

DÉVELOPPEMENT WEB DYNAMIQUE

Fatou NGOM
Ingénieur en informatique

Ecole Supérieur Polytechnique
Département Génie Informatique

Introduction au développement web

Plan

1. Introduction
2. Evolution du web
3. Technologie du web

C'est quoi le web?

- Technologie informatique permettant de consulter des pages regroupés sur des sites
- Désigne une toile d'araignée
- Le World Wide Web désigne la toile d'araignée mondiale

Le Web : une application Internet

- Le World Wide Web, la toile d'araignée mondiale, est un réseau de ressources qui s'appuie sur :
 - une norme unique de nommage (URL) ;
 - un protocole d'échange (HTTP) ;
 - un système de liaison hypertexte entre ressources (HTML)

Le W3C

Définition (World Wide Web Consortium - Wikipédia)

- Le W3C est un organisme de normalisation fondé en octobre 1994 comme un consortium chargé de promouvoir la compatibilité des technologies du World Wide Web.
- Le W3C n'émet pas de normes au sens européen, mais des recommandations à valeur de standards industriels.

Modèle client-serveur

- Le serveur : logiciels (e.g. Apache) situés sur les machines des hébergeurs.
- Le client : votre navigateur Web (Mozilla, Opera, IE, Safari, Chrome) se connecte à des serveurs Web :
 - émet des requêtes HTTP ;
 - reçoit des réponses HTTP (une ressource).

Les requêtes HTTP

HTTP est un protocole client-serveur basé sur la notion de requête - réponse

- GET permet de demander la ressource spécifiée
- POST permet de transmettre des données (d'un formulaire par exemple) à une URL qui désigne un script
- PUT permet de transmettre au serveur une ressource à l'URL
- DELETE permet d'effacer la ressource spécifiée
 - GET / HTTP/1.1
 - Host: www.google.fr

Web Statique au Web Dynamique

- Au tout début du World Wide Web (1991), les pages étaient statiques : leur contenu était fixé une fois pour toute et ne variait pas.
- on se contentait de diffuser de l'information mais c'était une véritable révolution pour l'époque (hypertexte)

Web Statique au Web Dynamique

- L'introduction des CGI (Common Gateway Interface) puis par la suite la généralisation des scripts exécutables sur le serveur dans différents langages (Perl, PHP, Python, Ruby) ont permis de faire varier le contenu des pages
- on a alors parlé de **Web dynamique**
- le contenu de la page varie en fonction de l'utilisateur, de ses préférences ou du sujet abordé.

Développer pour le Web

- Il faut maîtriser de nombreuses technologies :
 - structure du document : XML, XHTML, DOM
 - rendu : CSS (feuilles de style)
 - interaction : langage cote client (Javascript)
 - interaction : langage cote serveur (PHP, Perl, Python, ...)

Application

- structure du document : HTML
- rendu : CSS (feuilles de style)
- langage cote client: (Javascript)
- langage cote serveur: PHP

Introduction au langage HTML

Plan

1. Introduction
2. Structure
 - Structure de base
 - Texte et mise en forme
 - Listes
 - Tableaux
 - Hyperliens
 - Images
3. Formulaire
4. Nouveautés dans HTML 5

Historique

- 1991: première utilisation du terme HTML par Tim Berners-Lee
- 1993: HTML première version publiée par l'IETF
- 1995: HTML 2 normalisée par le W3C
- 1997 : HTML 3.2
- 1999: sortie de HTML 4.01
- 2001 : XHTML
- 2011: HTML 5

Principe

- Fichier texte contenant des informations de structuration.
- La structure est indiquée à l'aide de balises :
 - Le nom de la balise indique le type de mise en forme à appliquer.
 - On met une balise ouvrante au début du morceau de texte concerné :
 - **<nom balise>** début du texte
 - On met une balise fermante à la fin du texte concerné :
 - fin du texte **</nom balise>** (! au /)
 - Une balise ouvrante peut contenir des attributs de la forme nom="valeur"
 - **<nom balise nom1="val1" nom2="val2" ...>** texte
 - Les attributs permettent de préciser des informations concernant la mise en forme.
 - Les balises peuvent être imbriquées.
 - C'est le navigateur qui interprète les balises pour faire l'affichage.

Éléments abordés

- La structure d'une page
- Le texte et mise en forme
- Les listes
- Les liens hypertextes
- Les images
- Les tableaux
- Les formulaires
- Les nouveautés dans HTML5

Structure d'une page HTML

- Un document HTML est en réalité un document XML avec des balises prédéfinies
 - `<!DOCTYPE html>`
- la balise `html` marque le début du document

`<html>`

La balise `Head` contient les informations relatives à la page: le type d'encodage, le titre, la description, ...

