

# Intégration Web

MMI 1 – TP#4 S1



Danielo **JEAN-LOUIS**  
Michele **LINARDI**

# Formatage du texte - Généralités

- Fonctionne sur toutes les formes de texte
  - Balises : `<a>`, `<p>`, `<span>`, `<li>`...
- Permet de structurer, hiérarchiser l'information
  - Rend l'information plus simple à trouver

## Sources :

- [https://developer.mozilla.org/fr/docs/Learn/HTML/Introduction\\_to\\_HTML/HTML\\_text\\_fundamentals](https://developer.mozilla.org/fr/docs/Learn/HTML/Introduction_to_HTML/HTML_text_fundamentals)

# Hiérarchie du texte - Généralités

- Fonctionne via le CSS pour les internautes voyants
- Fonctionne via le HTML (sémantique) pour les internautes non-voyants

## Sources :

- [https://developer.mozilla.org/fr/docs/Learn/HTML/Introduction\\_to\\_HTML/HTML\\_text\\_fundamentals](https://developer.mozilla.org/fr/docs/Learn/HTML/Introduction_to_HTML/HTML_text_fundamentals)

# Formatage du texte

- Exclusivement fait grâce aux propriétés css
  - color : couleur du texte
  - font-weight : graisse du texte
  - font-family : police du texte
  - font-size : taille du texte
  - text-align : alignement du texte
  - text-decoration : "décore" le texte
  - text-transform : change la casse du texte
  - [...]

## Sources :

- [https://developer.mozilla.org/fr/docs/Web/CSS/CSS\\_Text](https://developer.mozilla.org/fr/docs/Web/CSS/CSS_Text)
- <https://developer.mozilla.org/fr/docs/Web/CSS/text-decoration>
- [https://developer.mozilla.org/fr/docs/Learn/CSS/Styling\\_text/initiation-mise-en-forme-du-texte](https://developer.mozilla.org/fr/docs/Learn/CSS/Styling_text/initiation-mise-en-forme-du-texte)
- <https://cssgradient.io/blog/css-gradient-text/> - texte en dégradé - en

# Point accessibilité – Casse du texte

- Évitez les paragraphes écrits en majuscules, ceci rend la lecture compliquées pour les personnes dyslexiques notamment
  - Limiter les majuscules aux titres ou textes courts

## Source(s) :

- <https://developer.mozilla.org/fr/docs/Web/CSS/text-transform#accessibilit%C3%A9>

# Formatage du texte - font-weight

- Accepte comme valeur :
  - Nombres compris en 100 et 900 définis
    - Changement de la règle avec CSS4 → Possibilité de mettre une valeur décimale
  - normal (400) (valeur par défaut) / bold (700)
  - lighter / bolder, valeurs relatives au parent

## Source(s) :

- <https://developer.mozilla.org/fr/docs/Web/CSS/font-weight>

# Formatage du texte - font-family

- Permet de changer la police du texte
- Accepte comme valeur une liste de polices ordonnée par priorité, la police prioritaire étant la plus à gauche
  - Exemple : **Cambria**, Cochin, Georgia, Times, 'Times New Roman', **serif**;

## Sources :

- <https://developer.mozilla.org/fr/docs/Web/CSS/font-family>
- <https://www.cssfontstack.com/> - liste des polices "sûres"
- <https://developer.mozilla.org/fr/docs/Web/CSS/@font-face>



# Formatage du texte – serif ou sans-serif

- Valeurs de secours si celles précédentes ne fonctionnent pas
  - On met l'un OU l'autre à la fin de la liste de polices
- Indispensable pour éviter d'avoir de mauvaises surprises

## Sources :

- <https://developer.mozilla.org/fr/docs/Web/CSS/font-family>
- <https://www.cssfontstack.com/> - liste des polices "sûres"
- <https://developer.mozilla.org/fr/docs/Web/CSS/@font-face>

# Formatage du texte – serif ou sans-serif



## **Sans-serif (sans empattement)**

- Représente la modernité
- Exemple de police : Helvetica, Futura, Arial...
- Plus présent sur le web

## **Serif (avec empattement)**

- Représente le sérieux, le professionnalisme
- Arrivé à l'époque de Gutenberg
- Exemple de polices : "Times New Roman", Georgia ...

