

420-1W1-DM Web 1

CSS et l'intégration du style

- Le HTML décrit la structure d'un document, tandis que le CSS décrit comment il devrait être affiché
- CSS est un langage qui décrit le style d'un document HTML
- CSS décrit comment un document HTML devrait être affiché

Pourquoi utiliser le CSS ?

- Le CSS est utilisé pour définir des styles pour nos pages web, incluant le design, l'agencement et les variations dans l'affichage sur des appareils et écrans différents.
- CSS sauve beaucoup de travail. Il peut contrôler l'agencement de plusieurs pages web d'un seul coup

Mais comment on peut organiser le HTML pour faciliter l'utilisation du CSS?

Le pouvoir des catégories

Est-ce qu'il existe des éléments, dans nos pages web, qui réfèrent à une même catégorie ? Prenez simplement notre pièce la *Jalousie du Barbouillé*. Dans cette page, vous retrouvez plusieurs *didascalies*, plusieurs *noms de personnage* (au début des dialogues) et plusieurs *indications à l'acteur* (ce que nous avons surligné). Il serait intéressant d'informer le navigateur que toutes les didascalies qu'elles sont des didascalies. Cette utilité sera importante pour le CSS. Heureusement, les catégories existent en HTML. Le principe est d'ajouter un attribut qui informe sur la catégorie des éléments. En gros, nous leur ajoutons un type.

Les catégories en HTML, l'attribut `class="categorie"`

(Référence)

Il s'agit d'un attribut global. C'est-à-dire qu'il est possible d'ajouter l'attribut `class` sur toutes les balises HTML. Cet attribut permet d'ajouter, à la base, une catégorie. Il est même possible d'ajouter plusieurs catégories. Pour y ajouter

plusieurs catégories, nous ajoutons un simple espace blanc entre les noms de catégories. Il y a quelques règles sur les noms utilisés pour les catégories.

- Il doit être **significatif**.
- Il **ne doit pas** y avoir d'**espaces** dans un nom.
- Il **ne faut pas** utiliser de **caractères spéciaux** (les tirets - sont acceptés).

Voici deux exemples, dans la pièce *La Jalousie du Barbouillé*, nous pouvons ajouter la catégorie **didascalie** à nos *didascalies*.

```
<section id="scene-1">
  <h3>Scène première (<em>Le Barbouillé</em>)</h3>
  <p class="didascalie">
    <em>
      Le barbouillé entre sur la scène par le côté jardin. Il avance
      songeux vers le centre de la scène. Il garde ces yeux vers le base.
      Il se place devant le public. Lentement, il relève tête et il
      regarde le public dans les yeux.
    </em>
  </p>

  <p>
    <strong>Le Barbouillé</strong>
    <mark>(au public)</mark>
    : Il faut avouer que je suis le plus malheureux de tous les hommes !
    J'ai une femme qui me fait enrager.
  </p>

  <p class="didascalie">
    <em>
      Le Barbouillé se déplace vers le côté cours tout en regardant le
      public.
    </em>
  </p>

  <p>
    <strong>Le Barbouillé</strong>
    <mark>(au public)</mark>
    : Au lieu de me donner du soulagement et de faire les choses à mon
    souhait, elle me fait donner au diable vingt fois le jour.
  </p>

  <p class="didascalie">
    <em>
      Le Barbouillé se tourne vers le côté jardins et il continue à
```

```

        discourir.
    </em>
</p>

<p>
    <strong>Le Barbouillé</strong>
    <mark>(au public)</mark>
    : Au lieu de se tenir à la maison, elle aime la promenade, la bonne
    chère, et fréquente je ne sais quelle sorte de gens.
</p>

<p class="didascalie">
    <em>
        Arrivé au côté jardin, le Barbouillé se retourne vers le public.
    </em>
</p>

<p>
    <strong>Le Barbouillé</strong>
    <mark>(au public)</mark>
    : Ah ! pauvre Barbouillé, que tu es misérable ! Il faut pourtant la
    punir. Si tu la tuais... L'intention ne vaut rien, car tu serais pendu.
    Si tu la faisais mettre en prison... La carogne en sortirait avec son
    passe-partout. Que diable faire donc ?
</p>
</section>

```

Nous pouvons, aussi, ajouter la catégorie *dialogue* à nos *répliques*. De plus, nous pouvons modifier les balises autour du nom des personnages. Nous avons vu, la semaine dernière, la balise `` qui donne un conteneur générique. Nous verrons son utilité dans une prochaine section.

```

<section id="scene-1">
    <h3>Scène première (<em>Le Barbouillé</em>)</h3>
    <p class="didascalie">
        <em>
            Le Barbouillé entre sur la scène par le côté jardin. Il avance
            songeux vers le centre de la scène. Il garde ces yeux vers le bas.
            Il se place devant le public. Lentement, il relève tête et il
            regarde le public dans les yeux.
        </em>
    </p>

    <p class="dialogue">
        <span>Le Barbouillé</span>
        <mark>(au public)</mark>

```

```

: Il faut avouer que je suis le plus malheureux de tous les hommes !
J'ai une femme qui me fait enrager.
</p>

<p class="didascalie">
  <em>
    Le Barbouillé se déplace vers le côté cours tout en regardant le
    public.
  </em>
</p>

<p class="dialogue">
  <span>Le Barbouillé</span>
  <mark>(au public)</mark>
  : Au lieu de me donner du soulagement et de faire les choses à mon
  souhait, elle me fait donner au diable vingt fois le jour.
</p>

<p class="didascalie">
  <em>
    Le Barbouillé se tourne vers le côté jardins et il continue à
    discourir.
  </em>
</p>

<p class="dialogue">
  <span>Le Barbouillé</span>
  <mark>(au public)</mark>
  : Au lieu de se tenir à la maison, elle aime la promenade, la bonne
  chère, et fréquente je ne sais quelle sorte de gens.
</p>

<p class="didascalie">
  <em>
    Arrivé au côté jardins, le Barbouillé se retourne vers le public.
  </em>
</p>

<p class="dialogue">
  <span>Le Barbouillé</span>
  <mark>(au public)</mark>
  : Ah ! pauvre Barbouillé, que tu es misérable ! Il faut pourtant la
  punir. Si tu la tuais... L'intention ne vaut rien, car tu serais pendu.
  Si tu la faisais mettre en prison... La carogne en sortirait avec son
  passe-partout. Que diable faire donc ?
</p>

