CONCEPTION ET DEVELOPPEMENT D'APPLICATIONS INTERNET

CDAI 1 - GENERALITES

Université Paris Dauphine
Master M2 MIAGE
Année 2014-2015

Bekhouche Abdesslem
Sobral Diogo

DAUPHINE
UNIVERSITÉ PARIS

Déroulement

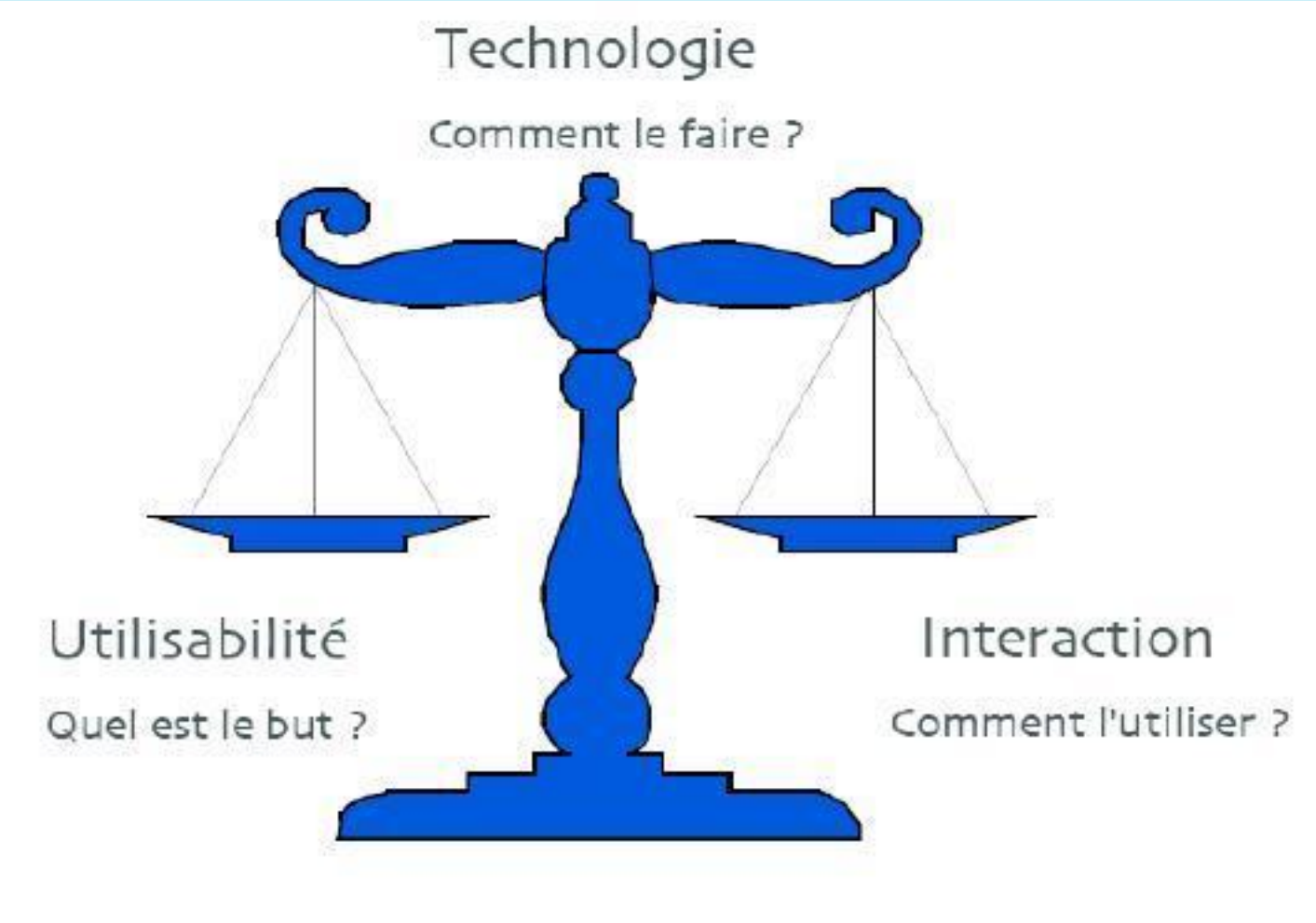
- Cours : Comment concevoir une application internet.
- TPs : Travaux progressifs devant aboutir à la réalisation d'une application internet
- Évaluation : Projet

Plan

1. Rappel HTML
2. Introduction au CSS
3. Introduction PhP
4. JavaScript
5. Java

Concevoir une application internet

Chercher l'équilibre ...



Généralités

- Qu'est-ce que le Web (ou World Wide Web, Toile, WWW, W3)?
  - Système hypertexte public : système contenant des documents liés entre eux par des hyperliens permettant de passer automatiquement d' un document à l'autre.

- Différence entre le Web et Internet?
  - Internet : réseau mondial d'ordinateurs permettant aux utilisateurs de communiquer (courrier électronique), de publier des informations (Web), de transférer des données (FTP), de travailler à distance (telnet et ssh). . .
  - Web : un aspect d'Internet.

Http : Généralités

- Protocole
  Ensemble normalisé de règles décrivant la manière de transmettre des informations, par exemple sur un réseau comme Internet entre un client et un serveur.

- HTTP
  HyperText Transfer Protocol, le plus utilisé des protocoles de communication sur le World Wide Web. Permet à un client Web d' indiquer quelle page il veut obtenir, et au serveur Web de lui répondre en lui donnant cette page.

