

Logols Learning

WEEKEND WEB DEVELOPMENT BOOT CAMP

TRAINING: JAVASCRIPT/TYPESCRIPT

What is Javascript?

- ▶ Javascript != Java
- ▶ LiveScript vs. ECMAScript vs. JavaScript
- ▶ Client Side Scripting vs. Full Language
- ▶ Javascript Engine or Virtual Machine
- ▶ “Safe” Programming Language – Ignores Memory/CPU
- ▶ Objects, but not really object oriented

- Javascript is a programming language.
- It's not Java and it doesn't have anything to do with Java.
- Its original name was Mocha then LiveScript and it was renamed to Javascript because Java was so popular at the time.
- ECMA made it into a standard and each standard is referred to as ECMAScript.
- Javascript was made for client side scripting, but is now being used as a full language. It is being used as both client and server side coding.
- The Javascript engine or virtual machine turns the code into something that works.
- Browsers implement this for client side scripting.
- Javascript is a safe programming language as there is nothing for it to interact with the CPU or memory.
- Javascript has objects, but it is not really an object oriented programming language.
- There is no inheritance or interfaces.

Javascript Linking

- ▶ Script tag in html can be used to write Javascript
- ▶ Javascript files can be linked in the head element of html

Example:

```
<script>  
    alert('test Javascript');  
</script>  
  
or  
  
<script src="/script.js"></script>
```

- Javascript can be written directly in the html page.
- To do that you can use the script element and put your javascript inside of the start and end tags.
- You can also create a separate javascript (.js) file.
- To link to that code use the script tag within the head tag.
- Specify the src attribute as the location of the javascript file.

Comments

- ▶ // - is used for comments
- ▶ /* *. – is used for multi-line comments

Example:

```
// this is a comment
```

```
/*
This is a multi-line comment
*/
```

- To create a single line comment use //
- To create multi-line comments use /* */

Declaring Variables

- ▶ let [name];
- ▶ let [name] = [value];
- ▶ let [name1] = [value1], [name2] = [value2], [name3] = [value3];
- ▶ You could use var, but it handles scope differently
- ▶ Use let or const

Example:

```
let message = "hello";  
alert(message);
```

- Variables can be declared with let or var.
- There is a difference with how scope works. I'm not really going to go into it.
- Just use let for now.
- These are some different ways to declare a variable.
- You can just specify the name without a value.
- You can specify the name and the value and you can specify multiple variables.
- const should be used if a variable will be assigned immediately and not changed.

Data Types

- ▶ string – Use single or double quotes
- ▶ number
- ▶ boolean
- ▶ function
- ▶ object
 - ▶ Object
 - ▶ Date
 - ▶ Array
- ▶ null – set and doesn't have a value
- ▶ undefined – not yet set
- ▶ Use typeof to find type

- Here are the data types available in Javascript.
- There is string, number, boolean, function, object, null, and undefined.
- You can define many custom objects.
- Use typeof to find the type of a variable.

Variable Scope

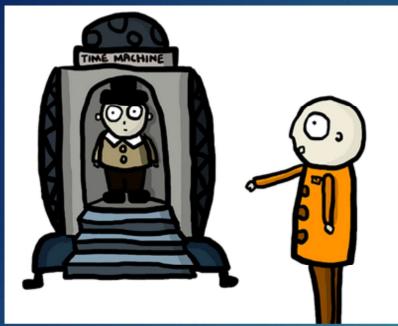
- ▶ Local or function scope
 - ▶ defined inside a function
- ▶ Global scope
 - ▶ The scope is global if the variable is declared outside of a function
- ▶ Lexical scope
 - ▶ Function inside a function has access to variables declared in outer function (closure)

- Variables can have local or function scope, global scope, or lexical scope.
- Be careful of global scoped variables.
- You may get unintended consequences if you use the same global variables as other Javascript libraries.
- To avoid this, try to use lexical scope where you create a function and have other functions and variables inside of it.
- Example: <https://toddmotto.com/everything-you-wanted-to-know-about-javascript-scope/>

What is this?

- ▶ this can be used instead of variable name
- ▶ Different value bound to this depending on how function is called
- ▶ Refers to outer most global object (window) by default
- ▶ Within a method, defined by the object that calls it.
 - ▶ Closures used, this becomes the outer function.

