CSS Class Naming and BEM

MA WEB DESIGN AND CONTENT PLANNING

> There are only two hard things in computer science: cache invalidation and naming things.
>
> - Phil Kalton

Why naming is important?

- A poorly chosen name will cause confusion and likely hangout your project moving forward.

- A good name will make understandable code and make things easier and reusable for all.

CSS Class Naming

- To make the CSS selectors as informative and readable as possible.

- Helps in development, debugging and other issues that can come when creating HTML based layouts.

We can usually fit a given class name into one of three categories:

Functional Class Names

Content-Based Class Names

Presentational Class Names

Functional class names

The styling of this element is based on their function or meaning.

For Example

```
<button class="important-text">Send Message</button>
```

Content-Based Class Names

Content-based class names are class names that describe the content they contain.

For Example

```
<button class="submit-button">Send Message</button>
```

Presentational Class Names

Presentational class names describe the way an element looks–like green-button or big-text. The name itself is describing the styles that are being applied.

For Example

```
<button class="green-button">Send
Message</button>
```

How to properly name CSS classes?

Choose a good name for HTML elements

For Example

```
 <div class=”submit”/> <!-- don't do? -->
<input class=”submit”/> <!-- would be better written as -->
```

Put the class name at the lowest possible level

For Example

/* Don’t do this */

HTML

```
 <main class='mainly'>
    <p>text/p>
 </main>
```

CSS

```
main.mainly p {
 }
```

/* Would be better written as */

HTML

```
<main>
   <p class=”paragraphy”>text</p>
</main>
```

CSS

```
.paragraphly {
}
```

Do not use camelCase

For Example

```
.redBox {
          border: 1px solid red;
 }
```

Use a word hyphen(-)

```
.red-box {
      border: 1px solid red;.
  }
```

Use fully descriptive words

Avoid abbreviation, apart from classics like nav, txt, url, btn…

Try BEM (Block, Element, Modifier)

For Example

```
/* Don’t do this */
HTML
 <button class="btn--secondary"></button>

CSS
.btn--secondary{
}

/*Would be better written as*/
HTML
<button class="btn btn--secondary"></button>

CSS
.btn{
}

.btn--secondary{
}
```

Try to avoid more than two words for a given name

For Example

```
/* Don’t Do this */
.button-dropdown-one{
}

/* Would be better written as */
 .button{
 }

 .button-dropdown{
 }
```

Parent-Child relationships

For Example

```
/* Don't do this */
.post {
    .title {...}
}
/* Would be better written as */
.post {

}
.post-title {

}
```

The most popular CSS Class naming methodologies are

SMACSS

(Scalable and Modular Architecture for CSS)

OOCSS

(Object Oriented CSS)

BEM

(Block, Element, Modifier)

BEM (Block, Element, Modifier)

- Developed by the team at Yandex in 2009
- BEM is a front end methodology.
- A popular industry standard for naming using prefixing, underscores and dashes.
- Help developers better understand the relationship between the HTML and CSS in a given project.

Block

- An independent, reusable part of your interface
- Think: Header, Menus, Images etc
- Can contain Elements and other Blocks
- Can contain one or more Modifier
- Words are separated by a hyphen (-)

For Example:

HTML	CSS

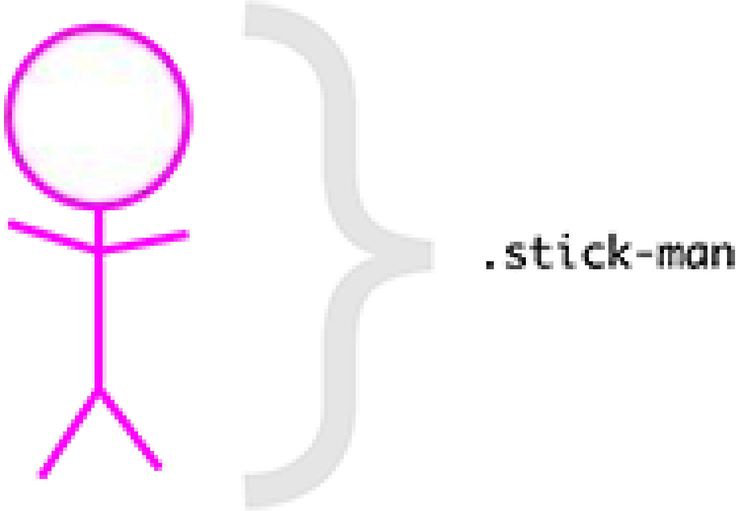
<form class=”site-search”>
</form>
	.site-search{
	}

The stick-man represents a component, such as a block of design.