```

</section>

À quoi ressemble le CSS

La syntaxe du CSS est assez simple. Le langage fonctionne par **propriété**. En gros, le principe est d'attribuer des valeurs à des propriétés prédéfinies. Ces propriétés sont des placés dans des blocs qui indiquent que cette propriété est appliquée à une balise. Sans plus attendre, voici la syntaxe générale du CSS :

```
balise {
    propriété-1: valeur-de-la-propriété;
    propriété-2: valeur-de-la-propriété;
    propriété-3: valeur-de-la-propriété;
/ / . . .
}

/*****
 * En CSS, les commentaires utilisent les mêmes caractères qu'en Java.
 *
 * Les commentaires de ligne : // Le commentaire...
 *
 * Les blocs de commentaires (sur plusieurs lignes) : /* Le commentaire ...*/
```

Ajouter du CSS au HTML

Avant de voir nos premières propriétés, voyons comment ajouter le CSS à un fichier HTML. Il existe deux solutions possibles : lié un fichier *CSS* ou ajouter dans une balise <style></style>.

Ajouter dans le document HTML, la balise <style></style>

(Référence)

Cette balise, placé dans la balise <head></head> permet de définir sur CSS pour le document uniquement. Elle s'utilise comme suit :

```
<head>
    <meta charset="utf-8">
    <title>Titre du site</title>
    <style>
        balise {
            propriété-1: valeur-de-la-propriété;
            propriété-2: valeur-de-la-propriété;
            propriété-3: valeur-de-la-propriété;
        / / . . .
        }
    </style>
```

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

Avantage de la balise

La balise permet de réduire le nombre de fichiers sur le site. De plus, vous êtes assuré d’avoir toujours le style dans votre page web.

Désavantage de la balise

Ceci alourdit de manière importante votre page web. De plus, votre code ne peut pas être réutilisé dans d’autre page web.

Utilisation de la balise dans le cadre de ce cours

À moins que ce soit expressément demandé, nous **n’utiliserons pas** cette balise. Pour vos TP, vous devrez utiliser la prochaine balise.

Lier un fichier CSS et HTML, la balise `<link href="lien" rel="type">`

(Référence)

Cette balise, placé dans la balise `<head></head>` permet de lier un fichier CSS dans une page web. Nous verrons plus tard qu’il est possible de lier d’autre ressource qu’un fichier CSS. Pour les fichiers CSS, cette balise s’utilise comme suit :

```
<head>
  <meta charset="utf-8">
  <title>Titre du site</title>
  <link rel="stylesheet" href="./css/styles.css">
</head>
```

Fichier CSS

Les fichiers CSS ont toujours l’extension `.css`. Vous créez votre fichier exactement comme les fichiers HTML dans Visual Studio Code, mais vous sélectionnez **CSS (.css)** comme type de fichier. Par habitude et convention personnelle, je place toujours mes fichiers CSS dans un dossier `./css/` situé à la racine du site web (à côté de notre page web). Ceci permet de simplifier la vue dans le dossier et la recherche des fichiers. J’applique la même logique aux images et vidéo.

Le code de base d’un fichier CSS

```
@charset "utf-8";
```

L'attribut href="lien"

Il s'agit ni plus ni moins que du même attribut que pour la balise ``. Son but, dans la balise `<link>` est d'informer du lien (chemin) vers notre ressource (fichier CSS).

Avantage de la balise

La balise permet de lier le CSS sans alourdir la page. De plus, vous pouvez réutiliser le fichier sur plus d'un site et vous pouvez toujours ajouter plus d'un fichier CSS. Il est donc possible de spécialiser nos fichiers.

Désavantage de la balise

Nous sommes dépendants de l'existence des fichiers CSS. S'il est déplacé, renommé ou supprimé, il faut modifier notre page web en conséquence.

Utilisation de la balise dans le cadre de ce cours

À moins que ce soit expressément demandé, nous **utiliserons toujours** cette balise. Pour simplifier l'apprentissage les premiers exemples dans ce cours utiliserons la balise `<style></style>`.

L'attribut rel="type"

(Référence)

Cet attribut permet de spécifier le type de ressource lié par la balise `<link>`. Concernant les fichiers de style CSS, il faut comprendre que nous utilisons le type `rel="stylesheet"`. Il en existe d'autres, mais nous les verrons plus tard. Pour le moment, il ne sont pas utiles.

Notre première propriété CSS, color

(Référence)

Nous avons vu un exemple, pendant le cours #4. Cette propriété permet de modifier la couleur d'écriture. Il existe plusieurs manières d'écrire une couleur :

- Avec son nom (exemple : rouge = **red**) ; il existe des codes de nom prédéfini (supporté par tous les navigateurs). Cliquez sur ce lien pour connaître les différents noms.
- Il existe la couleur hexadécimale (exemple : rouge = **#ff0000**)
- Il existe la couleur RGB (exemple rouge = **rgb(255, 0, 0)**)
- Il existe la couleur RGBA (exemple rouge = **rgb(255, 0, 0, 1)**)
- Le mot clé `currentcolor` qui permet de réutiliser automatiquement la couleur mentionnée dans la balise.

Cette propriété, comme toutes les autres, s'utilise simplement en indiquant son nom (`color`) et sa valeur.

```
balise {  
    color: valeur;  
}
```

Comment ajouter les titres en bleu foncé dans notre pièce.

Pour que nos textes soient écrits en bleu foncé, il faut spécifier, toujours dans le fichier CSS, la balise du titre (`h1`, `h2`, `h3`, etc.) et, dans notre bloc, attribuer la valeur `darkblue` à la propriété `color`.

```
h1 {  
    color: darkblue;  
}  
  
h2 {  
    color: darkblue;  
}  
  
h3 {  
    color: darkblue;  
}
```

Simplifions notre CSS

Parfois, comme dans l'exemple précédent, plusieurs balises auront le même style. Toutefois, répéter plusieurs fois le même code alourdit le fichier et il s'agit d'une grande source d'erreur de la part de la personne développée. Heureusement pour nous, le CSS supporte une syntaxe « *généreuse* » pour ces cas de figure. Nous pouvons appliquer un bloc de style à plusieurs balises, il suffit de séparer chaque balise par une virgule. La syntaxe générale est alors :

```
balise-1, balise-2, balise-3, /*...*/, balise-n {  
    propriete-1: valeur;  
    /*...*/  
    propriete-n: valeur;  
}
```