HTML : HyperText Markup Language

- normalisé par le W3C (World Wide Web Consortium) regroupant industriels (Microsoft, Google, Apple. . .) et académiques (INRIA, MIT. . .)

- format ouvert : lecture possible dans des conditions correctes sans contrainte matérielle ou logicielle

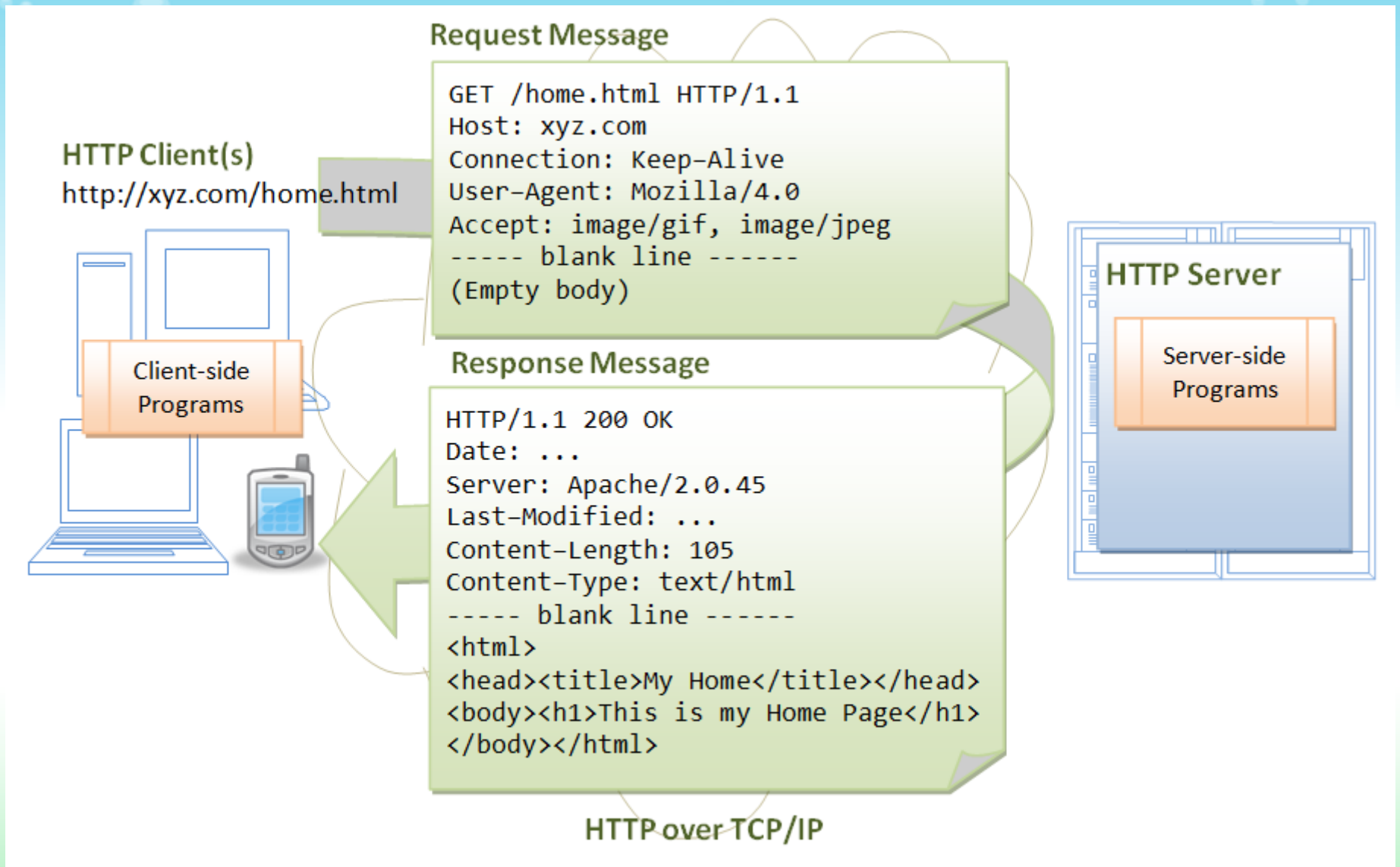
- un fichier texte avec des balises

- description de la structure et du contenu d'un document, accent sur l'accessibilité

- on ne décrit pas la présentation (ce sera le rôle de CSS)

- on ne décrit pas de comportement dynamique (ce sera le rôle de JavaScript et des langages côté serveur)

Client serveur



Source : https://rmohan.com/?p=1488

HTML suite …

- HTML est un langage qui alterne texte et balises (<blabla> ou </blabla>)
  - Les balises permettent de structurer chaque partie du document et servent entre autres au navigateur pour réaliser la mise en page du document.

- Les fichiers HTML
  - sont structurés en deux parties principales :
    l'en-tête <head> … </head>
    et le corps <body> … </body>

- En HTML, les blancs (espaces, tabulations, retours à la ligne) sont en général équivalents et servent juste à délimiter mots, balises. . . Leur nombre n'a pas d'importance.

