Introduction to JavaScript

Class 1

Welcome Ladies

Girl Develop It is here to provide affordable and accessible programs to learn software through mentorship and hands-on instruction.

Some "rules"

- We are here for you!
- Every question is important.
- Help each other.
- Have fun.

I am Denise Nguyen

I am here because I excited to help you ladies learn more about developing and programing!

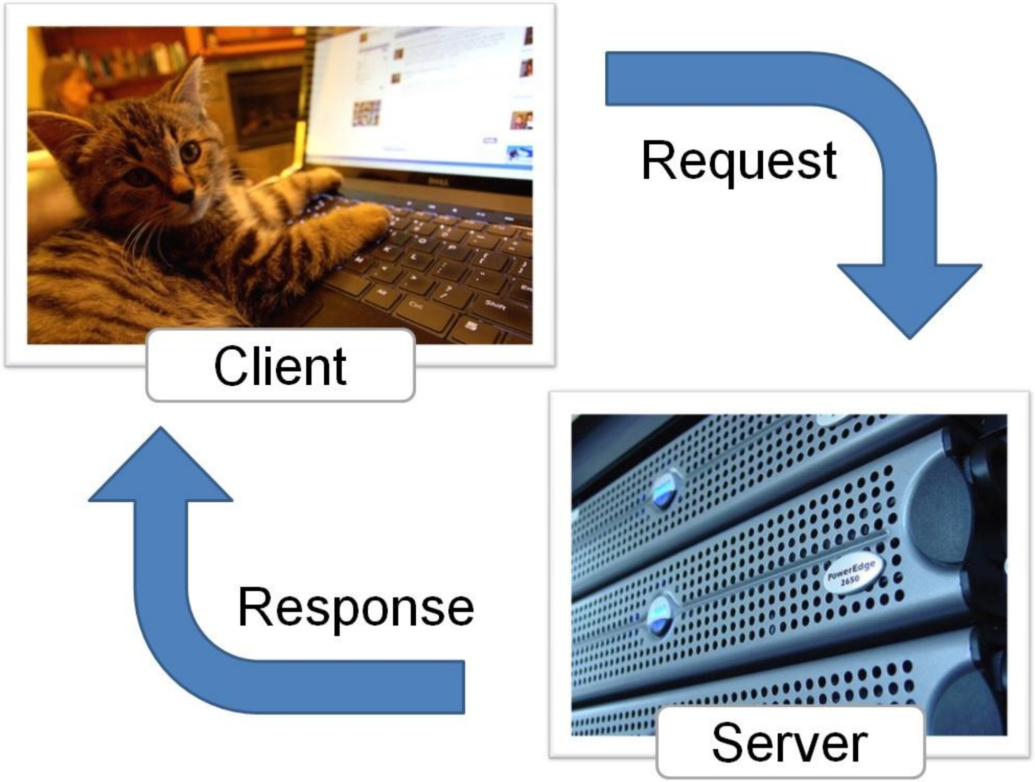
Tell us about yourself

- ★ Who are you?
- ★ What do you hope to get out of the class?
- ★ What's your would your superpower be?

