Scandiweb

Base template development guide

General

This document from Scandiweb.com contains Magento theme development guides and theme development case study. It will basically cover two topics – Magento theme development Base Template "theory" and it's "practical" approach based on real educational project – 123 Pallets, plus FAQ for 123 Pallets project development. Before continuing, there's a thing you should know – even if you might not yet have any idea what's the base template is about, it's very important! It is the core of a successful project.

Base template

First – what is base template? We call base template styles and theme layouts sets that match requested site design globally. It means creating base layout structure, create CSS styles that are global, not one page specific. It is important, because if there are global elements that look the same through all the site, it is totally wrong to style these elements only for one page specific.

For example, intern NR.1 styles Home Page, intern NR.2 styles product view, there are same elements on Home Page and Product View (for example buttons, etc). Intern NR.1 was also supposed to do the base templating, but he created his styles only Home Page specific, so Intern NR.2 has to write the same CSS, to match design on Product View, and that is what creates a lot of bugs.

Second thing is that there is always a chance that some new pages will appear in project, so it is up to us to make pages that will (maybe) appear in future to look good. And that also depends on good base template.

Base templating also includes development environment setup for other developers, but we will discuss that later in this document.

All practical examples are based on the educational project called **123 Pallets**, so, if you do not have any materials on this project, ask PMs.

Design analysis

First thing to do when you have received design files and you are ready to start is – analysis. By that we mean going through all PSD files and write down, note all global styles you see and that will be applied globally.

Styles to look at

- Font sizes, colors and fonts for global elements – p, h1 – h6, a, etc;

- Page column widths, paddings, margins, etc;

- Buttons and button styles;

- Inputs, checkboxes and selects;

- Input field labels and em elements for required fields;

- Field validation message styles;

- Position of global messages and their style;

- Tables;

Fonts, font styles and colors

Now when design is analysed you can start with basic things as setting proper font, colors and font styles. You will notice that for example a element style in different places is slightly different, however base template is all about specifying the most common style.

Make sure you choose the most approprite style for global elements, it will save your development time later on. Dont worry if you cant find one specific, you can also specify styles for larger blocks like header or footer, or main content block, etc. Most important thing is not to specify one and the same style on every a or span, etc. element again and again.
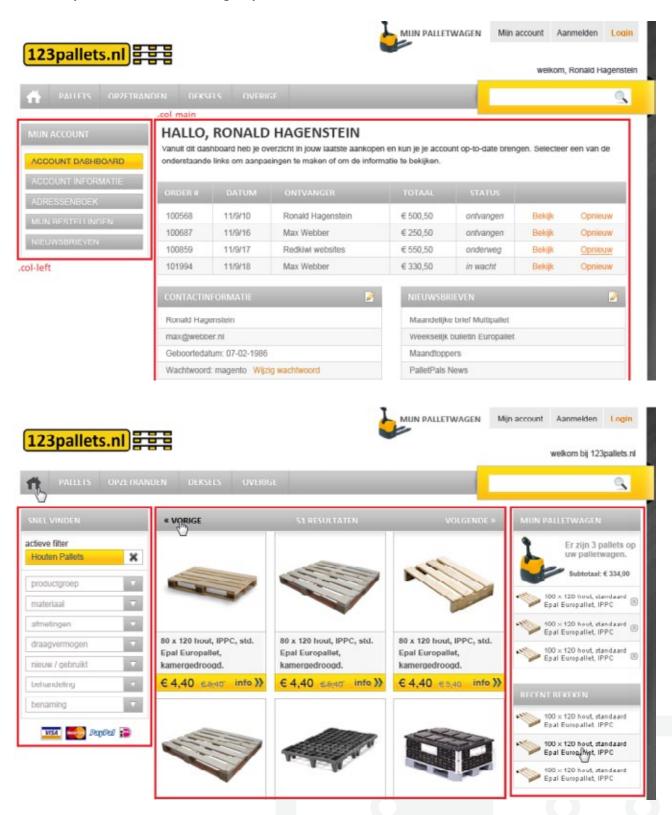

If on a project we use a custom font face (like Gill Sans for example) and implement it via @font-face. Then in styles css we should provide both weight - normal one and the one to use for bold marked text. Usually there are different font files used for this so in order to mek it work add something like this.
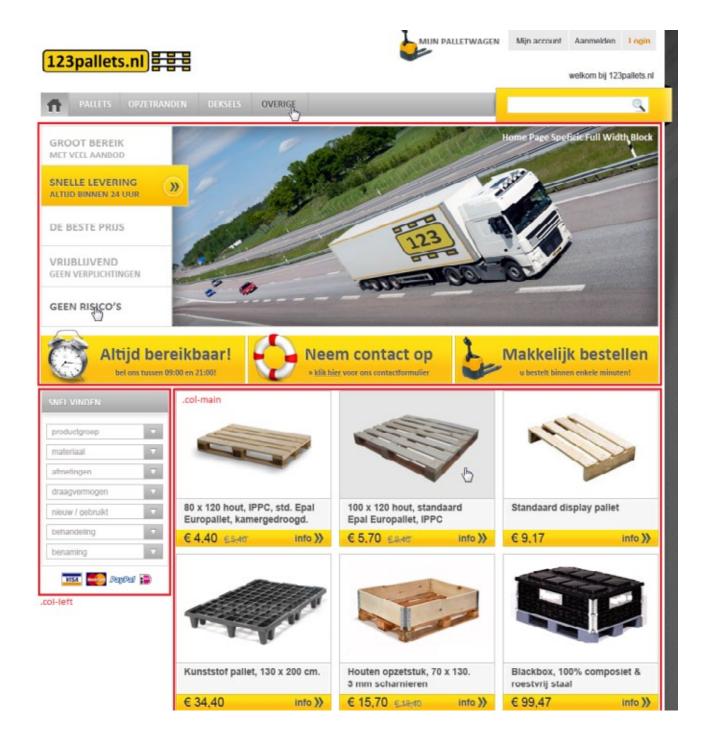
Here we specify font family which we will put in css and specify a font file which we will use for each of font weights we need.

```
@font-face {
font-family: 'GillSans';
font-style: normal;
font-weight: normal; // Normal font weight for all text
src: local('Gill Sans'), url('GillSans-Regular.ttf'), format('ttf');
}
@font-face {
font-family: 'GillSans';
font-style: normal;
font-weight: bold; // Bold font weight for text we will specify to be bold
src: local('Gill Sans Bold'), url('GillSans-Bold.ttf'), format('ttf');
}
body { font-family: "Gill Sans", GillSans, Arial, Helvetica, Verdana, sans-serif; }
```

