

CSS Architecture – BEM

Block Element Modifier



SoftUni Team
Technical Trainers




SoftUni



Software University

<https://softuni.org>

Table of Contents

- 
1. What is BEM?
 2. Benefits
 3. Problems with BEM
 4. Naming - BEM
 5. BEM in Action




sli.do

#front-end



Block, Element, Modifier Methodology


What is BEM?

- 
- A popular **naming convention** for classes in HTML and CSS
 - The goal is to help developers better understand the **relationship** between the HTML and CSS in a given project

```
/* Block component */  
.btn {}
```

```
/* Element that depends upon the block */  
.btn__price {}
```

What is BEM?

- 
- Parent Element - `.btn{}`
 - Child Elements (denoted by two underscores following the name of the block) `.btn__price { }`
 - Modifiers - `.btn--orange{}`

```
/* Modifier that changes the style of the block */  
.btn--orange {}  
.btn--big {}
```

The Markup

```
<a class="btn btn--big btn--orange" href="#">  
  <span class="btn__price">$9.99</span>  
  <span class="btn__text">Subscribe</span>  
</a>
```

- It's easy to understand:
 - Which classes are **responsible for what**
 - How they **depend** on one another
- Without writing CSS, developers are potentially capable of creating **different combinations** of buttons by changing a class in the markup



```
<a href="#" class="btn">
  <span class="btn__text">Standard button</span>
</a>
<a href="#" class="btn btn--orange btn--big">
  <span class="btn__price">$6</span>
  <span class="btn__text">Big button</span>
</a>
<a href="#" class="btn btn--blue btn--big">
  <span class="btn__price">$4</span>
  <span class="btn__text">Big button</span>
</a>
<a href="#" class="btn btn--green btn--big">
  <span class="btn__price">$9</span>
  <span class="btn__text">Big button</span>
</a>
```




```
/* Block */
.btn {
  text-decoration: none;
  background-color: white;
  color: #888;
  border-radius: 5px;
  display: inline-block;
  margin: 10px;
  font-size: 18px;
  text-transform: uppercase;
  font-weight: 600;
  padding: 10px 5px;
}

/* Element */
.btn__price {
  background-color: white;
  color: #fff;
  padding-right: 12px;
  padding-left: 12px;
  margin-right: -10px;
  font-weight: 600;
  background-color: #333;
  opacity: .4;
  border-radius: 5px 0 0 5px;
}
```

```
/* Element */
.btn__text {
  padding: 0 10px;
  border-radius: 0 5px 5px 0;
}

/* Modifier */
.btn--big {
  font-size: 28px;
  padding: 10px;
  font-weight: 400;
}

/* Modifier */
.btn--blue {
  border-color: #0074D9;
  color: white;
  background-color: #0074D9;
}

/* Modifier */
.btn--orange {
  border-color: #FF4136;
  color: white;
  background-color: #FF4136;
}
```

```
/* Modifier */
.btn--green {
  border-color: #3D9970;
  color: white;
  background-color: #3D9970;
}

body {
  font-family: "fira-sans-2", sans-serif;
  background-color: #ccc;
}
```



Why Should We Consider BEM?

Modularity

- Block styles are never **dependent** on other elements on a page
- Never experience problems from **cascading**
- Also get the ability to **transfer blocks** from finished projects to new ones



Reusability



- Composing **independent blocks** in different ways, and reusing them intelligently, reduces the amount of CSS code that you will have to maintain
- With a set of **style guidelines** in place, you can build a library of blocks, making your CSS super effective

Structure

- BEM methodology gives your CSS code a **solid structure** that remains simple and easy to understand



```
.button {  
  display: inline-block;  
  padding: 7px 12px;  
  border: 1px solid #D5D5D5;  
  background-image: linear-gradient(#EEE, #DDD);  
  font: 700 13px/18px Helvetica, arial;  
}  
.button--state-success {  
  color: #FFF;  
  background: #569E3D linear-gradient(#79D858, #569E3D) repeat-x;  
  border-color: #4A993E;  
}  
.button--state-danger {  
  color: #900;  
}
```



Problems with ~~BEM~~ CSS

Break from BEM Rules

Breaking BEM Rules

- Looks like it has parts of BEM going on, but it's **NOT BEM**

```
.nav .nav__listItem .btn--orange {  
  background-color: green;  
}
```

