THE KNOWLEDGE HOUSE

Intro to JavaScript

09/14/2021

Class Overview

- Fellows will understand...
  - The skills used to become a proficient programmer
- Fellows will be able to....
  - Know the basic workings of JavaScript
  - Integrate JavaScript to a web page

What is Javascript

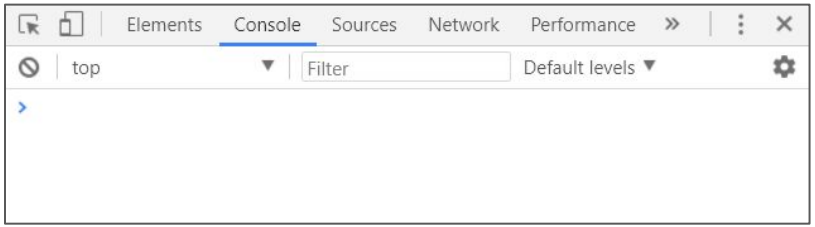
- JavaScript is a programming language most well-known as the scripting language for Web pages
- If HTML is the skeleton and CSS is the skin, JavaScript would be the brains
- JavaScript is also used for non-browser environments, such as Node.js, an open source server framework

Javascript is Not JAVA

- JavaScript
  - Created by Netscape
  - Needs Web browser to run
  - Usually used for light-weight projects
- JAVA
  - Created by Sun Microsystems
  - Builds standalone programs
  - Used to build enterprise projects

Using Javascript in Browser

- You can use JavaScript directly in the browser with the console
- This is mainly used for testing code before implementing it
- Access console on Mac
  - Cmd + Shift + C
- Access console on Windows
  - Ctrl + Shift + J
- Then click on the Console tab



Using Javascript in Web Pages

- In order to use JavaScript you must use the script tag
- Although the script tag can go in the head, it is usually placed at the bottom of the body to allow the web page to load faster
- Much like CSS, there is an internal and external method.

Using Javascript in Web Pages

- Internal

```
<body>
    <h1>Look at Console</h1>
    <script>
        console.log('Hello World');
    </script>
</body>
```

- External

```
<body>
    <h1>Look at Console</h1>
    <script src="script.js"></script>
</body>
```

File: page1.html

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

File: script.js

Javascript Operators

- These arithmetic operators, “(+, -, *, /, %)”, are used to compute values
  - + addition
  - - subtraction
  - * multiplication
  - / division
  - % modulo: gives the remainder of dividing two number
    - 13 % 5 would be 3 because 13 / 5 is 2 remainder 3
- The assignment operator, “=”, is used to assign values to variables
  - var x = 4;

Special Number Operators

- In Javascript there are special Math Object methods that are practical, for example if you need to do a square root.

```
// square root of 4
Math.sqrt(4)
// output => 2
```

Special Number Operators

- Need to take a number to a “power”, Math.pow

```
// 3^2
Math.pow(3,2)
=> 9
// 2^5
Math.pow(2,5)
=> 32
```

Special Number Operators

```
// round down to the nearest integer
Math.floor(4.16)
=> 4
Math.floor(4.99)
=> 4

//round up to the nearest integer
Math.ceil(4.16)
=> 5
Math.ceil(4.99)
=> 5
```

Special Number Operators

- ```
  // round to the nearest integer
  Math.round(4.16)
  => 4
  Math.round(4.99)
  => 5
  ```
  ```

Variables

- Variables are containers for storing data values.
- To create a variable use the keyword var, let, or const
  - var and let can be updated at any time
  - const cannot be updated once created
- Variables can be given values when being created or given values on a separate line.

```
let x;
x = 17;
let y = 38;
```

Commenting

- Comments are ignored, and will not be executed.
- This is used to document code and keep certain lines from being executed.

```
//Example of Single Line Comment
//var x = 6;
//var y = 9;
/*
    Example of Multi Line Comment
    var x = 6;
    var y = 9;
*/
```

Var and let/const

- So far we have been using var as examples, but var is a pre-ES6 globally scoped-variable.
- But when ES6(2015) came out, a new set of keywords were created that provide variables: let and const.
- Let and const were created to address the global scope issues created by var.
- Today, it is considered standard to use let and const instead of var.

Let and const continued..

- Starting with the let keyword.
- Let creates a variable that can be re-assigned later on.

```
Let favFood = “Burger”

favFood = “pizza”

//if you try this in the console you will see the
output is “pizza”
```

Let and const continued..

- Const creates variables that cannot be re-assigned.

