



Java EE

-

Cours 7



Cours de 2^e année ingénieur
Spécialisation « Génie Informatique »



Rappels



Feuilles de Styles : CSS

- Lorsque l'on réalise un site Web, il faut dissocier la partie **forme** de la partie **fond**.
- Le fond correspond au contenu de notre page Web (texte, images, ...)
- La forme correspond à la mise en page de notre contenu (couleurs, cadres, centrage, polices, tailles...)
- Le fond est stocké dans le fichier **HTML**.
- La forme est stocké dans une feuille de style : un fichier **CSS** (**Cascading Style Sheets**)



Feuilles de Styles : CSS

- Dans la plupart des cas, il faut éviter de définir la forme d'un site Web directement dans les fichiers HTML (même si c'est possible de le faire) pour plusieurs raisons :
 - **Réutilisabilité / Homogénéité** : Une fois défini, un fichier CSS peut être appliqué très facilement sur l'ensemble d'un site Web. Toutes les pages Web auront donc une même forme.
 - **Maintenance / Mise à jour** : Si l'on veut faire une modification sur le visuel de notre site, il suffit de modifier la propriété correspondante dans le fichier CSS et tout le site sera automatiquement affecté.
 - **Séparation des tâches** : La structure et la présentation sont gérées séparément
 - **Simplification** : Le code HTML est réduit en taille et en complexité



Feuilles de Styles : CSS

- Utilisation :

- Un fichier CSS est un fichier **texte** d'extension .css
- Il est lié à une page HTML par la balise suivante (dans la balise <head>)

```
<link rel="stylesheet" type="text/css" href="style.css">
```

nom du fichier CSS
- Un fichier CSS contient un ensemble de règles d'affichage.

Exemple : Le fichier CSS suivant fera afficher tous les titres h1 en italique

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



Feuilles de Styles : CSS

- L'inconvénient de l'écriture précédente est qu'elle va s'appliquer sur toutes les balises h1. Si l'on veut spécifier son utilisation à certaines balises, on doit utiliser un attribut *class* ou bien *id* pour cette balise.

Ex:

fichier html

```
<h1> Voici un Titre h1 </h1>
```

```
<h1> En voici un autre </h1>
```

```
<h1 class="rouge"> Celui-ci a une classe spéciale </h1>
```

Fichier css

```
.rouge  
{  
    color: red;  
}
```

Voici un Titre h1

En voici un autre

Celui-ci a une classe spéciale

- La différence entre *id* et *class* est que l'attribut *id* ne doit être utilisé qu'une fois dans tout le document, il est **unique**.



Feuilles de Styles : CSS

- Le dernier inconvénient que l'on pourrait avoir est que les balises ou classes définies dans le fichier CSS sont des balises HTML prédéfinies. Comment pouvoir créer un style s'appliquant uniquement à une portion de texte du texte suivant ?

```
<p>
```

```
  Lorem ipsum dolor sit amet, consectetur adipiscing  
  elit. Sed non risus. Suspendisse lectus tortor,  
  dignissim sit amet, adipiscing nec, ultricies sed,  
  dolor.
```

```
</p>
```



Feuilles de Styles : CSS

- Il faut utiliser les balises `` (ligne) ou `<div>` (bloc). Ce sont des balises qui n'ont aucun effet, leur seule utilité est de pouvoir appliquer un style CSS particulier entre leurs bornes.

Ex:

```
<p>
```

```
  Lorem ipsum dolor sit amet, <span class="rouge">consectetuer  
  adipiscing</span> elit. Sed non risus. Suspendisse lectus  
  tortor, dignissim sit amet, adipiscing nec, ultricies sed,  
  dolor.
```

```
</p>
```

Lorem ipsum dolor sit amet, **consectetuer adipiscing** elit. Sed non risus.
 Suspendisse lectus tortor, dignissim sit amet, adipiscing nec, ultricies sed,
 dolor.



