

# Intro to JavaScript

**Harry J. Wang, Ph.D.**

University of Delaware

# JavaScript Review

- Part of the slides are reviews for this course:  
<https://www.codecademy.com/learn/learn-javascript>

# JavaScript

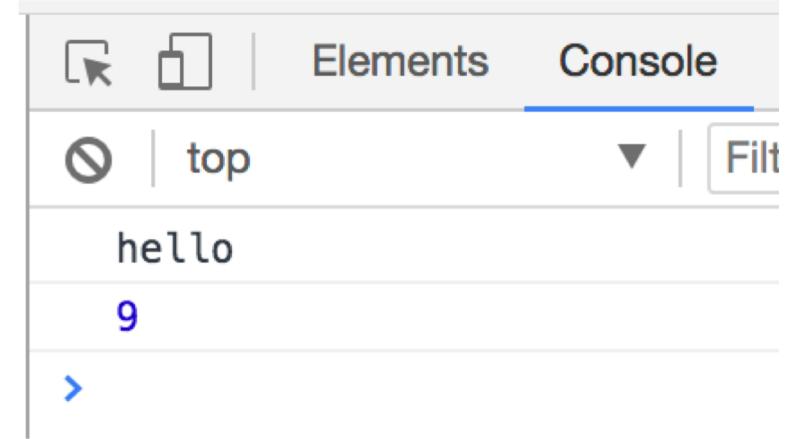
- JavaScript (JS), is a high-level, object-based, and interpreted programming language.
- Alongside HTML and CSS, JavaScript is one of the three core technologies of World Wide Web content production.
- JS is used to make webpages interactive
- All modern web browsers support JS by default with a built-in JavaScript engine.



# JS in HTML

- In HTML, JavaScript code must be inserted between <script> and </script> tags.
- Scripts can be placed in the <body>, or in the <head> section of an HTML page, or in both.
- The console.log() command is used to print, or log, text to the console.

```
<head>
  <meta charset="utf-8">
  <title>JS Exercise</title>
  <script>
    console.log("hello");
    console.log(4+5);
  </script>
</head>
```



# JS Comments

- A *single line comment* will comment out a single line and is denoted with two forward slashes // preceding a line of JavaScript code.

```
// The first 5 decimals of pi  
console.log('Pi is equal to ' + 3.14159);
```

- A *multi-line comment* will comment out multiple lines and is denoted with /\* to begin the comment, and \*/ to end the comment.

```
/*  
console.log('All of this code');  
console.log('Is commented out');  
console.log('And will not be executed');  
*/
```

- CSS comment: /\* to begin the comment, and \*/
- HTML comment: <!-- html comment -->

# Properties and Methods

- Objects in JS have properties and methods
- Properties are the **values** associated with a JavaScript object.
- Methods are the **actions** that can be performed on objects.

```
<script>
  //log the length of a string object
  console.log ("hello".length)
  // Log Hello in all uppercase letters
  console.log('hello'.toUpperCase());
  /* Use a string method to log the following statement without whitespace
  at the beginning and end of it. */
  console.log('    Remove whitespace    '.trim());
  console.log('hello'.startsWith('H')); // false
</script>
```

5
HELLO
Remove whitespace
false

See details about [standard built-in objects](#)

# Variables

- Variables are containers for storing data values.
- ***const***, short for constant, is a JavaScript keyword that creates a new variable with a value that ***cannot change***.

```
// constant variable  
const city = "New York";  
city = "Beijing" // this will cause an error
```

✖ ► Uncaught TypeError: Assignment to constant variable.  
at js.html:34

You cannot assign different values to Constant Variables .

- the ***let*** keyword is used to create the variables that can have reassigned values. (note ***var*** was used before the newest version of JavaScript (ES6) to define variables, their difference is about scopeing)
- Variables without values are ***undefined***

```
// use let keyword to declare variables  
let country = "USA";  
country = "China";  
console.log(country);
```

```
//undefined variables  
let zip_code;  
console.log(zip_code);
```

# Math Operators and String Concatenation

```
// math operators
let x = 4;
x += 2; // x equals 6

let y = 4;
y -= 2; // y equals 2
```

```
let z = 4;
z *= 2; // z equals 8

let r = 4;
r++; // r equals 5

let t = 4;
t--; // t equals 3
```

```
// String Interpolation/Concatenation
let course = "MISY350"
console.log("Come and Learn Web App Dev in " + course + " with Harry!")
// in ES6
console.log(`Come and Learn Web App Dev in ${course} with Harry!`)
```

