WAP INSTITUTE
LEARN SOFTWARE ENGINEERING

https://wapinstitute.com

---

75 Days Hard

JavaScript Notes

JavaScript

JavaScript is a programming language for adding functionality to web pages, allowing developers to create effects and controll the website user interfaces

Keywords or Reserve Words

In JavaScript, a keyword is a specific type of word used to convey the meaning of code to the compiler or interpreter.

List of some keywords

1. var – for creating a variable in javascript
2. const – for creating a constant in javascript
3. function – for creating a function in javascript

Variables

Variable is used to store a data in memory with the help of programming language

Example

```
<script>
    var x = 1234;
    alert(x)
<script>
```

Constant

Constant is aslo used to store a data in memory with the help of programming language but constant is not updatable

Example

```
<script>
    const x = 1234;
    alert(x)
<script>
```

Output Posiblities & Statements – if you want to display an any result in browser you have to use any of them

1. alert()
2. document.write()
3. console.log()
4. console.warn()
5. console.error()
6. innerHTML

Function

Functions are the way to organise or structure your code in a block in proramming languages

Example

```
<script>
    function demo()
    {
        // Your code goes here
    }
<script>
```

Function Components

Functions has two components

1. Function definition

2. Function call

Example

```
<script>
    // This is function definition
    function demo()
    {
        // Your code goes here
    }

   // This is function call
   demo()
<script>
```

Programming language

A programming language is a set of instructions or commands used to create software and instruct computers to perform specific tasks or functions.

Levels of Programming language

1. Low level – Used to create operating systems
2. Middle level – Used to create operating systems and application software
3. High level – Used to create application softwares

Compiler

Compiler is a type of software that translates code written in a programming language into machine code. It processes the entire code at once, translating it into a form that the computer can directly execute.

Interpreter

Interpreter is also a type of software that translates code written in a programming language into machine code. It processes the code line by line, translating it into a form that the computer can directly execute.

Function

A function is a self-contained unit of code designed to perform a specific task or set of tasks, allowing for the execution of a series of instructions in response to particular needs or events.

Types of Function

1. User defined function
2. Built-in function

User defined function

A function which is created by user or developer according to use cases in software development process

Components of User defined function

1. **Function definition :-** Means how to make a function

```
function demo() {
}
```

2. **Function call –** To executed to your created function
   demo()

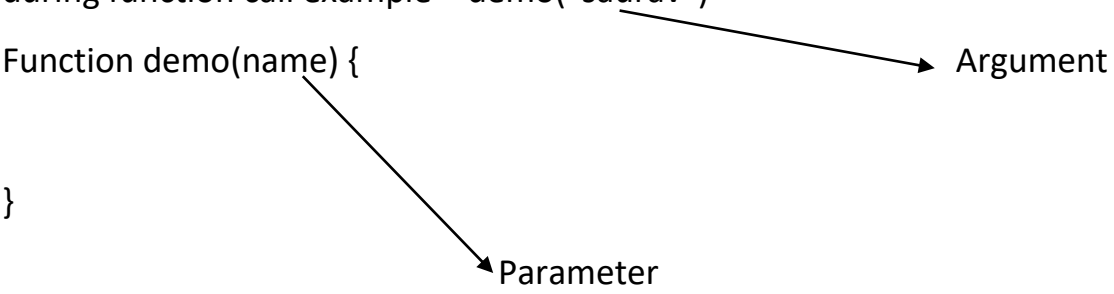
Call by value

demo(“saurav”) – this is the example of **call by value** whenever you pass any data during function call that process are called **call by value**

Argument

demo(“saurav”) – here “saurav” is an argument

Parameter – here **name** is parameter. Parameter is used to receive the data which passed during function call example – demo(“saurav”)



Events

Events is an action of mouse and keyboard performes by an user to call a function

Mouse events

1. onclick
2. ondblclick
3. onmouseover
4. onmouseout
5. onmousemove
6. onmouseup
7. onmousedown
8. oncontextmenu

example

```
<button onclick=”functionCall()”>test</button>
```

Selectors in JavaScript

1. document.body -> To select body tag
2. document.getElementById(“test”) -> To select an element or html tag with that id value
3. document.forms -> To select all the form tags of html page at one time

Document.body example

In this example we are changing the background color of body tag on button click

Some sample code of html page

```
<body>
    <button onclick="changeColor()">change color</button>
</body>
```

Js page code

```
function changeColor()
{
    var body = document.body;
     body.style.background = "red";
}
```

Document.getElementById example

In this example we are changing the font size of h1 tag on button click

Some sample code of html page

```
<body>
    <h1 id="title">CodingOtt</h1>
    <button onclick="changeSize()">change size</button>
</body>
```

Js page code

```
function changeSize()
{
    var h1 = document.getElementById("title");
     h1.style.fontSize = "80px";
}
```

Document.forms example

In this example we are reading the values of form fields on form submit

Some sample code of html page

```
<body>
    <form onsubmit="login(event)" name="loginForm">
        <input type="text" name="username" />
        <input type="text" name="password" />
        <button>submit</button>
    </form>
</body>
```

Js page code

```
function login(event)
{
   event.preventDefault()
   var form = document.forms.loginForm;
   var username = form.elements.username;
   var password = form.elements.password;
   console.log(username.value);
   console.log(password.value);
}
```

Reading input value without form tag on button click

Some sample code of html page

```
<body>
    <input type="text" id="age" />
   <button onclick="findAge()">find age</button>
</body>
```

Js page code

```
function findAge()
{
   var input = document.getElementById("age");
   alert(input.value)
}
```

Update input on button click

Some sample code of html page

```
<body>
    <input type="text" id="age" />
   <button onclick="updateAge()">find age</button>
</body>
```

Js page code

```
function updateAge()
{
   var input = document.getElementById("age");
   input.value = '22 Years';
}
```

Add or remove class value using className and classList

Example using **className**

```
<body>
    <p id="para">I am testing</p>
    <button onclick="addClass()">add class</button>
    <button onclick="removeClass()">remove class</button>
</body>

function addClass() {
    var para = document.getElementById("para");
    para.className = "democlass";
}

function removeClass() {
    var para = document.getElementById("para");
    para.className = "";
}
```

Example using **classList**

```
 <body>
      <p id="para">I am testing</p>
      <button onclick="addClass()">add class</button>
      <button onclick="removeClass()">remove class</button>
 </body>

 function addClass() {
    var para = document.getElementById("para");
     para.classList.add('firstclass','secondclass');
 }

 function removeClass() {
    var para = document.getElementById("para");
     para.classList.remove('firstclass','secondclass');
}
```

JavaScript Conditions

Conditions are the technique to take software logic decisions inside a programming language

Syntax & Uses

```
If(10 == 10) {
  alert("success");
}
else {
 alert("Failed");
}
```

Result -> Success (because 10 == 10 is always true)

Example 2nd

```
If(10 != 10) {
    alert(“success”);
}
else {
   alert(“Failed”);
}
```

Result -> Failed (because 10 != 10 is false)