

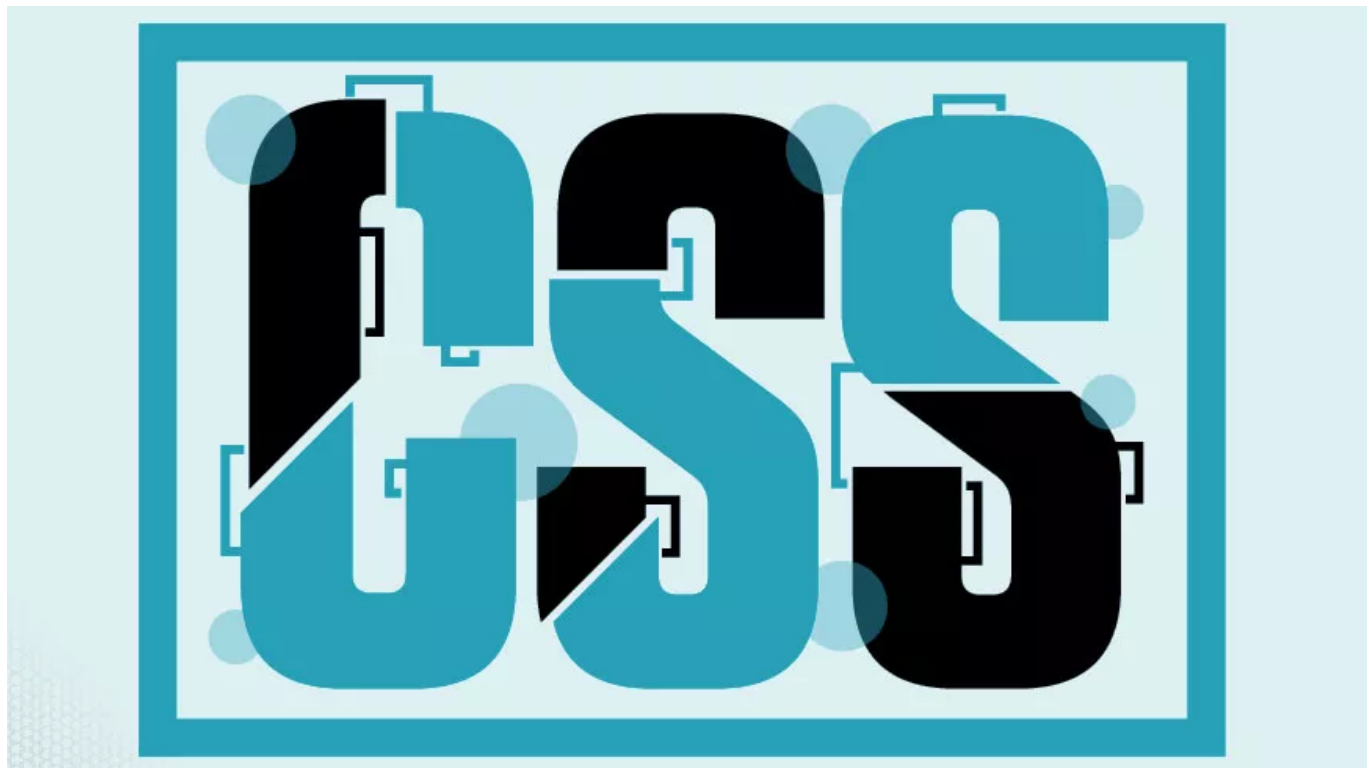
When you purchase through links on our site, we may earn an affiliate commission. [Here's how it works.](#)

[Home](#) > [Features](#) > [Web Design](#)

# A web designer's guide to CSS methodologies

By [Ahmed Abuelgasim](#) ( [Web Designer](#) ) last updated February 21, 2021

How to use Object-oriented CSS, BEM, SMACSS and Atomic CSS.



PAGE 1 OF 2: A GUIDE TO CSS METHODOLOGIES



Jump to: [OOCSS](#) | [ACSS](#) | [BEM](#) | [SMACSS](#)

A CSS methodology is a set of guidelines for writing modular, reusable and scalable code. Although CSS is an easy language to write, without an agreed-upon convention, the code gets messy almost as fast as it is written. Since each CSS declaration is defined on its own line, files get huge quickly, making them a nightmare to maintain.

## CSS METHODOLOGIES: QUICK LINKS

[Object-oriented CSS \(OOCSS\)](#)

[Atomic CSS \(ACSS\)](#)

[Block Element Modifier \(BEM\)](#)

[Scalable and Modular Architecture for CSS \(SMACSS\)](#)

To solve this and other CSS implementation issues (for a further explanation, jump to [Why do we need CSS methodologies?](#)), groups of coders around the world have developed different CSS methodologies, or sets of standard practices. Each comes with its own focus, advantages and disadvantages.