`<head>`

`<meta charset="utf-8" />`

`<meta name="description" content="Exemple de cours" />`

`<title>Mon premier exemple</title>`

`</head>`

Structure d'une page HTML

la balise body contient le corps de la page (les éléments visuels de la page)

```
<body>
```

Le document se termine par la fermeture des balise body et html

```
</body>
```

```
</html>
```

Structure d'une page HTML

```
1  <!DOCTYPE html>
2  <html lang="fr">
3      <head>
4          <meta charset="utf-8" />
5          <meta name="description" content="Exemple de cours" />
6          <meta name="author" content="Maxime MORGE" />
7
8          <title>Mon premier exemple</title>
9          <link rel="stylesheet" href="exemple.css" />
10     </head>
11     <body>
12         <p>Le corps de la page.</p>
13         <p>Il suffit de se mettre au <span class="motImportant"> travail</span>.</p>
14     </body>
15 </html>
```

Texte et mise en forme

Les titres

- Les titres sont définis par la balise `<hx>`
- ou x désigne un niveau
- Il existe 6 niveaux: 1 étant associé à la taille de caractère la plus grande et 6 la plus petite

Les paragraphes

- `<p>` texte `</p>`
texte forme un paragraphe (saut de ligne avant et après).
- `<hr/>` ligne horizontale
- `
` saut de ligne

Texte et mise en forme: texte formaté

Balises qui identifient un formatage :

- ``en gras``
- `<i>`en italique`</i>`
- `<u>`surligné`</u>`
- `<big>`en grand`</big>`
- `<small>`en petit`</small>`
- `_{`en indice`}`
- `^{`en exposant`}`

Listes

Balises qui structurent les listes :

- ``une liste non ordinale``
- ``une liste ordinale``
- ``un élément``
- `<dl>`une liste de définition`</dl>`
- `<dt>`une étiquette`</dt>`
- `<dd>`une données`</dd>`

Listes

entree 1

entree 2

entree 3

...

entree 1

entree 2

entree 3

...

- entree 1
- entree 2
- entree 3
- ...

1. entree 1
2. entree 2
3. entree 3
4. ...

Tableaux

Balises qui structurent les tableaux :

- `<table>`un tableau`</table>`
- `<thead>`un en-tête de tableau`</thead>`
- `<tbody>` un groupe du corps`</tbody>`
- `<tfoot>`un pied de tableau`</tfoot>`
- `<caption>`un titre`</caption>`
- `<tr>`une ligne`</tr>`
- `<th>`une cellule d'en-tête`</th>`
- `<td>`une cellule de données`</td>`
- `<col>`une colonne`</col>`
- `<colgroup>`une groupe de colonne`</colgroup>`

Structure de tableaux

Voir TP 1

Hyperliens: Liens hypertextes

- `texte`
 - texte devient un lien cliquable.
 - Lorsque l'on clique sur texte, on va à l'adresse (URL) adresse web.
- `Documentation HTML`
 - Le texte Documentation HTML est un lien vers la page web dont l'adresse est `http ://www.w3.org/TR/html401/`
- `Des trucs`
 - Le texte Des trucs est un lien vers la page web `truc.html` située au même endroit que la page courante.

Images

Les formats d'images acceptés

- JPEG (.jpg, .jpeg) : un grand nombre de couleurs, images bien compressées ;
- PNG (.png) : taille faible, peu de couleurs, images entrelacées (affichage progressif) ;
- GIF (.gif) : idem que PNG en moins bien.

Balise ``

- L'insertion d'une image dans une page est possible à l'aide de la balise **img**
- Deux attributs:
 - **src** : URL de l'image (absolue ou relative)
 - **alt** : un texte alternatif en cas de non chargement de l'image

Images

Balise simple

- ``

Par ex, ``

Critères de choix du format

- jpeg : idéal pour les photos et illustrations complexes contenant des millions de couleurs
- png : format sans compression 8 bits ou 24 bits ; possibilité de transparence
- gif : pratique pour les petites tailles (un logo par exemple) ; 256 couleurs max ; gestion de la transparence et de l'animation
- svg : format vectoriel ; ensemble d'objet graphique ; pas de perte de qualité ; plus léger et plus rapide

Principe

Objectifs :

- accéder a une page (dynamique) en spécifiant des paramètres ;
- permettre a l'utilisateur de saisir ces paramètres.

Comment :

- en utilisant des champs textuels, des listes déroulantes, des cases a cocher ;
- à chacun de des composants de saisie correspond un paramètre.

