


Join GitHub today

GitHub is home to over 40 million developers working together to host and review code, manage projects, and build software together.

[Sign up](#)[Dismiss](#)

Branch: master ▾ curriculum / 04-advanced-ui / 01_advanced_css.md

[Find file](#)[Copy path](#) Xaviju Add image

23f6832 11 days ago

[1 contributor](#)

103 lines (78 sloc) 3.44 KB

[Raw](#)[Blame](#)[History](#)

Advanced CSS

Custom Properties

Custom properties (sometimes referred to as CSS variables) are variables defined by CSS authors that contain specific values to be reused throughout a document (similar to JS variables). They are set using custom property notation (e.g., `--main-color: black;`) and are accessed using the `var()` function (e.g., `color: var(--main-color);`).

Complex websites have very large amounts of CSS, often with a lot of repeated values. For example, the same color might be used in hundreds of different places, requiring global search and replace if that color needs to change. Custom properties allow a value to be stored in one place, then referenced in multiple other places.

Using variables

```
.parent-element {
  --main-color: #000;
}

.child-element {
  background: var(--main-color);
}

/* Custom properties using fallback value */
.sibling-element[attr] {
  background: var(--main-color, #fff);
}
```

Updating the value from javascript

```
const element = document.querySelector('.parent-element');
element.style.setProperty("--main-color", "pink");
```

Class naming methodology [BEM]

On smaller sites, how you organize your styles isn't usually a big concern. However, in larger projects, how you organize your code is key. Nowadays, most of the companies use at least one method of CSS naming methodology to ensure performance, structure and maintainability.

BEM is the abbreviation of **Block, Element, Modifier**:

- **Block**: Standalone entity that is meaningful on its own (header, menu, input),
- **Element**: A part of a block that has no standalone meaning and is semantically tied to its block. (header -> title, menu -> item)
- **Modifier**: Use them to change appearance or behavior. (header -> title -> active, menu -> item -> highlighted)

Using BEM

```
<button class="button">
  Normal button
</button>
<button class="button button--state-success">
  Success button
</button>
<button class="button button--state-danger">
  Danger button
</button>

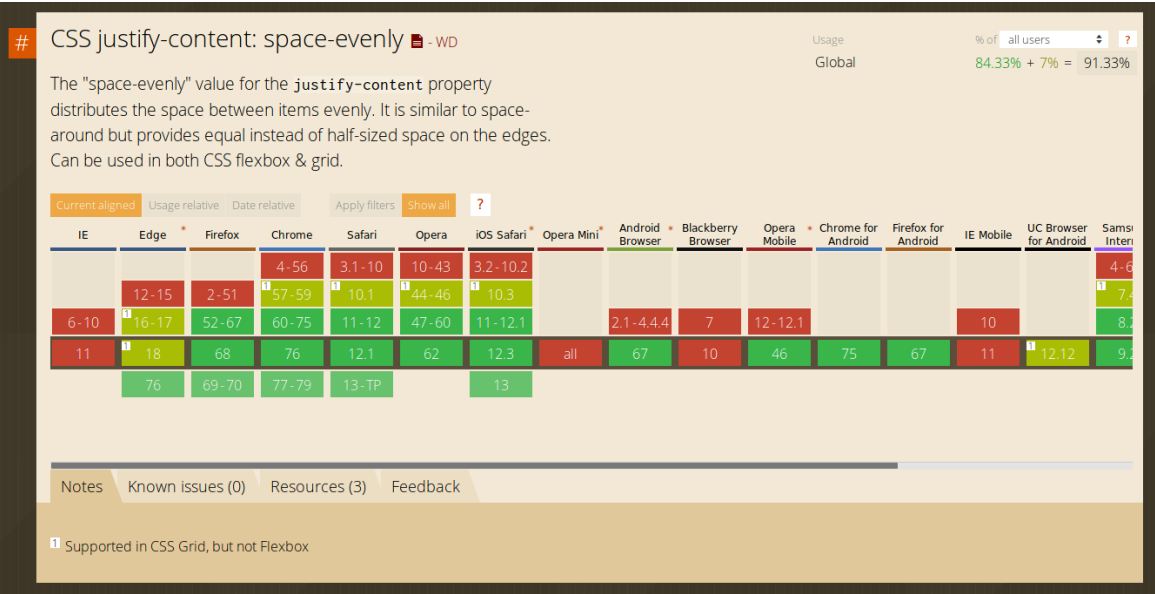
.button {
  display: inline-block;
  border-radius: 3px;
  padding: 0.5rem 0.75rem;
  border: 1px solid #D5D5D5;
  background-image: linear-gradient(#EEE, #DDD);
  font: 700 13px/18px Helvetica, arial;
}
.button--state-success {
  color: #FFF;
  background: #569E3D linear-gradient(#79D858, #569E3D) repeat-x;
  border-color: #4A993E;
}
.button--state-danger {
  color: #900;
}
```

Further reading:

- [Bem Documentation](#)
- [Bem Methodology for small projects](#)

CanIuse && Feature Queries

Not every CSS and JS feature is supported for every browser. Some projects may require to support an old version of a browser and we might not be able to use this feature. It is very important to ensure that the browsers we expect our customers are going to use are supported by our code. "Can I use" provides up-to-date browser support tables for support of front-end web technologies on desktop and mobile web browsers.



For example, can I use flexbox in every browser? No, some browsers like [Internet Explorer 11](#) do not support [standard CSS flexbox](#)

In case a browser does not support a feature, we can query if the browser does support this feature and offer an alternative:

```
div {
  display: flex;
}

@supports (display: grid) {
  div {
    display: grid;
  }
}
```