

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

But du cours

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

Quiz durant chaque cours

**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

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>

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)

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

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

Adresse web

Protocole

Extension (tld)

http://www.example.com

Nom de domaine

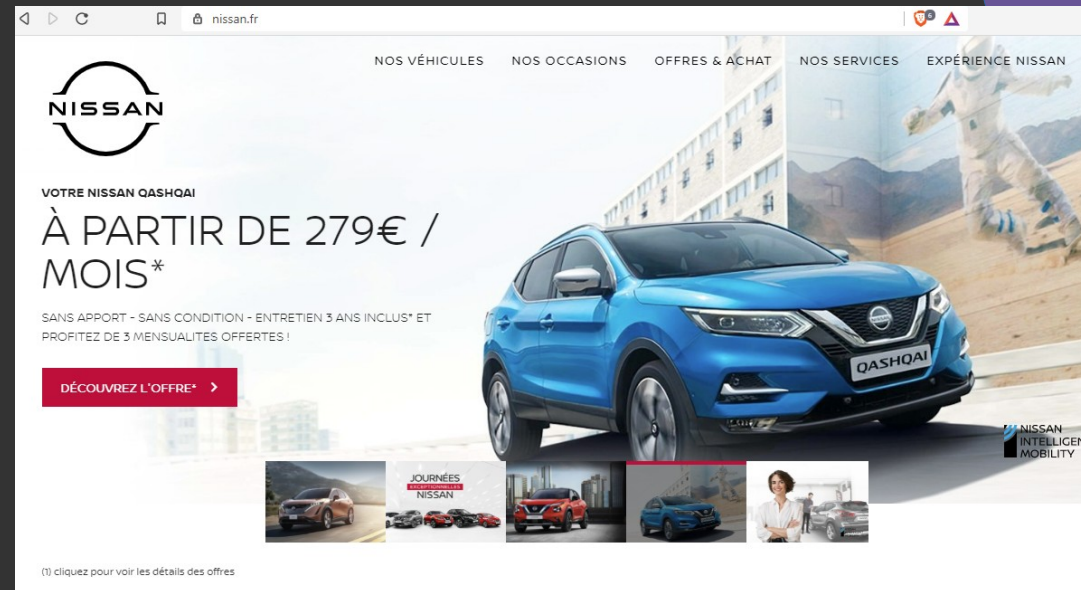
Sources :

- https://developer.mozilla.org/fr/docs/Apprendre/Comprendre_les_URL
- https://fr.wikipedia.org/wiki/Domaine_de_premier_niveau_national

Adresse web chaque paire nom / extension est unique



<http://nissan.com/>



<http://nissan.fr/>

Sources :

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

Adresse web

👤.la

Sources :

- https://developer.mozilla.org/fr/docs/Apprendre/Comprendre_les_URL
- 👤.la

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://www.w3.org/TR/2012/WD-html-markup-20121025/> - en

Pratiquons ! - Réalisons notre première page

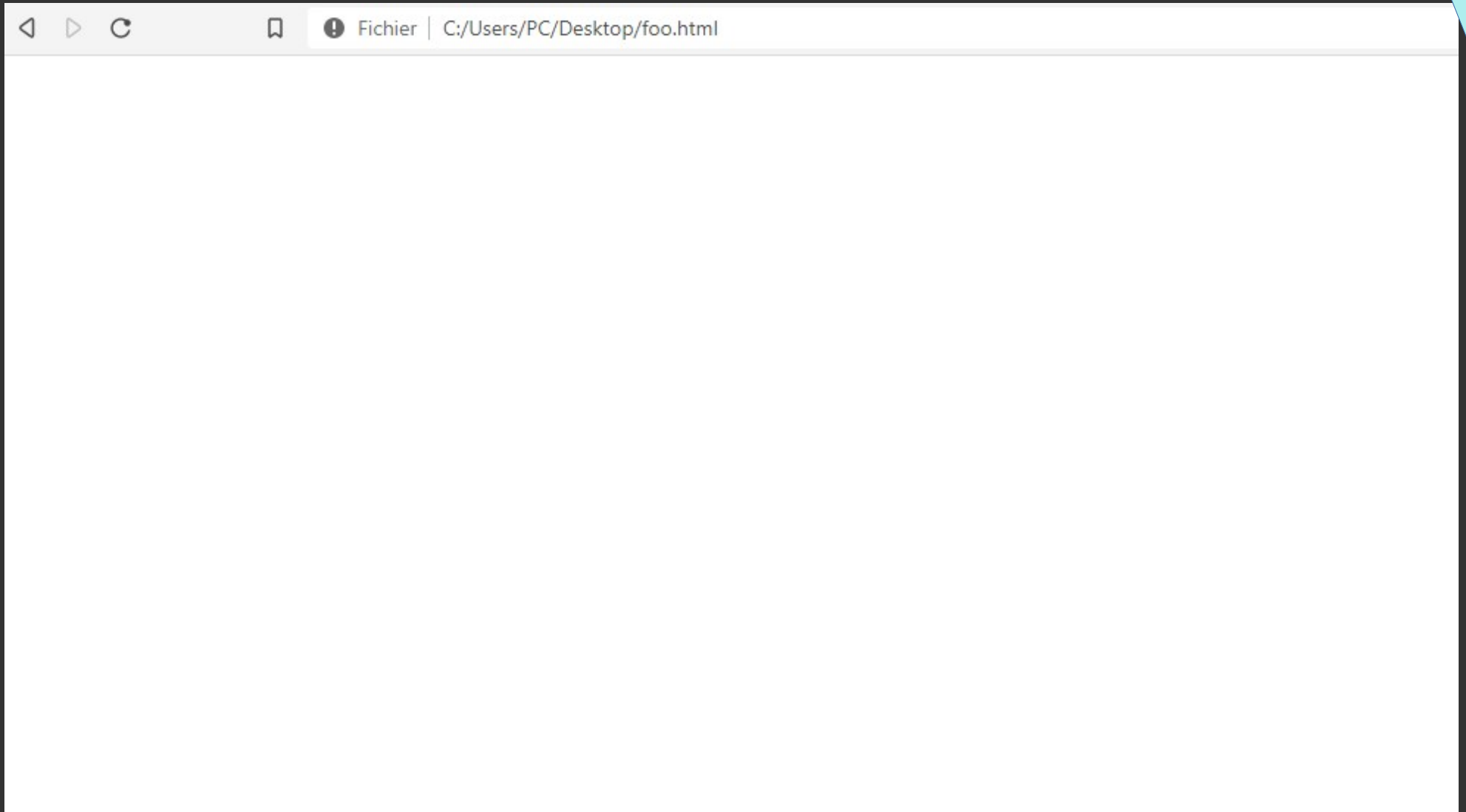
Pré-requis : avoir un logiciel de développement

