

Philly Tech Sistas

# Intro to JavaScript

Class 5

# What is JavaScript?

A Quick Overview



# Programming Language Types

Scripting  
JavaScript  
ReactJS  
Python

```
var x, y, z;  
x = 5;  
y = 6;  
z = x + y;
```

Tagging Language  
HTML  
XML

```
<h1>This is a Heading</h1>  
<p>This is a paragraph.</p>
```

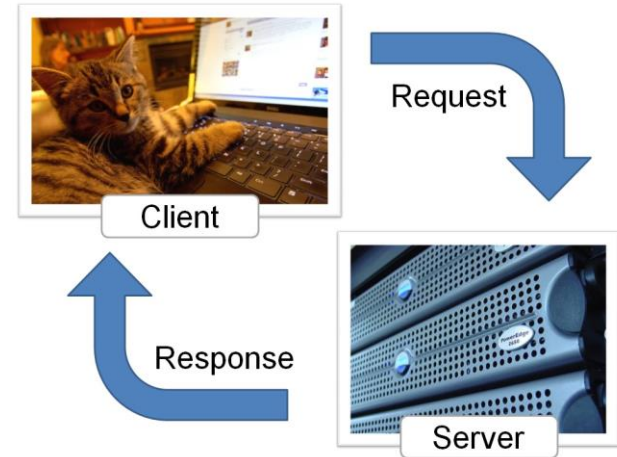
Object Oriented \ Compiled

Java  
C#  
C++

```
static void Main(string[] args)  
{  
    Console.WriteLine("Hello World!");  
}
```

# What is JavaScript?

- Interfaces with HTML and CSS
- Builds dynamic webpages
- Responds to user input
- Client-side (browser) language.



# Javascript can modify your webpage

*Multiple ways to print a message*

**Open a popup box**

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

**Write to your console**

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

**Write to the Web Page**

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



# Class Exercise: Your first JS Script

**Start a new project, Hello World. Click the JS tab**

Type the below in the JavaScript side. Refresh your code.

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

Now comment out the alert line . Refresh your code.

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

Now add the below line. . Refresh your code. Click the > in app.bsd to see the console.

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

Now comment out the console line.

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

Now print to the HTML window

```
document.write("Hello World!")
```

# What is a Variable?

- Place to store values
- Can change over time
- Can hold strings, numbers, booleans or arrays

```
let kittenAmount= 5;    // Store numbers  
let catName= 'Jane';    // Store words  
let doesSheHaveCats = true; //Store bools  
let foodList = ['lasagna','eclairs']; //arrays
```

# Variable Declarations

**Var (old school JS)**

`var greet= "hey";`

**Let (ES6, used now)**

`let greet = "hey";`

**Const (doesn't change)**

`const greet = "hey";`

Old school declaration

*Pros: Extremely*

*Permissible*

*Main Problem? Scope  
issues*

New Default

*Pro: Respects scope*

*Con: Updated but can't  
be re-declared*

*Pro: Respects Scope, best  
with objects*

*Cons: Can't be  
updated/re-declared, but  
its properties can*



# Variables and Text

Variable can also manipulate words including strings or groups of characters.

## Declare a Variable

```
let kittensName = 'Fluffy';
```

## To Combine Variables Use the Plus

```
let kittensName += ' Cottontail'; //+= Appends strings to existing var  
console.log("My cat's name is " + kittensName ); //concatenate two strings
```

## Insert a variable into a line of text

```
console.log( `The cat is  called ${kittensName}` ); //String interpolation, uses the backtick `
```

## Favorite String Functions

- Return String Length: `kittensName.length;`
- Position of a character in a string: `kittensName.search("kitten");`
- Make a string upper-case: `kittensName.toUpperCase();`
- Replace: `kittensName.replace("Fluff","puff");`



# Exercise: Play with Variables

In the JS tab.

Create 4 variables with the below names :

city, job, hobby, age, currentYear , isSingle (boolean - a true/false value)

Hint: `let job = 'web developer';`

Append two variables together in a sentence and write out

Hint: `console.log("Your name is " + name + " and your age is " + age);`

Now do some math calculate your .

