Starting soon!

Intro to JS (pt. 2)

JFSS CS Club 2022/23
Arya

Recap

What is JS? What is its use?

It’s a programming language used both on client-side and server-side to make websites functional.

How do you use JS in your website?

```
<script src="script.js"></script>
```

What are the data types in JS?

String, Number, Boolean, Null, Undefined

What are JS functions?

“Chunk” of code used over and over again to avoid repetition and write clean code

How to access HTML elements in JS?

Document Object Model (DOM) Manipulation

Recap

What is **document.getElementById()** ?

Gets the element with the specific ID from the HTML document

What is **console.log()** ?

Logs whatever you want in the browser debugging console

What is element.**innerHTML** ?

Gets the inner content of the particular element

What is **Number(**value**) ?**

Turns the value given to a number. If value cannot be turned to a number, it outputs NaN

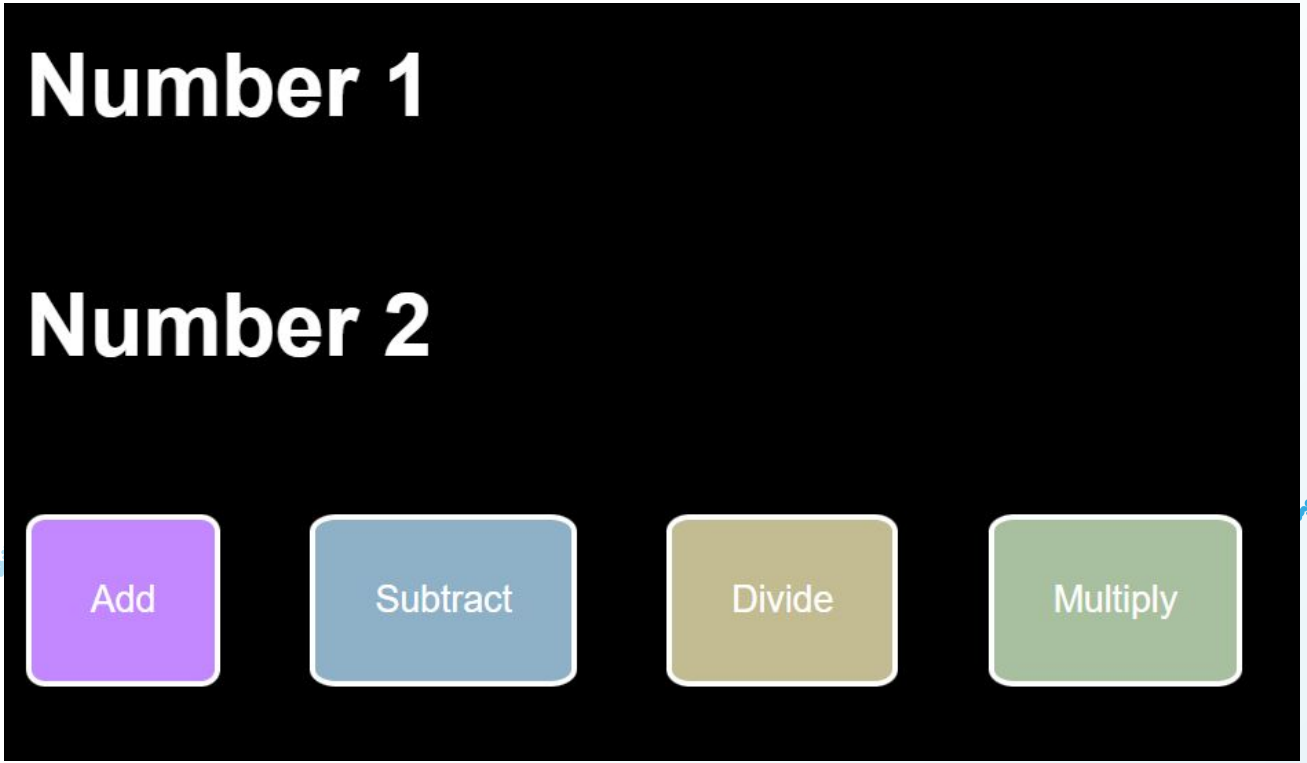
```
<button onClick="add( )">ADD</button>
<h1 id="result"></h1>
```

```
var num1 = document.getElementById("num1");
var num2 = document.getElementById("num2");
var result = document.getElementById("result")

// to get content inside the element
num1 = Number(num1.innerHTML);
num2 = Number(num2.innerHTML);

  function add(){
   result.innerHTML = num1 + num2;
  }
```