- 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>
```

Notre page générée



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://www.lesintegristes.net/2008/04/06/que-se-passe-t-il-en-mode-quirks-traduction/>
- <http://jkorpela.fi/quirks-mode.html>

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>

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>

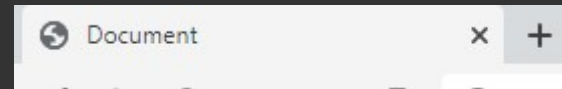
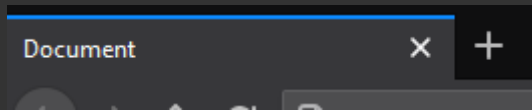
Sources :

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

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>

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.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

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>

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

Balises HTML



Source :

- https://developer.mozilla.org/fr/docs/Apprendre/HTML/Balises_HTML

Balises HTML – balises « classique »

- S'ouvre avec une balise et se ferme avec la même balise avec un anti-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>` : `Université`

- [...]

Sources :

- https://developer.mozilla.org/fr/docs/Web/HTML/%C3%89l%C3%A9ments_en_ligne
- https://developer.mozilla.org/fr/docs/Web/HTML/%C3%89l%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 » :
 - `` : ``
 - `
` : `
`
 - `<hr>` : `<hr>`
 - [...]

Sources :

- https://developer.mozilla.org/fr/docs/Web/HTML/%C3%89l%C3%A9ments_en_ligne
- https://developer.mozilla.org/fr/docs/Web/HTML/%C3%89l%C3%A9ments_en_bloc

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

* Certaines imbrications ne sont pas valides

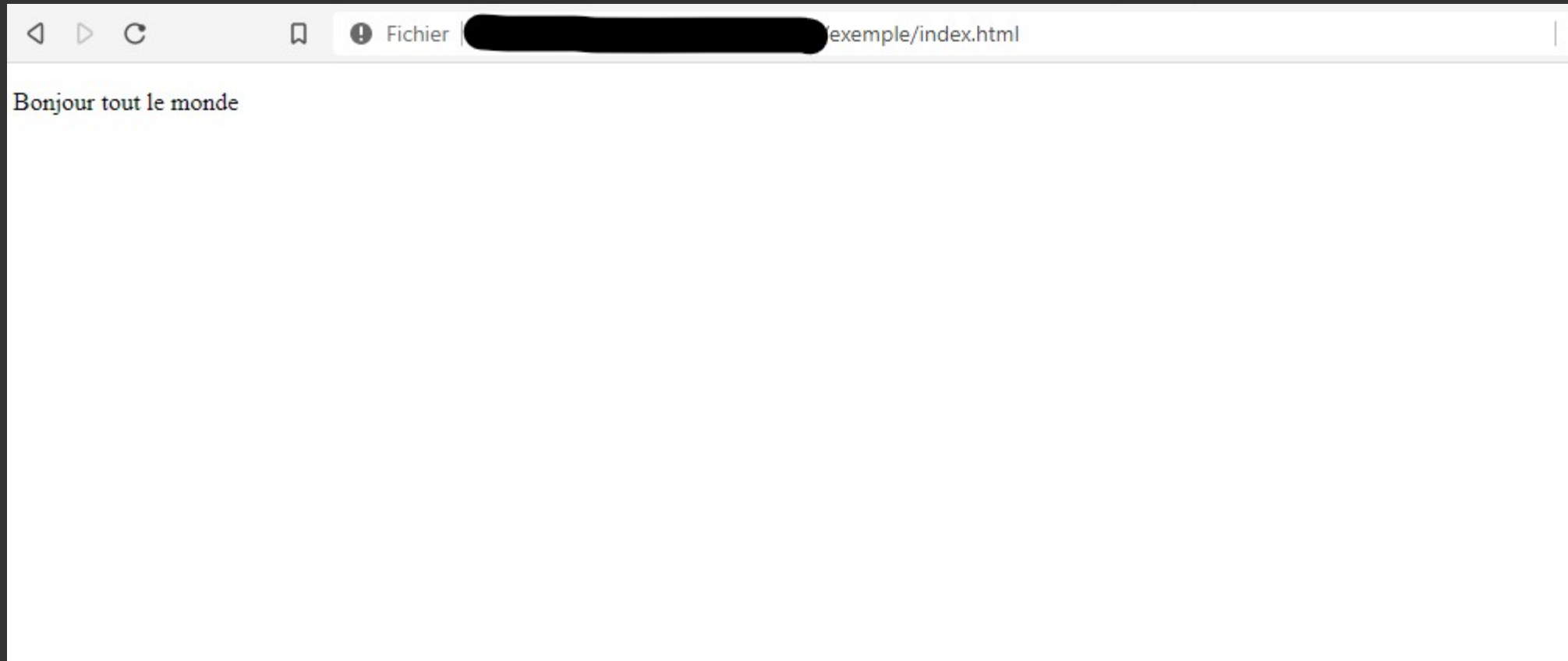
Pratiquons ! - Améliorons notre page

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%89l%C3%A9ments_en_ligne
- https://developer.mozilla.org/fr/docs/Web/HTML/%C3%89l%C3%A9ments_en_bloc

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
 - `` : liste à puces non ordonnées
 - `` : élément d'une liste à puces
 - [...]

Sources :

- https://developer.mozilla.org/fr/docs/Web/HTML/%C3%89l%C3%A9ments_en_ligne
- https://developer.mozilla.org/fr/docs/Web/HTML/%C3%89l%C3%A9ments_en_bloc

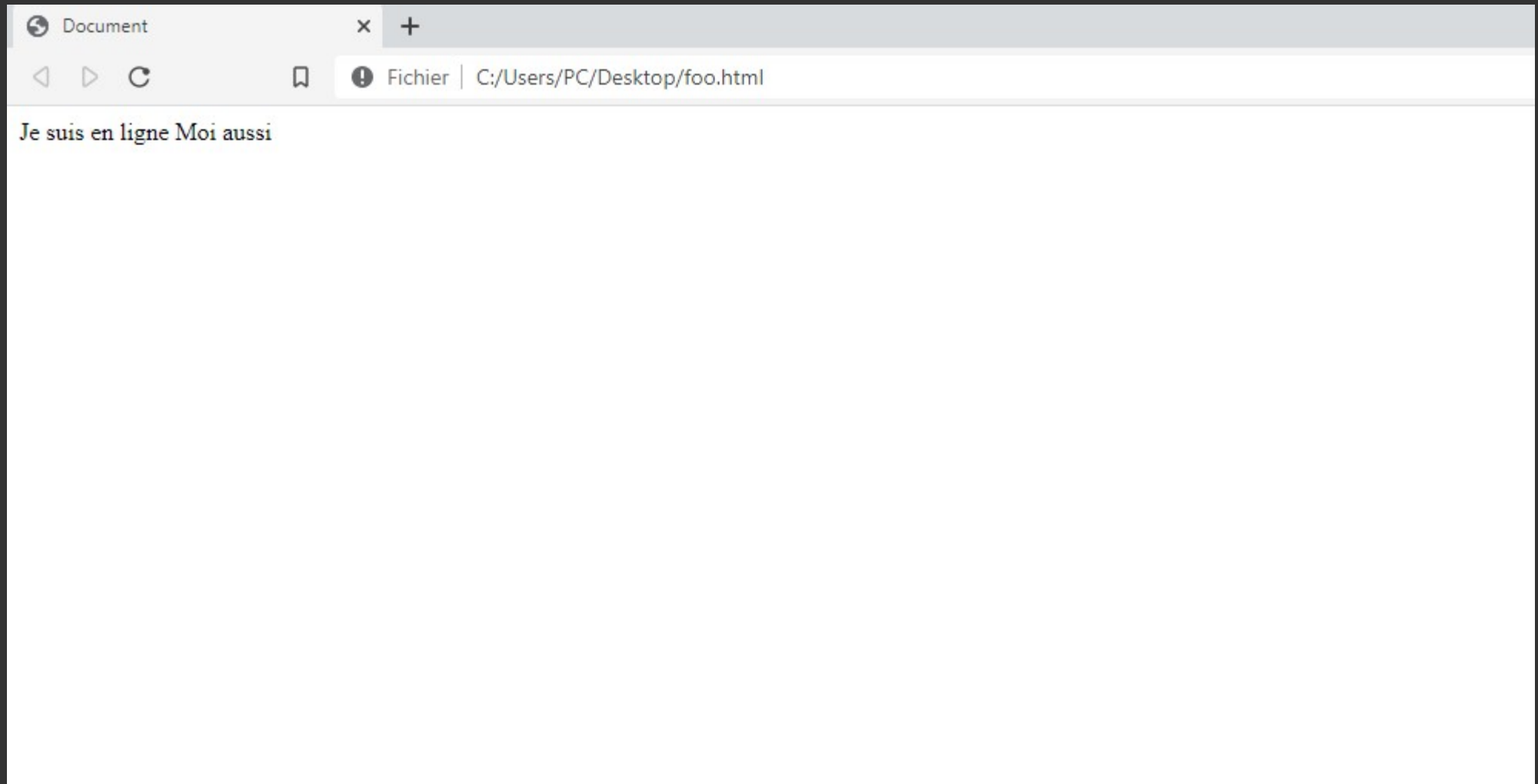
Pratiquons ! - Découvrons les balises “inline”

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>`

`Je suis en ligne`

`Moi aussi`

`</p>`

Mettons nos balises `` dans une balise de type “block” pour écrire du code valide

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