## **Sources :**

- <https://www.adobe.com/fr/creativecloud/design/discover/serif-vs-sans-serif.html>
- <https://www.get-beyond.fr/serif-sans-serif-difference/>
- <https://www.canva.com/learn/serif-vs-sans-serif-fonts/> - anglais

# Formatage du texte - font-family

- Par défaut, accès uniquement à des polices système
- Utilisation de la règle @font-face pour utiliser des polices non système

## Sources :

- <https://developer.mozilla.org/fr/docs/Web/CSS/font-family>
- <https://www.cssfontstack.com/> - liste des polices "sûres"
- <https://developer.mozilla.org/fr/docs/Web/CSS/@font-face>

# Formatage du texte - font-family

- Listez des polices cohérentes dans votre propriété
  - Ne mettez pas "Comic Sans MS" et "Impact" ensemble, par exemple
  - Note : VS Code propose des polices qui vont ensemble à l'auto-complétion

## Sources :

- <https://developer.mozilla.org/fr/docs/Web/CSS/font-family>
- <https://www.cssfontstack.com/> - liste des polices "sûres"
- <https://developer.mozilla.org/fr/docs/Web/CSS/@font-face>

# Formatage du texte - font-family



**Abc**

Impact

Abc

Comic Sans MS

Abc

Arial

Les trois polices sont à 40px  
pourtant elles n'occupent pas le  
même espace d'où l'importance  
d'avoir une liste de polices  
cohérentes pour la valeur de la  
propriété "font-family"

## Sources :

- <https://developer.mozilla.org/fr/docs/Web/CSS/font-family>
- <https://www.cssfontstack.com/> - liste des polices "sûres"
- <https://developer.mozilla.org/fr/docs/Web/CSS/@font-face>

# Variables / Propriétés personnalisées CSS

- Éviter la redondance des **valeurs de propriétés**
- Géré par tous les navigateurs modernes

## Sources :

- [https://developer.mozilla.org/fr/docs/Web/CSS/Using\\_CSS\\_custom\\_properties](https://developer.mozilla.org/fr/docs/Web/CSS/Using_CSS_custom_properties)

# Variables / Propriétés personnalisées CSS

```
.mon-selecteur {  
    /* Définition de la variable */  
    --couleur-marque: rgb(14, 25, 90);  
  
    /* utilisation de la variable */  
    color: var(--couleur-marque, blue);  
}
```

fonction "var" pour pouvoir utiliser  
les propriétés personnalisées

propriété  
personnalisée  
définie avant

valeur par défaut si  
la propriété n'existe  
pas (facultatif)

une variable CSS

## Sources :

- [https://developer.mozilla.org/fr/docs/Web/CSS/Using\\_CSS\\_custom\\_properties](https://developer.mozilla.org/fr/docs/Web/CSS/Using_CSS_custom_properties)

# Variables / Propriétés personnalisées CSS

- Les variables possèdent une portée (scope) définie par le sélecteur CSS
- Utilisation de la pseudo-classe ":root" pour rendre la variable globale

```
:root {  
  --couleur-arriere-plan: blue;  
}  
  
body {  
  --couleur-verte: rgb(0, 201, 50);  
  background-color: var(--couleur-arriere-plan, blue);  
  color: var(--couleur-verte);  
  border: 3px solid var(--couleur-verte);  
}
```

La variable est globale car définie dans ":root"

## Sources :

- [https://developer.mozilla.org/fr/docs/Web/CSS/Using\\_CSS\\_custom\\_properties](https://developer.mozilla.org/fr/docs/Web/CSS/Using_CSS_custom_properties)



