Naming conventions

BEM, or “Block-Element-Modifier”, is a *naming convention* for classes in HTML and CSS.

We will be using BEM for the following reasons:

- It helps create clear, strict relationships between CSS and HTML
- It helps us create reusable, composable components
- It allows for less nesting and lower specificity
- It helps in building scalable stylesheets

Example:

HTML

```html
<a class="btn btn--big btn--orange" href="https://css-tricks.com">
  <span class="btn__price">R199.99</span>
  <span class="btn__text">Subscribe</span>
</a>
```

CSS

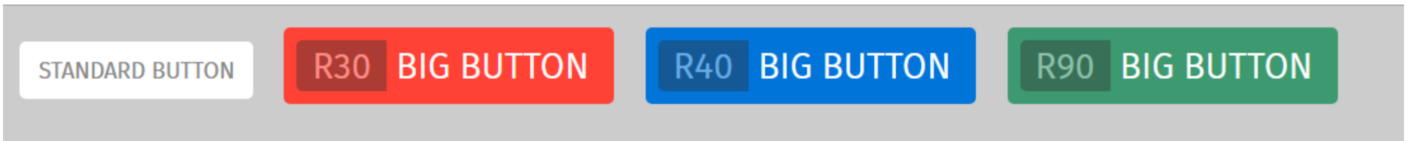
```css
/* Block component */
.btn {}

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

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

- In this CSS methodology a **block** is a top-level abstraction of a new component, for example a button: `.btn { }`. This block should be thought of as a parent.
- Child items, or **elements**, can be placed inside and these are denoted by two underscores following the name of the block like `.btn__price { }`.
- Finally, **modifiers** can manipulate the block so that we can theme or style that particular component without inflicting changes on a completely unrelated module. This is done by appending two hyphens to the name of the block just like `btn--orange`.

> If another developer wrote this markup, and we weren’t familiar with the CSS, we should still have a good idea of which classes are responsible for what and how they depend on one another. Developers can then build their own components and modify the existing block to their heart’s content. Without writing much CSS, developers are potentially capable of creating many different combinations of buttons simply by changing a class in the markup:



Revision #5
Created 17 September 2020 02:02:44 by Theuns Pretorius
Updated 5 October 2020 04:50:25