`let birthYear = currentYear - age;`

Hint: `console.log(`You were born in ${birthYear}`);`

Note: To embed strings in quotes, you must use the backtick ` not the single quote '

# What are Functions?

- Reusable pieces of code
- Typically called by function name
- Can accept input values

**Javascript has its own functions**

```
alert("My popup box!");  
console.log('My secret warning message to developers');
```

**But you can create your own functions!**

```
function turtleFact() {  
  console.log('A turtle is called a plastron.');
```



# Exercise: Create 3 functions

Create the following functions in the JS tab, then click refresh.

```
function myCat() {  
  document.write ("My cat is mean");  
}  
myCat();
```

```
function myMath() {  
  document.write("<p>");  
  document.write (2+2);  
  document.write("<p>");  
  document.write (5-3);  
  document.write("<p>");  
  document.write (5*3);  
}  
myMath();
```

```
function myAge() {  
  let myAgeIsBeautiful = true;  
  document.write("<p>");  
  document.write("Is my age beautiful? Y/N: " +  
myAgeIsBeautiful);  
}  
myAge();
```



# Exercise: Functions with Parameters

Create the following functions with parameters

```
function myCat(catName) {  
  document.write ("My cat is called " + catName);  
}  
myCat("Juniper");
```

```
function myMath(num1, num2) {  
  document.write("<p>");  
  document.write (num1+num2);  
}  
myMath(5,6);  
myMath(6,9);
```

```
function myAge(myAgelsBeautiful) {  
  document.write("<p>");  
  document.write("Is my age beautiful? Y/N: " +  
myAgelsBeautiful);  
}  
myAge(true);
```

# Break!

10 min!

# Ways to Declare Functions

## Standard Function Declaration

```
function square(num)
{
  return num * num;
}
```

## Function Expression (ES5)

*Set function to variable*

```
let square = function (num )
{
  return num * num;
}
```

## Arrow Functions (ES6)

*Function shorthand*

```
let isArray = (val) =>
{
  return Array.isArray(val);
}
```



# Exercise: Declare as ES6

Rewrite the previous functions as ES6 functions

```
let myCat = (catName) => {  
  document.write ("My cat is called " + catName);  
}  
myCat("Juniper");
```

```
let myMath = (num1, num2) => {  
  document.write("<p>");  
  document.write ("Add " + num1+num2);  
  document.write("<p>");  
  document.write ("Multiply: " + num1*num2);  
  
}  
myMath(5,6);  
myMath(6,9);
```

```
let myAge = (myAgelsBeautiful) => {  
  document.write("<p>");  
  document.write("Is my age beautiful? Y/N: " +  
myAgelsBeautiful);  
}  
myAge(true);
```



# Comparison Values

## "Falsy" values

- The number 0
- "" (empty string)
- undefined
- null
- !(variablename)

## "Truthy" values

- The variable has a value

```
var hasData = "myValue"
```

```
if (hasData)
```

```
console.log("This has data");
```

==  
(loose)  
(compares values, not type)  
If (5 == "5")

===  
(strict)  
If (5===5)

a == b	Equal (LOOSE)
a === b	Identical (STRICT)
a != b	Not equal
a < b	Less than
a > b	Greater than
a <= b	Less than or equal to
a >= b	Greater than or equal to

# Conditionals: The If Statement

Use if to decide which lines of code to execute, based on a condition.

```
if (condition) { // statements to execute }
```

## If Statement

```
let catsAreGood = true;
```

```
if (catsAreGood === true )  
{  
  console.log("Cats are good");  
}
```

## If/else statement

Provides an alternate set of instructions

```
let age = 28;  
if (age >= 16)  
{  
  console.log('Yay, you can drive!');  
}  
else  
{  
  console.log('Sorry, but you have to wait!');  
}
```



# Exercise: checkTemperature

- Create a new function called checkTemperature that will receive a parameter, temperature

```
let checkTemperature = (temperature) => {  
}
```

- If it is below 50 degrees, print to console "Coat Day!"
- Else, print to console "Warm Day!"

```
if (temperature < 50) { console.log("Coat Day!"); }  
else { console.log("Warm Day!"); }
```

**What Happens when you run with the below values?**

checkTemperature(40);

checkTemperature(50);

checkTemperature(60);

# Multiple Conditionals

If you have multiple conditions, you can use else if.

