

## COURS

## Le Web

## 1 Introduction

Le "World Wide Web", plus communément appelé "Web" a été développé au CERN (Conseil Européen pour la Recherche Nucléaire) par le Britannique Sir Timothy John Berners-Lee et le Belge Robert Cailliau au début des années 90. Pour faciliter les échanges d'information Tim Berners-Lee met au point le système **hypertexte**. Le système hypertexte permet, à partir d'un document, de consulter d'autres documents en cliquant sur des mots clés. Ces mots "cliquables" sont appelés hyperliens et sont souvent soulignés et en bleu.

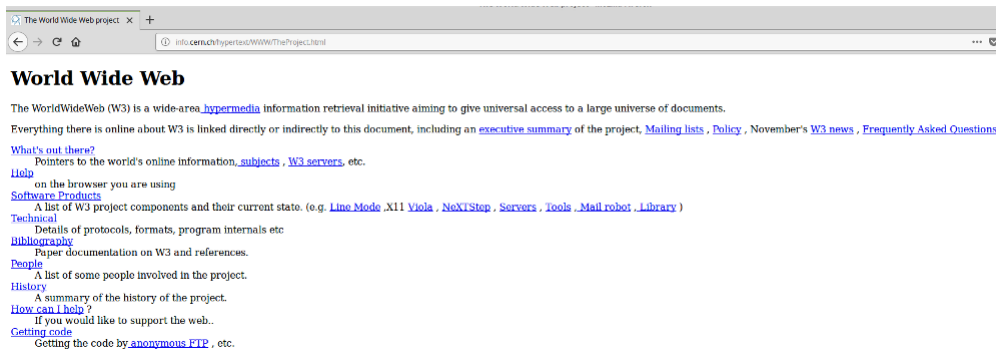


FIGURE 1 – première page web, les hyperliens sont soulignés et en bleu

Cette première page web est toujours consultable à l'adresse suivante :  
<http://info.cern.ch/hypertext/WWW/TheProject.html>

### À savoir

Le web se base sur trois choses :

- le **protocole HTTP** (HyperText Transfert Protocol) ;
- les **URL** (Uniform Resource Locator)
- le langage de description **HTML** (HyperText Markup Language) associé au langage de feuille de style CSS (Cascading Style Sheets) pour la mise en forme (ce dernier n'est pas obligatoire)

### Ne pas confondre

Beaucoup de personnes confondent "web" et "internet". Même si le "web" "s'appuie" sur internet, les deux choses n'ont rien à voir puisqu'"internet" est un "réseau de réseaux" s'appuyant sur le protocole IP (voir le module Internet) alors que, comme nous venons de le voir, le web est la combinaison de trois technologies : HTTP, URL et HTML. D'ailleurs on trouve autre chose que le "web" sur internet, par exemple, les emails avec le protocole SMTP (Simple Mail Transfert Protocol) et les transferts de fichiers avec le protocole FTP (File Transfert Protocol).

## 2 Les URLs

Dans la barre d'adresse de votre navigateur web vous trouverez, quand vous visitez un site, des choses du genre : "http://www.ac-rennes.fr/disciplines/informatiquelycee/index.html". Nous aurons l'occasion de parler du "www.ac-rennes.fr" plus tard. La partie "/disciplines/informatiquelycee/index.html" s'appelle une URL.

Une URL (Uniform Resource Locator) permet d'identifier une ressource (par exemple un fichier) 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... On parle d'une structure en arborescence, car elle ressemble à un arbre à l'envers.

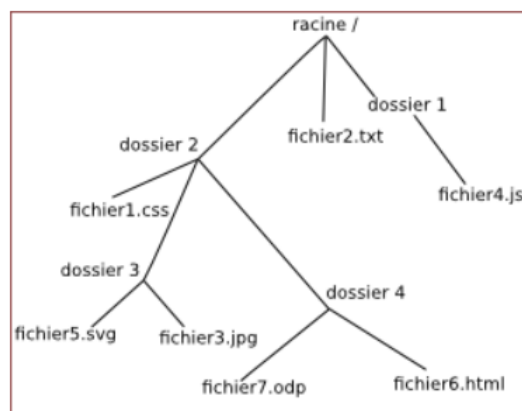


FIGURE 2 – structure en arborescence

Comme vous pouvez le constater, la base de l'arbre s'appelle **la racine** de l'arborescence et se représente par un /.

