



CSS3 meets GWT with GSS

Closure-styleSheets in GWT

Daniel Kurka
Software Engineer

Julien Dramaix
Software Engineer



Google™



**Daniel
Kurka**

invented mgwt

mgwt



SPEAKER



Google™  12

Julien
Dramaix



What is this about?

- Introduction to Closure-stylesheets (GSS)
 - Variables
 - Functions
 - Mixins
 - Conditionals
 - Linting
 - RTL flipping
- Integration with GWT
- Timeline
- Demo





DRY



CSS is missing things

- variables
 - repeating tokens in CSS files
- functions
 - no way to express derived parameters
- macros
 - duplicating code common code blocks



Closure-stylesheets

- are an extension to CSS
- write GSS, compile to CSS
- adding variables, conditionals, mixing
- support minification, linting, RTL flipping, renaming



Examples



Variables

```
@def BG_COLOR          rgb(235, 239, 249);  
  
@def DIALOG_BORDER_COLOR  rgb(107, 144, 218);  
@def DIALOG_BG_COLOR    BG_COLOR;  
  
body {  
  background-color: BG_COLOR;  
}  
  
.dialog {  
  background-color: DIALOG_BG_COLOR;  
  border: 1px solid DIALOG_BORDER_COLOR;  
}
```



compiles to

```
body {  
  background-color: #ebeff9;  
}  
.dialog {  
  background-color: #ebeff9;  
  border: 1px solid #6b90da;  
}
```



Functions

- Several arithmetic functions
 - add()
 - sub()
 - mult()
 - divide()
 - min()
 - max()
- Variable number of arguments
- Arguments may be purely numeric or need to have the same CSS unit
- multiply and divide only allow a CSS unit for the first parameter
- not as flexible as CSS calc, but yields more maintainable stylesheets



Functions

```
@def LEFT_HAND_NAV_WIDTH      180px;  
@def LEFT_HAND_NAV_PADDING    3px;  
  
.left_hand_nav {  
  position: absolute;  
  width: LEFT_HAND_NAV_WIDTH;  
  padding: LEFT_HAND_NAV_PADDING;  
}  
  
.content {  
  position: absolute;  
  margin-left: add(LEFT_HAND_NAV_PADDING, /* padding left */  
                  LEFT_HAND_NAV_WIDTH,  
                  LEFT_HAND_NAV_PADDING); /* padding right */  
}
```



compiles to

```
.left_hand_nav {  
    position: absolute;  
    width: 180px;  
    padding: 3px;  
}  
  
.content {  
    position: absolute;  
    margin-left: 186px;  
}
```



Functions

- `blendColorsHsb(startColor, endColor)` blends using HSB values
- `blendColorsRgb(startColor, endColor)` blends using RGB values
- `makeMutedColor(backgroundColor, foregroundColor [, saturationLoss])`
- `addHsbToCssColor(baseColor, hueToAdd, saturationToAdd, brightnessToAdd)`
- `makeContrastingColor(color, similarityIndex)`
- `adjustBrightness(color, brightness)`
-



Functions - selectFrom

```
/* Implies MYDEF = FOO ? BAR : BAZ; */  
@def MYDEF selectFrom(FOO, BAR, BAZ);  
  
/* @def FOO true; to have the effect of @def MYDEF = BAR. */
```



Providing your own functions

- Implement GssFunctionMapProvider
- pass “--gss-function-map-provider” to the compiler



Mixins

- reuse a list of parameterized declarations
- declaration
- a mixing can be used anywhere instead of a declaration



Mixins

```
@defmixin size(WIDTH, HEIGHT) {  
  width: WIDTH;  
  height: HEIGHT;  
}  
  
.logo {  
  @mixin size(150px, 55px);  
  background-image: url('http://www.google.com/images/logo_sm.gif');  
}
```



compiles to

```
.logo {  
  width: 150px;  
  height: 55px;  
  background-image: url('http://www.google.com/images/logo_sm.gif');  
}
```



Mixins - for browser specific code

```
@defmixin gradient(POS, HSL1, HSL2, HSL3, COLOR, FALLBACK_COLOR) {  
  background-color: FALLBACK_COLOR; /* fallback color if gradients are not supported */  
  background-image: -webkit-linear-gradient(POS, hsl(HSL1, HSL2, HSL3), COLOR);  
  /* Chrome 10+, Safari 5.1+ */  
  /* @alternate */  
  background-image: -moz-linear-gradient(POS, hsl(HSL1, HSL2, HSL3), COLOR);  
  /* FF3.6+ */  
  /* @alternate */  
  background-image: -ms-linear-gradient(POS, hsl(HSL1, HSL2, HSL3), COLOR); /* IE10 */  
  /* @alternate */  
  background-image: -o-linear-gradient(POS, hsl(HSL1, HSL2, HSL3), COLOR);  
  /* Opera 11.10+ */  
}  
  
.header {  
  @mixin gradient(top, 0%, 50%, 70%, #cc0000, #f07575);  
}
```



Mixins - for browser specific code

```
.header {  
  background-color: #f07575;  
  background-image: -webkit-linear-gradient(top,hsl(0%,50%,70%) ,#cc0000);  
  background-image: -moz-linear-gradient(top,hsl(0%,50%,70%) ,#cc0000);  
  background-image: -ms-linear-gradient(top,hsl(0%,50%,70%) ,#cc0000);  
  background-image: -o-linear-gradient(top,hsl(0%,50%,70%) ,#cc0000);  
}  
}
```