- It has:
 - Nested Selectors
 - The modifier doesn't describe what's going on
 - A block (such as **.nav**) should never override the styles of another block or modifier (such as **.btn--orange**)



Breaking BEM Rules

- Breaking BEM Rules goes for **HTML**, as well

```
<a class="btn" href="#">  
  <div class="nav__listItem">Item one</div>  
  <div class="nav__listItem">Item two</div>  
</a>
```

- The problems:
 - Never override modifiers in an **unrelated block**
 - Avoid making **unnecessary parent elements** when the child can exist by itself





Samuel Fine 👁️👄👁️
@samuelfine



Not sure I'm sold on BEM.

.site-search
.site-search__field
.site-search--full

Why not:

.site-search
.site-search input
.site-search .full

Naming Conventions

Naming – BEM


- It's not necessary to use **double-underscores** or **double-dashes** while naming
- In the **SMACSS** methodology, you're likely to find a CSS class name with three letters
 - Modifiers then follow the module name with a hyphen:

```
.btn { } /* Example Module */  
.btn-primary { } /* Modifier of the btn class */  
.btn.is-collapsed { } /* Btn Module with State */
```



Naming – BEM

- In **OOCSS**, blocks are similarly generic:

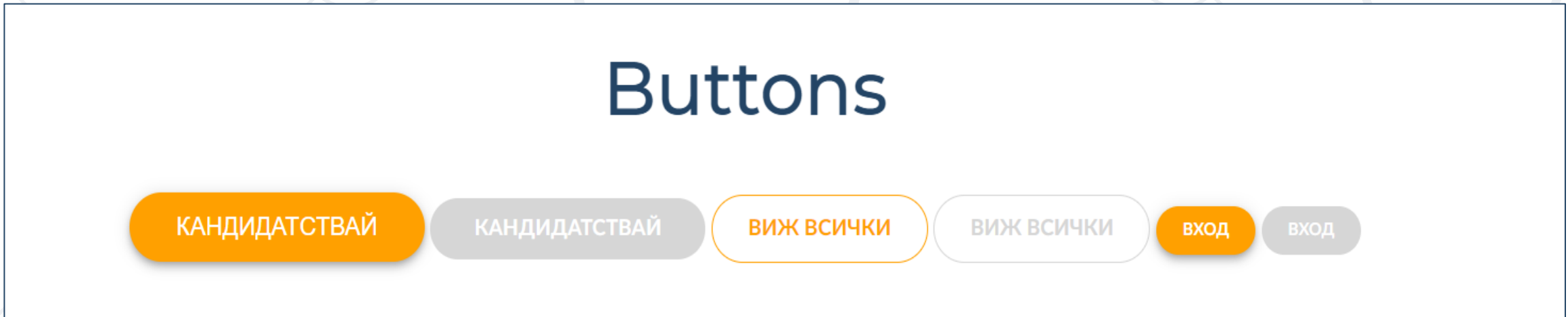


```
/* Example Module */  
.mod { }  
  
/* Part of Module */  
.inner { }  
  
/* Talk Module */  
.talk { }  
  
/* Variation of part inside Module */  
.talk .inner { }
```



