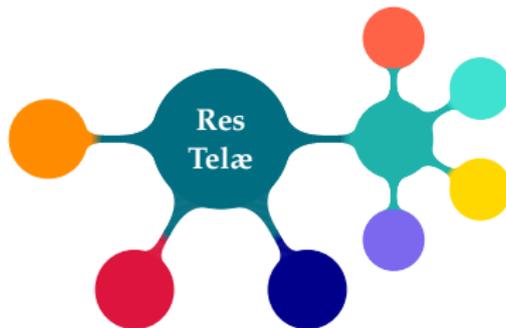


Zen, SASS, responsive design

Felip Manyer i Ballester



21 mai 2013

Plan

- 1 Créer un thème sous Drupal
 - Généralités
 - Concepts à mettre en œuvre
 - Typologie des thèmes Drupal
- 2 SASS et responsive
 - CSS
 - Responsive design
 - SASS
- 3 Zen
 - Installation et création d'un sous-thème
 - Structure des fichiers, grilles



- 1 Créer un thème sous Drupal
 - Généralités
 - Concepts à mettre en œuvre
 - Typologie des thèmes Drupal
- 2 SASS et responsive
 - CSS
 - Responsive design
 - SASS
- 3 Zen
 - Installation et création d'un sous-thème
 - Structure des fichiers, grilles

Généralités

- Drupal sépare clairement le rendu (présentation des pages web) de la logique. En toute fin de course, il charge un moteur de rendu (*theme engine*) (**PHP template**, Smarty, PHPTAL, etc.) de transformer le contenu sous une forme exploitable pour le client.
- Drupal peut présenter le contenu autrement qu'en utilisant HTML : texte brut (courriers électroniques), XML, JSON...
- La mission principale de Drupal est la **gestion de contenus**, mais un en sens désormais plus générique (services web, etc.)

Concepts à mettre en œuvre

Structuration

Balilage HTML :

- fichiers gabarits (*templates*)
- fonctions de rendu (*theme functions*)

Présentation

Définition du style avec un langage tel que CSS ou un sur-ensemble (SASS, LESS)

Surcharges

Surcharge des fonctions de rendu et de *preprocess* dans `template.php` (ou de CSS dans le `.info`)

Structuration : gabarits

system/page.tpl.php

```
<?php if ($breadcrumb): ?>  
  <div id="breadcrumb"><?php print $breadcrumb; ?></div>  
<?php endif; ?>  
<?php print $messages; ?>
```

zen/templates/page.tpl.php

```
<div id="content" class="column" role="main">  
  <?php print render($page['highlighted']); ?>  
  <?php print $breadcrumb; ?>  
  [...]  
</div><!-- /#content -->
```

Structuration : fonctions de rendu

Dans theme.inc

```
function theme_breadcrumb($variables) {  
  $breadcrumb = $variables['breadcrumb'];  
  $output = '<div class="breadcrumb">  
    . implode(' » ', $breadcrumb) . '</div>';  
  return $output;  
}
```

Dans template.php

```
function mon_joli_theme_breadcrumb($variables) {  
  $breadcrumb = $variables['breadcrumb'];  
  $output = '<div class="breadcrumb">  
    . implode(' > ', $breadcrumb) . '</div>';  
  return $output;  
}
```

Surcharges

Dans theme.inc

```
function template_process_page(&$variables) {  
  if (!isset($variables['breadcrumb'])) {  
    $variables['breadcrumb'] = theme('breadcrumb',  
      array('breadcrumb' => drupal_get_breadcrumb()));  
  }  
}
```

Dans template.php

```
function mon_joli_theme_preprocess_page(&$variables) {  
  $variables['ville'] = 'Montpellier';  
}
```

Types de thèmes

Trois possibilités au moment de choisir un thème :

- utiliser un thème déjà fait (Bartik, Marinelli, CTI Flex, Corporate Clean, Pixture Reloaded, etc.), éventuellement acquis à titre onéreux
- ~~repartir de zéro à la Linux From Scratch (get a life)~~
- utiliser un thème de base (Omega, Adaptive Theme, Fusion, **Zen**, etc.) : thème « blanc » comprenant les fondamentaux (gabarits, régions, CSS de mise en page, divers paramètres) et des outils permettant d'intégrer un design.