Autre représentation un peu plus "jolie" :

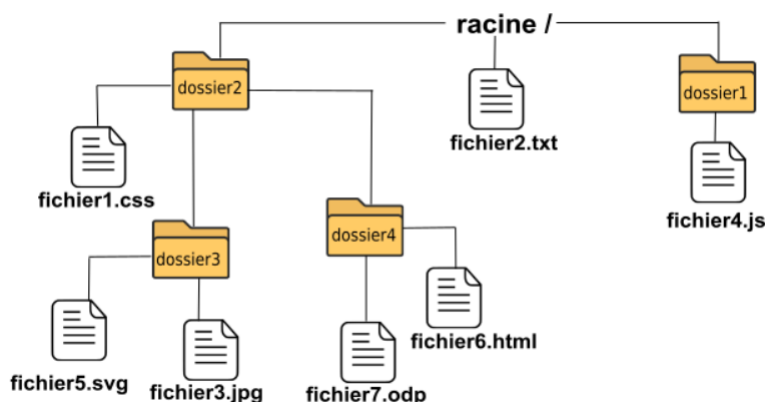


FIGURE 3 – structure en arborescence

**Chemin absolu ou chemin relatif?** Pour indiquer la position d'un fichier (ou d'un dossier) dans l'arborescence, il existe 2 méthodes : indiquer un chemin absolu ou indiquer un chemin relatif. Le chemin absolu doit indiquer « le chemin » depuis la racine. Par exemple l'URL du fichier fichier3.jpg sera : /dossier2/dossier3/fichier3.jpg

Remarquez que nous démarrons bien de la racine / (attention les symboles de séparation sont aussi des /)

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 fichier1.css, nous parlerons alors de chemin relatif : dossier3/fichier3.jpg

Remarquez l'absence du / au début du chemin (c'est cela qui nous permettra de distinguer un chemin relatif et un chemin absolu).

Imaginons maintenant que nous désirions indiquer le chemin relatif du fichier fichier5.svg depuis l'intérieur du dossier dossier4.

### Comment faire ?

Il faut "remonter" d'un "niveau" dans l'arborescence pour se retrouver dans le dossier dossier2 et ainsi pouvoir repartir vers la bonne "branche" (vers le dossier3). Pour ce faire il faut utiliser 2 points : ..

```
../dossier3/fichier5.svg
```

Il est tout à fait possible de remonter de plusieurs "crans" : ../../ depuis le dossier dossier4 permet de "retourner" à la racine.

#### À faire vous-même 1

Soit la structure en arborescence suivante :

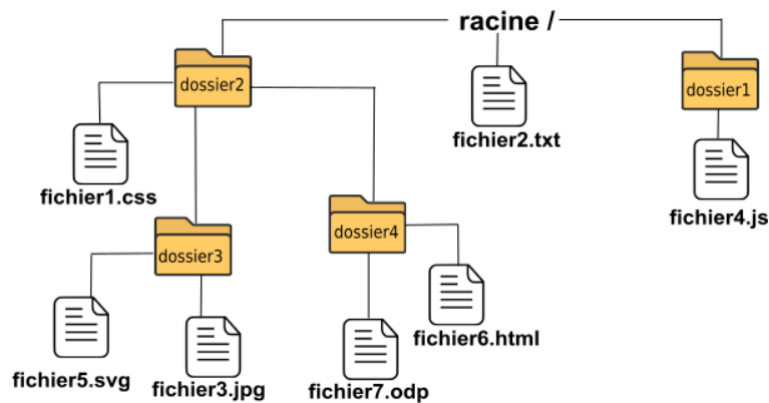


FIGURE 4 – Exemple de structure arborescente

- **Donnez** le chemin relatif permettant d'atteindre le fichier "fichier5.svg" depuis le dossier "dossier4".
- **Donnez** le chemin absolu permettant d'atteindre le fichier "fichier6.html".

#### Remarque importante

La façon d'écrire les chemins (avec des slash (/) comme séparateurs) est propre aux systèmes dits « UNIX », par exemple GNU/Linux ou encore Mac OS. Sous Windows, ce n'est pas le slash qui est utilisé, mais l'antislash (\). Pour ce qui nous concerne ici, les chemins réseau (et donc le web), pas de problème, c'est le slash qui est utilisé.

## À faire vous-même 2

À partir du dossier intitulé "chemin.zip" à [télécharger ici](#).