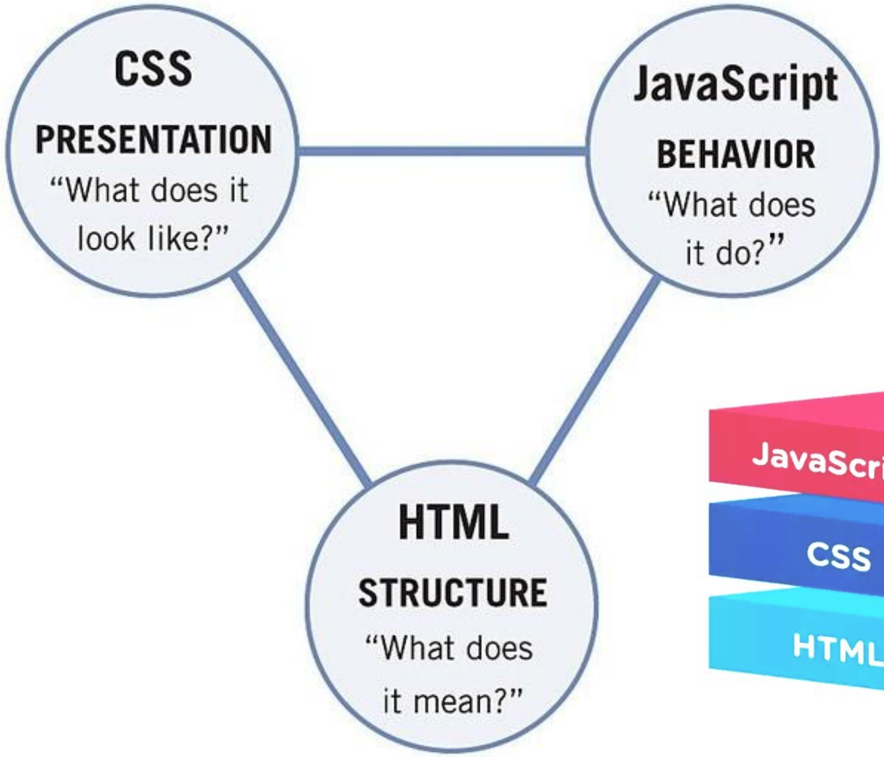
1

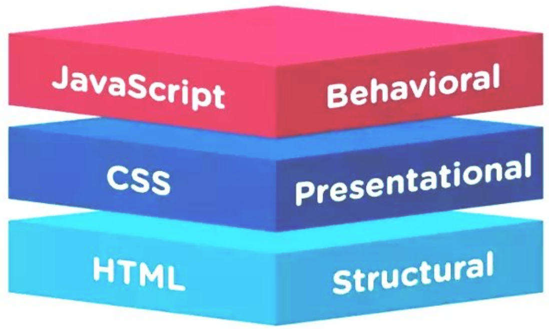
What is JavaScript?

JavaScript is not Java



JavaScript is the language of the web





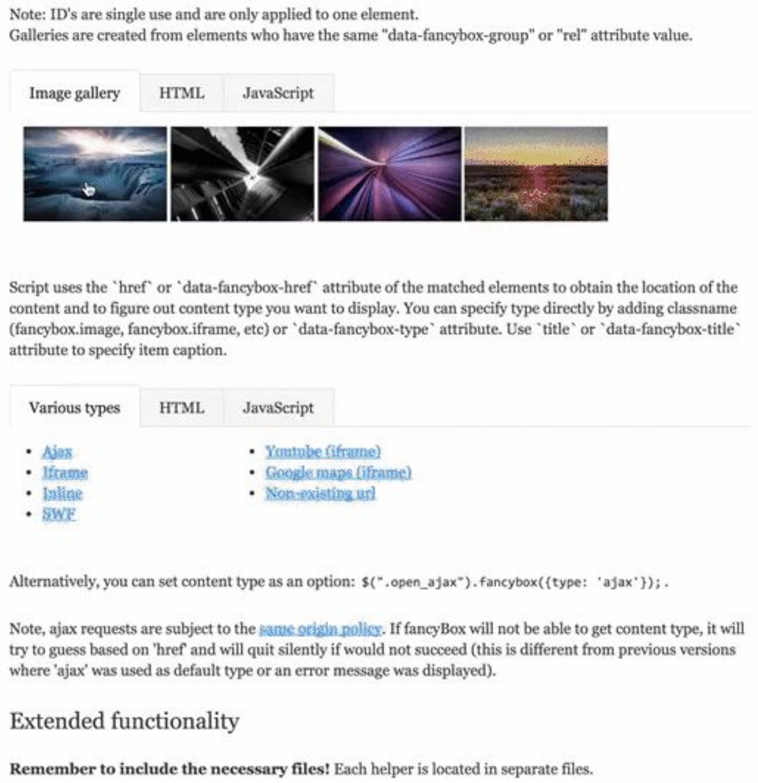
JavaScript works with HTML & CSS

What is JavaScript?

- Created by Brendan Eich as "LiveScript" in 1995, which got renamed to "JavaScript"
- Standardized by the ECMAScript specifications. This class covers ES5 (standardized in 2009)
- A client-side processing language. A browser reads the code and runs it directly.
- Interfaces with HTML & CSS.
- Lets you build dynamic web pages that respond to input from users.

What can JavaScript do?

Let's find out!



Note: ID's are single use and are only applied to one element.
Galleries are created from elements who have the same "data-fancybox-group" or "rel" attribute value.

