

Javascript Notes

__/__/__

Notes by Ipwebdevelopers.in

==
" Functions "

- Functions are building blocks in Javascript.
- A function is a self-contained block of statement that performs a specific task.
- A Programmer can invoke a function at different places in a script whenever it is needed of writing the same statements again and again

Syntax

function keyword function name ([par 1] , [par 2])
{
statement [s];
}

- Functions are defined using function keyword, followed by a function name and comma separated list of one or more optional parameters enclosed in parentheses.

==
" Javascript function Example "

```
<script>  
function msg()  
{
```

//_

```
    alert (" hello ! jpwebdevelopers");  
    }  
</script>
```

```
<input type="button" onclick="msg()" value=  
    "Print">
```

* Javascript Function Parameters

We can call function by passing arguments.

Example We pass parameters to calculate the sum of two numbers.

```
<html>
```

```
<head>
```

```
<title> sum of two numbers </title>
```

```
<script language="Javascript">
```

```
    function sum (a,b)
```

```
    {
```

```
        c = a+b;
```

```
        alert (" sum is " c);
```

```
    }
```

```
</script>
```

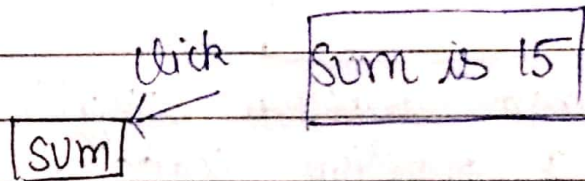
```
</head>
```



```
<body>  
<input type = "button" value = "sum"  
onclick = "sum(5,10); ">
```

```
</body>  
</html>
```

Output



* Function Invoke,

- The Javascript function invoke a function is used to execute a function contains the value and an argument provided individually
- The code inside a function is executed when the function is invoked.

Syntax

```
function myfunction (var)  
{  
    return var;  
}  
myfunction (value);
```

Example

— / — / —

```
return a * b;
```

3

// will return 20.

* function apply() method

- The Javascript function `apply()` method is used to call a function contains the value and an argument. contains element of an array

Syntax :- function.apply(^{optional}this Arg - [^{array like object.}Array])
this value is given for the call to a function. (Optional).

Example :-

```
<html>
<body>
<script>
var arr = [7, 5, 9, 1];
var max = Math.max.apply(null, arr);
document.writeln(max);
</script>
</body>
</html>
```

Output :- 9

* Return Statement :-

The return statement is used to return a value from the function to the place where the function was called.

Syntax :- `return (value);`

→ Here value is value returned by the function which is optional.

Example

```
<html>
<head>
  <script language = "javascript">
    function area(side)
    {
      C = side * side;
      return C;
    }
  </script>
</head>
<body>
  <script language = "javascript">
    d = area(5);
    document.write("area = " + d);
  </script>
</body>
</html>
```


Output

area = 25

* Javascript closures

- A closure can be defined as a javascript feature in which the inner function has access to the outer function variable.
- A closure is an inner function that has access to the outer function's variables.

For every closure we have three scopes:-

◦ Local scope

◦ Outer function scope

◦ Global Scope

Example

```
<html>
<head>
<script>
function fun()
{
var a = 4;
function innerfun()
{
return a;
}
return innerfun ;
}
var output = fun();
document.write(output());
document.writeln(output());
</script>
</head>
</body>
</html>
```

Output

44