

Université de Neuchâtel

Master en littératures

TG: Histoire du livre et sa troisième révolution (numérique)

Cours 7: CSS (bases)

Élodie Paupe

chaire de philologie classique et d'histoire ancienne

19 avril 2021

Objectifs du jour:

- Balises supplémentaires
- Mise en forme avec CSS (bases)
- Mise en forme avec CSS (avancé)

- Pour les démonstrations faites en commun, nous utilisons le contenu du dossier `\cours7\dossier_exercice`
 - `accueil.html`
 - `exercice.html`
 - `contact.html`
 - `stylesheet.css`
 - différents fichiers `.jpeg`
- Les exercices figurent dans le dossier `\cours7\exercices`

Balises supplémentaires

- `<header>` est l'élément qui contient un contenu introductif ou une série de liens pour la navigation, par exemple des titres, un logo, des informations sur l'auteur ou l'autorité responsable de la page.
 - `<nav>` est l'élément qui contient un menu de navigation. Il s'utilise en complément avec `<a>`. Les liens sont présentés dans une liste avec ``.

```
<nav id="primary_nav_wrap">
  <ul>
    <li><a href="accueil.html">Accueil</a> </li>
    <li><a href="exercice.html">Voyage en Italie</a></li>
    <li><a href="contact.html">Contact</a></li>
  </ul>
</nav>
```

- `` pour un insérer un logo (on le fera tout à la fin)

- `<hr>` s'emploie sans balise fermante et annonce un changement de contenu dans la page. Sa réalisation graphique est une ligne horizontale (*horizontal rule*).

- `<pre>` permet d'introduire du texte préformaté: il conservera ainsi les retours à la ligne, les tabulations, etc. Sa réalisation graphique se fait en police Courier.

- `<iframe>` crée un cadre dans la page qui peut contenir une autre page web. Cette balise peut supporter les mêmes attributs que ``.

```
<iframe width="560" height="315" src="https://www.youtube.com/embed/-gy1j00Tal8"
title="YouTube video player" frameborder="0"
allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture" allowfullscreen></iframe>
```

```
<iframe src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d96569.9972693191!2d14.176562366742917!3d40.85404211399215!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0x133b0866db7afaeb%3A0xd23a43cc658cb87e!2sNaples%2C%20Italia!5e0!3m2!1sfr!2sfr!4v1618817841697!5m2!1sfr!2sfr" width="600" height="450" style="border:0;" allowfullscreen="" loading="lazy"></iframe>
```


- `<figure>` et `<figcaption>` pour ajouter une légende à une photo:

```
<figure>
  <img src="" alt="" width="">
  <figcaption>La légende</figcaption>
</figure>
```

- `<footer>` est l'élément qui contient le pied de page d'un document ou d'une partie du document, notamment les informations relatives au copyright, des moyens de contact, le plan du site, etc.
 - `©` est l'entité HTML pour faire apparaître ©
 - Pour inclure un lien rapide vers la composition d'un message à partir d'une adresse électronique, on utilise ``

Rendu final sans CSS.

Balises de présentation

- `` : texte en gras
- `<i>` : texte en italique
- `<mark>` : texte marqué
- `<small>` : texte plus petit
- `` : texte effacé
- `<ins>` : texte inséré
- `<sub>` : texte en indice
- `<sup>` : texte en exposant

Exercise 1

Deux niveaux d'éléments

- Éléments de niveau bloc: ils commencent à une nouvelle ligne, comme
 - `<header>`
 - `<h1>` - `<h6>`
 - `<p>`
 - `<pre>`
 - `<hr>`
 - `` et ``
 - `<div>` : cette balise permet de diviser un document HTML.
- Éléments de niveau *inline*: ils ne commencent pas une nouvelle ligne. Les balises vues dans la diapositive précédentes en font partie.

Commenter son code

Lorsqu'on code, on souhaite parfois laisser des informations dans le code qui ne seront pas visibles sur le navigateur internet.