Ainsi, notre exemple précédent devient simplement :

```
h1, h2, h3 {  
    color: darkblue;  
}
```

Ordre application des styles

Après notre bloc de style, dans l'exemple précédent, ajoutez le code suivant :


```
h2 {
  color: deeppink;
}
```

Que remarquez-vous ? Réessayez avec plusieurs couleurs différentes l'une à la suite de l'autre. Choisissez des couleurs donc la différence est marquante, personnellement j'ai pris :

```
h1, h2, h3 {
  color: darkblue;
}
```

```
h2 {
  color: deeppink;
}
```

```
h2 {
  color: darkgreen;
}
```

```
h2 {
  color: darkred;
}
```

```
h2 {
  color: goldenrod;
}
```

```
h2 {
  color: grey;
}
```

La couleur qui s'affiche est toujours la dernière indiquée. Il s'agit du **principe de la cascade**. Nous pouvons définir plusieurs fois un bloc de style, mais ce sera toujours la dernière valeur indiquée qui sera sélectionnée. Maintenant, ajoutez le mot clé **!important** après l'une des valeurs, sauf la dernière, et avant le point virgule. N'oubliez pas le ! et d'ajouter un espace entre la valeur et le mot clé. Que remarquez-vous ?

```
h2 {
  color: darkgreen !important;
}
```

Maintenant, ajoutez un deuxième **!important** dans un second bloc, plus bas que votre précédent. Que remarquez-vous ?

```
h2 {
  color: darkgreen !important;
}
```

```
h2 {
    color: darkred;
}

h2 {
    color: goldenrod !important;
}
```

Peut importe si vous ajoutez des valeurs après une propriété avec ce mot clé, le navigateur utilisera la dernière valeur avec le mot clé **!important**. Selon vous, quelle est l'explication ?

Pourquoi **!important** ?

Ce mot clé appose une très haute priorité à votre valeur. Seule une autre propriété avec le mot clé **!important** pourra changer votre valeur à ce moment.

Qu'est-ce qui se passe avec deux fichiers CSS ?

Ajoutez un second fichier CSS à votre projet. Ajoutez-y le code suivant :

```
h2 {
    color: lawngreen !important;
}
```

Pour l'ajouter dans votre page HTML, ajoutez une nouvelle balise `<link>` situé après la précédente. Personnellement, j'ai appelé mon second fichier `app.css`. Mon `<head></head>` est devenu :

```
<head>
    <meta charset="utf-8">
    <title>La Jalousie du Barbouillé</title>
    <link rel="stylesheet" href="./css/style.css">
    <link rel="stylesheet" href="./css/app.css">
</head>
```

Que remarquez-vous ? Qu'est-ce qui se passe si nous inversons les deux balises `<link>` ? Qu'est-ce qui se passe si vous retirez les mots clés **!important** du second fichier (dans l'ordre des `<link>`) ? Par la suite, qu'est-ce qui arrive si nous enlevons tous les **!important** ? Finalement, qu'est-ce qui arrive si nous inversons à nouveau les fichiers ?

Principe de la cascade en CSS

Nous allons établir formellement le **principe de la cascade** en CSS.

Lors de la liaison des styles CSS, le navigateur appliquera la dernière valeur indiquée. Cette valeur est déterminée par le dernier style lu.

De plus, lorsque la page web lie plusieurs feuilles de style, la dernière propriété lue est présente dans la dernière feuille liée.

Bien évidemment, ce principe s'applique aux styles donnés dans les balises `<style></style>`. Il faut voir le contenu de ces balises comme une feuille de style.

Sélecteurs CSS

En plus des **sélecteurs de balise** (par exemple `<h1>`), il existe les sélecteur d'identifiant. Les ***Sélecteur d'identifiant*** sélectionne une balise spécifique par ID, écrivez l'id précédé d'un dièse # - si un élément une balise a été défini avec un id comme ceci : `id="top"` - utilisez le sélecteur `#top` pour sélectionner précisément cette balise

Exemple: ex2.html `<!DOCTYPE html>`

CSS Exemple 2

```
<p>First paragraph.</p>
<p id="second">Second paragraph.</p>
<p>Third paragraph.</p>
</div>

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>CSS Exemple 2</title>
  <style>
    p {
      color: red;
      background-color: yellow;
    }

    #second {
      background-color: lightblue;
    }
  </style>
</head>
<body>
<p>First paragraph.</p>
<p id="second">Second paragraph.</p>
<p>Third paragraph.</p>
</body>
</html>
```

- le sélecteur **p** sélectionne tous les paragraphes
- **#second** sélectionne seulement la balise avec **id="second"**, dans ce cas le deuxième paragraphe
- Notez que les IDs sont uniques dans un document HTML, donc seulement une seule balise peut avoir un ID spécifique
- Par conséquent, un sélecteur d'ID peut correspondre à seulement une balise (si elle existe)
- **Sélecteur de classe** : pour sélectionner les balises d'une classe spécifique (balises avec l'attribut **class** égal au nom d'une certaine classe, par exemple **class="important"**), utilisez le sélecteur **.important**
- **Sélecteur classe-élément** : pour sélectionner des balises d'une classe spécifique, combinez les sélecteurs de classe et de balise :
 - par exemple, pour sélectionner tous les paragraphes de la classe **important**, utilisez le sélecteur **p.important**

Exemple : ex3.html

<!DOCTYPE html>

CSS Exemple 3

```
<h1 class="important">CSS Exemple 3</h1>
<p>First paragraph.</p>
<p class="important">Second paragraph.</p>
<p>Third paragraph.</p>
</div>
```

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>CSS Exemple 3</title>
  <style>
    p {
      color: red;
      background-color: yellow;
    }

    .important {
      background-color: lightgreen;
    }

    p.important {
      border-width: 5px;
    }
  </style>
</head>
<body>
  <div>
```

