Web (Persistance)

Andrea G. B. Tettamanzi
Université de Nice Sophia Antipolis
Département Informatique
andrea.tettamanzi@unice.fr

CM - Séance 3

Cascade CSS
Introduction à PHP

Plan

- CSS
  - Assignation, cascade, héritage
- PHP
  - Avant PHP : CGI
  - Un peu d'histoire
  - Caractéristiques du langage

Assignation, cascade et héritage

- Le navigateur analyse le document (arbre d'éléments HTML)
- La navigateur calcule le style pour chaque éléments
  - Pour chaque propriété (de style) possible
  - En partant du haut de l'arbre (html, puis body, …)
  - Calcul en 3 étapes :
    - la valeur "spécifiée"
    - la valeur "calculée"
    - la valeur "réelle"

Valeurs spécifiées

1. Si la cascade (feuille(s) de style) donne une valeur, utiliser celle−ci ;
2. Autrement, si la propriété est héritée, utiliser la valeur de l'élément parent, qui est en général une valeur calculée ;
3. Autrement, utiliser la valeur initiale de la propriété. Cette valeur initiale est définie pour chaque propriété.

Cascade

- Les feuilles de style ont trois origines différentes :
  - Auteur,
  - Utilisateur
  - Navigateur
- L'auteur
  - produit des feuilles de style.
- L'utilisateur
  - peut être capable d'indiquer une information de style pour un document particulier. (style personnelle)
- Le navigateur
  - Style par défaut
- Ordre de priorité : Auteur, puis Utilisateur puis Navigateur
  - Parfois utilisateur prioritaire (cas d’handicap)

Cascade : identifier les styles à appliquer

- Trouver toutes les déclarations
  - Pour chaque propriété (selon média)
  - Pour chaque élément (sélecteurs)
- Trier selon l'origine
  - auteur > utilisateur > navigateur
- Trier selon spécificité des sélecteurs
  - les plus spécifiques prévalent
  - Les pseudo−éléments et les pseudo−classes sont considérés comme éléments et classes
- Trier selon ordre d'apparition
  - si deux règles ont les mêmes poids, origines et spécificités, c'est la dernière survenue qui l'emporte.
  - Règles issues de feuilles de style importées considérées comme étant survenues avant chacune de celles de la feuille de style elle−même.

Spécificité d'un sélecteur

- La spécifité d'un sélecteur est déterminée comme suit :
  - dans le sélecteur, compter le nombre d'attributs Id (= a) ;
  - puis celui des autres attributs et des pseudo−classes (= b) ;
  - et ensuite le nombre de noms des éléments (= c) ;
  - ignorer les pseudo−elements.
- Assembler les trois nombres a−b−c (dans un système de nombre avec une base étendue) pour obtenir la spécificité.
  - * {} /* a=0 b=0 c=0 −> specificité = 0 */
  - LI {} /* a=0 b=0 c=1 −> specificité = 1 */
  - UL LI {} /* a=0 b=0 c=2 −> specificité = 2 */
  - UL OL+LI {} /* a=0 b=0 c=3 −> specificité = 3 */
  - H1 + *[REL=up]{} /* a=0 b=1 c=1 −> specificité = 1-1 */
  - UL OL LI.red {} /* a=0 b=1 c=3 −> specificité = 1-3 */
  - LI.red[title] {} /* a=0 b=2 c=1 −> specificité = 2-1 */
  - #x34y {} /* a=1 b=0 c=0 −> specificité = 1-0-0 */
- Pour HTML, les valeurs de l'attribut "style" sont des règles de feuille de style. Ces règles n'ont pas de sélecteurs, mais dans l'optique du point 3 de l'algorithme de cascade, on considère qu'elles ont un sélecteur d'ID (spécificité : a=1, b=0, c=0). Et dans l'optique du point 4, on considère qu'elles surviennent après toutes les autres règles.

Media

- Adapter la présentation au média
  - Média = dispositif sur lequel le document est présenté
  - un écran, une feuille de papier, un synthétiseur de parole, un appareil braille, etc.
  - Certaines propriétés sont spécifiques à un média
- Association style – média
  - Préciser pour quel(s) média le style est valable
  - @import url("loudvoice.css") aural;
  - @media print { /* la feuille de style pour l'impression vient ici */ }
  - @media screen, print { BODY { line−height: 1.2 } }
  - <LINK rel="stylesheet" type="text/css" media="print, handheld" href="foo.css">

Catégories de médias

Media Types	**Media Groups**			
	continuous/paged	visual/aural/tactile	grid/ bitmap	interactive/static
aural	continuous	aural	N/A	both
braille	continuous	tactile	grid	both
emboss	paged	tactile	grid	both
handheld	both	visual	both	both
print	paged	visual	bitmap	static
projection	paged	visual	bitmap	static
screen	continuous	visual	bitmap	both
tty	continuous	visual	grid	both
tv	both	visual, aural	bitmap	both

Valeurs calculées

- Pas de calcul pour les valeurs absolues
  - les valeurs 'red' et '2mm' ne sont pas relatives à une autre valeur
- Calcul pour les valeurs relatives.
  - relatives (ex. les valeurs 'auto', '2em' et '12%' se rapportent à une autre valeur).
- Héritage de valeurs
  - Quand non spécifiées dans l’élément mais spécifiée dans un élément parent

