

```
// Kristofer Walters (Knowit Oslo)
```

```
#Maintainable.CSS {
```

```
// The Next Frontier Of  
// Front-End Engineering
```

```
}
```

```
<blink color="#FF0000">
```

Best practices

considered harmful

```
</blink>
```

Maintainable Code

- * Easy to understand
- * Easy to fix
- * Easy to add new stuff
- * Easy to adjust for new environments

Separation of Content and Presentation

```
<aside id="articles">
  <div class="article">
    <h3>Lorem Ipsum</h3>
    <p>...</p>
    <a href="...">Read more</a>
  </div>
  ...
</aside>
```

```
#articles { ... }  
.article { ... }  
.article h3 { ... }  
.article p { ... }  
.article a { ... }
```

```
#articles {  
  color: #009682;  
  left: 785px;  
  font-size: 0.8125em;  
  ...  
}
```

```
$my-brand-green: #009682;
```

```
@function rel-font-size($size) {  
  @return ($size / $context-size) * 1em;  
}
```

```
#articles {  
  color: $my-brand-green;  
  left: $page-width - $main-col-width;  
  font-size: rel-font-size(13);  
  ...  
}
```


Don't
Repeat
Yourself

```
background: -moz-linear-gradient(top, #CCC 0%, #FFF 100%);  
background: -webkit-gradient(linear, left top, left bottom, color-stop(0%,#CCC),  
color-stop(100%,#FFF));  
background: -webkit-linear-gradient(top, #CCC 0%,#FFF 100%);  
background: -o-linear-gradient(top, #CCC 0%,#FFF 100%);  
background: -ms-linear-gradient(top, #CCC 0%,#FFF 100%);  
background: linear-gradient(to bottom, #CCC 0%,#FFF 100%);  
filter: progid:DXImageTransform.Microsoft.gradient( startColorstr='#cccccc',  
endColorstr='#ffffff',GradientType=0 );
```

```
@import "compass/css3/images";
```

```
.button {  
  @include background-image(  
    linear-gradient(left top,  
      #FFF, #CCC));  
}
```

```
.article {
  float: left;
  ""
}

.article:after, .article:before {
  content: " ";
  display: table;
}

.article:after {
  clear: both;
}

// IE 6/7 - trigger hasLayout
.article {
  *zoom: 1;
}
```

```
@import "compass/utilities/clearfix",
```

```
.article {
```

```
  float: left;
```

```
  @include clearfix;
```

```
  ...
```

```
}
```



```
<aside id="articles">
  <div class="article">
    <h3>Lorem Ipsum</h3>
    <p>...</p>
    <a href="...">Read more</a>
  </div>
  ...
</aside>
```

```
#articles { ... }  
.article { ... }  
.article h3 { ... }  
.article p { ... }  
.article a { ... }
```



```
.article h3 { ... }
```

```
.article h3 { ... }
```

```
#sidebar .article h3 { ... }
```

```
.article h3 { ... }
```

```
#sidebar .article h3 { ... }
```

```
.author-page #sidebar .article h3 { ... }
```

```
.article h3, .article h4 { ... }  
#sidebar .article h3,  
#sidebar .article h4 { ... }  
.author-page #sidebar .article h3,  
.author-page #sidebar .article h4 { ... }
```

Object
Oriented
CSS

laoreet risus, id faucibus dui urna
sollicitudin velit.

[Read More](#)

Lorem Ipsum

Lorem ipsum dolor sit amet, consectetur
adipiscing elit. Fusce pulvinar, turpis
ac volutpat ultricies, massa nisl
laoreet risus, id faucibus dui urna
sollicitudin velit.

[Read More](#)

Lorem Ipsum

Lorem ipsum dolor sit amet, consectetur
adipiscing elit. Fusce pulvinar, turpis
ac volutpat ultricies, massa nisl

```
<div class="article">  
  <h3>Lorem Ipsum</h3>  
  <p>...</p>  
  <a href="...">Read more</a>  
</div>
```

```
<div class="mod">
  <div class="inner">
    <div class="hd highlighted">
      <h3 class="h3">Lorem Ipsum</h3>
    </div>
    <div class="bd">
      <p>...</p>
    </div>
    <div class="ft">
      <a href="...">Read more</a>
    </div>
  </div>
</div>
```


