

Concepts de base

Ressources nécessaires pour un site Web

- **Fichiers** : à disposition des visiteurs dans un format "standard".
- **Serveur** : Machine physique accessible via internet où héberger les fichiers. *serveur* = *machine physique* + *logiciel*. Logiciel-serveur capable de répondre aux demandes de fichiers.

Fonctionnement de base

1. Le visiteur indique une **URL**.
2. Le navigateur envoie une **demande** au serveur situé à l'adresse.
3. Le serveur reçoit la demande et **répond** en renvoyant les fichiers demandés.
4. Le navigateur les reçoit et les **interprète**.

point 2. et 3. -> *protocole HTTP*

Les standards

- **HTML** : description du contenu des pages
 - **CSS** : description du formatage des pages
- Les standards sont établis par le W3C (*World Wide Web Consortium*)

Les principes

- Les **blancs** répétés sont ignorés.
- On peut insérer des commentaires `<!-- commentaire -->`.
- On écrit les balises en minuscules.

Définitions

Navigateur Logiciel capable d'envoyer des demandes de documents via internet. De réceptionner les informations renvoyées et d'interpréter les standards utilisés pour le web et les transformer en un document affichable.

Client Personne qui visite un site web.

Serveur La machine qui héberge un site web, ou le logiciel qui est à l'écoute des demandes (*Apache*).

Protocole Ensemble de règles, conventions, qui permettent de mettre en oeuvre une communication.

FTP *File Transfer Protocol* : protocole permettant l'échange de fichiers.

HTTP *HyperText Transfer Protocol* : protocole permettant d'échanger des hypertextes.

Définitions (cont)

Hyper texte Texte accompagné de liens (*hyperliens*) permettant de voyager d'une partie à l'autre.

HTML *HyperText Markup Language* : langage permettant d'annoter un hypertexte ; principal standard utilisé pour présenter des pages web.

URL *Unified Resource Locator* : Permet de localiser une ressource, exprimée dans un format standardisé. (en FR : adresse réticulaire/universelle).

Cache Endroit où un navigateur stocke une copie des éléments déjà téléchargés afin d'éviter de devoir les télécharger à nouveau.

URL

<http://www.monsite.be/vacances/images/plage.jpg>

(le fichier .html est dans vacances)

- **Protocole** : HTTP

- **Nom de domaine** : www.monsite.be

- **Adresse absolue de la ressource** : /vacances/images/plage.jpg

- URL RACCOURCIE :

/vacances/images/plage.jpg

- URL RELATIVE :

images/plage.jpg ou ../vacances/images/plage.jpg

Les balises HTML

`<html>...</html>` Tout le document

`< head >...</ head >` La partie information du document

`<body>...</body>` La partie contenu du document

Les balises de head

`< title >...</ title >` Titre du document

`< meta charset ="utf - 8"/>` Encodage du fichier

Les balises de body

<code><h1>...</h1></code> jusque <code><h6>...</h6></code>	Titres et sous-titres
<code><p>...</p></code>	Paragraphes
<code>...</code>	Portions de texte importantes
<code>...</code>	Portions de texte en emphase
<code>
</code>	Retour à la ligne forcé
<code><hr /></code>	Ligne horizontale
<code> ...</code>	Liste ordonnée
<code> ...</code>	Liste à points

`<table><caption>...</caption><tr><th>...</th></tr></table></table>`

colspan="x" et rowspan="x" **colspan** : nombre de colonnes recouvertes / **rowspan** : nombre de lignes recouvertes

`` Balise image (peut ajouter les attributs width= et height= pour fixer la taille)

`...` pour une ancre : href = "#nomAncre" / autre page : page.html#nomAncre

`` Permet de créer une ancre

`<audio src ="*source" controls ></ audio>` Marche aussi avec <video>. Autres attributs possible : autoplay et loop

Attributs généraux id="" / title="" / class, style

Balises div et span

`<div>` Permet de regrouper un ensemble d'éléments HTML dans un "groupe"

`` Pareil mais pour une portion de texte, groupe de mots

En entourant plusieurs paragraphes de balises <div> par exemple, cela permet de définir une règle CSS portant sur tous les paragraphes de la balise <div>.

Exemple:

```
<div class="intro">
<p>...</p> <p>...</p> <p>...</p>
</div>
```

```
.intro { font-style:italic}
```

Formulaires

La validation se déroule souvent en deux temps :

- **Validation locale (sur le client)** : vérifications rapides qui ne nécessitent pas de données -> pratique mais pas sécurisante !
- **Validation par le serveur** : vérifications plus complexes utilisant des bases de données (login et mdp, autorisations...)
- **Méthode GET** : Données incluses dans l'adresse web.
- **Méthode POST** : Données transmises au sein de la demande.