 - `` : conteneur générique pour le texte
 - `<a>` : lien
 - `<select>` : liste d'options
 - [...]

Sources :

- https://developer.mozilla.org/fr/docs/Web/HTML/%C3%89l%C3%A9ments_en_ligne
- https://developer.mozilla.org/fr/docs/Web/HTML/%C3%89l%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/Web/HTML/Attributs>

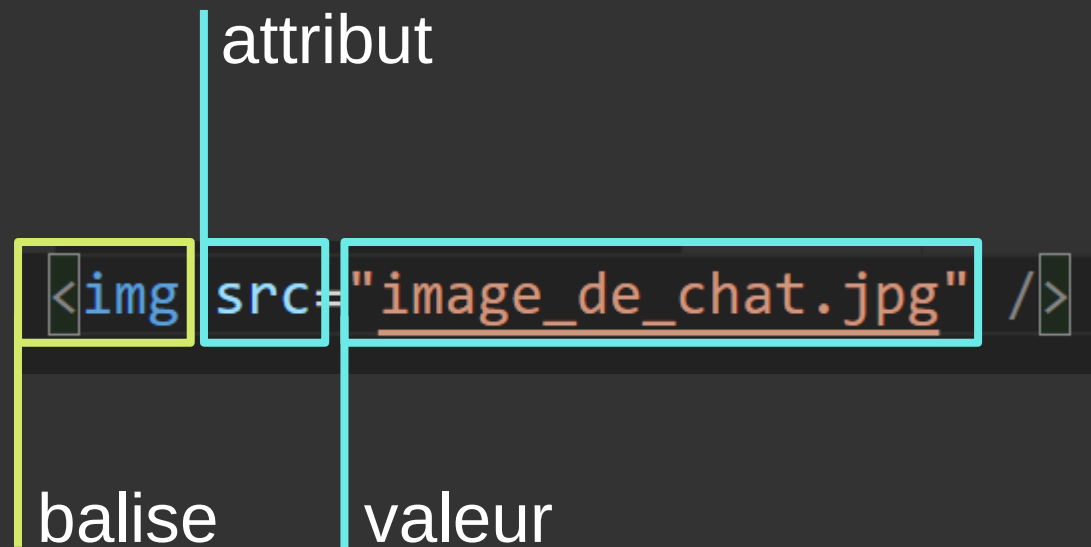
Balises HTML - Attributs

- 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-attributes
 - 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/Web/HTML/Attributs>

Balises HTML - Attributs



Sources :

- https://developer.mozilla.org/fr/docs/Apprendre/HTML/Balises_HTML
- <https://developer.mozilla.org/fr/docs/Web/HTML/Attributs>

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 :

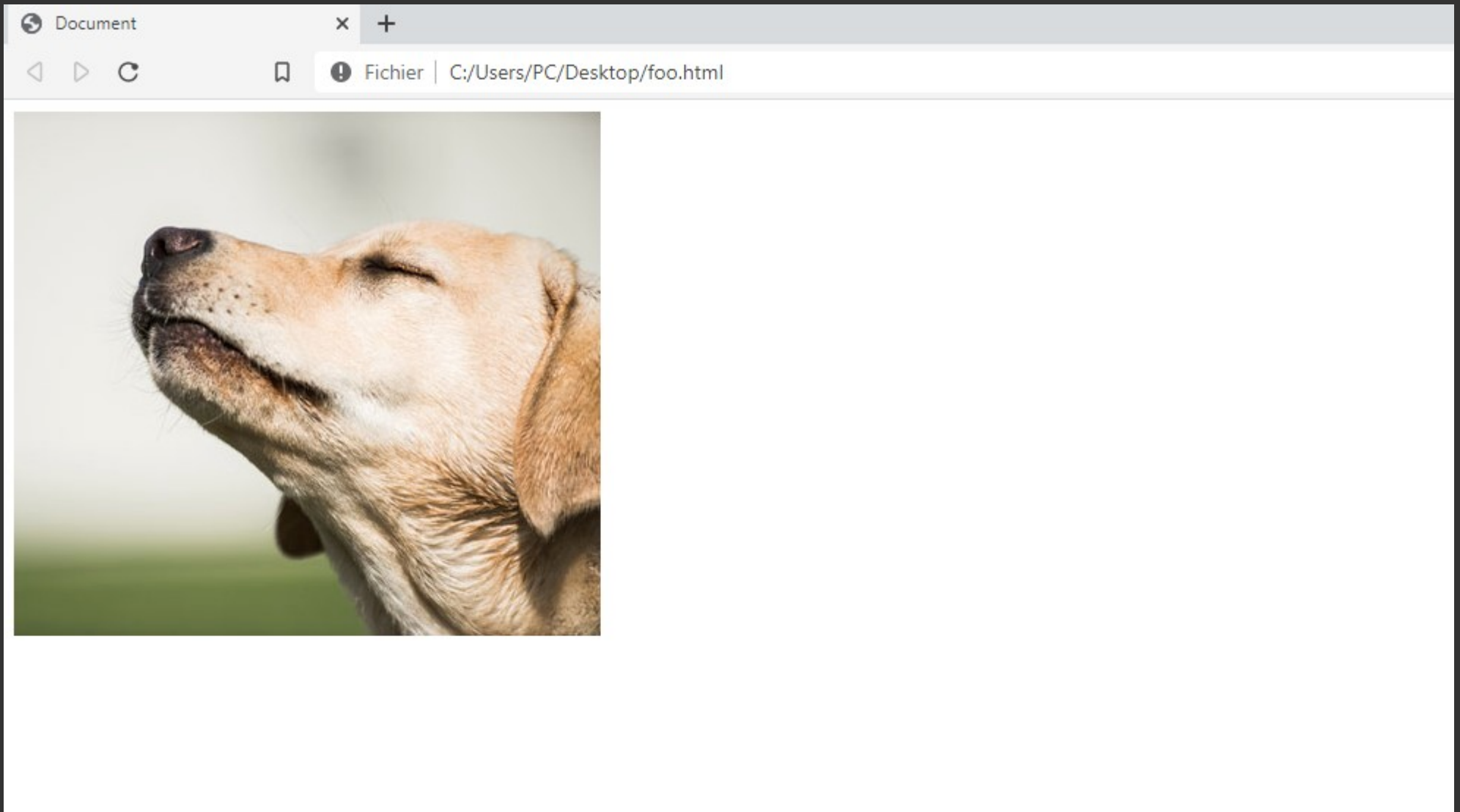
```

```

Sources :

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

Notre page générée



Attribut src – Chemins relatifs et absolus

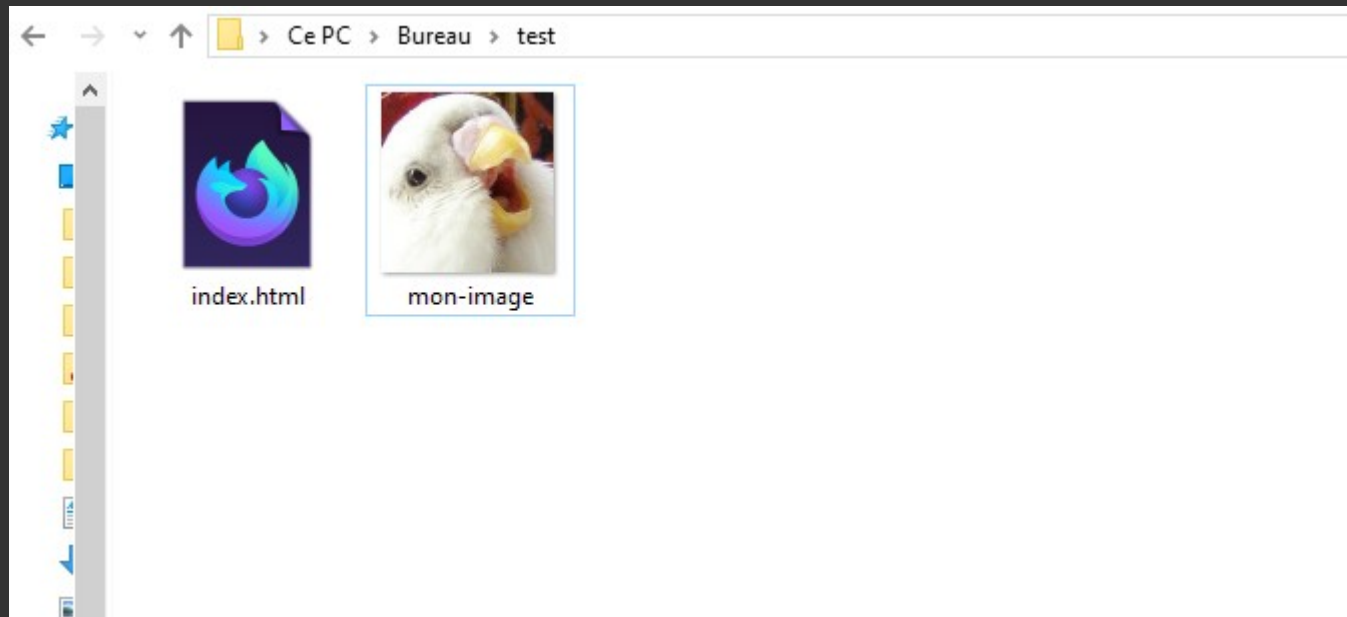
- En HTML les fichiers ne sont pas embarqués, ils sont liés
 - Si je déplace mon fichier d'image sans mettre à jour l'attribut "src" de ma balise son image ne s'affichera plus. Cette règle ne s'appliquant pas si l'image est un serveur distant
