

# Intégration Web

Septembre 2020 – MMI 1 – TP#1 S1





Danielo **JEAN-LOUIS**  
Développeur front-end

## But du cours

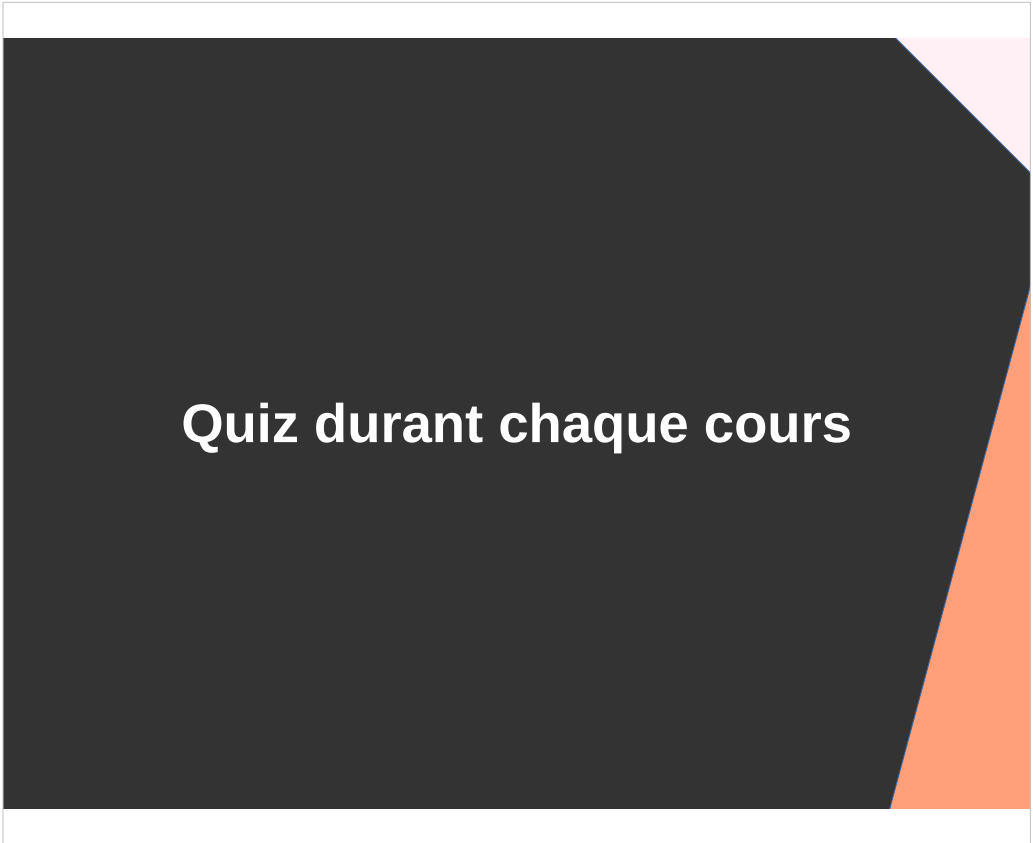
- Voir les bases de la programmation d'un site web
  - Langage HTML
  - Langage CSS
- Sensibilisation au web : design et programmation
  - Accessibilité
  - Bonnes pratiques

- On retrouve Apple, Netflix ou encore Microsoft au W3C. Netflix va notamment contribuer à une technologie (<https://www.w3.org/TR/media-source/>) utilisée dans l'épisode Bandersnatch de Black Mirror
- Le W3C définit également les standards du javascript ou du CSS
- Face à la lenteur du W3C, Apple (Safari), Mozilla (Firefox) et Opera (Opera) forment le WHATWG (Web Hypertext Application Technology Working Group). L'idée étant de définir ensemble les standards du web.
-

## But du cours

- Être capable de réaliser un site web responsive avec uniquement du html et du CSS

- On retrouve Apple, Netflix ou encore Microsoft au W3C. Netflix va notamment contribuer à une technologie (<https://www.w3.org/TR/media-source/>) utilisée dans l'épisode Bandersnatch de Black Mirror
- Le W3C définit également les standards du javascript ou du CSS
- Face à la lenteur du W3C, Apple (Safari), Mozilla (Firefox) et Opera (Opera) forment le WHATWG (Web Hypertext Application Technology Working Group). L'idée étant de définir ensemble les standards du web.
-



## **Quiz durant chaque cours**

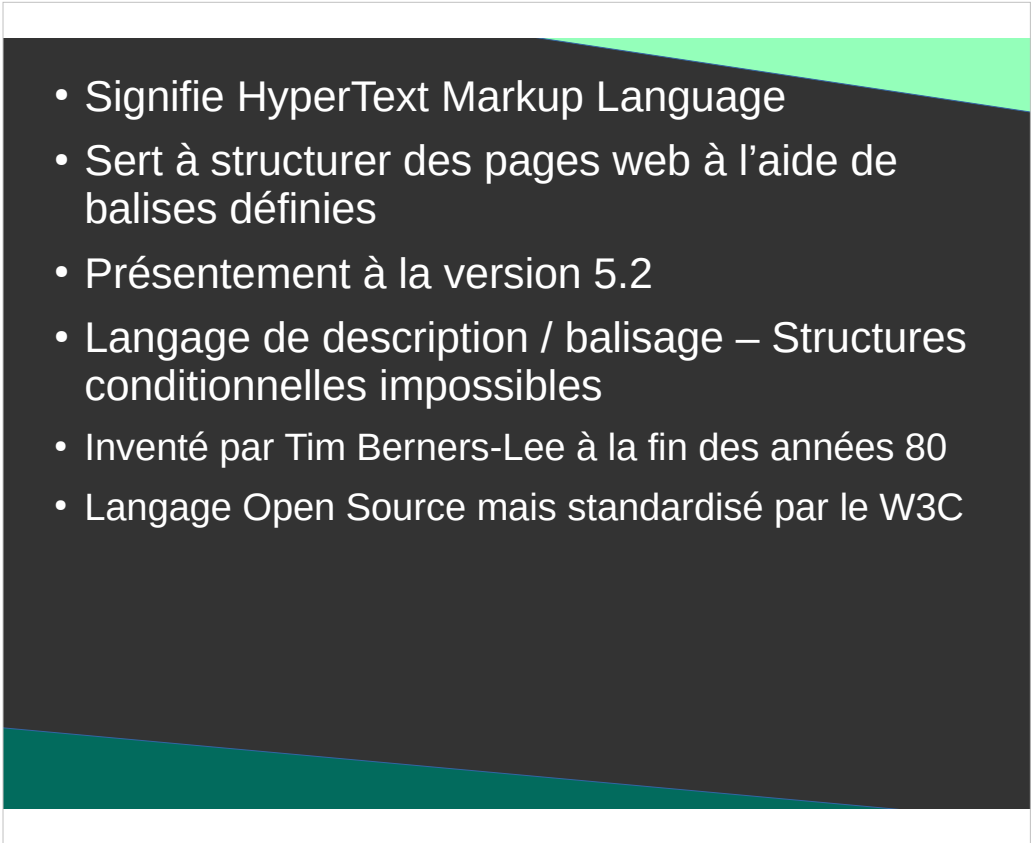
- Le but du quiz n'est pas de vous sanctionner, mais d'acquérir les compétences que vous avez obtenu durant le cours. De fait, il ne portera que sur le contenu vu en classe



**Évaluation en fin de cycle : réaliser /  
ajouter des fonctionnalités à un site**



**Que signifie HTML ?**

- 
- Signifie HyperText Markup Language
  - Sert à structurer des pages web à l'aide de balises définies
  - Présentement à la version 5.2
  - Langage de description / balisage – Structures conditionnelles impossibles
  - Inventé par Tim Berners-Lee à la fin des années 80
  - Langage Open Source mais standardisé par le W3C

- Tim Berners-Lee a également inventé :
  - le www (world wide web) ou web, système incluant :
    - adresses web
    - protocole HTTP (communication client-serveur)
    - Le HTML
  - Il invente également le W3C
- Le HTML n'est pas un langage de programmation. Il ne permet pas de mettre en place des conditions (if, else, boucle while...)



## W3C - World Wide Web Consortium

- Inventé par Tim Berners-Lee en oct. 94
- Consortium standardisant le HTML
- Les membres sont des entreprises qui paient
- WHATWG - 2004
  - Consortium entre éditeurs de navigateurs

### Sources :

- <https://www.w3.org/>
- <https://www.w3.org/TR/media-source/>
- <https://www.youtube.com/watch?v=WLqc0EX8Bmg>

- On retrouve Apple, Netflix ou encore Microsoft au W3C. Netflix va notamment contribuer à une technologie (<https://www.w3.org/TR/media-source/>) utilisée dans l'épisode Bandersnatch de Black Mirror
- Le W3C définit également les standards du javascript ou du CSS
- Face à la lenteur du W3C, Apple (Safari), Mozilla (Firefox) et Opera (Opera) forment le WHATWG (Web Hypertext Application Technology Working Group). L'idée étant de définir ensemble les standards du web.