Conditionals

- defining variables in conditionals



Conditionals

```
@if (BROWSER_IE) {  
  @if (BROWSER_IE6) {  
    @def GOOG_INLINE_BLOCK_DISPLAY inline;  
  } @elseif (BROWSER_IE7) {  
    @def GOOG_INLINE_BLOCK_DISPLAY inline;  
  } @else {  
    @def GOOG_INLINE_BLOCK_DISPLAY inline-block;  
  }  
} @elseif (BROWSER_FF2) {  
  @def GOOG_INLINE_BLOCK_DISPLAY -moz-inline-box;  
} @else {  
  @def GOOG_INLINE_BLOCK_DISPLAY inline-block;  
}  
  
.goog-inline-block {  
  position: relative;  
  display: GOOG_INLINE_BLOCK_DISPLAY;  
}  
  
```



Conditionals

- “java -jar closure-stylesheets.jar --define BROWSER_FF2 --pretty-print conditionals-example.gss”

```
.goog-inline-block {  
  position: relative;  
  display: -moz-inline-box;  
}
```

- “java -jar closure-stylesheets.jar --pretty-print conditionals-example.gss”

```
.goog-inline-block {  
  position: relative;  
  display: inline-box;  
}
```



Linting

- check your CSS for errors
- will fail the compile with an error message



Linting

```
.logo {  
  width: 150px;  
  height: 55px;  
  background-image: url('http://www.google.com/images/logo_sm.gif');  
  border-color: #DCDCDC;  
  border-color: rgba(0, 0, 0, 0.1);  
}
```



Linting

```
.logo {  
  width: 150px;  
  height: 55px;  
  background-image: urel('http://www.google.com/images/logo_sm.gif');  
  border-color: #DCDCDC;  
  border-color: rgba(0, 0, 0, 0.1);  
}
```

Unknown function "urel" in linting-example.gss at line 4 column 21:

```
background-image: urel('http://www.google.com/images/logo_sm.gif');  
^
```

Detected multiple identical, non-alternate declarations in the same ruleset.

If this is intentional please use the /* @alternate */ annotation.

```
border-color:[rgba(0,0,0,0.1)] in linting-example.gss at line 7 column 1:
```

```
}
```

```
^
```

```
2 error(s)
```



RTL flipping

```
.logo {  
  margin-left: 10px;  
}  
  
.shortcut_accelerator {  
  /* Keyboard shortcuts are untranslated; always left-to-right. */  
  /* @noflip */ direction: ltr;  
  border-right: 2px solid #ccc;  
  padding: 0 2px 0 4px;  
}
```



RTL flipping

```
.logo {  
  margin-right: 10px;  
}  
  
.shortcut_accelerator {  
  direction: ltr;  
  border-left: 2px solid #ccc;  
  padding: 0 4px 0 2px;  
}
```



GSS



Integration with GWT

- Why ?
 - Closure-stylesheets are powerful
 - easier to maintain CSS
 - GWT needs better CSS3 support
- What does it look like?
 - A new GSSResource
 - similar to CSSResource



CSS: application.gss

```
@def PADDING_RIGHT 50px;
@def PADDING_LEFT 50px;

@mixin size(WIDTH, HEIGHT) {
    width: WIDTH;
    height: HEIGHT;
}

.header, span[data-role^="header"] {
    display: block;
    @ mixin size(150px, 50px);
}

.header-with-padding {
    @ mixin size(add(PADDING_RIGHT, 150px, PADDING_LEFT), 50px);
    padding-right: PADDING_RIGHT;
}
```



Java: define GssResource

```
public interface ApplicationResources extends ClientBundle {  
  
    public interface ApplicationGss extends GssResource {  
        int paddingRight();  
  
        String header();  
  
        @ClassName("header-with-padding")  
        String headerWithPadding();  
    }  
  
    @Source("application.gss")  
    public ApplicationGss style();  
}
```



Usage:

```
ApplicationResources resources = GWT.create(ApplicationResources.class);
ApplicationGss style = resources.style();
style.ensureInjected();

// ...
myWidget.addStyleName(style.header());
```



CssResource features

- @sprite
- @eval
- value function



Image sprite (@sprite)

```
.addIcon {  
    gwt-sprite: 'addIcon';  
}
```



Runtime substitution (@eval)

```
@def BLACK eval('mypackage.GssSampleApplication.black()');

.black {
    color: eval('mypackage.GssSampleApplication.black()');
}
```



Value function

```
@def ICON_WIDTH value('addIcon.getWidth', 'px');
```



How far is this done?

- supports all Closure-stylesheet features
- supported CssResource features
 - image sprites
 - runtime substitution
 - value function
 - RTL support



What is missing?

- conditionals based on permutations

```
@if user.agent safari {  
    /* ... */  
}
```

```
@if locale en {  
    /* ... */  
}
```



Roadmap

- Ready to use, but there might still be issues
- please test & report issues at: [https://github.com/jDramaix/gss.gwt/
issues](https://github.com/jDramaix/gss.gwt/issues)
- Expect missing user agent conditionals in Q1
- Unclear: How will CssResource transition look like
- If everything goes smoothly, expect GssResource in GWT 3.0, but you can use it right now!



Demo



Summary

- simple and maintainable CSS with Closure-styleSheets
 - many exciting features
 - widely used inside of Google
- GssResource
 - brings Closure-styleSheets to GWT, aka CssResource for CSS
 - brings all CSS3 features to GWT
 - thanks to Julien Dramaix for this big contribution
- Roadmap
 - Test and report bugs!



Questions?



Julien Dramaix



gplus: <https://plus.google.com/+JulienDramaix>

twitter: @jDramaix

Daniel Kurka



www.m-gwt.com

gplus: [daniel-kurka.de/+](https://plus.google.com/u/0/+daniel.kurka.de)

twitter: @dankurka