Introductory Web Programming

What this class is

This is an overview of JavaScript and how it interacts with the browser to make interactive websites

What you'll learn

- A simple overview of JavaScript
- An explanation of the Document Object Model
- An overview of the event system in JavaScript
- Simple examples of writing code

What you'll learn

- A simple overview of JavaScript
- An explanation of the Document Object Model
- An overview of the event system in JavaScript
- Simple examples of writing code

What you'll learn

- A simple overview of JavaScript
- An explanation of the Document Object Model
- An overview of the event system in JavaScript
- Simple examples of writing code

What you'll learn

- A simple overview of JavaScript
- An explanation of the Document Object Model
- An overview of the event system in JavaScript
- Simple examples of writing code

What you won't learn

- The basics of HTML and CSS
- The full JavaScript programming language
- Libraries like jQuery or Angular or React

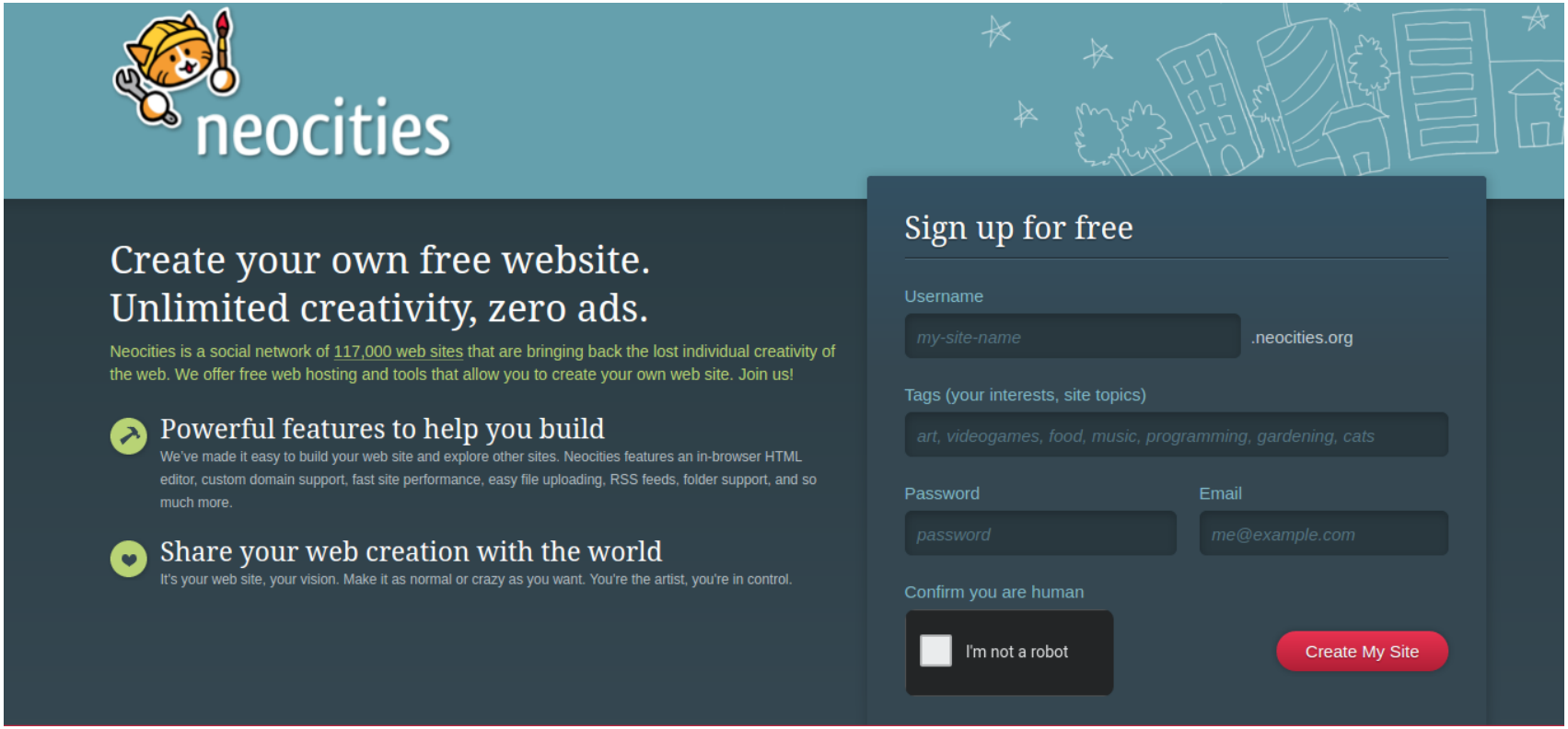
What you won't learn

- The basics of HTML and CSS
- The full JavaScript programming language
- Libraries like jQuery or Angular or React