```

        border-style: solid;
    }
</style>
</head>
<body>
<h1 class="important">CSS Exemple 3</h1>
<p>First paragraph.</p>
<p class="important">Second paragraph.</p>
<p>Third paragraph.</p>
</body>
</html>

```

- `p` correspond à tous les paragraphes
- `.important` correspond à toutes les balises avec `class="important"`
 - dans ce cas-ci, le `<h1>` et le deuxième `<p>`
- `p.important` correspond à tous les paragraphes avec `class="important"`
 - dans ce cas-ci, seulement le deuxième `<p>`
- **Sélecteur de groupe**: pour sélectionner plusieurs balises différentes, séparez les différentes balises avec des virgules
 - par exemple, pour sélectionner plusieurs niveaux de titres, tels que `h1`, `h2` et `h3`, utilisez le sélecteur `h1, h2, h3`
 - ceci peut aider à éviter de répéter les mêmes déclarations pour plusieurs balises différentes

Voici un résumé des sélecteurs importants :

Sélecteur	Exemple	Description
<code>.classe</code>	<code>.introduction</code>	Sélectionne tous les éléments avec la <code>class="introduction"</code>
<code>.classe1.classe2</code>	<code>.section1.section2</code>	Sélectionne tous les éléments avec les deux classes <code>class="section1 section2"</code>
<code>.classe1 .classe2</code>	<code>.section1 .entete</code>	Sélectionne tous les éléments entete qui sont enfants d'un élément section1
<code>#id</code>	<code>#top</code>	Sélectionne l' élément avec <code>id="top"</code>
<code>.class, element</code>	<code>.introduction, p</code>	Sélectionne tous les éléments avec la <code>class="introduction"</code> et tous les éléments <code><p></code>

Sélecteur	Exemple	Description
.class > element	.introduction>p	Sélectionne tous les éléments <p> ayant pour parent immédiat un élément avec la <code>class="introduction"</code>
.class + element	.introduction+p	Sélectionne le premier élément <p> suivant l'élément avec la <code>class="introduction"</code>
.class ~ element	.introduction~p	Sélectionne tous les éléments <p> qui précède un élément ayant la <code>class="introduction"</code>

Feuilles de style externes

- Déplaçons le contenu de la balise <style> dans un fichier css externe, en utilisant la balise <link>

```
<link rel="stylesheet" type="text/css" href="ex5.css">
```
- Le résultat devrait toujours être le même.

Exemple : ex5.html <!DOCTYPE html>

```
<h1>CSS Exemple 5</h1>
<h2>Grouping Selector</h2>
<h3>Apply style to headers 1, 2 and 3</h3>
<h4>Header 4 keeps the default style</h4>
</div>
```

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>CSS Exemple 5</title>
  <link rel="stylesheet" type="text/css" href="ex5.css">
</head>
<body>
<h1>CSS Exemple 5</h1>
<h2>Grouping Selector</h2>
<h3>Apply style to headers 1, 2 and 3</h3>
```

```
<h4>Header 4 keeps the default style</h4>
</body>
</html>
```

Fichier : ex5.css

```
h1, h2, h3 {
    color: red;
    background-color: yellow;
}
```

Les unités de longueur (Référence)

Tout comme les couleurs, il existe de nombreuses unités de longueur en CSS. Chacune d'elle permet de mesurer la taille d'écrire ou d'un trait d'une manière différente. Il existe deux types d'unité les unités relatives et les unités absolues.

Les unités relatives

Les unités relatives sont mesurées par rapport à une référence. Prenez votre écran en exemple. Je pourrais vous donner une unité qui s'appelle écran. Cette unité mesure la longueur des objets par rapport à votre écran. Ainsi, un objet qui a une longueur d'un écran a la même longueur que votre écran. Une longueur de deux écrans correspond à deux fois votre écran. Pour le CSS, nous avons les unités suivantes :

Nom de l'unité	Symbole (utilisé en CSS)	Signification	Équivalent
Pixel	px	Représente la taille selon le nombre de pixels. Cette unité dépend de l'écran ou du média.	
	em	Représente la taille selon la taille font-size du parent. Si la balise parent a une taille de 16px alors 1em vaut ce nombre. Attention à la composition	1em 16px

Nom de l'unité	Symbole (utilisé en CSS)	Signification	Équivalent
	rem	Représente la taille par rapport à l'élément <code><html></html></code> . Si la taille de base du navigateur est 16px alors 1rem vaut ce nombre pour tous les éléments.	
Pourcentage	%	Représente la taille par rapport à l'élément parent. Par exemple, 1 % correspond à 1% <i>taille-parent</i> /100	
	ex	Représente la taille par rapport à la taille de la lettre minuscule x de la police d'écriture utilisée.	

Notez que la taille par défaut du navigateur est généralement de 16px.

Le danger de la composition

L'un des dangers de l'unité **em** est la composition. Pour faire une histoire courte, lorsque nous avons plusieurs balises imbriquées l'une dans l'autre, nous appelons l'ensemble une composition. Le problème de l'unité **em** est qu'elle définit la taille selon la taille d'écriture de son parent. Prenons par exemple le code suivant :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Exemple</title>
    <link href="style.css" rel="stylesheet" >
  </head>
```



```

<body>
<!-- -->
    <h1>Un titre</h1>
    <h2>Un sous-titre</h2>
    <p>
        Ceci est un texte sans être contenu dans une balise <strong>span</strong>.
        <br >
        <span>
            Cette balise contient du texte :<br>
            <span>
                Qui est aussi contenu dans une autre balise.
            </span>
        </span>
    </p>
</body>
</html>

```

Les balises `<p>` et `` imbriquée l'une dans l'autre est une composition. Or, si nous définissons la taille d'écriture des balises `` à 2em . Alors, le texte de la balise `<p>` sera d'une taille normale, le texte de la première balise `` sera le double de la taille de `<p>` (sa balise parente) et le texte de la deuxième balise `` sera le double de son parent **à lui** soit la première balise `` .

```

<!DOCTYPE html>

```

Exemple

```

<style>
    .bigSpan {
        font-size: 2em;
    }
</style>
</head>
<body>
<!-- -->
    <h2>Un titre</h2>
    <h3>Un sous-titre</h3>
    <p>
        Ceci est un texte sans être contenu dans une balise <strong>span</strong>.
        <br >
        <span class="bigSpan">
            Cette balise contient du texte :<br>
            <span class="bigSpan">
                Qui est aussi contenu dans une autre balise.
            </span>
        </span>
    </p>

```

```
</p>
</body>
```

À l'inverse, si nous définissons la taille des balises `` à `2rem` , alors la taille de la deuxième balise sera identique à la première, car la taille est définie à partir de la balise `<html>` et elle est donc globale à toute la page.