- [Cool CSS animation examples](#)

They are not frameworks or libraries, rather rules for writing CSS code that encourage developers to stick to conventions that make code easier to write and maintain, saving hours of development time. These methodologies are not mutually exclusive and can be used together in a way that best suits developers.

In this article we will take a look at the pros and cons of a few of the most popular CSS methodologies: object-oriented CSS, Atomic CSS (related to [atomic design](#)), BEM and SMACSS. Use the boxout opposite to jump to a particular methodology. Alternatively, hop to [page 2](#) to see how they can be combined together in a custom methodology.

Want to create a site without coding? You need a [website builder](#). And to keep your site running smoothly, pick the perfect [web hosting](#) service. Need to upgrade your storage? Try these [cloud storage](#) options.

## OOCSS

## Object-Oriented CSS

- **In a nutshell:** Divide layout into objects, then abstract their CSS into modules

OOCSS involves identifying objects on a page and separating their structural and visual CSS styles into two declaration blocks. These blocks can then be reused by different elements, and changes need only be made in one place, leading to better consistency.

Declaration blocks are applied to elements using single-class selectors to avoid specificity issues. This technique also separates content from container, so objects look the same wherever they appear. Classes also decouple mark-up from CSS. Using `.title` instead of `h2` for heading `<h2 class="title">` allows it to be changed to `<h3 class="title">` without changing the CSS.

To further separate HTML and CSS, class names should not include property values. A class 'blue' would require renaming in HTML and CSS if the colour changed.

Using OOCSS a button's CSS and markup can be defined as:

```
.button { padding: 10px 16px; }  
.primary-skin { color: blue; }  
.secondary-skin { color: green; }  
<button class="button primary-skin">primary skin button</button>  
<button class="button secondary-skin">secondary skin button</button>  
<div class="primary-skin">primary skin div</div>
```

OOCSS introduces many useful concepts, but its lack of rules leads to variations in interpretation that can result in inconsistencies. It has, however, been used as inspiration for stricter methodologies.

## ACSS

## Atomic CSS

- **In a nutshell:** Create a class selector for every repeating CSS declaration

**ACSS** encourages developers to define single-purpose class selectors for every reusable declaration. Unlike OOCSS, which discourages CSS property values in class names, ACSS welcomes it. Using ACSS styles can be defined and applied to elements as:

```
.mb-sm { margin-bottom: 16px; }  
.mb-lg { margin-bottom: 32px; }  
.color-blue { color: #1e90ff; }
```

```
<div class="mb-lg">
  <p class="mb-lg color-blue">Blue text</p>
  <img class="mb-sm" />
</div>
```

There are programmatic approaches to ACSS that automatically generate CSS based on classes or attributes that users add to the HTML. **Atomizer** is one such tool, allowing the previous HTML to be redefined as:

```
<div class="Mb(32px)">
  <p class="Mb(32px) C(#1e90ff)">Blue text</p>
  <img class="Mb(16px)" />
</div>
```

This would automatically generate the following CSS upon build:

```
.Mb\(16px\)
{ margin-bottom: 16px; }
.Mb\(32px\)
{ margin-bottom: 32px; }
.C\(#1e90ff\) { color: #1e90ff; }
```

The main benefit of ACSS is the ease of maintaining consistent code and not having to invent classes for components requiring a single CSS rule.

However, ACSS used on its own can lead to an unmanageable number of classes and bloated HTML files. It is therefore common to only use ACSS principles to create helper classes that define consistent, reusable declaration blocks.

## BEM

# Block Element Modifier

- **In a nutshell:** Use a standard naming convention for classes

BEM encourages developers to divide layouts into blocks and nested elements. Variations from the average appearance of a block or element should also be identified and applied using modifiers.

CSS declarations are applied using a single class name of format **block-name** for blocks and **block-name\_\_element-name** for elements, with two underscores in between. Modifier names are appended to classes, prefixed with an underscore or two hyphens for better clarity, for example **block-name\_\_element-name\_modifier-name** or **block-name\_\_element-name--modifier-name**. An object is a block if it can exist without ancestors, otherwise it's an element.

Blocks can have nested blocks and elements, but elements cannot. Modifiers must be used alongside block and element classes, not instead of them.