Script uses the `href` or `data-fancybox-href` attribute of the matched elements to obtain the location of the content and to figure out content type you want to display. You can specify type directly by adding classname (fancybox.image, fancybox.iframe, etc) or `data-fancybox-type` attribute. Use `title` or `data-fancybox-title` attribute to specify item caption.

- Ajax
- Iframe
- Inline
- SWF
- Youtube (iframe)
- Google maps (iframe)
- Non-existing url

Alternatively, you can set content type as an option: `$(".open_ajax").fancybox({type: 'ajax'});`.

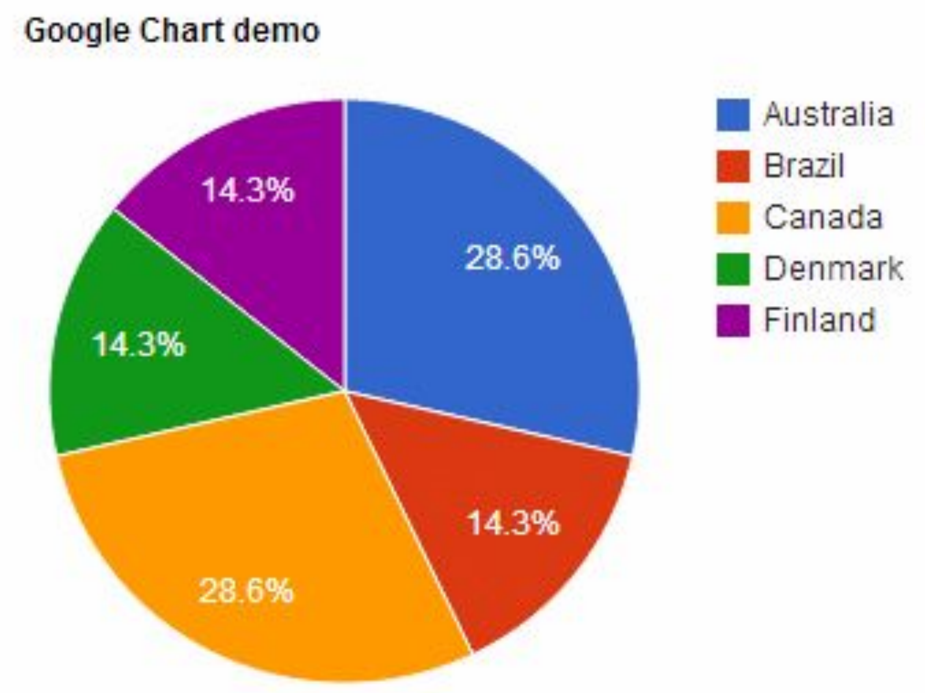
Note, ajax requests are subject to the same origin policy. If fancyBox will not be able to get content type, it will try to guess based on 'href' and will quit silently if would not succeed (this is different from previous versions where 'ajax' was used as default type or an error message was displayed).

Extended functionality

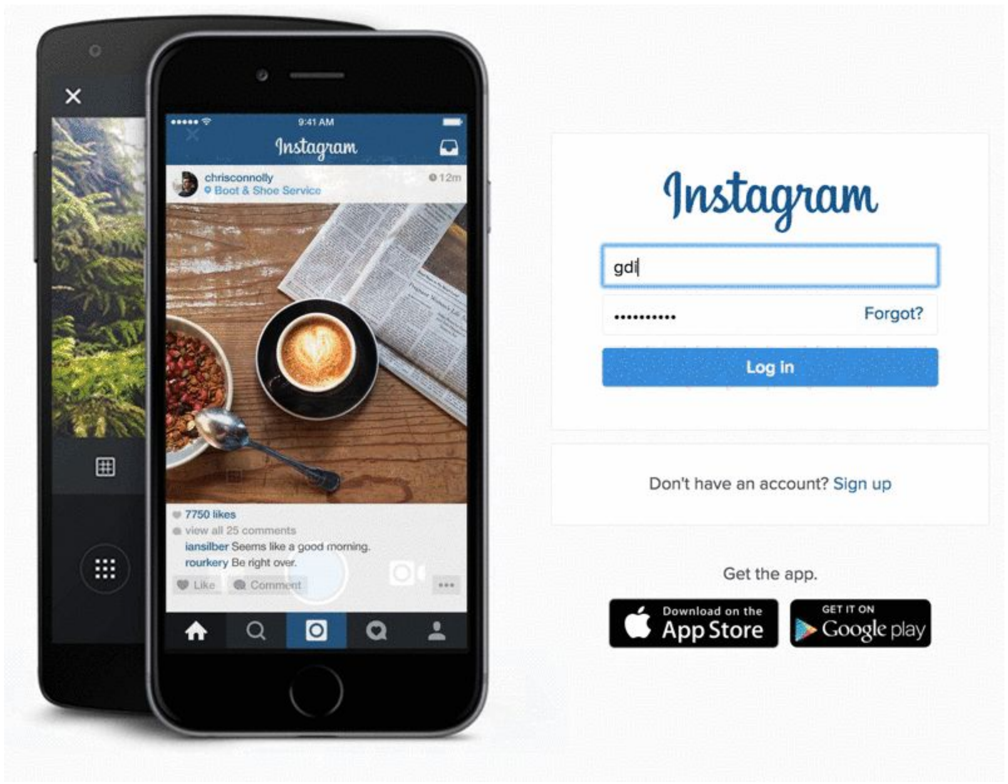
Remember to include the necessary files! Each helper is located in separate files.