What you won't learn

- The basics of HTML and CSS
- The full JavaScript programming language
- Libraries like jQuery or Angular or React

Make an account



HTML in a nutshell

```
<!doctype html>
<html>
  <body>
    <h1>This is a heading</h1>
    <p>
      This is a paragraph of text,
      where some of the text is <b>bold</b> and
      after this paragraph, there will be a numbered list
    </p>

    <ol>
      <li>lists are made of "list items"</li>
      <li>like these</li>
    </ol>
  </body>
</html>
```

CSS in a nutshell

Selectors and Properties

```
selector {
    property: value;
    property: value;
    property: value;
}
```

CSS in a nutshell

Selectors and Properties

```
selector {
    property: value;
    property: value;
    property: value;
}
```

CSS in a nutshell

Selectors and Properties

```
selector {
    property: value;
    property: value;
    property: value;
}
```

What's a programming language?

Programming languages are languages for describing computation

JavaScript

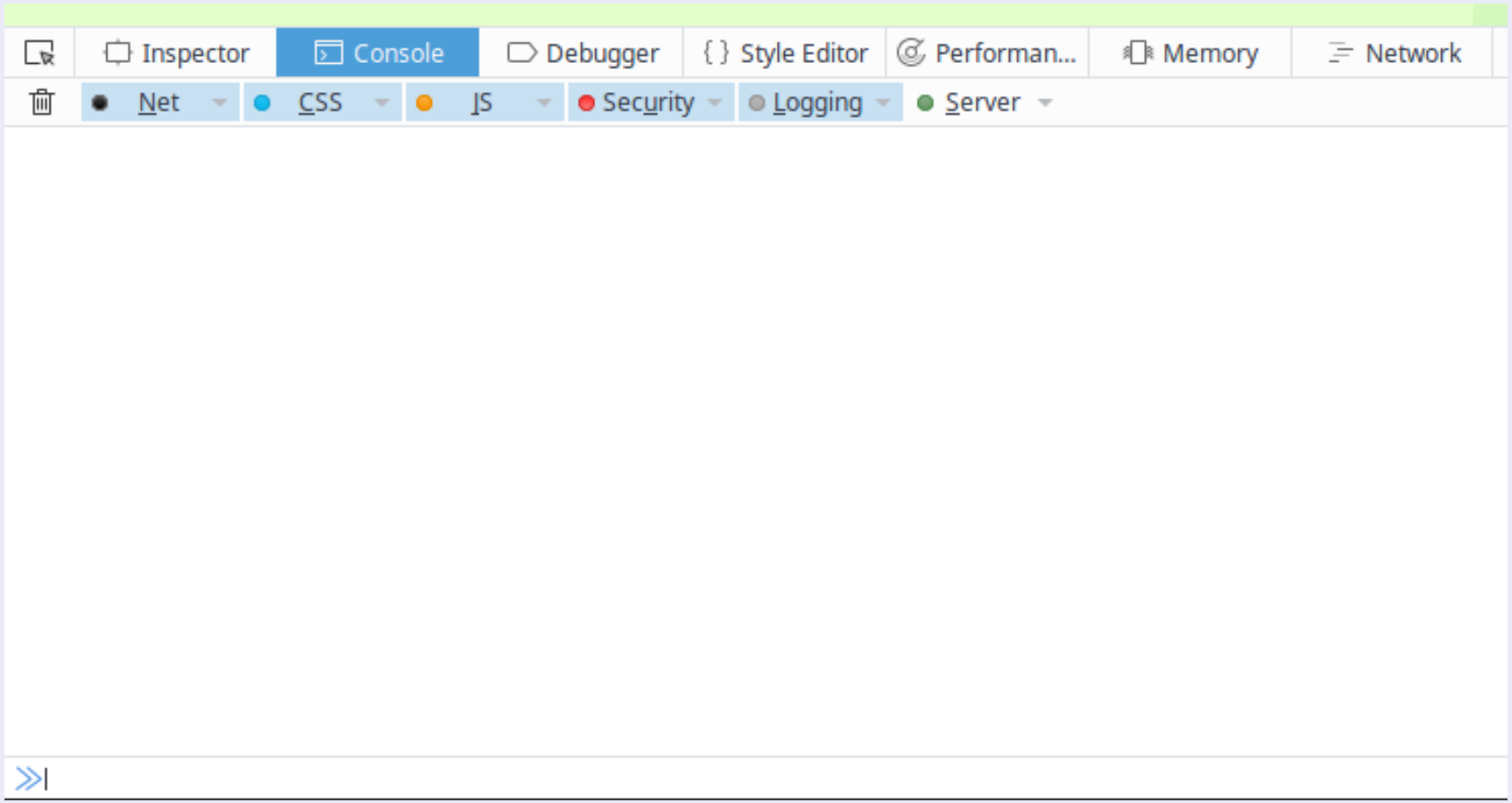
JavaScript has a special relationship to the browser

Learning programming

Learning programming requires specificity that's unnatural.
It's always hard at first.