# Formatage du texte - font-family

```
:root {  
  --liste-polices: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;  
}  
  
.article-contenu {  
  font-family: var(--liste-polices);  
  /* [...] */  
}  
  
.article-titre {  
  font-family: var(--liste-polices);  
  /* [...] */  
}
```

```
.article-contenu {  
  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;  
  /* [...] */  
}  
  
.article-titre {  
  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;  
  /* [...] */  
}
```

Les deux codes font la même chose mais l'utilisation de variables rend le code plus facile à modifier

## Sources :

- <https://developer.mozilla.org/fr/docs/Web/CSS/font-family>
- <https://www.cssfontstack.com/> - liste des polices "sûres"
- <https://developer.mozilla.org/fr/docs/Web/CSS/@font-face>

**Vous n'êtes pas obligé d'utiliser  
des variables, c'est pour votre  
confort de travail**

## font-family – Bonnes pratiques

- Évitez de mélanger trop de polices d'écriture au sein même d'un site  
**Deux ou trois** polices différentes maximum
- Pensez aux dyslexiques, préférez les polices simples à celles (trop) stylisées

### Sources :

- <https://developer.mozilla.org/fr/docs/Web/CSS/font-family>

# Formatage du texte - font-size

- Définit la taille du texte
- Accepte comme valeurs (toujours positives) :
  - Pourcentage (%)
  - Longueur, unités possibles :
    - px
    - rem
    - em
    - ...

## Sources :

- <https://developer.mozilla.org/fr/docs/Web/CSS/font-size>
- <https://developer.mozilla.org/fr/docs/Web/CSS/length>

# Point accessibilité – Taille de texte

- 16 px est considéré comme étant la taille recommandée pour un corps de texte sur le web
  - En dessous ça devient compliqué à lire
- Préférer les tailles de polices relatives comme le rem, le % ou l'em. **Mais le rem reste moins prise de tête pour les calculs**
  - Ne pas oublier que certains utilisateurs peuvent zoomer ou même changer la taille de base de la font-size du navigateur

# Formatage du texte – rem / em / %

- Unités relatives
- rem se base sur l'élément racine (r pour root) de la page, souvent la font-size définie sur <html> avec une valeur de 16px
- em sera plus utilisé pour les marges notamment de la balise <p>
  - Se base sur la font-size du parent direct, de même pour l'unité pourcentage

## Sources :

- <https://24ways.org/2006/compose-to-a-vertical-rhythm/> - en

## Formatage du texte – rem

- 1 rem représente 100 % de la valeur de la font-size du root (racine)
- ~~Balise <html>~~ - Préférez la valeur du navigateur
- Valeur du navigateur
- S'applique à toutes les propriétés avec une valeur en rem (width, font-size, gap, margin...)

# Pratiquons ! - Exemple unité rem

Pré-requis :

- Avoir la ressource `ressources/exemple-unite-rem`

A télécharger ici :

<https://download-directory.github.io/?url=https://github.com/DanYellow/cours/tree/main/integration-web-s1/travaux-pratiques/numero-4/ressources>

Testez le site pour mieux comprendre l'unité rem. Il y a des commentaires également dans le code du site dans le fichier css.



# Formatage du texte – rem

- PhotoShop (et autres) utilisent l'unité pixel
  - Il faudra donc faire la conversion px → rem
- Calcul pixel → rem :

$$\text{taille en rem} = \frac{\text{taille police \acute{e}l\acute{e}ment (px)}}{\text{taille police racine* (px)}}$$

\* : Représente la balise `<html>` ou la valeur par défaut de la police d'écriture du navigateur. Rappelons qu'il est préférable de laisser le navigateur laisser gérer la chose

## Formatage du texte – em / %

- 1 em / 100 % représente 100 % de la valeur de la **font-size du parent direct**
- Calcul pixel → em / % :

$$\text{taille en em/\%} = \frac{\text{taille police \acute{e}l\acute{e}ment (px)}}{\text{taille police parent (px)}}$$

# Formatage du texte – em / %

Exemple :

