

URL

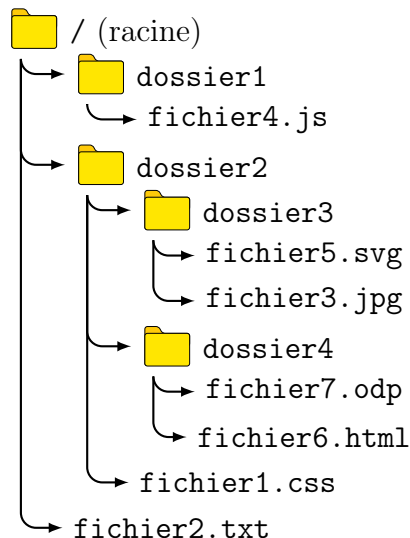
Chemin absolu ou relatif



Dans la barre d'adresse de votre navigateur web vous trouvez, quand vous visitez un site, des choses du genre : <https://upload.wikimedia.org/wikipedia/commons/0/05/Url.png>

- La partie `https://` indique le protocole de communication utilisé pour recevoir la ressource.
- La partie `upload.wikimedia.org` contient le nom de domaine du site (wikimedia), le sous-domaine (upload) et l'extension (.org).
- La partie `/wikipedia/commons/0/05/Url.png` est une URL (Uniform Resource Locator). C'est ce qui permet d'identifier une ressource (par exemple un fichier html ou autre, ici un fichier image .png) sur un réseau.

L'URL indique l'endroit où se trouve une ressource sur un ordinateur. Un fichier peut se trouver dans un dossier qui peut lui-même se trouver dans un autre dossier, etc. On parle d'une structure en arborescence, car elle ressemble à un arbre (à l'envers si on met la racine en haut) :



Il existe deux types de chemins (on dit aussi adresses) dans une arborescence : **absolu** et **relatif**.

1. Chemin absolu

Le chemin absolu doit indiquer le chemin depuis la racine.

Par exemple l'URL du fichier `fichier3.jpg` est : `/dossier2/dossier3/fichier3.jpg`

Remarquez que nous démarrons bien de la racine `/`.



Les symboles de séparation sont aussi des `/`.

2. Chemin relatif

Imaginons maintenant que le fichier `fichier1.css` fasse appel au fichier `fichier3.jpg` (comme un fichier HTML peut faire appel à un fichier CSS). Il est possible d'indiquer le chemin non pas depuis la racine, mais depuis le dossier (`dossier2`) qui accueille le fichier `fichier1.css`. Nous parlerons alors de chemin relatif.

Dans notre cas, le chemin relatif est le suivant : `dossier3/fichier3.jpg`

Car depuis le `dossier2`, il faut aller dans le `dossier3`, dans lequel se trouve le fichier voulu.

Remarquez l'absence du `/` au début du chemin : c'est ce qui distingue les adresses relatives et absolues.

Imaginons maintenant que nous désirions indiquer le chemin relatif du fichier `fichier1.css` depuis l'intérieur du dossier `dossier1`.

Comment faire ?

Il faut « remonter » d'un niveau dans l'arborescence pour pouvoir repartir vers la bonne « branche ».

Pour ce faire on utilise une notation avec 2 points : `..`

Le chemin relatif est alors le suivant : `../dossier2/fichier1.css`

Il est tout à fait possible de remonter de plusieurs niveau :

`../../` depuis le dossier `dossier4` permet de remonter jusqu'à la racine.

3. En résumé

Chemin absolu :

- débute par /
- indique l'adresse depuis la racine

Chemin relatif :

- ne débute pas par /
- indique l'adresse depuis la position actuelle (autrement dit depuis le répertoire contenant le fichier dans lequel est donnée l'adresse vers le fichier vers lequel il pointe)
- `../` signifie un niveau plus haut dans l'arborescence

4. Exercice

On considère encore la structure en arborescence donnée plus haut.

1. Donner le chemin absolu du fichier `fichier5.svg`.
2. Le fichier `fichier1.css` contient une adresse relative pointant vers le fichier `fichier5.svg`.
Quelle est cette adresse ?
3. Encore dans le fichier `fichier1.css` se trouve une adresse relative pointant vers le fichier `fichier2.txt`.
Quelle est cette adresse ?
4. Le fichier `fichier6.html` contient une adresse relative pointant vers le fichier `fichier1.css`.
Quelle est cette adresse ?
5. Encore dans le fichier `fichier6.html` se trouve une adresse relative pointant vers le fichier `fichier4.js`.
Quelle est cette adresse ?
6. Encore dans le fichier `fichier6.html` se trouve une adresse relative pointant vers le fichier `fichier7.odp`.
Quelle est cette adresse ?