

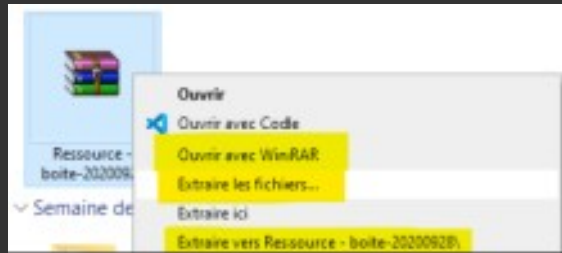
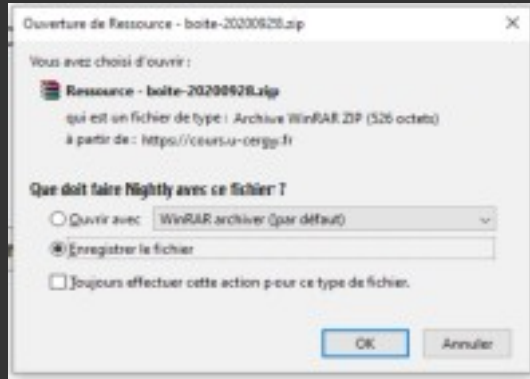
Intégration Web

MMI 1 – TP#1 S1



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

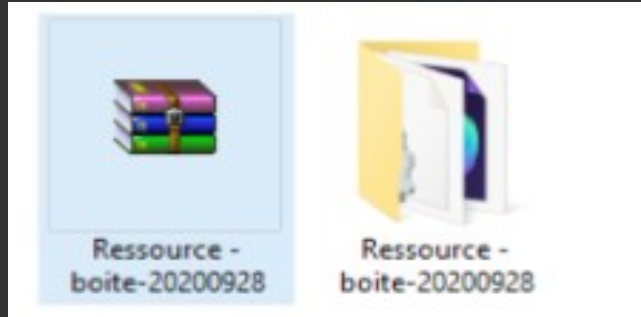
Petit point sur les ressources de cours



Menu contextuel sur l'archive (clic droit). Choisir un des choix surligné.

- Les ressources se présentent sous la forme d'une archive une fois téléchargées
- L'archive doit être impérativement extraite sinon vous ne serez pas capables de réaliser correctement les pratiques

Petit point sur les ressources de cours



- Une fois extraite, vous devez obtenir un dossier avec le contenu de l'archive
- Libre à vous de travailler dans le dossier OU en copier le contenu pour le coller ailleurs
- **En aucun vous devez travailler dans l'archive**
- Une fois fait, glissez le dossier dans VS Code

Pratiquons ! - Notre première page web (Partie 1)

Pré-requis :

- Avoir la ressource ressources/bases
- A télécharger ici :
<https://download-directory.github.io/?url=https://github.com/DanYellow/cours/tree/main/integration-web-s1/travaux-pratiques/numero-1/ressources>

Sources :

- <https://code.visualstudio.com/Download>

Analysons tout ça ensemble

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

A screenshot of a code editor showing the HTML DOCTYPE declaration. The text "<!DOCTYPE html>" is displayed in a light blue font on a dark background. The opening and closing angle brackets are highlighted with small white boxes.

- 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 pas 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 http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Ma première page</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>

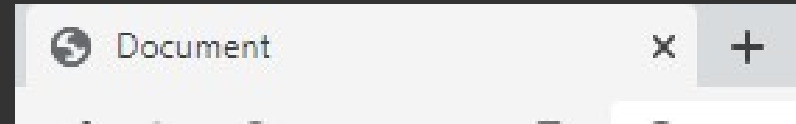
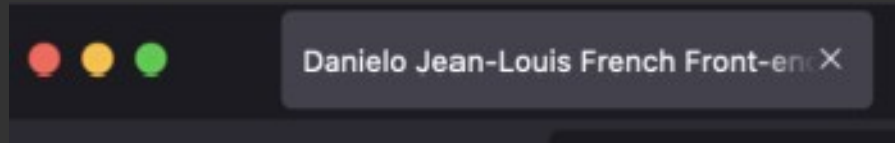
Sources :

- <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>Ma première page</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 <body>

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

- Cette balise est **unique**, il ne doit pas en avoir **plusieurs** par fichier html
- 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>Bonjour la promotion MMI</p>
```

Balise "classique"