Ces commentaires sont destinées aux personnes qui liront le code source, le développeur d'un site ou d'un livre numérique. Ils peuvent justifier certains choix d'encodage, signaler une information importante, indiquer la structure d'un document complexe, etc.

Pour commenter le code `HTML`, on utilise la syntaxe suivante:

```
<!-- Voilà, ici, je place un commentaire. -->
```

Objectifs du jour:

- Balises supplémentaires ✓
- Mise en forme avec CSS (bases)
- Mise en forme avec CSS (avancé)

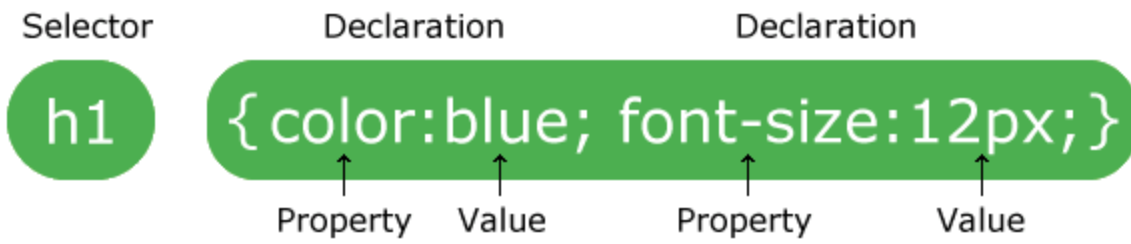
Mise en force avec CSS

Syntaxe de base

Le langage CSS s'exprime de la façon suivante:

```
body {  
  background-color: lightblue;  
}
```

On sélectionne la balise HTML concernée, les informations de mise en forme sont données ensuite entre accolades: pour chaque propriété, une ou plusieurs valeurs sont données.



[Source](#)

On peut également sélectionner les éléments en fonction de l'attribut `class` qu'on a fait porter aux éléments HTML.

Ainsi, si on a le code HTML suivant:

```
<div class="poem">  
  <p>blablabla</p>  
</div>  
<div class="note">  
  <p>blablabla</p>  
</div>
```

On pourra commander de façon différentes la mise en forme des deux sections dans le CSS en sélectionnant la `class` :

```
.poem {  
  text-align: center;  
  font-size: 12pt;  
}  
  
.note {  
  text-align: left;  
  font-size: 10pt;  
}
```

On peut encore sélectionner:

- un élément porteur d'un attribut `class` spécifique. Pour `<p class="red"> : p.red`
- un élément porteur d'un attribut `id` (= un seul élément du fichier). Pour ` : #logo`
- tous les éléments: `*`
- certaines éléments: `p, div`

On peut déclarer la mise en forme à différents endroits.

- Interne = à un seul endroit dans le document HTML
- Externe = dans un autre fichier
- *Inline* = dans le code HTML

La syntaxe qui vient d'être présentée est utilisée telle quelle dans le cas d'un document interne ou externe. Elle est raccourcie dans le cas d'un usage interne.

Interne et la balise `<style>`

La feuille de style encodée en CSS est placée au début du document dans le `<head>` .

Son contenu est placée dans la balise `<style>` :

```
<head>
  <title>Voyage en Italie</title>
  <meta charset="utf-8">
  <style>
    body {
      background-color: linen;
    }

    h1 {
      color: maroon;
      margin-left: 40px;
    }
  </style>
</head>
```

Externe

La feuille de style CSS n'est pas contenue dans le document HTML, mais dans un fichier à part qui peut donc être commun à plusieurs documents HTML.

On relie chaque fichier HTML à sa feuille de style CSS en ajoutant l'élément suivant dans la balise `<head>` :

```
<link rel="stylesheet" href="mystyle.css">
```

Où on remplace *mystyle.css* par le nom de votre fichier, enregistré en .css.

