

ऑनलाइन पाठ्य सामग्री

2DCA2

INTERNET & E-COMMERCE

इकाई – चार

सुश्री तुलना त्रिवेदी
फैकल्टी, कंप्यूटर विज्ञान एवं अनुप्रयोग
प्रशांत पाराशर
ट्यूटर, प्रबंधन विभाग



माखनलाल चतुर्वेदी राष्ट्रीय पत्रकारिता एवं संचार विश्वविद्यालय

B -38, विकास भवन , एम पी नगर, जोन -I, भोपाल

1. जावास्क्रिप्ट

जावास्क्रिप्ट एक बहुत शक्तिशाली क्लाइंट-साइड स्क्रिप्टिंग भाषा है। प्रारंभ में जावास्क्रिप्ट "वेब पेजेस को जीवंत बनाने" के लिए बनाया गया था। जावास्क्रिप्ट का उपयोग मुख्य रूप से वेबपेज के साथ एक उपयोगकर्ता की संवादात्मकता को बढ़ाने के लिए किया जाता है। दूसरे शब्दों में, वेबपेज जावास्क्रिप्ट की मदद से अधिक जीवंत और संवादात्मक हो जाता है। गेम और मोबाइल एप्लिकेशन डेवलपमेंट में भी जावास्क्रिप्ट का व्यापक रूप से उपयोग किया जा रहा है।

1.1 जावास्क्रिप्ट का इतिहास

जावास्क्रिप्ट 1995 में ब्रेंडन ईच द्वारा विकसित किया गया था, जो उस समय के एक लोकप्रिय ब्राउज़र नेटस्केप में दिखाई दिया था। भाषा को शुरू में लाइवस्क्रिप्ट कहा गया था और बाद में इसका नाम बदलकर जावास्क्रिप्ट रखा गया था। उस समय जावा बहुत लोकप्रिय था, इसलिए यह तय किया गया कि जावा के "छोटे भाई" के रूप में एक नई भाषा को प्रारंभ करने में काफी मदद मिलेगी। जावास्क्रिप्ट का जैसे जैसे विकास हुआ यह अपनी विशिष्टता, जिसे ECMA स्क्रिप्ट कहते हैं, के साथ एक पूर्ण स्वतंत्र भाषा के रूप में विकसित हो गई और अब इसका जावा से कोई संबंध नहीं है। कई प्रोग्रामर हैं, जो यह सोचते हैं, कि जावास्क्रिप्ट और जावा समान हैं। वास्तव में, जावास्क्रिप्ट और जावा में कोई संबंध नहीं है। जावा एक बहुत ही जटिल प्रोग्रामिंग भाषा है, जबकि जावास्क्रिप्ट केवल एक स्क्रिप्टिंग भाषा है। जावास्क्रिप्ट का ज्यादातर सिंटैक्स प्रोग्रामिंग लैंग्वेज "C" से प्रभावित है।

1.2 रनिंग जावास्क्रिप्ट

एक स्क्रिप्टिंग भाषा होने के नाते, जावास्क्रिप्ट स्वयं नहीं चल सकता है। वास्तव में, जावास्क्रिप्ट कोड चलाने के लिए ब्राउज़र की आवश्यकता होती है । जब कोई उपयोगकर्ता एक HTML पेज का अनुरोध जावास्क्रिप्ट के साथ करता है, तो स्क्रिप्ट ब्राउज़र को भेजी जाती है और ब्राउज़र द्वारा इसे निष्पादित किया जाता है।

जावास्क्रिप्ट का मुख्य लाभ यह है, कि सभी आधुनिक वेब ब्राउज़र जावास्क्रिप्ट का समर्थन करते हैं और साइट विजिटर चाहे इंटरनेट एक्सप्लोरर, गूगल क्रोम, फ़ायरफ़ॉक्स या किसी अन्य ब्राउज़र का उपयोग क्यों न करते हो, जावास्क्रिप्ट सभी वेब ब्राउज़र पर निष्पादित किया जा सकता है। इसके अलावा, जावास्क्रिप्ट विंडोज, लिनक्स या मैक सहित किसी भी ऑपरेटिंग सिस्टम पर चलता है। इस प्रकार, जावास्क्रिप्ट VBScript (अब पदावनत) के मुख्य नुकसान की भरपाई करता है, जो सिर्फ़ IE और विंडोज तक सीमित है।

आज, जावास्क्रिप्ट न केवल ब्राउज़र में, बल्कि सर्वर पर, या वास्तव में किसी भी डिवाइस पर निष्पादित किया जा सकता है, जिसमें एक विशेष प्रोग्राम होता है, जिसे जावास्क्रिप्ट इंजन कहा जाता है। ब्राउज़र में एक एम्बेडेड इंजन होता है, जिसे कभी-कभी "जावास्क्रिप्ट वर्चुअल मशीन" कहा जाता है।