-

## Page web

- Fichier texte au format .html ou .htm
- Peut contenir du texte, des médias, des liens, des fichiers CSS ou javascript
- Peut être développé sous TextEdit (MacOS) ou Bloc-Notes (Windows)
- Le plus populaire : Visual Studio Code (Linux, MacOS, Windows)

- Les liens sont également appelés « hypertextes »
- L'avantage d'utiliser un logiciel fait pour le développement web est qu'il propose la coloration syntaxique, en un coup d'œil il est possible de savoir ce qui bon ou non. Ces logiciels permettent également l'installation d'extensions qui améliorent encore plus l'expérience de développement.
- Il existe également : SublimeText (2 et 3) (freemium), Brackets (gratuit), Atom (gratuit), Notepad++, Dreamweaver

## Site web

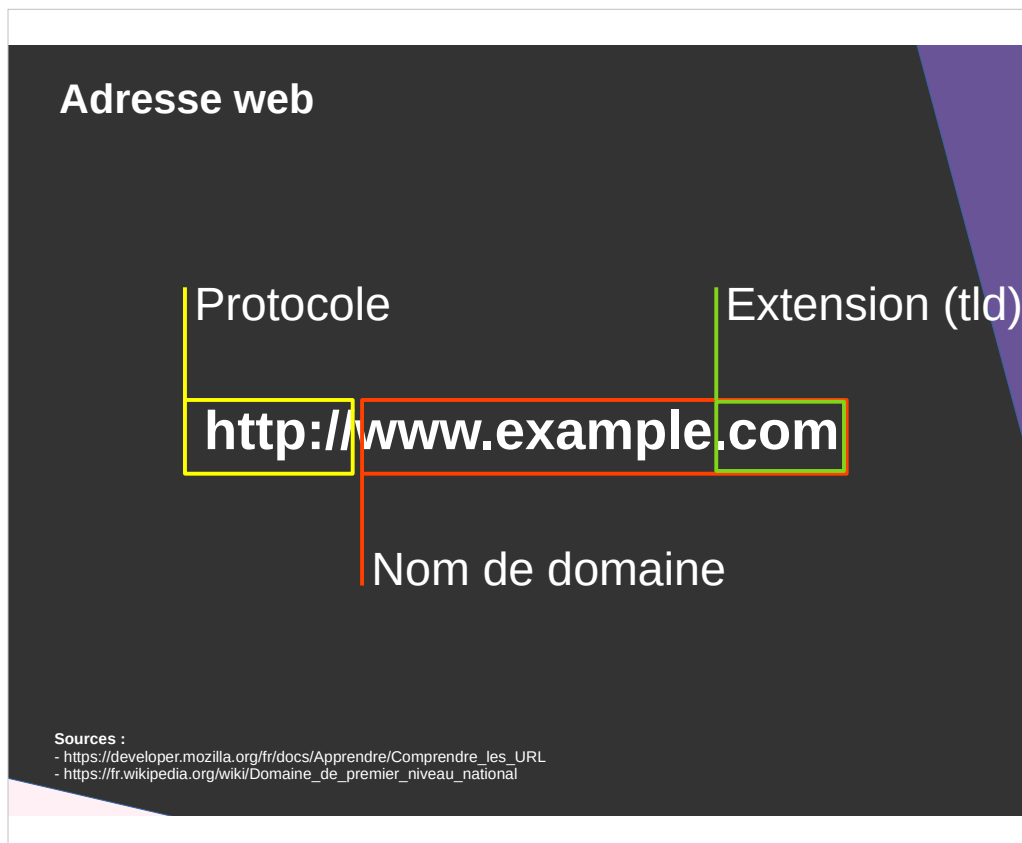
- Ensemble de pages web liées par des liens
- Accessible via Internet, réseau informatique mondial et public grâce à une adresse web
- Le fichier index.html fait office de porte d'entrée pour un site
- Affiché grâce à un navigateur
  - Si la page est locale, il suffit d'ouvrir une des pages de votre site avec un navigateur

- Un site Internet n'existe pas, seul le site web existe. Internet désigne « le réseau mondial accessible au public » - wikipedia
- Un intranet est un réseau internet limité à un lieu précis (entreprise par exemple)
- Le darknet est un réseau internet où les utilisateurs sont anonymes
- Un site peut ne pas avoir de page index.html, mais il faudra faire une configuration serveur, inutile de se casser la tête

## Navigateur web

- Logiciel permettant de lire des sites web
- Chrome est le plus populaire (PC et mobile) (+60 % de pdm)
- Il lit la page web du haut en bas
- Un navigateur n'est plus uniquement sur PC
- Accès à un site via une adresse web

- Chrome a pour concurrent Firefox, Safari ou encore Internet Explorer (à ne pas utiliser) et Edge (Version de Microsoft de Chrome)
- Une page étant lue de haut en bas, les éléments critiques doivent être le plus haut possible sur la page
- Un navigateur se trouve également sur mobile (smartphone et tablette), tv

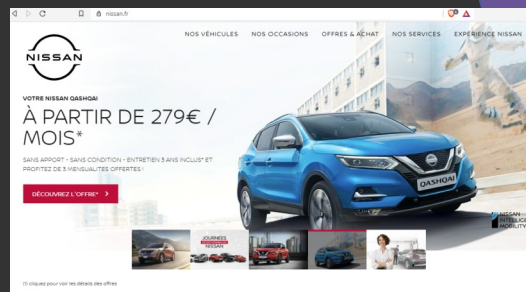


- La paire nom (exemple) / extension (com) est unique. Exemple : nissan.
- Au début du web certaines entreprises, comme Microsoft, n'avaient pas acheté leurs domaines, la loi étant floue à l'époque ils ont dû le payer très cher.
- A la base, chaque extension avait un rôle défini:
  - .fr : pour France | be : pour Belgique
  - .org : organisation
  - .net : pour les sociétés en rapport avec Internet ou le réseau
- etc.

## Adresse web chaque paire nom / extension est unique



<http://nissan.com/>



<http://nissan.fr/>

### Sources :

- <http://nissan.com/>
- <http://nissan.fr/>



- De nos jours les adresses web gèrent également l'unicode (standard universel d'encodage de caractères)

## Balises HTML

- Cœur du langage HTML
- Chacune des 140 balises a un rôle qui lui est propre
- Les balises s'enchaînent comme une pile d'assiettes
- S'appelle "tag" en anglais

### Sources :

- [https://developer.mozilla.org/fr/docs/Apprendre/HTML/Balises\\_HTML](https://developer.mozilla.org/fr/docs/Apprendre/HTML/Balises_HTML)
- <https://www.w3.org/TR/2012/WD-html-markup-20121025/> - en

- L'algorithme se basait notamment sur la bonne utilisation des balises pour classer les sites sur sa page de résultats
- Si les rôles ne doivent plus être utilisés de façon aussi stricte, certains ne sont pas interchangeables, Une balise de texte ne peut pas afficher une image et vice-versa.



## Pratiquons ! - Réalisons notre première page

Pré-requis : avoir un logiciel de développement

- Brackets, Visual Studio Code (VS Code), Sublime Text2 (ou 3) au pire "bloc-notes" fera l'affaire
- Ouvrir la ressource "Ressource TP#1 – squelette html" sur ENT
- Copier puis coller le contenu dans un fichier nommé "index" au format ".html" puis sauvegardez

## Squelette d'une page web

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <!-- Contenu de votre site |-->
</body>
</html>
```

- Ceci est la base d'une page web. Affichons là, ouvrir le fichier avec votre navigateur web préféré



Voici comment le code de la diapo précédente  
s'affiche dans une navigateur  
C'est un peu vide, mais on va quand même  
expliquer ce qu'on a écrit

## Squelette d'une page web – Balise <!DOCTYPE>

```
<!DOCTYPE html>
```

- Permet au navigateur de savoir quelle version de html est utilisée – ici html5
- Indispensable pour passer la validation W3C
- Balise pas vraiment auto-fermante et pas du tout classique
- Absence du <!doctype> active le mode quirks (appelé aussi mode compatibilité)
- **La balise est indispensable de nos jours**

### Sources :

- [https://developer.mozilla.org/fr/docs/Web/HTML/Quirks\\_Mode\\_and\\_Standards\\_Mode](https://developer.mozilla.org/fr/docs/Web/HTML/Quirks_Mode_and_Standards_Mode)
- <https://www.lesintegristes.net/2008/04/06/que-se-passe-t-il-en-mode-quirks-traduction/>
- <http://jkorpele.fi/quirks-mode.html>

- Le mode Quirks permet de gérer des pages qui ont été écrites avant la rédaction de standards du web. Par exemple en mode quirks les noms de classes CSS ne sont pas sensibles à la casse.
- Chaque navigateur a donc son propre mode quirks

## Squelette d'une page web – balise <html>

```
<html lang="fr">  
</html>
```

- Définit la racine structurelle d'un document html
- **Toutes les balises de la page** doivent se trouver dedans
- La balise doit contenir une balise <head> et une balise <body> (dans cet ordre)
- L'attribut lang permet de définir la langue de la page
- Cette balise est **unique**, il ne peut en avoir plusieurs par page

Source :  
- <https://www.w3.org/TR/2012/WD-html-markup-20121025/html.html#html>

- L'attribut lang est un attribut global, il peut donc être associé à n'importe quelle balise. Cet attribut permet notamment aux moteurs de recherche de savoir la langue utilisée, mais surtout aux lecteurs de page pour la synthèse vocale.
- Utiliser un attribut lang fr sur un mot d'une langue étrangère fait lire le mot avec l'accent français. Exemple "newsletter" (en) devient "*niouseletté*" (fr)
- L'attribut lang permet aussi de valider l'orthographe dans les formulaires

## Squelette d'une page web – balise <head>

```
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
```

- Cette balise est **unique**, il ne doit pas en avoir plusieurs par page
- Contient toutes les balises <meta> de la page
  - Métadonnée : information générale de la page
- Contient la balise <title>

Sources :

- <https://www.w3.org/TR/2012/WD-html-markup-20121025/head.html>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/head>

- La balise <head> suit l'ouverture de la balise <html>.
- Une métadonnée peut être par exemple le titre de la page (balise <title></title>) ou le chemin vers un fichier CSS (balise <link rel="stylesheet" href="style.css">)

## Squelette d'une page web – balise <title>

```
<title>Document</title>
```

- Utile au référencement, le contenu est affiché sur la page de résultats des moteurs de recherche
- Contenu affiché dans l'onglet des navigateurs



- La balise est unique, il ne peut avoir qu'une seule balise <title> par page

### Sources :

- <https://www.w3.org/TR/2012/WD-html-markup-20121025/title.html>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/title>

- S'il y a plusieurs balises <title> dans la page, le navigateur utilisera la première balise

## Squelette d'une page web – balise <meta>

- Attribut “name” pour désigner le type
- Attribut “content” pour en définir le contenu
- Exemples :
  - description : pour afficher un descriptif de la page, utilisé par les moteurs de recherche
  - viewport : définir le comportement de l’affichage du site

### Sources :

- <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/meta>  
- [https://developer.mozilla.org/fr/docs/Mozilla/Mobile/Balise\\_meta\\_viewport](https://developer.mozilla.org/fr/docs/Mozilla/Mobile/Balise_meta_viewport)  
- [https://developer.apple.com/library/archive/documentation/AppleApplications/Reference/SafariHTMLRef/Articles/MetaTags.html#//apple\\_ref/doc/uid/TP40008193](https://developer.apple.com/library/archive/documentation/AppleApplications/Reference/SafariHTMLRef/Articles/MetaTags.html#//apple_ref/doc/uid/TP40008193)  
- [https://developer.apple.com/library/archive/documentation/AppleApplications/Reference/SafariWebContent/UsingtheViewport/UsingtheViewport.html#//apple\\_ref/doc/uid/TP40006509-SW25](https://developer.apple.com/library/archive/documentation/AppleApplications/Reference/SafariWebContent/UsingtheViewport/UsingtheViewport.html#//apple_ref/doc/uid/TP40006509-SW25)

- Sur mobile le viewport désigne la zone d’affichage du site, comment le site doit se comporter sur mobile.
- On peut par exemple décider que l’utilisateur ne peut pas zoomer dans le site grâce au viewport



## Squelette d'une page web – balise <body>