## **If Else\Else\Else**

```
let age = 20;
```

```
if (age >= 35)
```

```
{
```

```
  console.log('You can vote AND hold office');
```

```
}
```

```
else if (age < 35)
```

```
{
```

```
  console.log('You can vote!');
```

```
}
```

```
else
```

```
{
```

```
  console.log('Sorry, no voting for you');
```

```
}
```



# Exercise: Update checkTemperature

- **Part 1: Update the first If statement to `>=`**
- `if (temperature <= 50)`
- Re-run the the code with `checkTemperature(50);`
  
- **Part 2: Add an Extra Else Statement, if `> 50`**
- `if (temperature <= 50) { console.log("Coat Day!"); }`  
`else if (temperature > 50) { console.log("Warm Day!"); }`  
`else console.log("The Weather is fine at " + temperature);`

# Expanded Comparisons

You can also chain comparison statements to handle ranges.

## If Elseif\Elseif\Else

```
let age = 20;
```

```
if (age >= 35)
```

```
{
```

```
    console.log('You can vote AND hold office');
```

```
}
```

```
else if (age < 35 && age >= 18)
```

```
{
```

```
    console.log('You can vote!');
```

```
}
```

```
else
```

```
{
```

```
    console.log('Sorry, no voting for you');
```

```
}
```

## && (and)

```
If (age >= 0 && age <= 5)
```

```
    console.log("baby");
```

## || or

```
If (age === 55 || age === 40)
```

```
    console.log("middle aged");
```

```
If (age != 100)
```

```
    console.log("You are young!");
```



# Exercise: Update checkTemperature

Update Check Temperature to test temperature ranges.

```
if (temperature >= 60 && temperature <= 70)
{
    console.log("Summer Day");
}
else if (temperature >= 50 && < 60)
    console.log("Spring Day"); //if there's one if line you don't need {}
else if ( temperature >= 100 || temperature <= 0)
{
    console.log("These are dangerous temperatures!");
}
else { console.log("Enjoy the temperature of " + temperature);
```

```
checkTemperature(40);
checkTemperature(50);
checkTemperature(101);
checkTemperature(0);
checkTemperature(-5);
checkTemperature(65);
```

# Switch\Case

Use switch case, if you don't have ranges, as it's easier to handle

## If Else\Else\Else

```
var age = 20;
if (age >= 35)
{
    console.log('You can vote AND hold
office');
}
else if (age >= 18)
{
    console.log('You can vote!');
}
else
{
    console.log('Sorry, no voting for you');
}
```

## Switch\Case

```
var age = 20;
switch(age)
{
    case 35:
        console.log('At 35, You can vote & hold office!');
        break;
    case 18:
        console.log('You can vote!');
        break;
    default:
        console.log('Sorry, no voting for you');
        break;
}
```





# Exercise: Coding Niceties

- Functions use variables (internal & external)
- Functions are typically used to return data
- Update the update checkTemperature to use internal variables
- **Part 1: Update check temperature to use an internal variable**
- Declare the variable at the start of the function  
`let message = "";`
- Instead of writing to console.log, set the responses to the variable message  
`message = "Coat Day";`
- Return the variable at the end of the function  
`return message;`
- **Part 2: Create an external variable called temp. Call checkTemperature passing it in.**  
`let temp = 90;`  
`console.log(checkTemperature(temp));`



# Exercise: Code Hint

```
let checkTemperature = (temperature) => {  
  let message="";  
  if (temperature >= 60 && temperature < =70)  
  {  
    message = "Summer Day";  
  }  
  else if (temperature >=50 & < 60)  
    message = "Spring Day";  
  else if ( temperature >= 100 || temperature <= 0)  
  {  
    message = "These are dangerous temperatures!";  
  }  
  else  
    message = "Enjoy the temperature of " +  
    temperature;  
  return message;  
}
```

```
let temp = 90;  
console.log(checkTemperature(temp));
```

# Introducing Arrays

Arrays are just lists of data.

```
let rainbowColors = ['Red', 'Orange', 'Yellow', 'Green'];
```

## Accessing Items

You can access items by using using bracket notation, starting at zero.

```
rainbowColors[0] = 'pink';
```



# Exercise: Play with your food!

1. Create a new project called arrays
2. Click on the JS tab
3. Create a food array & initialize to your favorite foods  
`let foods=['lasagne','popcorn'];`
4. Change the value of at least one position (also called index)  
`foods[1]='caramel popcorn';`
5. Print out the array to the console  
`console.log(foods);`

# Can You Do It? Entry Level Salaries

## entry level programmer Careers

### Average salaries for "entry level programmer" jobs in United States

Showing 13 salaries for "entry level programmer" in United States

Programmer	\$58,821 per year	Entry Level Programmer Analyst	\$44,906 per year	Programmer Analyst	\$81,979 per year
Entry Level Software Engineer	\$86,435 per year	Software Developer	\$108,676 per year	Entry Level Developer	\$71,182 per year
Web Developer	\$76,052 per year	Software Engineer	\$111,600 per year	Computer Programmer	\$48,055 per year

