

1 Structuration d'une page Web

Le **World Wide Web** constitue l'ensemble des applications utilisables sur le réseau **Internet** (ne pas confondre les deux). Le Web, développé dans les années 1980 par Tim Berners-Lee et Robert Cailliau, est à l'origine principalement une collection de documents consultables à l'aide d'un logiciel **navigateur**.

Le navigateur envoie une **requête** avec le protocole **HTTP** sur le réseau Internet pour interroger un **serveur** qui renvoie alors le document souhaité qui peut s'afficher sur l'ordinateur **client**.

Aujourd'hui on peut construire des applications Web complètes qui permettent de gérer une interaction avec l'utilisateur.

La construction d'une page Web repose sur 3 piliers :

1. Un document écrit en **langage HTML** permettant de structurer le contenu du document (titre, paragraphes, sections...)
2. Une **feuille de style CSS** permettant de définir la mise en page du document (couleurs, police, mise en page...)
3. Des **programmes permettant de dynamiser la page** pour modifier son contenu ou son apparence lors de l'interaction avec l'utilisateur (exécutés côté client pour une application Web écrite en JavaScript par exemple ; ou côté serveur, écrits en PHP ou Python, par exemple).

Ce 3ème pilier n'est pas obligatoire et la création de pages Web statiques est encore très commune.

2 Construction d'un fichier HTML

Le langage HTML (Hypertext Markup Language) est un **langage de description** qui repose sur des **balises**. La version actuelle (depuis 2014) est HTML 5.

Une **balise** permet de définir un **élément** du document HTML, un élément pouvant être un titre, un paragraphe, un retour à la ligne...

La structure d'une balise est la suivante :

```
<nom attribut1="valeur" attribut2="valeur">contenu de l'élément</nom>
```

Un élément est donc défini par une **balise ouvrante**, un **contenu** et une **balise fermante**.

Les balises peuvent être précisées à l'aide d'attributs (facultatifs ou non) pour lesquels on définit une valeur.

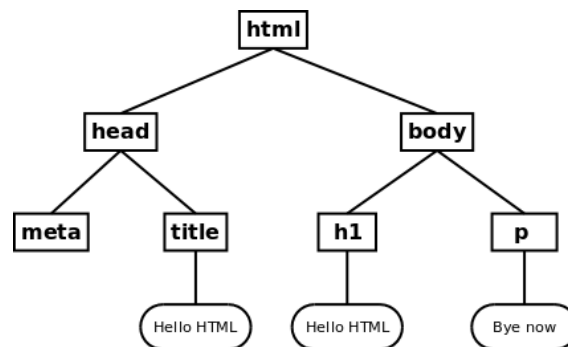
Un élément peut contenir d'autres éléments à l'intérieur (comme un jeu de poupées russes).

La structure minimale d'un document HTML est la suivante :

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta charset="UTF-8" />
    <title>Hello HTML</title>
  </head>
  <body>
    <h1>Hello HTML</h1>
    <p>Bye now</p>
  </body>
</html>
```

L'**arbre DOM** (Document Object Model) est la façon dont le navigateur « voit » le document HTML.

Avec l'exemple de la page minimale, l'arbre DOM est :



Un fichier HTML se développe ainsi toujours sur la structure suivante :

1. un prologue définissant la **DOCTYPE** (avertit le navigateur que le langage utilisé est HTML5).
2. l'élément racine **html** (conteneur de l'ensemble du document).
3. un élément d'en-tête **head** (définit par exemple le titre de la page et l'encodage des caractères).
4. un élément de corps **body** qui présente le contenu qui s'affiche dans le navigateur.

Le balisage à l'intérieur du corps repose aujourd'hui en HTML5 sur plusieurs éléments sémantiques permettant de structurer le contenu selon le sens des parties qu'il présente (header, main, article, section, aside, nav, footer).

Il reste toujours possible de cibler des portions de document avec la balise **div**, de type **block**, ou la balise **span**, de type **inline** (qui restent des balises génériques de sectionnement).

3 Feuille de style CSS

Une feuille de style (Cascading Style Sheet) permet la mise en forme de chaque élément du document HTML.

Le langage CSS apparu en 1996 est aujourd'hui utilisé en version 3.

Un élément peut être stylé par l'attribut *style* de sa balise, ou à l'intérieur d'une *balise style* indépendante. Mais la meilleure façon de procéder est de définir les **règles de style** dans un fichier annexe (*style.css*) qui est lié au fichier HTML par une balise **link** dans son en-tête.

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

Les règles de style ont la structure suivante :

```
selecteur {  
  propriété1 : valeur;  
  propriété2 : valeur;  
}
```

Quelques exemples de sélecteurs :

1. sélecteurs d'un élément :

```
p {  
  font-size : 1.2em;  
}
```

Avec le nom d'un élément, on sélectionne tous les éléments identiques du document (ici les paragraphes).

2. groupe de sélecteurs :

```
h1, h2, h3 {  
  text-decoration : underline;  
}
```

Avec la virgule, on sélectionne en même temps plusieurs éléments (ici les titres de niveaux 1, 2 et 3).

3. sélecteurs de classe :

```
.italique {  
  font-style : italic;  
}
```

Avec le point, on sélectionne les éléments d'une certaine classe (ici les éléments qui ont la valeur *italique* pour leur attribut **class**).

4. sélecteurs d'identifiant :

```
#titreSpecial {  
  color : rgb(12,45,127);  
}
```

Avec le dièse #, on sélectionne l'unique élément de l'identifiant (ici, l'**unique** élément qui a la valeur *titreSpecial* pour son attribut **id**).

5. combinaison de sélecteurs :

```
h2 strong{  
  color : red;  
}
```

Avec l'espace (!!!), on sélectionne seulement les éléments enfants d'un autre (ici, les éléments *strong* qui sont à l'intérieur d'un élément *h2*).

4 Interaction homme-machine (IHM) : application Web et langage JavaScript

L'interaction homme-machine dans une application Web repose sur des composants graphiques de la page Web permettant de réagir à des événements.

Les composants graphiques peuvent être :

- des éléments du document : titre, paragraphe, lien hypertexte...
- des éléments particuliers liés à un formulaire par exemple : champ de saisie, bouton, case à cocher...
- des entités du navigateur : la page, le navigateur, l'écran

Les événements peuvent être :

- un passage de la souris sur un élément
- un clic de souris (gauche, droit, molette)
- une modification de la taille de la fenêtre
- ...

L'écriture d'une l'application Web consiste alors à l'aide d'un langage de programmation (JavaScript en général) à associer des fonctions, permettant d'effectuer les tâches désirées, qui sont appelées lors de la réalisation des événements possibles.

Ceci est réalisé en associant un **gestionnaire d'événement** aux éléments à contrôler.

À partir d'exemples vus en TP, nous pourrons observer des points communs et différences entre le langage Python ou JavaScript.