Intoduction to Sass and Compass

D.R.Y = Don't Repeat Yourself.

Sass Introduction

- Sass is a superset of CSS.
- The code structure is **more clear** and make CSS *more like a programming language *.
- The file extension can be ***.scss** or **.sass *.
- The difference is that **in *.sass** you **don't** have to write *braces ({, }) and semicolon (;) *.
- No browser can interprets Sass formats.
  You have to **compile Sass to CSS** manually (sass-convert) or automatically (sass --watch).
- Built-in in Rails 3.1.

Nesting

- CSS

```
#profile-edit-icon .description * {
    vertical-align: middle;
}
#profile-edit-icon .tip-link {
    background: url(/index/my/ico_bulb.png) 0 0 no-repeat;
    display: inline-block;
    display: -moz-inline-box;
    width: 21px;
    height: 21px;
}
#profile-edit-icon .select {
    float: left;
    width: 24.9%;
}
```

- Sass (*.scss) - Nesting structure.

```
#profile-edit-icon {
    .description * {
        vertical-align: middle;
    }
    .tip-link {
        background: url(/index/my/ico_bulb.png) 0 0 no-repeat;
        display: inline-block;
        display: -moz-inline-box;
        width: 21px;
        height: 21px;
    }
    .select {
        float: left;
        width: 24.9%;
    }
}
```

- Sass (*.sass) - Less lines. CSS Coder must change their behavior.

```
#profile-edit-icon
    .description *
        vertical-align: middle
    .tip-link
        background: url(/index/my/ico_bulb.png) 0 0 no-repeat
        display: inline-block
        display: -moz-inline-box
        width: 21px
        height: 21px
    .select
        float: left
        width: 24.9%
```

Mixin

Define (@mixin) a group of CSS attributes and reuse it (@include).

1. Define (@mixin)

```
@mixin rounded($radius: 1px) {
   border-radius: $radius;
   -moz-border-radius: $radius;
   -webkit-border-radius: $radius;
   *behavior: url("/static/css3-pie/pie.htc");
```

```
6}
```

2. Use (@include)

```
1.view-content {
2    background: #f6eccd;
3    @include rounded(4px);
4}
5.crop {
6    border: solid 1px #ebe2c6;
7    @include rounded();
8}
```

3. Output

```
 1.view-content {
 2    background: #f6eccd;
 3    border-radius: 4px;
 4    -moz-border-radius: 4px;
 5    -webkit-border-radius: 4px;
 6    *behavior: url("/static/css3-pie/pie.htc");
 7}
 8.crop {
 9    border: solid 1px #ebe2c6;
10    border-radius: 1px;
11    -moz-border-radius: 1px;
12    -webkit-border-radius: 1px;
13    *behavior: url("/static/css3-pie/pie.htc");
14}
```

Extend

- Original CSS

```
 1.x2a {
 2    background: transparent url(/index/bar/bg_layout_frame_t.png) repeat-x 0 0;
 3    height: 24px;
 4    left: 0;
 5    position: absolute;
 6    top: 0;
 7    width: 99%;
 8}
 9.x3a {
10    background: transparent url(/index/bar/bg_layout_frame_b.png) repeat-x 0 0;
11    bottom: 0;
12    height: 24px;
13    left: 0;
14    position: absolute;
15    width: 99%;
16}
```

- After Using @extend

```
 1.x2a {
 2    background: transparent url(/index/bar/bg_layout_frame_t.png) repeat-x 0 0;
 3    height: 24px;
 4    left: 0;
 5    position: absolute;
 6    top: 0;
 7    width: 99%;
 8}
 9.x3a {
10    @extend .x2a;
11    background-image:url (/index/bar/bg_layout_frame_b.png);
12    bottom: 0;
13    top: default;
14}
```