1.3 टूल्स की आवश्यकता

आरंभ में , वेब पेजेस को प्रदर्शित करने के लिए ब्राउज़र और कोड लिखने के लिए एक टेक्स्ट एडिटर की आवश्यकता होती है। नोटपैड ++, विज़ुअल स्टूडियो कोड, सबलाइम टेक्स्ट , एटम या किसी भी अन्य टेक्स्ट एडिटर के साथ अपनी पसंद के टेक्स्ट एडिटर का उपयोग कर सकते हैं, जिसके साथ आप सहज हैं। गूगल क्रोम , फ़ायर फ़ॉक्स,

माइक्रोसॉफ्ट एज , इंटरनेट एक्स्प्लोरर आदि किसी भी वेब ब्राउज़र का उपयोग कर सकते हैं ।

2. सिंटेक्स और कन्वेंशन

अगर जावास्क्रिप्ट कोड को HTML डॉक्यूमेंट में ही रखा जाता है, तो जावास्क्रिप्ट कोड को <script> टैग्स (<script>और </ script>) के भीतर रखें । यह टैग ब्राउजर को जावास्क्रिप्ट कोड और बाकी कोड को अलग करने में मदद करता है। चूंकि अन्य क्लाइंट-साइड स्क्रिप्टिंग भाषाएं उपयोग में हैं, उदाहरण के लिए : VBScript, इसीलिए यह अनुशंसा की जाती है, कि उपयोग की गई स्क्रिप्टिंग भाषा को निर्दिष्ट किया जाए । <Script> टैग के भीतर टाइप एट्रिब्यूट का उपयोग करें और इसके मान को टेक्स्ट / जावास्क्रिप्ट पर सेट करें।

```
<script type = "text / javascript">
```

हैलो वर्ल्ड उदाहरण

```
<html>
```

```
<head>
```

```
<title> My First JavaScript Code !!! </ title >
```

```
<script type = "text / javascript">
```

```
alert ("Hello World!");
```

```
</ script>
```

```
</ head>
```

```
<body>
```

```
</ body>
```

```
</ html>
```

नोट: HTML5 में टाइप = "टेक्स्ट / जावास्क्रिप्ट" आवश्यक नहीं है।

जावास्क्रिप्ट के बारे में कम से कम तीन प्रमुख चीजें हैं-

- 1) HTML / CSS के साथ पूर्ण एकीकरण।
- 2) साधारण चीजें साधारण तरीके से की जाती हैं।
- 3) सभी प्रमुख ब्राउज़रों द्वारा जावास्क्रिप्ट निष्पादित किया जाता है और डिफ़ॉल्ट रूप से सक्षम है ।

जावास्क्रिप्ट एकमात्र ब्राउज़र तकनीक है, जो इन तीन चीजों को जोड़ती है।

जावास्क्रिप्ट ब्राउज़र इंटरफ़ेस बनाने के लिए सबसे व्यापक टूल है। जावास्क्रिप्ट सर्वर, मोबाइल एप्लिकेशन आदि बनाने की भी अनुमति देता है। यही विशेषता जावास्क्रिप्ट को अद्वितीय बनाता है। जावास्क्रिप्ट को शुरुआत में केवल ब्राउज़र की भाषा के रूप में बनाया गया था, लेकिन अब इसे कई अन्य एनवायरनमेंट में भी उपयोग किया जाता है। आजकल जावास्क्रिप्ट का एक अद्वितीय स्थान है, क्योंकि जावास्क्रिप्ट HTML / CSS के पूर्ण एकीकरण के साथ सबसे व्यापक रूप से अपनाया जाने वाला ब्राउज़र भाषा के रूप में स्थापित है ।

3. वैरिएबल्स

जावास्क्रिप्ट वैरिएबल्स केवल भंडारण स्थान (storage location) का एक नाम है। वैरिएबल्स का मतलब है, वो कुछ भी, जो बदल सकता है। जावास्क्रिप्ट में वैरिएबल शामिल होते हैं, जो डेटा वैल्यू रखते हैं और इसे कभी भी बदला जा सकता है। एक वैरिएबल घोषित करने के लिए जावास्क्रिप्ट आरक्षित कीवर्ड वैर (var) का उपयोग करता है। एक वैरिएबल का एक अद्वितीय नाम होना चाहिए। जब यह घोषित किया जाता है या उपयोग करने से पहले इक्वल टू (=) ऑपरेटर का उपयोग करके किसी वैरिएबल को वैल्यू असाइन (assign) किया जा सकता है।

3.1 सिंटैक्स

```
var<variable-name>;
```

```
var< variable-name> = <value>;
```

जावास्क्रिप्ट में वैरिएबल (जिन्हें आइडेंटिफायर के रूप में भी जाना जाता है) को घोषित करते समय कुछ नियम लागू होते हैं

