Basic Javascript

The code I demonstrated in class is on gitlab: jsdemo.html and jsdemo.js.

To include scripts on a page, we can either embed the script directly:

```
<script type="text/javascript">
   console.log("Hello from JS!")
</script>
```

or reference an external file:

```
<script src="jsdemo.js"></script>
```

When you do `console.log(args)`, it dumps the contents of `args` into the page's console, which you can see in the browser's developer tools (probably `control-shift-I` in Chrome).

The basic syntax of Javascript is a lot like C, except variable declarations don't have types so just use the keyword `var`:

```
var x = 0; // Sum all integers from 1 to 100
for(var i = 1; i <= 100; i++)
{
    x = x + i;
}
console.log("The sum is", x)
```

Document Object Model

The elements on the page are accessed using the **DOM,** which stands for Document Object Model. For example, if your HTML has an element with a particular ID:

```
<p id="p3">Hello world</p>
```

Then you can get a handle to it from Javascript using:

```
var p3 = document.getElementById("p3")
```

And here are some examples of things you can manipulate:

```
p3.innerHTML = "Some new HTML content.";
p3.style.backgroundColor = "#f00"
```

The jsdemo files linked above show an example where a paragraph can be clicked to change its content. The key to that is to put the manipulations into a function definition:

```
var hijack = false
function clickedParagraph()
{
    hijack = !hijack
    var p3 = document.getElementById("p3")
    if(hijack) {
        p3.innerHTML = "HAHAHA, I hijacked your page's content!!!! Brwhahahah.";
        p3.style.backgroundColor = "#f00"
    }
    else {
        p3.innerHTML = "This is a static paragraph again.";
        p3.style.backgroundColor = "#fff"
    }
}
```

and then use the attribute onclick in the HTML element:

```
  <p id="p3" onclick="clickedParagraph()">
    This is a static paragraph.
  </p>
```

Other DOM attributes

Here are just a few other things you can do with the DOM. You can change the source of an image. The element continues to exist in its place on the page, but the image it displays changes. Here is the HTML followed by the Javascript:

```
<img id="mypic" src="profile.jpg">
```

```
var mypic = document.getElementById("mypic")
mypic.src = "kittens.jpg"
```

You can access child elements with a children array:

```
<ul id="cities">
  <li>Brooklyn</li>
  <li>Chicago</li>
  <li>Los Angeles</li>
</ul>
```

```
var cities = document.getElementById("cities")
cities.children[0].setAttribute("class", "active")
cities.children[2].setAttribute("class", "disabled")
```