```
<!DOCTYPE html>
```

Exemple

```
<style>
    .bigSpan2 {
        font-size: 2rem;
    }
</style>
</head>
<body>
<!-- -->
    <h2>Un titre</h2>
    <h3>Un sous-titre</h3>
    <p>
        Ceci est un texte sans être contenu dans une balise <strong>span</strong>.
        <br >
        <span class="bigSpan2">
            Cette balise contient du texte :<br>
            <span class="bigSpan2">
                Qui est aussi contenu dans une autre balise.
            </span>
        </span>
    </p>
</body>
```

Les unités absolues

Les unités absolues ne dépendent pas des éléments de la page ou de l'écran. Elles seront identiques pour tous les écrans. Parmi ces unités, nous retrouvons les unités de longueur essentiellement.

- `in` représente un pouce soit environ 2.54 centimètres ;
- `cm` représente un centimètre ;
- `mm` représente un millimètre ;
- `pt` représente un point soit, par convention, 1/72 de pouce ;
- `pc` représente un pica soit 12 points

Des propriétés sur le texte

Depuis le début de ce cours, nous parlons de la taille de la police d'écriture (qui la même chose que la taille du texte). Bien que les unités s'appliqueront aussi aux autres propriétés CSS, nous allons commencer par voir comment modifier du texte en CSS.

Exemple HTML de cette section

Pour la prochaine section, nous utiliserons le code HTML suivant :

```
<p>
  <span class="normal">
    Ceci est un texte en HTML modifié par la classe normal.
  </span>
  <br >
  <span class="ex-1">
    Ceci est un texte en HTML modifié par la classe ex-1.
  </span>
  <br >
  <span class="ex-2">
    Ceci est un texte en HTML modifié par la classe ex-2.
  </span>
  <br >
  <span class="ex-3">
    Ceci est un texte en HTML modifié par la classe ex-3.
  </span>
  <br >
  <span class="ex-4">
    Ceci est un texte en HTML modifié par la classe ex-4.
  </span>
  <br >
</p>
```

Prenez note que la classe `normal` correspond uniquement aux valeurs de base du navigateur.

La taille du texte, la propriété `font-size` (Référence)

Cette propriété permet de modifier la taille d'écriture du texte selon la taille donnée. Il existe plusieurs tailles prédéfinies dans le navigateur : *xx-small*, *x-small*, *small*, *medium*, *large*, *x-large* et *xx-large*. Il est aussi possible d'utiliser les mots clés *smaller* et *larger* qui signifie respectivement **plus petit que** et **plus grand que** la taille de l'élément parent.

Par la suite, il est possible d'utiliser l'une des unités précédentes dans le calcul de la taille de notre texte. Prenons l'exemple CSS suivant :

```

.ex {
    font-size: 1ex;
}
.ex-1 {
    font-size: 2em;
}
.ex-2 {
    font-size: 64px;
}
.ex-3 {
    font-size: 150%;
}
.ex-4 {
    font-size: 3ex;
}

```

<!DOCTYPE html>

Exemple

```

<style>
    .ex {
        font-size: 1.5ex;
    }
    .ex-1 {
        font-size: 2em;
    }
    .ex-2 {
        font-size: 64px;
    }
    .ex-3 {
        font-size: 150%;
    }
    .ex-4 {
        font-size: 2.5ex;
    }
</style>
</head>
<body>
    <p class="ex">
        <span class="normal">
            Ceci est un texte en HTML modifié par la classe normal.
        </span>
        <br >
        <span class="ex-1">
            Ceci est un texte en HTML modifié par la classe ex-1.
        </span>
    </p>

```

```

    <br >
    <span class="ex-2">
        Ceci est un texte en HTML modifié par la classe ex-2.
    </span>
    <br >
    <span class="ex-3">
        Ceci est un texte en HTML modifié par la classe ex-3.
    </span>
    <br >
    <span class="ex-4">
        Ceci est un texte en HTML modifié par la classe ex-4.
    </span>
    <br >
</p>
</body>

```

La taille du texte, la propriété font-weight (Référence)

Cette propriété modifie le « poids » de l'écriture. Le poids correspond à l'épaisseur du trait. C'est à-dire que plus le trait est gros, plus le texte est gras. Il existe plusieurs valeurs possibles soit les tailles nommées : *normal*, *bold*, *bolder*, *lighter* ou les nombres 100, 200, 300, 400, 500, 600, 700, 800 et 900. À noter que les tailles 400 et 700 correspondent respectivement à *normal* et *bold*. Il faut aussi noter que ce ne sont pas toutes les polices d'écriture qui supportent toutes les tailles mentionnées.

Prenons l'exemple CSS suivant :

```

.ex2-1 {
    font-weight: 400;
}
.ex2-2 {
    font-weight: bold;
}
.ex2-3 {
    font-weight: 700;
}
.ex2-4 {
    font-weight: 900;
}

```

<!DOCTYPE html>

Exemple

```

<style>
    .ex2 { font-size: 16px; }

```

```

        .ex2-1 {
            font-weight: 400;
        }
        .ex2-2 {
            font-weight: bold;
        }
        .ex2-3 {
            font-weight: 700;
        }
        .ex2-4 {
            font-weight: 900;
        }
    </style>
</head>
<body>
    <p class="ex2">
        <span class="normal">
            Ceci est un texte en HTML modifié par la classe normal.
        </span>
        <br >
        <span class="ex2-1">
            Ceci est un texte en HTML modifié par la classe ex2-1.
        </span>
        <br >
        <span class="ex2-2">
            Ceci est un texte en HTML modifié par la classe ex2-2.
        </span>
        <br >
        <span class="ex2-3">
            Ceci est un texte en HTML modifié par la classe ex2-3.
        </span>
        <br >
        <span class="ex2-4">
            Ceci est un texte en HTML modifié par la classe ex2-4.
        </span>
        <br >
    </p>
</body>

```

Le style du texte, la propriété font-style (Référence)

Cette propriété applique un effet visuel (*style*) au texte. Elle prend trois valeurs possible : *normal*, *italic* et *oblique*. Elle sert principalement à mettre du texte en italique. En général, la valeur *oblique* a le même impact que la valeur *italic*.

Prenons l'exemple CSS suivant :

```
.ex3-1 {  
    font-style: italic;  
}  
.ex3-2 {  
    font-style: oblique;  
}  
.ex3-3 {  
    font-weight: 900;  
    font-style: italic;  
}  
.ex3-4 {  
    font-weight: 900;  
}
```

<!DOCTYPE html>

Exemple