1. **le sauvegarder** dans votre répertoire "Homework".
2. **décompresser** le fichier.
3. **réaliser** ci-dessous l'arborescence du dossier "chemin".

## À faire vous-même 3

L'objectif est d'ouvrir le fichier "Fichier1.txt" en double cliquant sur le "ouvrirfichier1". Cependant sur un système d'exploitation (ici Windows en salle de SNT) il faut utiliser des fichiers exécutables **.bat** par exemple.

1. **Ouvrir** le Fichier1.txt en double cliquant, Bloc Note s'ouvre et vous laisse voir le message noté, **faire** de même avec ouvrirFichier1.txt, bien remarquer le code noté de ce fichier il va vous servir pour la suite..
2. **Faire** un copier-coller de ce fichier et le renommer "Fichier1.bat" <sup>a</sup> (modification de l'extension, Windows vous demande si vous êtes certains, répondre oui).
3. **Double cliquer** sur le ouvrirFichier1.bat et constater l'ouverture du fichier "Fichier1.txt".

---

a. l'extension "bat" rend ainsi le fichier exécutable contrairement à l'extention "txt"

## À faire vous-même 4

L'objectif est à présent d'ouvrir différents fichiers (Fichier1, Fichier2, Fichier3, Fichier4 et Fichier5) à partir des fichiers présents dans l'arborescence ("ouvrirFichier1", "ouvrirFichier2", "ouvrirFichier3", "ouvrirFichier4" et "ouvrirFichier5" potentiellement présents à différents endroits ). N'oubliez pas la remarque importante page 3 concernant l'écriture des chemins sous Windows, différents des autres systèmes d'exploitation et des URLS .

C'est à vous !

## 3 HTML et CSS

### 3.1 Initiation

Nous allons nous intéresser à un acteur fondamental du développement web, le couple HTML+CSS (Hyper Text Markup Language et Cascading Style Sheets).

Dans un premier temps, nous allons exclusivement nous intéresser au HTML. Qu'est-ce que le HTML, voici la définition que nous en donne Wikipedia :

L'Hypertext Markup Language, généralement abrégé HTML, est le format de données conçu pour représenter les pages web. C'est un langage de balisage. HTML permet également de structurer sémantiquement et de mettre en forme le contenu des pages, d'inclure des ressources multimédias, dont des images, des formulaires de saisie, et des programmes informatiques. Il est souvent utilisé conjointement avec des langages de programmation (JavaScript) et des formats de présentation CSS (feuilles de style en cascade).

Pour l'instant, nous allons retenir deux éléments de cette définition «conçu pour représenter les pages web» et «un langage de balisage».

Grâce au HTML vous allez pouvoir, dans votre navigateur (Firefox, Chrome, Opera,...), afficher du texte, afficher des images, proposer des hyperliens (liens vers d'autres pages web), afficher des formulaires et même maintenant afficher des vidéos.

Pour aborder le HTML, nous allons, dans un premier temps utiliser le site [jsfiddle.net](https://jsfiddle.net).

Après avoir lancé votre navigateur web, tapez `http://jsfiddle.net/` dans la barre d'adresse.

Vous devriez voir apparaître ceci :