1)नाम को एक अक्षर से शुरू होना चाहिए (a to z या A से Z), अंडरस्कोर (_), या डॉलर (\$) चिन्ह।

2) पहले अक्षर के बाद अंकों (0 से 9) का उपयोग किया जा सकता है, उदाहरण के लिए
value1।

3) जावास्क्रिप्ट वैरिएबल केस सेंसिटिव है, उदाहरण के लिए x और X अलग वैरिएबल हैं।

सही जावास्क्रिप्ट वैरिएबल

```
var x = 10;
```

```
var_val = "Bhopal";
```

गलत जावास्क्रिप्ट वैरिएबल

```
var 123 = 30;
```

```
var *aa = 320;
```

3.2 जावास्क्रिप्ट वैरिएबल का उदाहरण

जावास्क्रिप्ट वैरिएबल का एक सरल उदाहरण देखते हैं।

```
<Script>
```

```
var x = 10;
```

```
var y = 20;
```

```
var z = x + y;  
document.write (Z);  
</ Script>
```

3.3 जावास्क्रिप्ट में दो प्रकार के वैरिएबल हैं -

- लोकल वैरिएबल ।
- ग्लोबल वैरिएबल ।

3.3.1 जावास्क्रिप्ट लोकल वैरिएबल

एक जावास्क्रिप्ट लोकल वैरिएबल ब्लॉक या फ़ंक्शन के अंदर घोषित किया जाता है। यह केवल फ़ंक्शन या ब्लॉक के भीतर ही उपलब्ध है। उदाहरण के लिए -

```
<Script>  
function abc () {  
var x = 10; // लोकल वैरिएबल  
}  
</ Script>
```

या,

```
<Script>  
if (10 <13) {  
var y = 20; // जावास्क्रिप्ट लोकल वैरिएबल  
}  
</ Script>
```

3.3.2 जावास्क्रिप्ट ग्लोबल वैरिएबल

एक जावास्क्रिप्ट ग्लोबल वैरिएबल किसी भी फंक्शन से प्राप्य है। एक वैरिएबल जो कि फंक्शन के बाहर घोषित किया गया या विंडो ऑब्जेक्ट के साथ घोषित किया गया है, उसे ग्लोबल वैरिएबल के रूप में जाना जाता है। उदाहरण के लिए:

```
<Script>

var data = 200; // ग्लोबल वैरिएबल

function a () {

document.write (data );

}

function b(){

document.write(data);

}

a (); // जावास्क्रिप्ट फंक्शन को कॉल करना

b ();

</ Script>
```

3.3.3 याद रखने लायक बिंदु -

- 1) वैरिएबल एकल डेटा वैल्यू संग्रहीत करता है, जिसे बाद में बदला जा सकता है।
- 2) उपयोग करने से पहले वैरिएबल को इनिशियलाइज़ किया जाना चाहिए।
- 3) एक ही लाइन में कई वैरिएबल्स को परिभाषित किया जा सकता है।

जैसे var one = 1, two = 2, three = "three";

4) जावास्क्रिप्ट में वैरिएबल, शिथिल टाइप के वैरिएबल हैं। यह किसी भी डेटा टाइप की वैल्यू को अपने जीवन काल में कभी भी स्टोर कर सकता है।

4. एक्सप्रेसंस

कोड की कोई भी इकाई जो कि मूल्यांकित होकर एक वैल्यू देती है , एक्सप्रेसंस कहलाती है । चूंकि एक्सप्रेसंस वैल्यू को प्रोड्यूस करते हैं, वे कहीं भी एक प्रोग्राम में दिखाई दे सकते हैं, जहां जावास्क्रिप्ट एक वैल्यू की अपेक्षा करता है, जैसे कि "फंक्शन इनवोकेशन के आर्गुमेंट" । मूल रूप से जावास्क्रिप्ट में निम्नलिखित एक्सप्रेसन श्रेणियां हैं -

4.1 अरिथमेटिक एक्सप्रेसंस -

अरिथमेटिक एक्सप्रेसंस एक संख्यात्मक मान का मूल्यांकन करती हैं। उदाहरणों में निम्नलिखित शामिल हैं।

`20;` // यहाँ 20 एक एक्सप्रेसन है ,जो जावास्क्रिप्ट इंटरप्रेटर द्वारा संख्यात्मक मान 20 का मूल्यांकन करता है।

`20 + 13;` // यह एक और एक्सप्रेसन है, जिसका मूल्यांकन संख्यात्मक मूल्य 33 को प्रोड्यूस करने के लिए किया जाता है।

4.2 स्ट्रिंग एक्सप्रेसंस

स्ट्रिंग एक्सप्रेसंस एक स्ट्रिंग का मूल्यांकन करने वाले एक्सप्रेसन हैं। उदाहरणों में निम्नलिखित शामिल हैं -