HTML : structure d'un document

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
  "http://www.w3.org/TR/html4/strict.dtd">
<html lang="fr">
  <head>
    <!-- En-tête du document -->
  </head>
  <body>
    <!-- Corps du document -->
  </body>
</html>
```

- La déclaration <!DOCTYPE ...> précise la version d'HTML utilisée.
- La langue du document est précisée avec l'attribut lang de la balise principale <html> .

HTML : les listes

- Les listes classiques :
  - Les listes non numérotées délimitées par les balises <ul> ... (unordered list).
  - Les listes numérotées délimitées par les balises <ol> ... (ordered list).
  - Tous les éléments d'une liste numérotée ou non sont délimités par les balises <li> ... (list item)

- Les lexiques sont délimités par les balises <dl> ... </dl> (definition list) et leurs entrées par les balises <dt> ... </dt> (term) et <dd> ... </dd> (definition).

- Exemples

```
<ol> <li>un</li> <li>deux</li> </ol>

<dl> <dt>lapin</dt> <dd>rongeur à oreilles</dd> </dl>
```

HTML : les tableaux

- Les tableaux sont délimités par les balises <table>… </table> .
- Les balises <tr> … </tr> (table row) délimitent les lignes.
- Les balises <td> … </td> (table data) délimitent les cellules.

Attention ! On déclare les lignes à l'intérieur du tableau et les cellules à l'intérieur des lignes.

Exemples

```
<table>
  <tr> <td> l1, c1 </td> <td> l1, c2 </td> </tr>
  <tr> <td> l2, c1 </td> <td> l2, c2 </td> </tr>
</table>
```

HTML : les images

- Pour insérer une image dans un document HTML, on utilise la balise <img> .
  - o L'attribut src permet de préciser où se trouve l'image.
  - o L'attribut alt permet de remplacer l'image par un texte quand elle n'est pas disponible. Il est obligatoire de l'utiliser, pour que tout agent (malvoyants, navigateur texte, incidents techniques, robots) ne pouvant voir votre image puisse avoir un texte alternatif.

```
<img src="http://www.lri.fr/images/lri.png" alt="LRI">
<img src="../images/montblanc.png" alt="Mont Blanc">
```

- Les formats d'images utilisables pour le Web sont :
  - o Le JPEG (.jpg), un format adapté aux photos.
  - o Le GIF (.gif) et le PNG (.png), des formats adaptés aux autres types d'image ; le PNG est à préférer dans tous les cas (transparence, profondeur de couleurs. . .) sauf besoin d'images animées (à utiliser avec parcimonie !).

HTML : les formulaires

- Permettent d'interagir avec l'utilisateur en lui proposant d' entrer des informations.
- En HTML : uniquement l'interface de formulaire
- L'essentiel du travail sera fait par le script qui traitera la soumission du formulaire

HTML : les formulaires suite …

- Un formulaire HTML est placé à l'intérieur d'une balise <form> .

- Celle-ci prend les attributs suivants :
  - action URL du script auquel sera soumis le formulaire.
  - method Méthode HTTP, valant soit "get" soit "post" .
  - enctype Encodage HTTP. Peut valoir :
    "application/x-www-form-urlencoded" (valeur par défaut, tous les caractères sont encodés), "multipart/form-data" (pas d'encodage) et "text/plain" (espaces encodés, caractéres spéciaux non).

- Exemples

```
<form action="action.php" method="get">
  <div><input type="submit"></div>
</form>
```

HTML : les formulaires suite …

En HTML, il est interdit de mettre des champs de formulaire directement à l'intérieur d'un <form> . Il faut d'abord les regrouper :

- Dans des paragraphes <p> si les champs de formulaires sont à l'intérieur de paragraphes de textes (rare).
- Dans des ensembles de champ <fieldset> pour regrouper des champs de formulaire de sémantique proche. On pourra alors donner une légende à l' ensemble de champs avec la balise <legend> .
- Dans des divisions sans contenu sémantique sinon.

Exemples

```
<fieldset>
  <legend>Mensurations</legend>
  <input type="text" name="taille">
  <input type="text" name="poids">
</fieldset>
```

HTML : les formulaires suite …

- La plupart des champs sont naturellement accompagnés d'une étiquette (<label>).
- On peut la placer où on veut, en général juste à gauche ou à droite du champ.
- Son attribut for référence l'attribut id du champ correspondant.
- Dans les navigateurs graphiques, un clic sur l'étiquette d'un champ permet en général de sélectionner le champ.

Exemples

```
<label for="taille">Taille:</label>
<input type="text" name="taille" id="taille">
```

HTML : les formulaires suite …

- La balise <input> a une utilisation très vaste dans les formulaires. Elle représente un champ de saisie.
- L'attribut type détermine le type (texte, mot de passe, liste, etc.) du champ.
- L'attribut name (nom du paramètre de la requête HTTP) est obligatoire (sauf pour les types "reset" et "submit") ; il permet de préciser au serveur à quelle saisie on fait référence.

Exemple :

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

HTML : les formulaires suite ...

- Saisie d'une ligne de texte

```
<input type="text" name="prenom" value="Jordy" maxlength="50">
```

- Saisie d'un mot de passe

```
<input type="password" name="pwd" value="12345678">
```

- Choix multiples parmi une liste

```
<input type="checkbox" name="pub[]" value="site"
  checked="checked" id="pub-site">
<label for="pub-site">Recevoir des offres de notre site</label>

<input type="checkbox" name="pub[]" checked="checked"
  value="externe" id="pub-externe">
<label for="pub-externe">Recevoir des offres externes</label>
```

HTML : les formulaires suite ...

- Choix unique parmi une liste

```
Recevoir de la pub:
<input type="radio" name="pub" value="oui" id="pub-oui"
  checked="checked">
<label for="pub-oui">oui</label>

<input type="radio" name="pub" value="non" id="pub-non">
<label for="pub-non">non</label>
```

- Fichiers joint

```
<label for="fichier">Fichier:</label>
<input type="file" name="fichier" id="fichier">
```

- Ré-initialisation d'un formulaire

```
<input type="reset" value="Tout effacer">
```

- soumettre le formulaire

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

HTML : les formulaires suite ...et fin

- Champs cachés
  - type ="hidden" permet de cacher des champs au client mais leur contenu est envoyé avec le formulaire.
  - Ceci permet de préciser des informations, en utilisant l'attribut value , concernant l'interaction client/serveur.
  - C'est à utiliser avec précaution car cela peut être à l'origine de problèmes de sécurité assez graves : ne pas oublier que le client peut éditer la page à la main pour changer la valeur de ces champs!