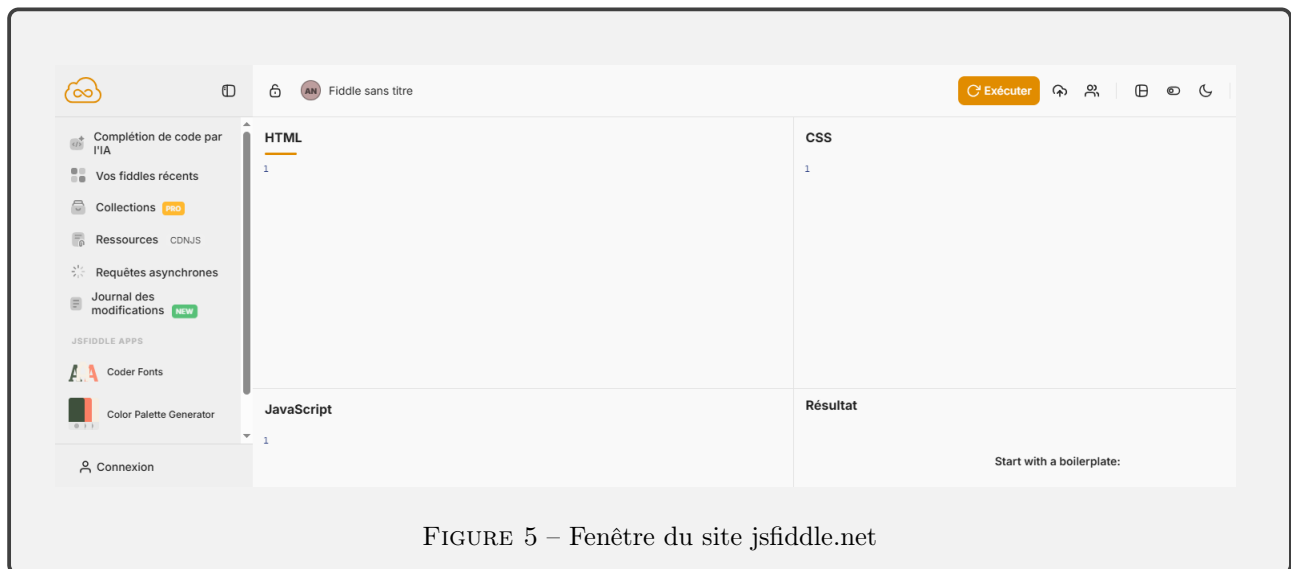


FIGURE 5 – Fenêtre du site jsfiddle.net

Nous allons pour l'instant uniquement utiliser la fenêtre « HTML » et la fenêtre « Resultat »

## À faire vous-même 5

Écrivez le code HTML suivant :

```
<h1>Hello World! Ceci est un titre</h1>
<p>Ceci est un <strong>paragraphe</strong>. Avez-vous bien compris ?</p>
```

**Qu'est-ce qui s'affiche dans la fenêtre ?**

Comme déjà évoqué ci-dessus, en HTML tout est une histoire de balise que l'on ouvre et que l'on ferme. Une balise ouvrante est de la forme `<nom_de_la_balise>`, les balises fermantes sont de la forme `</nom_de_la_balise>`.

En observant attentivement le code, vous devriez forcément remarquer que toute balise ouverte doit être refermée à un moment ou un autre. La balise ouvrante et la balise fermante peuvent être sur la même ligne ou pas, cela n'a aucune espèce d'importance, la seule question à se poser ici est : ai-je bien refermé toutes les balises que j'ai ouvertes ?

Enfin pour terminer avec les généralités sur les balises, il est important de savoir qu'une structure du type :

```
<balise1>
<balise2>
</balise1>
</balise2>
```

est interdite, la balise2 a été ouverte après la balise1, elle devra donc être refermée avant la balise1.

En revanche, l'enchaînement suivant est correct :

```
<balise1>
  <balise2>
  </balise2>
</balise1>
```

Notez que dans notre exemple nous respectons bien cette règle « d'imbrication » des balises avec la balise `<p>` et la balise `<strong>`.

Il est important de comprendre que chaque balise a une signification qu'il faut bien respecter (on parle de la sémantique des balises). Par exemple le texte situé entre la balise ouvrante et fermante `<h1>` est obligatoirement un titre important (il existe des balises `<h2>`, `<h3>` ... qui sont aussi des titres, mais des titres moins importants (sous-titre)). La balise `<p>` permet de définir des paragraphes, enfin, la balise `<strong>` permet de mettre en évidence un élément important.

Vous devez aussi savoir qu'il existe des balises qui sont à la fois ouvrantes et fermantes (`<balise/>`) : un exemple, la balise permettant de sauter une ligne, la balise `<br/>`.

Il est possible d'ajouter des éléments à une balise ouvrante, on parle d'attribut. Une balise peut contenir plusieurs attributs :

```
<ma_balise attribut_1= "valeur_1" attribut_2="valeur_2">
```

Il existe beaucoup d'attributs différents, nous allons nous contenter de 2 exemples avec l'attribut *id* (id pour identifiant) et *class*. Nous verrons l'intérêt de ces attributs dans l'activité suivante.

## À faire vous-même 6

Écrivez le code HTML suivant et **exécuter** :

```
<h1>Ceci est un titre</h1>
<h2 class="titre_1">Ceci est un sous titre</h2>
<p id="para_1">Ceci est un <strong>paragraphe</strong>. Avez-vous bien compris ?</p>
```

**Excécuter** une première fois puis **retirer** les attributs et **ré-exécuter**. Y a t'il une différence de rendu ?

Le HTML n'a pas été conçu pour gérer la mise en page (c'est possible, mais c'est une mauvaise pratique). Le HTML s'occupe uniquement du contenu et de la sémantique, pour tout ce qui concerne la mise en page et l'aspect « décoratif » (on parle du « style » de la page), on utilisera le **CSS** (Cascading Style Sheets).

