

Functions

Functions are the main “**building blocks**” of the program. They allow the code to be called many times without repetition.

Quite often we need to perform a similar action in many places of the script. Function is representation and implementation of task. Once we define a function once ,we can use it as many as we need.

Syntax:

```
function function-name ( parameter-1,parameter-2,.....,parameter-n)  
    {  
        Definition of function  
    }
```

“**function**” is the key word.

“**parameters**” are variables to receive data(arguments).

Ex:

```
function test( a)  
    {  
        console.log(“a is “,a);  
    }
```

Calling a function

A function can be executed only when it is get called. And also a function can be called for multiple times.

To call a function ,knowledge of function prototype is required. “**function prototype**” is the representation of function name ,parameters and return value of it.

Prototype of a function= name of function + parameters + return type

Ex:

```
//define function  
    function test( a)  
        {  
            console.log(“a is “,a);  
        }  
  
//call the function  
    test( 10 );// prints “a is 10”
```

Function with return value

A function can return the result after execution of its business logic. For this ,the ketword “return” is used.

Syntax:

```
return value;
```

Ex:

```
//function receives an argument and returns the same
```

```
function test( a)  
  {  
    return a;  
  }
```

When calling a function that return some value, it should hold by a variable.

```
let result= test( 100 );
```

Consider the following function which returns the sum of given arguments

```
function sumOfTwoNumbers( firstValue , secondValue)  
  {  
    let sum= firstValue + secondValue ;  
    return sum;  
  }  
  
//call the function  
let result= sumOfTwoNumbers( 100 ,200 );  
console.log(“sum is “,result);
```

Global and local variables

Variables which declared outside of all functions are called **global variables** which are accessible by the entire file.

Variables which declared inside of a function are called **local variables** and can be accessible only within the function. When we try to access local variables from outside of function,it results error.

Object

There are eight data types in JavaScript. Seven of them are called “primitive”, because their values contain only a single thing.

An object is a collection of related data and/or functionality.

Syntax:

```
{  
    key : value ,  
    key : value,  
    .....  
}
```

Ex:

Let us create a student object as object literal.

```
Let student= {  
    Sno:100,  
    Name: “ Alex “,  
    Age : 20  
};
```

Accessing values of an object

TO access values from an object we can use any of the following syntaxes.

Object-name . key

OR

Object-name[“key”]

Ex:

```
student.sno; // gives 100
```

```
student[“sno”]; // gives 100
```

Deleting key,value pair from an object:

Syntax:

```
delete object-name . key;
```

Ex: to remove “city” from student object,use

```
delete student . city ;
```

Adding or modifying an object

To add/modify an object , we use following syntax

To add:

Object-name . new-key = new-value;

To modify:

Object-name. key = new-value ;

Creating object using “Object constructor”

Syntax:

let object-name = new Object ();

Here , “**Object**” is a class and it’s constructor is executed to return a empty object.

It will create an empty object and we can add key,value pairs to it.

Ex:

//create an empty object employee

let employee= new Object();

//add key , value pairs to the above empty object

employee . empNo =100;

employee. Name="Alex";