Image Lightboxes

Girl Develop It
don't be shy. develop it.



Google Charts API



Instagram

gdi

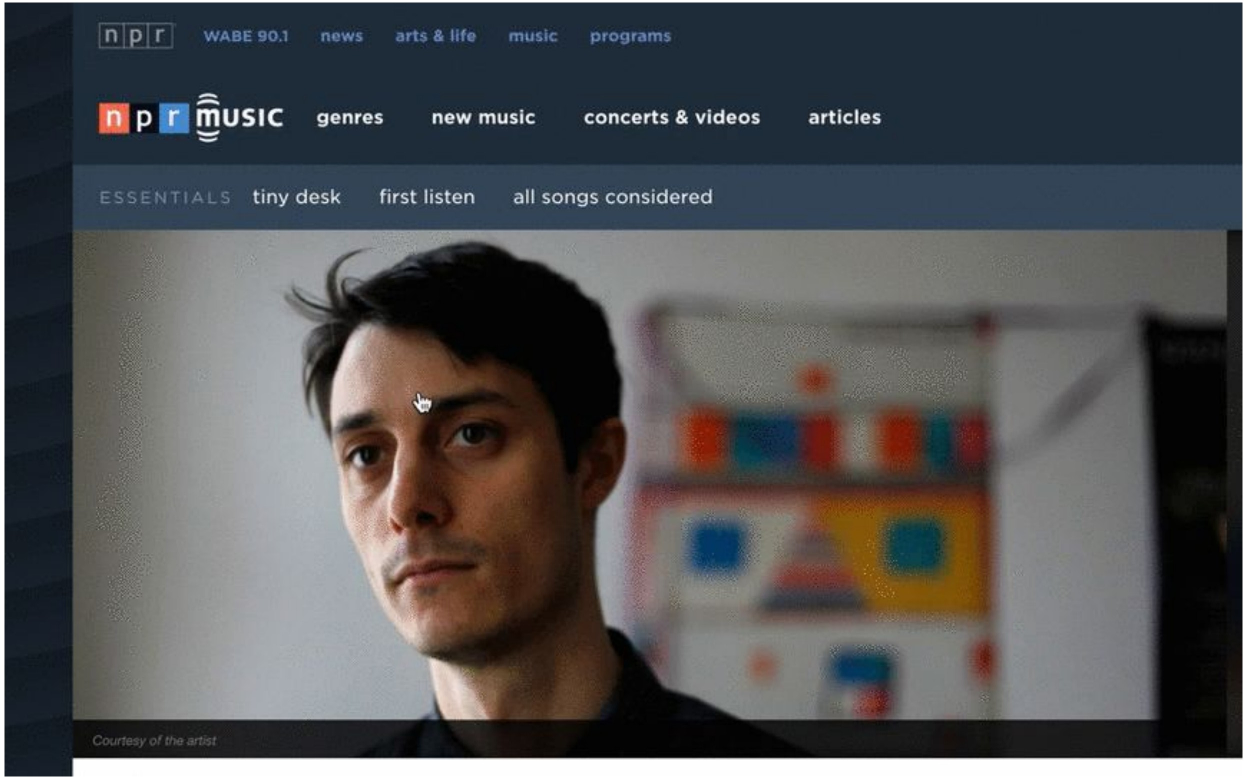
••••••••••• Forgot?

