<div class="div-min-max">
  Je suis un élément avec une largeur de 500 pixels au maximum. Mais, je ne peux pas être plus large que la page
</div>

```

```

<!DOCTYPE html>
<html lang="en">
<head>

```

```

<title>CSS Height and Width Exemple</title>
<link rel="stylesheet" type="text/css" href="css_dimension.css">
</head>
<body>
  <div class="ex5">
    <div class="div-50">
      Je suis un élément avec une hauteur de 200 pixels et une largeur de 50% maximum
    </div>
    <div class="div-500">
      Je suis un élément avec une hauteur de 100 pixels et une largeur de 500 pixels p
    </div>
    <div class="div-auto">
      Je suis un élément avec une hauteur et une largeur automatique selon mon contenu
    </div>
    <div class="div-max">
      Je suis un élément avec une largeur de 600 pixels au maximum. Alors je peux être
    </div>
    <div class="div-min-max">
      Je suis un élément avec une largeur de 500 pixels au maximum. Mais, je ne peux p
    </div>
  </div>
</body>
</html>

```

Fichier `css_dimension.css`

```

.ex5 .div-50 {
  height: 200px;
  width: 50%;
  background-color: powderblue;
}

.ex5 .div-500 {
  height: 200px;
  width: 500px;
  background-color: #b1ede9;
}

.ex5 .div-auto {
  height: auto;
  width: auto;
  background-color: #cae0de;
  font-size: 24px;
}

.ex5 .div-max {

```



```

    max-width: 600px;
    height: 100px;
    background-color: powderblue;
}

```

```

.ex5 .div-min-max {
    max-width: 500px;
    min-width: 300px;
    background-color: #b1ede9;
}

```

On remarque qu'il existe d'autres propriétés comme `max-height` et `max-width` qui définissent une dimension **maximale** de l'élément HTML. Leur contraires `min-height` et `min-width` définissent la dimension **minimale** de l'élément HTML.

C'est une bonne façon de définir quand la barre de défilement apparaît selon la taille de l'écran. De plus, dans certaines situations, on veut s'assurer qu'un élément HTML ne soit pas trop écrasé ou étiré. Ce qu'on peut faire en définissant le `max-*` et le `min-*` en même temps.

Une autre combinaison intéressante est d'ajouter `margin: auto` à un `max-width` pour garder un élément centré et que ce soit les marges qui s'ajustent si l'écran change de dimension.

```

.ex5 .div-margin-auto {
    max-width: 500px;
    margin: auto;
    border: 3px solid #73AD21;
}

```

```

<div class="div-margin-auto">

```

```

    Je suis un élément avec une largeur de 500 pixels au maximum et avec une marge automatique
</div>

```

La propriété float

- La propriété `float` est utilisée pour positionner et formater du contenu, par exemple pour placer une image à gauche d'un texte placé dans un conteneur
- La propriété `float` peut avoir les valeurs suivantes :
 - `left`: la balise flotte à gauche de son conteneur
 - `right`: la balise flotte à droite de son conteneur
 - `none`: valeur par défaut ; la balise ne flotte pas, elle est placée où elle apparaît dans le code HTML
 - `inherit`: la balise hérite de la valeur de `float` de sa balise parente

Exemple: css_float.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>CSS Exemple 1</title>
  <link rel="stylesheet" type="text/css" href="css_float.css">
</head>
<body>
<div id="floating-div">
  <h1>Hello</h1>
  <p>Some Text. Some Text. Some Text. Some Text. Some Text. Some Text. Some
    Text.
    Some Text. Some Text. </p>
</div>
<div>
  <h1>Goodbye</h1>
  <p>Some other text. Some other text. Some other text. Some other text. Some
    other text. Some other text. </p>
  <h2>Hi</h2>
  <p>Some other text. Some other text. Some other text. Some other text. Some
    other text. Some other text. </p>
  <h2>Bye</h2>
  <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
    tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim
    veniam,
    quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo
    consequat. Duis aute irure dolor in reprehenderit in voluptate velit
    esse cillum
    dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non
    proident,
    sunt in culpa qui officia deserunt mollit anim id est laborum.</p>
</div>
</body>
</html>
```