```
<body>  
  <!-- Contenu de votre site |-->  
</body>
```

- Cette balise est **unique**, il ne doit pas en avoir plusieurs par page.
- On y écrit les balises html qui affichent du contenu

### Sources :

- <https://www.w3.org/TR/2012/WD-html-markup-20121025/body.html>  
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/body>

Un commentaire html n'est affiché que dans le code du source du site. Il faut absolument pas écrire des données sensibles entre commentaires. Le code source est facilement affichable depuis le navigateur web.

## Balises HTML

- Deux types de fermeture de balises:
  - classiques
  - auto-fermantes

```
<p>Hello world</p>
```

Balise "classique"

```

```

Balise "auto-fermantes"

Source :  
- [https://developer.mozilla.org/fr/docs/Apprendre/HTML/Balises\\_HTML](https://developer.mozilla.org/fr/docs/Apprendre/HTML/Balises_HTML)

- Une balise fermante ne peut pas devenir auto-fermante et inversement
- Le slash (ou barre oblique) sur une balise auto-fermante n'est pas obligatoire

## Balises HTML



Source :  
- [https://developer.mozilla.org/fr/docs/Apprendre/HTML/Balises\\_HTML](https://developer.mozilla.org/fr/docs/Apprendre/HTML/Balises_HTML)

- Les balises classiques vont de paire, la valeur de la balise ouvrante doit être égale à celle de la balise fermante
- Les balises classiques possèdent du contenu contrairement à celles auto-fermantes

## Balises HTML – balises "classique"

- S'ouvre avec une balise et se ferme avec la même balise avec un slash
- Le contenu se trouve entre les deux balises
- Exemple de balises "classique" :

- `<div></div>` : `<div id="hello">World</div>`
- `<p><p>` : `<p>Hello world</p>`
- `<a></a>` : `<a href="cyu.fr">Université</a>`
- [...]

### Sources :

- [https://developer.mozilla.org/fr/docs/Web/HTML/C3%89%C3%A9ments\\_en\\_ligne](https://developer.mozilla.org/fr/docs/Web/HTML/C3%89%C3%A9ments_en_ligne)
- [https://developer.mozilla.org/fr/docs/Web/HTML/C3%89%C3%A9ments\\_en\\_bloc](https://developer.mozilla.org/fr/docs/Web/HTML/C3%89%C3%A9ments_en_bloc)

## Balises HTML – balises auto-fermante

- Ne peut contenir du contenu
- Possède des attributs parfois nécessaires à son fonctionnement
- Exemple de balises « classique » :
  - `<img />` : ``
  - `<br>` : `<br>`
  - `<hr>` : `<hr>`
  - [...]

### Sources :

- [https://developer.mozilla.org/fr/docs/Web/HTML/C3%89%C3%A9ments\\_en\\_ligne](https://developer.mozilla.org/fr/docs/Web/HTML/C3%89%C3%A9ments_en_ligne)
- [https://developer.mozilla.org/fr/docs/Web/HTML/C3%89%C3%A9ments\\_en\\_bloc](https://developer.mozilla.org/fr/docs/Web/HTML/C3%89%C3%A9ments_en_bloc)

- La balise `<img>` permet d'afficher des images
- La balise `<br>` permet de faire un saut de ligne (faire "entrée" dans du texte html ne fonctionnera pas)
- La balise `<hr>` affiche une ligne séparatrice horizontale

## Balises HTML

- Imbrication (infinie\*) possible

```
<ul>
  <li>Pikachu</li>
  <li>Emolga</li>
  <li>Posipi</li>
  <li>Dedenne</li>
  <li>Pachirisu</li>
</ul>
```

Source :

- [https://developer.mozilla.org/fr/docs/Apprendre/HTML/Balises\\_HTML](https://developer.mozilla.org/fr/docs/Apprendre/HTML/Balises_HTML)

\* Certaines imbrications ne sont pas valides

- Certaines balises ne sont pas imbriquables. Ça fonctionnera mais ça sera considéré comme étant invalide par un validateur de code html.  
Exemple : une balise <p> dans une balise <p> est considéré comme étant incorrect

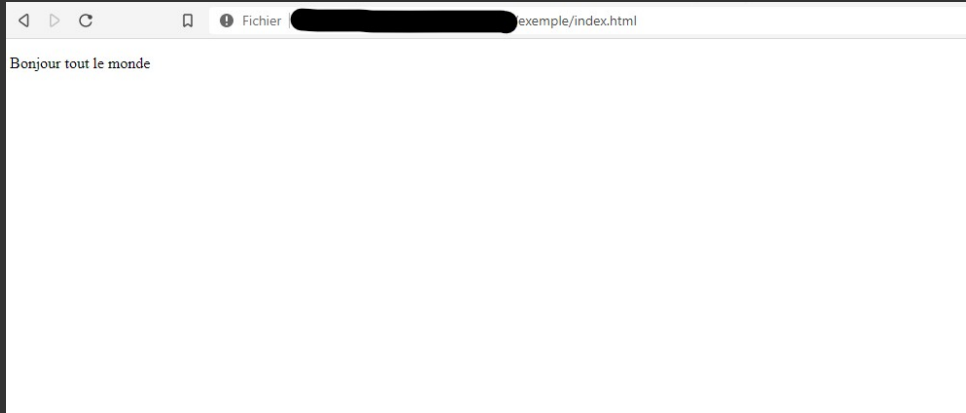
## Pratiquons ! - Améliorons notre page

Pré-requis : avoir un squelette de page web (voir diapo 16)

- Ouvrir son fichier html et ajouter au bon endroit :

```
<p>Bonjour tout le monde</p>
```

# Notre page générée





## Balises HTML

- Deux types d'affichage pour les balises :
  - block
  - inline ("en ligne" en français)

### Sources :

- [https://developer.mozilla.org/fr/docs/Web/HTML/%C3%89%C3%A9ments\\_en\\_ligne](https://developer.mozilla.org/fr/docs/Web/HTML/%C3%89%C3%A9ments_en_ligne)
- [https://developer.mozilla.org/fr/docs/Web/HTML/%C3%89%C3%A9ments\\_en\\_bloc](https://developer.mozilla.org/fr/docs/Web/HTML/%C3%89%C3%A9ments_en_bloc)

- Les positionnements “block” et “inline” sont les deux seuls affichages possibles par défaut pour une balise html. Il en existe d'autres, nous aurons l'occasion de les voir.

## Pratiquons ! - Découvrons les balises “block”

Pré-requis : avoir un squelette de page web (voir diapo 16)

- Ouvrir son fichier html et ajouter au bon endroit :

**`<p>Bonsoir tout le monde</p>`**

Ajoutez autant de balises `<p>` que vous vous voulez, elles formeront une “tour”.

## Pratiquons ! - Découvrons les balises “block”

Ajoutez autant de balises <p> que vous vous voulez, elles formeront une liste de balises <p>

## Balises HTML – balises bloc

- Occupe tout l'espace disponible en largeur
- Force un retour à la ligne après elle
- Exemple de balises bloc
  - `<div>` : conteneur générique
  - `<p>` : paragraphe
  - `<nav>` : navigation
  - `<ul>` : liste à puces non ordonnées
  - `<li>` : élément d'une liste à puces
  - [...]

### Sources :

- [https://developer.mozilla.org/fr/docs/Web/HTML/%C3%89%C3%A9ments\\_en\\_ligne](https://developer.mozilla.org/fr/docs/Web/HTML/%C3%89%C3%A9ments_en_ligne)
- [https://developer.mozilla.org/fr/docs/Web/HTML/%C3%89%C3%A9ments\\_en\\_bloc](https://developer.mozilla.org/fr/docs/Web/HTML/%C3%89%C3%A9ments_en_bloc)

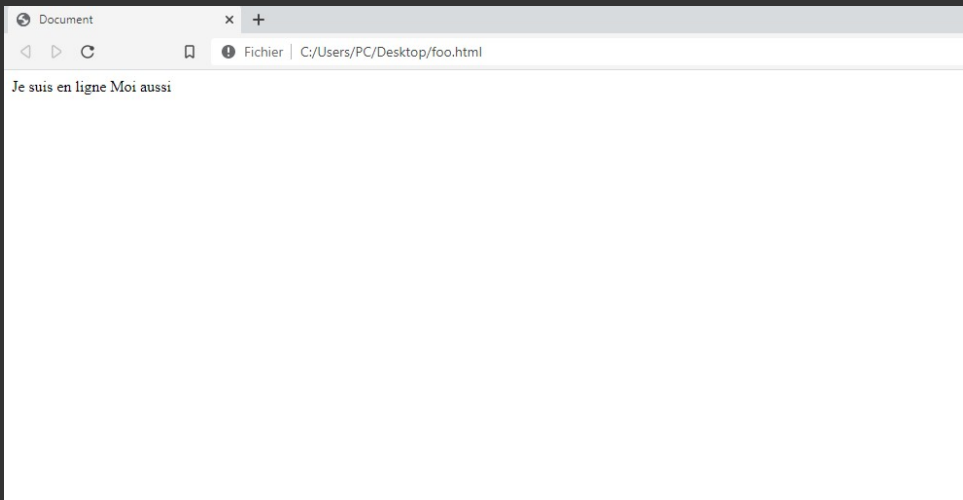
## Pratiquons ! - Découvrons les balises "inline"

Pré-requis : avoir un squelette de page web (voir diapo 16)

- Ouvrir son fichier html et ajouter au bon endroit :

```
<span>Je suis en ligne</span>  
<span>Moi aussi</span>
```

# Notre page générée



## Pratiquons ! - Découvrons les balises "block"

- Même si on rajoute un retour à la ligne dans le code html, les balises inline se suivront toujours, et ce, jusqu'à l'infini
  - La fenêtre de votre navigateur ne sert pas de limite
- Une balise inline se doit d'être dans une balise de type "block"