- Parent du parent : 20px
- Parent : 0.5em ( $20 \times 0.5 = 10\text{px}$ )
- Élément cible : 2em ( $10 \times 2 = 20\text{px}$ )
- → Taille de police de l'élément cible : 20px

**Privilégiez l'unité rem pour vous  
éviter des prises de tête**

# Formatage du texte – text-align

- Permet de modifier l'alignement du texte
- Aligné à gauche par défaut (text-align: left;)

## Sources :

- <https://developer.mozilla.org/fr/docs/Web/CSS/text-align>
- <https://developer.mozilla.org/fr/docs/Web/CSS/text-justify>

# Formatage du texte – text-align

- Accepte les valeurs suivantes :
  - left, right, center, justify
- La valeur "justify" donne accès à une nouvelle propriété "text-justify" pour ajouter de la granularité à la justification du texte

## Sources :

- <https://developer.mozilla.org/fr/docs/Web/CSS/text-align>
- <https://developer.mozilla.org/fr/docs/Web/CSS/text-justify>

# Point accessibilité – Alignement du texte

- La justification du texte dans le web est problématique
  - Crée des lézardes (voir ci-contre)
  - Rend difficile la lecture notamment pour les dyslexiques
- Évitez la valeur "justify" pour la propriété "text-align"

ecte turnadis pisici elit, sed  
modest empora incidunt ut  
re et dolore magna aliqua.  
enimenad mi meniat, quis  
rud exercitatio ullamco  
ris nisi utaliniq id ex ea  
modi consequat. Quis aute  
repreheni deriti voliptate  
resse cillum doloresu fugiat  
n pariatur. Excepteuren asint  
ecat cupiditat non proidens



Les lézardes rendent compliqué la lecture, l'œil a du mal à suivre

## Sources :

- <https://designforhackers.com/blog/justify-text-html-css> – anglais
- [www.codebuilder.fr/blog/redaction/justifier-texte-web-bonne-pratique/](http://www.codebuilder.fr/blog/redaction/justifier-texte-web-bonne-pratique/)
- [https://fr.wikipedia.org/wiki/L%C3%A9zarde\\_\(imprimerie\)](https://fr.wikipedia.org/wiki/L%C3%A9zarde_(imprimerie))

# Point accessibilité – Alignement du texte

- Le centrage du texte est aussi problématique
  - Trop de textes centrés rend compliqué la lecture : l'utilisateur a dû mal à "suivre" les lignes

## Sources :

- <https://designforhackers.com/blog/justify-text-html-css> – anglais
- [www.codebuilder.fr/blog/redaction/justifier-texte-web-bonne-pratique/](http://www.codebuilder.fr/blog/redaction/justifier-texte-web-bonne-pratique/)



# Formatage du texte – text-decoration

- Décore le texte
- Propriété raccourcie regroupant :
  - text-decoration-line
  - text-decoration-color
  - text-decoration-style
- Permet par exemple de souligner ou barrer du texte avec une certaine couleur

## Source :

- <https://developer.mozilla.org/fr/docs/Web/CSS/text-decoration>
- [https://developer.mozilla.org/fr/docs/Web/CSS/CSS\\_Text\\_Decoration](https://developer.mozilla.org/fr/docs/Web/CSS/CSS_Text_Decoration)

# Point accessibilité – Soulignement

- Réservé historiquement pour les (hyper)liens
  - **A utiliser avec parcimonie si ce n'est pas pour un lien**

## Sources :

- [https://developer.mozilla.org/fr/docs/Learn/HTML/Introduction\\_to\\_HTML/HTML\\_text\\_fundamentals#italique\\_gras\\_soulignement%E2%80%A6](https://developer.mozilla.org/fr/docs/Learn/HTML/Introduction_to_HTML/HTML_text_fundamentals#italique_gras_soulignement%E2%80%A6) – Voir encadré bleu