- this is a keyword in Javascript that can be sometimes confusing.
- It can be used instead of the variable name.
- A different value is bound to this depending on how a function is called.
- The default is the outer most global object (window) by default.
- Example: <https://toddmotto.com/everything-you-wanted-to-know-about-javascript-scope/>



EXAMPLE

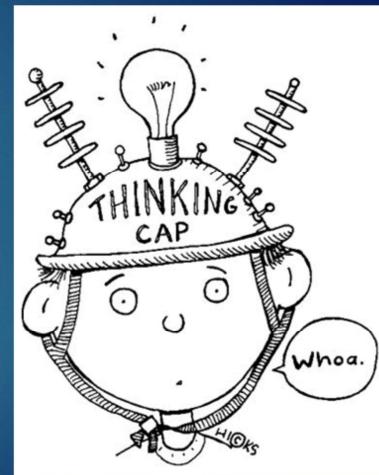
VARIABLES

Let's go through some more examples.

- Create variables
- Show conversions
- Show variable scope
- Show this examples

ASSESSMENT

VARIABLES



- Write on the board, the html to link to a Javascript file name “script.js”
- Write on the board the symbols for a single line comment.
- Write on the board the declaration of a string variable with the value “hello world”

Comparison Operators

- ▶ == equal to
- ▶ === equal value and equal type
- ▶ != not equal
- ▶ !== not equal value or not equal type
- ▶ > greater than
- ▶ < less than
- ▶ >= greater than or equal to
- ▶ <= less than or equal to

- Here are the comparison operators available.
- They are similar to what is available in C#.
- === compares both the value and the type.

Logical Operators

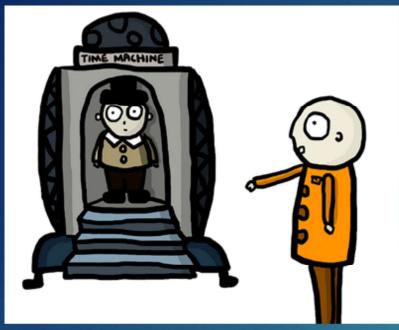
- ▶ && logical and
- ▶ | | logical or
- ▶ ! Logical not

- Here are the Logical Operators.
- Again similar to C#.

Conditional Statements

- ▶ If
- ▶ else
- ▶ else if
- ▶ switch

- For conditional statements you again have if, else, else if, and switch.
- Let's look at some examples.
- Example: https://www.w3schools.com/js/js_if_else.asp
- Example: https://www.w3schools.com/js/js_switch.asp



EXAMPLE

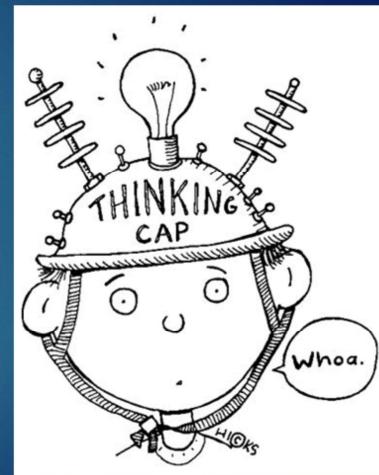
CONDITIONS

Let's go through some more examples.

- Create various conditions

ASSESSMENT

CONDITIONS



- What are the logical operators?
- Write an if statement on the board that checks if a bool variable is true.
- Write an if else statement on the board that checks if a bool variable is true.
- Write a nested if statement on the board that checks two different bool variables.
- Write if, else, else if statements to write to the console the text representation of the numbers 1 to 3.
- Write a switch statement to write to the console the text representation of the numbers 1 to 3.

Assignment

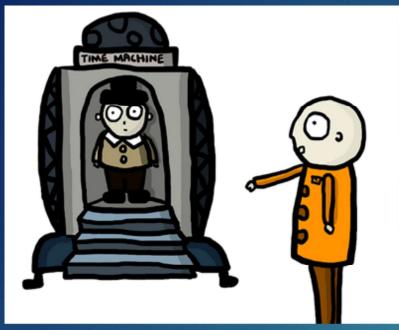
- ▶ A status report is needed of all government employees. Statuses are:
 - ▶ 1: Alive, 2: Zombie, 3: Dead, 4: Unknown
- ▶ Given a number variable, write if else statements and console out the persons status.
- ▶ Using the same number variable, modify your code to perform the same operation with a switch statement.



