



Open Apereo 2014

Innovate - Incubate - Implement

Meet Sakai's **MORPHEUS**

NYU mar22@nyu.edu [@alien_resident](https://twitter.com/alien_resident)



Sakai, Meet **MORPHEUS**

Mobile Optimized Responsive Portal for
Higher Education Using Sass

About This Presentation

I am using `reveal.js` for my slides. They're on github
<https://github.com/alienresident/Sakai-Meet-Morpheus>



Mark Reilly
User Experience Designer
New York University.



Sakai 10

/portal

/reference

/portal/trunk/portal-render-engine-

impl/pack/src/webapp/vm/morpheus/

/reference/trunk/library/src/webapp/skin/morpheus

Now-*ish*

Templates we're committed in February
Sass is under active development

Why a new *another* new skin?

Usability Testing

- Pilot
- Faculty
- Never used Sakai
- Remote Testing

Usability Issues

- Place Finding
- Navigation
- Messaging
- Hiding & Showing Items

Usability Issues *Solutions*

- Quick Fixes
- Long Term Fixes
 - New Icons
 - iframes: Fix Mental Model
 - Group Navigation Items by Function
 - Better Messaging

User Interface

- Organized
- Modular
- Reusable

Need to use a CSS preprocessor

Sass was my choice

Responsive Design

- Fluid Images
- Fluid Grids
- Media Queries

Responsive Design *Issues*

- Navigation
- HTML Source Order
- Div-itis: Convolutd Markup
- Lack of Classes
- Upgrading?

Digging into MORPHEUS



What it is!

- New VM Templates
- New CSS (compiled from Sass)

New VM Templates

Apache Velocity templates

- Converted to HTML 5
- Simplified markup
- Added new Semantic tags

HTML 5

```
<!DOCTYPE html>
<html lang="en-US">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
```

HTML 5 *contd*

IE 8 HTML 5 shiv

```
<!--[if lt IE 9]>  
  <script src="//cdnjs.cloudflare.com/ajax/libs/html5shiv/3.7/html5shiv.min.js"></script>  
<![endif]-->
```

HTML 5 *contd*

IE 8 CSS files

```
<!--[if (lt IE 9)]>
  <link href="/library/skin/morpheus-default/portal-ie.css" rel="stylesheet">
<![endif]-->
<!--[if (gte IE 9)|(IEMobile)|(!IE)]><!-->
  <link href="/library/skin/morpheus-default/portal.css" rel="stylesheet">
<!--<![endif]-->
```

HTML 5 *contd*

Morpheus Script in skin

```
<script src="${pageSkinRepo}/${pageSkin}/js/morpheus.scripts.js"></script>
```

Add your own javascript to your skin!

```
<script src="library/morpheus-default/js/morpheus.scripts.js"></script>
```

HTML 5 *contd*

New Snippets folder

Discrete snippets of markup that are conditionally called throughout the templates.

- header
- login form
- role switcher
- tool body
- etc

HTML 5 *contd*

Adding Classes while leaving IDs

- adding multiple classes
 - reusable and modular
 - single responsibility theory
 - classes for layout, states, and js hooks
- Leave IDs for legacy javascript

SASS

Syntactically Awesome StyleSheets

- Variables
- Mixins
- Extends
- Nesting
- @import & Partials

learn more at sass-lang.com/guide

SASS contd

Variables

```
$instution-color: #fc3;  
  
a {  
  color: $instution-color;  
}  
  
nav {  
  background-color: $instution-color;  
}
```

SASS *contd*

Mixins

```
@mixin default-type-size {  
  font-size: 16px;  
  line-height: 1.5em;  
}  
p {  
  @include default-type-size;  
}  
footer {  
  @include default-type-size;  
}
```

```
p {  
  font-size: 16px;  
  line-height: 1.5em;  
}  
footer {  
  font-size: 16px;  
  line-height: 1.5em;  
}
```

SASS *contd*

Mixins (*arguments*)

```
@mixin default-type-size($color) {  
  font-size: 16px;  
  color: $color;  
}  
p {  
  @include default-type-size(#333);  
}  
footer {  
  @include default-type-size(#eee);  
}
```

```
p {  
  font-size: 16px;  
  color: #333333;  
}  
footer {  
  font-size: 16px;  
  color: #eeeeee;  
}
```

SASS *contd*

Mixins (*more arguments*)

```
@mixin type-size($size, $color) {  
  font-size: $size;  
  color: $color;  
}  
p {  
  @include type-size(16px, #333);  
}  
footer {  
  @include type-size(14px, #eee);  
}
```

```
p {  
  font-size: 16px;  
  color: #333333;  
}  
footer {  
  font-size: 14px;  
  color: #eeeeee;  
}
```