Fichier : css_float.css

```
#floating-div {
  background-color: #44ff44;
  float: right;
  width: 20vw;
  min-width: 100px;
}

div {
```

```

    background-color: #ff4444;
}
<!DOCTYPE html>
<div id="floating-div">
  <h1>Hello</h1>
  <p>Some Text. Some Text. Some Text. Some Text. Some Text. Some
    Text.
    Some Text. Some Text. </p>
</div>
<div>
  <h1>Goodbye</h1>
  <p>Some other text. Some other text. Some other text. Some
    other text. Some other text. </p>
  <h2>Hi</h2>
  <p>Some other text. Some other text. Some other text. Some
    other text. Some other text. </p>
  <h2>Bye</h2>
  <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
    tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim
    veniam,
    quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo
    consequat. Duis aute irure dolor in reprehenderit in voluptate velit
    esse cillum
    dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non
    proident,
    sunt in culpa qui officia deserunt mollit anim id est laborum.</p>
</div>

```

Exercice #3

Reprenons l'exercice de La Jalousie du Barbouillé. Nous allons déplacé certains éléments pour que le contenu soit plus lisible.

Modifiez l'entête de la page pour qu'elle ait une **hauteur** de 200 pixels fixe : - Vous pouvez appliquer cela sur l'id **#top**. - Modifiez l'image dans l'entête pour qu'elle ait une **hauteur** de 200 pixels et que sa **largeur** soit calculée automatiquement. - Créez trois colonnes dans l'entête. La première contiendra l'image et prendra 20% de la **largeur**. La colonne du milieu sera de 60% de **largeur** et contiendra le titre principal. La dernière colonne sera vide aura une **largeur** de 20%. Utiliser `float: left;` et `width: %` pour vous aidez.

Mettez en forme le pied de page en ajoutant la même **couleur de fond** que l'entête. Ajoutez un **padding** de quelques pixels et **alignez le texte** au centre.

Ajouter une **marge** pour chaque didascalie (**classe didascalie**) à gauche et en

haut de 30 pixels : - Ajouter un **padding** de 10 pixels. - Ajouter une **bordure** à gauche de 6 pixels avec la couleur #f7a156. - Ajouter une **couleur de fond** plus pâle comme #fff7f0.

Ajoutez une **bordure grise** (#e1e4e5) de 2 pixels autour des sections source, autre-exemple, représentation et pièce. Ajoutez aussi un **padding** et une **marge** de 20 pixels. (**Truc**: vous pouvez utiliser les ids comme #source et #autre-exemple)

Ajoutez une **bordure** dans le bas de la scène 1 de 6 pixels de couleur grise (#e1e4e5).

Choisissez une réplique et ajoutez **visibility: hidden**. Faites la même chose pour une autre réplique mais avec **display: none**. Remarquez l'espace qui est laissée selon la propriété utilisée.

Barre de navigation

Il existe deux types de barre de navigation. La barre de navigation horizontale qui est généralement au haut de la page. L'autre type est la barre de navigation verticale qui est soit à droite ou à gauche de l'écran. La propriété **position** est pratique pour placer cette barre de navigation. Cette propriété définit comment les propriétés **top**, **bottom**, **left** ou **right** affectent la position de l'élément HTML.

Il existe 5 valeurs possibles pour la propriété **position** :

- **static** : C'est la valeur et le comportement par défaut.
- **relative** : La position est ajustée par rapport à la position de base. (déplacement)
- **fixed** : La position de l'élément est toujours à la même place dans la page. Peut importe comme l'utilisateur défile la page.
- **absolute** : La position de l'élément est calculé selon le point haut gauche de l'élément parent.
- **sticky** : C'est un mélange de **fixed** et **relative**. L'élément est **relative** jusqu'à ce que une certaine hauteur est dépassée. Alors l'élément devient **fixed**.

Les valeurs **top**, **right**, **bottom** et **left** définissent le décalage des éléments.

Voir l'exemple **css__position** pour un exemple du comportement.

Important de ne pas oublier que l'axe des Y est positif **vers le bas**. C'est dans le sens de la lecture.

