CSS variables

Contents

Bootstrap includes many CSS custom properties (variables) in its compiled CSS for real-time customization without the need to recompile Sass. These provide easy access to commonly used values like our theme colors, breakpoints, and primary font stacks when working in your browser's inspector, a code sandbox, or general prototyping.

All our custom properties are prefixed with `bs-` to avoid conflicts with third party CSS.

Root variables

Here are the variables we include (note that the `:root` is required) that can be accessed anywhere Bootstrap's CSS is loaded. They're located in our `_root.scss` file and included in our compiled dist files.

```
{{< root.inline >}}
{{- $css := readFile "dist/css/bootstrap.css" -}}
{{- $match := findRE ":root {([^}]*)}" $css 1 -}}

{{- if (eq (len $match) 0) -}}
{{- errorf "Got no matches for :root in %q!" $.Page.Path -}}
{{- end -}}

{{- index $match 0 -}}

{{< /root.inline >}}
```

Component variables

We're also beginning to make use of custom properties as local variables for various components. This way we can reduce our compiled CSS, ensure styles aren't inherited in places like nested tables, and allow some basic restyling and extending of Bootstrap components after Sass compilation.

Have a look at our table documentation for some [insight into how we're using CSS variables]({{< docsref "/content/tables#how-do-the-variants-and-accented-tables-work" >}}).

We're also using CSS variables across our grids—primarily for gutters—with more component usage coming in the future.

Examples

CSS variables offer similar flexibility to Sass's variables, but without the need for compilation before being served to the browser. For example, here we're resetting our page's font and link styles with CSS variables.

```css
body {
  font: 1rem/1.5 var(--bs-font-sans-serif);
}
a {
  color: var(--bs-blue);
}
```

Grid breakpoints

While we include our grid breakpoints as CSS variables (except for `xs`), be aware that **CSS variables do not work in media queries**. This is by design in the CSS spec for variables, but may change in coming years with support for `env()` variables. Check out this Stack Overflow answer for some helpful links. In the mean time, you can use these variables in other CSS situations, as well as in your JavaScript.