Héritage

- Les éléments enfants (contenu) héritent des valeurs de leurs éléments parents (contenant)
  - Chacune des propriété définit si elle est héritée ou non.
  - color, oui ; background-image (image de fond) non
- Supposons un élément accentué (ici EM) dans un élément H1 :
  - <h1 style="color:red">Le titre est important !</h1>
  - Si aucune couleur n'est précisée pour l'élément EM, le mot accentué "est" héritera de la couleur de l'élément parent, ainsi l'élément H1 ayant une couleur rouge, EM le sera également.
- Une propriété de style définie pour HTML ou BODY sera héritée pour chaque élément du document.
  - Définition propre à un élément prioritaire sur l’héritage

Avant PHP : CGI

- Au commencement était la Common Gateway Interface (CGI)
- CGI fournit une interface entre un serveur Web et un programme (binaire exécutable) qui génère du contenu Web
- Le serveur Web passe au programme un certain nombre de variables d'environnement, dont REQUEST_METHOD
- Avec la méthode GET,
  - les paramètres de l'URL sont passés au programme dans la variable QUERY_STRING
- Avec la méthode POST, le contenu posté est passé par « stdin »
- Le « stdout » du programme est renvoyé au navigateur comme réponse (page HTML dynamique)

Histoire de PHP

- Créé en 1994 par Rasmus Lerdorf pour sa page Web
- PHP Tools (Personal Home Page Tools)
- Originairement un simple jeu de binaires CGI écrits en langage C
- De plus en plus de fonctionnalités sont ensuite ajoutées
- En juin 1995 Rasmus fourni le code source au grand publique
- ~~PHP~~ → FI (Forms Interpreter), syntaxe à la Perl
- Octobre 1995 : FI → PHP Construction Kit, syntaxe à la C
- Support cookies, BD, fonctions utilisateur
- En 1998, 1 % des serveurs Web avait PHP installé
- PHP 3.0 (PHP Hypertext Processor), avec Gutmans et Suraski
- En janvier 2013, 39 % (2,1 millions) des hôtes utilisait PHP.

Caractéristiques du langage

- PHP est un préprocesseur de pages HTML
- Typage dynamique
- Orienté objet
- Interprété côté serveur
- Nécessite :
  - D'un serveur Web avec support de PHP activé
  - Les fichier avec extension .php sont traités par PHP
  - Les fichiers .php sont comme des fichiers HTML avec en plus des balises « magiques »
  - Pour développer en locale : Apache + PHP + mySQL
- Génération de contenu HTML par « echo » (ou « print »)

Hello World en PHP

```
<html>
 <head>
  <title>Test PHP</title>
 </head>
 <body>
 <?php echo '<p>Bonjour le monde</p>'; ?>
 </body>
</html>
```

Syntaxe

- Entre le C/Java et le Shell Script
  - « ; » comme délimiteur
  - commentaires
    - comme en java/C/C++, entre les signes /* et */
    - comme en java/C/C++, en commençant une ligne par //
    - comme en shell Unix, avec #
- Balises d'ouverture et fermeture PHP
  - <?php …?> (recommandées)
  - <script language="php"> … </script>
  - <? … ?> (« balises courtes » déconseillées)
  - <% … %> (« balises ASP », également déconseillées

Styles de programmation

- On peut distinguer deux styles de programmation en PHP
  - Côté obscur : echo au fur et à mesure
  - Côté lumineux : calcul, mise en forme, puis echo
- Idée de fond : séparation des aspects
  - Code PHP (calcul)
  - Balises HTML (structure du document)

Exemple – style côté obscur

```
<HTML><HEAD>
<TITLE>HTML avec
PHP</TITLE></HEAD>
<BODY>
  <H1>HTML + PHP</H1>
  <p>Nous sommes le
  <?php echo date
        ("j/m/Y"); ?>
</p>
</BODY></HTML>
```

```
<HTML><HEAD>
<TITLE>HTML avec
PHP</TITLE></HEAD>
<BODY>
  <H1>HTML + PHP</H1>
  <p>Nous sommes le
     08/10/2014</p>
</BODY></HTML>
```

Exemple – style côté lumineux

```
<?php //calcul préalable
$date = "<p>Nous sommes
le ".date("j/m/Y").
"</p>"; ?>
<HTML><HEAD>
<TITLE>HTML avec
PHP</TITLE></HEAD>
<BODY>
  <H1>HTML + PHP</H1>
  <?php echo $date; ?>
</BODY></HTML>
```

```
<HTML><HEAD>
<TITLE>HTML avec
PHP</TITLE></HEAD>
<BODY>
  <H1>HTML + PHP</H1>
  <p>Nous sommes le
     08/10/2014</p>
</BODY></HTML>
```

Fonction « date »

- **string date (string format [, int timestamp])**
  - renvoie une date sous forme d'une chaîne, au format donné par la chaîne format. La date est fournie par le paramètre timestamp (un entier), sous la forme d'un timestamp. Par défaut, la date courante est utilisée.

```
<?php // Aujourd'hui, le 12 avril 2006, 10:16:18 am
$aujourdhui = date("F j, Y, g:i a"); // April 12, 2006, 10:16 am
$aujourdhui = date("m.d.y");         // 04.12.06
$aujourdhui = date("j, m, Y");       // 12, 04, 2006
$aujourdhui = date("Ymd");           // 20060412
$aujourdhui = date('\C\'\e\s\t\ \l\e\ jS \j\o\u\r\.');
                                     // C'est le 12th jour.
$aujourdhui = date("D M j G:i:s T Y"); // Wen Apr 12 10:16:18 Paris 2006
$aujourdhui = date("H:i:s");         // 10:16:18
// notation française
$aujourdhui = date("d/m/y");         // 12/04/06
$aujourdhui = date("d/m/Y");         // 12/04/2006
?>
```

Merci de votre attention