```

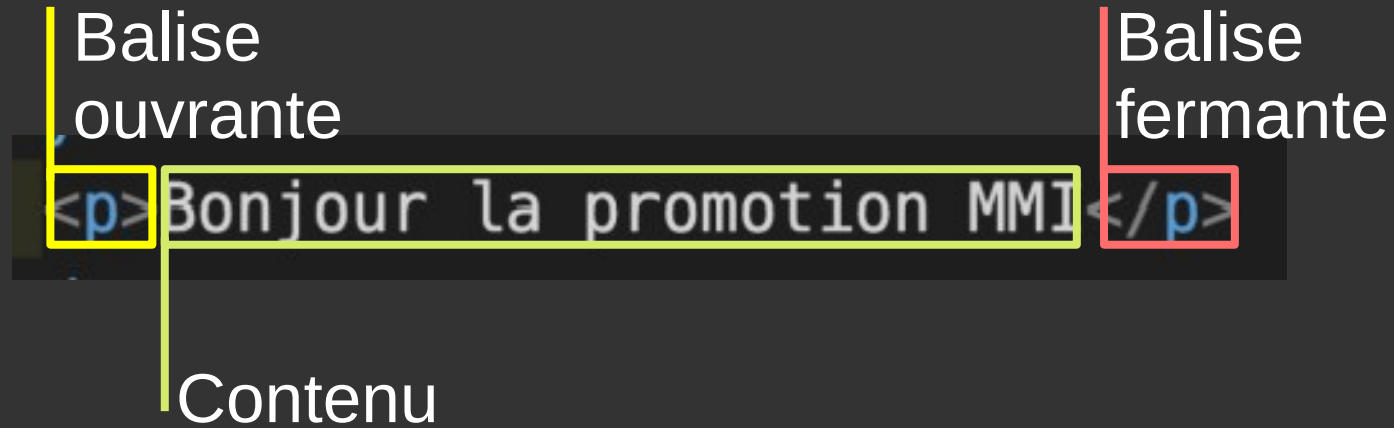
```

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
- https://developer.mozilla.org/fr/docs/Learn/Getting_started_with_the_web/HTML_basics - anglais

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><p>Hello world</p></div>`
 - `<p></p>` : `<p>Bonjour la promotion MMI</p>`
 - `<a>` : `Site 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 "classique"

- Les balises ne peuvent pas se chevaucher

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



Interdit :

La balise `<p>` se ferme avant la balise `<div>`

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



Correct :

Les balises s'ouvrent et se ferment dans le même ordre

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-fermantes

- Ne peut contenir du contenu (pas de balise fermante)
- Possède des attributs parfois nécessaires à son fonctionnement
- La barre oblique à la fin est optionnelle en HTML5
- Exemple de balises "auto-fermantes" :

- `` :

```

```

- `
` :

```
<br>
```

- `<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>
```

```
<section>
  <ul>
    <li>
      <a href="">
        
      </a>
    </li>
    <li>
      <a href="">
        
      </a>
    </li>
    <li>
      <a href="">
        
      </a>
    </li>
  </ul>
</section>
```

Source :

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

* Certaines imbrications ne sont pas valides

Pratiquons ! - Notre première page web (partie 2)

Pré-requis :

- Avoir la ressource ressources/bases
- A télécharger ici :
<https://download-directory.github.io/?url=https://github.com/DanYellow/cours/tree/main/integration-web-s1/travaux-pratiques/numero-1/ressources>

Sources :

- <https://code.visualstudio.com/Download>

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 ! - Notre première page web (partie 3)

Pré-requis :

- Avoir la ressource ressources/bases
- A télécharger ici :
<https://download-directory.github.io/?url=https://github.com/DanYellow/cours/tree/main/integration-web-s1/travaux-pratiques/numero-1/ressources>

Sources :

- <https://code.visualstudio.com/Download>

Balises type bloc - Caractéristiques

- Occupe tout l'espace disponible en largeur
- Force un retour à la ligne après elle (la pile d'assiette)
- Exemple de balises bloc
 - `<div>` : conteneur générique
 - `<p>` : paragraphe
 - `` : 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 ! - Notre première page web (partie 4)

Pré-requis :

- Avoir la ressource ressources/bases
- A télécharger ici :
<https://download-directory.github.io/?url=https://github.com/DanYellow/cours/tree/main/integration-web-s1/travaux-pratiques/numero-1/ressources>

Sources :

- <https://code.visualstudio.com/Download>

Balises `` et `<p>`

- Balises permettant d'afficher du texte (il en existe d'autres)
- `` : balise inline
- `<p>` : balise bloc
- Possibilité d'imbriquer des `` dans un `<p>` mais pas l'inverse

Sources :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/p>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/span>

Balises `` et `<p>`

- Possibilité d'imbriquer un `` dans un ``
- Impossible d'imbriquer un `<p>` dans un `<p>`