On crée généralement un **sous-thème** permettant de conserver la plupart des fonctionnalités dans le thème parent, et de surcharger à la demande.



- 1 Créer un thème sous Drupal
 - Généralités
 - Concepts à mettre en œuvre
 - Typologie des thèmes Drupal
- 2 SASS et responsive
 - CSS
 - Responsive design
 - SASS
- 3 Zen
 - Installation et création d'un sous-thème
 - Structure des fichiers, grilles

Cascading stylesheets : genèse

CSS (cascading stylesheets) est un langage de présentation (première séparation entre la structuration (HTML) et la présentation), largement connu de l'audience

6 août 1991 projet WorldWideWeb rendu public sur UseNet par Tim Berners-Lee

1992–1993 premiers essais « côté client » (simple personnalisation) : Viola, Harmony

1995–1996 les *cascading HTML stylesheets* d'Håkon Wium Lie s'imposent

Cascading stylesheets : les temps héroïques

- CSS 1 la guerre des navigateurs entre Internet Explorer et Netscape. Spécification relativement pauvre, implémentations concurrentes. La mise en page avec des tableaux perdure.
- CSS 2 ère IE5/6. Ambitions précipitées : gestion des fontes, adaptabilité aux différents supports de lecture : instruction @media, qui se limitera longtemps à screen et print
- CSS 2.1 se concentre sur ce qui est implémentable, repousse le reste à CSS 3
- CSS 3 beaucoup de changements, dont les **media queries**

CSS et plasticité du web : instructions @media

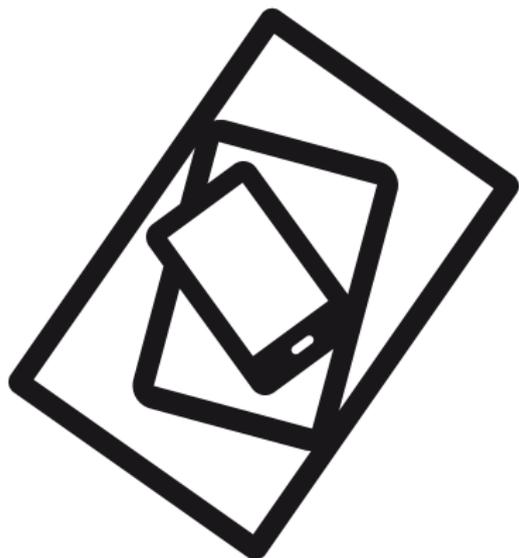
Types de media (CSS 2)

- braille
- embossed
- handheld
- **print**
- projection
- **screen**
- speech
- tty
- tv

Caractéristiques du media (2007)

color	nb bits/couleur
color-index	index des couleurs
device-aspect-ratio	ratio dispositif
device-height	hauteur dispositif
device-width	largeur dispositif
grid	largeur fixe
height	hauteur fenêtre
monochrome	monochrome
orientation	portrait/paysage
resolution	résolution
scan	pour les TV
width	largeur fenêtre

Pléthore de clients sur le Web : de 7 à 77"



- usages mobiles en pleine explosion : le trafic des appareils mobiles devrait **dépasser** celui des machines de bureau en **2015** (IDC)
- le « responsive design » (conception de sites web adaptatifs) est une solution **élégante** permettant d'échapper à l'enfer de la **duplication** de code

Responsive design : mise en œuvre

- on utilise une extension de l'instruction CSS 2.1 @media, permettant de définir des **points de coupure**

```
@media all and (min-width:480px)
@media all and (min-width:480px) and (max-width:959px)
@media all and (min-width:960px)
```

