Javascript Variables, Data Types, and innerHTML

Javascript Structure

Javascript is made up of statements that contain the following

- Values
- Operators
- Expressions
- Keywords
- Comments

A semicolon is used to separate statements in Javascript

```
var x = 115;
```

Variables

- Variables are a way of storing data in Javascript
- **Variables are case sensitive**
- Cannot contain spaces
- Must begin with a letter, underscore, or dollar sign
- Can only contain letters, numbers, or dollar signs

Creating a Variable

To create a variable use the var statement

```
var y = 89;
```

Variables can be redeclared later in your code

```
var y = 10;

var y = 20;
```

A few notes on naming variables

The first character can be a letter, underscore _, or dollar sign $.

After the first character, you can use numbers, as well as letters, underscores, or dollar signs.

Don’t use any of JavaScript’s reserved keywords. (You’ll learn this as we go)

Let

Let works the same as a variable but it cannot be reassigned

```
let y = 10;

let y = 20;  //y will still equal 10
```

Instead if you wanted to change the value of y you would need to do the following

```
let y = 10;

y = 20;
```

Const

Const cannot be reassigned or redeclared

Const is a “static” variable type

```
const  one = 1;

const two = 2;

const one = 3; //this will cause an error

two = 3; //this will cause an error
```

Lets try it

Use either alert() or console.log() and try re assigning var, let and const to see what happens

```
var x = 10;

var x = 20;

let y = 10;

let y = 20;

const z = 10;

const z = 20;
```

Data Types

Javascript contains 8 different data types. Here are some of the types we will focus on

- String
- Number
- Boolean
- Bigint
- Object*

String

A string is defined by using either single or double quotes

```
var firstText = “This is a string”;

var secondText = ‘This is also a string’;
```

But what if I want a ‘ or “ in my string?

To add a single or double quote to your string use a backslash \

```
var text = “Mark and Jordan\’s class is the best!”;
```

This backslash is called an “escape sequence”.

Lets try it

Create a variable, let, or const of type string. Include a “ or ‘ in one of your strings

```
var one = 'Hello';
var two = "Mark and Jordan\’s class is the best!";
var three = '\"Mark and Jordans class is the best!\" said everyone';
```

NOTE: Be careful when copy and pasting ‘ and “. You may need to manually type them in

Escape Sequence - New Line

Another common escape sequence is \n

let twoLines = “Mark and Jordan\’s class\n is the best!”

Mark and Jordan’s class

is the best!

Number

A number is defined as a number with or without a decimal

Note that a number cannot contain any letters

```
var y = 5;

var x = 8.2;
```

Boolean

A boolean is data type that can only contains the following pair values

- YES / NO
- ON / OFF
- TRUE / FALSE
- 0/1

Booleans can be used to determine if an expression is true or false

```
Boolean(11 > 5);
```

Booleans will become important when we begin learning about loops

Undefined

A null value is defined as nothing or having no value

You can define a null value by writing the following

```
var x;
```

This allows you to create a variable but not assign a data type to it

Empty Value

Like Undefined you can assign an empty value to a variable by using an empty string

```
var x = “”;
```

As opposed to getting an undefined message from the browser you will instead get nothing

Lets try it

Create an null value and empty string variable

```
var x;

var y = ‘’;

alert(x);

alert(y);
```

Why is a Data Type important

Data Types are important when we begin dealing with operations

For example it does not make sense to include a string type when adding 2 number types

```
var text = “Hello”;

var x = 4;

var y = 2;

var result = text + x + y;
```

Let’s try this!

TypeOf Operator

If you are ever unsure of what type your variable is you can use the typeof operator

```
var x = “Hello”;

alert(typeof x);
```

You can also use typeof on values

```
alert(typeof 5)

alert (typeof “Hello”);
```

InnerHTML - This is where things get fun!

InnerHTML allows you to change the value or add an element to your webpage

To change an element on a page it must have an ID

You can assign variables, text, numbers, bools, etc to an InnerHTML

InnerHTML is written as follows:

```
document.getElementById("ID of element").innerHTML = "what you want to change";
```

InnerHTML Demo

Lets try it

document.getElementById(“ID of element”).innerHTML = “what you want to change”;

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <title>Jordan</title>
  <script>
    function myFunction(){
      document.getElementById("changeMe").innerHTML = 'Goodbye';
    }
  </script>
</head>
<body>
  <p id="changeMe">Hello</p>
  <button onclick="myFunction()">Change Me</button>
</body>

</html>
```