`'hi';`

`'hello' + 'students';` // "Hello students" स्ट्रिंग का मूल्यांकन करता है।

4.3 लॉजिकल एक्सप्रेसंस

बूलियन वैल्यू "true" या "false" मूल्यांकन करने वाले एक्सप्रेसन को लॉजिकल एक्सप्रेसन माना जाता है। एक्सप्रेसंस के इस सेट में अक्सर लॉजिकल ऑपरेटरों && (AND), || (OR) and !(NOT) का उपयोग शामिल होता है। उदाहरण इस प्रकार है -

```
12 > 9; // बूलियन वैल्यू के लिए मूल्यांकन सही है।
```

```
20 < 10; // बूलियन वैल्यू के लिए मूल्यांकन गलत है।
```

```
true; // बूलियन मूल्य के लिए सही मूल्यांकन है।
```

```
a === 20 && b === 30; // a और b के वैल्यूज के आधार पर सही या गलत का मूल्यांकन करता है।
```

4.4 लेफ्ट - हैंड - साइड एक्सप्रेसंस

यह एल वैल्यूज के रूप में भी जाना जाता है। लेफ्ट - हैंड - साइड एक्सप्रेसन वे हैं, जो असाइनमेंट एक्सप्रेसन के बाईं ओर दिखाई देते हैं। लेफ्ट हैंड साइड एक्सप्रेसन के उदाहरणों में निम्नलिखित शामिल हैं -

```
// वैरिएबल जैसे कि i और total
```

```
i = 10;
```

```
total = 0;
```

```
// अर्रे के तत्व
```

```
array[0] = 20;
```

```
array[1] = 'hello';
```

4.5 असाइनमेंट एक्सप्रेसंस

जब किसी वैरिएबल को वैल्यू असाइन करने के लिए एक्सप्रेसन = ऑपरेटर का उपयोग करते हैं, तो इसे असाइनमेंट एक्सप्रेसन कहा जाता है। उदाहरण इस प्रकार है -

```
avg = 55;
```

`var b = (a = 1);` // यहाँ असाइनमेंट एक्सप्रेशन ($a = 1$) मूल्यांकित होकर एक वैल्यू प्रदान करता है, जो कि `b` को असाइन की जाती है, `b = (a = 1)` एक अन्य असाइनमेंट एक्सप्रेशन है। `var` एक्सप्रेशन का हिस्सा नहीं है।

= ऑपरेटर अपने लेफ्ट-साइड ऑपरेंड के रूप में एक एल वैल्यू की अपेक्षा करता है। असाइनमेंट एक्सप्रेशन का वैल्यू राइट-साइड ऑपरेंड की वैल्यू है, जैसे उपरोक्त उदाहरण में `55`। साइड इफेक्ट के रूप में, = ऑपरेटर बाईं ओर के वैल्यू के लिए दाईं ओर वैल्यू प्रदान करता है।

5. ब्रांचिंग एंड लूपिंग स्टेटमेंट

5.1 ब्रांचिंग स्टेटमेंट

विभिन्न स्थितियों के आधार पर निष्पादन के प्रवाह को तय करने के लिए कंडीशनल स्टेटमेंट्स का उपयोग किया जाता है। यदि कोई स्थिति सत्य है, तो एक क्रिया करें और यदि स्थिति असत्य है, तो दूसरी क्रिया करें।

जावास्क्रिप्ट में मुख्य रूप से तीन प्रकार के कंडीशनल स्टेटमेंट्स हैं।

1) **इफ स्टेटमेंट** (If statement)

2) **इफ एल्स स्टेटमेंट** (If Else statement)

3) **इफ एल्स इफ एल्स स्टेटमेंट** (If Else If Else statement)

5.1.1 इफ स्टेटमेंट (IF statement)

IF statement एक कंडीशनल ब्रांचिंग स्टेटमेंट है। IF statement में, यदि कंडीशन सही (True) है, तो स्टेटमेंट का एक समूह निष्पादित होता है। और यदि कंडीशन असत्य (False) है, तो निम्नलिखित कथनों को छोड़ दिया जाता है।

सिंटेक्स - इफस्टेटमेंट

If (condition)

{

```
//Statement 1;  
//Statement 2;  
}
```

Example:

: IF स्टेटमेंट के लिए सरल प्रोग्राम

```
<html>  
<body>  
<script type="text/javascript">  
var num = prompt("Enter Number");  
if (num > 0) {  
alert("Given number is Positive!!!"); }  
</script>  
</body>  
</html>
```

5.1.2 इफ एल्स स्टेटमेंट (If Else Statement)

इफ एल्स दो-तरफा डिसीजन स्टेटमेंट है। इसका उपयोग डिसीजन लेने और स्टेटमेंट को कण्डीशनली निष्पादित करने के लिए किया जाता है।

