

Date: 02-6-2020

Morning Session: 9 am – 11 PM

By ~ Sundeeep Charan Ramkumar Today

Topics: Functions

Functions : to perform a similar action in many places (removing redundancy)

Syntax :

```
//Declaration Function
function reusableFunction() {
  console.log("welcome")
}
//calling function
reusableFunction()
```

You can call or invoke this function by using its name followed by parentheses, like this: `reusableFunction()`; Each time the function is called it will print out the message "Hello World" on the dev console. All of the code between the curly braces will be executed every time the function is called.

methods Using functions inside objects:

```
var mr = {  
  name: "yodraj",  
  from: "hyderabad",  
  introduce: function () {  
    console.log(  
      "Hi I am " +  
      mr.name +  
      " and I am from " +  
      mr.from  
    );  
  }  
};  
console.log(mr.introduce());
```

Use all the built in methods:

Common methods for strings:

```
var name = "yodraj"  
console.log(name);
```

name.length() // it will give length of string 6.

name.toLowerCase // convert to lowercase answer // yodraj,

name.toUpperCase // convert to Uppercase // YODRAJ

name.slice // slice expects two inputs 1) start , 2) end // *string.slice(start, end)*

Start : Required. The position where to begin the extraction. First character is at position 0

End : Optional. The position (up to, but not including) where to end the extraction. If omitted, slice() selects all characters from the start-position to the end of the string // *name.slice(0,2)* // "yod"

name.indexOf // *name.indexOf(r)* // 3

name.replace // *name.replace("r" "h")* // yodhaj // its not changing the original value it will be a new value from it..

Functions in Arrays:

```
var a = [1,2,3,4];  
console.log(a);
```

Push() : Adds new elements to the end of an array, and returns the new length
// *a.push(5,6,7,8)* // output = [1,2,3,4,5,6,7,8]

pop() : Removes the last element of an array, and returns that element
//*a.pop()* // output = [1,2,3,4,5,6,7]

Unshift() : Adds new elements to the beginning of an array, and returns the new length // *a.unshift(8,9,10)* // output = [8,9,10,1,2,3,4,5,6,7]

concat() : Joins two or more arrays, and returns a copy of the joined arrays

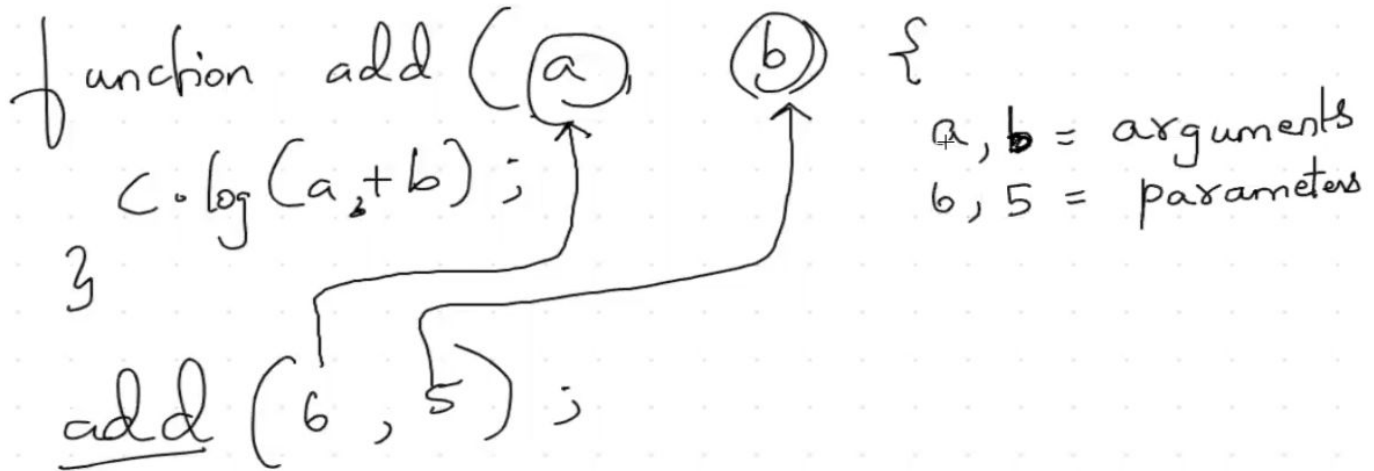
reverse() : Reverses the order of the elements in an array

methods Related to objects:

hasOwnProperty() : method returns a boolean indicating whether the object has the specified property as its own property

Built In Functions:

Accepting Inputs:



```
function addTwoNumbersWithInputs(number1, number2 = 10) {  
  console.log(typeof number2);  
  if (typeof number1 !== "number" || typeof number2 !== "number")  
    return "Invalid data";  
  else return number1 + number2;  
}
```

Note: step 1 (declaration) , step 2 (Assigning)

```
// If you are not using the return keyword, by default it will use undefined  
var key1 = "hi";  
var key2 = "there";  
var sum = addTwoNumbersWithInputs(key1, key2);  
console.log(sum);
```

```
> // Problem statement
// Input - Array of object
//           // { name: string, mark: number }
// Output - Array of object
//           // Containng the passing mark students
data
< undefined
```

```
> var studentsData = [
  { name: "Sundeep", mark: 45},
  {name: "Charan", mark: 47},
  {name: "Lakshay", mark: 91 },
  {name: "Aniket", mark: 93},
  {name: "Manjunath", mark: 94}
]

function filterStudentsMark (students) {
  var passingStudentsData = []
  for (studentData of students) {
    if (studentData.mark >= 50) {
      passingStudentsData.push(studentData)
    }
  }
  return passingStudentsData
}

filterStudentsMark(studentsData)
```

```
< ▼ (3) [{...}, {...}, {...}] ⓘ
  ► 0: {name: "Lakshay", mark: 91}
  ► 1: {name: "Aniket", mark: 93}
  ► 2: {name: "Manjunath", mark: 94}
    length: 3
  ► __proto__: Array(0)
```

```
>
```