Sources :

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/p>
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/span>

Balise

- Permet de gérer des retours à la ligne **uniquement** quand ils sont importants :
 - poèmes
 - adresses
 - [...]
- Ne pas utiliser la balise le cas échéant

Sources :

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

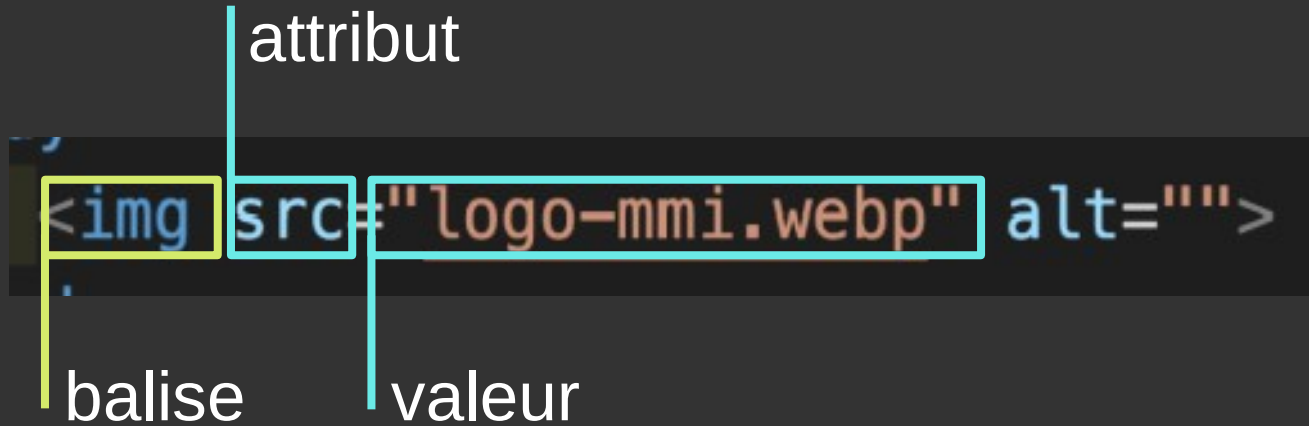
Balises type inline - Caractéristiques

- Peut contenir d'autres balises "inline"
- Occupe la place de son contenu
- 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

Balises HTML - Attributs



Sources :

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

Balises HTML - Attributs

- Représentent les informations complémentaires d'une balise
- Se placent **toujours dans la balise ouvrante** d'une balise
- Chaque attribut (+ valeur) doivent être espacé pour plus de lisibilité

Sources :

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

Balises HTML - Attributs

- 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)

Sources :

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

Balises HTML - Attributs

- La valeur d'un attribut doit être entre guillemets (simples / ' ou doubles / ")
 - Si on ouvre avec des guillemets simples (') on ferme avec des guillemets simples (') idem pour les guillemets doubles
 - On ne peut donc pas mixer l'ouverture et la fermeture
- Certains attributs sont propres à certaines balises
- Ils peuvent être indispensables pour certaines balises

Sources :

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

Chemins relatifs et absolus

- Chemin absolu:
 - A utiliser si votre ressource est extérieure à votre site

```