सिंटैक्स: इफ-एल्स स्टेटमेंट

```
if (condition)  
{  
    // one Statement or number of statements  
}  
else  
{  
    // one Statement or number of statements  
}
```

उदाहरण :

Example :

```
<html>  
<head>  
<title> If...Else Statments!!! </title>  
<script type="text/javascript">  
    // Get the current hours
```

```

var hours = new Date().getHours();
if(hours<12)
document.write("Good Morning!!!<br />");
else
document.write("Good Afternoon!!!<br />");
</script>
</head>
<body>
</body>
</html>

```

5.1.3 इफ एल्स इफ- एल्स स्टेटमेंट (If Else If Else statement)

इफ एल्स इफ- एल्सस्टेटमेंट का उपयोग दो या दो से अधिक कंडीशंस की जाँच करने के लिए किया जाता है।

सिंटैक्स:इफएल्सइफ-एल्सस्टेटमेंट

```

If (condition)
{
//one,Statementornumberofstatements
}
elseif (condition)
{
//oneStatementornumberofstatements
}
else
{
//oneStatementornumberofstatements
}

```

Example:

```

<html>
<head>
<script type="text/javascript">
var one = prompt("Enter the first number");
var two = prompt("Enter the second number");
one = parseInt(one);
two = parseInt(two);

```

```
if (one == two)
document.write(one + " is equal to " + two + ".");
else if (one<two)
document.write(one + " is less than " + two + ".");
else
document.write(one + " is greater than " + two + ".");
</script>
</head>
<body>
</body>
```

5.1.4 स्विच स्टेटमेंट

स्विच का उपयोग विभिन्न परिस्थितियों पर विभिन्न क्रियाओं को करने के लिए किया जाता है। इसका उपयोग एक ही एक्सप्रेशन की कई अलग-अलग वैल्यू से तुलना करने के लिए किया जाता है। एक ब्रेक स्टेटमेंट का उपयोग प्रत्येक केस स्टेटमेंट के साथ किया जाता है।

Syntax : Switch statement

```
switch(expression) {
case condition 1:
    //Statements;
break;
case condition 2:
    //Statements;
break;
case condition 3:
    //Statements;
break;
    case condition n:
        //Statements;
break;
default:
    //Statement;
}
```

Example : Simple Program for Switch Statement

```
<html>
<head>
<script type="text/javascript">
var day = prompt("Enter a number between 1 and 7");
switch (day)
{
case (day="1"):
document.write("Sunday");
break;
case (day="2"):
document.write("Monday");
break;
case (day="3"):
document.write("Tuesday");
break;
case (day="4"):
document.write("Wednesday");
break;
case (day="5"):
document.write("Thursday");
break;
case (day="6"):
document.write("Friday");
break;
case (day="7"):
document.write("Saturday");
break;
default:
document.write("Invalid Weekday");
break;
}
</script>
</head>
</html>
```

5.2 लूपिंग स्टेटमेंट

जावास्क्रिप्ट लूप का उपयोग बार-बार कोड के एक ब्लॉक को चलाने के लिए किया जाता है, जब तक कि एक निश्चित कंडीशन पूरी नहीं हो जाती। जावास्क्रिप्ट व्हाइल , डू व्हाइल और फॉर सहित कोड के एक ब्लॉक को चलाने के लिए कई विकल्प प्रदान करता है।

5.2.1 व्हाइल लूप

व्हाइल लूप एक एंट्री-नियंत्रित लूप स्टेटमेंट है। यह जावास्क्रिप्ट में सबसे बुनियादी लूप है। यह किसी स्टेटमेंट को बार-बार निष्पादित करता है, जब तक कि एकसप्रेशन सत्य है। एक बार जब एकसप्रेशन असत्य हो जाता है, तो लूप समाप्त हो जाता है।

सिंटैक्स

```
while (condition)
{
    //Statements;
}
```

Example : Fibonacci Series Program using While Loop

```
<html>
<body>
<script type="text/javascript">
var num1=0,num2=1,num3=0;
document.write("Fibonacci Series:"+ "<br>");
while (num2<=10)
{ num3 = num 1+ num 2;
num 1 = num 2;
num 2 = num 3;
document.write(num 3 + " <br>");
}
</script>
</body>
</html>
```


5.2.2 डू-व्हाइल लूप

डू -व्हाइल लूप एक एग्जिट कंट्रोल्ड लूप स्टेटमेंट है। व्हाइल लूप के ही समान ही है, डू -व्हाइल, केवल अंतर यह है कि डू -व्हाइल लूप के अंत में जाँच की जाती है । लूप कम से कम एक बार निष्पादित होता है, भले ही स्थिति असत्य (फॉल्स) हो।

सिंटैक्स

```
do
{
    // Statement ;
}
```

While (Condition) ;