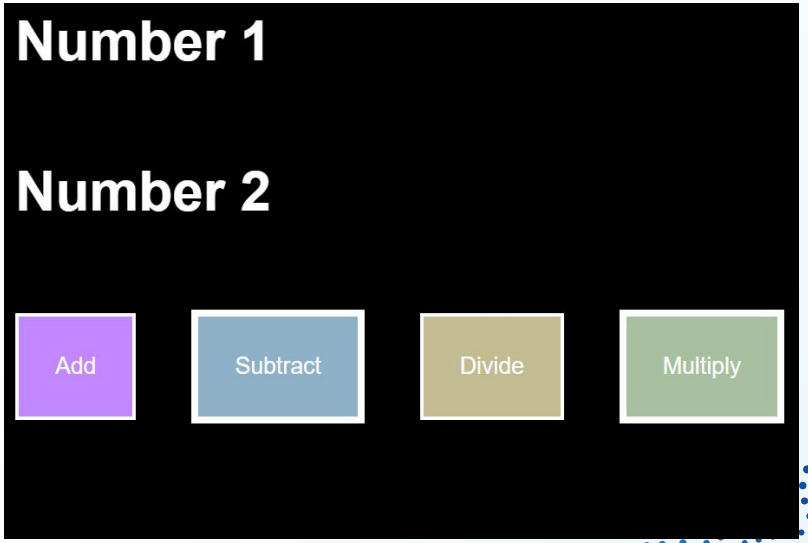
Activity Time



Activity Time

Make a Calculator (~20 minutes)

Clues will be dropped once in a while



- Access all the HTML data using document.getElementById("idName")
- Add different buttons for each
- Add a different function for each button
- Change the same HTML manipulation for each button
- Use the arithmetic operators (**+**, **-**, **/**, *****)

```
function subtract (){
  result.innerHTML = num1 - num2;
}
```

Activity Time

Answer

https://replit.com/@aryaholmukhe/JS

JavaScript Variables

- var
  Function scoped variable
  Ex: **var a = 2;**
- let
  Block scoped variable
  Ex: **let a = 2;**
- const
  Variable is constant (can't be changed)
  Ex: **const Pi = 3.1415925;**