Barre de navigation = liste de liens

- Une barre de navigation est d'abord définie dans le document HTML

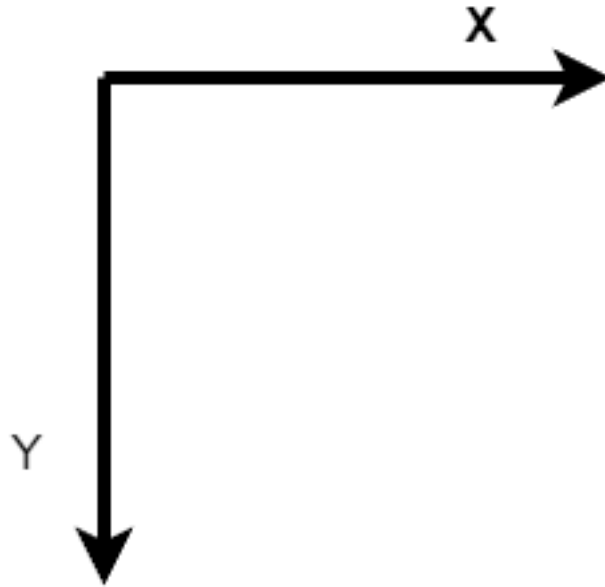


Figure 1: Axe de transformations en HTML

- Dans nos exemples, la barre de navigation est construite à partir d'une liste HTML standard
- Une barre de navigation est essentiellement une liste de liens, donc on utilise les balises `` et `` :
 - le `ul` a la classe `menulist`
 - le `ul` est à l'intérieur d'une `div` avec comme ID `menubar`
 - * on pourrait aussi remplacer la `div` par `navbar` si c'est la barre de navigation principale du site
 - chaque `li` a la class `menuitem`, et contient un lien (une balise `<a>`)
- À part cette liste, le document HTML ne contient rien de spécial
 - la *magie* s'opère dans le fichier CSS

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>CSS Exemple 2</title>
  <link rel="stylesheet" type="text/css" href="css_nav.css">
</head>
<body>
<div id="menubar">
  <ul class="menulist">
    <li class="menuitem">
```

```

        <a class="active" href="">Home</a>
    </li>
    <li class="menuitem">
        <a href="download">Download</a>
    </li>
    <li class="menuitem">
        <a href="doc">Documentation</a>
    </li>
    <li class="menuitem">
        <a href="blog">Blog</a>
    </li>
    <li class="menuitem">
        <a href="contact">Contact</a>
    </li>
</ul>
</div>

<div id="topheader">
    <h1>Hello</h1>
    <p>Some Text. Some Text. Some Text. Some Text. Some Text. Some
        Text. Some Text. Some Text. </p>
</div>
<div>
    <h1>Goodbye</h1>
    <p>Some other text. Some other text. Some other text. Some
        other text. Some other text. </p>
    <h2>Hi</h2>
    <p>Some other text. Some other text. Some other text. Some
        other text. Some other text. </p>
</div>
</body>
</html>

```

Fichier css_nav.css

```

#menubar {
    /* text-align: center;*/ /* for centering menu */
    /* float: right;*/ /* uncomment for vertical menu floating on the right */
    display: block /* block will use full width, flex will not use full width */
    /* make it sticky */
    /*position: -webkit-sticky; /* Safari */
    /*position: sticky; top: 0;*/
}

ul.menulist {
    position: relative;

```

```

        overflow: auto;
        list-style-type: none;
        margin: 0;
        padding: 0;
        background-color: black;
        border: 2px solid limegreen;
        font-family: monospace;
        font-size: 1.2em;
        /*width: auto;*/ /* for centering menu */
        /*display: inline-block;*/ /* for centering menu */
    }

    li.menuitem {
        text-align: center;
        float: left; /* comment out to make a vertical menu */
    }

    li.menuitem a {
        display: block;
        color: limegreen;
        font-weight: bolder;
        padding: 8px 8px;
        text-decoration: none;
    }

    li.menuitem a.active {
        background-color: limegreen;
        color: white;
    }

    li.menuitem a:hover:not(.active) {
        background-color: limegreen;
        color: black;
    }

```

Exemple: css_nav.html <!DOCTYPE html>

```

<div id="menubar">
    <ul class="menulist">
        <li class="menuitem">
            <a class="active" href="">Home</a>
        </li>
        <li class="menuitem">
            <a href="download">Download</a>
        </li>
        <li class="menuitem">

```