.mod { ... }

.hd { ... }

.bd { ... }

.ft { ... }

.h3 { ... }

.highlighted { ... }

Block
Element
Modifier

#{block}

#{block} + #{element}

#{block} + #{modifier}

#{block} + #{element} + #{modifier}

```
.global-nav { ... }
```

```
.global-nav { ... }  
.global-nav-item { ... }  
.global-nav-logged-in { ... }
```

```
.global-nav { ... }
```

```
.global-nav_item { ... }
```

```
.global-nav--logged-in { ... }
```

```
.global-nav { ... }  
.global-nav_item { ... }  
.global-nav--logged-in { ... }  
.global-nav_item--disabled { ... }
```

```
.global-nav { ... }  
.global-nav_item { ... }  
.global-nav--logged-in { ... }  
.global-nav_item--disabled { ... }
```

```
.global-nav--logged-in  
  .global-nav_item { ... }
```



```
.globalNav { ... }  
.globalNav-item { ... }  
.globalNav_loggedIn { ... }  
.globalNav-item_disabled { ... }
```

```
.globalNav_loggedIn  
  .globalNav-item { ... }
```

```
<ul class="nav nav_loggedIn">
  <li class="nav-item">
    <a href="nav-link"> ... </a>
  </li>
  <li class="nav-item nav-item_cur">
    <a href="nav-link"> ... </a>
  </li>
</ul>
```

Scalable and Modular Architecture for CSS

Base Rules

Layout Rules

Module Rules

State Rules

Theme Rules

```
@import url("base.css");  
@import url("layout.css");  
@import url("modules.css");  
  
@import url("theme.css");
```

```
@import "base/*.scss";  
@import "layout/*.scss";  
@import "modules/*.scss";  
  
@import "theme/*.scss";
```

SOLID?

Single responsibility principle

A class should have only a single responsibility

Open/Closed principle

open for extension,
closed for modification

Liskov substitution principle

objects in a program should be replaceable with instances of their subtypes without altering the correctness of that program

Interface segregation principle

many client-specific interfaces
are better than one general-purpose
interface

Dependency inversion principle

depend upon abstractions,
not upon concretions

Individuals and interactions
over processes and tools

Working software

over comprehensive documentation

Customer collaboration

over contract negotiation

Responding to change

over following a plan

Default styles

Individual form controls receive styling, but without any required base class on the `<form>` or large changes in markup. Results in stacked, left-aligned labels on top of form controls.

Example

Legend

Label name

Example block-level help text here.

Check me out

```
1. <form>
2.   <fieldset>
3.     <legend>Legend</legend>
4.     <label>Label name</label>
5.     <input type="text" placeholder="Type something...">
6.     <span class="help-block">Example block-level help text here.</span>
7.     <label class="checkbox">
8.       <input type="checkbox"> Check me out
9.     </label>
10.    <button type="submit" class="btn">Submit</button>
11.  </fieldset>
12. </form>
```

Strap-on V3.0

FINN.no style guide



What's the big deal anyways?

It's all about finding the standard! By defining small and highly reusable building blocks there is no longer need for lots and lots of custom css.

About Objectoriented CSS (OOCSS)

Tools

BSniff - tiles duplicator checking

Who's to blame?

The Strap-on project is owned by the Frontend Core Team consisting of [Espen](#) (Political lubricator), [Anders](#) (Penetrator), [Snorre](#) (Penetrator)

Layout

Template

Base Classes

Property	Description
page	Main wraps site content.
head	Site header (e.g. horizontal navigation, logo, login)
body	Main content area, body of the page.
foot	Site footer

```
1. <div class="page">
2.   <div class="head"><!-- Head --></div>
3.   <div class="body"><!-- Body --></div>
4.   <div class="foot"><!-- Foot --></div>
5. </div>
```

Layout

- Template
- Grid
- Margin/Padding
- Module
- Media Block

Core

- Fonts
- Color
- Forms
- Fieldsizes
- Tables
- Buttons
- Lists

Components

- Collapse
- Modal
- Speechbubbles
- Ratings
- Popover
- Paging
- Omgnew
- Icons
- Crumbs
- Decoration
- Businesscard
- Ribbons
- Tooltip
- Tabs
- Videoplayer

Grid overlay

“Always code as if the person who ends up maintaining your code is a **violent psychopath** who knows where you live.”

kris@wltrs.org @kwltrs

Sass

<http://sass-lang.com/>

Compass

<http://compass-style.org/>

OOCSS

<https://github.com/stubbornella/oocss/wiki>

BEM

<http://bem.info/method/>

SMACSS

<http://smacss.com/>