```
<p>  
  <span>Je suis en ligne</span>  
  <span>Moi aussi</span>  
</p>
```

Mettons nos balises <span> dans une balise de type "block" pour écrire du code valide

- La balise <span> est souvent utilisée avec la balise <p>
- Elle ne doit en aucun cas être utilisée pour mettre une balise de type "block"

## Balises HTML – balises inline

- Peut contenir une balise "inline"
- Occupe la place de son contenu
- Une balise "inline" ne peut pas contenir une balise "block"
- Exemple de balises inline
  - `<span>` : conteneur générique pour le texte
  - `<a>` : lien
  - `<select>` : liste d'options
  - [...]

### Sources :

- [https://developer.mozilla.org/fr/docs/Web/HTML/%C3%89%C3%A9ments\\_en\\_ligne](https://developer.mozilla.org/fr/docs/Web/HTML/%C3%89%C3%A9ments_en_ligne)
- [https://developer.mozilla.org/fr/docs/Web/HTML/%C3%89%C3%A9ments\\_en\\_bloc](https://developer.mozilla.org/fr/docs/Web/HTML/%C3%89%C3%A9ments_en_bloc)



## Balises HTML - Attributs

- Représente les informations complémentaires d'une balise
- Se place **toujours sur la balise ouvrante** d'une balise classique
- Une balise peut contenir plusieurs attributs **uniques**
- La valeur d'un attribut ne peut être qu'une primitive :
  - Chaîne de caractères
  - Nombre (entier, négatif ou décimal)
- La valeur d'un attribut doit être entre guillemets (simples / ' ou doubles / " )

### Sources :

- [https://developer.mozilla.org/fr/docs/Apprendre/HTML/Balises\\_HTML](https://developer.mozilla.org/fr/docs/Apprendre/HTML/Balises_HTML)
- <https://developer.mozilla.org/fr/docs/Web/HTML/Attributs>

## Balises HTML - Attributs

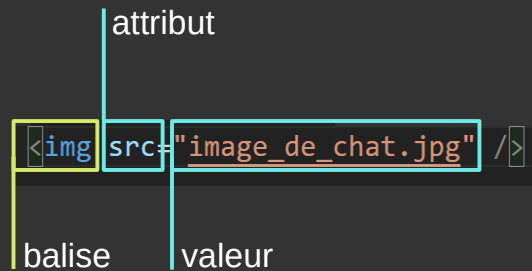
- Certains attributs sont propres à certaines balises
- Les balises auto-fermantes ne se basent que sur les attributs pour s'afficher
- Si les attributs sont prédéfinis, ils peuvent complétés par les data-attributs
  - Attributs s'écrivant sous la forme data-valeur.  
Exemple : data-nombre-eleves
  - Utiles pour passer des informations vers d'autres langages

### Sources :

- [https://developer.mozilla.org/fr/docs/Apprendre/HTML/Balises\\_HTML](https://developer.mozilla.org/fr/docs/Apprendre/HTML/Balises_HTML)
- <https://developer.mozilla.org/fr/docs/Web/HTML/Attributs>

- Si des attributs sont exclusifs à certaines balises, mettre ces attributs sur des balises inappropriées ne va pas provoquer de problèmes. Il ne se passera juste rien
- Les data-attributs sont des balises permettant au HTML de communiquer avec le javascript.

## Balises HTML - Attributs



The diagram shows the HTML tag `` with three annotations: a yellow box around `<img` labeled "balise", a cyan box around `src="image_de_chat.jpg"` labeled "valeur", and a cyan line pointing to the `src` text labeled "attribut".

Sources :

- [https://developer.mozilla.org/fr/docs/Apprendre/HTML/Balises\\_HTML](https://developer.mozilla.org/fr/docs/Apprendre/HTML/Balises_HTML)
- <https://developer.mozilla.org/fr/docs/Web/HTML/Attributs>

- Ici, j'indique à ma balise image la source de mon image pour l'afficher

## Pratiquons ! - Affichons notre première image

Pré-requis :

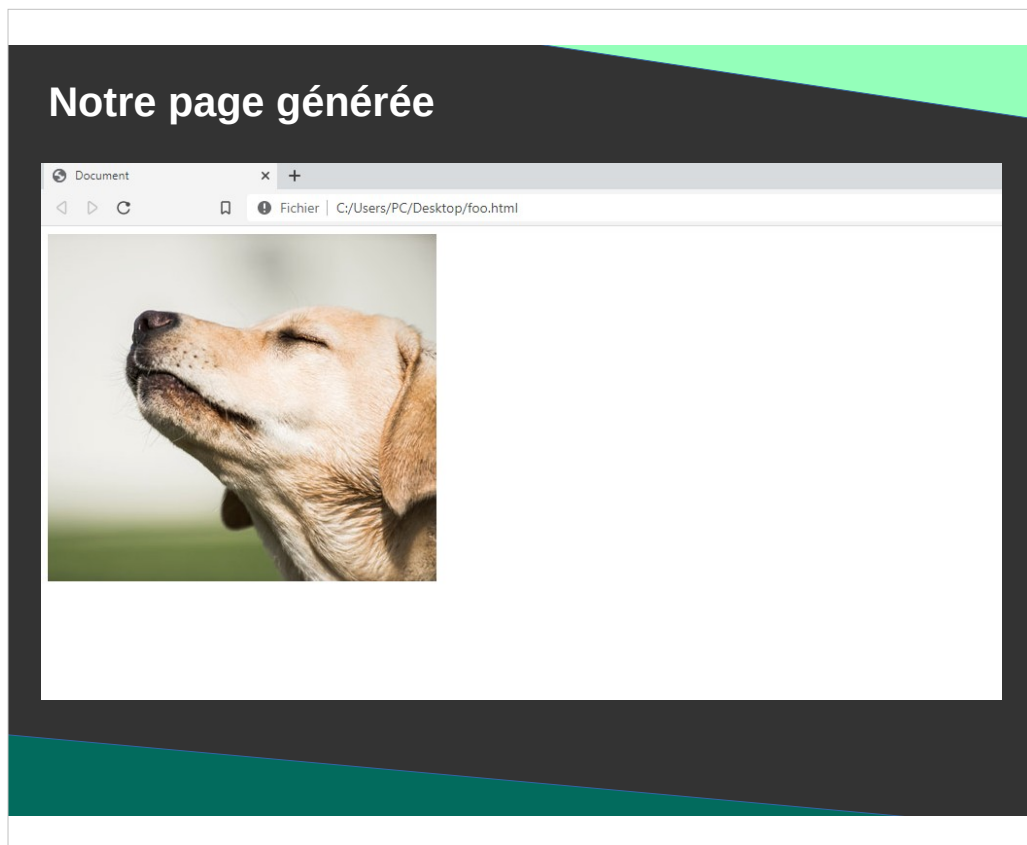
- Avoir un squelette de page web (voir diapo 16)
- Avoir un fichier image (.jp(e)g, .png ou .gif)
- Ouvrir son fichier html et ajouter au bon endroit :