Quelques balises HTML

- Pour insérer une image dans un fichier html, on utilise la balise suivante :

```

```

- Le format de l'image peut être bmp, jpg, gif, ...
 - L'attribut alt est obligatoire si l'on respecte la norme W3C
-
- La balise `<FIELDSET>` permet un regroupement de champs de saisie en catégories logiques (thématiques).
 - Elle peut être associée à une balise `<LEGEND>` pour ajouter une légende au cadre.

Informations personnelles

Nom : Prénom : Adresse :



Gestion de BDD en Java

- Java peut gérer des bases de données relationnelles en utilisant les JDBC (Java DataBase Connectivity).
 - En JEE, notre serveur étant codé en Java, il est également possible d'utiliser les jdbc pour communiquer avec une BDD.
- Java peut également gérer des fichiers XML en utilisant DOM ou SAX (cf le cours correspondant). Nous pouvons donc également stocker des informations via ce format.
- Il existe des frameworks spécifiques (appelés frameworks de persistance) permettant de mapper des tables de bases de données (relationnelles ou XML) directement en objets Java.
 - Ceux-ci sont utilisés via un fichier .jar à ajouter au projet Java.
 - Ils peuvent être orientés uniquement Java EE ou bien plus largement Java SE.
 - Le plus connu est **Hibernate**



Gestion des ressources externes

- Le serveur JEE peut contenir un ensemble de fichiers de ressources : images, CSS, videos, XML, ...
- Sous Eclipse, tous les fichiers de ressources doivent être placés dans le dossier WebContent, qui sera ensuite déployé par Eclipse à la racine de notre serveur.
- Il est possible de créer une architecture de dossiers dans le dossier WebContent afin de "classer" les ressources, elle sera déployé ainsi à la racine de notre serveur.
- Pour accéder à l'URL d'un fichier sur notre serveur, il y a plusieurs possibilités :
 - Utiliser la propriété "wtp.deploy" => Ne fonctionne que sous Eclipse
 - Utiliser l'URL de notre site Web => <http://localhost:8080/...>
 - Utiliser la syntaxe suivante dans une Servlet, qui nous donne le chemin physique de l'emplacement où est déployé notre serveur sur la machine :
`getServletContext().getRealPath(this.getServletConfig().getServletName())`



Les Cookies



Historique

- Inventés par **Lou Montulli** alors qu'il travaillait chez Netscape Communications.
- Leur but premier est d'adapter dynamiquement le contenu d'un site Web aux habitudes de navigation de l'internaute.
- Ils sont souvent l'objet de polémique, car ils peuvent permettre à un site de récupérer des informations sur votre activité Internet. Par exemple sur les différents sites que vous avez visités et éventuellement les données que vous avez échangés.



Fonctionnement

- Les cookies sont des fichiers textes, **stockés du côté du client** par son navigateur Web.
- C'est le site qui demande au navigateur du client de créer un cookie dans l'en-tête HTTP de sa réponse. Le navigateur peut être configuré pour le refuser.
- L'en-tête HTTP réservé à l'utilisation des cookies s'appelle *Set-Cookie*, il s'agit d'une simple ligne de texte de la forme :

`Set-Cookie : NOM=VALEUR;domain=NOM_DOMAINE;expires=DATE`



Fonctionnement

- Chaque navigateur a sa propre manière de stockage des cookies (un fichier ou plusieurs, crypté ou pas, ...)
- Un cookie a une durée de vie limitée, fixée par le concepteur du site. Ils peuvent aussi expirer à la fin de la session de l'utilisateur.
- Les cookies ne sont donc pas forcément supprimés après fermeture du navigateur. Ainsi, même après fin de la session, ils conservent leurs informations sur la machine du client.





Pourquoi les utiliser?

- Enregistrer des informations sur le visiteur ou sur son parcours dans le site.
- Reconnaître les habitudes d'un visiteur et personnaliser la présentation de son site pour chaque visiteur.
- Mémoriser des informations sur le long terme pour chaque visiteur.