```
<input type="hidden" name="monnaie_utilisee" value="EUR">
<input type="hidden" name="customerCB" value="c2415-345-8563">
```

- Saisie de plusieurs lignes de texte

```
<textarea name="bio" cols="40" rows="5">
  Fille de Josiane Balasko,
  Marilou Berry fait ses premiers pas au cinéma à 8 ans...
</textarea>
```

HTML vs XHTML

- XHTML : une application d'XML
- Les balises sans contenus <hr> , s'écrivent <hr /> en XHTML.
- Certains éléments peuvent ne pas être refermées en HTML (un deux), mais la fermeture est obligatoire en XHTML.
- Les valeurs des attributs peuvent ne pas être entre guillemets (<img src=toto.png alt=toto>) en HTML, guillemets obligatoires en XHTML.
- Les noms des éléments et des attributs sont insensibles à la cass en HTML (<HTMl laNg=fr>), contrairement à XHTML ou tout doit être en minuscule.
- Attributs xmlns et xml:lang sur la balise <html> en XHTML.
- Et quelques autres petites subtilités. . .

Avantages respectifs de HTML et XHTML

- Avantages de HTML 5.0
  - Meilleur support par les navigateurs (Internet Explorer 6/7 comprennent mal XHTML).
  - Moins de contraintes. . .
  - Beaucoup plus utilisé sur le Web.

- Avantages de XHTML 2.0
  - Plus de contraintes. . . (donc plus simple !).
  - Syntaxe claire, sans ambiguïté.
  - Familiarité avec XML, utile dans d'autres contextes.
  - Facilité d'utilisation dans des contextes XML (p.ex. XSLT).
  - IE 9 comprend mieux le XHTML.

HTML : outils

- N'importe quel éditeur de texte (par exemple le bloc-notes de Windows, emacs, vim. . .). Privilégier les éditeurs avec coloration syntaxique. Bon choix : Scite (libre, léger, simple d'utilisation, multi-plateforme, coloration syntaxique pour HTML/CSS PHP/JavaScript/. . .), WordPad ...
- N'importe quel navigateur, et plusieurs navigateurs avec un moteur de rendu différent
- Utiliser la fonction « Afficher la source » ou encore "Outils de développement" des navigateurs Web
- Firefox présente de nombreux avantages pour le développement/analyse de sites Web
- Google Chrome, Safari ...
- Pour vérifier qu'une page se conforme bien aux normes de HTML, utiliser le validateur du W3C : http://validator.w3.org/

HTML : références

- http://www.w3schools.com/html/
- http://www.w3.org/html/
- http://playground.html5rocks.com/

Exemple ...

CSS : Définition

- CSS : Cascading StyleSheets
- Recommandation du W3C
- Plusieurs niveaux : CSS1 (1996), CSS2 (1998), CSS2.1 (2005),CSS3, CSS4 (en cours)
- Support par les navigateurs très inégal, jamais complet (en particulier, IE6 et IE7 ont plusieurs limitations importantes).
- Le navigateur IE9 supporte mieux la version CSS4

- Principe
  - Support par structure et le contenu
  - CSS décrit :
    - la mise en forme du texte
    - la mise en page des boîtes les unes par rapport aux autres

CSS : Style en ligne

- Manière le plus simple d'utiliser les CSS.
- Rajouter un attribut style sur les balises HTML.
- On peut utiliser si on a besoin d'une balise supplémentaire.
- Encombre le code HTML avec des indications de mise en forme : ce n'est pas ce qu'on veut !

Exemple :

```
Ce mot en <em style="color: red;">emphase</em> est aussi en
rouge.
```

```
<span style="text-decoration: underline">Cette expression de
plusieurs mots</span> est soulignée.
```

CSS : styles en tête

- Intégration des propriétés de style à l'en-tête de la page avec la balise <style>
- Utilisation des sélecteurs pour définir à quels élément les propriétés s'appliquent
- Inconvénients : mélange HTML et CSS dans le même document, impossible de réutiliser les propriétés CSS dans plusieurs documents

Exemple:

```
<!DOCTYPE ... >
<html>
  <head> ...
    <style type="text/css">
      em { color: red; }
    </style>
  </head>
  <body> ... </body>
</html>
```

CSS : feuille de styles liée

- Mettre la feuille de style CSS dans un fichier à part (en général, on utilise l'extension .css).
- Permet d'utiliser la même feuille de style pour plusieurs pages Web.
- Rajouter une balise <link> dont l'attribut rel est positionné à "stylesheet" dans l'en-tête du document.
- Possibilité d'ajouter media="screen" ou media="print" , etc., pour choisir différentes feuilles de style suivant le mode d'affichage.