[https://www.indeed.com/career/salaries/entry%20level%20programmer?from=acme-keyword-salaries&rawkeyword=Entry+Level+Programmer&keyword=entry+level+programmer&reason=indexedserp\\_url](https://www.indeed.com/career/salaries/entry%20level%20programmer?from=acme-keyword-salaries&rawkeyword=Entry+Level+Programmer&keyword=entry+level+programmer&reason=indexedserp_url)

# Homework





# Exercise: The Calculator

1. Write a function called `squareNumber` that will take one argument (a number), square that number, and return the result.  
Hint: <https://github.com/philly-tech-sistas/intro-to-javascript/blob/gh-pages/solutions/calculator-squareNumber.js>
2. Write a function called `halfNumber` that will take one argument (a number), divide it by 2, and return the result.  
Hint: <https://github.com/philly-tech-sistas/intro-to-javascript/blob/gh-pages/solutions/calculator-halfNumber.js>
3. Write a function called `percentOf` that will take two numbers, figure out what percent the first number represents of the second number, and return the result. It should also log a string like "2 is 50% of 4."  
Hint: <https://github.com/philly-tech-sistas/intro-to-javascript/blob/gh-pages/solutions/calculator-percentOf.js>
4. Write a function called `areaOfCircle` that will take one argument (the radius), calculate the area based on that, and return the result. It should also log a string like "The area for a circle with radius 2 is 12.566370614359172." (Use `Math.floor` to round down to two decimal places)  
Hint: <https://github.com/philly-tech-sistas/intro-to-javascript/blob/gh-pages/solutions/calculator-areaOfCircle.js>



# Exercise: String Functions

- Create a function, `wordLength`, that receives a parameter and returns its length. Then, print out the results.

```
let wordParam = "Supercali";  
let wordLength = (word) => { return word.length ; }  
console.log("The word " + wordParam + " has " + wordLength + " letters ");
```

- Create a function, `findWord`, that searches for a word in a phrase. If the word does not exist let users know. Check [https://www.w3schools.com/jsref/jsref\\_search.asp](https://www.w3schools.com/jsref/jsref_search.asp) for more.

```
let searchParam = "Flakes";  
let phraseParam = "Corn Flakes are Great!";  
let findWord= (word, phrase) => {  
  let wordExists = false;  
  if (phrase.search(word) > 0)  
    wordExists = true;  
  return wordExists;  
}  
console.log("Does " + searchParam + " exist in phraseParam ? " + findWord(searchParam, phraseParam) );
```





# Exercise: Pluralize

- Create a function that accepts a word, it checks to see if the word is not empty (i.e. that it is truthy)
- If the word is not empty, then it appends the letter S to the end and prints the word to the console



# Bonus Exercises

## Which is bigger?

Write a function that compares two numbers and returns the larger one. Be sure to try it out with some different numbers. Bonus: add error messages if the numbers are equal or cannot be compared.

Hint: <https://philly-tech-sistas.github.io/intro-to-javascript/puzzles.html#bigger>

## Calculate Tip

Create a function that accepts a number and returns the 15% tip

## Replace a word in a phrase

Build a function that replaces a word in a phrase, using the replace function

```
kittensName.replace("Fluff","puff");
```

[https://www.w3schools.com/jsref/jsref\\_replace.asp](https://www.w3schools.com/jsref/jsref_replace.asp)

**Learn More**



# Learn More

Using Strings: [https://www.w3schools.com/js/js\\_string\\_methods.asp](https://www.w3schools.com/js/js_string_methods.asp)

Philly Tech Sisters Intro to JavaScript Resources: <https://github.com/philly-tech-sistas/intro-to-javascript>

ECMA Script: <https://tc39.es/ecma262/> (Outlines rules JavaScript should adhere to)

Mozilla JavaScript Guide: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide>

JavaScript Weekly: <https://javascriptweekly.com/> (Email round-up of JavaScript news)

# Free Resources to Continue Learning

Codecademy: <https://www.codecademy.com/>

- Offers a variety of courses for front end, back end, and more
- Has a free version, pro subscription, and paid 8-10 week specialized courses

Freecodecamp: <https://www.freecodecamp.org/>

- Offers curriculums for a variety of paths with certificates upon completion
- Completely free