```

```

Sources :  
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/img#attr-alt>

- nom-de-mon-image.extension-du-fichier doit être remplacé par le nom de votre image suivi de l'extension du fichier
- L'attribut "alt" a trois rôles :
  - Référencement : permet aux moteurs de recherches de savoir quelles images sont sur votre site web
  - Accessibilité : permet aux lecteurs de sites (utilisés notamment pour les personnes ne pouvant pas voir correctement) de lire en "lisant" l'image
  - Gestion d'erreur : Si jamais l'image ne s'affiche pas pour une raison, le contenu de la balise "alt" s'affichera à la place



Voici notre image affichée. Ici notre chemin de fichier est relativement simple, car il est dans le même dossier que notre fichier html.

Si le fichier est local mais pas dans le même dossier, il faudra prendre en compte le chemin relatif, soit le chemin du fichier html qui doit afficher la page vers l'image que l'on souhaite afficher

## Attribut src – Chemins relatifs et absolus

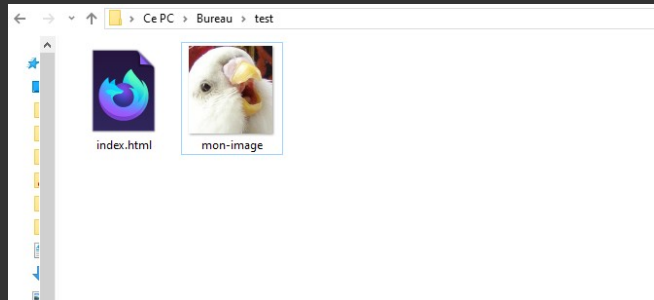
- En HTML les fichiers ne sont pas embarqués, **ils sont liés**
  - Si je déplace mon fichier d'image sans mettre à jour l'attribut "src" de ma balise <img> son image ne s'affichera plus. Cette règle ne s'appliquant pas si l'image est un serveur distant
  - La valeur de l'attribut "src" peut-être un chemin relatif ou absolu

### Sources :

- <https://www.alsacreations.com/astuce/lire/78-Quelle-est-la-difference-entre-les-chemins-relatifs-et-absolus-.html>
- [https://developer.mozilla.org/fr/docs/Apprendre/HTML/Introduction\\_%C3%A0\\_HTML/Creating\\_hyperlinks#URL\\_absolue\\_vs.\\_URL\\_relative](https://developer.mozilla.org/fr/docs/Apprendre/HTML/Introduction_%C3%A0_HTML/Creating_hyperlinks#URL_absolue_vs._URL_relative)

## Chemins relatifs et absolus

- Chemin relatif :
  - Lien pointant vers une ressource dont l'emplacement est relatif au fichier qui l'appelle



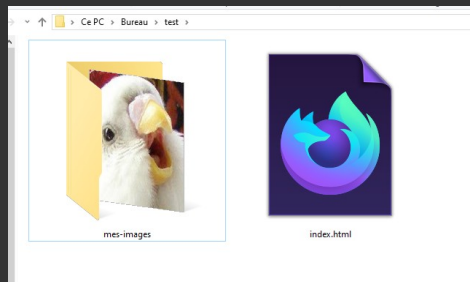
Chemin relatif = `./mon-image.jpg`

### Sources :

- <https://www.alsacreations.com/astuce/lire/78-Quelle-est-la-difference-entre-les-chemins-relatifs-et-absolus-.html>
- [https://developer.mozilla.org/fr/docs/Apprendre/HTML/Introduction\\_%C3%A0\\_HTML/Creating\\_hyperlinks#URL\\_absolue\\_vs.\\_URL\\_relative](https://developer.mozilla.org/fr/docs/Apprendre/HTML/Introduction_%C3%A0_HTML/Creating_hyperlinks#URL_absolue_vs._URL_relative)

# Chemins relatifs et absolus

- Chemin relatif :
  - "/" indique au navigateur qu'il doit chercher l'image à partir du dossier courant
    - Exemple `src= "/mon-image.jpg"`
  - Si les fichiers ne sont pas dans le même dossier, le caractère slash (/) sert de séparateurs entre les dossiers



Chemin relatif = `./mes-images/mon-image.jpg`

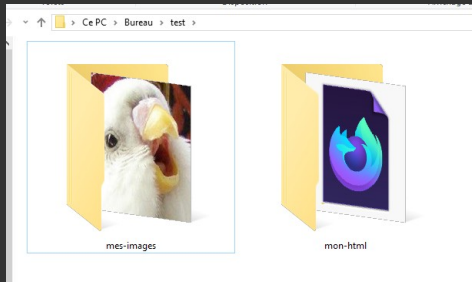
## Sources :

- <https://www.alsacreations.com/astuce/lire/78-Quelle-est-la-difference-entre-les-chemins-relatifs-et-absolus-.html>
- [https://developer.mozilla.org/fr/docs/Apprendre/HTML/Introduction\\_%C3%A0\\_HTML/Creating\\_hyperlinks#URL\\_absolue\\_vs.\\_URL\\_relative](https://developer.mozilla.org/fr/docs/Apprendre/HTML/Introduction_%C3%A0_HTML/Creating_hyperlinks#URL_absolue_vs._URL_relative)



## Chemins relatifs et absolus

- Chemin relatif :
  - Si le fichier que je veux charger est dans un répertoire parent (un niveau au-dessus), on doit écrire "../". Une fois au bon niveau, je mets le nom du dossier où est contenu mon image



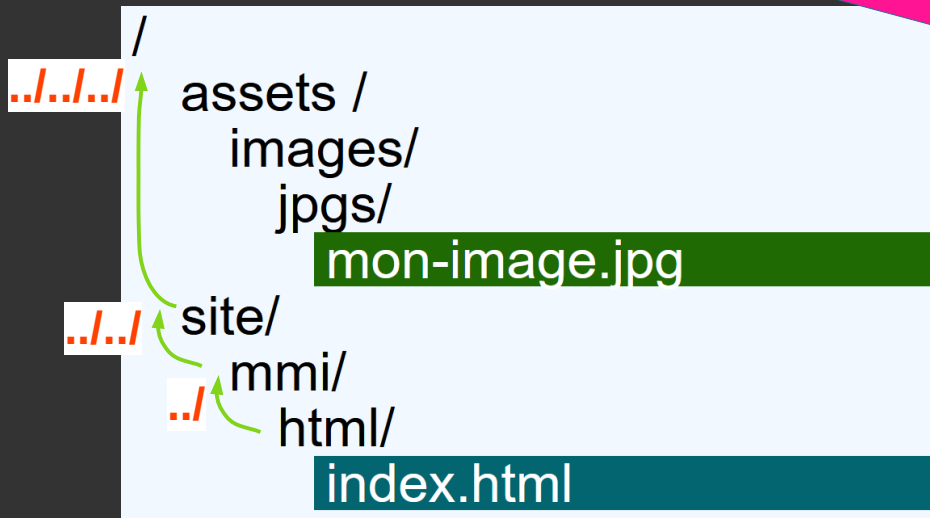
Chemin relatif = ../mes-images/mon-image.jpg

**Sources :**

- <https://www.alsacreations.com/astuce/lire/78-Quelle-est-la-difference-entre-les-chemins-relatifs-et-absolus-.html>
- [https://developer.mozilla.org/fr/docs/Apprendre/HTML/Introduction\\_%C3%A0\\_HTML/Creating\\_hyperlinks#URL\\_absolue\\_vs\\_URL\\_relative](https://developer.mozilla.org/fr/docs/Apprendre/HTML/Introduction_%C3%A0_HTML/Creating_hyperlinks#URL_absolue_vs_URL_relative)

- Si le fichier cible (celui que je veux afficher / charger) est dans un dossier parent situé à un niveau placé 3 paliers plus haut, il faut mettre trois fois "../" (../../../) pour atteindre le bon niveau et écrire le bon chemin

## Chemins relatifs et absolus



### Sources :

- <https://www.alsacreations.com/astuce/lire/78-Quelle-est-la-difference-entre-les-chemins-relatifs-et-absolus-.html>
- [https://developer.mozilla.org/fr/docs/Apprendre/HTML/Introduction\\_%C3%A0\\_HTML/Creating\\_hyperlinks#URL\\_absolue\\_vs.\\_URL\\_relative](https://developer.mozilla.org/fr/docs/Apprendre/HTML/Introduction_%C3%A0_HTML/Creating_hyperlinks#URL_absolue_vs._URL_relative)

## Chemins relatifs et absolus

- Chemin absolu:
  - A utiliser si votre ressource est extérieure à votre site. Si vous devez faire un lien vers une image du site de l'université → chemin absolu

```

```

- Utiliser un chemin absolu localement créera du travail supplémentaire sur les liens quand le site sera en ligne  
Un chemin comme "C:/Users/PC/Desktop/mon-image.jpg" est à éviter.

### Sources :

- <https://www.alsacreation.com/astuce/lire/78-Quelle-est-la-difference-entre-les-chemins-relatifs-et-absolus-.html>
- [https://developer.mozilla.org/fr/docs/Apprendre/HTML/Introduction\\_%C3%A0\\_HTML/Creating\\_hyperlinks#URL\\_absolue\\_vs.\\_URL\\_relative](https://developer.mozilla.org/fr/docs/Apprendre/HTML/Introduction_%C3%A0_HTML/Creating_hyperlinks#URL_absolue_vs._URL_relative)

# Quiz

"Test - Chemins relatifs et absolus" sur ENT

## Balises HTML – Exemple d'attributs

- id="val" : identifiant unique, utilisable dans le CSS.  
**La valeur d'un id est unique dans une page**

```
<div id="hello">World</div>
```

- class="val" : Utilisable dans le css. Peut être multiple dans une page

```
<span class="hello">World</span>
```

### Sources :

- [https://developer.mozilla.org/fr/docs/Apprendre/HTML/Balises\\_HTML](https://developer.mozilla.org/fr/docs/Apprendre/HTML/Balises_HTML)
- <https://developer.mozilla.org/fr/docs/Web/HTML/Attributs>

- si je donne la valeur « hello » à l'attribut id d'une balise dans une page une autre balise ne peut pas avoir l'attribut id avec la valeur hello

## Balises HTML – Exemple d'attributs

- `src="val"` : Utilisé pour les balises permettant d'afficher un média (vidéo, image, son)

```
<audio
  controls
  src="/media/cc0-audio/t-rex-roar.mp3"
>
  Votre navigateur ne supporte pas cette balise.
</audio>
```

- `title="val"` : Utilisable par toutes les balises pour afficher une information supplémentaire au survol avec la souris

```
<span title="hello">World</span>
```



Rendu html  
(au survol)

Sources :

- [https://developer.mozilla.org/fr/docs/Apprendre/HTML/Balises\\_HTML](https://developer.mozilla.org/fr/docs/Apprendre/HTML/Balises_HTML)
- <https://developer.mozilla.org/fr/docs/Web/HTML/Attributs>

## Balises HTML - Attributs

Attribut

```
<a href="https://www.u-cergy.fr/">Site de l'université de Cergy-Pontoise</a>
```

Balise

Valeur

Sources :

- [https://developer.mozilla.org/fr/docs/Apprendre/HTML/Balises\\_HTML](https://developer.mozilla.org/fr/docs/Apprendre/HTML/Balises_HTML)
- <https://developer.mozilla.org/fr/docs/Web/HTML/Attributs>

- Ici, j'indique à ma balise image la source de mon image pour l'afficher

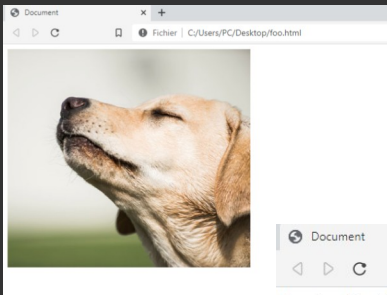
## html – Récapitulatif

- Langage de description
  - Les structures comme “if” ou “while” sont impossibles
- Utilisation de balises qui peuvent être
  - auto-fermantes `<br>`
  - “classique” `<div id="hello">World</div>`
- Des attributs peuvent être ajoutés pour ajouter des informations complémentaires à une balise



- La valeur de l'attribut "id" doit être unique sur une page.
- Une balise html peut être :
  - inline : sa largeur est égale à son contenu
  - block : sa largeur est égale à l'espace disponible mais elle peut être définie
- Les balises peuvent s'imbriquer (gare à leurs fermetures) et au respect de la sémantique

# Voilà ce que nous avons produit jusqu'à présent



Bonjour tout le monde  
Bonsoir tout le monde

Je suis en ligne Moi aussi



Ça fonctionne, mais ce n'est pas très beau, alors on va embellir tout ça grâce au css

## CSS – Feuilles de style

- Signifie Cascading Style Sheets
- Décrit la présentation d'une page html
- Version actuelle 2.1, la version 3 devrait être finalisée un jour mais elle est utilisable partiellement

- Si la version 3 est utilisable présentement, il est possible que certaines spécifications changent. Ce fut le cas de la spécification de l'élément flexbox
- Le html est la structure et le css l'habillement d'un site web

## CSS – Feuilles de style

- Syntaxe :
  - Sélecteur : élément sur lequel le style s'applique
  - Propriété : effet que l'on souhaite donner (taille, couleur de texte, espacement)
  - Valeur : valeur de la propriété ("pink", 42px, etc.)

## CSS – Feuilles de style

- Toutes les propriétés possèdent une valeur par défaut, ces valeurs ne sont pas visibles dans le fichier css
- Certaines valeurs ne s'appliquent qu'à certaines propriétés. La propriété "color" ne peut pas avoir comme valeur "42px"

## CSS – Feuilles de style

Propriété    Valeur

color: red;

Déclaration CSS

## CSS – Feuilles de style

- Les navigateurs ne connaissent pas toutes les propriétés
  - Vérifiable grâce au site : <https://caniuse.com/>
- Chaque navigateur à sa façon d'interpréter le css, des différences de rendu sont possible
- Les commentaires s'écrivent entre `/* */`

Source :  
- <https://caniuse.com/>

- De nos jours, Chrome, Edge (depuis janvier 2020 – Appelé aussi Edge Chromium) et Opera possèdent le même moteur de rendu, si Firefox et Safari (dans une moindre mesure) ont quant à eux des moteurs de rendu différents.
- Les différences de rendu entre les navigateurs sont donc très minimes
- Les commentaires en css sont visibles par tous, par de données critiques dedans



## CSS – Feuilles de style

- Le CSS peut s'appliquer :
  - Directement dans la balise html
  - Entre les balises <style></style> propre à la page
  - Dans un fichier distant au format .css

Sources :  
- <https://caniuse.com/>

## CSS – Dans la balise html

- Utilisation de l'attribut "style"

