Theme switcher With CSS Variables

Using CSS Variables to creaate themes we should be able to creaate more than one and then switch between them. I've experimented with using multiple roots with different classes and then use javascript to switch the class of the HTML element to match the theme.

I understand the theory but I had a little bit of a hard time understanding how to add/switch a class from the HTML element. Looking at code from @justmarkup article helped clarify the idea.

Initially I had decided to use buttons to make the theme switcher but a select drop down menu look better. We also have to hide one element rather than several if we choose to do so.

The HTML with the switcher element and the content, Lorem Ipsum for now, looks like this.

```
<h1>Theme Switcher Demo!</h1>

<label id="theme-changer" class="hidden">
  Choose theme
  <select name="theme" id="theme">
    <option value="default">default</option>
    <option value="theme-blue">blue</option>
    <option value="theme-red">red</option>
    <option value="theme-green">green</option>
    <option value="theme-grey">grey</option>
  </select>
</label>

    <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit.</p>
    <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit.</p>
    <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit.</p>
    <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit.</p>
    <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit.</p>
```

```
    <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit.</p>
    <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit.</p>
    <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit.</p>
    <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit.</p>
```

The CSS is broken in two parts. The first part defines multiple root elements with classes. We'll use these classes to change the currently selected theme from Javascript.

```
:root.default {
  --global-backgroundcolor: white;
  --global-h1-size: 2em;
}

:root.theme-blue {
  --global-backgroundcolor: lightblue;
  --global-h1-size: 3em;
}

:root.theme-red {
  --global-backgroundcolor: indianred;
  --global-h1-size: 2.5em;
}

:root.theme-green {
  --global-backgroundcolor: lightgreen;
  --global-h1-size: 2em;
}

:root.theme-grey {
  --global-backgroundcolor: lightgray;
  --global-h1-size: 4em;
}
```

The second block of Javascript uses the themes we created. We don't specify which team to use but, since all themes have the same basic variables we only use the variable name, Javascript will take care of the switching.

The h1 element has different values depending on the theme we selected. This also allow us to see the differences between themes.

```
body {
  background-color: var(--global-backgroundcolor);
  width: 80%;
  margin: 0 auto;
}

h1 {
  font-size: 3em;
  font-size: var(--global-h1-size);
}

.hidden {
  display: none;
}
```

The Javascript portion of the project is what makes the switch happen. We use three different feature detection tests:

- `window.CSS` checks if the browser implements the CSS Object Model (CSSOM)
- `window.CSS.supports` is the Javascript equivalent to CSS `@supports`
- `windows.CSS.supports('--a', 0)` is the actuall support test

These are called using logical and meaning that they must all return true for the expression to evaluate to true.

We capture different elements (using `document.querySelector`) into constants that we'll use later in the script.

We set the initial theme we'll use to the 'default' theme.

We reveal the theme chooser by removing the `hidden` class from the select element.

We add a change event listener to capture the child element (representing the child theme) we selected, make it the current selection and use it as the new class for our root element.

If we don’t pass the test it’s because the browser doesn’t implement CCSSOM (unlikely), it doesn’t implement **support** or it doesn’t support CSS variables. If any of these conditiona are not met we pop an alert to the user… there’s not much more we can do on the JS side.

```
if (window.CSS && window.CSS.supports && window.CSS.supports('--a', 0)) {
  const themeChanger = document.querySelector('#theme-changer');
  const root = document.querySelector(':root');
  const themeChooser = document.querySelector('#theme');

  root.className = 'default';

  themeChanger.classList.remove('hidden');

  themeChooser.addEventListener('change', function(e) {
    selectOption = this.options[this.selectedIndex];
    currentTheme = selectOption.value;
    root.className = currentTheme;
  });
} else {
  alert("You browser doesn't support CSS custom variables yet");
}
```

This is a very simple proof of concept with a very big omission:

As implemented, the script can’t save the theme selection for resumption on future visits; you’ll have to select your theme everytime you visit. This could be solved with either Local Storage or Indexed DB.

Full working example on this pen at Codepen.