JavaScript: how to add CSS styles dynamically

Sometimes you need to change the CSS styles of a page dynamically. JavaScript comes handy in such case.

Let's start with a typical minimal test case with the following markup structure.

```
<p>This should be green</p>
<script><!-- JavaScript code --></script>
```

The first approach we can try is adding a `style` element just before the target element. We can accomplish this task in this way:

```
const p = document.querySelector('p');
const style = document.createElement('style');

style.textContent = 'p { color: green; }';

document.body.insertBefore(style, p);
```

The contents of the `style` element are parsed as CSS code right away and the style rule is applied because it comes before the target element.

This approach is useful when you just have to modify a couple of styles and you don't want to use the `style` property several times on the same element.

But when you have to apply multiple style rules with a certain degree of complexity, a much better approach is inserting a new stylesheet directly into the `head` element, like so:

```
const link = document.createElement('link');
link.href = 'style.css';
link.rel = 'stylesheet';
link.media = 'screen';
link.type = 'text/css';

document.head.appendChild(link);
```

Both approaches are valid, provided that your JavaScript code is executed just after the document structure is complete. It's always a better solution to insert our scripts just before the closing `body` tag in order to maximize and optimize performance.