```
Const car = “Volvo”
car = “Toyota

//output: Uncaught TypeError: Assignment to constant
variable
```

Primitive Data Types

- A primitive data value is a single simple data value with no additional properties and methods.
- Number
  - written with, or without decimals (123 or 12.3)
  - Numbers with decimals are called Floats; Numbers without are called Ints
- String
  - series of characters
  - can be written with double(“text”) or single quotes(‘text’) or backticks (`text`)
- Boolean
  - can only have two values: true or false
  - used in conditional testing (More in the later slides)

Primitive Data Types

- Null
  - something that doesn't exist
  - “Nothing”
  - Its type is actually an object
  - The intentional absence of a value
- Undefined
  - Is like null but instead of being an object, its type is undefined
- To see the type of a value use “typeof value”
  - Example: typeof null
  - Returns object

Boolean Data Type

- As mentioned in a previous slide, Boolean is a logical type and has two value, true and false.

A	B	A and B	A or B	Not A
False	False	False	False	True
False	True	False	True	True
True	False	False	True	False
True	True	True	True	False

Boolean (Truthy & Falsey)

Everything below becomes false when converted to a Boolean

- False
- 0
- ‘ ‘ (empty string)
- NaN
- Null
- undefined

Boolean (Truthy & Falsey)

A way to check the truthiness or falsiness of a value is by adding a ! in front of a value. The value returned will be the inverse of the value in boolean.

But if you add two ! , such as !! you get the original one.

!!3 => true

!!0 => false

!!-4 => true

!! null => false

Boolean (Truthy & Falsey)

Logical operators return boolean values of true or false

The AND binary operator is &&

- True && True => True
- True && False => false

The OR binary operator is ||

- True || False => True
- False || True => True

Boolean (Truthy & Falsey)

The third “unary” operator requires only one value

- NOT , denoted as !
- !true
- //=> false

Mixing Data Types

- To join the two types use the addition symbol (+).
  - This is called concatenating
- The outcome of adding any data type to a string will always be a string

```
console.log(10 + “eggs”);
console.log(10 + 2 + “eggs”);
console.log(“eggs” + 10 + 2);
```

Prints 10eggs

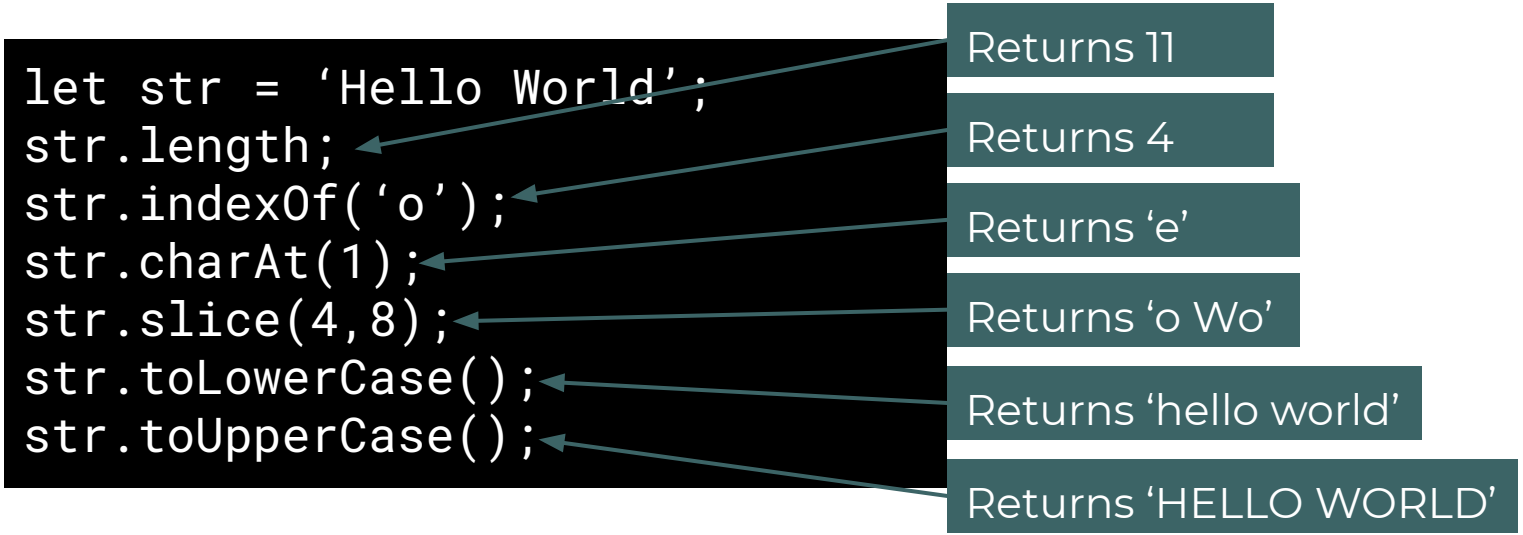
Prints 12eggs

Prints eggs102

Methods

- All data types have methods that can be used to get more information about them or allow you to edit them as needed.
- Methods are accessed with “.” or a dot
- .length - Returns amount of character in string, spaces are counted.
- .indexOf(str) - Returns position of first occurrence in string, returns -1 if not found. When dealing with indices, start counting from 0
- .charAt(index) - Returns character at specified position, returns -1 if not found.
- .slice(startIdx, endIdx) - Returns part of string starting from the first argument to but not including the second argument
- .toLowerCase()/.toUpperCase() - Lower cases/Capitalizes all letters in string.

Methods


let str = 'Hello World';
str.length;
str.indexOf('o');
str.charAt(1);
str.slice(4,8);
str.toLowerCase();
str.toUpperCase();
Returns 11
Returns 4
Returns 'e'
Returns 'o Wo'
Returns 'hello world'
Returns 'HELLO WORLD'

Helpful functions

- .parseInt()
  - Converts string into integer.
- .parseFloat()
  - Converts string into floating point number.

