Using Javascript to manipulate the DOM

JavaScript gives us multiple ways to insert content into the DOM. One generic and simple way and a more flexible and complicated alternative.

Inserting new elements: The simple version

The simple way to insert content into an existing element is to use `appendChild` to insert the element after the existing children of the specified element

We're using the following HTML element as the host of our new element.

```html
<div id="container"></div>
```

In the JavaScript we capture the container element in a variable and then create a button element.

Next we assign attributes to the button we just created. An `id` and the text that will becoe the label of the button using `innerHTML`.

Finally we attach the button element to the container div using `appendChild`.

```javascript
const container = document.getElementById("container");
const button = document.createElement("button");

button.id = "clicky";
button.innerHTML = "Click Me";

container.appendChild(button);
```

- you can't currently add an attribute to the element when it's created; you must use `setAttribute` or similar method elsewhere on the script
- You may think about adding an `id` attribute to the element, but it's unlikely

that you need to do so. You already have a reference to the element when you create it.

- Only reason why you may need an id or class is if you'll reference the element from a separate script

- Once you add the element to the page, it will follow all the rules you apply via CSS, either a stylesheet, or Javascript

See the following pages on MDN for additional methods to manipulate DOM nodes.

- [Node.removeChild()](#)
- [Node.replaceChild()](#)
- [Node.insertBefore()](#)
- [ParentNode.append()](#)

Placing new elements: insertAdjacentHTML

One disadvantage of appending the child to an existing document is that you don't have a way to choose where the child element is placed.

`insertAdjacentHTML` takes care of this. It takes two parameters: a position string and a string represting the element we want to position at the given location.

The position parameter can be:

- **beforebegin**: Before the element itself.
- **afterbegin**: Just inside the element, before its first child.
- **beforeend**: Just inside the element, after its last child.
- **afterend**: After the element itself.

So `inserAdjacentHTML` gives us the flexibility of putting content where we need it inside the existing elements of the page so we are safe when inserting content into exsiting elements.

The following pen illustrates how the four different values work:

So we've looked at different ways to insert HTML in an existing document using both appendChild and insertAdjacentHTMLL and discussed what they do different. Which one works best will depend on your usecase and how you've structure the document and the script that adds to it.