

JavaScript Day - 14

DOM Styles and Classes

Styles and Classes

There are generally two ways to style an element:

- Create a class in CSS and add it: `<div class="...">`
- Write properties directly into style: `<div style="...">`.

JavaScript can modify both classes and style properties.

We should always prefer CSS classes to style. The latter should only be used if classes “can’t handle it”

className and classList

Changing a class is one of the most often used actions in scripts.

If we assign something to `elem.className`, it replaces the whole string of classes. Sometimes that's what we need, but often we want to add/remove a single class.

There's another property for that: `elem.classList`.

The `elem.classList` is a special object with methods to add/remove/toggle a single class.

So we can operate both on the full class string using `className` or on individual classes using `classList`. What we choose depends on our needs.

className and classList

Methods of classList:

- `elem.classList.add/remove("class")` – adds/removes the class.
- `elem.classList.toggle("class")` – adds the class if it doesn't exist, otherwise removes it.
- `elem.classList.contains("class")` – checks for the given class, returns true/false.

className and classList

Besides, classList is iterable, so we can list all classes with for..of, like this:

```
<script>
  for (let name of document.body.classList) {
    alert(name); // main, and then page
  }
</script>
```

Element style

The property `elem.style` is an object that corresponds to what's written in the "style" attribute. Setting `elem.style.width="100px"` works the same as if we had in the attribute style a string `width:100px`.

For multi-word property the camelCase is used:

```
1 background-color => elem.style.backgroundColor
2 z-index         => elem.style.zIndex
3 border-left-width => elem.style.borderLeftWidth
```