```
<style>  
    .ex3-1 {  
        font-style: italic;  
    }  
    .ex3-2 {  
        font-style: oblique;  
    }  
    .ex3-3 {  
        font-weight: 900;  
        font-style: italic;  
    }  
    .ex3-4 {  
        font-weight: 900;  
    }  
</style>  
</head>  
<body>  
    <p class="ex2">  
        <span class="normal">  
            Ceci est un texte en HTML modifié par la classe normal.  
        </span>  
        <br >  
        <span class="ex3-1">  
            Ceci est un texte en HTML modifié par la classe ex3-1.  
        </span>  
        <br >  
        <span class="ex3-2">  
            Ceci est un texte en HTML modifié par la classe ex3-2.
```

```

    </span>
    <br >
    <span class="ex3-3">
        Ceci est un texte en HTML modifié par la classe ex3-3.
    </span>
    <br >
    <span class="ex3-4">
        Ceci est un texte en HTML modifié par la classe ex3-4.
    </span>
    <br >
</p>
</body>

```

La décoration du texte, la propriété text-decoration (Référence)

Cette propriété permet d'ajouter une décoration au texte. Par les décorations, nous entendons des lignes. Ces lignes peuvent être sous le texte (souligné), par-dessus le texte et dessus le texte (barré). Toutefois, cette propriété possède une syntaxe particulière. Elle est appelée **super propriété**. Pour l'utiliser, la syntaxe est la suivante :

```
text-decoration: text-decoration-line text-decoration-color text-decoration-style;
```

Les valeurs de text-decoration-line (Référence)

Cette propriété permet de définir le type de ligne. Il existe trois possibilités :

Mot clé	Effet	Exemple
overline	Mets une ligne par-dessus le texte.	Cette phrase a une ligne par dessus.
underline	Mets une ligne sous le texte	Cette phrase a une ligne sous.
line-through	Mets une ligne sur le texte	Cette phrase a une ligne sur elle.

De plus, il est possible de combiner plusieurs mots clés ensemble. Pour y parvenir, nous ajoutons les deux mots clés.

```
text-decoration-line: line-through underline;
```






Cette phrase a une ligne sous et sur elle.

Les valeurs de text-decoration-color (Référence)

Cette propriété a été ajoutée avec **CSS3**. Elle permet de modifier la couleur de la décoration. Par défaut, cette valeur est la même que celle du texte. De plus, nous lui donnons les mêmes valeurs que pour la couleur du texte.

```
text-decoration: underline;  
text-decoration-color: pink;
```






Cette phrase a une ligne sous et rose!

					
Navigateur					
Version minimale	57.0	79.0	36.0	12.1	44.0

Les valeurs de text-decoration-style (Référence)

Cette propriété permet de modifier le style de la ligne. Elle a été ajoutée avec le **CSS3** et elle possède cinq valeurs possibles. De plus, elle est toujours utilisée avec une valeur **text-decoration-line**. Par défaut, sa valeur est **solid**.

Mot clé	Effet	Exemple
solid	La ligne sera pleine .	Ma phrase avec décoration.
wavy	La ligne sera ondulée .	Ma phrase avec décoration.
dotted	La ligne sera sous la forme de points .	Ma phrase avec décoration.
dashed	La ligne sera sous la forme de tiret .	Ma phrase avec décoration.
double	La ligne sera doublée .	Ma phrase avec décoration.

					
Navigateur					
Version minimale	57.0	79.0	36.0	12.1	44.0

Exemple en combinant les décorations

```
.ex4-1 {  
    text-decoration: overline green solid;  
}  
.ex4-2 {  
    text-decoration: underline green solid;  
}  
.ex4-3 {
```

```

        text-decoration: underline blue solid;
    }
    .ex4-4 {
        text-decoration: underline #056A92 double;
    }
<!DOCTYPE html>

```

Exemple

```

<style>
    .ex4-1 {
        text-decoration: overline green solid;
    }
    .ex4-2 {
        text-decoration: line-through red solid;
    }
    .ex4-3 {
        text-decoration: underline blue solid;
    }
    .ex4-4 {
        text-decoration: underline #056A92 double;
    }
</style>
</head>
<body>
    <p class="ex2">
        <span class="normal">
            Ceci est un texte en HTML modifié par la classe normal.
        </span>
        <br >
        <span class="ex4-1">
            Ceci est un texte en HTML modifié par la classe ex4-1.
        </span>
        <br >
        <span class="ex4-2">
            Ceci est un texte en HTML modifié par la classe ex4-2.
        </span>
        <br >
        <span class="ex4-3">
            Ceci est un texte en HTML modifié par la classe ex4-3.
        </span>
        <br >
        <span class="ex4-4">
            Ceci est un texte en HTML modifié par la classe ex4-4.
        </span>
        <br >
    </p>

```

```
</p>
</body>
```

La taille du texte, la propriété text-align (Référence)

Cette propriété permet de modifier l'alignement **horizontal** du texte dans son élément parent. Par défaut, le texte est toujours aligné à **gauche** (*left*). À noter qu'il faut un élément qui prend toute la ligne comme un **h1**, **p** ou **div**. Vous pouvez essayer avec un **span** ça ne fonctionnera pas par défaut. Voici les quatre valeurs possibles.

```
.ex5-1 {
    text-align: left; /* La ligne sera alignée sur la gauche */
}
.ex5-2 {
    text-align: right; /* La ligne sera alignée sur la droite */
}
.ex5-3 {
    text-align: center; /* La ligne sera alignée au centre */
}
.ex5-4 {
    text-align: justify; /* La ligne sera justifiée (comme un article de journal) */
}
```

```
<!DOCTYPE html>
```

Exemple

```
<style>
    .ex5-1 {
        text-align: left;
    }
    .ex5-2 {
        text-align: right;
    }
    .ex5-3 {
        text-align: center;
    }
    .ex5-4 {
        text-align: justify;
    }
</style>
</head>
<body>
    <p class="ex2">
        <div class="normal">
```

```

        Ceci est un texte en HTML modifié par la classe normal.
    </div>
    <div class="ex5-1">
        Ceci est un texte en HTML modifié par la classe ex5-1.
    </div>
    <br >
    <div class="ex5-2">
        Ceci est un texte en HTML modifié par la classe ex5-2.
    </div>
    <div class="ex5-3">
        Ceci est un texte en HTML modifié par la classe ex5-3.
    </div>
    <div class="ex5-4">
        Ceci est un texte en HTML modifié par la classe ex5-4.
    </div>
    <span class="ex5-3">
        Ceci est un texte en HTML modifié par la classe ex5-3. Mais pas vraiment, car j
    </span>
</p>
</body>

```

Les ombres sur un texte, la propriété text-shadow (Référence)

Apparue avec le **CSS3**, la propriété **text-shadow** permet d'afficher un effet d'ombre sous le texte. Il utilise la syntaxe suivante :

text-shadow: *horizontal* verticale blur color;

Horizontal détermine le décalage horizontal de l'ombre. Cette valeur est numérique. La valeur négative est acceptée et elle s'effectue sur la gauche.

verticale Détermine le décalage vertical de l'ombre. Cette valeur est numérique. La valeur négative est acceptée et elle s'effectue vers le haut.

blur n'est pas obligatoire, mais elle détermine le dégradé, le temps que la couleur s'efface, de la couleur de l'ombre.

color n'est pas obligatoire, mais elle détermine la couleur de l'ombre.