Practical Exercises

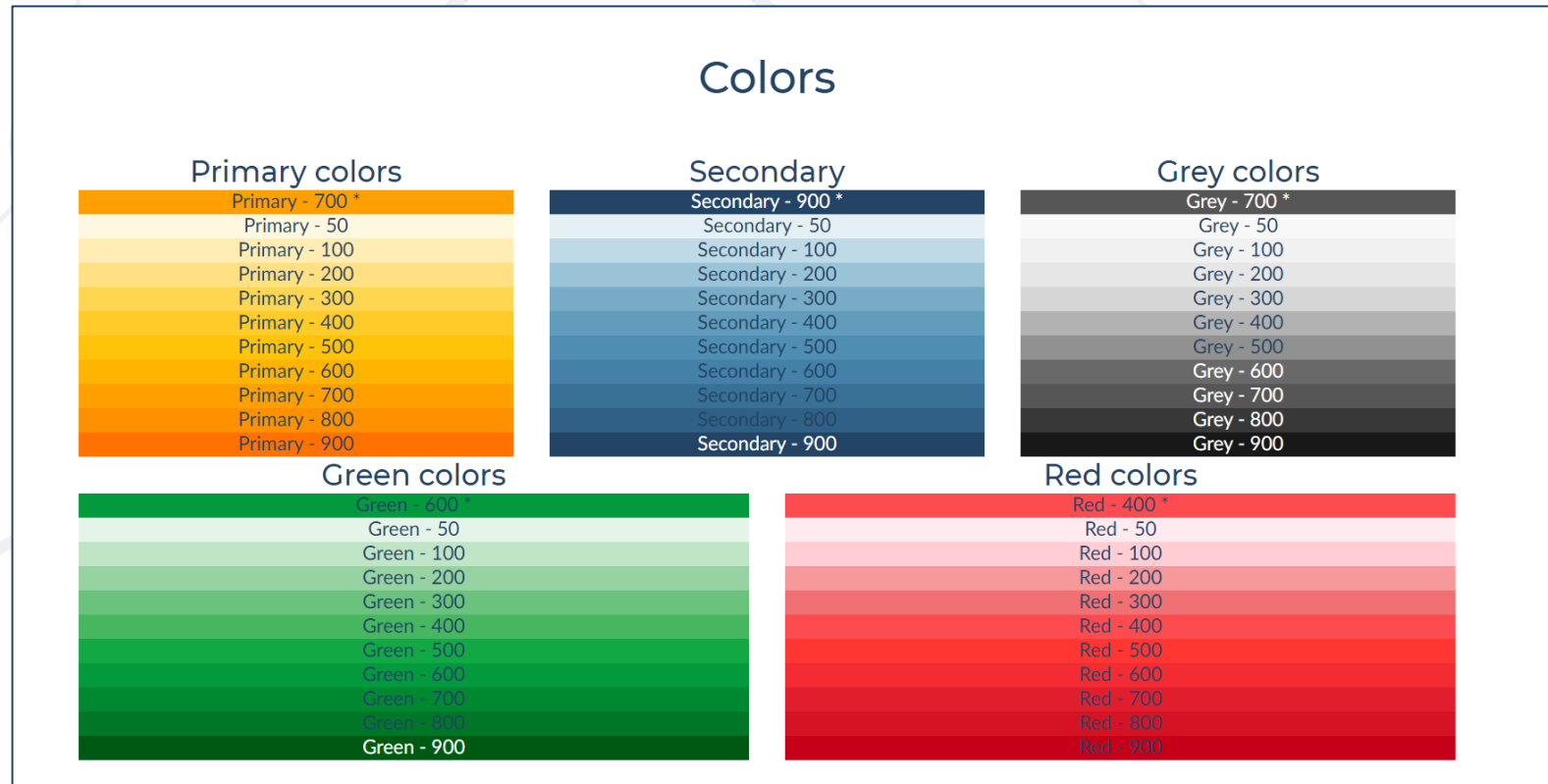
- Create a web page using the BEM methodology like the following:



- Use the following design guideline

Colors – Design Guidelines

- Create a web page using the BEM methodology like the following:



- Use the following design guideline

- What is BEM?
- Benefits - **Modularity, Reusability, Structure**
- Problems with BEM
- Naming - BEM
- BEM in **Action**



Questions?



SoftUni Diamond Partners



NETPEAK



Postbank

Решения за твоето утре



SoftwareGroup
doing it right



SmartIT



Coca-Cola HBC
Bulgaria

INDEAVR

Serving the high achievers

tek
experts



SBTech
we know sports



INFRAGISTICS®

SUPERHOSTING.BG



telenor

- This course (slides, examples, demos, exercises, homework, documents, videos and other assets) is **copyrighted content**
- Unauthorized copy, reproduction or use is illegal
- © SoftUni – <https://softuni.org>
- © Software University – <https://softuni.bg>



- Software University – High-Quality Education, Profession and Job for Software Developers
 - softuni.bg
 - Software University Foundation
 - softuni.foundation
- Software University @ Facebook
 - facebook.com/SoftwareUniversity
- Software University Forums
 - forum.softuni.bg