```

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

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

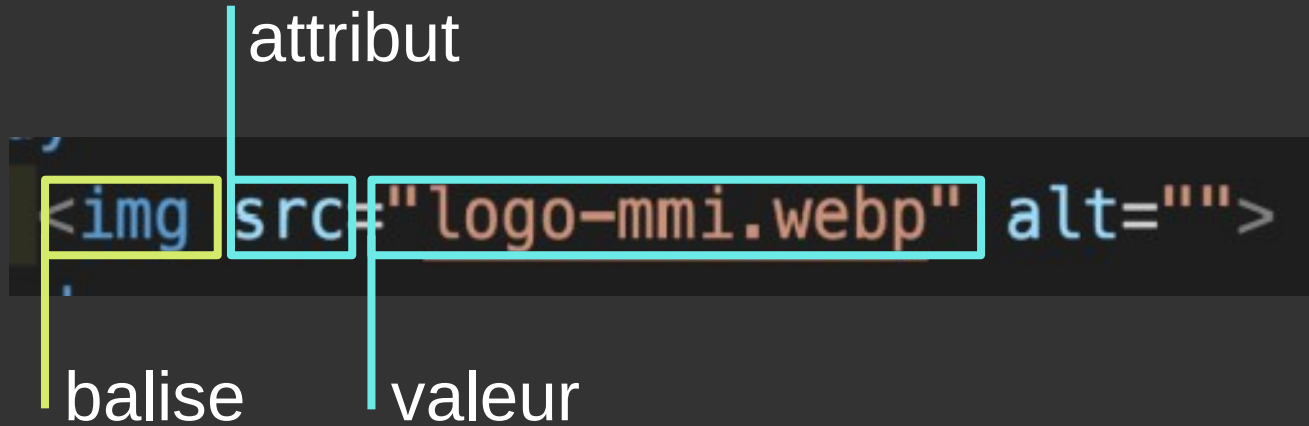
Balises HTML - Attributs

- Si les attributs sont prédéfinis, ils peuvent complétés par les data-attributes
 - Attributs s'écrivant sous la forme data-{valeur}="param"
Exemple : data-etudiants="63"
 - 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 ! - Découvrons les attributs – Partie 1

Pré-requis :

- Avoir la ressource ressources/attributs
- A télécharger ici :
<https://download-directory.github.io/?url=https://github.com/DanYellow/cours/tree/main/integration-web-s1/travaux-pratiques/numero-1/ressources>

Point accessibilité – l'attribut "alt"

- Permet aux lecteurs d'écran de "lire" l'image
- **Doit toujours être présent même vide**
- Sert au référencement du site
- Mieux vaut un alt vide qu'un alt mal nommé
- S'affiche lorsque l'image est absente
 - Ne s'affichera pas si l'image prend du temps à charger

Attribut src – Chemins relatifs et absolus

- En HTML les fichiers ne sont pas embarqués, **ils sont liés**
 - Déplacer un fichier peut avoir des conséquences
 - 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

Attribut src – Chemins relatifs et absolus

- Le chemin est le cheminement entre le fichier qui charge le fichier (ex : index.html) et le fichier qui va être chargé (ex : image.png)
- Caractères clés :
 - ./ : *racine du chemin
 - / : séparateur entre les dossiers
 - ../ : *permet de remonter d'un niveau dans la structure des dossiers. Se place principalement au début du chemin

* : Ces caractères s'utilisent **uniquement** pour les chemins relatifs

Chemins relatifs

- Ne jamais, jamais utiliser l'anti-slash (\) comme séparateur de dossiers dans une url ou un chemin de ressource

```

```



Anti-slash dans l'url = interdit

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 absolus

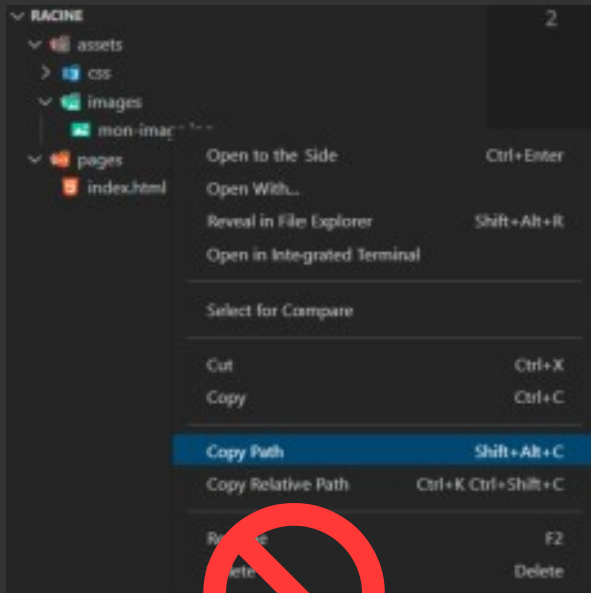
- Chemin pointant vers une **ressource externe** à votre projet
 - **N'est pas influencé par la structure de vos dossiers**
 - Ex : Si vous déplacez votre fichier html, ça ne changera rien
 - Exemple : lien vers une image sur instagram
- Ne pas utiliser les chemins absolus propres à votre ordinateur

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 absolus

- Ne jamais, jamais utiliser le chemin absolu proposé par votre ordinateur
 - Ni la fonctionnalité de VS Code "Copy path"



- En utilisant la fonctionnalité "copy path" (clic droit sur le fichier), on obtient un chemin comme : "E:\projects\racine\assets\images\mon-image.jpg"
- **Ce chemin ne va fonctionner QUE sur cet ordinateur.** On ne veut pas ça

Chemins relatifs

- Chemin pointant vers une **ressource interne** à votre projet
 - **Est influencé par la structure de vos dossiers**
 - Ex : Si vous déplacez votre fichier html, ceci aura des conséquences
 - Exemple : lien vers une autre page html
- Le chemin est toujours construit par rapport au fichier qui charge la ressource