Inline et l'attribut `style`

Les informations de mise en forme sont placées directement dans la balise grâce à l'attribut `style` :

```
<body style="background-color: linen;">  
  <h1 style="color: maroon; margin-left: 40px;">Titre de niveau 1</h1>  
  <p>Un paragraphe</p>  
</body>
```

Quelques propriétés de couleurs

- `background-color` pour définir la couleur d'arrière-plan
- `color` pour définir la couleur du texte
- `border-color` pour définir la couleur d'une bordure

Appeler des couleurs en HTML:

- à l'aide de **140 noms de couleurs**
- à l'aide de valeurs
 - **RGB**: une valeur à trois composante qui mélange les trois couleurs à des degrés divers avec la syntaxe suivante `rgb(x, x, x)` où x est un nombre entre 0 et 255
 - **HEX**: une valeur hexadécimale définit la couleur avec la syntaxe suivante `#RRGGBB`
 - HSL, RGBA ou HSLA

Exercices 2 et 3

- `background`
 - `background-image` pour définir une image d'arrière-plan avec la syntaxe:
`background-image: url("IMG_7300.JPG");`
 - `background-repeat` , `background-attachment` , `background-position`
- `border`
 - `border-style` pour définir le type de bordure: `dotted`, `dashed`, `solid`, `double`, `groove`, `ridge`, `inset`, `outset`, `none`, `hidden`
 - on peut préciser `border-top-style` ou `right` , `bottom` , `left`
 - `border-width` pour définir l'épaisseur de la bordure avec des valeurs
 - spécifiques: `px`, `pt`, `cm`, `em`, etc.
 - pré-définies: `thin`, `medium`, `thick`
 - `border-radius` pour arrondir les angles de la bordure (valeur en `px` ou en `%`)

Police de caractère

- En CSS, on considère 5 familles:
 - serif
 - sans-serif
 - monospace
 - cursive
 - fantasy



Sans-serif



Serif



Serif
(red serifs)

Generic Font Family	Examples of Font Names
Serif	Times New Roman Georgia Garamond
Sans-serif	Arial Verdana Helvetica
Monospace	Courier New Lucida Console Monaco
Cursive	<i>Brush Script MT</i> <i>Lucida Handwriting</i>
Fantasy	COPPERPLATE Papyrus

Habituellement, on déclare également une ou deux polices "de secours" au cas où la première n'était pas lisible. On peut aussi déclarer une famille.

Les polices les plus courantes et donc les plus sûres sont:

- Arial (sans-serif)
- Verdana (sans-serif)
- Helvetica (sans-serif)
- Tahoma (sans-serif)
- Trebuchet MS (sans-serif)
- Times New Roman (serif)
- Georgia (serif)
- Garamond (serif)
- Courier New (monospace)
- Brush Script MT (cursive)

- `font` permet d'agir sur la police de caractère.
 - `font-family` pour définir la police
 - `font-size` pour définir la taille de la police
 - avec des valeurs absolues: par exemple la taille en px
 - avec des valeurs relatives: par exemple la taille en em ou en pourcentages
 - `font-style` pour définir l'italique avec les valeurs `normal`, `italic` ou `oblique`

Exercise 4

L'exercice précédent a montré que, par défaut, un élément de niveau bloc (par exemple `<h1>`) prend toute la place à disposition, ce qui n'est pas très esthétique.

On peut définir une largeur absolue ou relative avec la propriété `width` .

- `width: 1000px` créera un élément conteneur d'une largeur fixe de 1000 px, ce qui peut poser problème pour consulter le site sur un écran de petite taille.
- `width-max: 1000px` créera un élément conteneur d'une largeur maximale de 1000 px.

Ces valeurs doivent être définies en fonction des appareils qui vont servir à consulter le document.

Pour le moment, le contenu est encore collé à gauche. On va ajouter des marges au document pour qu'il soit centré sur l'écran. Pour cela, on fait appel à la propriété `margin`. La valeur s'exprime en longueur spécifique (px, pt, cm, em, etc.), en pourcentage ou se calcule automatiquement (`auto`).

