

JAVASCRIPT

Mastering Javascript



github: 2Kronos

# Console

* Display important messages like erros[1]
* If we want to see the a specific line of code that is running in the background we can print or log to our console.
* We use the .log() method to print whats inside the parenthesis on to the console.

# Data types

The first 7 are the primitive data types.

1. Number: Any number, including numbers with decimals: 4, 8, 1516, 23.42.
2. BigInt: Any number, greater than 253-1 or less than -(253-1), with n appended to the number: 1234567890123456n.
3. String: Any grouping of characters on your keyboard (letters, numbers, spaces, symbols, etc.) surrounded by single quotes: ' ... ' or double quotes " ... ", though we prefer single quotes. Some people like to think of string as a fancy word for text.
4. Boolean: This data type only has two possible values— either true or false (without quotes). It’s helpful to think of booleans as on and off switches or as the answers to a “yes” or “no” question.
5. Null: This data type represents the intentional absence of a value, and is represented by the keyword null (without quotes).
6. Undefined: This data type is denoted by the keyword undefined (without quotes). It also represents the absence of a value though it has a different use than null. undefined means that a given value does not exist.
7. Symbol: A newer feature to the language, symbols are unique identifiers, useful in more complex coding. No need to worry about these for now.
8. Object: Collections of related data.

# MATHS

Can be done in the console

* There is an order that must be adhered to when maths in JavaScript.
* The order can be controlled by using brackets .
* Best practiced when calculating money is to calculate in cents instead of dollars because calculation with floats are sometime inaccurate.
* Math.round() [use to round off numbers]
* Order precednace left to right paranthesis, exponengts, multiply and division and modulus, addition and subtraction.

## Maths in JS

* Math is a built in objects that provides a collection of properties and the methods
* Round down a number Math.floor(x)
* Round up a number Math.ceil(x)
* Eliminate any decimal portion Math.trunc(x)
* Raise to the power Math.pow(x, y)
* Square root Math.sqrt(x)
* Natural log of a number Math.log(x)
* Math trigonometry Math.sin(x), Math.cos(x), Math.tan(x),
* Absolute value of a function Math.abs(x)
* Sign of a number +/- Math.sign(x)
* Max of numbers Math.max(x, y, z).
* Min of numbers Math.min(x, y, z)

### Math Random

* Math.random() use to generate a random number

 randomNum1 = Math.floor(Math.random() \* max) + min;

* With this code we can control what range the random number we will get based on the values of max and min.

# STRINGS

Can be done in the console as well

1. ‘…….’
2. “…….”
3. `……..` (template strings)

* Use single quotes ‘’ by default  
  Double quotes can when you use single notes in your string for word like it’s

" I'm learning Javascript"

* Can also use an escape character \’

This combo act as one character the comma is the text   
' I\'m learning javascript' (same out come)

It does not start or end the sting

* \” creates a double quote as text
* \n new line character
* Use alert(); to make a pop up on your window
* Can add string together using + (concatenation)
* string + number = string
* Strings follow order of operation left to write  
    
  '$' + 20.95 + 7.99

'$20.957.99' (here it just put the 7.99 with the rest of the word because it is a string)

'$' + (20.95 +7.99) (order of operation)

'$28.939999999999998'

'$' + (2095 +799)/ 100 (float number inaccuracy)

'$28.94'

## Template strings

`……`

* Can do interpolation = insert value directly into a string

`Item (${ 1 + 1}): $${(2095 + 799)/100}` dollar

'Item (2): $28.94'

${} = insert value directly into string

* Multi line strings

## Auto type conversion

* Can check whtat type a value is using typeof (automatic type conversion) can be used to add signs to number in my example I was adding dolar signs to numbers

# HTML

* Nesting element inside of a element

<p> paragraph of the text <button>Hi</button></p>

* Multiple spaces are combined into 1 space on webpage
* New lines count as spaces

## HTML Attributes

* Attributes change the behavior of an element

<button title=" tooltip"> hello</button>

* In this line of code if you hover over the button element you will a small pop that displays the words in the double quotes
* Always out on the opening tag
* The title is the attribute name tells what we changing of the element.

<button title =  " good job" class=" red-button"> hello</button>

* class attribute = adds a lable to an element
* html tag contains the entire webpage.
* Head tag contains information about the page and everything that is not visible on the webpage.

onclick=""

* This code will run Javascript code when ever it is clicked

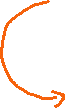
# CSS

* Uses a style tag to write CSS inside

  button{

        background-color: blueviolet;

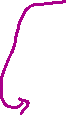
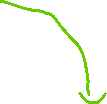
        color: aliceblue;



        border: none;



    }



* This a HTML element which we want to change.
* Left side is called the CSS property
* This is called a css value it tells the computer what we are changing the property too



# Variables

* The keyword let creates a new variable.
* Can’t use special words for naming like let
* Can’t start variable name with a number
* Can’t use special characters. But you can use $ and \_

## Re-assigning variables

* Don’t use the let

# Booleans

* Use in comparison operators
* Two ways to check if values are equal

console.log(5 == '5.00')