Site column layouts

By default Magento has 4 types of page layouts : 1 column, 2 columns with right narrow column, 2 columns with left narrow column, 3 columns. Proper style should be applied to all of these layouts. .main-container block will always have current layout's class attached - col2-right-layout-main, etc.

www.scandiweb.com

One column layout was not available in design, however it's obvious that it will simply stretch to whole page width. For the home page you will create a new, custom page layout that will be based on 2 columns left layout and will have one full width block on the top of it.

Buttons

All buttons in site must be styled using classes and not css selectors using parents and nesting!

By that, we mean following style :

```css
@font-face {
font-family: 'GillSans';

/*Default button*/
button.button span { background:transparent url(../images/button_sprite) 100% -48px
no-repeat; ....}
button.button span span { background-position:0 0; padding:0 20px 0 30px; }
button.button:hover span { background-position:100% -144px; }
button.button:hover span span { background-position:0 -96px; }
/*Small blue button*/
button.button.btn-small-blue span { height:28px; background-position:100% -220px; ...}
button.button.btn-small-blue span span { background-position:0 -192px; padding:0 10px
0 15px; }
button.button.btn-small-blue:hover span { background-position:100% -276px; }
button.button.btn-small-blue:hover span span { background-position:0 -248px; }
/*Small white button with border*/
button.button.btn-small-white-border span { height:28px; background-position:100%
-332px; .... }
button.button.btn-small-white-border span span { background-position:0 -304px;}
button.button.btn-small-white-border:hover span { background-position:100% -388px; }
button.button.btn-small-white-border:hover span span { background-position:0 -360px; }
```

NOT
```css
.top-cart .actions button { float:none; ... }
.top-cart .actions button.button span,
.top-cart .actions button.button span span {
background-image:url(../images/bkg_btn-onbg.gif);
}
```

So, if someone needs specific buttons style, he/she just adds css class to button and it's done!

Button types on 123 Pallets:

bestel pallets ›› might!! be used as default for all buttons

e-mail@voorbeeld.nl zend specific for newsletter

€ 4,40 €5,40 info ›› specific on product list

www.scandiweb.com

All buttons should have hover effect - style change when user mover mouse pointer over the button. If styles are specified in given design, make it as it says, if not usually making button darker or lighter on hover is enough (Ask PM about style if not sure). In 123 Pallets should make text and style arrows darker on hover.

Input fields, checkboxes, radio buttons and selects

Check global style for input fields, checkboxes, radio buttons and selects. Checkboxes, radio buttons and selects can not be styled using CSS, JavaScript plugins are used in such cases. If no design for these elements is given, leave default.

Input fields, checkboxes, radio buttons and selects

Check global style for input fields, checkboxes, radio buttons and selects. Checkboxes, radio buttons and selects can not be styled using CSS, JavaScript plugins are used in such cases. If no design for these elements is given, leave default.

www.scandiweb.com

Global messages

Global messages are messages that appear after user submits form, buys product, etc. and page reloads. Usually at the top of main content.

You can easily call these messages any time by adding following code to any phtml file that is used in site and reloading page:

```php
<?php
Mage::getSingleton('catalog/session')->addSuccess("Mmmmm, messagessss! ");
Mage::getSingleton('customer/session')->addError("Mmmmm, messagessss! ");
Mage::getSingleton('checkout/session')->addNotice("Mmmmm, messagessss! ");
?>
```

This snippet adds all 3 types of global messages : error, success, notice. Once it appears, just style it.

Tables

Make sure you style tables globally, so they all by default look similar. Of course, table like Shopping Cart is specific and will have it's own style.

www.scandiweb.com

Sidebar blocks

Usually all sidebar blocks have similar style so style the all together at once.

Overall tips

1. Think! Analyse! Make things global!

2. Write CSS in Magento style.

3. Check design using Pixel Perfect for FireBug extension or similar tools.

4. Use CSS Reloader extension for browser

5. Validate your HTML - W3C Validator

6. Configure your Code Editor - 4 spaces instead tab, remove trailing whitespaces on save

7. Zend Code Standarts - http://framework.zend.com/manual/en/coding-standard.php-file-formatting.html

www.scandiweb.com