Example : Simple Program using Do-While Loop

```
<html>
```

```
<body>
```

```
<script type ="text/javascript">
```

```
vari = 0;
```

```
do
```

```
{
```

```
document.write(i+"<br>")
```

```
i++;
```

```
}
```

```
while (i<= 5);
```

```
</script>
```

```
</body>
```

```
</html>
```

व्हाइल लूप और डू व्हाइल लूप के बीच अंतर

| व्हाइल लूप | डू व्हाइल लूप |
|--|--|
| व्हाइल लूप में , पहले यह कंडीशन की जांच करता है और फिर प्रोग्राम को निष्पादित करता है। | डू व्हाइल लूप में, पहले यह प्रोग्राम को निष्पादित करता है और फिर कंडीशन की जांच करता है। |
| यह एक एंट्री कंट्रोल्ड लूप है। | यह एक एग्जिट कंट्रोल्ड लूप है। |
| बॉडी से पहले कंडीशन आएगी । | बॉडी के बाद कंडीशन आएगी । |
| यदि कंडीशन असत्य है, तो यह लूप को समाप्त करेगा । | यह कम से कम एक बार चलता है, भले ही कंडीशन असत्य हो। |
| यह एक काउंटर-कंट्रोल्ड लूप है। | यह एक ईटेरेटिव कंट्रोल्ड लूप है। |

5.2.3 फॉर लूप

फॉर लूप लूपिंग का एक कॉम्पैक्ट रूप है। इसमें तीन महत्वपूर्ण भाग शामिल हैं:

1. लूप प्रारंभ करना
2. कंडीशन को टेस्ट करना
3. ईट्रेशन

ये तीनों भाग अर्धविराम (;) द्वारा अलग की गई एक ही पंक्ति में आते हैं।

सिंटैक्स

```
for(initialization; test-condition; increment/decrement)
```

```
{
```

```
    //Statements;
```

```
}
```

Example : Palindrome Program using For Loop

```
<html>
<body>
<script type="text/javascript">
function  palindrome()
{ varrevstr = "";
varstr = document.getElementById("strr").value;
vari = strr.length;
for(var j=i; j>=0; j--)
    {
revstr = revstr+strr.charAt(j);
    }
if(strr == revstr)
    {
alert(strr+" - is Palindrome");
    }
else
    {
alert(strr+" - is not a Palindrome");
    }
}
</script>
```

```
<form>
```

```
Enter a String or Number: <input type="text" id="strr"  
name="checkpalindrome"><br>
```

```
<input type="submit" value="Check" onclick="palindrome();">
```

```
</form>
```

```
</body>
```

```
</html>
```

5.2.4 ब्रेक एवं कंटिन्यू स्टेटमेंट

5.2.4.1 ब्रेक स्टेटमेंट

ब्रेक स्टेटमेंट का उपयोग लूप से बाहर निकलने के लिए किया जाता है। यह जल्दी से एक लूप से बाहर निकलने के लिए उपयोग किया जाता है, जो कर्ली ब्रेसिज़ के ब्लॉक को तोड़ता है।

सिंटैक्स - break;

5.2.4.2 कंटिन्यू स्टेटमेंट

कंटिन्यू स्टेटमेंट अगले लूप के साथ लूप को जारी रखता है। यह ब्लॉक के शेष कोड को छोड़ देता है।

सिंटैक्स - continue;

6. जावास्क्रिप्ट फ़ंक्शन

फ़ंक्शन स्टेटमेंट का एक सेट है, जो इनपुट्स लेता है, कुछ विशिष्ट गणना करता है, और आउटपुट उत्पन्न करता है। मूल रूप से, एक फ़ंक्शन स्टेटमेंट का एक सेट है, जो कुछ कार्यों को करता है या कुछ गणना करता है और फिर यूजर को परिणाम देता है।

अभिप्राय यह है कि कुछ सामान्य या बार-बार किए जाने वाले कार्य को एक साथ रखा जाए और एक फ़ंक्शन बनाया जाए, ताकि विभिन्न इनपुट्स के लिए एक ही कोड को बार-बार लिखने के बजाय, फ़ंक्शन को कॉल किया जा सके।

अन्य प्रोग्रामिंग भाषाओं की तरह, जावास्क्रिप्ट भी फंक्शन के उपयोग का समर्थन करता है। जावास्क्रिप्ट बिल्ट-इन फंक्शन और यूजर-डिफ़ाइंड फंक्शन दोनों को सपोर्ट करता है। उदाहरण के लिए अलर्ट () जावास्क्रिप्ट में एक अंतर्निहित फंक्शन है। लेकिन जावास्क्रिप्ट हमें यूजर के द्वारा परिभाषित फंक्शन्स को भी बनाने की अनुमति देता है।