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

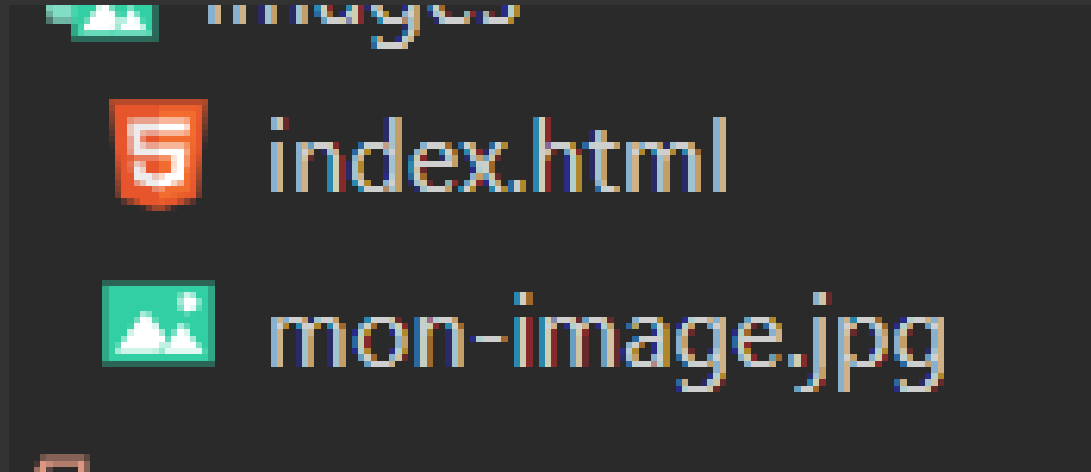
Le chemin est toujours calculé par rapport au fichier qui veut charger la ressource

Chemins relatifs

Dans le cadre du cours, nous chargerons nos fichiers depuis les fichiers HTML (et parfois CSS)

Chemins relatifs – Exemple 1

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



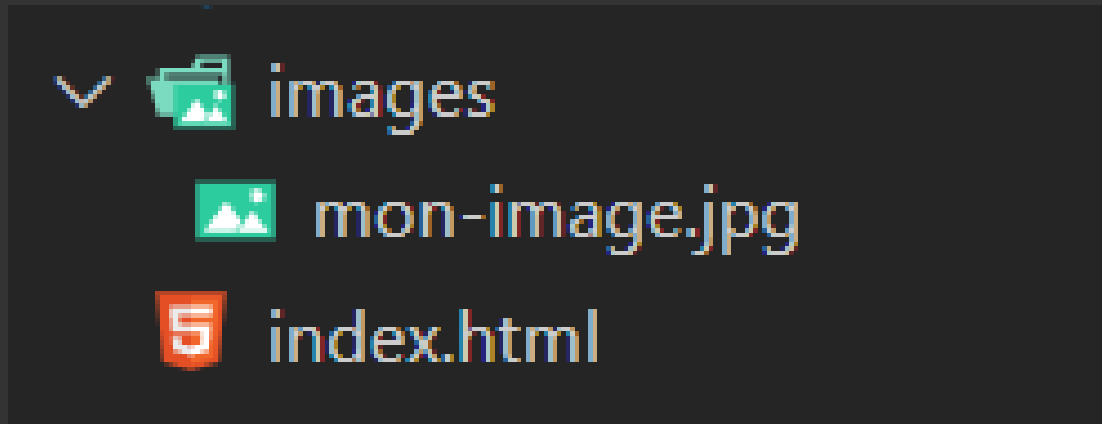
Chemin relatif = ./mon-image.jpg

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 – Exemple 2

- Si les fichiers ne sont pas dans le même dossier, le caractère slash (/) sert de séparateurs entre les dossiers