Variable

```
 1$misc-icon: "/index/space/sp_ico_misc.png";
 2$css3pie: "/static/css3-pie/pie.htc";
 3
 4.mod-main {
 5    .msg span.exclamation-icon {
 6        background-image: url($misc-icon);
 7        background-position: 0 -24px;
 8    }
 9    .msg .close-link {
10        background-image: url($misc-icon);
11    }
12    .storage .share-icon {
13        background-image: url($misc-icon);
14    }
15    .notice .hr {
16        -moz-border-radius: 3px;
17        -khtml-border-radius: 3px;
18        -webkit-border-radius: 3px;
19        border-radius: 3px;
20        *behavior: url($css3pie);
```

```
21    }
22}
```

Other Goodies

- @import

```
1@import "_hp_access.scss";
```

```
1@import "_hp_access.css";
```

  - Include all content of "_hp_access.css" after compiling.

- Math calculation.
  - Example - Provides different column width when the amount of column is different.

```
1@mixin cell-width($total-width, $num-columns, $grid-gutter) {
2    width: $total-width / $num-columns + $grid-gutter;
3}
4
5.cell {
6    @include cell-width(960px, 10, 10px);
7}
```

- Color functions - hsl, hsla, darken, saturate, complement, grayscale, mix, ...
- @media

```
 1@media screen {
 2    .sidebar {
 3        width: 300px;
 4        float: left;
 5        @media (max-width: 480px) {
 6            width: auto;
 7            float: none;
 8        }
 9    }
10}
```

- @if, @else, @for, @while, @each, @function - Useful when your code can be reusable.

Command-Line

- Convert existing CSS to Sass.

```
sass-convert a.css a.scss
```

- Manually compile Sass to CSS.

```
sass compile -t expanded a.scss a.css
```

- Automatically compile Sass to CSS.

```
sass --watch a.scss:a.css
```

Sass References

- http://sass-lang.com/
- hlb's slideshare - Sass & Compass / W3CTech Shanghai

Compass Introduction

- Compass is *Charityware *.
- Compass is a framework built up on Sass.
- Sass makes CSS syntax become more flexible.

  However, when it comes to re-usability, you still have to reinvent the wheel.

  Compass provides **real world CSS libraries, frameworks, or tools** you may need:
  - Blueprint - CSS Framework.
  - Page Layout
  - Page Reset

- CSS3.
  - border-radius, box-shadow, text-shadow, box-flex, opacity, transform...

```
1@import "compass";
2#foo {
3    @include border-radius(4px);
4    @include opacity(0.5);
5    @include single-box-shadow(#aaa, 0px, 0px, 5px, false, true);
6}
```

  - CSS3 Pie

```
1@import "compass";
2@import "compass/css3/pie";
3$pie-behavior: url(/static/css3-pie/pie.htc);
4#foo {
5    @include border-radius(4px);
6    @include pie;
7}
```

- Inline Block

```
1@import "compass";
2#foo {
3    @include inline-block;
4}
```

- text-ellipsis

```
1@import "compass/typography/text/ellipsis";
2$use-mozilla-ellipsis-binding: true;
3.ellipsis {
4    @include ellipsis;
5}
```

- word-wrap
- nowrap
- clearfix

```
1@import "compass";
2#foo {
3    @include clearfix;
4}
```

- pie-clearfix

```
1@import "compass";
2#foo {
3    @include pie-clearfix;
4}
```

- Data URI

```
1@import "compass";
2#foo {
3    background-image: inline-image(logo.png);
4    *background-image: url(/lib/compass/images/logo.png);
5}
```

- Spriting
  - FamFamFam silk icons exists at `images/silk` folder.
  - Suppose we need to make user_*.png as sprites in our CSS.

```
@import "compass";
@import "compass/utilities/sprites";
@import "silk/*.png";
#foo {
  @include silk-sprite(user_green);
}
```

- `silk` is the magic keyword.

- Unlike Sass, Compass needs to have **a project directory** with **a project configuration file** in it.
- You always need to **compile a project** instead of a file.

```
1$ compass compile <path/to/project>
```

Reference

- Compass Core Framework

Conclusion

- We **MUST have Compass** in miiiCasa. Only Sass is not enough.