```
<!DOCTYPE ... >
<html>
  <head> ...
    <link rel="stylesheet" href="feuille.css" type="text/css">
  </head>
  <body> ... </body>
</html>
```

CSS : classes

- On veut parfois rajouter encore plus de structure et de sémantique à un document HTML.
- On utilise l'attribut class sur n'importe quelle balise (ou, à défaut, sur une balise).
- Après, on peut utiliser CSS pour appliquer une mise en forme commune à tout ce qui fait partie d'une class particulière.

Exemple (Mettre en bleu italique les noms de personnes)

HTML

```
<p>Je voudrais remercier en particulier
<span class="personne">Madame Machin</span>
et <span class="personne">Monsieur Bidule</span>.</p>
```

CSS

```
.personne { color: blue; font-style: italic; }
```

CSS : syntaxe

- Ensemble de règles de la forme :

```
sélecteur {
  propriété: valeur;
}
```

- sélecteur : indique à quelles parties du documents la règle s'applique
- propriété : propriété spécifique de mise en forme à modifier
- valeur : son sens dépend de la propriété.
- Les feuilles de style peuvent être validées avec un validateur approprié, cf. http://jigsaw.w3.org/css-validator/
- Commentaires entre /* et */.

CSS : pour aller plus loin

- Sélecteurs de CSS
  - o sélecteurs simples, multiples, universel
  - o sélecteurs de classes
  - o sélecteurs d'identifiant
  - o sélecteurs contextuels
  - o pseudo éléments
  - o pseudo classes

- Mise en forme
  - o propriété de longueur
  - o police, espacement, alignement et indentation
  - o propriété des listes, couleur
  - o ...

- Mise en page
  - o Marge, bordure, espacement, visibilité, tables
  - o...

CSS : outils

- N’importe quel éditeur de texte
- N’importe quel navigateur graphique
- Fonction Aucun style de Firefox
- Extension cqstyle pour Google Chrome
- Validateur CSS : http://jigsaw.w3.org/css-validator/

CSS : références

- Les spécifications de CSS :
  http://www.w3.org/TR/REC-CSS1
  http://www.w3.org/TR/CSS21/
  http://www.w3.org/Style/CSS/current-work

- Wiki de support des standards du Web :
  http://www.webdevout.net/

Exemple ...

Rôle des programmes coté serveur

- Le Web, ce n'est pas qu'un ensemble de documents HTML statiques !

- Les programmes côté serveur permettent :
  - de traiter des soumissions de formulaire ;
  - d'afficher de manière uniforme l'ensemble des pages d'un site ;
  - de proposer des applications interactives ;
  - de permettre à l'utilisateur d'ajouter ou modifier du contenu ;
  - etc

Langage coté serveur

PHP : un des langages les plus populaires, s'intègre très facilement avec Apache (libre).

ASP et ASP.NET : destiné à être utilisé avec IIS (Microsoft, commercial)

ColdFusion (Adobe, commercial)

JSP (Java Server Pages) : permet de mêler instructions Java et code HTML; nécessite un serveur d'applications Java (p. ex.,Tomcat) en plus d' Apache (Sun, gratuit voire libre).

Servlets Java : véritables programmes Java, plutôt pour les applications complexes côté serveur avec peu d'interaction côté client ; nécessite un serveur d'applications Java en plus d'Apache (Sun, gratuit voire libre).

Framework d'application Web

- Les langages présentés ci-avant restent assez basiques et généralistes.
- N'encouragent pas forcément une organisation propre d'un site Web.
- **Framework** : ensemble d'un langage de programmation, d'une bibliothèque de fonctions, d'outils externes, de bonnes pratiques à suivre. . .
- Permet d'abstraire la création d'une page Web.
- Suit en général le modèle **MVC** (voir transparent suivant).
- Inclut parfois la génération de code **JavaScript** côté client pour créer directement une application Web fortement dynamique (p. ex., validation de formulaire) ; intégration **Ajax** également.
- Fortement recommandé pour créer des applications complexes. . . mais également complexe à maîtriser !

Modèle MVC

- Principe de génie logiciel, utilisé dans d'autres domaines
- Particulièrement adapté au cas des applications Web!
- Séparation propre entre :

Modèle : données manipulées par l'application et fonctions de manipulation de ces données ; réutilisable pour d'autres applications Web

Vue : présentation des données ; facilement échangeable pour changer l'apparence et la structure du site

Contrôleur : contrôle la manière dont l'utilisateur interagit, au travers de la vue, avec les données du modèle

Frameworks coté serveur les plus populaires

ASP.NET : DotNet
ColdFusion : Model-Glue, Fusebox
Java : Struts, Spring, JavaServer Faces, Google Web Toolkit
Perl : Catalyst
PHP : CakePHP, Symphony, Zen
Python : Django
Ruby : Ruby on Rails (a eu beaucoup d'influence !)
Smalltalk : Seaside

. . . et beaucoup d'autres

PHP et HTML

- Script PHP : document HTML (par exemple), dans lequel est incorporé du code PHP.
- Le code PHP est à l'intérieur d'une pseudo-balise <*?php* ... *?*> (ou <*?* ... *?*> , ou <*?*= ... *?*> qui est un raccourci pour <*? echo* ... *?*>).

Exemple

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">

<html>
  ...
  <body>
  <h1><?php echo 2+2; ?></h1>
</body>
</html>
```

Introduction au PhP

- Un script PHP est une suite d'instructions terminées par des points-virgules.

Exemple (Écrire une phrase avec l'instruction echo)

```
echo 'Ceci apparaîtra dans la page générée.';
```

- Ces instructions contiennent des éléments variables ou constants, peuvent être conditionnées ou encore itérées plusieurs fois.

