JavaScript

Jargons In Functions

@startwithmani

@M.serisha Kothapalli

swipe

Function Statement / Function Declaration

This is the regular traditional way of creating a function. where we have **function** keyword followed by **Name of the function** along with **paranthesis**. This is also called as **FUNCTION DECLARATION.**

Function Expression

Function which acts like a **value**...meaning you can intialize variable with function as a value.

Function statement

```javascript
function greet(){
  console.log('Hello User');
}
```

Function Expression

```javascript
var greet = function (){
  console.log('Hello User');
}
```

The main difference comes into picture in concept of hoisting

@startwithmani
@M.serisha Kothapalli

swipe

Anonymous Function

Function **without** any **name**...these doesn't have their own identity.

then how can you utilize this function without any name??

```
function(){
    console.log('Hello Mani');
}
```

When you want to assign function as a value.
We have utilized it in function expression in previous slide.

Named Function Expression

This is exactly same as **function expression** but rather than intializing it with anonymous function, we **intialize** it with **function with name** which is nothing but function statement.

```
var sum = function add(x, y){
        return x + y;
    }
```

The variable funName **add** is accessible within **function scope**, but not outside. So
if we try to call the function from outside, we get reference error

So if you try to call the above function with add() outside the scope....you will get an error....

@startwithmani
@M.serisha Kothapalli

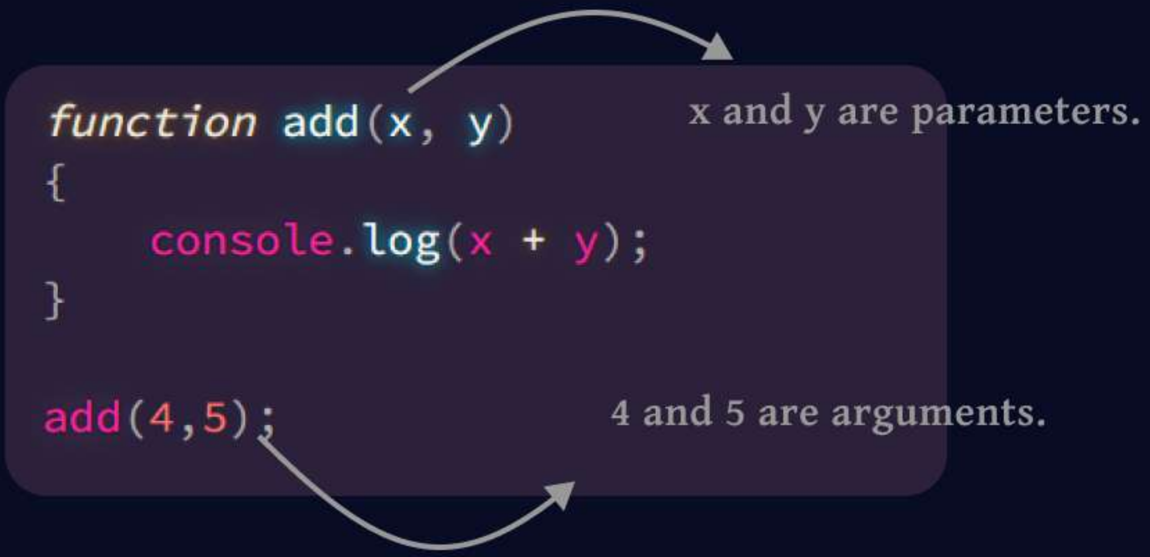
swipe

Parameters

Variables which you pass for function declaration.

Arguments

Variables which you pass for function call.



```
function add(x, y)
{
    console.log(x + y);
}

add(4,5);
```

x and y are parameters.

4 and 5 are arguments.

*** NOTE *** : **Never interchange these terms**

@startwithmani
@M.serisha Kothapalli

swipe

First Class Functions / First Class Citizens

This term is not just dedicated to JavaScript but to all programming languages where functions are treated like any other variable, Meaning....

- Function can be passed as an **argument** to other functions,
- can be **returned** by another function.
- can be **assigned as a value** to a variable.

Functions which have above abilities are called **FIRST CLASS FUNCTIONS**

Higher Order Functions

A function that accepts **functions** as **parameters** and/or **returns function**.

Callback Functions

A **function** which is passed as an **argument** into another function is called **callback function.**

Arrow Functions

**** NOTE ** :** Before I start explaining about arrow functions..There is one false statement which many people say is that function expressions and arrow functions are same.

No, Arrow functions are **alternatives** to **function expressions** with their own **limitations**.....I reiterate my sentence they are alternatives with limitations....they are **not completely same**.

ES6 arrow functions provide an alternative way to write a shorter syntax compared to the function expression.But they doesn't behave exactly same in all scenarios.

```
let sum = (a, b) => a + b;
```

@startwithmani
@M.serisha Kothapalli

swipe

IIFE - Immediately Invoked Function Expression

Immediately invoked function expression is a function defined as an expression and executed immediately after creation.

Meaning....

Function is **created** as an expression and **executed immediately**

- function will be **enclosed** within the Grouping Operator **().**
- An IIFE **can/cannot** have a **name**.
- you can also use **arrow functions**.

```
(function(a,b)
 {
   return a + b;

 })(10,20);
```

@startwithmani
@M.serisha Kothapalli

swipe

Did you find it helpful??

Like this post!

Share with your friends

Save it for later

Follow for more!

@startwithmani

@M.serisha Kothapalli