

HIGHER ORDER FUNCTION

A “higher-order function” is a function that accepts functions as parameters and/or returns a function.

```
Function demo( ){  
  Clg("hello this is demo starts");  
  returns function( ){  
    clg("HOF")  
  }  
}  
  
demo( )( );
```

CALLBACK FUNCTION

A javascript callback function is function which is to be executed after another function has finished execution

A more formal definition would be - Any function that is passed as an argument to another function so that it can be executed in that other function is called as a callback function.

```
<script>  
  function per1(name,callback){  
    return function(){  
      console.log('my name is '+ name)  
    }  
    callback()  
  }  
  function per2(){  
    console.log('callback function')  
  }  
  let per=per1("manoj",per2())
```

```
per1()  
console.log(per());  
  
</script>
```

OBJECT

JavaScript Objects

A JavaScript object is an entity having state and behavior (properties and method). For example: car, pen, bike, chair, glass, keyboard, monitor etc.

JavaScript is an object-based language. Everything is an object in JavaScript.

JavaScript is template based not class based. Here, we don't create class to get the object. But, we directly create objects.

We can create an object by using 3 ways

1. By using literals

- Create a single object, using an object literal

Syntax:

```
var obj={  
    color:"white",  
    width:123,  
}
```

2. Create an object by using new keyword

- Create a single object, using new keyword.

Syntax:

```
var obj1=new Object();
```

3. Create an object by using constructor function

Syntax:

```
function person(name,age,sal){
```

```
    this.name=name;  
    this.age=age;  
    this.sal=sal;  
}
```