PhP : notion à retenir

- variable et affectation : *$var = valeur*
- instruction conditionnelle : *if (C) { T } else { E }*
- boucle : *for (I;C;P) { B }*
- $_GET et $_POST

Exemple

```
echo "<p>Votre login est: " . $_POST["login"] . "</p>";
echo "<p>Vous avez coché les genres: ";
for($i=1;$i<=count($_POST['genre']);$i=$i+1) {
  echo $_POST['genre'][$i] . " ";
}
echo "</p>";
```

Javascript : définition

- PHP, CGI. . . : permettent des comportements dynamiques côté serveur. Nécessitent un échange entre le navigateur et le serveur Web (soumission d'un formulaire, clic sur un lien) pour chaque comportement dynamique souhaité.

- JavaScript : permet des comportements dynamiques côté client : manipulation des fenêtres, changement dynamique du code HTML/CSS, interaction fine avec les formulaires. . .

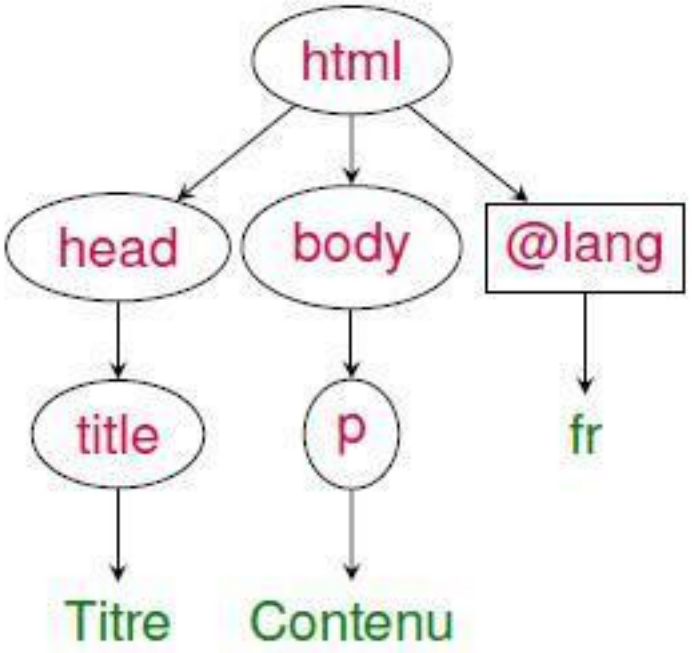
- Permet la manipulation du DOM (Document Object Model), lareprésentation du document HTML comme un arbre, les balises étant les noeuds de l'arbre.

- « Dynamic HTML » (DHTML) : JavaScript + DOM + CSS

- Rien à voir avec Java!

DOM : exemple

```
<html lang="fr" xmlns="...">
  <head><title>Titre</title></head>
  <body><p>Contenu</p></body>
</html>
```



Langage JavaScript

- Langage de Programmation
- Script (ou programme) JavaScript : fichier texte, instructions terminées par des points-virgules
- Normalisé sous le nom d'EcmaScript (JavaScript est historiquement le nom de l'implémentation Netscape, JScript étant l'équivalent chez Microsoft)
- En pratique, actuellement, seuls les navigateurs graphiques supportent JavaScript. Conséquence : sauf applications complexes, un site doit être utilisable sans JavaScript.

Liaison avec HTML

- toto.js contenant des fonctions JavaScript (function)
- Dans le <*head*> du HTML :

```
<script src="toto.js" type="text/javascript"></script>
```

- Gestionnaires d'événement comme attributs des balises HTML (cf plus loin).

- On peut aussi mettre directement du code JavaScript à l'intérieur des balises <*script*> mais :
  - o Mélange de plusieurs langages dans le même document: ajoute de la complexité.
  - o Le document complet doit rester du HTML/XHTML valide !
  - o Impossible de partager les mêmes fonctions dans plusieurs pages Web sans les recopier à chaque fois.

- Dernière possibilité : utiliser le pseudo-protocole *javascript :* dans un lien.

JavaScript : généralité

- Notion de variable et d'affectation : *var = valeur*
- Instruction conditionnelle : if (C) { T } else { E }

```
if (sortie=='q') { message='Merci et au revoir!'; }

if (a > b) { message=a+" est plus grand que "+b; }
else {
  if (a==b) { message=a+" est égal a "+b; }
  else { message=a+" est plus petit que "+b; }
}
```

- Boucle : for (I;C;P) { B }

```
for( i=0; i<10; i=i+1 ) { /* faire quelque chose */ }

fact=1;
for( i=13; i>0; i=i-1 ) { fact = fact*i; }
```

JavaScript : modèle objet

- JavaScript basé sur le modèle de programmation objet.
- Variables : objets complexes, ayant des propriétés (membres) et des fonctionnalités (fonctions membres, méthodes).
- En JavaScript, on accède au membre blah de l'objet toto avec toto.blah, et on utilise la méthode bouh de l'objet toto avec toto.bouh(arguments).

Exemple :
Par exemple, un objet *voiture* pourrait avoir une propriété *couleur* et des fonctionnalités *tourneGauche*(), *tourneDroite*(), *avance*(distance).

```
voiture.couleur="bleu";
voiture.avance(100);
voiture.tourneGauche();
```

- En pratique, les objets JavaScript qu'on utilisera représenteront le document HTML, les noeuds du documents, la fenêtre. . .

JavaScript : exemple

JavaScript :

```
function Test() {
  document.getElementById("paragraphe").style.color = "blue";
}
```

HTML :

```
<p id="paragraphe" style="color: red;">un texte</p>
<a href="" onclick="Test()">Test</a>
```

Explication : L'exemple contient un paragraphe avec le nom id paragraphe et un lien qui si on le clique appelle la fonction Test(). Cette fonction change la propriété CSS color du paragraphe, de telle sorte que le paragraphe perde sa couleur rouge et devienne bleu.