Dans **JSFIDDLE**, il est possible d'écrire du **CSS** dans la fenêtre en haut à droite.

## À faire vous-même 7

**Ajouter** le code CSS suivant dans l'espace dédié au CSS et **exécuter** :

```
h1
{
    text-align: center;
    background-color: red;
}
h2
{
    font-family: Verdana;
    font-style: italic;
    color: green;
}
```

**Qu'est-ce qui s'affiche dans la fenêtre ?**

Dans l'exemple du "À faire vous-même 3", les propriétés « text-align » et « background-color » seront appliquées au contenu de toutes les balises de type h1 (avec respectivement les valeurs « center » et « red »).

Vous trouverez à l'adresse suivante <https://www.w3schools.com/> dans la partie HTML la référence du langage HTML. **Amusez-vous** à modifier l'apparence de votre page web à partir de information fournies dans **HTML color**

## À faire vous-même 8

Écrivez le code HTML suivant :

```
<h1>Ceci est un titre</h1>
<h2>Ceci est un sous titre</h2>
<p id="para_1">Ceci est un <strong>paragraphe</strong>. Avez-vous bien compris ?</p>
```

Écrivez le code CSS suivant :

```
#para_1
{
    font-style: italic;
    color: green;
}
```

**Qu'est-ce qui s'affiche dans la fenêtre ? Que remarquez-vous ?**

Il est donc possible de cibler un paragraphe et pas un autre en utilisant l'*id* du paragraphe (en CSS l'*id* se traduisant par le signe #).

Il est aussi possible d'utiliser l'attribut class à la place de l'id. Dans le CSS on utilisera le point . à la place du #.

## La différence entre "id" et "class" n'est pas très importante.

L'attribut "class" permet de donner le même nom à plusieurs reprises dans une même page.

Si nous avons eu un 3e paragraphe, nous aurions pu avoir : `<p class="para_1">Voici un 3ème paragraphe</p>`, mais nous n'aurions pas pu avoir : `<p id="para_1"> Voici un 3e paragraphe </p>`, car le nom `para_1` a déjà été utilisé pour le 1er paragraphe.

## 3.2 Plus réaliste...

### Remarque importante

JSFIDDLE est un très bel outil, mais il ne peut pas être utilisé pour la réalisation d'un vrai site internet (ou d'une vraie application web).

Nous allons à partir de maintenant apprendre à créer une page web comme il se doit. Nous allons ainsi créer 2 fichiers : un fichier qui contiendra du HTML (`index.html`) et un fichier qui contiendra du CSS (`style.css`).

### À faire vous-même 9

À l'aide d'un éditeur de texte (bloc note par exemple), **créer** un nouveau fichier.

**Sauvegardez-le** en précisant son nom, par exemple `"index.html"`.

**Écrivez** le code suivant dans votre éditeur de texte (sans oublier de sauvegarder quand vous avez terminé) :

```
<!doctype html>
<html lang="fr">
  <head>
    <meta charset="utf-8">
    <title>Voici mon site</title>
  </head>
  <body>
    <h1>Hello World! Ceci est un titre</h1>
    <p>Ceci est un <strong>paragraphe</strong>. Avez-vous bien compris ?</p>
  </body>
</html>
```

**Testez** votre code à l'aide d'un navigateur web (Firefox, Chrome, Edge, etc) en "double-cliquant" sur le fichier `index.html`

Dans l'exemple du "À faire vous-même 5", vous reconnaissez le code se trouvant entre les balises `<body>` :

```
<body> .....</body>
```

Tout votre code HTML devra se trouver entre ces 2 balises.

Le reste des balises devraient vous être inconnues. Passons-les en revue :

La première ligne (`<!doctype html>`) permet d'indiquer au navigateur que nous utiliserons la dernière version du HTML, le fameux HTML5.

La balise `<html>` est obligatoire, l'attribut `lang="fr"` permet d'indiquer au navigateur que nous utiliserons le français pour écrire notre page.

