

B__E_M



Hi, there!

I'm Luciano Battagliero

@battaglir

**Block,
Element,
Modifier**

Yandex

~2009

A little bit of context

OOCSS ~2010

SMACSS ~2011

What is it?

It's a methodology

Heavily complemented by a set
of **tools, libraries** and a complete
technology stack

At its **core** BEM is an **unified**
semantic for **different**
implementations

*That being said, this talk will be
focused mainly on CSS*

**What does
BEM solve?**

Chaos

/'kā,äs/

noun

1. Complete disorder and confusion

There are **two types** of **problems**
we face in **CSS**

Layout or cosmetic problems

Architectural problems

BEM attempts to **help solving**
architecture related **problems**

Entity

A generic **term** to **refer** to **blocks**,
elements or **modifiers**

Block

An **independent** and **self-sufficient**
component of an interface

Provides structure, behaviour and
appearance **encapsulation**

```
<dialog class="modal"> ... </dialog>
```

HTML

```
<dialog class="modal"> ... </dialog>
```

HTML


```
.modal { ... }
```

CSS

Blocks must be **context independent**,
thus they should **not** have direct
influence over **other blocks**

```
<dialog class="modal">  
  <button class="btn"> ... </button>  
</dialog>
```

HTML

```
<dialog class="modal">  
  <button class="btn"> ... </button>  
</dialog>
```

HTML

```
/* Don't do this */  
.modal .btn { ... }
```

CSS

```
/* Do this */  
.modal__btn { ... }
```

CSS

Blocks **can contain** other **blocks**

```
<header class="header">  
  <nav class="nav"> ... </nav>  
</header>
```

HTML


```
<header class="header">  
  <nav class="nav"> ... </nav>  
</header>
```

HTML

Multiple instances of a block could be used across the interface

```
<button class="btn"> ... </button>
```

```
...
```

```
<dialog class="modal modal--message">
```

```
  <button class="modal__confirm btn">
```

```
    ...
```

```
  </button>
```

```
</dialog>
```

HTML

```
<button class="btn"> ... </button>
```

```
...
```

```
<dialog class="modal modal--message">
```

```
  <button class="modal__confirm btn">
```

```
    ...
```

```
  </button>
```

```
</dialog>
```

HTML

```
<button class="btn"> ... </button>
```

```
...
```

```
<dialog class="modal modal--message">
```

```
  <button class="modal__confirm btn">
```

```
    ...
```

```
  </button>
```

```
</dialog>
```

HTML

Element

An **internal part** of a block that
can **not** be **used outside** of it

Elements **can contain** other
elements or **blocks**


```
<ul class="paginator">  
  <li class="paginator__page">  
    <a class="paginator__link"> ... </a>  
  </li>  
</ul>
```

HTML

```
<ul class="paginator">
  <li class="paginator__page">
    <a class="paginator__link"> ... </a>
  </li>
</ul>
```

HTML

```
<ul class="paginator">  
  <li class="paginator__page">  
    <a class="paginator__link"> ... </a>  
  </li>  
</ul>
```

HTML

Elements should **not** attempt to
be a **representation** of the
DOM structure

```
<ul class="paginator">
  <li class="paginator__page">

    <!-- Don't do this -->
    <a class="paginator__page__link"> ... </a>

  </li>
</ul>
```

HTML

```
<ul class="paginator">
  <li class="paginator__page">

    <!-- Don't do this -->
    <a class="paginator__page__link"> ... </a>

  </li>
</ul>
```

HTML

```
<ul class="paginator">
  <li class="paginator__page">

    <!-- Don't do this -->
    <a class="paginator__page__link"> ... </a>

  </li>
</ul>
```

HTML

A block may not contain any element

Modifier

A **variation** on the **appearance** or
behavior of a **block** or an **element**

```
<a class="btn btn--large"> ... </a>
```

HTML

```
<a class="btn btn--large"> ... </a>
```

HTML

```
<a class="btn btn--large"> ... </a>
```

HTML

```
<a class="btn btn--is-disabled"> ... </a>
```

HTML

```
<a class="btn btn--is-disabled"> ... </a>
```

HTML

```
<a class="btn btn--is-disabled"> ... </a>
```

HTML

Multiple modifiers can be **used simultaneously** on the same block or element

```
<a class="btn btn--large btn--is-disabled">
```

```
...
```

```
</a>
```

HTML

```
<a class="btn btn--large btn--is-disabled">  
  ...  
</a>
```

HTML

```
<a class="btn btn--large btn--is-disabled">  
  ...  
</a>
```

HTML

Naming conventions

The **main purpose** of a **class name** is to be used as a **hook** for adding style or behaviour

A class name should **communicate information** which helps to **understand its purpose**