- On peut viser spécifiquement certaines marges (`margin-top` , `margin-right` , `margin-bottom` , `margin-left`).
- On peut aussi travailler sur les marges globalement (`margin`).

Exercise 5

- `text-align` pour définir l'alignement horizontal du texte avec les valeurs `center`, `left`, `right` ou `justified`
- `vertical-align` pour définir la place d'un contenu dans un élément. La propriété peut prendre les valeurs `top`, `middle` ou `bottom`
- `text-decoration` pour définir une ligne au-dessus (`overline`), un soulignement (`underline`), une ligne par-dessus (`line-through`)
 - il est possible de préciser la nature du trait (`dotted`, `wavy`, `dashed`)
 - il est possible de préciser la couleur

On pourra ainsi avoir `style="text-decoration: darkslateblue dashed underline"`

- `text-transform` pour contrôler la dimension majuscule (`uppercase`) ou minuscule (`lowercase`). `capitalize` ajoute une majuscule au début de chaque mot.
- `text-indent` pour ajouter une indentation à la première ligne du texte. La valeur est donnée en px.
- `text-shadow` pour ajouter une ombre au texte. On joindra trois valeurs: deux valeurs chiffrées en px qui donnent l'ombre horizontale et verticale + la couleur. Si on ajoute une troisième valeur numérique, on obtiendra une ombre plus diffuse.
 - `text-shadow: darkslateblue 3px 3px 20px`

- `letter-spacing` pour augmenter/diminuer l'espace entre les lettres. La valeur est donnée en px (par exemple 5px ou -5px)
- `word-spacing` pour augmenter/diminuer l'espace entre les mots.

- `line-height` pour définir la hauteur des lignes (mais pas la taille de la police!).

Exercise 6

Si des informations contradictoires sont données parce que les indications se multiplient, la priorité est fixée ainsi:

1. Inline
2. Externe ou interne (la dernière information encodée dans le head)
3. Aucune mise en forme et rendu HTML

Exercice 7

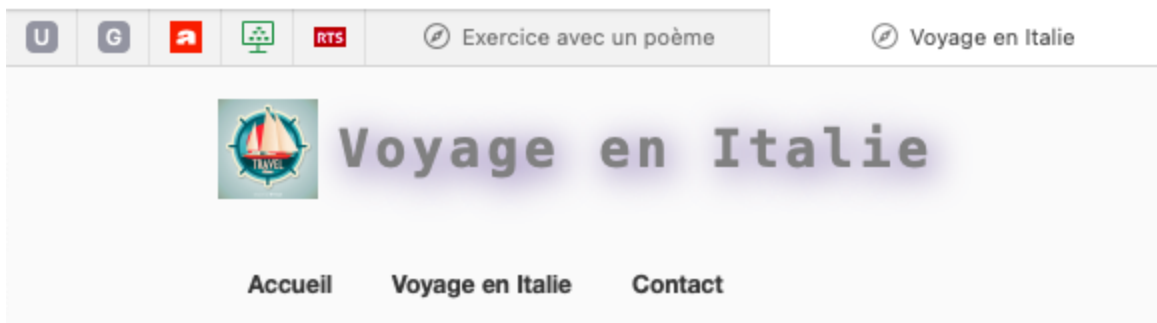
Créons une feuille de style pour le blog de voyage.

