

JavaScript Functions

Function -

Function in any programming language is a block of code which is used for performing a particular kind of task. Similar to that, JavaScript has its own function creation capability. So these functions work when we call them after the time of creation.

A JavaScript function is defined with the `function` keyword, followed by a name, followed by parentheses `()`.

Function names can contain letters, digits, underscores, and dollar signs (same rules as variables).

The parentheses may include parameter names separated by commas: `(parameter1, parameter2, ...)`

The code to be executed, by the function, is placed inside curly brackets: `{ }`

Below are the syntaxes for function declaration:

```
function name(parameter1, parameter2, parameter3) {  
    // code to be executed  
}
```

- Function parameters are listed inside the parentheses `()` in the function definition.
- Function arguments are the values received by the function when it is invoked.
- Inside the function, the arguments (the parameters) behave as local variables.

Function Invoke -

Function invoke means calling the function. Code present inside function is executed when any function is invoked.

Below are the reasons of function invoking:

- When an event occurs (when a user clicks a button)
- When it is invoked (called) from JavaScript code
- Automatically (self invoked)

Function Return -

Return is a reserved keyword in JavaScript, which is used for stopping the execution of code. If the function was invoked from a statement, JavaScript will "return" to execute the code after the invoking statement.

Functions often compute a return value. The return value is "returned" back to the "caller".

There are different ways of function declaration in JavaScript.

- Function Expression
- Anonymous Functions
- Immediately Invoked Function Expression
- Constructor Functions
- Hoisting
- Arrow Functions

Function Expression -

A function can be declared using a function expression. It is declared quite differently from the general syntax because it uses a variable to denote the name of the function.

```
var greet = function () {  
    console.log("Welcome to Javascript");  
};  
greet();
```

Anonymous Functions -

Anonymous function declaration allows function names to appear hidden in the declaration itself. In the general function declaration syntax, the function name is attached to the function keyword, but an anonymous function is declared using only the function keyword and a parenthesis.

```
function () {  
    // function body  
}
```

Immediately Invoked Function Expression -

If you want to execute a function immediately after the declaration, use IIFE. This is executed by wrapping the anonymous function in parentheses and ending it with a semicolon.

```
(function () {  
    console.log("Welcome to Javascript");  
})();
```

Constructor Functions -

The concept of a function constructor is to create a function object which executes in the global scope. It can be used to create multiple objects that are similar. The function constructor has similar functionalities as the function expression. A constructor function is called with the new keyword to create an object.

```
var F = new Function(arg1, functionBody)  
var F = new Function(arg1, arg2, functionBody)  
var F = new Function(arg1, arg2, ....., argN, functionBody)
```

Hoisting -

With JavaScript functions, it is possible to call functions before actually writing the code for the function statement and they give a defined output. This property is called hoisting. Hoisting is the ability of a function to be invoked at the top of the script before it is declared.

```
greet();  
  
function greet() {  
    console.log("Hello world");  
}
```

Arrow Function -

This is a feature available in the ES6 version of JavaScript and as such has not stayed in the space for as long as the other features in the

function declaration. It is generally a cleaner way of creating JavaScript functions and it is similar to the function expression.

```
let name = (arguments1, arguments2, arguments 3...) => {  
  statements  
};
```

Above are the different ways of function declaration in javascript.