Loops

- ▶ for
- ▶ for / of
- ▶ while
- ▶ do / while

- Here are the loops available. Similar to c#.
- The only difference is that for / in replaces foreach.
- Example: https://www.w3schools.com/js/js_loop_for.asp
- Example: https://www.w3schools.com/js/js_loop_while.asp



EXAMPLE

LOOPS

Let's go through some more examples.

- Create various loops

ASSESSMENT

LOOPS



- Write on the board a declaration of an array and initialize it with the numbers 1 to 3.
- Write a while loop that loops through an integer array 1 to 3 and writes each number to the console.
- Write a do while loop that loops through an integer array 1 to 3 and writes each number to the console.
- Write a for loop that loops through an integer array 1 to 3 and writes each number to the console.
- Write a for in loop that loops through an integer array 1 to 3 and writes each number to the console.

Assignment

- ▶ A status report is needed of all government employees. Statuses are:
 - ▶ 1: Alive, 2: Zombie, 3: Dead, 4: Unknown
- ▶ Given an array of number variable, write loops with if else statements and console out everyone's status.
- ▶ Use all loop types.
- ▶ Given another array of string variables with names, write out the name and their status.



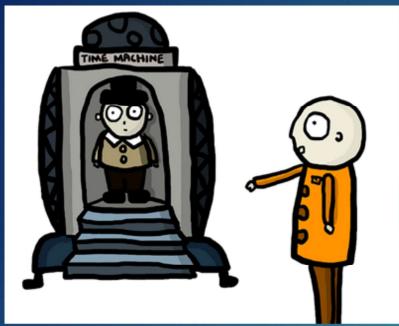
function Syntax

- ▶ Function performs an action
- ▶ Can also be a type in Javascript

Example:

```
function square(num) {  
    return num * num;  
}  
alert(square(4));
```

- Functions perform an action.
- In Javascript as you saw a function can also be a type.
- This is similar to a delegate in C#.
- The example shows a simple square function and how it's called.



EXAMPLE

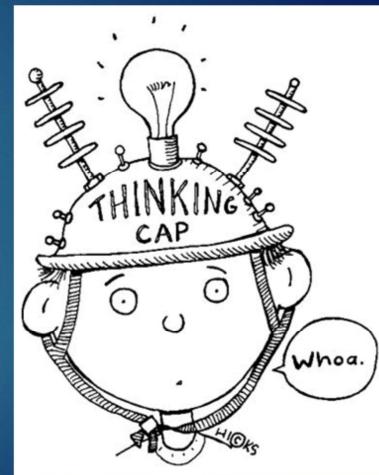
FUNCTIONS

Let's go through some more examples.

- Create basic functions
- Create functions in functions
- Create closures

ASSESSMENT

FUNCTIONS



- Write a method definition that does not return anything and takes two integers as parameters.
- Write a method definition that returns a string and takes two integers as parameters.

Assignment

- ▶ A status report is needed of all government employees. Statuses are:
 - ▶ 1: Alive, 2: Zombie, 3: Dead, 4: Unknown
- ▶ Modify your previous program to create a method that handles the condition given a parameter for status and for name that returns the concatenated string.
- ▶ Write a void method that takes a string parameter and writes it to the console.



What is TypeScript?

- ▶ Superset of the Javascript language
- ▶ Transpiler – interprets Typescript to Javascript
- ▶ Strongly Typed
- ▶ Also Support for:
 - ▶ Generics
 - ▶ Classes
 - ▶ Interfaces
 - ▶ Namespaces
 - ▶ Etc...



- Typescript is a preprocessor it is a superset of the Javascript language.
- This means it is a language that uses Javascript, but adds to it.
- Before you run the website the TypeScript files are transpiled into regular Javascript.
- Typescript is strongly typed and offers support for generics, classes, interfaces, and namespaces.
- This gives the developer the feel of developing in an object oriented language.