SASS *contd*

Extends

```
%default-type-size {  
  font-size: 16px;  
  line-height: 1.5em;  
}  
p {  
  @extend %default-type-size;  
}  
footer {  
  @extend %default-type-size;  
}
```

```
p, footer {  
  font-size: 16px;  
  line-height: 1.5em;  
}
```

SASS *contd*

Extends *contd*

```
%default-type-size {  
  font-size: 16px;  
  line-height: 1.5em;  
}  
p {  
  @extend %default-type-size;  
}  
footer {  
  @extend %default-type-size;  
  color: #eeeeee;  
}
```

```
p, footer {  
  font-size: 16px;  
  line-height: 1.5em;  
}  
footer {  
  color: #eeeeee;  
}
```

SASS *contd*

Nesting

```
nav {  
  ul {  
    margin: 0;  
    padding: 0;  
    list-style: none;  
  
    li {  
      display: inline-block;  
    }  
  }  
}
```

```
nav ul {  
  margin: 0;  
  padding: 0;  
  list-style: none;  
}  
  
nav ul li {  
  display: inline-block;  
}
```

SASS *contd*

@import & Partials

```
// _reset.scss
html,
body,
ul,
ol {
  margin: 0;
  padding: 0;
}
```

```
// base.scss
@import 'reset';

body {
  font-size: 100% Helvetica, sans-serif;
  background-color: #efefef;
}
```

```
/* base.css */
html, body, ul, ol {
  margin: 0;
  padding: 0;
}
body {
  font-size: 100% Helvetica, sans-serif;
  background-color: #efefef;
}
```


SASS contd

Sass while powerful can be misused. You can create brittle, inflexible, and unmaintainable CSS with Sass as with vanilla CSS. What you need is an architecture that is modular and scalable.

SMACSS

Scalable and Modular Architecture for CSS



By Jonathan Snook a web developer and designer formerly at Yahoo! He worked on the redesign of Yahoo! mail
learn more at smacss.com

SMACSS

At the very core of SMACSS is categorization

There are five types of categories:

1. Base
2. Layout
3. Module
4. State
5. Theme

SMACSS

Base

Base rules are the defaults.

```
html, body, form {  
  margin: 0;  
  padding: 0;  
}  
  
input[type="text"] {  
  border: 1px solid #999;  
}  
  
a {  
  color: #039;  
}  
  
a:hover {  
  color: #03C;  
}
```

SMACSS

Layout

Layout rules divide the page into sections. Layouts hold one or more modules together.

```
#article {  
  width: 80%;  
  float: left;  
}  
  
#sidebar {  
  width: 20%;  
  float: right;  
}  
  
.l-fixed #article {  
  width: 600px;  
}  
  
.l-fixed #sidebar {  
  width: 200px;  
}
```

SMACSS

Module

Modules are the reusable, modular parts of our design. They are the callouts, the sidebar sections, the product lists and so on.

```
.pod {  
  width: 100%;  
  background: #ddd;  
}  
  
.pod input[type=text] {  
  width: 50%;  
  border: 1px solid #666;  
}
```

SMACSS

Module *contd*

```
.pod {  
  width: 100%;  
  background: #ddd;  
}  
  
.pod input[type=text] {  
  width: 50%;  
  border: 1px solid #666;  
}  
  
.pod-callout {  
  width: 200px;  
}  
  
.pod-callout input[type=text] {  
  width: 180px;  
}
```

```
<div class="pod pod-callout"> ... </div>
```

SMACSS

State

State rules are ways to describe how our modules or layouts will look when in a particular state. Is it hidden or expanded? Is it active or inactive?

```
.tab {  
  background-color: purple;  
  color: white;  
}  
  
.is-tab-active {  
  background-color: white;  
  color: black;  
}
```


SMACSS

Theme

Theme rules are similar to state rules in that they describe how modules or layouts might look.

```
/* in module-name.css */  
.mod {  
  border: 1px solid;  
}  
  
/* in theme.css */  
.mod {  
  border-color: blue;  
}
```

SMACSS & SASS

In MORPHEUS we are combining the best of SMACSS & SASS
To get Scalable and Modular Architecture for SASS *or* SMASASS
or SMACASS

~~To get Scalable and Modular Architecture for SASS *or* SMASASS
or SMACASS~~

Let's just stick to MORPHEUS!

