Summary DOM modification techniques

Hide tutorial navigation

Modifying an existing element

We covered various ways that you can modify aspects of an existing element:

Modifying attributes

You can set an attribute on an element by setting the property of the same name. For example, to change the `src` of an `<img>`:

```
imgEl.src = "http://www.dogs.com/dog.gif";
```

In addition, you can also use the `setAttribute` method, like so:

```
imgEl.setAttribute("src", "http://www.dogs.com/dog.gif");
```

If you want to remove an attribute, you should do that with `removeAttribute` - like to remove the `disabled` attribute off a button, effectively enabling it:

```
imgEl.removeAttribute("disabled");
```

Modifying styles

You can change styles just like how you change attributes, by accessing the `style` property of the element, and setting the corresponding property. For example, to change the color:

```
headingEl.style.color = "hotpink";
```

Remember to "camelCase" the multi-word CSS property names, since hyphens are not valid JS property names:

```
headingEl.style.backgroundColor = "salmon";
```

Modifying class names

To add a class to an element, you can set the `className` property:

```
mainEl.className = "warning";
```

That will override any existing classes, so you should do this instead if you just want to add to the list of classes:

```
mainEl.className += " warning";
```

In newer browsers, you can use the `classList` functionality instead:

```
mainEl.classList().add("warning");
```

Modifying inner HTML / text

To completely replace the contents of an element with an arbitrary string of HTML, use `innerHTML`:

```
mainEl.innerHTML = "cats are the <strong>cutest</strong>";
```

If you don't need to pass in HTML tags, you should use `textContent` instead:

```
mainEl.textContent = "cats are the cutest";
```

Generally, you should be careful when using either of these 2 properties, because they will also remove event listeners (which we teach in the next tutorial).

Creating elements from scratch

There is a whole set of functions that you can use to create entirely new elements and add them to the page.

To create a new element, use the aptly named `createElement`:

```
var imgEl = document.createElement("img");
```

To append it to the page, call `appendChild` on the target parent element:

```
document.body.appendChild(imgEl);
```

Similarly, you can also use `insertBefore`, `replaceChild`, removeChild`, and` insertAdjacentHTML`.`