Balise principale

```
<form action="adresse web" method="POST" name="nom">
```

contenu formulaire

```
</form>
```

- contenu formulaire : du texte contenant en particulier des balises représentant les différents composants de saisie;
- adresse web : l'adresse de la page web dynamique à laquelle on souhaite accéder via ce formulaire ;
- POST : la méthode de transmission des paramètres (on peut également mettre GET) ;
- nom : le nom du formulaire (optionnel)

Saisie de texte

```
<input type="text" name="nom">
```

- Crée un champ de saisie pour une ligne de texte.
- nom est le nom du paramètre correspondant à ce composant.
- On peut ajouter les attributs suivants :
 - size="un nombre" : la taille du champ en caractères ;
 - value="une valeur" : texte pré-saisi
 - utile pour modifier des informations.
- On peut remplacer type="text" par type="password" si on veut afficher des * au lieu des lettres lors de la saisie.
- Pas de balise fermante.

Saisie d'un grand texte

```
<textarea name="nom" rows="h" cols="l">
```

contenu pré-saisi

```
</textarea>
```

- Crée un champ de saisie pour du texte sur plusieurs lignes.
- nom est le nom du paramètre correspondant à ce composant.
- h est la hauteur du composant en nombre de lignes.
- l est la largeur du composant en nombre de caractères.
- le contenu pré saisi peut être vide et ne contient de balise.

Liste déroulante

```
<select name="nom">  
  <option value="val1">Texte 1</option>  
  <option value="val2">Texte 2</option>  
  ...  
</select>
```

- Crée une liste déroulante ayant comme sélection possible

Texte 1, Texte 2, ...

- La valeur du paramètre nom est donnée par la sélection choisie par l'utilisateur :

val1 pour Texte 1

val2 pour Texte 2

...

- L'attribut value est optionnel.

Par défaut c'est le texte dans la balise <option></option>

On peut ajouter `selected="true"` dans une des balises options pour pré-sélectionner cette option

Bouton de soumission

```
<input type="submit" value="texte">
```

- Crée un bouton déclenchant le chargement de la page de destination (attribut action de la balise <form>).
- texte est un texte qui sera affiché sur le bouton.

```
<input type="reset" value="texte">
```

- Crée un bouton déclenchant la réinitialisation du formulaire, en utilisant les valeurs pré saisies lorsqu'elles existent.
- texte est un texte qui sera affiché sur le bouton.

Méthodes de transmission de paramètres

Deux méthodes de transmission des paramètres

- GET
 - Les paramètres sont encodés avec l'adresse de la page :
 - a la fin de l'adresse, on ajoute le caractère ?
 - puis pour chaque paramètre on ajoute nom=val
 - les paramètres sont séparés par le caractère &
 - Utile pour spécifier des paramètres dans un lien hypertexte.
- POST
 - Les paramètres sont encodés séparément de l'adresse web.
 - Plus pratique pour les formulaires

Les balises structurantes de HTML 5

`<header">`

`<! - - représente l'ente de la page - ->`

`</headers>`

- Entête de la page:
 - Logo du site
 - bannière

Les balises structurantes de HTML 5

`<nav>`

<! - - représente les liens de navigation du site- ->

`</nav>`

- On peut y définir le menu principale du site

`<nav>`

``

``

``

``

``

`</nav>`

Les balises structurantes de HTML 5

`<section>`

`<!-- contenu la section -->`

`</section>`

- Elle sert à regrouper des contenus en fonction de leur thématique

`<section>`

`<h1>Ma section</h1>`

`<p> Mon premier paragraphe </p>`

`</section>`

Les balises structurantes de HTML 5

`<aside>`

`<!-- placer les informations complémentaires -->`

`</aside>`

- Elle permet de représenter des informations complémentaires au document que l'on visualise

Les balises structurantes de HTML 5

`<article>`

`<!-- placer les informations complémentaires -->`

`</article>`

- Cette balise sert à englober une portion autonome de la page.

`<article>`

`<h1>Mon article</h1>`

`<p> Mon premier paragraphe </p>`

`</article>`

Les balises structurantes de HTML 5

`<footer>`

<! - - placer les information comme les mentions légales par exemples- ->

`</footer>`

- la balise footer permet de représenter le pieds de page. Elle se trouve en général tout en bas de notre page.



Introduction au langage CSS

Plan

1. Introduction
2. Intégration
3. Syntaxe
4. Style

Qu'est que le CSS

- Web = Fond + **Forme** + Interactivité
- Gérer la mise en forme
- Technologie = HTML + **CSS** + Javascript
- Feuilles de style en cascade, en anglais Cascading Style Sheets (CSS)
- Avantages :
 - séparation des préoccupations
 - simplifier le code HTML
 - factoriser le code

Comment intégrer le CSS

- Dans le code HTML
 - Directement dans les balises via l'attribut `style` (*méthode la moins recommandée*)
- Dans l'entête de la page
 - La balise `<head>`
- Dans un fichier séparé (*méthode la plus recommandée*)

Intégrer le CSS dans l'entête de la page

```
1  <!DOCTYPE html>
2  <html lang="fr">
3      <head>
4          <meta charset="utf-8" />
5          <meta name="description" content="Exemple de cours" />
6          <style type="text/css">
7              p{
8                  text-align:center
9              }
10         </style>
11     </head>
12     <body>
13         <p>Le corps de la page</p>
14     </body>
15 </html>
```