BEM can be applied to a list, where **list-block--inline** and **list-block\_\_item--active** display lists horizontally and highlight items respectively:

```
<ul class="list-block list-block--inline">
  <li class="list-block__item">Item 1</li>
  <li class="list-block__item">Item 2</li>
</ul>
<ul class="list-block">
  <li class="list-block__item list-block__item--active">Item 1</li>
  <li class="list-block__item">Item 2</li>
</ul>
```

BEM is a highly effective naming convention that creates predictably behaving CSS that is easy to manage, maintain and scale. BEM does have downsides, however, including the difficulty in inventing class names for deeply nested objects, the ridiculously long class names and bloated HTML that may sometimes result, and also the lack of consistency that is caused by the inability to share CSS between objects.

## SMACSS

## Scalable and Modular Architecture for CSS

- **In a nutshell:** Split CSS code across multiple files for better performance and organisation

SMACSS works by dividing CSS into five categories – base, layout, module, state and theme – commonly split into separate files. Base styles override the default styles and are mainly applied using element selectors:

```
h1 { font-size: 20px; }  
a { text-decoration: none; }
```

Layout styles are for major objects like headers and sidebars. They are applied using IDs or classes with generic helper declarations optionally prefixed with **l-**:

```
#header { height: 50px; }  
.l-right { float: right; }
```

Module styles are for smaller, reusable objects like buttons and lists, each commonly with its own file. They are applied using classes, with nested items classes commonly prefixed with the ancestor class:

```
.list { ... }  
.list--icon { ... }  
.list--text { ... }
```

State styles are for changeable states, like hidden or disabled. They are commonly applied with class names prefixed with **is-** or **has-** and chained to other selectors:

```
.button { ... }  
.button.is-disabled { ... }
```

Theme styles are optionally used for changing the visual scheme. SMACSS provides well-organised CSS code split logically across multiple files. Using SMACSS does, however, introduce specificity traps by allowing IDs and relying on selector chaining for state and some layout declarations.

**Next page: Learn how to combine different methodologies to create your own**

## Get Weekly Tips and Inspiration

Sign up below to get the latest from Creative Bloq, plus exclusive special offers, direct to your inbox!

[SIGN ME UP](#)

- ☐ Contact me with news and offers from other Future brands
- ☐ Receive email from us on behalf of our trusted partners or sponsors

By submitting your information you agree to the [Terms & Conditions](#) and [Privacy Policy](#) and are aged 16 or over.

---

## Ahmed Abuelgasim

Ahmed is a front-end developer at Potato London

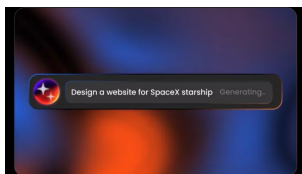
---

### TOPICS

[CSS](#)[WEB DESIGN](#)[MAGCONTENT](#)[WEB DESIGNER](#)[WEB DESIGN](#)

---

### RELATED ARTICLES



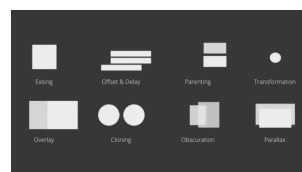
Yikes, now AI can design a whole website from a text prompt



The best PS5 external hard drives in May 2023



Sign up for our essential online UX design course: UX Design Foundations



This handy UX design cheatsheet is invaluable

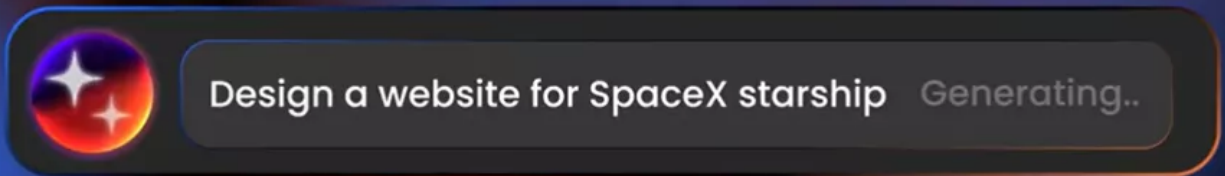
## RECOMMENDED

## The best drawing tablets: The best graphics tablets in M05 2023

### How to make and sell an NFT

MOST READ

MOST SHARED



1 Yikes, now AI can design a whole website from a text prompt

2 "Uncommonly good" video game logo design drives Reddit wild

3 Can you spot the mistake in this iconic Nintendo logo?

4 Unleash Your Inner Superhero with Filmora's Superhero Effects Pack

5 Seriously, this really is Sony's hyped PS5 handheld design













Creative Bloq is part of Future plc, an international media group and leading digital publisher. **Visit our corporate site.**

---

[About us](#)[Contact Future's experts](#)[Terms and conditions](#)[Privacy policy](#)[Cookies policy](#)[Advertise with us](#)[Accessibility Statement](#)[Careers](#)

---

© Future Publishing Limited Quay House, The Ambury, Bath BA1 1UA. All rights reserved. England and Wales company registration number 2008885.