```
<p style="color: red; /*font-size: 30px;*/ text-decoration:underline;">  
  Bonjour tout le monde  
</p>
```

La propriété "color" permet de définir la couleur du  
texte d'une balise

Bonjour tout le monde

Et il s'affiche en rouge, comme défini dans le code

- La balise html fait office de sélecteur

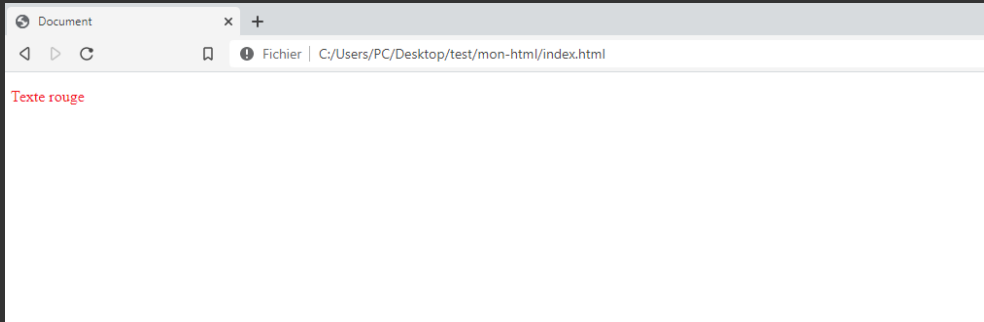
## Pratiquons ! - Écrivons notre premier css

Pré-requis :

- Avoir un squelette de page web (voir diapo 16)
- Au moins une balise `<p>` dans la balise `<body>`
- Ajouter l'attribut "style" à la balise `<p>` avec la valeur "color: red"

```
<p style="color: red">Texte rouge</p>
```

# Notre page générée



## CSS – Dans la balise html - Inconvénients

- Maintenabilité
  - Chaque balise doit avoir son propre attribut "style" → Si on doit changer la couleur de toutes les balises <p> ça sera très fastidieux
- Lisibilité
  - Une balise avec 5+ valeurs dans l'attribut "style" peut rendre rapidement illisible son fichier html s'il y a beaucoup de balises

## CSS – Balise <style>

- Peut-être placée dans la balise <head> ou <body> (<head> par convention)
- Souvent vue avec l'attribut "type" et la valeur "text/css"
- Plusieurs occurrences possible sur la même page
- Utilise un système de sélecteur

Source :  
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/style>

- La balise <style> peut être placée dans la balise <head> ou <body> On préférera la balise <head> par convention et éviter un mélange entre style et texte dans la balise <body>

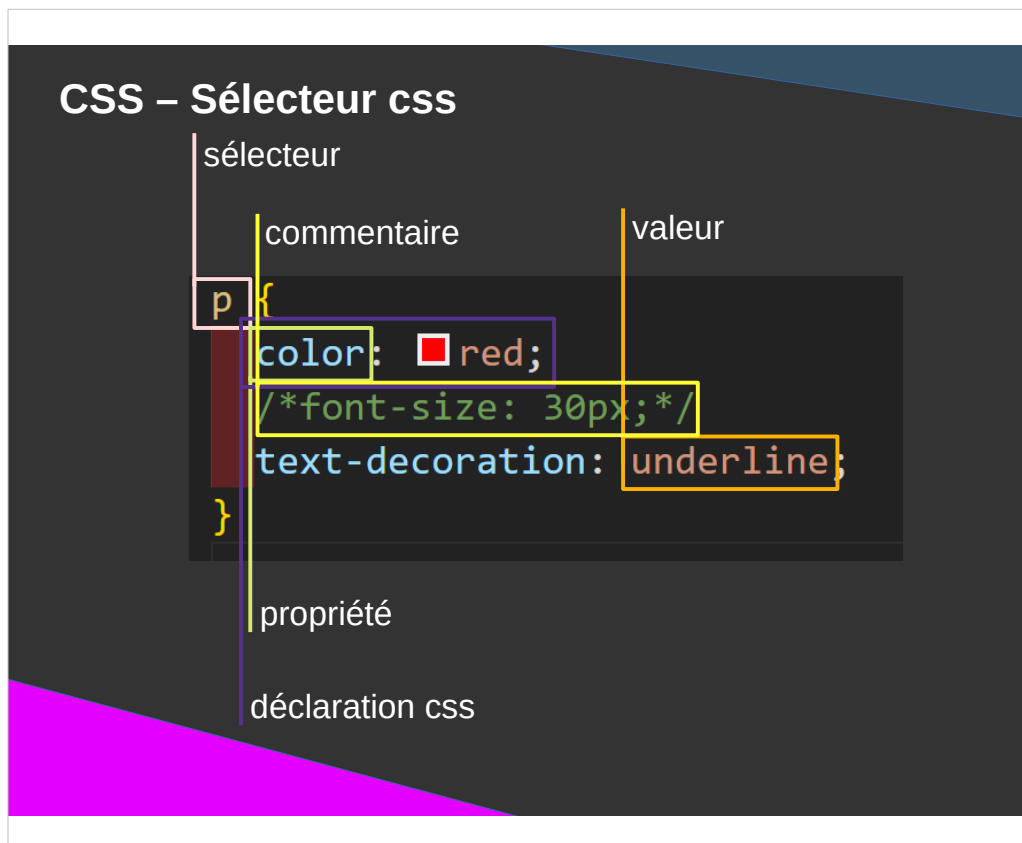
## CSS – Sélecteur css

- Sert à appliquer une ou plusieurs déclarations css dans un fichier lié OU entre les balises `<style></style>`
- N'importe quelle balise html peut servir de sélecteur et donc ses déclarations s'appliquent à toutes les balises qui ont le même nom

```
p {  
  color: purple;  
  font-size: 25px;  
}
```

Tous les textes entre balises `<p>` de la page auront le texte violet et une taille de texte de 25 px

- La balise `<style>` peut être placée dans la balise `<head>` ou `<body>` On préférera la balise `<head>` par convention et éviter un mélange entre style et texte dans la balise `<body>`



- La balise `<style>` peut être placée dans la balise `<head>` ou `<body>`. On préférera la balise `<head>` par convention et éviter un mélange entre style et texte dans la balise `<body>`.
- Le point-virgule (;) permet de séparer les sélecteurs CSS, et ce, quelque soit la méthode utilisée.

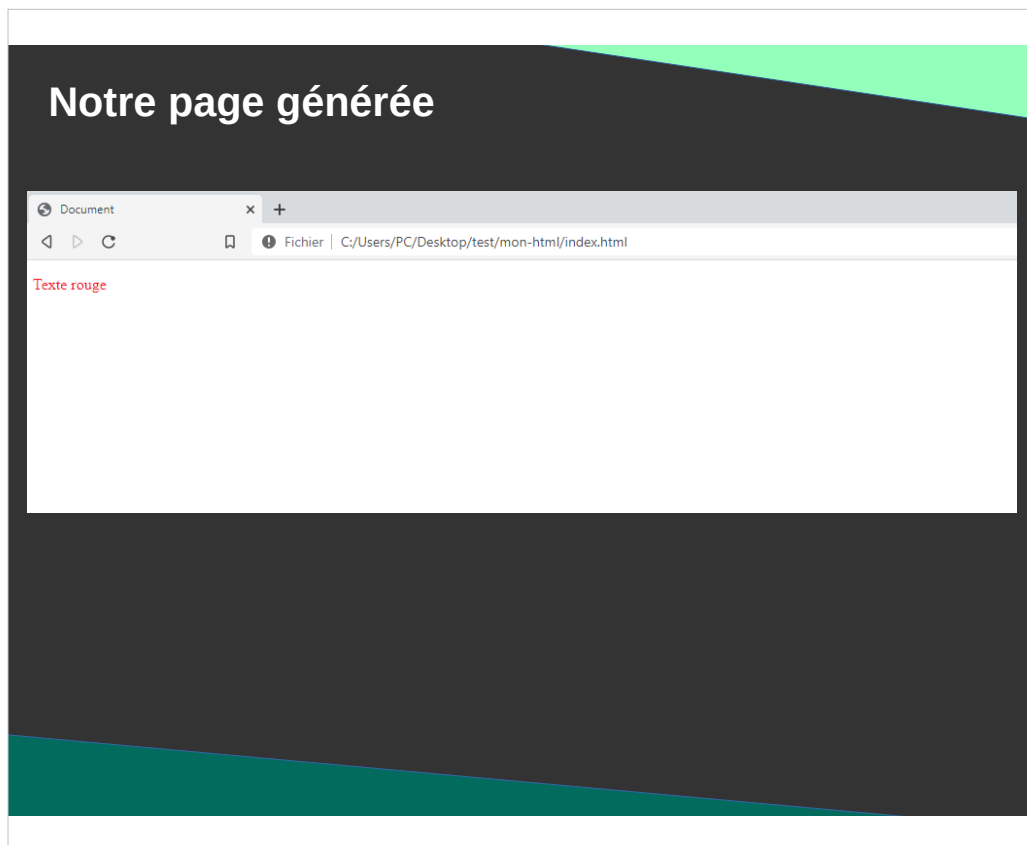


## Pratiquons ! - Écrivons notre balise <style>

Pré-requis :

- Avoir un squelette de page web (voir diapo 16)
- Au moins une balise `<p>` dans la balise `<body>`
- Ajouter dans la balise `<head>` le contenu suivant