The console

Ctrl-Shift-K



Syntax and Semantics

Every programming language has a *grammar* and it has to be followed exactly

Numbers and Strings

Numbers

```
10
20 + 30
0 * 3 + 0.2
```

Strings

```
"text"
'also text'
"this is an 'inner quote'"
```

Variables

Variables are generalized pronouns: names that can refer to data

Type into the console

```
var thisIsAName = 10
thisIsAName
thisIsAName = 20
thisIsAName
```

Variables exercise

Try for yourself

- Create a variable in the console
- Assign it a number
- Retrieve the data out of the variable

Loading code into the browser

```
<head>
  <script>
    ...
  </script>
</head>
```

```
<head>
  <script src="yourCode.js">
  </script>
</head>
```

Sequencing code

To write multiple steps of code, separate them with by semicolons.

```
var thisIsAName = 10;
thisIsAName;
thisIsAName = 20;
thisIsAName;
```

Functions

Functions hold sequences of code that can be called again and again.

```
function funName (arg1) {
    var arg2 = arg1;
    return arg2;
}

console.log(funName(10));
```

Functions exercise

Try for yourself

1. open a new file, call it `funEx.js`
2. In this file:
   - Create a function that
     - takes three arguments
     - multiplies all of them together
     - uses `console.log` to write the value to the console
   - Call the function with three arguments
3. Create an HTML file that looks like the following

Example HTML

```
<html>
  <head>
    <script src="funEx.js"></script>
  </head>
</html>
```

Objects

Objects are ways of holding data like dictionaries or contact lists: key/value pairs

Type along

```
var obj = { property1: 10,
            property2: 20}

console.log(obj);
console.log(obj.property1);
console.log(obj.property2);
```

The pineal gland of the browser

The Document Object Model (DOM) connects the webpage you can see with the JavaScript code

The DOM

The DOM provides the *state of the webpage* as an *object* you can manipulate

Creating elements

```
var newElement = document.createElement("h1");
var textInTheThing =
  document.createTextNode("There's some text in here.");
newElement.appendChild(textInTheThing);

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

Type the following into a file called LoadTest.html

```
<html>
  <head>
    <script>
       var newElement = document.createElement("h1");
       var textInTheThing =
         document.createTextNode("There's some text in here.");
       newElement.appendChild(textInTheThing);

       document.body.appendChild(newElement);
    </script>
  </head>
  <body>
  </body>
</html>
```

Making sure it loads

You probably saw something like

What went wrong?

Loading order

The code needed to run after, not before, `body` existed

A taste of events

```
<html>
  <head>
    <script>
        window.onload = function () {
          var newElement = document.createElement("h1");
          var textInTheThing =
            document.createTextNode("There's some text in here.");
          newElement.appendChild(textInTheThing);

          document.body.appendChild(newElement);
        }
    </script>
  </head>
  <body>
  </body>
</html>
```

How to change elements

To change elements, you usually will change their CSS classes

Adding and removing classes

```
var newElement = document.createElement("h1")
    .appendChild(document.createTextNode("Text here"));
document.body.appendChild(newElement);
newElement.classList.add("superbold");
```

Finding elements

In order to modify already existing elements, we need to find them first

Finding elements by id

```
var aHeading = document.getElementById("heading");
aHeading.classList.add("superbold");
```

Finding elements by class

```
var listItems = document.getElementsByClassName("listy");

for(var i=0; i < listItems.length; i = i + 1){
    listItems.item(i).classList.add("superbold");
 }
```

JavaScript event model

Clicking, typing, moving the mouse, etc. create *events*

Events are sensors

Events are the *senses* of your code

Event handlers

```
window.onload = function () {
    document.body
    .addEventListener("mousemove",console.log);
}
```

Attaching events to buttons

```
<html>
  <head>
    <script>
       window.onload = function () {
           var button = document.getElementById("button");
           button.addEventListener("click", function () {
              console.log("the button was pressed");
           });
       }
    </script>
  </head>
  <body>
    <button id="button">Push me</button>
  </body>
</html>
```

Retrieving data from forms

```
<html>
  <head>
  <script>
    window.onload = function () {
      var button = document.getElementById("button");
      button.addEventListener("click", function () {
          console.log("the button was pressed");
      });
    }
  </script>
  </head>
  <body>
    <input id="input" value="stuff"/>
    <button id="button">Press me</button>
  </body>
</html>
```

Using the Mozilla Developer Network

`https://developer.mozilla.org/en-US/`

Questions?

Thank you for coming out!