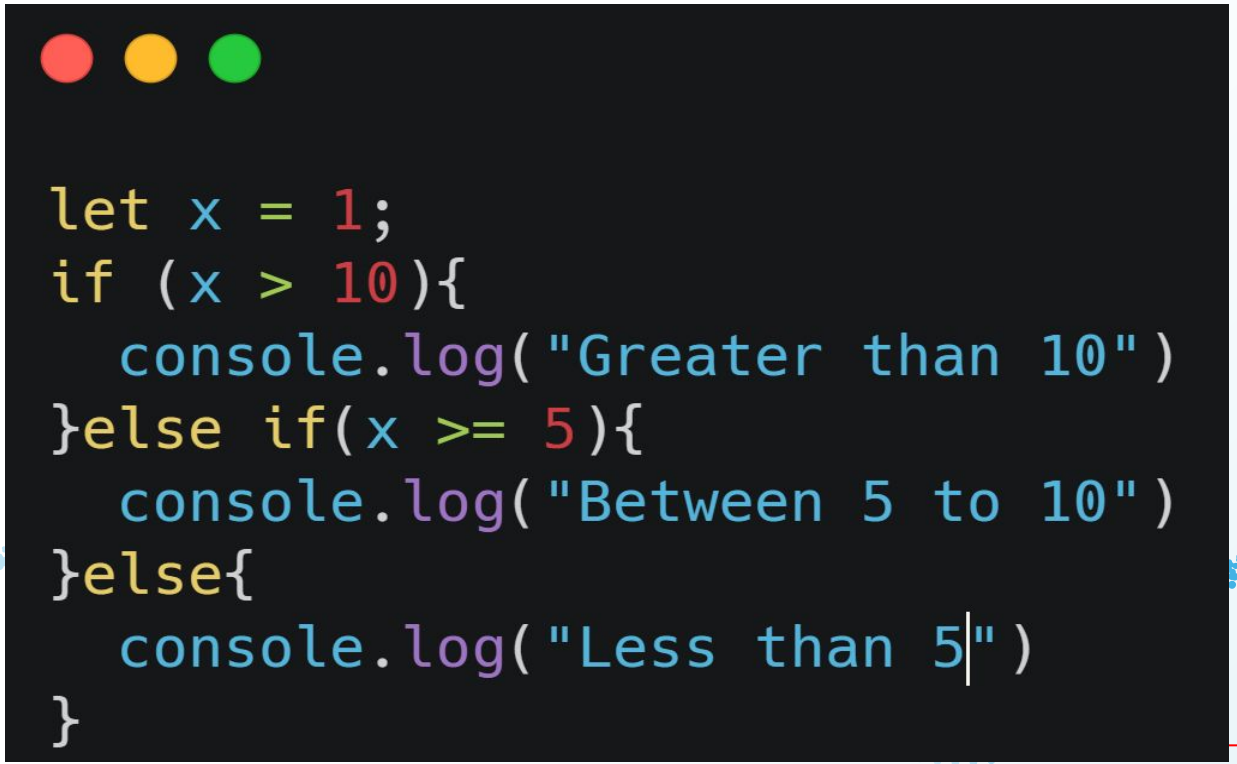
```
function greet() {
    // variable a can be used here
    var a = 'hello';
    console.log(a);
}
```

```
function greet() {
    let a = 'hello';

    // variable b cannot be used here
    if(a == 'hello'){
        // variable b can be used here
        let b = 'world';
        console.log(a + ' ' + b);
    }

     // variable b cannot be used here
    console.log(a + ' ' + b); // error
}
```

Conditionals



```
let x = 1;
if (x > 10){
  console.log("Greater than 10")
}else if(x >= 5){
  console.log("Between 5 to 10")
}else{
  console.log("Less than 5")
}
```

JS Loops

- The For loop

Expression 1 is executed (one time) before the execution of the code block.

Expression 2 defines the condition for executing the code block.

Expression 3 is executed (every time) after the code block has been executed.

```
for (expression 1; expression 2; expression 3) {
  // code block to be executed
}
```

JS Loops

- The For loop

```
for (let i = 0; i < 5; i++) {
  console.log(`The number is ${i}`);
}

/**
  * Output:
  * The number is 0
  * The number is 1
  * The number is 2
  * The number is 3
  * The number is 4
**/
```