- ⇒ nécessite un navigateur supportant CSS 3, ou une couche intermédiaire de compatibilité en JavaScript (**shim**, shiv, modernizr) pour notre ami IE 8 (oubliez IE ≤ 7)
- préférez des **unités relatives** (pourcentages, em, ex), utilisez une **grille**, donnez une **largeur de 100 %** aux images
- utilisation des tableaux problématique

Responsive design : mobile first

- le mobile **n'est pas** un état dégradé de la version bureau : UX correcte (même si minimaliste) et ergonomique de base
- ⇒ étudier d'abord la version mobile. Ce sera le **style par défaut** dans la CSS **pour tous** les dispositifs : pas encore de point de coupure
- mobile : empilement sur une colonne (**pas de défilement horizontal**)
- ensuite rajouter des spécificités pour chaque dispositif identifié

C'est la doctrine *mobile first* (le mobile d'abord).

Responsive design : modules utiles

- Views Responsive Grid** Format de grille adaptatif pour Views. Voir aussi Views Fluid grid.
- Better Jump Menus** afficher un menu sous forme de liste déroulante
- Responsive Images [and Styles]** charger des images adaptées au dispositif. 7.x-2.x utilise uniquement des *media queries*. ☐ modules similaires
- Breakpoints** gestion avancée des points de coupure
- Picture** autre approche de la gestion des images

Syntactically awesome stylesheets (SASS)



- un langage de présentation se présentant comme un sur-ensemble de CSS 3 (au même titre que LESS)
- fichiers doivent être transformés en CSS avant utilisation
- syntaxe SCSS compatible avec CSS (transition en douceur)
- principales caractéristiques : variables, règles imbriquées, « mixins », héritage

SASS : variables

SCSS

```
$blue: #3bbfce;
$margin: 16px;

.content-navigation {
  border-color: $blue;
  color: darken($blue, 9%);
}

.border {
  padding: $margin / 2;
  margin: $margin / 2;
  border-color: $blue;
}
```

CSS

```
content-navigation {
  border-color: #3bbfce;
  color: #2b9eab;
}

.border {
  padding: 8px;
  margin: 8px;
  border-color: #3bbfce;
}
```

SASS : Imbrication (nesting)

SCSS

```
table.hl {  
  margin: 2em 0;  
  td.ln {  
    text-align: right;  
  }  
}  
li {  
  font: {  
    family: serif;  
    weight: bold;  
    size: 1.2em;  
  }  
}
```

CSS

```
table.hl {  
  margin: 2em 0;  
}  
table.hl td.ln {  
  text-align: right;  
}  
  
li {  
  font-family: serif;  
  font-weight: bold;  
  font-size: 1.2em;  
}
```

SASS : mixins

SCSS

```
@mixin table-base {
  th {
    text-align: center;
    font-weight: bold;
  }
  td, th {padding: 2px}
}

@mixin left($dist) {
  float: left;
  margin-left: $dist;
}

#data {
  @include left(10px);
  @include table-base;
}
```

CSS

```
#data {
  float: left;
  margin-left: 10px;
}

#data th {
  text-align: center;
  font-weight: bold;
}

#data td, #data th {
  padding: 2px;
}
```

SASS : héritage

SCSS

```
.error {  
  border: 1px #f00;  
  background: #fdd;  
}  
  
.error.intrusion {  
  font-size: 1.3em;  
  font-weight: bold;  
}  
  
.badError {  
  @extend .error;  
  border-width: 3px;  
}
```

CSS

```
.error, .badError {  
  border: 1px #f00;  
  background: #fdd;  
}  
  
.error.intrusion,  
.badError.intrusion {  
  font-size: 1.3em;  
  font-weight: bold;  
}  
  
.badError {  
  border-width: 3px;  
}
```

Compass



- une bibliothèque pour SASS, comprenant de nombreux *mixins*, un gestionnaire de *packages* pour obtenir des librairies additionnelles, un exécutable pour générer des CSS
- les meilleurs motifs réutilisables du Web
- facilite l'utilisation de CSS 3 : gère les *vendor prefixes*
- gestion automatisée des sprites CSS
- « rythmes typographiques »