 - La valeur de l'attribut "src" peut-être un chemin relatif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

Chemins relatifs et absolus

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



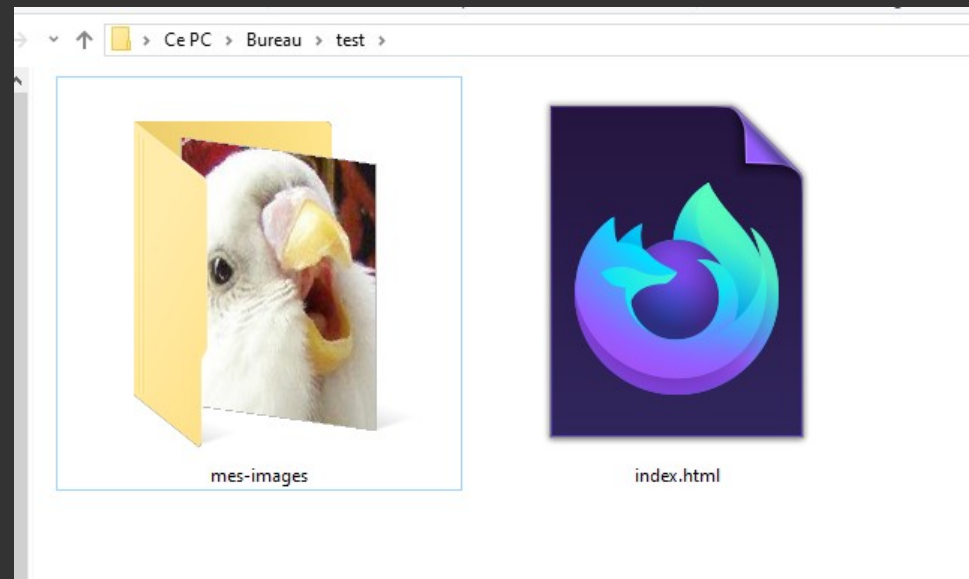
Chemin relatif = ./mon-image.jpg

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

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



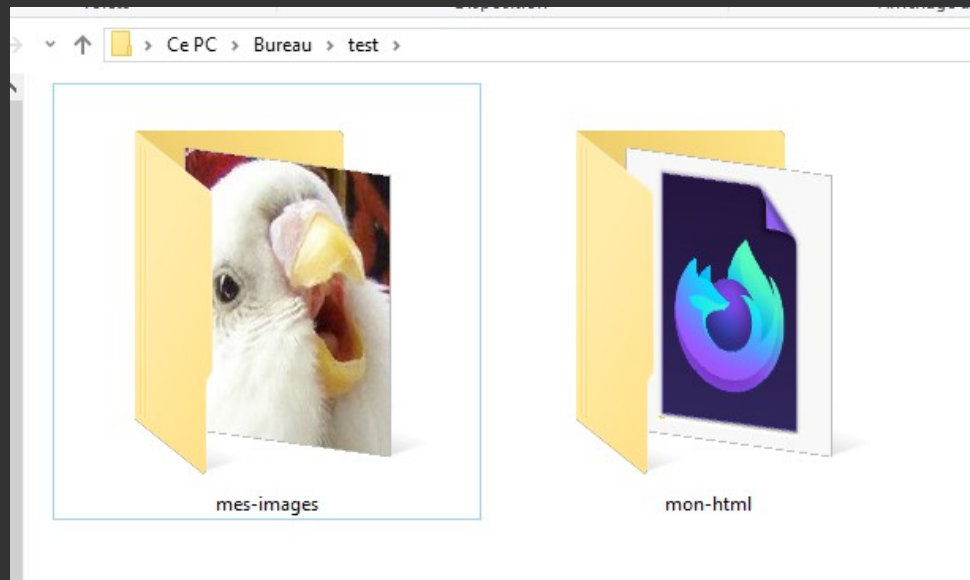
Chemin relatif = `./mes-images/mon-image.jpg`

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

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

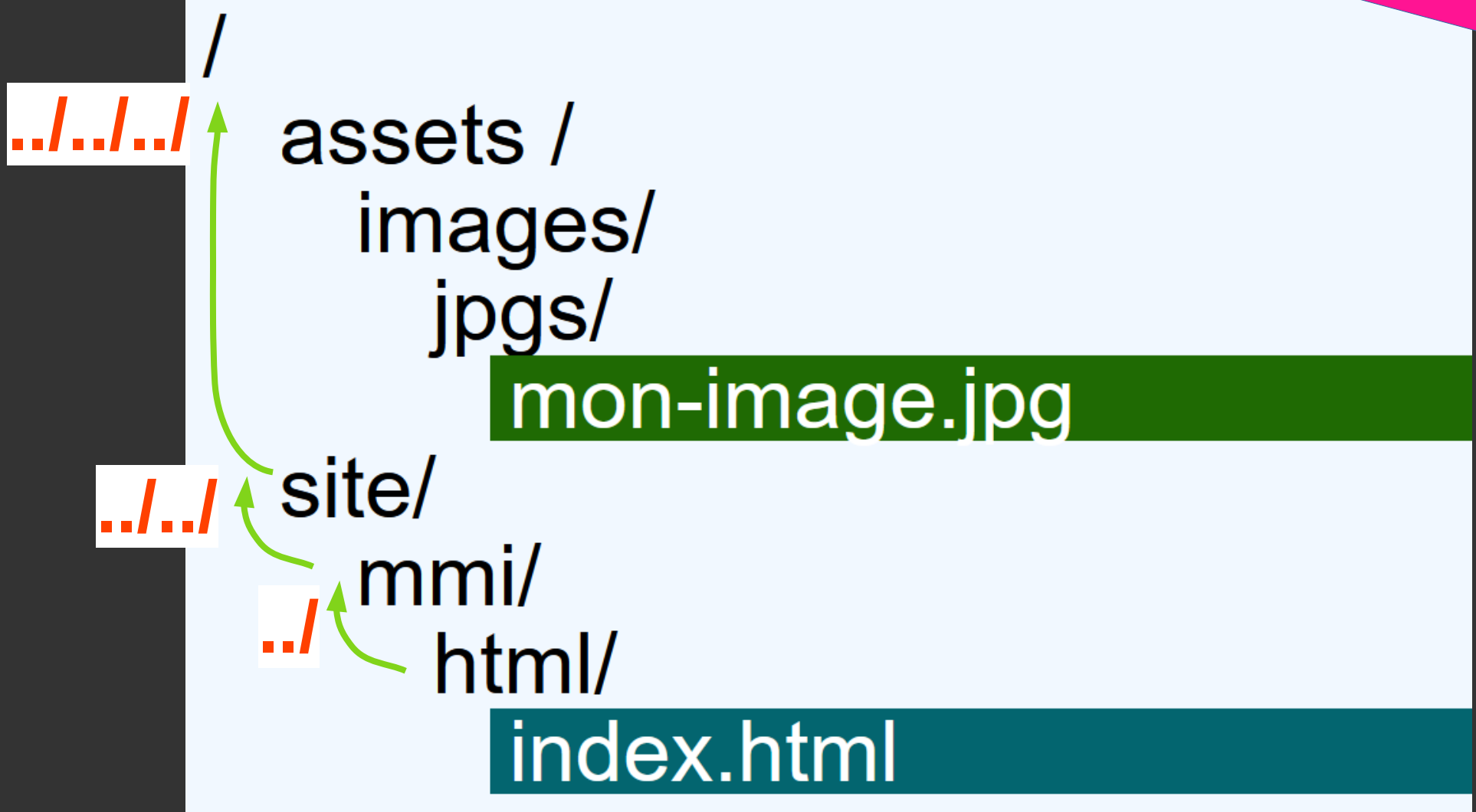


Chemin relatif = ../mes-images/mon-image.jpg

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

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

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.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

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/Web/HTML/Attributs>

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/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 :

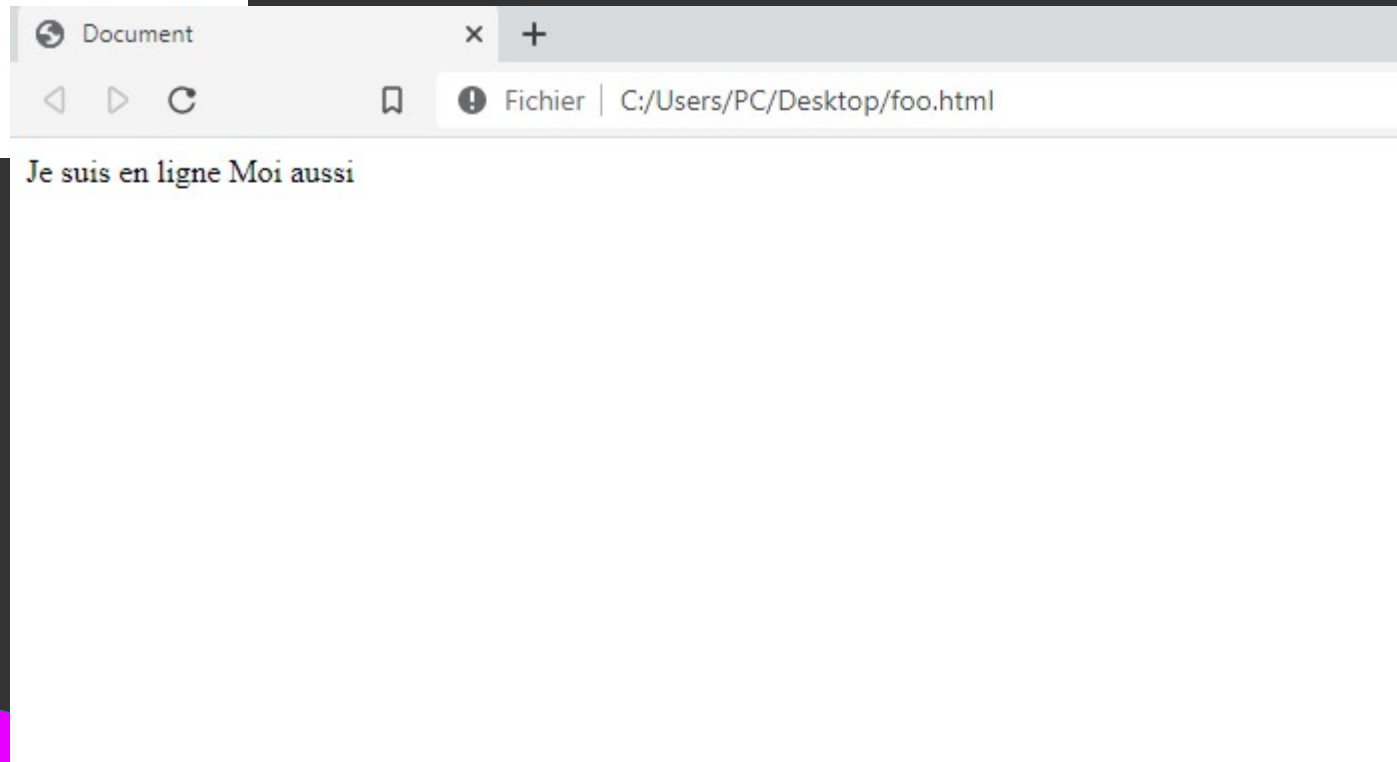
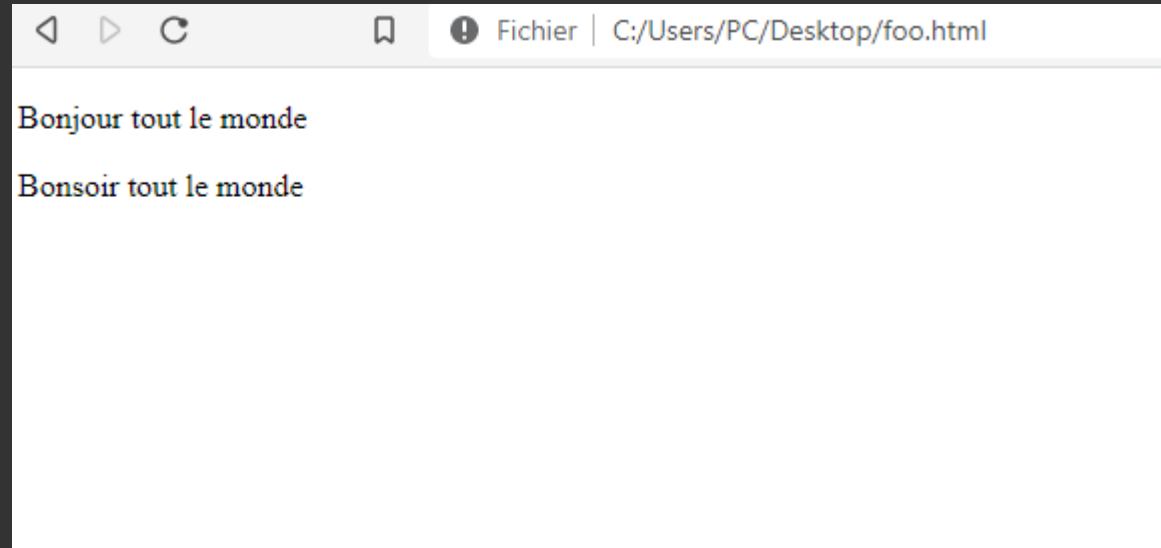
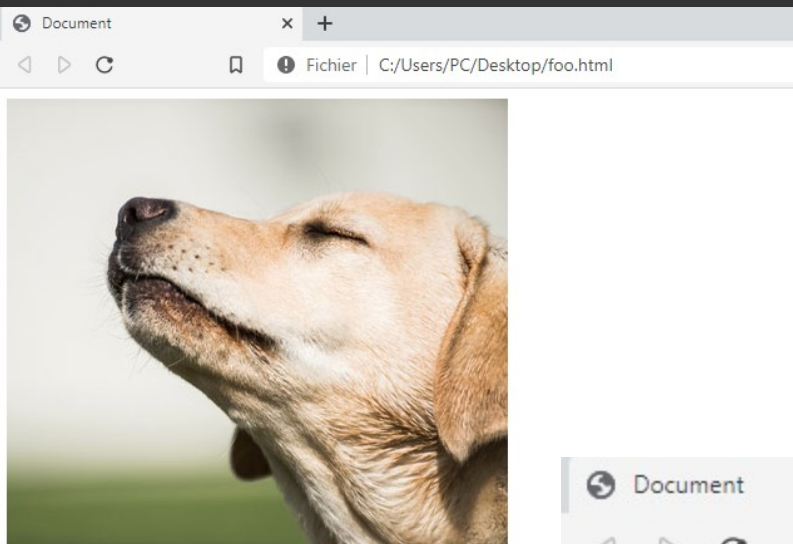