TypeScript Types

- ▶ * - any
- ▶ Built-in Types
 - ▶ Number
 - ▶ String
 - ▶ Boolean
 - ▶ Void
 - ▶ Null
 - ▶ Undefined
- ▶ User-defined Types
 - ▶ enum
 - ▶ class
 - ▶ Interface
 - ▶ array
 - ▶ tuple

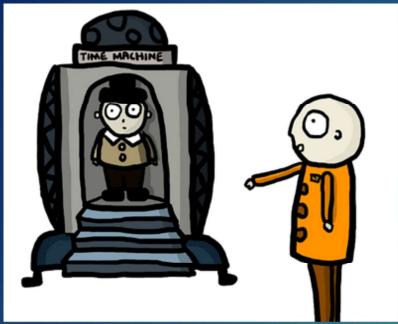
- These are the types that can be specified in TypeScript.
- They mostly map to the Javascript types.
- You have Number, String, Boolean, Void, Null, and Undefined.
- Then there are user-defined types.
- Like in C#, anyone can create user-defined types.

TypeScript Syntax

- ▶ Typescript is written in .ts files that are transpiled to .js files
- ▶ Also option to create .d.ts declaration files for intellisense
- ▶ Variable Declaration: let[name] :[type] = [value];

```
Function: function [name] ([param1]:[type], [param2]:[type]) : [return type] {}  
  
class [name] {  
    name:string;  
  
    constructor(name:string) {  
        this.name = name;  
    }  
  
    write():void {  
        console.log("Name is " + this.name);  
    }  
}
```

- Here is the TypeScript syntax.
- The files are .ts files. These are the ones that get transpiled into .js files.
- A variable declaration is done by using let followed by the variable name : variable type = the value.
- Functions can also use the type as parameters and as a return type.
- Let's look at the example.



EXAMPLE

TYPESCRIPT

Let's go through some examples.

- getElementsByTagName example
- GetByClassName example
- Document.QuerySelectorAll example
- Inner html update
- Attribute update
- Style update
- Create element
- Remove child
- Append child
- Replace child
- event

ASSESSMENT

TYPESCRIPT



- Write on the board a class named Car with a variable named miles and a constructor that set miles to 100.
- Write on the board a function that returns a number and is passed a number called Miles. The function should be named Drive.

Assignment

- ▶ A status report is needed of all government employees. Statuses are:
 - ▶ 1: Alive, 2: Zombie, 3: Dead, 4: Unknown
- ▶ Write the same program in TypeScript Syntax that you just created in JavaScript. Make use of typing.



QUICK REVIEW

JAVASCRIPT / TYPESCRIPT



- Write on the board, the html to link to a Javascript file name “script.js”
- Write on the board the symbols for a single line comment.
- Write on the board the declaration of a string variable with the value “hello world”
- Write an if statement on the board that checks if a bool variable is true.
- Write an if else statement on the board that checks if a bool variable is true.
- Write a nested if statement on the board that checks two different bool variables.
- Write if, else, else if statements to write to the console the text representation of the numbers 1 to 3.
- Write a switch statement to write to the console the text representation of the numbers 1 to 3.
- Write on the board a declaration of an array and initialize it with the numbers 1 to 3.
- Write a while loop that loops through an integer array 1 to 3 and writes each number to the console.
- Write a do while loop that loops through an integer array 1 to 3 and writes each number to the console.

- Write a for loop that loops through an integer array 1 to 3 and writes each number to the console.
- Write a for in loop that loops through an integer array 1 to 3 and writes each number to the console.
- Write a method definition that does not return anything and takes two integers as parameters.
- Write a method definition that returns a string and takes two integers as parameters.
- Write on the board a class named Car with a variable named miles and a constructor that set miles to 100.
- Write on the board a function that returns a number and is passed a number called Miles. The function should be named Drive.

Additional Resources

- ▶ JSFiddle
 - ▶ <https://jsfiddle.net/>
- ▶ free Code Camp
 - ▶ <https://www.freecodecamp.org/>
- ▶ Microsoft Virtual Academy
 - ▶ <https://mva.microsoft.com/en-us/training-courses/javascript-fundamentals-for-absolute-beginners-14194>
- ▶ MDN Web Docs
 - ▶ https://developer.mozilla.org/en-US/docs/Web/JavaScript/Language_Resources
- ▶ TypeScript Documentation
 - ▶ <https://www.typescriptlang.org/docs/home.html>