Les balises `<head>...</head>` délimitent ce que l'on appelle l'en-tête. L'en-tête contient, dans notre exemple, 2 balises : la balise `<meta charset="utf-8">` qui permet de définir l'encodage des caractères (utf-8) et la balise `<title>` qui définit le titre de la page (attention ce titre ne s'affiche pas dans le navigateur, ne pas confondre avec la balise `<h1>`).

## À faire vous-même 10

À l'aide d'un éditeur de texte, **créer** un nouveau fichier.

Toujours à l'aide d'un éditeur de texte, **créer** un fichier qui va contenir le CSS de notre page (par exemple style.css). Complétez ce fichier à l'aide du code suivant :

**Écrivez** le code suivant dans votre éditeur de texte (sans oublier de sauvegarder quand vous avez terminé) :

```
h1
{
    text-align: center;
    background-color: red;
}
p
{
    font-family: Verdana;
    font-style: italic;
    color: green;
}
```

Pour l'instant notre CSS ne sera pas appliqué à notre page, pour ce faire, il faut modifier notre code HTML en ajoutant une ligne qui va permettre d'associer notre code CSS à notre page.

**Modifiez** le code HTML avec la ligne suivante `<link rel="stylesheet" href="style.css">` :

```
<!doctype html>
<html lang="fr">
  <head>
    <meta charset="utf-8">
    <title>Voici mon site</title>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <h1>Hello World! Ceci est un titre</h1>
    <p>Ceci est un <strong>paragraphe</strong>. Avez-vous bien compris ?</p>
  </body>
</html>
```

Vous trouverez à l'adresse suivante <https://www.w3schools.com/> dans la partie CSS la référence du langage CSS.

**Changer** la couleur du texte et **Amusez-vous** à modifier l'apparence de votre page web à partir de l'information fournies dans **CSS color** et **Testez** votre code à l'aide d'un navigateur web en "double-cliquant" sur le fichier index.html

Dans l'exemple que nous venons de voir, les fichiers "index.html" et "style.css" se trouvent dans le même dossier. Il est souvent utile de placer les fichiers CSS dans un dossier "CSS". Il faudra alors modifier le chemin du fichier style.css dans le code HTML afin que le fichier index.html trouve le fichier style.css.

Indiquer ici le nouveau chemin

Attention pour les adresses URLs, ce sont les "slashes" / qui sont utilisés.

### 3.3 Quelques balises utiles

Pour terminer, voici quelques balises très utilisées :

### 3.4 La balise <a>

```
<a href="mon_autre_page.html">Cliquez ici pour vous rendre sur mon autre page</a>
```

La balise **a** permet de créer des liens hypertextes, ce sont ces liens hypertextes qui vous permettent de "voyager" entre les pages d'un site ou entre les sites. Les liens hypertextes sont par défaut soulignés et de couleur bleue (modifiable grâce au CSS). La balise <a> possède un attribut *href* qui a pour valeur le chemin du fichier que l'on cherche à atteindre ou l'adresse du site cible (exemple : <a href="http://www.google.fr"> Cliquez ici pour vous rendre sur google.fr</a>). Entre la balise ouvrante et fermante, on trouve le texte qui s'affichera à l'écran (c'est ce texte qui est souligné et de couleur bleue). La balise peut sans problème se trouver en plein milieu d'un paragraphe.

#### À faire vous-même 11

**Créer** un fichier par exemple *essai\_baliseA.html* et **copier** <a href="http://www.google.fr"> Cliquez ici pour vous rendre sur google.fr</a> entre les balises <body> et </body>. **Essayer** le code en double cliquant sur *essai\_baliseA.html*.

**Créer** sur cette même page *essai\_baliseA.html* ou dans un autre fichier un lien vous permettant d'accéder à un de vos sites préférés.

#### 3.4.1 La balise <img>

Comme vous devez déjà vous en douter, la balise image sert à insérer des.....images :

```

```

la balise <img> est à la fois ouvrante et fermante comme la balise <br>. Elle possède 2 attributs :

- **src** qui doit être égal au nom du fichier image (ou au chemin si le fichier image se trouve dans un autre dossier).
- **alt** qui doit être égal à une description de votre image (cet attribut est utilisé notamment par les systèmes de description des pages web utilisées par les non-voyants), il faut donc systématiquement renseigner cet attribut.

#### À faire vous-même 12

**Créer** un fichier par exemple *essai\_baliseImgA.html* et **insérer** le code suivant "" entre les balises <body> et </body>, **Essayer** le code en double cliquant sur *essai\_baliseImg.html* et **constater** qu'aucune image n'apparaît mais seulement "avion" donné à l'attribut *alt*.

