

CSE 331: Web Development in 2005

The web browser let's the web developer customize the default behavior of certain actions via Javascript. It can

- Access and modify the HTML tree.
- Register callbacks to take actions when buttons are pressed, a timer goes off, etc.
- Send additional HTTP requests to the server and take action on the responses.

The **Document Object Model** (DOM) is an API for Javascript to manipulate the HTML on the page.

- `document.getElementById("foo")` allows Javascript to get a reference to a certain element.
- `x.innerHTML` is a (mutable!) string containing the HTML source code for the children of `x`.
- `document.createElement("tag")` allows creating a new HTML tag element.
- `document.createTextNode("...text...")` allows creates a new text element.
- `x.appendChild(y)` inserts HTML element `y` as a new child of HTML element `x`.

Registering **callbacks**:

- `x.onclick`: a Javascript function to call when the button is pressed.
- For example:

```
x.onclick = () => {
  console.log("clicked!");
}
```

Sending additional HTTP requests:

- `fetch("/path/to/request")` sends a GET request to `/path/to/request`
- You can pass arguments via query parameters: `fetch("/path/to/request?x=10&y=hello")`
- You can send POST requests by passing a configuration object as the second argument to `fetch`:

```
fetch("/path/to/request", {
  method: "POST",
  headers: { "Content-Type": "application/json" },
  body: JSON.stringify({ x: 10, y: "hello" })
});
```

Promises and **async/await**:

- The return value from `fetch` is *not* the response, but rather a **Promise** to return the response later.
- You can **await** the **Promise** to actually get the value.
- **await** can only be called if the surrounding function is declared **async** (!!!)
- The response can be parsed as JSON with the `.json()` method, which *also* returns a **Promise**.
- Full example of fetching:

```
async function foo() {
  const response = await fetch(...);
  const actualData = await response.json();
  // do something with actualData
}
```