PhP vs JavaScript

Caractéristique	PHP	Javascript
Exécution	Exécuté sur le serveur.	Exécuté chez le client.
Nécessaire à l'exécution	Un interpréteur PHP doit être installé sur le serveur.	Tous les navigateurs possèdent un interpréteur Javascript (mais peut être désactivé).
Manipulation de fichiers	Lecture, écriture, ajout possible dans des fichiers texte et éventuellement binaire situés sur le serveur.	Totalement incapable de manipuler les fichiers.
Cookies	Il est possible d'utiliser les cookies dans les deux langages, mais l'utilisation est simplifiée en PHP.	
Données issues de formulaires (POST)	PHP permet de récupérer les données d'un formulaire.	Au contraire, Javascript permet uniquement d'accéder aux différents champs d'un formulaire tant que celui-ci est apparent sur la page.
Données passées par URL (GET)	Ici encore, les deux langages permettent de récupérer les variables passées par URL. Cependant, PHP permet de manipuler plus facilement des données et même de les encoder et décoder.	
Manipulation de base de données	PHP permet d'interroger tout type de base de donnée et de récupérer les tuples (résultat) d'une requête.	Impossible en Javascript.
Création et manipulation d'image	Manipulation et création d'image possible grâce à la librairie GD.	Javascript ne permet uniquement d'afficher des images.
Richesse	PHP dispose d'un nombre très important de fonctions qui se chiffre à plus de 2000.	Très petit nombre de fonctions comparé à PHP, tout au plus une centaine.
Avenir	Ajouts continuellement de nouvelles fonctionnalités malgré qu'elles ne soient pas toujours compatibles avec les anciennes versions.	Stable, l'ajout de nouvelles fonctionnalités est rare.
Récupérer le navigateur du client	Possible dans les deux langages, cependant Javascript permet d'avoir plus de précisions.	
Information sur le serveur	Il est tout à fait possible en PHP de récupérer une multitude d'informations concernant le serveur.	Impossible en Javascript.
Information sur le système du visiteur	En dehors du nom du système d'exploitation du visiteur, on ne peut rien obtenir.	En Javascript, il est possible d'établir la résolution de l'écran, ainsi que les plugins ... de l'utilisateur.
Réagir aux événements chez le client	Impossible en PHP, puisqu'il est exécuté côté serveur.	Javascript permet de réagir aux événements : (dé)chargement d'une page, validation de formulaire, clic, focus d'un champ de formulaire ...

La programmation par objet et Java

- Programmation par Objets
  - Unité logique : l'objet
  - Objet est défini par
    - un état
    - un comportement
    - une identité

- **Etat** : représenté par des attributs (variables) qui stockent les valeurs.
- **Comportement** : défini par des méthodes (procédures) qui modifient des états.
- **Identité** : permet de distinguer un objet d'un autre objet

La programmation par objet et Java

Les objets communiquent entre eux par des messages

- Un objet peut recevoir un message qui déclenche :
  - une méthode qui modifie son état

  et / ou
  - une méthode qui envoie un message à un autre objet

Héritage et polymorphisme

- **Définition**

  Technique offerte par les langages de programmation pour construire une classe à partir d'une (ou plusieurs) autre classe en partageant ses attributs et opérations.