# Formatage du texte – listes

- Permet de lister des éléments
  - `<ul>` : non ordonnée. Ex : liste de courses
  - `<ol>` : ordonnée. Ex : classement
- Chaque élément de liste doit être contenu dans une balise `<li>`

## Sources :

- [https://developer.mozilla.org/fr/docs/Learn/HTML/Introduction\\_to\\_HTML/HTML\\_text\\_fundamentals#listes](https://developer.mozilla.org/fr/docs/Learn/HTML/Introduction_to_HTML/HTML_text_fundamentals#listes)
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/ul>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/li>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/ol>

# Formatage du texte – listes

- Chaque groupe de `<li>` doit être contenu dans un `<ul>` ou `<ol>`
- Possibilité d'imbriquer une liste dans une liste
  - `<ol>` ou `<ul>` imbriqués doivent être dans un `<li>`
  - Note : un `<ul>` peut contenir un `<ol>` et inversement

## Sources :

- [https://developer.mozilla.org/fr/docs/Learn/HTML/Introduction\\_to\\_HTML/HTML\\_text\\_fundamentals#listes](https://developer.mozilla.org/fr/docs/Learn/HTML/Introduction_to_HTML/HTML_text_fundamentals#listes)
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/ul>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/li>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/ol>

# Pratiquons ! - Formattage texte

Pré-requis :

- Avoir la ressource **ressources/formattage-texte**

A télécharger ici :

<https://cours-prod.cyu.fr/mod/url/view.php?id=203391>

Connectez-vous avec votre compte Teams/Office 365 pour télécharger la ressource

# Formatage du texte – Titres

- Balise `<h1>`-`<h6>`
  - Plus le chiffre est petit plus le titre est important
  - Permettent de définir du texte important du point de vue référencement
- Une seule balise `<h1>` par page mais...

## Sources :

- <https://cours-prod.cyu.fr/mod/lesson/view.php?id=184001&pageid=1828>
- [https://developer.mozilla.org/fr/docs/Web/HTML/Element/Heading\\_Elements](https://developer.mozilla.org/fr/docs/Web/HTML/Element/Heading_Elements)

# Formatage du texte – Titres

- ...HTML5 change les règles grâce aux balises *sectionnantes*
  - <section>, <article>, <header>, <body>...

## Sources :

- <https://cours-prod.cyu.fr/mod/lesson/view.php?id=184001&pageid=1828>
- [https://developer.mozilla.org/fr/docs/Web/HTML/Element/Heading\\_Elements](https://developer.mozilla.org/fr/docs/Web/HTML/Element/Heading_Elements)

# Formatage du texte – Titres

```
<body>  
  <h1>Je suis un titre</h1>  
  <div>  
    <h1>Je suis un autre titre</h1>  
  </div>  
</body>
```



Ce code est invalide car chaque `<h1>` **n'est définit pas** dans une balise sectionnante.

La balise `<div>` n'est pas sectionnante

```
<body>  
  <h1>Je suis un titre</h1>  
  <section>  
    <h1>Je suis un autre titre</h1>  
  </section>  
</body>
```



Ce code est valide car chaque `<h1>` **est définit** dans une balise sectionnante.

La balise `<section>` est sectionnante tout comme `<body>`

## Sources :

- <https://cours-prod.cyu.fr/mod/lesson/view.php?id=184001&pageid=1828>
- [https://developer.mozilla.org/fr/docs/Web/HTML/Element/Heading\\_Elements](https://developer.mozilla.org/fr/docs/Web/HTML/Element/Heading_Elements)



# Formatage du texte – Titres

- Si vous ne savez pas si le titre aura une réelle importance, utilisez des classes (comme toujours) et la balise `<p>`
  - Vous changerez la balise le cas échéant

## Sources :

- <https://cours-prod.cyu.fr/mod/lesson/view.php?id=184001&pageid=1828>
- [https://developer.mozilla.org/fr/docs/Web/HTML/Element/Heading\\_Elements](https://developer.mozilla.org/fr/docs/Web/HTML/Element/Heading_Elements)



**Questions ?**