- https://developer.mozilla.org/fr/docs/Apprendre/HTML/Balises_HTML
- <https://developer.mozilla.org/fr/docs/Web/HTML/Attributs>

html – Récapitulatif

- Langage de description
 - Les structures comme “if” ou “while” sont impossibles
- Utilisation de balises qui peuvent être
 - auto-fermantes `
`
 - “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



Pas très beau non ?

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

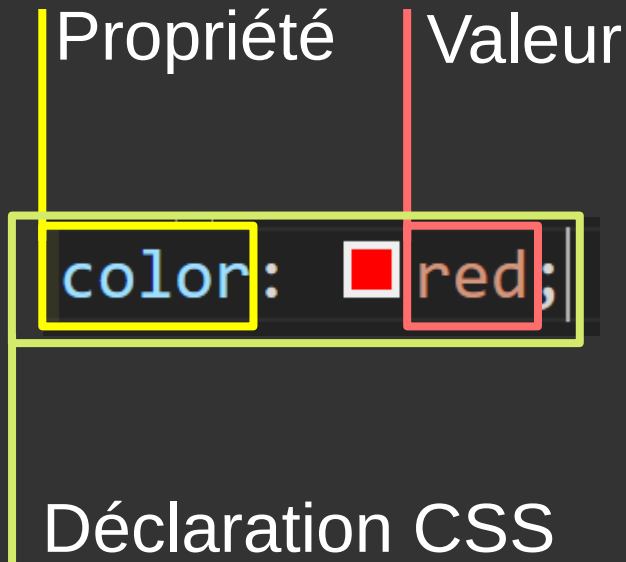
CSS – Feuilles de style

- 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



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 `/* */`

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

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

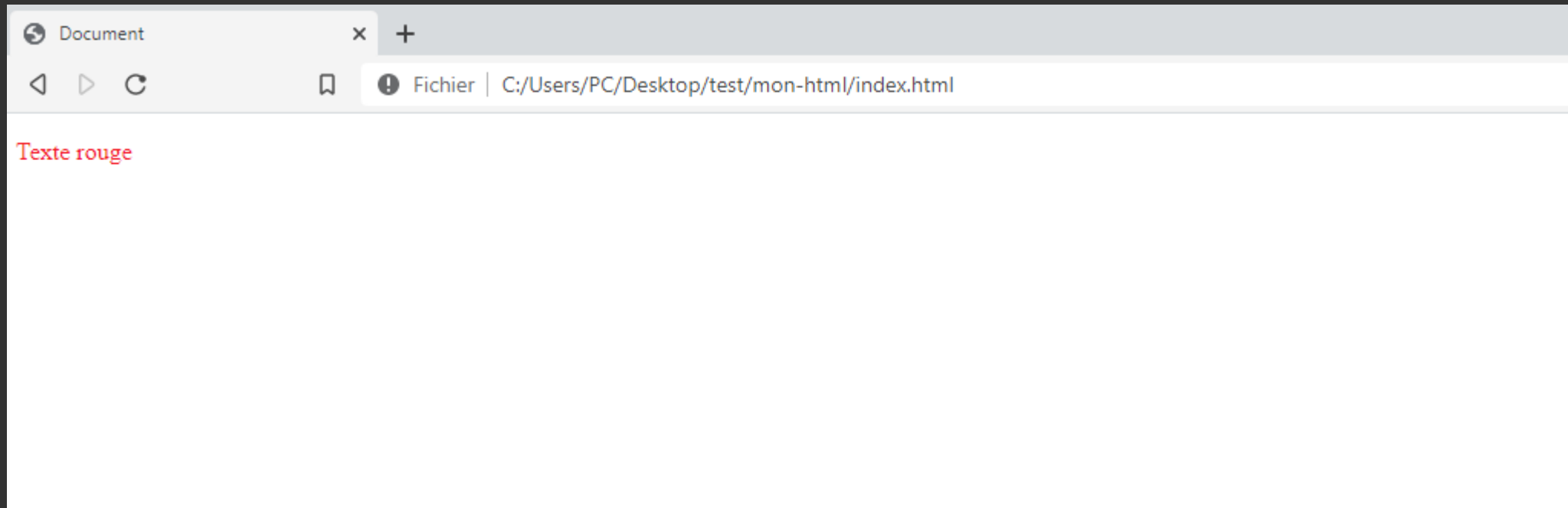
Pratiquons ! - Écrivons notre premier css

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>

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

CSS – Sélecteur css


sélecteur

commentaire

valeur

p

{

color:  red;

/*font-size: 30px;*/

text-decoration: underline;

}

propriété

déclaration css

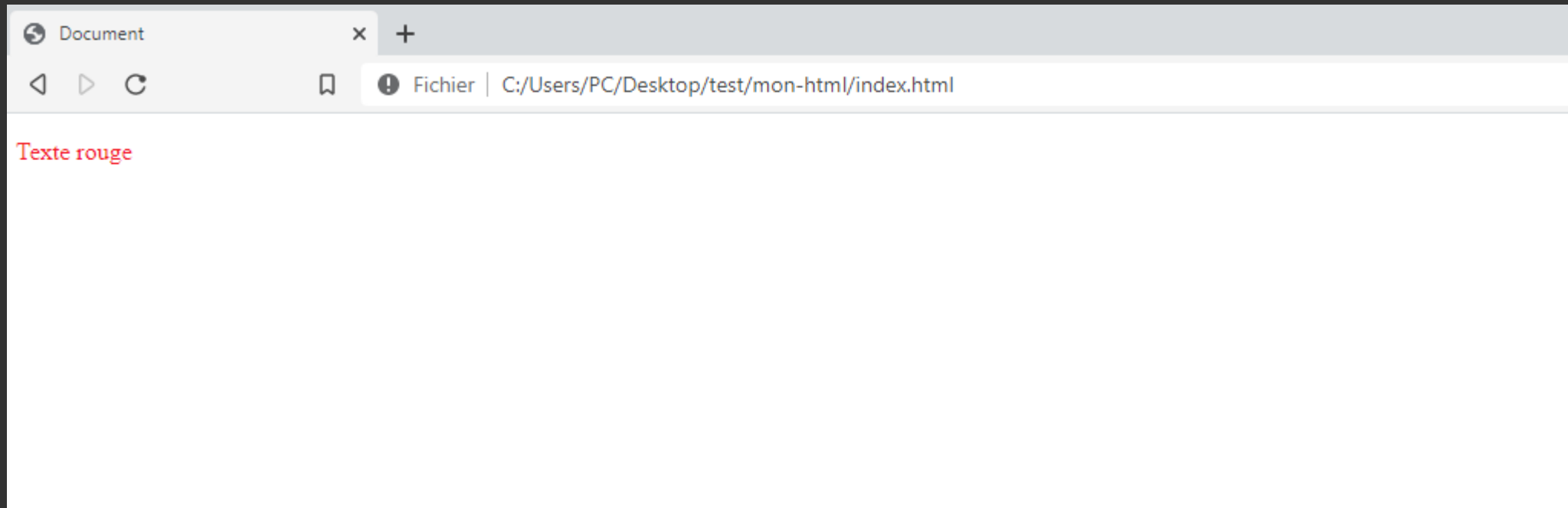
Pratiquons ! - Écrivons notre balise <style>

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>
```