```
<style>
  p {
    color: red;
  }
</style>
```



Oui, le résultat est identique à notre essai précédent, mais le style de nos balises est externalisé. C'est bien meilleur pour la lisibilité

## CSS – Balise <style> - Inconvénient

- Maintenabilité
  - Si on a 20 pages, il faudra changer le contenu de la balise <style> dans les 20 pages

## CSS – Feuille de style externe

- Extension .css
- Feuille liée via la balise <link> dans la balise <head>

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

- Possibilité de mettre plusieurs balises <link> dans la même page
- Utilise un système de sélecteur

Source :  
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/link>

- Le fichier externe est liée et non embarqué. De ce fait le chemin (attribut “href”) doit être changé si un des fichiers venait à changer d'emplacement

## CSS – Sélecteur de classe

- Contrevient à limitation des sélecteurs html
  - Un sélecteur html s'applique à toutes les balises de la page
- Utilisation de l'attribut "class", attribut global donc utilisable sur n'importe quelle balise html
- Une classe peut être utilisée par plusieurs balises html
- Une balise html peut avoir plusieurs classes css

Source :  
- [https://developer.mozilla.org/fr/docs/Web/CSS/S%C3%A9lecteurs\\_de\\_classe](https://developer.mozilla.org/fr/docs/Web/CSS/S%C3%A9lecteurs_de_classe)

- Les sélecteurs html sont aussi appelés "sélecteurs de type"

## CSS – Sélecteur de classe

- Ciblage plus spécifique
- Dans le fichier css, on cible une classe en la préfixant d'un point "."
- Les noms de classes sont sensibles à la casse :
  - **MaClasse** ce n'est pas la même chose que "maclasse"

Source :  
- [https://developer.mozilla.org/fr/docs/Web/CSS/S%C3%A9lecteurs\\_de\\_classe](https://developer.mozilla.org/fr/docs/Web/CSS/S%C3%A9lecteurs_de_classe)

## Pratiquons ! - Écrivons notre fichier .css

Pré-requis :

- Avoir un squelette de page web (voir diapo 16)
- Au moins une balise `<p>` dans la balise `<body>`
- Créer un fichier .css au même niveau que le fichier index.html appelé style.css

## Pratiquons ! - Écrivons notre fichier .css

- Dans le fichier html rajouter la ligne suivante dans la balise <head>

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

- Dans le fichier "style.css" fraîchement créé, écrire le contenu suivant

```
p {  
  color: red;  
}
```





Oui, le résultat est identique à notre essai précédent, mais le style de nos balises est externalisé. C'est bien meilleur pour la lisibilité.

Mais voyons les classes ensemble

## Pratiquons ! - Ajoutons nos premières classes

Pré-requis :

- Pré-requis des précédentes pratiques
- Un fichier css appelé "style.css" et lié à un fichier css
- Dans le fichier "style.css", écrire le contenu suivant

```
.texte-important {  
    color: red;  
    font-weight: bold;  
}
```

Sources :

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

Nouvelle propriété css "font-weight", elle permet de faire varier la graisse du texte.

Note : Si après avoir fait varier la graisse d'une texte, il paraît étrange, c'est parce que la police n'est pas faite pour la graisse utilisée. Le navigateur interprète donc ce comportement à sa façon.

## Pratiquons ! - Ajoutons nos premières classes

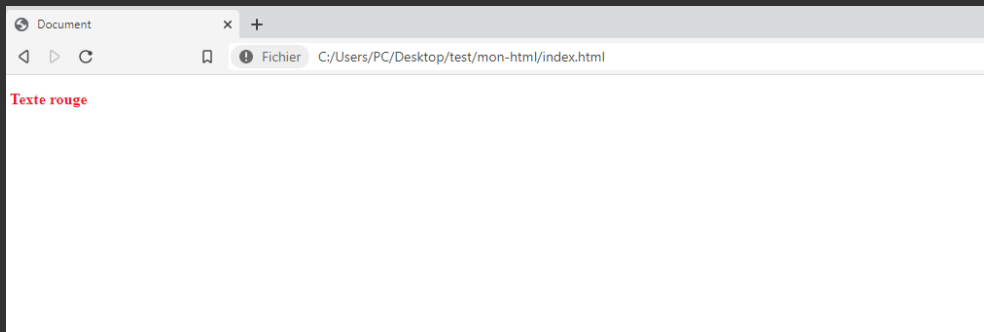
- Dans le fichier "index.html", écrire le contenu suivant

```
<p class="text-important">Texte rouge</p>
```

### Sources :

- [https://developer.mozilla.org/fr/docs/Web/CSS/S%C3%A9lecteurs\\_de\\_classe](https://developer.mozilla.org/fr/docs/Web/CSS/S%C3%A9lecteurs_de_classe)
- [https://developer.mozilla.org/fr/docs/Web/CSS/Selector\\_list](https://developer.mozilla.org/fr/docs/Web/CSS/Selector_list)

## Notre page générée



Voilà ! Notre texte est rouge, en gras mais surtout sa classe est réutilisable

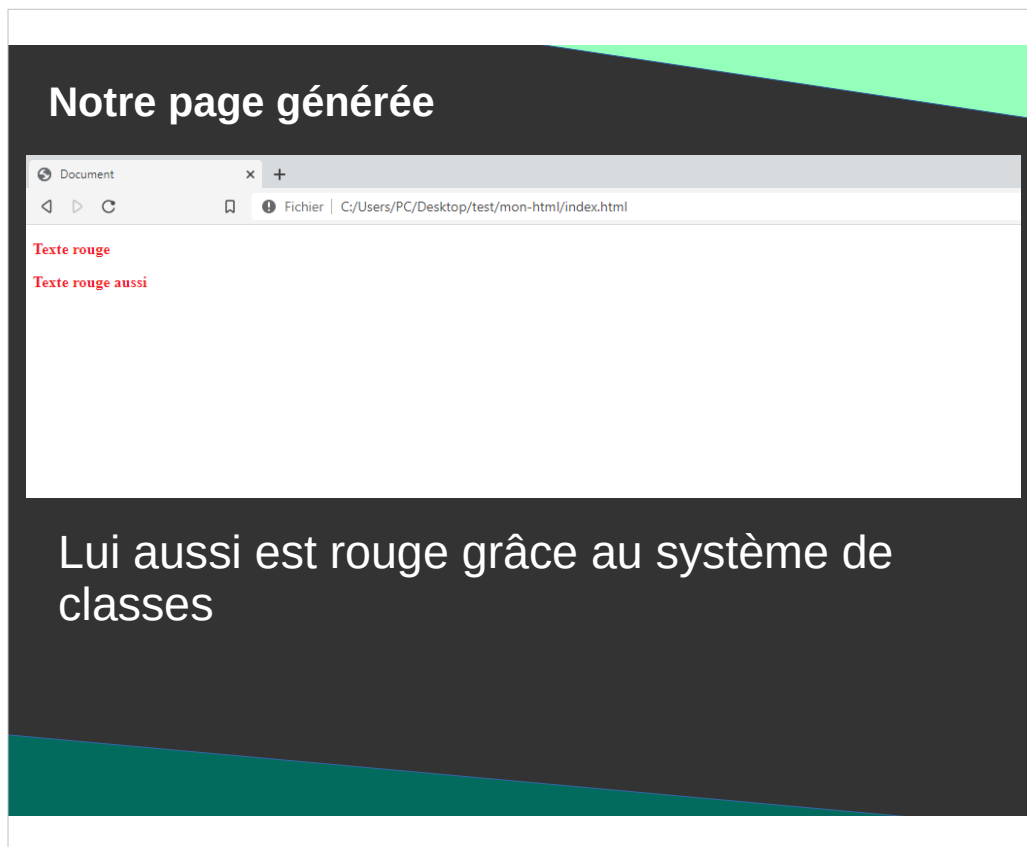
## Pratiquons ! - Ajoutons nos premières classes

- Dans le fichier "index.html", ajouter le contenu suivant

```
<span class="text-important">  
  Texte rouge aussi  
</span>
```

### Sources :

- [https://developer.mozilla.org/fr/docs/Web/CSS/S%C3%A9lecteurs\\_de\\_classe](https://developer.mozilla.org/fr/docs/Web/CSS/S%C3%A9lecteurs_de_classe)
- [https://developer.mozilla.org/fr/docs/Web/CSS/Selector\\_list](https://developer.mozilla.org/fr/docs/Web/CSS/Selector_list)



Oui, le résultat est identique à notre essai précédent, mais le style de nos balises est externalisé. C'est bien meilleur pour la lisibilité.

Mais voyons les classes plus en profondeur

## CSS – Sélecteur de classe

- Un élément html peut avoir plusieurs classes, elles doivent être séparées par un espace :

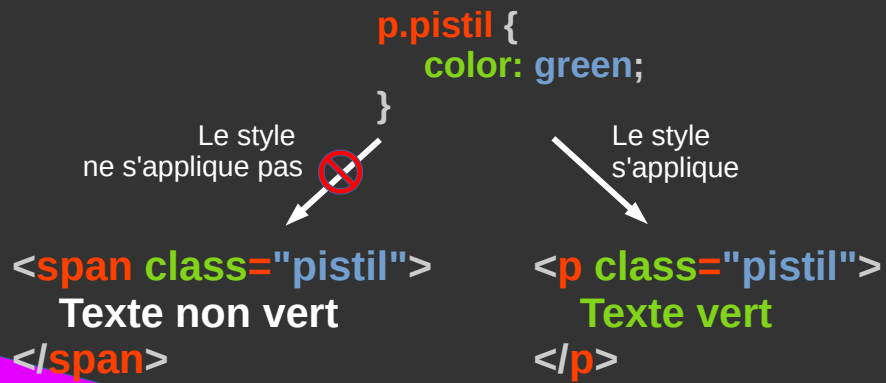
```
<span class="text-important informations">  
  Texte rouge aussi  
</span>
```

- L'ordre des classes dans l'attribut "class" n'influe en rien sur l'interprétation des déclarations

Source :  
- [https://developer.mozilla.org/fr/docs/Web/CSS/S%C3%A9lecteurs\\_de\\_classe](https://developer.mozilla.org/fr/docs/Web/CSS/S%C3%A9lecteurs_de_classe)

## CSS – Sélecteur css

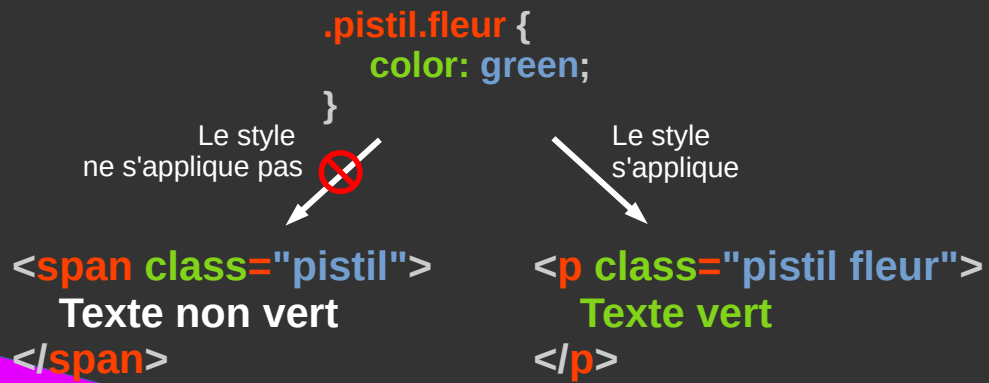
- Dans un sélecteur css, si la classe est préfixée d'une balise html, on cible la balise ayant la classe en question





## CSS – Sélecteur css

- Cette règle s'applique également avec une combinaison de classes



## CSS – Sélecteur css

- Il est possible d'écrire des sélecteurs plus précis en imbriquant plusieurs sélecteurs
- Toutefois seul le dernier sélecteur de la liste voit son style être modifié

Le style s'applique ici