6.1 फंक्शन की परिभाषा/डिक्लेरेशन

जावास्क्रिप्ट में यूजर -परिभाषित फंक्शन का उपयोग करने से पहले, इसे बनाना आवश्यक है। फंक्शन की परिभाषा को कभी-कभी फंक्शन डिक्लेरेशन या फंक्शन स्टेटमेंट भी कहा जाता है।

जावास्क्रिप्ट में एक फंक्शन बनाने के नियम -

- 1) प्रत्येक फंक्शन को कीवर्ड फंक्शन के साथ शुरू होना चाहिए।
- 2) एक यूजर परिभाषित फंक्शन नेम जो अद्वितीय होना चाहिए।
- 3) कोष्ठक के भीतर संलग्न पैरामीटर की एक सूची अल्पविराम द्वारा अलग होनी चाहिए।
- 4) कर्ली ब्रेसिज़ के भीतर संलग्न फंक्शन के बॉडी स्टेटमेंट की एक सूची होनी चाहिए {}।

जावास्क्रिप्ट में एक फंक्शन बनाने के लिए मूल सिंटैक्स नीचे दिया गया है।

Syntax:

```
function function name(Parameter1, Parameter2, ..)
```

```
{
```

```
    // Function body
```

```
}
```

6.2 फ़ंक्शन पैरामीटर

पैरामीटर एक फ़ंक्शन को दी गई अतिरिक्त जानकारी हैं। वे फ़ंक्शन नेम के बाद कोष्ठक के भीतर फ़ंक्शन में पास किये जाते हैं और कॉमा के द्वारा अलग होते हैं। जावास्क्रिप्ट में एक फ़ंक्शन में किसी भी संख्या में पैरामीटर हो सकते हैं और साथ ही साथ इसमें एक भी पैरामीटर नहीं हो सकता है। उदाहरण के लिए, फ़ंक्शन calcAddition का कार्य दो संख्याओं की गणना करना है। जिन दो नंबरों पर एडिशन ऑपरेशन किया जाना है, उन्हें इस फ़ंक्शन के पैरामीटर के रूप में पास किया जाता है।

उदाहरण –

```
function calcAddition(number1, number2)
{
  return number1 + number2;
}
```

उपरोक्त उदाहरण में, calcAddition नामक एक फ़ंक्शन बनाया गया है। यह फ़ंक्शन दो नंबरों को पैरामीटर्स के रूप में स्वीकार करता है और इन दो नंबरों के एडिशन को वापस करता है।

6.3 कॉलिंग फ़ंक्शन

किसी फ़ंक्शन को परिभाषित करने के बाद, अगला चरण फ़ंक्शन का उपयोग करने के लिए कॉल करना है। एक फ़ंक्शन को फ़ंक्शन नेम का उपयोग करके कॉल किया जा सकता है, जिसमें कोष्ठक के बीच संलग्न पैरामीटर की वैल्यू को अल्पविराम से अलग किया जाता है और अंत में अर्धविराम लगाया जाता है। नीचे दिया गया सिंटैक्स यह दिखाता है कि जावास्क्रिप्ट में फ़ंक्शन को कैसे कॉल करते हैं -

सिंटैक्स

```
functionname (Value1, Value2, ..);
```

उदाहरण

```
<html>
```

```
<script type = "text/javascript">
```

```
// Function definition
```

```
function welcomeMsg(name) {
```

```
    document.write("Hello " + name + " welcome to Computer Department");
```

```
}
```

```
    // creating a variable
```

```
    varnVal = "Admin";
```

```
    // calling the function
```

```
    welcomeMsg(nVal);
```

```
</script>
```

```
</html>
```

Output:

Hello Admin welcome to Computer Department

6.4 रिटर्न स्टेटमेंट

कुछ परिस्थितियां एसी होती हैं, जहां कुछ ऑपरेशन करने के बाद कुछ वैल्यूज को किसी फ़ंक्शन से वापस करना पड़ता है। ऐसे मामलों में, जावास्क्रिप्ट में रिटर्न स्टेटमेंट का उपयोग किया जाता है। यह एक वैकल्पिक स्टेटमेंट है और अधिकांश समय किसी जावास्क्रिप्ट फ़ंक्शन में अंतिम स्टेटमेंट होता है। पहले उदाहरण calcAddition को नामित फ़ंक्शन के साथ देखें। यह फ़ंक्शन दो संख्याओं की गणना करता है और फिर

परिणाम वापस करता है। रिटर्न स्टेटमेंट का उपयोग करने का सबसे बुनियादी सिंटैक्स है-

सिंटैक्स

```
return value;
```

रिटर्न स्टेटमेंट की शुरुआत कीवर्ड Return से होती है, जिसे फंक्शन से वापस किया जाता है। वैल्यू को सीधे वापस करने के बजाय एक एक्सप्रेशन का भी उपयोग किया जा सकता है।