Notre page générée



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>

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

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"**

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

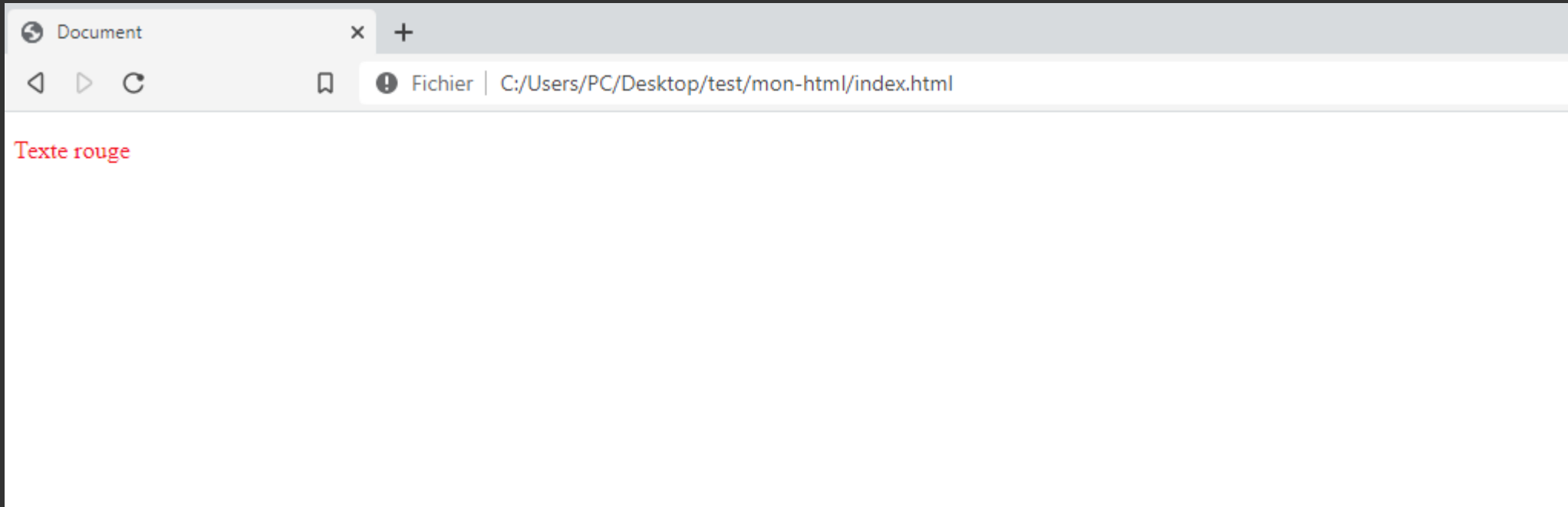
- Dans le fichier html rajouter la ligne suivante dans la balise <head>

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

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

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


Notre page générée



Pas très surprenant

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>

Pratiquons ! - Ajoutons nos premières classes

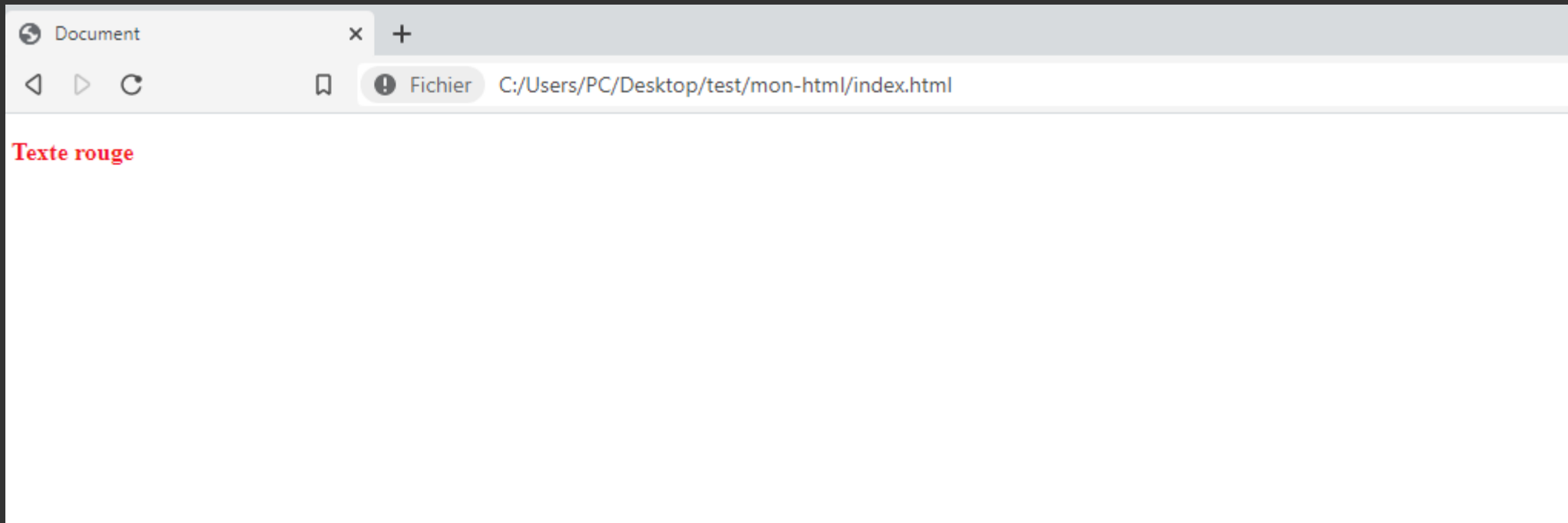
- Dans le fichier "index.html", écrire le contenu suivant

```
<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/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">
```