Log in

Don't have an account? Sign up

Get the app.

Keep track of users with **Cookies** or storing data with **local storage**.



Interactive elements like tabs, sliders, etc.

Script Tags

You can mix JavaScript and HTML. The **script tag** tells your browser the stuff inside is code, not content.

```
<script>
  CODE GOES HERE
</script>
```

JavaScript Files

Just like CSS, you can split a long block of JavaScript into its own file.

```
<script src="path/to/file.js"></script>
```

Let's Develop It

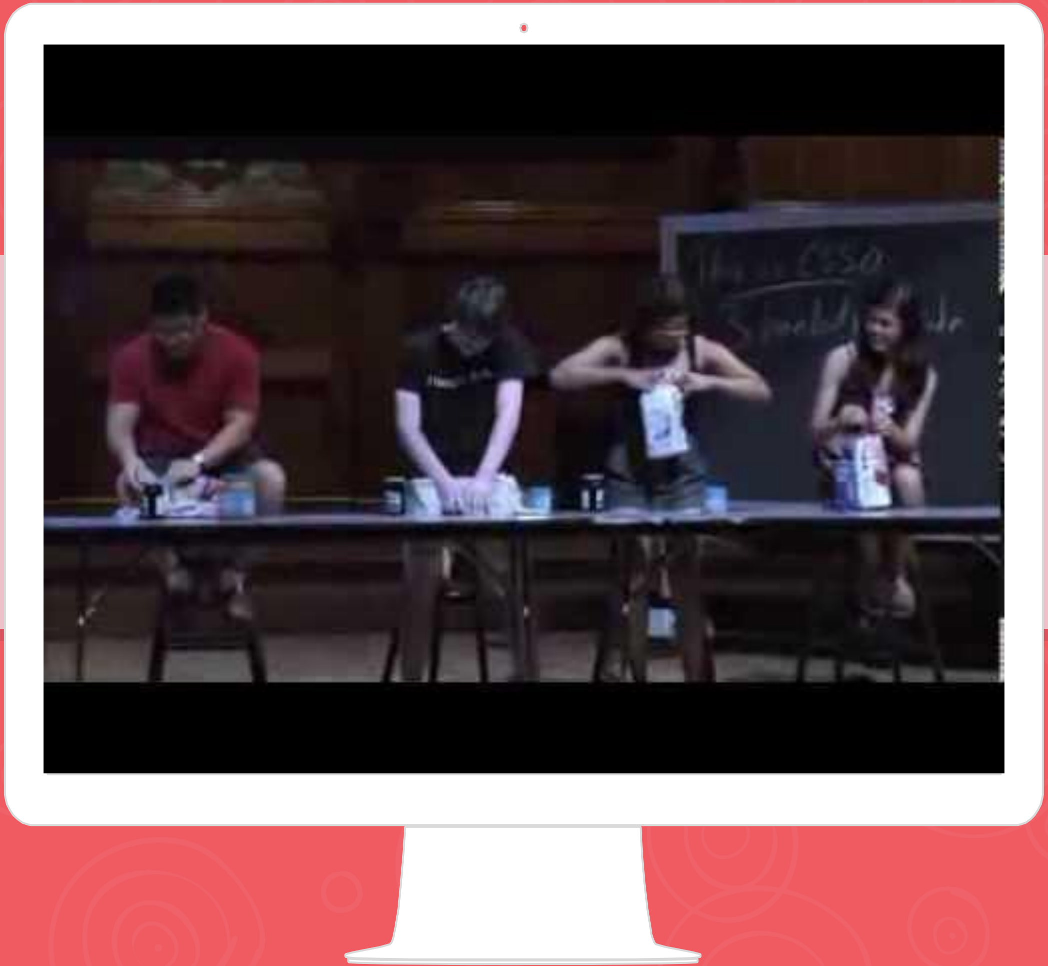
- Make a folder called `gdi`.
- Inside, make a new page called `index.html`.
- Write this code inside.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Test Page</title>
  </head>
  <body>
   <p>This is my awesome JavaScript Code.</p>
    <script>
      alert('Hello World!');
      console.log('Secret message');
    </script>
  </body>
</html>
```

Computers need simple, clear instructions

- Computers are great at ***processing***. They are bad at ***understanding***.
- When you write a program, you must break down every step into simple pieces.

Example: Make a Sandwich

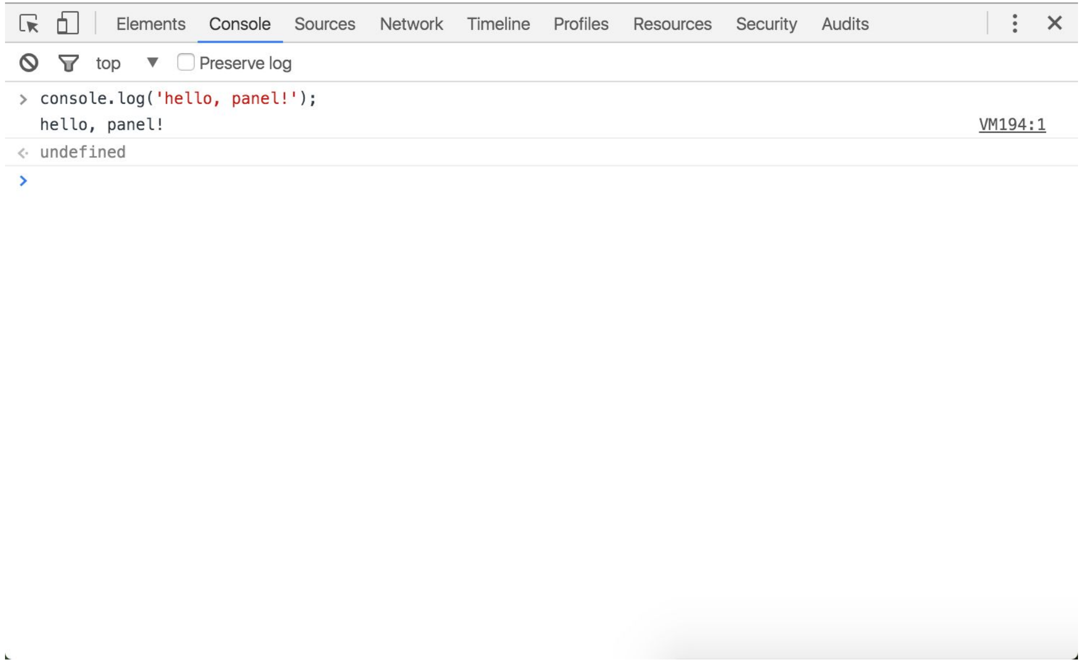


How does JavaScript work?

1. You visit a website with JavaScript code on it.
2. Your browser (e.g., Chrome) reads the code line-by-line.
3. The browser runs each line of code as it reads it.
4. Based on these instructions, the browser performs calculations and changes the HTML and CSS on the page.
5. If the browser finds code it doesn't understand, it stops running and creates an error message.

Console

You can see what's going on in the **console**.