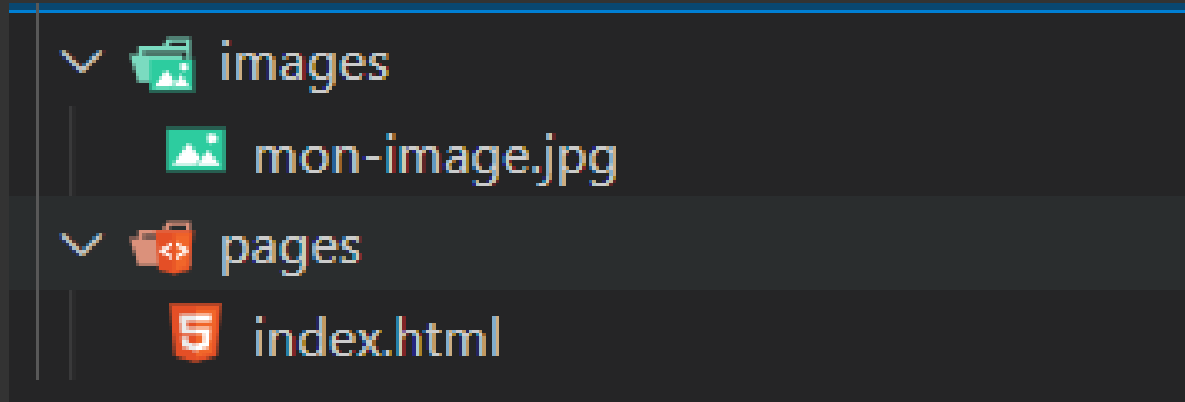
Chemin relatif = ../images/mon-image.jpg