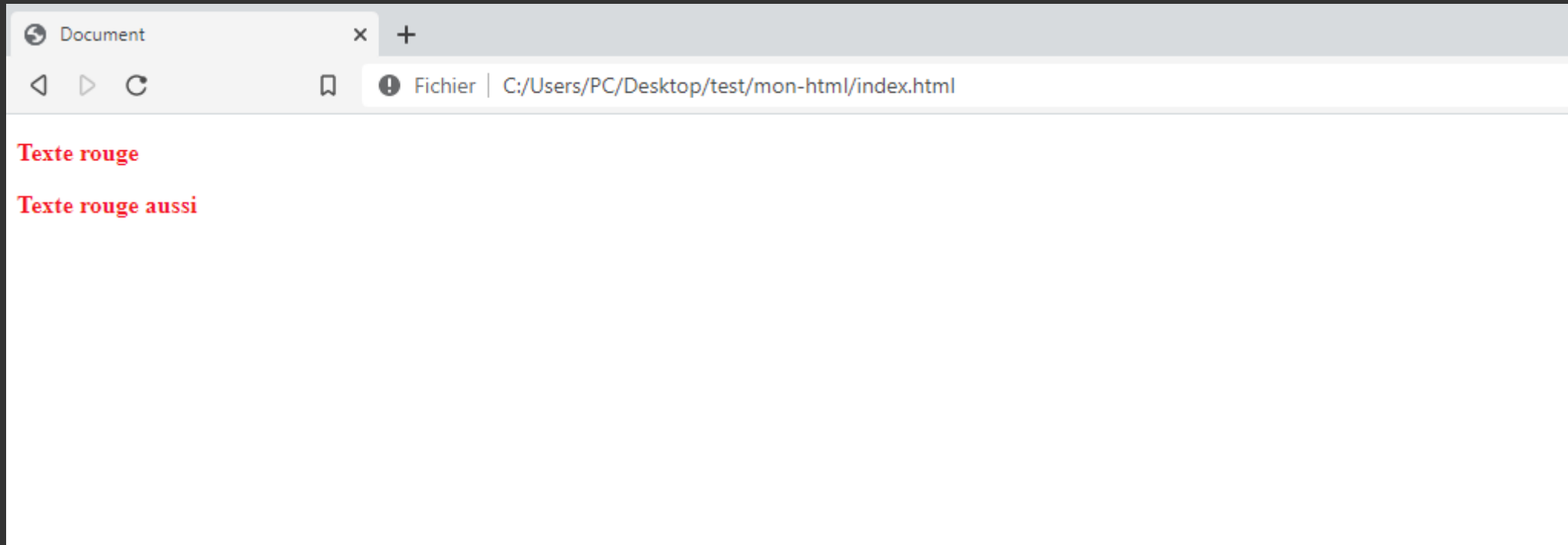
Texte rouge aussi

```
</span>
```

Sources :

- https://developer.mozilla.org/fr/docs/Web/CSS/S%C3%A9lecteurs_de_classe
- https://developer.mozilla.org/fr/docs/Web/CSS/Selector_list

Notre page générée



Lui aussi est rouge grâce au système de classes

CSS – Sélecteur de classe

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

```
<span class="text-important informations">
```

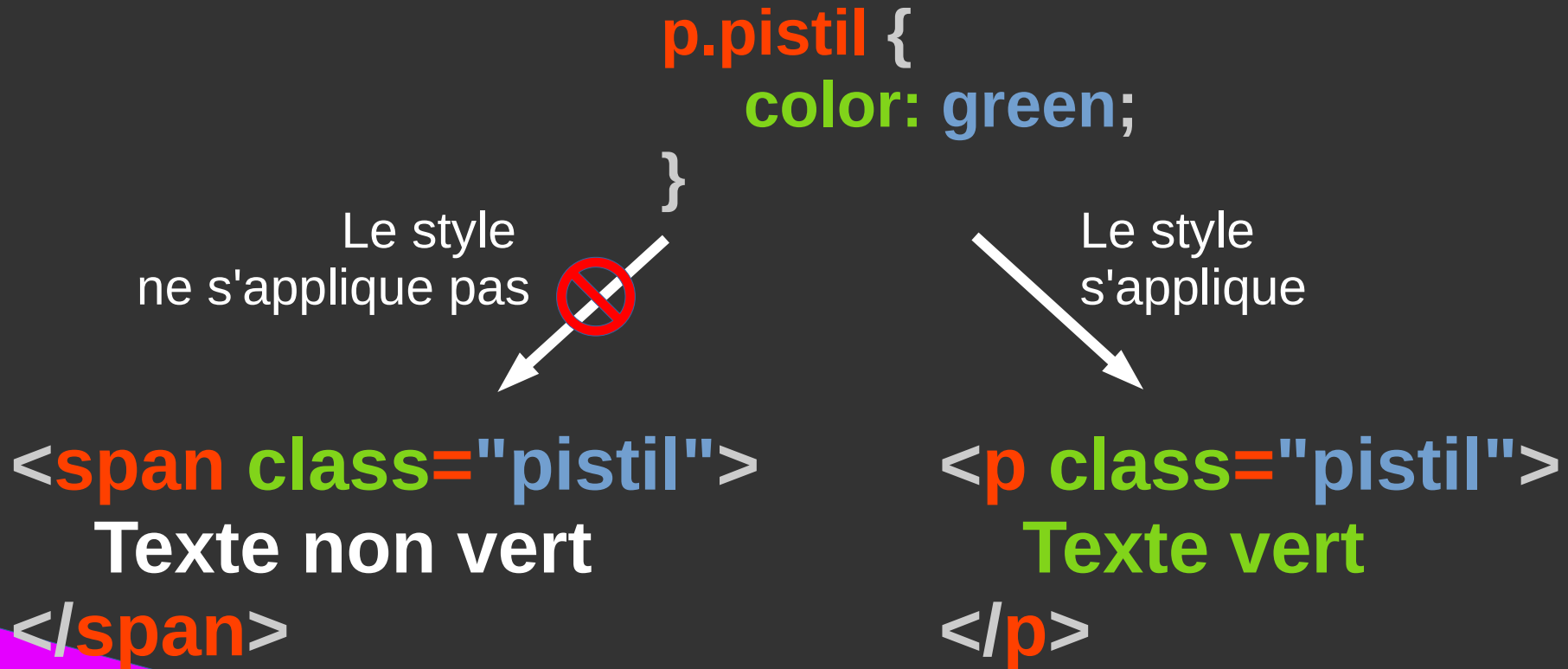
Texte rouge aussi

```
</span>
```

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

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

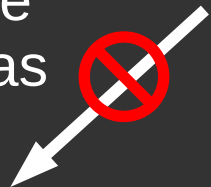


CSS – Sélecteur css

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

```
.pistil.fleur {  
  color: green;  
}
```

Le style
ne s'applique pas



```
<span class="pistil">  
  Texte non vert  
</span>
```

Le style
s'applique



```
<p class="pistil fleur">  
  Texte vert  
</p>
```

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

```
.herbier .fleur {  
    color: green;  
}
```

Le style
s'applique
ici

```
<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

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"

CSS – Pratiquons ensemble

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

Pour aller plus loin


Devoir "Sélecteurs css" sur ENT dans le dossier Travaux pratiques #1

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;  
}
```

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;  
}
```

CSS – Feuilles de style – La cascade

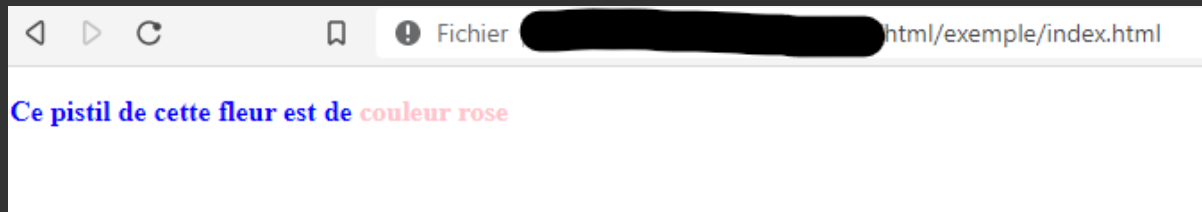
```
<body>
  <div>
    <p class="description">
      Ce pistil de cette fleur est de
      <span class="text-pistil"> couleur rose </span>
    </p>
  </div>
</body>
```

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 `` 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>

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



Mauvaise pratique

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

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 ?