INTÉGRATION

Intégrer le CSS dans un fichier séparé

exemple.html

```
1 <!DOCTYPE html>
2 <html lang="fr">
3   <head>
4     <meta charset="utf-8" />
5     <meta name="description" content="Exemple de cours" />
6     <title>Mon premier exemple</title>
7     <link rel="stylesheet" href="exemple.css" />
8   </head>
9   <body>
10    <p>Le corps de la page</p>
11  </body>
12 </html>
```

exemple.css

```
1 p
2 {
3   color: red;
4   text-align: center;
5 }
6 }
```


Syntaxe CSS

La syntaxe

```
selecteur1 {  
    propriete1 : valeur1 ;  
    propriete2 : valeur2 ;  
}  
  
selecteur2 {  
    propriete1 : valeur1 ;  
    propriete2 : valeur2 ;  
}
```

exemple2.css

```
1 h1 {  
2  font-family: "Trebuchet MS", Arial, Helvetica, sans-serif;  
3  font-variant: small-caps;  
4 }  
5 p {  
6  font-family: "Times New Roman", Georgia, Serif;  
7  font-style: italic;  
8 }  
9
```

Sélecteur CSS

Sélecteur	Exemple	Description
*	*	sélectionne tous les éléments
element	p	sélectionne tous les éléments <p>
#id	li # home	sélectionne l'élément id="#home"
.class	.gauche	sélectionne tous les éléments class=".gauche"
el1, el2	table, td, th	sélectionne tous les éléments <table>, <th> et td
el1 el2	footer a	sélectionne tous les liens dans un pied de page
:hover	a : hover	sélectionne tous les liens survolés
:active	a : active	sélectionne tous les liens activés
:link	a : link	sélectionne tous les liens non visités
:after	dt : after	insert quelque chose après tous les éléments <dt>
:before	dt : before	insert quelque chose avant tous les éléments <dt>
:visited	a : visited	sélectionne tous les liens visités

Formatage du texte: la taille

- Pour modifier la taille du texte on utilise la propriété CSS **font-size** en indiquant soit la taille **absolue**(en pixel, méthode très précis) soit la taille **relative** (en pourcentage, plus souple)
- Pour indiquer une valeur relative il y a plusieurs manière:
 - xx-small : minuscule ;
 - x-small : très petit ;
 - small : petit ;
 - medium : moyen ;
 - large : grand ;
 - x-large : très grand ;
 - xx-large : très très grand

Formatage du texte: la police

- Pour indiquer la taille du texte on utilise la propriété CSS **font-family**.

Exemple :

```
p {  
  font-family:Arial ;  
}
```

Formatage du texte: Mettre en gras, italique, souligné

- Pour mettre la texte en **italique** on utilise la propriété CSS **font-style** qui peut prendre trois valeurs:
 - Normal, le texte est normal (par défaut)
 - Italic, le texte sera mis en italique
 - Oblique, le texte sera passé en oblique (légèrement différent de l'italique)

Exemple :

```
p {  
    font-family: Arial ;  
    font-style: italic ;  
}
```

Formatage du texte: Mettre en gras, italique, souligné

- Pour mettre la texte en **gras** on utilise la propriété CSS **font-weight** qui peut prendre deux valeurs:
 - Normal, le texte est normal (par défaut)
 - **bold**, le texte sera mis en gras

Exemple :

```
h1 {  
    font-style: bold ;  
}
```

Formatage du texte: Mettre en gras, italique, souligné

- La propriété CSS qui permet de **souligné** un texte est **texte-decoration**. Elle peut prendre les valeurs suivantes:
 - underline :souligné
 - line-through : barré
 - overline: souligné au dessus
 - blink : clignotant
 - none : normal

Formatage du texte: alignement

- La propriété CSS qui permet de faire des **alignements** (à gauche, centré, à droite, justifié) un texte est **texte-align** . Elle peut prendre les valeurs suivantes:
 - left : texte aligné à gauche
 - center : texte centré
 - right: texte aligné à droite
 - justify : le texte sera justifié

La couleur et le fond: couleur du texte

- La propriété CSS qui permet de modifier la couleur du texte est **color** .

Exemple :

```
h1 {  
    color: red ;  
}  
  
p {  
    color: #FFFFFF  
}
```

La couleur et le fond: couleur du fond

- La propriété CSS qui permet de modifier la couleur de fond est **background-color**.

Exemple :

```
body {  
    background-color: black ;  
    color: white;  
}
```

La couleur et le fond: image du fond

- La propriété CSS qui permet d'insérer une image de fond est **background-color** .

Exemple :

```
body {  
    background-image: url("monimage.png") ;  
    color: white;  
}
```