* This double equals tries to convert both values into the same type and then compare them so the answer wil be true. They should not be equal because one is a string and the other is a number
* In JS we use === triple equal
* Not eqaul will be !==
* To avoid conversion
* Logic operators help us to combine Boolean values they have lowers pro

# User Inputs

username = window.prompt("whats your username");

* Window prompt open a pop up on the widow tap

 document.getElementById("myH1").textContent

* Changes the content of an H1 element

<label>Username</label>

  <input id ="myText"> <br>

* These lines of code create an input box
* myText.value this gets the value of the input

# Type conversion

* type conversion = change the datatype of a value to another (strings, numbers, booleans)
* Used in window prompt because the input from the window prompt is a string to convert it to a number w
* Boolean will always be true if
* If an a input is Not A Number it will NaN

# Rock Paper Scissors game

* Math.random())
* Generates a random number between 1 and 0

# Check property

* .checked = property that determines the checked state of an HTML checkbox or radio button element

 <input type="checkbox" id="myCheckBox">

  <label for="myCheckBox">subscribe</label><br><br>

* This creates a check box the for attribute and the id of the input must be the same so when we click on the label it will select the check box

 <input type="radio" id="visaBtn" name="card">

  <label for="visaBtn">Visa</label><br>

* Creates a radio button and giving it the name attribute is how we are allowed to check one box at a time

myCheckBox.checked

* If this is checked it will evaluate to true or fals

# Ternary Operator

* Shortcut to if{} and else{} statements help to assign a variable based on a condition  
  condition ? codeIfTrue : codeIfFalse;
* Alternative to an if-else statement

# Switches

* Alternative to else if statment

let day = 1;

switch (day) {

  case 1:

    console.log("It is Monday");

    break;}

In the code above this will check is eqaual to the day set and it will compile the code that is equal to the case  
  
switch(true){

  case testScore >= 90:

      letterGrade = "A";

      break;}

console.log(letterGrade);

In this instance the code will evaluate to either true or false and it will display the ouput which is stored in the letterGrade variable.

# String Methods

* Allow us to manipulate and work with strings.

let userName = "Bro Code";

console.log(userName.charAt(0));

* This allows to get the first character of a string or whatever letter in the string specified buy the number

 console.log(userName.indexOf("o"));

* This shows us the first occurrence of a specific character in a string

console.log(userName.lastIndexOf("o"));

* This shows us the last occurrence of a specific character in a string

 console.log(userName.length);

* This gets the length of the spring

 let userName = "BroCode";

 userName = userName.trim();

 console.log(userName);

* Trims white spaces before or after the string

 userName = userName.toUpperCase();

* Turns the stirng into upper case

userName = userName.toLowerCase();

* Turns the string to lower case

 userName = userName.repeat(3);

* Use to repeat a string by a specified amount

 let result = userName.startsWith(" ");

* To see if a string starts with a certain character it will evaluate to true or false

let result = userName.endsWith(" ");

* Check to see if thee strings ends with a certain character

 let result = userName.includes(" ");

* Does the string include any of the specified characters

let phoneNumber = "123-456-789";

 phoneNumber = phoneNumber.replaceAll("-", "");

* Replace string with another string

phoneNumber = phoneNumber.padStart(15, "0");

* Pad the string so that it will be a specific amount of length and pad it with the specified charcters the padding will start from the front.

phoneNumber = phoneNumber.padEnd(15, "0");

* Pad the string so that it will be a specific amount of length and pad it with the specified charcters the padding will start from the end.

# String slicing

* Creating a substring from a portion of another string
* String.slice(start, end)
* And if you going from a certain index till the end you do not have to spe

const fullName = "Bro Code";

let firstName = fullName.slice(0, 2);

* Ending index is exclusive so (index 0, 1, 2) since o is on 2 it will not be included and our output will be Br

let lastName = fullName.slice(4,8)

* The reason the starting index is 4 and not 3 because we are including the space

let lastChar = fullName.slice(-1)

* We can also use a negative index by decreasing the number we will work our way backwords

let firstName = fullName.slice(0, fullName.indexOf(" "));

* Everything from the first index to the index of the first space

let lastName = fullName.slice( fullName.indexOf(" "));

* Start from the index that has a space and go till the end

# REFERENCE LIST

[1] ‘Learn JavaScript | Codecademy’. Accessed: Jun. 25, 2024. [Online]. Available: https://www.codecademy.com/enrolled/courses/introduction-to-javascript

[2] Bro Code, *JavaScript tutorial for beginners 🌐*, (Oct. 14, 2023). Accessed: Jul. 28, 2024. [Online Video]. Available: https://www.youtube.com/watch?v=Ihy0QziLDf0

[3] SuperSimpleDev, *JavaScript Tutorial Full Course - Beginner to Pro (2024)*, (May 09, 2024). Accessed: Jul. 28, 2024. [Online Video]. Available: https://www.youtube.com/watch?v=EerdGm-ehJQ

## Github Repo

[4] 2Kronos, *2Kronos/Javascript-Learning-Journey*. (Jul. 23, 2024). HTML. Accessed: Jul. 28, 2024. [Online]. Available: https://github.com/2Kronos/Javascript-Learning-Journey