Ex :

- Garder en mémoire un panier d'achat dans le cas d'un site de vente
- Retenir les identifiants de connexion à une éventuelle partie privée : lorsque le visiteur revient sur le site, il ne lui est plus nécessaire de taper son nom et son mot de passe pour se faire reconnaître, puisqu'ils sont automatiquement envoyés par le *cookie*



Limitations

- Les cookies sont soumis a plusieurs contraintes :
 - Leur nombre total est limité à 300.
 - La taille maximale d'un cookie est de 4 ko.
 - Il ne peut exister au maximum que 20 cookies par domaine.



Structure

- Un Cookie contient plusieurs paramètres dont 2 principaux :
 - **Name** : Nom du Cookie
 - **Value** : Valeur du Cookie
- Il peut également stocker d'autres informations plus générales :
 - **MaxAge** : La durée de vie maximale du cookie (en secondes), -1 s'il est détruit à la fermeture du navigateur
 - **Comment** : Commentaire décrivant le but de ce cookie
 - **Domain** : Le nom du domaine pour ce cookie
 - **Path** : L'URL de la page sur le serveur qui a créé ce cookie
 - **Secure** : Décrit si le navigateur peut envoyer ce cookie à travers n'importe quel protocole, ou s'il doit être sécurisé
 - **Version** : Décrit la version du protocole avec lequel ce cookie est accepté



Les Cookies en JEE

- Le conteneur JEE se charge de gérer le parsing de la requête HTTP entrante et sortante.
- Il crée un tableau contenant les informations de tous les cookies envoyés par le navigateur du client. Ce tableau peut être récupéré dans l'objet request.
- Pour récupérer les cookies provenant de la requête du client, il suffit d'utiliser la méthode *getCookies()* de l'objet *HttpServletRequest*.

```
Cookie[] getCookies()
```

- La classe *javax.servlet.http.Cookie* représente l'objet Java encapsulant toutes les opérations nécessaires à la manipulation des cookies.



Les Cookies en JEE

- Le serveur JEE peut également envoyer une demande de création de Cookie au Client. Pour cela, il doit créer un objet Cookie puis appeler la méthode *addCookie()* de l'objet *HttpServletResponse*.

```
response.addCookie(MonCookie);
```

- Le constructeur de la classe *Cookie* crée un cookie avec un nom et une valeur passés en paramètre. Il est toutefois possible de modifier la valeur de ce cookie ultérieurement grâce à sa méthode *setValue()*.

```
Cookie MonCookie = new Cookie("nom", "valeur");
```

- La méthode *getCookies()* retourne un tableau contenant l'ensemble des cookies présents chez le client. Il est ainsi possible de parcourir le tableau afin de retrouver un cookie spécifique grâce à la méthode *getName()* de l'objet *Cookie*.
- La récupération de la valeur d'un cookie se fait grâce à la méthode *getValue()* de l'objet *Cookie*.

```
String Valeur = Cookie.getValue();
```



Exemple

```
Cookie[] mesCookies = request.getCookies(); // Je récupère tous les cookies
if (mesCookies!=null)
{
    for (int i=0; i<mesCookies.length; i++)
    {
        // S'il existe un Cookie nommé "nomCookie"
        if (mesCookies[i].getName().equals("nomCookie"))
        {
            // J'ai trouvé le cookie que je cherchais, j'affiche sa valeur:
            System.out.println(mesCookies[i].getValue());
        }
    }
}
```



Webographie

- Cours complet sur les CSS :

<http://www.cssdebutant.com/>

- Site de sun sur la classe Cookie:

<http://java.sun.com/j2ee/1.4/docs/api/javax/servlet/http/Cookie.html>

- Description de l'utilisation des Cookies:

<http://www.commentcamarche.net/contents/securite/cookies.php3>