For Example

```
.stick-man {

 }
```



Element

- Parts of a block and have no standalone meaning.
- Think: Header sub title, Menu item, Image caption etc
- Can contain other Elements and Blocks
- Can contain one or more Modifiers
- The element name is separated from the block name with two underscores: “__”.

For Example:

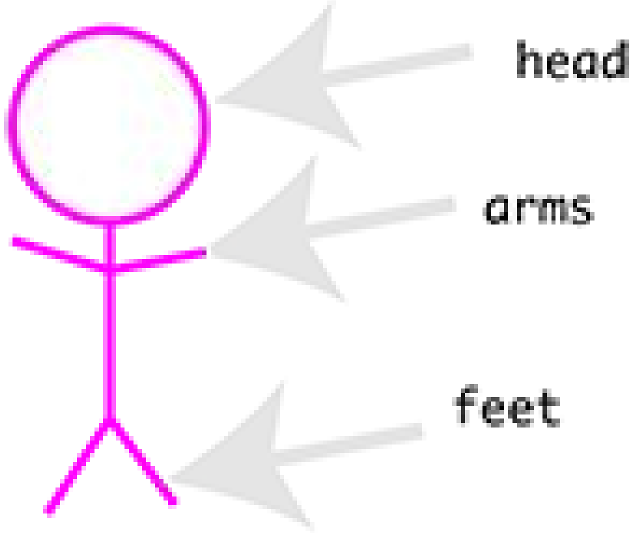
HTML	CSS

<form class="site-search">
        <input class="site-search__button"
        type="submit" value="search">
</form>
	site-search__button{
	}

The stick-man has a head, two arms, and feet.

For Example

```
.stick-man__head {
}
.stick-man__arms {
}
.stick-man__feet {
}
```



Modifier

- Used to change the appearance, behavior or state of a block or element.
- Think: theme, active/inactive state, alignment etc.
- Are separated from the block or element name by a single underscore(_) or by adding two hyphens(--).

For Example:

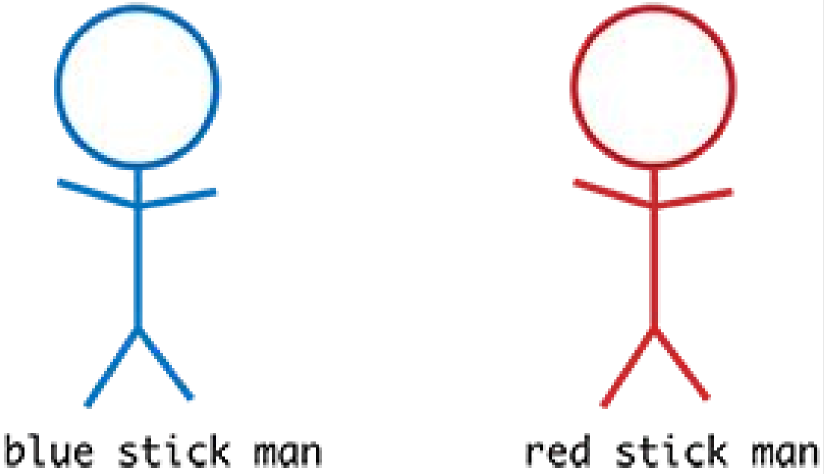
HTML	CSS

<form class="site-search site-search--themed">
<input class="site-search__button--large"
type="submit" value="search">
</form>
	site-search--themed{
	}
	site-search__button--large{
	}

Here, we have stick man in blue and red colour. These are modifications of the component.

For Example

```
.stick-man--blue {
}
.stick-man--red {
}
```



What does BEM CSS solve?

- Thinking in Blocks, Elements and Modifiers helps you figure out what your design is made of
- BEM methodology gives your CSS code a solid structure that remains simple and easy to understand.
- Pre planned design with independent blocks help reduces the maintenance time.
- Makes working on larger sites easier
- BEM is about communicating between technologies and the people using them.

References

https://seesparkbox.com/foundry/naming_css_stuff_is_really_hard

https://blog.toughbyte.com/blabla-9fd86eae4e6c

https://www.hallme.com/blog/css-naming-conventions/

http://bdavidxyz.com/blog/how-to-name-css-classes/

https://medium.freecodecamp.org/css-naming-conventions-that-will-save-you-hours-of-debugging-35cea737d849

http://philbaker.me/2014/02/css-class-naming-and-bem-contemporary-practice/

https://alistapart.com/article/meaningful-css-style-like-you-mean-it

http://getbem.com/naming/

More References

https://en.bem.info/methodology/naming-convention/

https://webdevstudios.com/2017/03/28/evolution-css-class-naming-methodologies/

https://www.slideshare.net/RolandLsslein/bem-css-seriously

Thank You