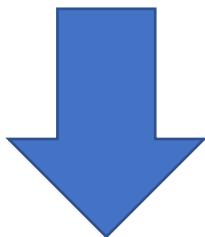
# Comparison Operators

- Comparison operators are used in logical statements to determine equality or difference between variables or values.
- Given  $x = 5$ :

Operator	Description	Comparing	Returns
==	equal to	$x == 8$	FALSE
		$x == 5$	TRUE
		$x == "5"$	TRUE
====	equal value and equal type	$x === 5$	TRUE
		$x === "5"$	FALSE
!=	not equal	$x != 8$	TRUE
!==	not equal value or not equal type	$x !== 5$	FALSE
		$x !== "5"$	TRUE
		$x !== 8$	TRUE
>	greater than	$x > 8$	FALSE
<	less than	$x < 8$	TRUE
>=	greater than or equal to	$x >= 8$	FALSE
<=	less than or equal to	$x <= 8$	TRUE

# if/else and else if Statements

```
// if/else statement
let forecast = "sunny";
if (forecast == "rainy") {
  console.log("Bring Umbrella!");
} else {
  console.log("Bring Sunglasses!");
}
```



```
// ternary operator
forecast == "rainy" ? console.log("Bring Umbrella!")
:console.log("Bring Sunglasses!");
```

```
// else if statement
let stopLight = 'green';
if (stopLight === 'red') {
  console.log('Stop');
} else if (stopLight === 'yellow') {
  console.log('Slow down');
} else if (stopLight === 'green') {
  console.log('Go!');
} else {
  console.log('Caution, unknown!');
}
```

# switch Statement

- The `break` keyword "jumps out" of the switch block.
- If you forget a break then script will run from the case where criteria is met, and will run the case after that regardless if criteria was met.

```
// switch statement with break
let groceryItem = 'papaya';
switch (groceryItem) {
  case 'tomato':
    console.log('Tomatoes are $0.49');
    break;
  case 'papaya':
    console.log('Papayas are $1.29');
    break;
  case 'lime':
    console.log('Limes are $1.49');
    break;
  default:
    console.log('Invalid item');
}
```

```
// switch statement without break
switch (groceryItem) {
  case 'tomato':
    console.log('Tomatoes are $0.49');
  case 'papaya':
    console.log('Papayas are $1.29');
  case 'lime':
    console.log('Limes are $1.49');
  default:
    console.log('Invalid item');
```

Wrong  
output  
without  
break



Papayas are \$1.29  
Limes are \$1.49  
Invalid item

# Variable Truthiness

- All variables that have been created and set are truthy (and will evaluate to true if they are the condition of a control flow statement) unless they contain one of the seven values listed below:

- `false`
- `0` and `-0`
- `""` and `''` (empty strings)
- `null`
- `undefined`
- `NaN` (Not a Number)

```
// truthy variables
let wordCount = 1;
if (wordCount) {
  console.log("Great! You've started your work!");
} else {
  console.log('Better get to work!');
}

// swapping the truthiness and falsiness of values using !
let favoritePhrase = '';
if (!favoritePhrase) {
  console.log("This string doesn't seem to be empty.");
} else {
  console.log('This string is definitely empty.');
}
```

- JavaScript provides an operator for swapping the truthiness and falsiness of values - the exclamation point (!).

# Logical Operators

- “all must be true”: &&
- “either can be true”: ||

```
//Logical Operators
if (stopLight === 'green' && pedestrians === false) {
  console.log('Go!');
} else {
  console.log('Stop');
}
```

# In-class Exercise

- Develop the first HTML file with Javascript code to log to console.
- Develop a grade calculator that given a number grade and log the letter grade to the console.
  - $\geq 90$ , A
  - $\geq 80$ , B
  - $\geq 70$ , C
  - $\geq 60$ , D
  - $< 60$ , F

---

Your grade is 85, which is B.

# Library

- Libraries contain methods that you can call without creating an instance of an object.
- Math library contains mathematical methods.

```
// random number between 0 and 1
console.log(Math.random());
// Returns the square root of 16
console.log(Math.sqrt(16));
//rounded downwards to the nearest integer
console.log(Math.floor(3.1415926));
```

See more about the Math library: [https://www.w3schools.com/jsref/jsref\\_obj\\_math.asp](https://www.w3schools.com/jsref/jsref_obj_math.asp)

# Questions?