Objectifs du jour:

- Balises supplémentaires ✓
- Mise en forme avec CSS (bases) ✓
- Mise en forme avec CSS (avancé)

Transformer un menu `<nav>` grâce au CSS

```
<nav id="primary_nav_wrap">
  <ul>
    <li><a href="accueil.html">Accueil</a> </li>
    <li><a href="exercice.html">Voyage en Italie</a></li>
    <li><a href="contact.html">Contact</a></li>
  </ul>
</nav>
```



```
#primary_nav_wrap
{
  margin-top:15px;
  display:flex;
  height: 50px;
}

#primary_nav_wrap ul
{
  list-style: none;
  position: relative;
  float: left;
  margin: auto 5px auto;
  padding:0;
}

#primary_nav_wrap ul a
{
  display:block;
  color:#333;
  text-decoration:none;
  font-weight:700;
  font-size:12px;
  line-height:32px;
  padding: 0 15px ;
  font-family:"HelveticaNeue","Helvetica Neue",Helvetica,Arial,sans-serif
}

#primary_nav_wrap ul li
{
  position:relative;
  float:left;
  margin:0;
  padding:0;
}

#primary_nav_wrap ul li:hover
{
  background:#ddd
}
```


Décortiquons pas à pas

On peut commenter son code CSS à l'aide de la syntaxe suivante:

```
/* Ceci est un commentaire */  
a {  
  color: green;  
}
```

```
#primary_nav_wrap
{
  margin-top:15px;
  display:flex;
  height: 50px;
}
```

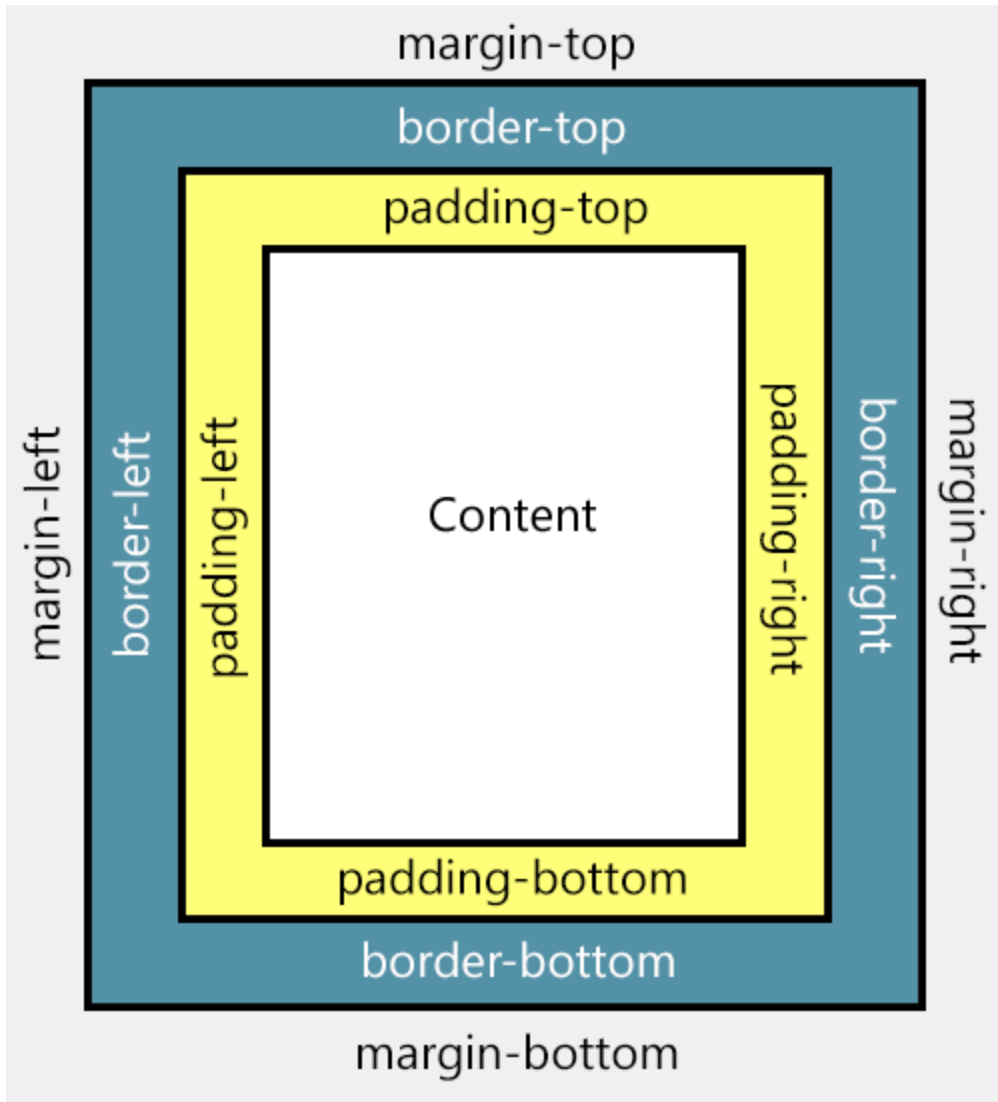
- La propriété `display` précise si et comment un élément doit être affiché (`inline` , `block` , etc.)

```
#primary_nav_wrap ul
{
    list-style: none;
    position: relative;
    float: left;
    margin: auto 5px auto;
    padding: 0;
}
```