```
.herbier .fleur {  
  color: green;  
}  
  
<section class="herbier">  
  <p class="pistil fleur">  
    Texte vert  
  </p>  
</section>
```

## CSS – Sélecteur de classe

- Plusieurs sélecteurs peuvent partager les mêmes déclarations css, chaque sélecteur devant être espacé d'une virgule
- L'attribut "id" peut être également utilisé pour le même rôle, mais on privilégiera l'attribut "class". Les id sont ciblés par un croisillon "#" dans le css
- Le point précédent s'applique également aux attributs

Source :  
- [https://developer.mozilla.org/fr/docs/Web/CSS/S%C3%A9lecteurs\\_CSS](https://developer.mozilla.org/fr/docs/Web/CSS/S%C3%A9lecteurs_CSS)

## CSS – Feuille de style externe - Avantages

- Maintenabilité
  - Si on change le fichier css, tous les fichiers html liés à un fichier css verront les modifications
  - Le style et la structure sont séparés
- Cohérence dans le design
  - Toutes les pages html liées aux mêmes fichiers css auront les mêmes designs au niveau de leurs balises

## CSS – Feuille de style externe - Avantages

- Gestion des supports
  - La balise <link>, utilisée pour lier un fichier css, accepte comme attribut "media", ceci permet d'utiliser des fichiers css pour certains supports. Par exemple une imprimante

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

Le style ne s'applique qu'aux imprimantes grâce à l'attribut "media" et la valeur "print"

- En absence d'attribut "media", le style s'applique à tout type de média :
  - speech
  - screen
  - print
  - all (valeur par défaut)

## CSS – Pratiquons ensemble

Sélecteur css	HTML ciblé
<code>p {}</code>	<code>&lt;p&gt;[...]&lt;/p&gt;</code>
<code>.ma-classe {}</code>	<code>&lt;div class="ma-classe"&gt;[...]&lt;/div&gt;</code> <code>&lt;img class="ma-classe" src=""&gt;</code>
<code>#un-id {}</code>	
<code>span.ma_classe.info {}</code>	
<code>div .ma-classe {}</code>	
<code>section.ma-classe, ul {}</code>	
	<code>&lt;input class="champ-txt"</code> <code>name="nom" type="text"&gt;</code>
<code>* {}</code>	

# Quiz

"Test - Sélecteurs CSS" sur ENT

- ~10 minutes
- Tentatives illimitées
- Meilleure note prise en compte

## CSS – Sélecteur de balises "\*"

- Sélectionne toutes les balises
- **Ce sélecteur doit être utilisé avec extrême parcimonie**, des comportements inattendus peuvent apparaître

- Le sélecteur "\*" est principalement utilisé avec la déclaration css "box-sizing : border-box". Cette déclaration permet de changer la façon dont les dimensions du modèle de boîte est calculé.  
<https://developer.mozilla.org/fr/docs/Web/CSS/box-sizing>



## Pour aller plus loin

Devoir "Sélecteurs css" sur ENT dans le dossier Travaux pratiques #1  
ou  
jeu : <https://flukeout.github.io/> (en anglais)

## CSS – Feuilles de style – La cascade

- Comme le html, le navigateur lit un fichier css de haut en bas. Si une déclaration est redéfinie plus tard, c'est celle la plus basse qui est prise en compte

```
p {  
  color: red;  
  text-decoration: underline;  
}
```

Les balises <p> s'afficheront en violet car la déclaration concernant la couleur est la plus basse dans le fichier

```
p {  
  color: purple;  
}
```

- Le navigateur lit tous le fichier css avant d'en appliquer le style. Si vous déclarez au début d'un fichier css que le texte fait 10px puis 30px, à l'affichage le texte ne s'affichera pas avec une taille de 10px puis de 30px, il s'affichera directement avec la taille de 30px
- A noter que le fonctionnement de la cascade fonctionne également avec les valeurs de propriétés non reconnues. Dans une déclaration css, si une valeur n'est pas reconnue, alors c'est la dernière valeur connue qui est prise en compte

## CSS – Feuilles de style – La cascade

- Les propriétés css sont héréditaires :
  - Si une classe / balise html a une déclaration, sa descendance en héritera sauf si elle surcharge la déclaration
  - On peut forcer l'hérédité avec la valeur "inherit" pour la propriété que l'on souhaite

```
.text-pistil {  
  color: inherit;  
}
```

- Bonne pratique : Mettre la propriété « font-family » sur la balise html la plus haute dans le code (<body> ou <html>)

## CSS – Feuilles de style – La cascade

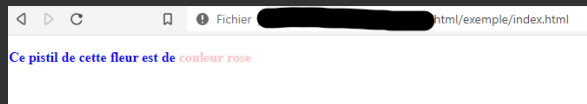
```
<body>
  <div>
    <p class="description">
      Ce pistil de cette fleur est de
      <span class="text-pistil"> couleur rose </span>
    </p>
  </div>
</body>
```

index.html

```
.text-pistil {
  color: pink;
}

.description {
  color: blue;
  font-size: 16px;
  font-weight: bold;
}
```

style.css



rendu dans le navigateur

Le texte de ma balise `<span>` est en gras car il est contenu dans une balise `<p>` possédant la déclaration css "font-weight:bold;"

## CSS – Feuilles de style – Imbrication

- Tout comme les balises html, les sélecteurs css peuvent s'imbriquer

```
div p.description .text-pistil {  
  text-decoration: underline;  
  font-size: 25px;  
}
```

Ici je sélectionne toute classe "text-pistil" contenue dans une balise <p> ayant la classe "description" elle-même contenue dans une balise <div>

- Avec ce sélecteur, on voit qu'il est possible de mélanger balise et classe, il suffit juste de préfixer la classe avec la balise html que l'on souhaite

## CSS – Feuilles de style – Imbrication

- Attention à la sur-imbrication, les sélecteurs trop complexes rendent difficiles la lecture et la réutilisation du sélecteur

```
article section div.imbrication ieme div p.description .text-pistil {  
  text-decoration: underline;  
  font-size: 25px;  
}
```

Mauvaise pratique

- Si la sur-imbrication de balises html est une bonne idée (dans une certaine limite), une sur-imbrication de règles css rend très difficile la lecture du css ou sa réutilisation
- Fichier css plus lourd, donc chargement du site plus long
- Bonne pratique : éviter de faire de l'imbrication html, un sélecteur css

## CSS – Bonnes pratiques

- Éviter les caractères spéciaux dans les noms de sélecteurs :
  - Le tiret (-) et l'underscore ( \_ ) sont autorisés
- Mettre son css dans un fichier externe
- Nommer de façon claire et concise ses sélecteurs css
- Séparer le css en plusieurs fichiers si le fichier est trop conséquent

- Le css reste un fichier texte, plus il y aura du texte, plus le fichier sera lourd, des noms de sélecteurs css court, diminue la taille du fichier. Attention à l'excès, le sélecteur doit rester facile à comprendre. Par exemple, "list-elmnt" est plus compréhensible que "lmnt"

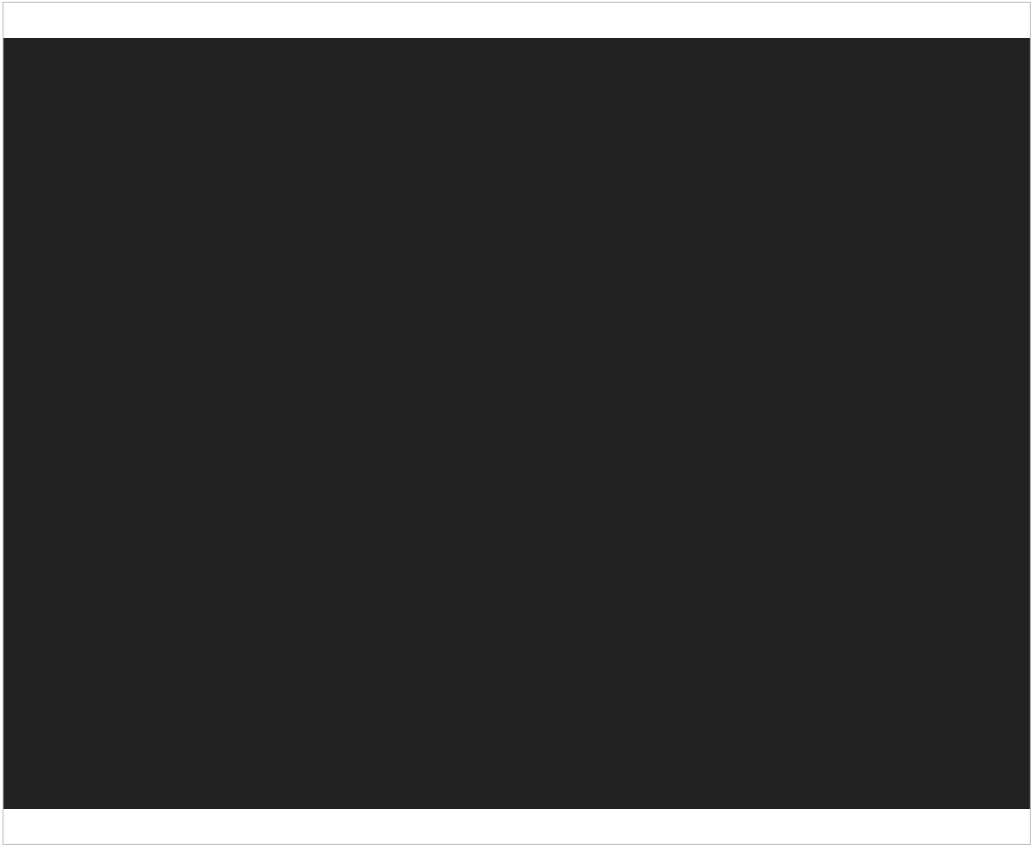
## CSS – Bonnes pratiques

- Utiliser l'attribut id comme sélecteur css
- Éviter la redondance des règles
- Éviter les sélecteurs trop spécialisés
- Éviter les sélecteurs directement sur la balise
- Limiter le plus possible le css dans les balises html



## CSS – Récapitulatif

- Le css est un langage permettant d'ajouter du style aux pages html
- Externalisé le css s'écrit dans un fichier .css
- Les propriétés css s'appliquent en cascade
- Les propriétés css peuvent être héritées



**Questions ?**