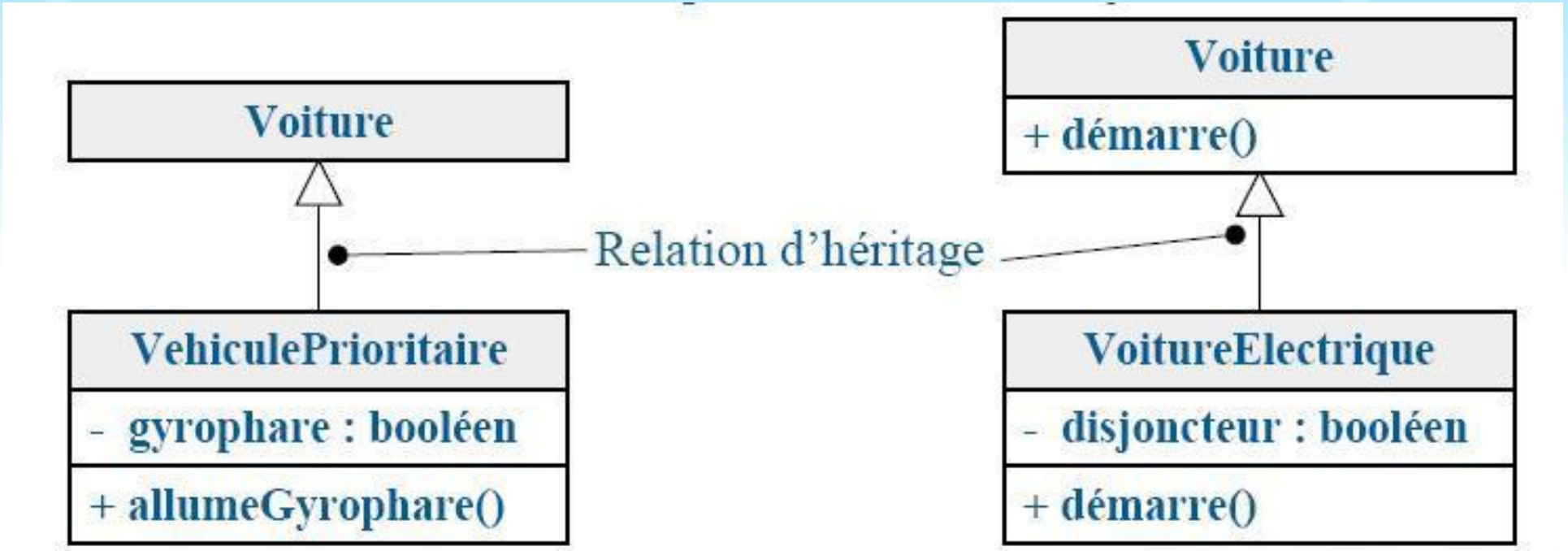
- **Intérêts**

  - **Spécialisation, enrichissement** : une nouvelle classe réutilise les attributs et les opérations d 'une classe en y ajoutant et/ou des opérations particulières à la nouvelle classe;

  - **Redéfinition** : une nouvelle classe redéfinit les attributs et opérations d'une classe de manière à en changer le sens et/ou le comportement pour le cas particulier défini par la nouvelle classe;

  - **Réutilisation** : évite de réécrire du code existant et parfois on ne possède pas les sources de la classe à hériter

Héritage et polymorphisme



Héritage et polymorphisme

- Définitions
  - La classe VehiculePrioritaire hérite de la classe Voiture
  - Voiture est la classe mère et VehiculePrioritaire la classe fille
  - Voiture est la super-classe de la classe VehiculePrioritaire
  - VehiculePrioritaire est une sous-classe de Voiture

- Attention !
  - Un objet de la classe VehiculePrioritaire ou VoitureElectrique est forcément un objet de la classe Voiture
  - Un objet de la classe Voiture n'est pas forcément un objet de la classe VehiculePrioritaire ou VoitureElectrique

Héritage et polymorphisme

- Autre notion ...
  - extends
  - super () et super
  - Redéfinition et surcharge
  - les classes abstraites
  - les interfaces

Héritage et polymorphisme

- **Les classes**
  - Elles sont complètement implémentées
  - Une autre classe peut en hériter

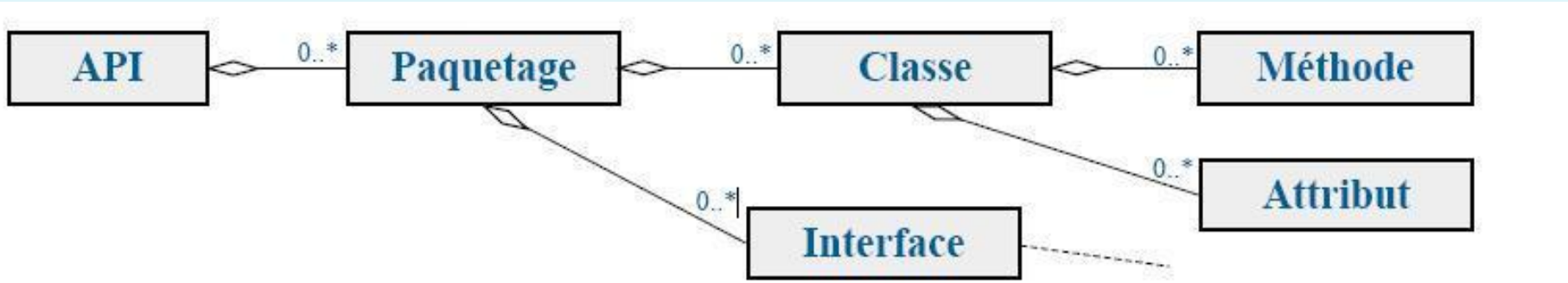
- **Les classes abstraites**
  - Elles sont partiellement implémentées
  - Une autre classe peut en hériter mais doit donner une implémentation aux méthodes abstraites
  - Elles ne peuvent pas être instanciées mais peuvent avoir un constructeur

- **Les interfaces**
  - Elles ne sont pas implémentées
  - Toute classe qui implémente une ou plusieurs interfaces doit implémenter toutes leurs méthodes (abstraites)

Les indispensables : package, collections et exception

- Le langage Java propose une définition très claire du mécanisme d'empaquetage qui permet de classer et de gérer les API externes

- Les API sont constituées :



- Un package est donc un groupe de classes associées à une fonctionnalité

- Exemples de packages
  - java.lang : rassemble les classes de base Java (Object, String, System, ...)
  - java.util : rassemble les classes utilitaires (Collections, Date, ...)
  - java.io : lecture et écriture
  - ...

Les indispensables : les exceptions

- **Définition**

Une exception est un signal qui indique que quelque chose d'exceptionnel (comme une erreur) s'est produit. Elle interrompt le flot d'exécution normal du programme

- **A quoi ça sert**
  - Gérer les erreurs est indispensable :
    - Mauvaise gestion peut avoir des conséquences catastrophiques (Ariane 5)
  - Mécanisme simple et lisible :
    - Regroupement du code réservé au traitement des erreurs (pas de « mélange » avec l'algorithme)
    - Possibilité de « récupérer » une erreur à plusieurs niveaux d'une application (propagation dans la pile des appels de méthodes)

- **Vocabulaire**
  - Lancer ou déclencher (throw) une exception consiste à signaler les erreurs
  - Capturer ou attraper (catch) une exception permet de traiter les erreurs

CONCLUSION

- HTML, Javascript, CSS ... côté client;
- ASP,PHP, Java, JSP, Servlet ... côté serveur;

- Questions ?