```
let str = '123.45';
parseInt(str);
parseFloat(str);
```

Returns 123

Returns 123.45

Helpful functions

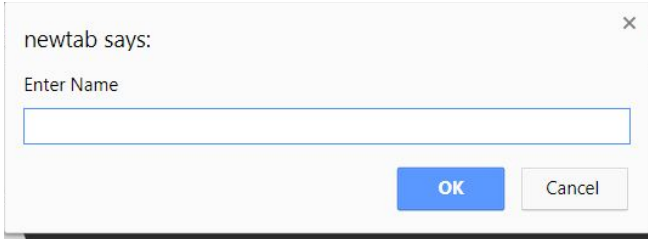
- Alternatively the unary plus operator can turn a variable into an integer

```
let str = ‘123’;
str = +str // output: 123
```

Get User Input for Console

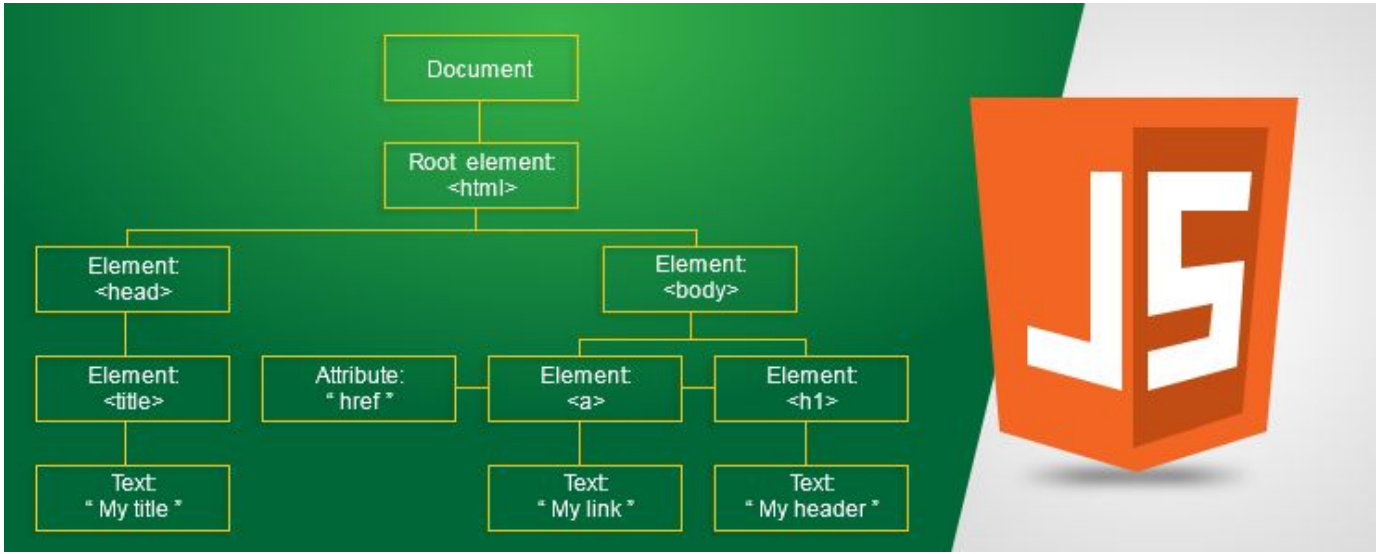
- .prompt() displays a dialog box that prompts the user for input.
- The method returns the input value if the user clicks "OK". If the user clicks "cancel" the method returns null.
- There is the option to add text to the dialog box to tell the user what to enter

```
let user = prompt('Enter Name');
```



Using JavaScript with HTML (DOM)

- With the HTML DOM (Document Object Model), JavaScript can access and change all the elements of an HTML document.



Accessing the DOM

- Access the DOM with the keyword “document”.
- Then use dot notation to get properties to edit and methods to perform.

```
document.body.bgColor = ‘blue’;
```

Changes the web page background color to blue

Editing Specific Elements

- To access a specific element give the element an id

```
<h1 id='myText'></h1>
<script>
    let myText = document.getElementById('myText');
    myText.innerHTML = 'Hello World';
    myText.style.color = '#FF00FF';
</script>
```

Gets element with the id 'myText'

Changes element's text to 'Hello world'

Changes element's color to '#FF00FF'

Task: Using Data Types with DOM

- [known brand] wants to welcome their visitors as soon as they reach their page
- Create HTML web page with text element saying “Hello my name is ”
- Use prompt(“Enter name”) to get input within JavaScript
- Use DOM to add input from prompt to the HTML text element

THE KNOWLEDGE HOUSE

Checklist

Skills Learned

Competency: Programming Fundamentals

- ❏ JavaScript

Homework

- ❏ Using Data Types with DOM