LONGO CAP CERBERO

- 1 Créer un thème sous Drupal
 - Généralités
 - Concepts à mettre en œuvre
 - Typologie des thèmes Drupal
- 2 SASS et responsive
 - CSS
 - Responsive design
 - SASS
- 3 Zen
 - Installation et création d'un sous-thème
 - Structure des fichiers, grilles

Présentation de Zen : le thème

*Zen is a powerful, yet **simple**, HTML5 starting theme with a responsive, mobile-first grid design.*



- thème de base créé en 2006 par John Albin Wilkins
- 2 branches stables actuellement : 7.x-3.x et **7.x-5.x** (Zen 5).
- documentation en ligne fournie, et code commenté avec soin

Présentation de Zen : le créateur

- co-maintenu par John Albin Wilkins, de palantir.net (organisations publiques et non gouvernementales)
- développeur (front-end) clef dans la communauté : nombreux modules (Menu Block, Menu Position, Fences, sans compter le cœur), chef de la *Drupal 8 Mobile Initiative*
- co-auteur de *Drupal 7 Module Development*, gagnant du TPOTM (trivial patch of the month)

```
- // Check only if it's a template and not registered  
  by the theme or engine  
+ // Check only if it's a template and not registered  
  by the theme or engine.
```

Préparatifs

Installations des dépendances optionnelles sous Debian :

Installer SASS

```
sudo apt-get install rubygems  
sudo gem update  
sudo gem install sass
```

Installer Compass

```
sudo gem install compass
```

Installation de Zen et création d'un sous-thème

- télécharger Zen **5** (<http://drupal.org/project/zen>)
- le placer par exemple dans `sites/all/themes`
- dans le répertoire du site, taper `drush zen "Mon joli thème" --without-rtl`
- (ou bien suivre le README : copier STARTERKIT et renommer les fichiers et les fonctions)
- dans `mon_joli_theme.info`, choisir entre `css/layouts/fixed-width.css` (pas de responsive) et `css/layouts/responsive-sidebars.scss`

Structure des fichiers

`config.rb` divers paramètres, dont bascule entre dev et prod, et activation de FireSass

`.info` notamment choix du *layout* (fixe ou adaptatif)

`css/` fichiers CSS, pouvant être utilisés directement

`sass/` répertoire contenant les fichiers SASS, les CSS sont alors **générées** et **écrasées** et ne doivent pas être sous gestion de version

`templates/` fichiers contenant les gabarits (balisage). Ce sont des êtres sensibles. Les respecter.

`template.php` surcharges fonctions de *preprocess* et de *theme*

Outils

FireSass extension de Firebug permettant d'afficher les numéros de ligne du fichier SASS et non CSS

Ctrl-Maj-M mode de visualisation en responsive design de Firefox

compass `compass compile` pour générer les CSS, `compass clean` pour les supprimer, `compass watch` pour régénérer les fichiers concernés à chaque sauvegarde

Compass.app interface graphique pour Compass (\$10)

Zen Grids

- Zen Grids est une collection de *mixins* Compass permettant de gérer aisément les grilles à haut niveau (↘ « charge cognitive »), pouvant être utilisée en dehors de Zen (<http://zengrids.com/>)
- avantage de SASS : pas de classes « de présentation » : la CSS est calculée lors de sa génération

👉 démonstration : <http://zengrids.com/help/#example-1>



Conclusion

- CSS 3, responsive design : de nouvelles perspectives pour un Web ouvert, universel, à défendre face aux risques de fragmentation qui persistent
- Zen : la puissance du moteur de thème de Drupal mise en toute simplicité à disposition du *themer*, avec des outils désormais incontournables (SASS, Compass, grilles)
- sites non adaptatifs obsolètes. Intégration croissante du responsive dans nos outils (Drupal 8)
- propagez la bonne parole