**Télécharger** une image d'avion et l'**enregistrer** dans un dossier **img**. **Modifier** le fichier *essai\_baliseImgA.html* en conséquence pour qu'il puisse trouver l'image de l'avion. **Essayer** le code en double cliquant sur *essai\_baliseImg.html*.

### 3.4.2 Les balises <form>, <input> et <button>

Les formulaires sont des éléments importants des sites internet, ils permettent à l'utilisateur de transmettre des informations. Un formulaire devra être délimité par une balise <form> (même si ce n'est pas une obligation) :

```
<form>
....
</form>
```

Il existe différentes balises permettant de construire un formulaire, notamment la balise <input>. Cette balise possède un attribut type qui lui permet de jouer des rôles très différents.

La balise <button> nous sera aussi d'une grande utilité.

#### À faire vous-même 13

Créez un fichier html (par exemple *essai\_baliseForm.html* contenant le code suivant :

```
<!doctype html>
<html lang="fr">
  <head>
    <meta charset="utf-8">
    <title>Voici mon site</title>
  </head>
  <body>
    <form>
      <p>voici un champ de texte : <input type="text"/></p>
      <p>voici une checkbox <input type="checkbox"/></p>
      <button>Cliquez ici !</button>
    </form>
  </body>
</html>
```

**Testez** votre code à l'aide d'un navigateur web en "double-cliquant" sur le fichier html que vous venez de créer.

Je vous invite à tester quelques attributs différents de la balise <input> que vous trouverez à l'adresse suivante <https://www.w3schools.com/> dans la partie HTML Tags et en particulier le type "password" et **saisir** le mot de passe, enfin le type "email" pour lequel ne pas hésiter à entrer une adresse mail sans le @ et **constater**. **Libre à vous** d'essayer ce qui vous est proposé sur le site .

### 3.4.3 Les balises div et span

Ces 2 balises génériques sont très utilisées (surtout la balise div).

#### À quoi servent-elles ?

À organiser la page, à regrouper plusieurs balises dans une même entité.

#### Pourquoi 2 balises génériques ?

Parce que **div** est une balise de type "**block**" et que **span** est une balise de type "**inline**" !

Sans vouloir trop entrer dans les détails, il faut bien comprendre que l'ordre des balises dans le code HTML a une grande importance. Les balises sont affichées les unes après les autres, on parle du flux normal de la page.

C'est ici qu'entrent en jeu les balises de type "block" et les balises de type "inline".

**À retenir sur les balises Block et Inline**

Les contenus des balises de type "block" (p, div, h1, ...) se placent sur la page web les uns en dessous des autres. Les contenus des balises de type "inline" (strong, img, span, a) se placent sur la page web les uns à côté des autres. Cela revient à dire qu'une balise de type "block" prend toute la largeur de la page alors qu'une balise de type "inline" prend juste la largeur qui lui est nécessaire.

**À faire vous-même 14**

**Créez** un fichier html (par exemple *essai\_baliseDiv.html* contenant le code suivant :

```
<!doctype html>
<html lang="fr">
  <head>
    <meta charset="utf-8">
    <title>Voici mon site</title>
  </head>
  <body>
    <div>div est une balise de type "block"</div>
    <p>la balise p est une autre balise de type block</p>
    <span>En revanche, span est une balise de type "inline"</span>
    <a href="www.google.fr">Et voici une autre balise de type "inline"</a>
    <h1>h1 est bien une balise de type "block"</h1>
    <span>la malheureuse balise span est "obligée" de se placer en dessous</span>
  </body>
</html>
```

**Testez** votre code à l'aide d'un navigateur web en "double-cliquant" sur le fichier html que vous venez de créer.

**À faire vous-même 15**

**Élaborez** une page HTML en utilisant les balises vues ci-dessus.

## 4 Les cookies

À faire vous-même ...

Après avoir regardé les 2 vidéos ci-dessous (ne pas hésiter pas à prendre des notes pendant le visionnage des vidéos), **rédigé** un texte de quelques lignes qui résumera les thèmes abordés dans ces vidéos.



À faire vous-même 17

À l'aide des informations présentes sur le site de la CNIL et dans la vidéo ci-dessous, paramétrez votre navigateur web afin de limiter vos traces lors de vos navigations sur le web.