*Remember that a class name
can not be “unsemantic”^[*]*

The “official” syntax

```
/* Basic syntax */  
.block__element_modifier
```

CSS

```
/* Basic syntax */  
.block__element_modifier
```

CSS

```
/* Basic syntax */  
.block__element_modifier
```

CSS

```
/* Basic syntax */  
.block__element_modifier
```

CSS

```
/* Key-value modifier */  
.block-or-element_modifier_value
```

CSS

```
/* Key-value modifier */  
.block-or-element_modifier_value
```

CSS

```
/* Entities with compound names */  
.block-name__element-name_modifier-name
```

CSS


```
/* Entities with compound names */  
.block-name__element-name_modifier-name
```

CSS

```
/**  
 * All possible combinations  
 */
```

```
.block  
.block_modifier  
.block__element  
.block__element_modifier
```

CSS

```
/**  
 * Don't do any of these  
 */
```

```
.element  
.block_modifier__element  
.block__element__element
```

CSS

The “popular” syntax

```
/* Basic syntax */  
.block__element--modifier
```

CSS

```
/* Basic syntax */  
.block__element--modifier
```

CSS

```
/* Basic syntax */  
.block__element--modifier
```

CSS

```
/* Basic syntax */  
.block__element--modifier
```

CSS


```
/* Entities with compound names */  
.block-name__element-name--modifier-name
```

CSS

```
/* Entities with compound names */  
.block-name__element-name--modifier-name
```

CSS

```
/**  
 * All possible combinations  
 */
```

```
.block  
.block--modifier  
.block__element  
.block__element--modifier
```

CSS

```
/**  
 * Don't do any of these  
 */
```

```
.element  
.block--modifier__element  
.block__element__element
```

CSS

The “CamelCase” syntax

```
/* Basic syntax */  
.Block-element--modifier
```

CSS

```
/* Basic syntax */  
.Block-element--modifier
```

CSS

```
/* Basic syntax */  
.Block-element--modifier
```

CSS


```
/* Basic syntax */  
.Block-element--modifier
```

CSS

```
/* Entities with compound names */  
.BlockName-elementName--modifierName
```

CSS

```
/* Entities with compound names */  
.BlockName-elementName--modifierName
```

CSS

```
/**  
 * All possible combinations  
 */
```

```
.Block  
.Block--modifier  
.Block-element  
.Block-element--modifier
```

CSS

```
/**  
 * Don't do any of these  
 */
```

```
.element  
.Block--modifier-element  
.Block-element-element
```

CSS

Syntax comparison

```
/* “Official” */
```

```
block-name__element-name_modifier-name
```

```
/* “Popular” */
```

```
block-name__element-name--modifier-name
```

```
/* “CamelCase” */
```

```
BlockName-elementName--modifierName
```

CSS

```
/* “Official” */
```

```
block-name__element-name_modifier-name
```

```
/* “Popular” */
```

```
block-name__element-name--modifier-name
```

```
/* “CamelCase” */
```

```
BlockName-elementName--modifierName
```

CSS


```
/* “Official” */
```

```
block-name__element-name_modifier-name
```

```
/* “Popular” */
```

```
block-name__element-name--modifier-name
```

```
/* “CamelCase” */
```

```
BlockName-elementName--modifierName
```

CSS

```
/* “Official” */
```

```
block-name__element-name_modifier-name
```

```
/* “Popular” */
```

```
block-name__element-name--modifier-name
```

```
/* “CamelCase” */
```

```
BlockName-elementName--modifierName
```

CSS

*It has been scientifically proven
that BEM class names are ugly*

Mix

A **combination** of different **entities**
on a **single DOM node**

```
<input class="search__input input" />
```

HTML

```
<input class="search__input input" />
```

HTML

Implementation guidelines

Only **define entities** using **class selectors**

```
.icon { ... }
```

CSS

Use **descendant selectors** only
when needed

```
/* Don't do this */  
.menu .menu__item { ... }
```

CSS

```
/* Do this */  
.menu--horizontal .menu__item { ... }
```

CSS

Do **not** use **type** or **id** selectors

```
/* Don't do this */  
button { ... }
```

CSS

```
/* Do this */  
.button { ... }
```

CSS


```
/* Don't do this */  
#breadcrumb { ... }
```

CSS

```
/* Do this */  
.breadcrumb { ... }
```

CSS

Do **not** declare **styles** outside
of **blocks**

```
/* Don't do this */  
ul {  
    list-style: none;  
}
```

CSS

```
/* Do this */  
.menu {  
    list-style: none;  
}
```

CSS

*This usually means no global styles
and no global “resets”*

**That's just a
part of it!**

There's a lot **more** about
BEM. Go to bem.info
and find out!

Questions?

Thanks!