

Let's Learn JS Fundamentals

Agenda – (Functions)

- Functions Declarations
- Anonymous functions / Function expressions
- Arrow functions
- Immediately Invoked Function Expression (IIFE)
- Callback Function

Function Declaration

function keyword

function name

function parameters

```
function functionName(parm1, parm2,...) {  
    statement 3;  
    statement1;  
}  
  
return something; }  
}
```

function statements

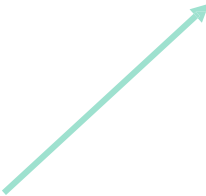
function return

Syntax:

```
function functionName (inputarguments) {  
    return stmt  
}  
  
functionName()
```

Anonymous Function /Function Expression (Func. with no name)

function with no name it is called a Anonymous Function



```
let funVar = function (inputarguments){  
  return stmt  
}
```

```
const greet = function() {}
```



name of the function is omitted, and stored in a variable 'greet' it becomes Function Expression

Arrow Functions



// arrow function expression

```
const add = (a, b) => {
```

```
    return a + b;
```

```
}
```

// very simple and concise syntax

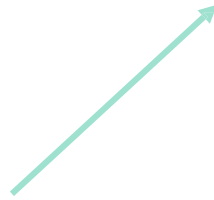
```
const add = (a, b) => a + b;
```

name of the function is replaced by “fat arrow”

```
let funVar = () => return
```

Immediately Invoked Function Expression (IIFE):

A function that runs as soon as it is defined



```
(function() {  
    console.log("IIFE");  
})  
  
();
```

Callback Function

The value of the variable call is passed using a callback function as an argument of another function

```
function greetStudent(name, call) {  
    console.log("Hello, " + name + "!");  
    call(); // Call the function passed as the callback  
}  
  
// Define a callback function  
  
function displayThanks() {  
    console.log("Thanks for learning with us!");  
}  
  
// Use the greetStudent function with the callback  
greetStudent("Everyone", displayThanks);
```

Key Takeaways

- How do we use functions to make reusable code?
- How does the function return something?