Principes du CSS

On peut utiliser la balise `<style>...</style>` dans le `<head>` de l'html

`<link href ="style.css" rel ="stylesheet " type="text/css " />` A placer dans le `<head>`

- Format **inline** : attribut "style="
 - Format **interne** : balise <style>
 - Format **externe** : balise <link>
- Inline et interne pour des tests plus ponctuels.

Les sélecteurs simples

de Balise nom de la balise. ex:h2 { color : blue }

d'identificateur #nomId. ex:#para1 { color : red }

de classe .nomClasse. ex:.important { color : red }

Différences entre identificateurs (id = "") et classes (class = "")

- Un identificateur doit cibler un élément **unique**
- Plusieurs éléments peuvent avoir la même classe
- Un élément peut avoir plusieurs classes

Combinaison

- Balise et classe : p.intro : les <p> de classe "intro"
- Balise et identificateur : p#para1 : le <p> d'id "para1"
- Regroupement : th, td : les th et td
- Universel : * : tout

Les sélecteurs d'arborescence

Arborescence

Sans restriction **em** : toutes les balises

Ancêtre **ul em** : toutes les balises descendant d'un

Fils **li > em** : toutes les balises fils (directs) d'un

Frère **strong ~ em** : tous les venant après un

Adjacent **strong + em** : tous les suivant (directement ou pas) un (uniquement le 1er)

#grandcadre tous les qui se trouvent dans l'élément d'identificateur "grandcadre"

table td.numerique tous les <td> de classe "numérique" qui se trouvent dans une <table>

p.intro + p tous les <p>aragraphes qui suivent (directement ou pas) un <p>aragraph de classe "intro"

header table em tous les qui se trouvent dans une <table> à l'intérieur de la section <header>

#cadre h2 + p tous les <p>aragraphes suivant immédiatement un titre <h2> à l'intérieur de l'élément d'identificateur "cadre"

h1 + p + p + p tous les 3e <p>aragraphes suivant un titre <h1>

Les sélecteurs d'attributs

But Exemple

Sans restriction **video** : toutes les vidéos

Présence ****video[loop]** : toutes les vidéos à jouer en boucle

Valeur **a[target="_blank"]** : les liens à ouvrir dans un nouvel onglet

Test valeur chaine de caracteres

Les sélecteurs d'attributs (cont)

Préfixe **a[href^="http"]** : Liens qui commencent par HTTP

Suffixe **a[href\$=".pdf"]** : Liens qui finissent par .pdf

Contient **a[href*=".be"]** : Liens qui contiennent .be

- On peut aussi utiliser les conditions sans préciser de balise :
[id^="prem"] : tous les éléments dont l'id commence par "prem"

- On peut combiner plusieurs conditions en les collant :
img[href^="http"][target="_blank"] : les liens vers une adresse http devant être ouverts dans un nouvel onglet

Pseudo-classes

a:hover Lien survolé par le curseur

a:visited Lien qui a déjà été visité

a:active Lien qui est activé (lors du clic)

nav a:visited Liens déjà visités dans <nav>

li:first-child li qui est le premier fils de son parent

li:last-child li dernier fils de son parent

li:nth-child(4) li qui est le 4e fils de son parent

li:nth-last-child(4) li 4e fils en partant de la fin

em:only-child les em qui n'ont aucun frère

em:first-of-type premier fils de type em

em:last-of-type dernier fils de type em

em:nth-of-type(4) 4e fils de type em

em:nth-last-of-type(4) 4e fils de type em depuis la fin

em:only-of-type les em qui n'ont aucun frère em

Négation **p:not(intro)** : paragraphe qui n'est pas de classe "intro"

Pseudo-classes (cont)

table a:not([target="_blank"]) : liens qui se trouvent dans une table, sauf ceux qui s'ouvrent dans un nouvel onglet.

Vide p:empty : paragraphes vides (aucun fils et aucun texte)

Dans les pseudo-classes du type "nth", on peut utiliser une formule simple (comme 2n ou 3n+3) au lieu d'une constante numérique ! Ou encore "odd" (impair) et "even" (pair).

Pseudo-Elements

p::first-letter la 1re lettre d'un paragraphe

p::first-line la 1re ligne d'un paragraphe

::before

::after

Les pseudo-éléments (notés avec ::) permettent de sélectionner des morceaux de contenu qui ne correspondent pas à des éléments HTML.

Layout

Tous les éléments sont affichés selon le même box model :

Modèle en 'oignon' à 3 couches avec au centre le contenu.

- Autour du contenu : le rembourrage (**padding**)

- Autour du padding : la bordure (**border**)

- Autour de la bordure : la marge (**margin**)

padding : Tpx Rpx Bpx Lpx

Toujours le même ordre : Top right bottom et left

en cas de 2 valeurs : 1ere pour Top/Bottom et 2eme pour left/right