```

        <a href="doc">Documentation</a>
    </li>
    <li class="menuitem">
        <a href="blog">Blog</a>
    </li>
    <li class="menuitem">
        <a href="contact">Contact</a>
    </li>
</ul>
</div>
<div id="topheader">
    <h1>Hello</h1>
    <p>Some Text. Some Text. Some Text. Some Text. Some Text. Some
        Text. Some Text. Some Text. </p>
</div>
<div>
    <h1>Goodbye</h1>
    <p>Some other text. Some other text. Some other text. Some
        other text. Some other text. </p>
    <h2>Hi</h2>
    <p>Some other text. Some other text. Some other text. Some
        other text. Some other text. </p>
</div>

```

Référence

- W3Schools CSS Tutorial: https://www.w3schools.com/css/css_navbar.asp

Exercice #4

Finalisons certains changements sur la page de la Jalousie du Barbouillé.

Déplacez la liste des personnages pour qu'elle prenne toute la page à droite en dessous de l'entête : - Entourez toutes les sections (la pièce, représentation, sources et autres exemples) dans une nouvelle balise `<div>`. Définissez une marge à droite de 25%; - Pour que **la liste des personnages** prennent ce 25%, forcez la largeur de la liste des personnages à 25%. - Forcez la liste des personnages à droite avec une position fixe `position: fixed;` à droite `right: 0;` - Vous pouvez ajouter `height: 100%;` Si vous voulez que la liste des personnages prennent toute la section de droite.

Créer un menu avec la table des matières : - Placer le menu en dessous de l'entête principale. - Inspirez-vous de l'exemple plus haut. Vous pouvez utiliser `#table-matieres` au lieu de `menuitem`. - Vous pouvez utiliser la couleur de fond

(#f0ff0) et la couleur du texte rgb(25, 155, 25) à la place des autres valeurs pour votre table des matières.

Tableaux en CSS et HTML

Voir l'autre page de note de cours 06_tableau sur LÉA.

Exercice #5

Ajoutez une nouvelle section dans la page de la Jalousie du Barbouillé entre **Source** et **Représentation** qui s'appelle **Statistiques des plus gros films au box-office**. Ajouter la section à la table des matières.

Prenez les 5 premiers films de la liste -> https://fr.wikipedia.org/wiki/Liste_des_plus_gros_succ%C3%A8s_d_office_mondial et créez un tableau avec le titre, le réalisateur, l'année, le budget et les recettes.

Par défaut, la couleur de fond peut être `aliceblue` pour l'entête et pour chaque ligne paire `tr:nth-child(even)`. Ajoutez une couleur de fond grise (rgb(197, 209, 219)) pour le `tr:hover` de chaque ligne.

Mettez le tableau à la pleine largeur (100%) de la section. Vous pouvez mettre `border-collapse: collapse;`.

Alignez le texte du tableau à gauche.

Ajoutez un **padding** de 10 pixels entre chaque ligne.

Voici un exemple du résultat

```
<!DOCTYPE html>

<table class="">
  <caption>Liste des 5 plus gros succès du box-office mondial</caption>
  <thead>
    <tr>
      <th scope="col" class="headerSort" tabindex="0" role="columnheader button"
        title="Tri croissant">Rang
      </th>
      <th scope="col" class="headerSort" tabindex="0" role="columnheader button"
        title="Tri croissant">Titre
      </th>
      <th scope="col" class="headerSort" tabindex="0" role="columnheader button"
        title="Tri croissant">Réalisateur
      </th>
      <th scope="col" class="headerSort" tabindex="0" role="columnheader button"
        title="Tri croissant">Année
      </th>
      <th scope="col" class="headerSort" tabindex="0" role="columnheader button"
```

```

        title="Tri croissant">Budget
    </th>
    <th scope="col" class="headerSort" tabindex="0" role="columnheader button"
        title="Tri croissant">Recettes
    </th>
</tr>
</thead>
<tbody>
    <tr>
        <td>1</td>
        <td><i>Avatar</i>
        </td>
        <td>James Cameron</td>
        <td align="center">2009</td>
        <td align="right">237&nbsp;000&nbsp;000&nbsp;$</td>
        <td align="right">2&nbsp;847&nbsp;397&nbsp;339&nbsp;$</td>
    </tr>
    <tr>
        <td>2
        </td>
        <td><i>Avengers: Endgame</i>
        </td>
        <td>Anthony et<br>Joe
            Russo
        </td>
        <td align="center">2019
        </td>
        <td align="right">356&nbsp;000&nbsp;000&nbsp;<abbr class="abbr" title="dollar">$
        </td>
        <td align="right">2&nbsp;797&nbsp;501&nbsp;328&nbsp;<abbr class="abbr"
            title="dollar">$</abbr>
        </td>
    </tr>
    <tr>
        <td>3
        </td>
        <td><i>Titanic</i>
        </td>
        <td>James Cameron
        </td>
        <td align="center">1997
        </td>
        <td align="right">200&nbsp;000&nbsp;000&nbsp;<abbr class="abbr" title="dollar">$
        </td>
        <td align="right">2&nbsp;201&nbsp;647&nbsp;264&nbsp;<abbr class="abbr"
            title="dollar">$</abbr>
    </tr>

```