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 – Exemple 3

- 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

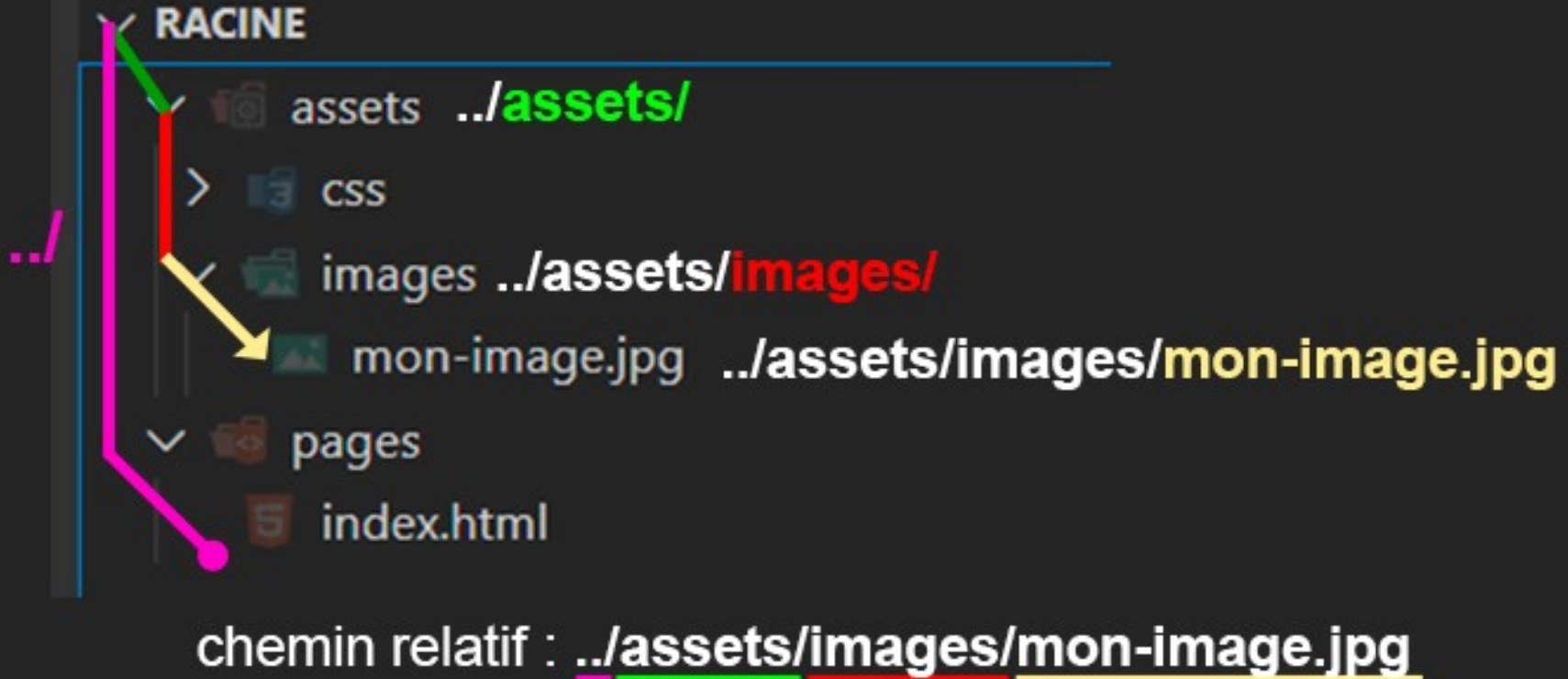


Chemin relatif = ../images/mon-image.jpg

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 – En détails

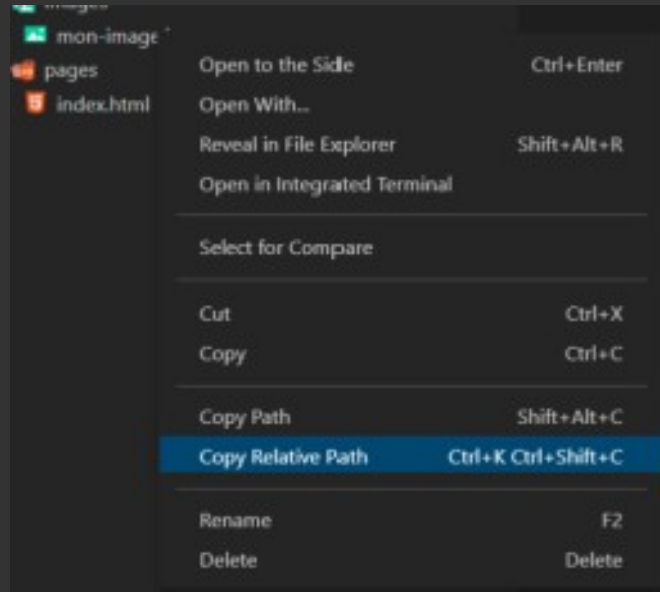


Sources :

- <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 – Astuce VS Code

- Le logiciel permet de récupérer le chemin relatif d'un fichier **par rapport au projet**. Pour ce faire :



Menu quand vous faites un clic droit sur un fichier

- Faites un clic droit sur le nom du fichier (dans le panneau de gauche OU les onglets, il faut que le fichier soit ouvert)
- Sélectionnez l'option "copy relative path" (copier le chemin relatif)
- Collez-le où vous en avez besoin
 - N'oubliez pas de faire un ajustement si nécessaire

Note : VSCode propose un système d'auto-complétion quand vous écrivez vos chemins

Règles de nom de fichier

- Préférer les noms en minuscule (et extension de fichier)
 - Windows n'est pas sensible à la casse mais UNIX, oui.
Donc : mon-image.jpg \neq MON-IMAGE.JPG
 - +90 % des serveurs sont sous système UNIX
- Éviter les espaces
 - Remplacer par un tiret (-) ou un underscore (_)
- Éviter les caractères spéciaux
 - Accents, apostrophes...
- Préférer des noms clairs et représentatifs
 - Pour savoir le contenu de vos fichiers sans les ouvrir

Pratiquons ! - Découvrons les attributs – Partie 2/3

Pré-requis :

- Avoir la ressource `ressources/attributs`

A télécharger ici :

<https://download-directory.github.io/?url=https://github.com/DanYellow/cours/tree/main/integration-web-s1/travaux-pratiques/numero-1/ressources>

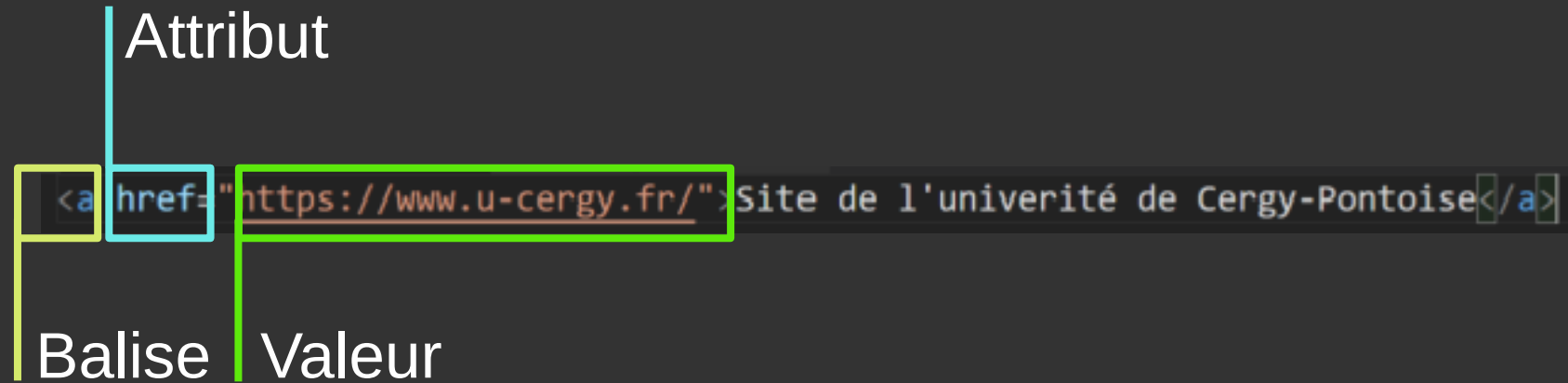
Balises HTML - Attributs

- Permettent d'ajouter des fonctionnalités supplémentaires aux balises
- Certains attributs sont exclusifs à certaines balises
 - Exemple : l'attribut "src" n'aura aucun effet sur les balises `<p>` ou ``
- L'ordre des attributs dans une balise (ouvrante) n'a aucune importance