```
> console.log('hello, panel!');
  hello, panel!                                        VM194:1
<· undefined
>
```

Let's Develop It

Open the console.

In Chrome, use the keyboard shortcut:

Mac: Command + Option + J

Windows: Control + Shift + J

Open your practice page.

Do you see anything in the console?

Try typing in 2 + 2 and hitting enter.

Statements

Each instruction in JS is a "statement", like:

```
console.log('Hello World!');
console.log('I am glad to meet you');
console.log('I am fuzzy');
```

Comments

You can leave comments in your code—notes that people can read but

```
/*
I can make long comments
with multiple lines here
*/
console.log('Hello World!'); // Or make short comments
here
```

Getting results onto your screen

Open a popup box.

```
alert('Hello World!');
```

Display a message in your console.

```
console.log('Hello World!');
```

Add something to the page.

```
document.write('Hello World!');
```

Let's Develop It

- ▷ Open `index.html`.
- ▷ Add a comment to the code.
- ▷ Try different ways of printing a message.
- ▷ Create a new file called `mycode.js.`
- ▷ Move your code to this file and link it to your page.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Test Page</title>
  </head>
  <body>
   <p>This is my awesome JavaScript Code.</p>
    <script>
      alert('Hello World!');
      console.log('Secret message');
    </script>
  </body>
</html>
```

Variables

Just like '**x**' in algebra, a variable is a named container for a value that can change.