```

.ex6-1 {
    text-shadow: 1px 1px #FF0000; /* Légère ombre rouge */
}
.ex6-2 {
    text-shadow: 8px 2px 8px #FF0000; /* ombre rouge très diffuse vers la droite */
}
.ex6-3 {

```

```

    /* ombre rouge assez diffuse vers la gauche et le haut*/
    text-shadow: -4px -4px 4px Blue;
}
.ex6-4 {
    /* Deux ombres noires*/
    text-shadow: 1px 1px, 3px 3px;
}
<!DOCTYPE html>

```

Exemple

```

<style>
    .ex6-1 {
        text-shadow: 1px 1px #FF0000;
    }
    .ex6-2 {
        text-shadow: 8px 2px 8px #FF0000;
    }
    .ex6-3 {
        text-shadow: -4px -4px 4px Blue;
    }
    .ex6-4 {
        text-shadow: 1px 1px, 3px 3px;
    }
</style>
</head>
<body>
    <p class="ex2">
        <div class="normal">
            Ceci est un texte en HTML modifié par la classe normal.
        </div>
        <div class="ex6-1">
            Ceci est un texte en HTML modifié par la classe ex6-1.
        </div>
        <br >
        <div class="ex6-2">
            Ceci est un texte en HTML modifié par la classe ex6-2.
        </div>
        <div class="ex6-3">
            Ceci est un texte en HTML modifié par la classe ex6-3.
        </div>
        <div class="ex6-4">
            Ceci est un texte en HTML modifié par la classe ex6-4.
        </div>
    </p>
</body>

```

La police de caractères **font-family** (Référence)

Le choix d'une police de caractères est très important pour un site web. Cela touche l'identité du site, ainsi que la lisibilité. C'est encore plus vrai quand on mélange cela avec les autres modifications de texte que nous avons vues plus haut. Dans le cadre du cours, nous allons utiliser des polices de caractères installées dans votre système. Mais rien n'empêche de charger des polices de caractères.

En CSS, il existe cinq familles de polices génériques :

- Les **polices Serif** ont un petit trait sur les bords de chaque lettre. Ils créent un sentiment de formalité et d'élégance.
- Les **polices sans empattement** ont des lignes épurées (pas de petits traits attachés). Ils créent un look moderne et minimaliste.
- **Polices Monospace** - ici toutes les lettres ont la même largeur fixe. Ils créent un look mécanique.
- Les **polices cursives** imitent l'écriture humaine.
- Les **polices Fantasy** sont des polices décoratives/ludiques.

Les **polices sans empattement** sont considérées comme les plus faciles à lire pour un site web.

La propriété **font-family** est utilisée pour définir la police de caractères. On peut mettre une ou plusieurs polices. La police sera choisie en ordre, selon la disponibilité dans le système de la personne consultant le site web. Généralement, on choisira une police précise jusqu'à une police générique.

```
.ex7-1 {  
    /* La première police est utilisée, sauf si elle n'existe pas dans ce cas, la deuxième s'ajoute  
    font-family: "Times New Roman", Times, serif;  
}  
.ex7-2 {  
    font-family: Arial, Helvetica, sans-serif;  
}  
.ex7-3 {  
    font-family: "Lucida Console", "Courier New", monospace;  
}  
.ex7-4 {  
    font-family: cursive;  
}  
<!DOCTYPE html>
```

Exemple

```

<style>
    .ex7-1 {
        font-family: "Times New Roman", Times, serif;
    }
    .ex7-2 {
        font-family: Arial, Helvetica, sans-serif;
    }
    .ex7-3 {
        font-family: "Lucida Console", "Courier New", monospace;
    }
    .ex7-4 {
        font-family: cursive;
    }
</style>
</head>
<body>
    <p class="ex2">
        <div class="normal">
            Ceci est un texte en HTML modifié par la classe normal.
        </div>
        <div class="ex7-1">
            Ceci est un texte en HTML modifié par la classe ex7-1.
        </div>
        <div class="ex7-2">
            Ceci est un texte en HTML modifié par la classe ex7-2.
        </div>
        <div class="ex7-3">
            Ceci est un texte en HTML modifié par la classe ex7-3.
        </div>
        <div class="ex7-4">
            Ceci est un texte en HTML modifié par la classe ex7-4.
        </div>
    </p>
</body>

```

Exercice 1

Reprenons l'exercice de La Jalousie du Barbouillé. Nous allons styliser le texte.

- Ajouter un fichier `jalousie.css` dans le répertoire du projet de la pièce de théâtre.
- Dans le fichier HTML, ajouter la balise `<link>` dans le `<head>` et lier le fichier `jalousie.css`.
- Sous-lignez tous les titres `<h1>` et `<h2>`. De plus, ajouter une autre couleur et changer le style (ex: dotted, dashed, ...) de la sous-ligne seulement pour

les `<h1>`.

- Ajouter une couleur de fond à l'entête (`id="top"`). Choisissez une couleur pâle comme un bleu (`aliceblue`).
- Modifier la couleur et la grosseur de la police de toutes les occurrences du personnage le **Barbouillé** à l'intérieur des **dialogues** ou des **didascalies**. La couleur peut être un gris foncé et la grosseur peut être 50% plus grosse. Ajouter une ombre de 2 pixels de hauteur et de largeur avec du `blur` de 2 pixels.
- Ajouter une ligne en forme de vague (`wavy`) sous le nom **Barbouillé** à chaque fois que celui-ci est dans une balise ``.
- Modifier la police de caractères de toutes les didascalies pour "Geneva" et si la police n'existe pas, "sans-serif" devra être utilisée.
- Rayez toutes les occurrences en rouge du nom **Docteur** dans les **didascalies** et les **dialogues**.
- Aligner le texte au centre du titre principal.
- Aligner le texte au centre de chaque titre de section (personnages, pièce, ...)

Styler les hyperliens (Référence)

Les hyperliens `<a>` peuvent être stylisés avec les mêmes propriétés que le texte comme `color`, `background`, `font-family`, ...

Par contre, il existe des pseudo-classes qui permettent de modifier les hyperliens dans différents états.

Voici les quatre états de hyperliens sont : - `a:link` - un lien normal non visité - `a:visited` - un lien que l'utilisateur a visité - `a:hover` - un lien lorsque l'utilisateur passe la souris dessus - `a:active` - un lien au moment où il est cliqué

L'ordre de définition est important! `a:hover` doit venir après `a:link` et `a:visited` `a:active` doit venir après `a:hover`.

La raison est que l'état `a:link` englobe `a:hover`. Même chose pour l'état `a:visited`. Alors, `a:hover` ne servirait à rien si elle est définit avant.

Regardons un exemple intéressant qui change le style par défaut des hyperliens.