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> - Liste tous les attributs disponibles

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. La valeur 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>

Pratiquons ! - Découvrons les attributs – Partie 4

Pré-requis :

- Avoir la ressource `ressources/attributs`

A télécharger ici :

<https://download-directory.github.io/?url=https://github.com/DanYellow/cours/tree/main/integration-web-s1/travaux-pratiques/numero-1/ressources>

Point accessibilité – l'attribut "title"

- Permet d'ajouter des informations supplémentaires à un élément
- Ne doit pas se substituer à du texte critique visible sans interaction (survol uniquement)
 - **Ne s'affiche pas pour les utilisateurs au clavier**

html – Récapitulatif

- Langage de description
 - Les structures comme “if” ou “while” sont impossibles
- Utilisation de balises qui peuvent être
 - auto-fermantes `
`
 - “classique” `<p>Hello world</p>`
- 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 via le CSS
- Les balises peuvent s'imbriquer
 - Attention à leurs fermetures
- Certaines balises ont un rôle qui leur est propre
 - Gare au respect de la sémantique

Pour s'entraîner :
"Test – Bases du HTML" sur ENT

A propos des tests

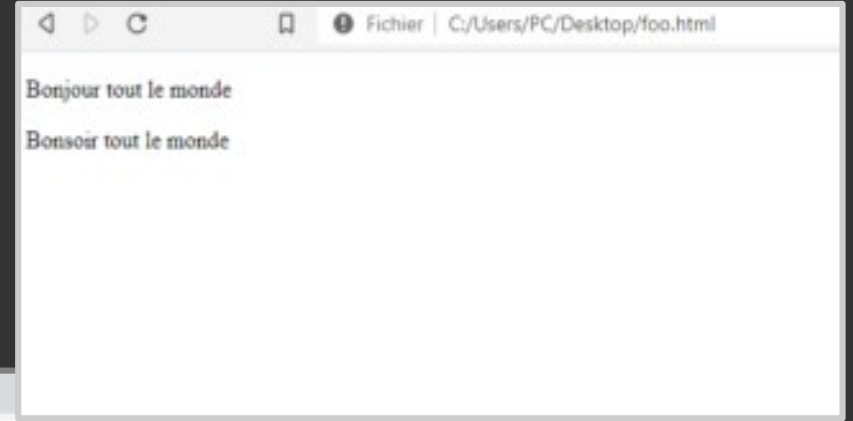
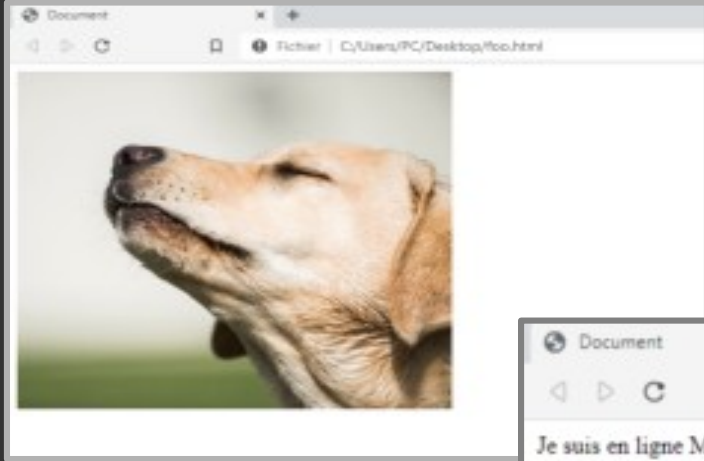
- Tests obligatoires mais non notés
 - Vous êtes obligé de faire le test du TP1 pour accéder à celui du TP2, etc.
 - Indispensable pour accéder au test final qui sera noté

A propos des tests

- Reprennent les notions des TP passés
 - Ex : test TP#3 = questions TP#1,2,3
 - Des questions nécessitent parfois des recherches en ligne
- Plus de détails dans la description des tests et la partie "annonces" sur ENT

Questions ?

Voilà ce que nous avons produit jusqu'à présent



Pas très beau non ?

CS5



Dans le prochain TP