Declaring a Variable

- To declare (create) a variable, just type the word `var` and the variable name.

```
var numberOfKittens;
```

- It is a good idea to give your variable a starting value. This is called initializing the variable.

```
var numberOfKittens = 5;
```

Variable Values

- When you first create a variable, it does not have a value (it is `undefined`).
- You can set a value for a variable.
- Variables can hold different types of data.
- The value of a variable can change over time.

Naming Variable

- The variable name is case-sensitive.
- A new variable should have a unique name.
- Variable names need to start with a letter, `$`, or `_`.
- Avoid reserved words.
- Choose clarity and meaning for humans to read later.

Using a Variables

Once you have created a variable, you can use it in your code. Just type the name of the variable.

Using a Variable

- To declare (create) a variable, just type the word `var` and the variable name.

```
var numberOfKittens = 5;
console.log(numberOfKittens);
```

Let's Develop It

- In your **JavaScript file**, create a *variable* and give it a valid name and value. Then, display the value.
- You can run it in `mycode.js.`

```
<!DOCTYPE html>
<html>
  <head>
    <title>Test Page</title>
     <!-- LINK FOR JS FILE -->
    <script src="mycode.js"></script>
  </head>
  <body>
   <p>This is my awesome JavaScript Code.</p>
  </body>
</html>
```

Data Types

- **string** string of characters

```
var userName = 'Jane Lane';
```

- **number** integer or floating point

```
var myAge = 30;
```

- **boolean** true or false

```
var myAge = 30;
```

- **boolean** true or false

```
var favoriteThings;
```

- **null** an explicitly empty value

```
var goodPickupLines = null;
```

Numbers

Variables can be numbers, either integers or floats (decimals).

- JavaScript automatically converts integers to floats
- `NaN` = Not-A-Number

```
var numberOfKittens = 5;
var cutenessRating = 9.6;
```

Arithmetic Operators

- Once you have numbers, you can do math with them!

```
var numberOfKittens = 5;
var numberOfPuppies = 4;
var numberOfAnimals = numberOfKittens + numberOfPuppies;
```

Arithmetic Operators

Example	Name	Result
`-a`	Negation	Opposite of `a`
`a+b`	Addition	Sum of `a` and `b`
`a-b`	Subtraction	Difference of `a` and `b`.
`a*b`	Multiplication	Product of `a` and `b`.
`a/b`	Division	Quotient of `a` and `b`.
`a%b`	Modulus	Remainder of `a` divided by `b`.

Let’s Develop It

- Create two variables and try some arithmetic operators. Don't forget to display your results!
- You can run it in `mycode.js.`

```
<!DOCTYPE html>
<html>
  <head>
    <title>Test Page</title>
      <!-- LINK FOR JS FILE -->
     <script src="mycode.js"></script>
  </head>
  <body>
   <p>This is my awesome JavaScript Code.</p>
  </body>
</html>
```

Strings

- Variables can be strings (groups of characters). You put your string in single or double quotes.

```
var kittensName = 'Fluffy';
```

- If you want to use a quote in your string, you'll need to escape it with a backslash.

```
console.log('I\'d like to use an apostrophe');
```

String Operators

- You can put strings together with a +, the concatenation operator.

```
var kittensName = 'Fluffy ';
var fullName = kittensName + 'McDougle';
console.log(fullName); // Outputs 'Fluffy McDougle'
```

String Operators

- You can also use += to add things to the end of a string.

```
var kittensName = 'Admiral ';
kittensName += 'Snuggles';
console.log(kittensName); // Outputs 'Admiral Snuggles'
```

Concatenate

Let's Develop It

- Create two variables, a first name and a last name, and then put them together to make a full name. Don't forget to display your results!
- You can use concatenation to mix strings and numbers. When you do this, JavaScript will treat the number like a string.
- You can run it in `mycode.js.`

```
var numberOfFruit = 6;
var typeOfFruit = 'bananas';
var allTheFruit = 'I have ' + numberOfFruit + ' ' + typeOfFruit + '!';
console.log(allTheFruit);
```

Let’s Develop It

Create a program to calculate the tip at a restaurant. It should:

- Have variables for the bill pre-tip and the tip percentage.
- Calculate the total bill.
- Output a sentence like "Your total bill, with tip, is $14.75".
- Bonus: Use `toFixed()` to round the bill total to 2 decimals.

YOU DID IT!

Any questions?