- `position` permet de définir la position de l'élément sur la page (`static` , `relative` , `absolute` , `fixed` ou `sticky`)
- `margin` permet de spécifier les marges. La valeur peut être absolue (px, cm, etc.) ou un pourcentage de l'élément qui contient

Il est possible d'éviter l'emploi de `margin-left/bottom/right/top`

- `margin: 12px` = les quatre marges ont 12px
- `margin: 12px 20px` = la marge du haut et du bas ont 12px et celles de gauche et droite en ont 20.
- `margin: 12px` = les quatre marges ont 12px
- `margin: 12px 20px 6px` = la marge du haut a 12px, celles de gauche et droit 20, celle du bas 6
- `margin: 12px 20px 6px 50px` = la marge du haut a 12px, celle de droite 20, celle du bas 6px, celle de gauche 50px



Source: Jecinta Morgan

```
#primary_nav_wrap ul a
{
    display:block;
    color:#333;
    text-decoration:none;
    font-weight:700;
    font-size:12px;
    line-height:50px;
    padding: 0 15px;
    font-family:"HelveticaNeue","Helvetica Neue",Helvetica,Arial,sans-serif
}
```

- `padding` fonctionne de la même façon que `margin` pour les valeurs.

```
#primary_nav_wrap ul li
{
    position:relative;
    float:left;
    margin:0;
    padding:0;
}
```

- `float` permet de définir comme un élément s'intègre dans une page (par exemple une image dans le texte)

```
#primary_nav_wrap ul li:hover  
{  
    background:#ddd  
}
```


Ajouter un logo au menu

Logo = une image spéciale avec un comportement unique qui n'apparaît qu'une seule fois dans la page.

- Où la placer dans le code?
- Comment la sélectionner?

```
<header>
  
  <nav id="primary_nav_wrap">
    <ul>
      <li><a href="accueil.html">Accueil</a> </li>
      <li><a href="exercice.html">Voyage en Italie</a></li>
      <li><a href="contact.html">Contact</a></li>
    </ul>
  </nav>
  <h1>Voyage en Italie</h1>
</header>
```



```
<header>
  
  <h1>Voyage en Italie</h1>
  <nav id="primary_nav_wrap">
    <ul>
      <li><a href="accueil.html">Accueil</a> </li>
      <li><a href="exercice.html">Voyage en Italie</a></li>
      <li><a href="contact.html">Contact</a></li>
    </ul>
  </nav>
</header>
```



```
#logo{  
  width: 50px;  
  height: 50px;  
  float: left;  
  margin-right: 10px;  
  margin-left: 5px;  
}
```

Objectifs du jour:

- Balises supplémentaires ✓
- Mise en forme avec CSS (bases) ✓
- Mise en forme avec CSS (avancé)

Devoirs pour la semaine prochaine: améliorer votre anthologie

À voir la fois prochaine

- Citations avec `<q>` , `<blockquote>` et `<cite>`
- Insérer un tableau
- Insérer du son ou de la vidéo

Des envies ou des besoins?