MORPHEUS

Directory Structure

`/reference/trunk/library/src/webapp/skin/`

```
morpheus-default/  
morpheus-examp-u/  
morpheus-master/  
morpheus-rtl/
```

```
morpheus-default/ // generated css  
morpheus-examp-u/ // generated css  
morpheus-master/ // Sass files and compile script  
morpheus-rtl/ // generated css
```

MORPHEUS

Modules

```
@import "modules/MODULE_NAME/_MODULE_NAME";
```

```
morpheus-master/sass/modules/MODULE_NAME/_MODULE_NAME.scss
```

A more realistic example:

```
@import "modules/alerts/_alerts";
```

```
morpheus-master/sass/modules/alerts/_alerts.scss
```

MORPHEUS

Modules

Optionally there might import additional files more files

```
@import "_MODULE_NAME_variables"; // default variables that can be overridde  
@import "_MODULE_NAME_mixins"; // mixins for that module only  
@import "_MODULE_NAME_extendables"; // extendables for that module only
```

In our previous example it would pull in these files

```
morpheus-master/sass/modules/alerts/_alerts_variables.scss  
morpheus-master/sass/modules/alerts/_alerts_mixins.scss  
morpheus-master/sass/modules/alerts/_alerts_extendables.scss
```

MORPHEUS

Sakai Properties

```
## Sakai 10 sakai.properties  
portal.templates=morpheus  
portal.neoprefix=  
skin.default=morpheus-default
```

MORPHEUS

Custom themes

Morpheus will ship with 3 themes.

1. morpheus-default
2. morpheus-examp-u
3. morpheus-rtl

`morpheus-master/sass/themes/`

```
_morpheus-default.scss  
_morpheus-examp-u.scss  
_morpheus-rtl.scss
```

```
_morpheus-nyu.scss
```

MORPHEUS

Custom themes

Examples of Basic Customizations.

1. Institution Colors
2. Logos
3. Button Colors

```
../sass/themes/_morpheus-default.scss
```

```
$primary-color: #009dce;  
$black-value: #333;
```

```
$masthead-image: "my-U-logo.png";
```

```
$button-gradient-stop-1: $primary-color;  
$button-gradient-stop-2: shade($button-gradient-stop-1, 3%);
```


MORPHEUS

Custom themes

Example of More Advanced Customizations.

1. Webfonts
2. Icon Fonts

```
$fontstack: 'Helvetica Neue', Helvetica, Arial, sans-serif;  
$webfont-name: 'Open Sans';
```

```
$icon-font-name: 'Sakai-Icon-Font';  
$tool-icons: (sakai-motd '\f015') (sakai-schedule '\f073') ... ;
```

Custom Icon font based off of **Font-Awesome** using **icomoon**. All the portal assets including the font SVGs will be on github github.com/alienresident/sakai-portal-assets

MORPHEUS

Custom themes

Example of More Advanced Customizations *contd*

3. Custom Modules

```
@import "modules/CUSTOM_MODULE_NAME/_CUSTOM_MODULE_NAME";
```

```
../sass/modules/CUSTOM_MODULE_NAME/_CUSTOM_MODULE_NAME.scss
```

MORPHEUS

Styleguides

Each skin comes with a flat HTML file that acts as a styleguide.

This is also a design pattern for sakai elements.

This is where you can see your styled elements on one page.

Here's an example from [NYU Classes](#).

<http://i5.nyu.edu/~mar22/nyuclasses/styleguide/>

I'll be adding more Sakai components to these.

MORPHEUS

Color Schemer

One each styleguide comes with a generated color guide. These are created using one base color and uses color theory models to create a color model. Let's look at [color scheme designer](#)

```
$primary-color: #009dce;
$black-value: #333;
$cs-primary:    $primary-color;
$cs-scheme:     triad;
// Options: mono, complement, triad, tetrad, analogic, accented-analogic
$cs-hue-offset: 60;

// $colour-stack-amounts: 25%, 50%, 75%, 85%, 90%;

$black:    tint-stack($black-value);
$primary:  tint-stack(cs-primary());
$secondary: tint-stack(cs-secondary());
$tertiary: tint-stack(cs-tertiary());
$quadrory: tint-stack(cs-quadrory());
```

Conclusion

Road Map: What's next?

1. Finish adding modules from NYU Classes code base
2. Responsive menus for devices
3. Responsive typography
4. Flesh out examples skins
5. User Testing
6. Documentation

Sakai 11

1. Tools...
2. Pattern Library a mini 'bootstrap' for Sakai

Conclusion

Thanks to Gonzalo Silverio and the NYU Classes team at NYU

Questions?

Contributors?