```
.ex8 a {
  outline: none;
  text-decoration: none;
  padding: 2px 1px 0;
}

.ex8 a:link {
  color: #265301;
}
```



```

.ex8 a:visited {
    color: #437A16;
}

.ex8 a:focus {
    border-bottom: 1px solid;
    background: #BAE498;
}

.ex8 a:hover {
    border-bottom: 1px solid;
    background: #CDFEAA;
    cursor: progress;
}

.ex8 a:active {
    background: #265301;
    color: #CDFEAA;
}

```

<!DOCTYPE html>

Exemple

```

<style>

.ex8 a {
    outline: none;
    text-decoration: none;
    padding: 2px 1px 0;
}

.ex8 a:link {
    color: #265301;
}

.ex8 a:visited {
    color: #437A16;
}

.ex8 a:focus {
    border-bottom: 1px solid;
    background: #BAE498;
}

.ex8 a:hover {
    border-bottom: 1px solid;
    background: #CDFEAA;
    cursor: progress;
}

.ex8 a:active {
    background: #265301;
}

```

```

        color: #CDFEAA;
    }
</style>
</head>
<body>
    <div class="ex8">
        <a href="https://www.youtube.com/watch?v=L8QCAt0jtpM">Un lien intéressant!</a>
    </div>
        <a href="https://www.youtube.com/watch?v=L8QCAt0jtpM">Un lien standard</a>
    <p>Mettez votre souris au dessus des liens!</p>
</body>

```

Avez-vous remarqué la propriété `cursor`. Elle permet de modifier le style du curseur. Consulter ce lien pour en savoir plus sur les valeurs possibles de `cursor`.

Styliser les listes (Référence)

Les listes `` et `` peuvent utiliser les mêmes propriétés que le texte. Il y a quand même quelques propriétés de plus à connaître.

- `line-height` Modifie l'espace entre les éléments.
- `list-style-type` définit le type de puce pour les listes non ordonnées ou bien des nombres, des lettres ou des chiffres romains pour une liste ordonnée.
- `list-style-position` définit si les puces seront à l'intérieur ou à l'extérieur du début de l'élément de liste.
- `list-style-image` permet d'utiliser une image à la place des puces. Mais l'utilisation des propriétés `background` est généralement préférée.
- `list-style` peut être utilisé pour regrouper `list-style-type`, `list-style-position` et `list-style-image` dans une même propriété.

```

.ex9 ul {
    background: #3399ff;
}

```

```

.ex9 ul li {
    background: #cce5ff;
    color: darkblue;
    margin: 5px;
}

```

```

.ex10 ul {
    list-style-type: square;
    list-style-position: inside;
    line-height: 100%;
}

```

<!DOCTYPE html>

Exemple

```
<style>
    .ex9 ul {
        background: #3399ff;
        border-left: 5px solid red;
        background-color: #f1f1f1;
        list-style-type: none;
    }
    .ex9 ul li {
        color: darkblue;
        margin: 5px;
    }
    .ex10 ul {
        list-style-type: square;
        list-style-position: inside;
        line-height: 100%;
    }
</style>
</head>
<body>
    <div>
        <ul>
            <li>Café</li>
            <li>Thé</li>
            <li>Boisson</li>
        </ul>
    </div>
    <div class="ex9">
        <ul>
            <li>Café</li>
            <li>Thé</li>
            <li>Boisson</li>
        </ul>
    </div>
    <div class="ex10">
        <ul>
            <li>Café</li>
            <li>Thé</li>
            <li>Boisson</li>
        </ul>
    </div>
</body>
```

Exercice 2

Reprenons l'exercice de La Jalousie du Barbouillé. Nous allons styliser les liens et les listes.

- Modifier tous les hyperliens pour être 1,25 fois plus gros que le texte standard.
- Enlever le sous-ligné sous les hyperliens par défaut, mais le sous-ligné doit rester, si la souris **est au dessus** de l'hyperlien (**hover**, **focus**).
- Changer la couleur de fond des hyperliens quand la souris est au dessus du lien (**hover**, **focus**) pour un mauve pâle (ex: #e6aafe).
- Changer la couleur de l'hyperlien quand celui-ci est visité (**visited**) pour un mauve foncé (ex: #8222b9).
- Changer le type du curseur (**cursor**) quand la souris est au dessus d'un hyperlien (**hover**).
- Inversez la couleur de l'hyperlien avec la couleur de fond quand celui-ci est actif (**active**). Par exemple, mettre la couleur de fond mauve foncé et la couleur du lien en mauve pâle.
- Choisissez les deux hyperliens “Scène première” et “Scène deuxième” dans la table des matières et changer les couleurs pour orange pâle et orange foncé à la place du mauve.
- Modifier les puces de la liste des personnages pour des carrés.
- Ajouter la même couleur de fond que l'entête pour chaque élément de la liste.
- Modifier la couleur de fond de la section **liste-perso** pour une couleur plus foncée comme un bleu “cornflowerblue”.
- Modifier la grosseur de la police de caractères de la table des matières pour que le premier niveau de la liste soit 50% plus gros que le texte standard. Pour le deuxième niveau d'éléments de la liste, ce doit être 25% plus petit que le premier niveau. (Indice: **em**)
- Enlever les puces pour la liste de la table des matières

Notes écrites par Godefroy Borduas, modifiées par Denis Rinfret et Pierre-Luc Boulanger.