JS Loops

- ## The While loop

  Loops as long as the condition is true

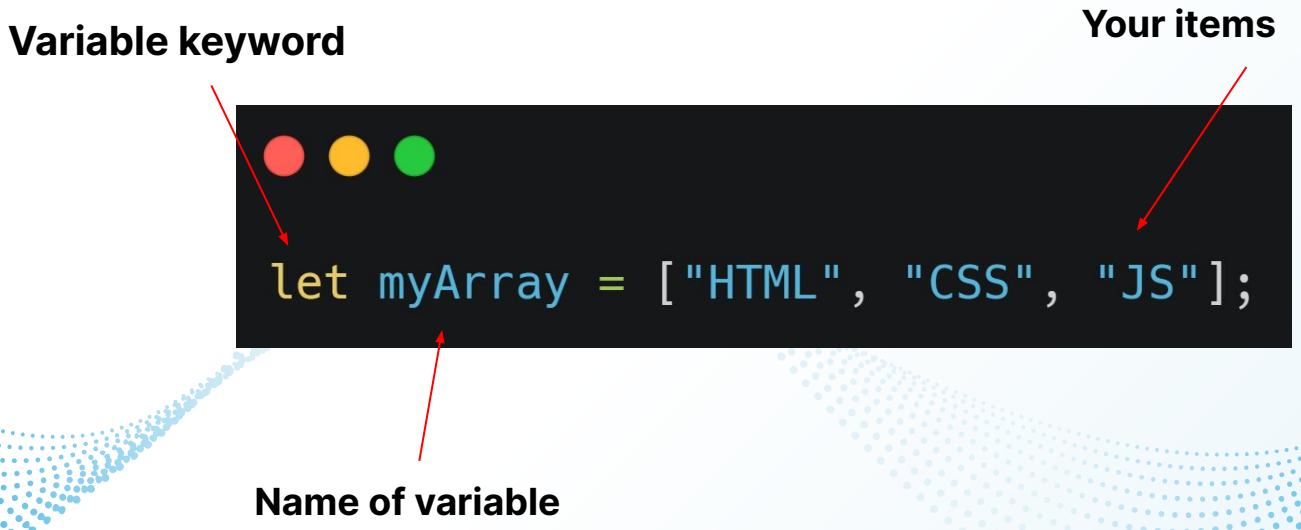
```
while (condition) {
  // code block to be executed
}
```

```
let i = 0;
while (i < 10) {
  console.log(`The number is ${i}`);
  i++;
}

/**
  * Output:
  * The number is 0
  * The number is 1
  * The number is 2
  * The number is 3
  * The number is 4
  * The number is 5
  * The number is 6
  * The number is 7
  * The number is 8
  * The number is 9
**/
```

JS Arrays

- To store and access multiple items



JS Array Methods

```
let numbers = ["zero", "one", "two", "three", "four", "five"];
console.log(numbers.length); //gets the length; output: 6
console.log(numbers[2]); //Output: "two"
numbers.pop(); //Removes the last item
number.push("five", "six"); //Adds to the end of the array
numbers.at(1) // Gets the value at the given index; output: one
```

numbers.at(x) vs numbers[x]?

- .at(x) allows for **negative numbers**

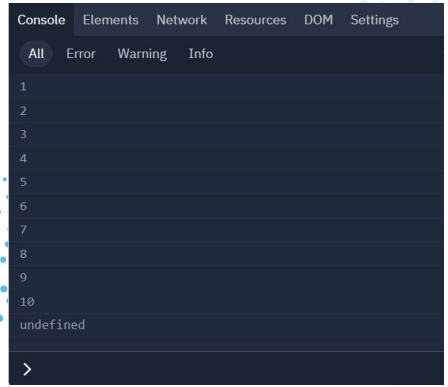
JS Array Methods

```
numbers = [3,5,6,2,5,6,4,2,1];
numbers.sort(function(a, b){return a - b});
//Output: [1, 2, 2, 3, 4, 5, 5, 6, 6]
```

Practice!

Beginner

Write an array with number from 1-10. Console log these numbers using one of the taught loops




```
oddNums([1, 2, 3, 4, 5, 6, 7])
// Output: 16
oddNums([1, 1, 1, 1, 1)
// Output: 5
```

Intermediate

Given an array of numbers, add all of the odd numbers and output the sum.

Advanced

Given an **array** and **n**, write a function which returns the **n**th smallest number.

```
nthSmallest([3,1,2], 2) //Output: 2

nthSmallest([15,20,7,10,4,3], 3) //Output: 7

nthSmallest([-5,-1,-6,-18], 4) //Output: -1

nthSmallest([2,1,3,3,1,2], 3) //Output: 2
```

Practice Answers

Beginner

```
let n = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
// With a while loop
let i = 0
while(i <= 10){
  console.log(n[i])
  i++;
}
// With a for loop
for ( let i = 0; i <= n.length; i++){
  console.log(n[i])
}
```

Intermediate

```
function oddNums(arr){
    sum = 0;
    for(i = 0; i<=arr.length-1; i++){
        if(arr[i] % 2 != 0){
            sum += arr[i];
            console.log(arr[i])
        }
    }
  return sum;
}
```

Advanced

```
function nthSmallest(arr, pos){
  return arr.sort((a,b)=>a-b)[pos-1];
}
```