7. ऐरे ऑब्जेक्ट

ऐरे ऑब्जेक्ट एक वैरिएबल में कई वैल्यूज को संग्रहीत करता है। यह एक ही प्रकार के तत्वों के एक निश्चित आकार के अनुक्रमिक संग्रह(sequential collection) को संग्रहीत करता है। एक ऐरे का उपयोग डेटा के कलेक्शन को संग्रहीत करने के लिए किया जाता है, लेकिन एक ऐरे को एक ही प्रकार के वैरिएबल के कलेक्शन के रूप में समझना अक्सर अधिक उपयोगी होता है।

सिंटैक्स

```
vararray_name = [item1, item2, ...];
```

उदाहरण

```
var fruits = new Array ( "apple", "orange", "mango" );
```

ऐरे पैरामीटर स्ट्रिंग्स या इन्टिजर की एक सूची है। जब ऐरे कंस्ट्रक्टर के साथ एक एकल संख्यात्मक पैरामीटर निर्दिष्ट किया जाता है, तो ऐरे की प्रारंभिक लंबाई निर्दिष्ट की जाती है। किसी ऐरे की अधिकतम लंबाई 4,294,967,295 है।

ऐरे को केवल वैल्यूज को निर्दिष्ट करके बनाया जा सकता है -

```
var fruits = [ "apple", "orange", "mango" ];
```


उपयोग करने के लिए क्रमिक संख्याओं का उपयोग करें और एक ऐरे के अंदर वैल्यूज को निम्नानुसार सेट करें –

fruits [0] पहला Element है।

fruits [1] दूसरा Element है।

fruits [2] तीसरा Element है।

7.1 ऐरे प्रॉपर्टीज

यहाँ उनके विवरण के साथ ऐरे ऑब्जेक्ट के डिस्क्रिप्शन की एक सूची दी गई है-

| अनुक्रमांक | प्रॉपर्टी | डिस्क्रिप्शन |
|------------|-------------|--|
| 1 | कंस्ट्रक्टर | ऐरे फ़ंक्शन का एक संदर्भ देता है, जिसने ऑब्जेक्ट बनाया है। |
| 2 | इंडेक्स | प्रॉपर्टी मैच की गई स्ट्रिंग के शून्य-आधारित सूचकांक का प्रतिनिधित्व करती है। |
| 3 | इनपुट | यह प्रॉपर्टी केवल मैच किये गए नियमित एक्सप्रेसंस द्वारा बनाई गई ऐरे में मौजूद है। |
| 4 | लंबाई | किसी ऐरे में एलिमेंट्स की संख्या को दर्शाता है। |
| 5 | प्रोटोटाइप | प्रोटोटाइप प्रॉपर्टी आपको किसी ऑब्जेक्ट में प्रॉपर्टी और विधियाँ जोड़ने की अनुमति देता है।ब्रेसिज़ |

8. इवेंट की अवधारणा

HTML के साथ जावास्क्रिप्ट का इंटरक्शन उन घटनाओं के माध्यम से नियंत्रित किया जाता है ,जब उपयोगकर्ता या ब्राउज़र किसी पेज में बदलाव करते हैं।

जब पेज लोड होता है, तो इसे एक ईवेंट कहा जाता है। जब उपयोगकर्ता एक बटन क्लिक करता है, तो वह क्लिक भी एक इवेंट है। अन्य उदाहरणों में किसी भी की(Key) को दबाने, एक विंडो को बंद करने, एक विंडो का आकार बदलने आदि जैसी इवेंट्स शामिल हैं।

डेवलपर्स इन इवेंट्स का उपयोग जावास्क्रिप्ट कोडेड प्रतिक्रियाओं को निष्पादित करने के लिए कर सकते हैं, जिससे विंडोज बंद करने के लिए बटन, उपयोगकर्ताओं को प्रदर्शित होने वाले संदेश, मान्य किए जाने वाले डेटा और वस्तुतः किसी अन्य प्रकार की प्रतिक्रिया कल्पनाशील करने योग्य बनाती हैं।

इवेंट्स डॉक्यूमेंट ऑब्जेक्ट मॉडल (DOM) लेवल 3 का एक हिस्सा हैं और हर HTML एलिमेंट में इवेंट्स का एक सेट होता है, जो जावास्क्रिप्ट कोड को ट्रिगर कर सकता है।

इवेंट और जावास्क्रिप्ट के बीच संबंध को समझने के लिए यहां कुछ उदाहरण दिए गए हैं -

जब कोई उपयोगकर्ता वेबसाइट पर जाता है, तो वो टेक्स्ट और इमेजेस पर क्लिक करता है और दिए गए लिंक पर क्लिक करता है , चीजों पर होवर आदि जैसे कार्य करता है। इस तरह के उदाहरण को जावास्क्रिप्ट इवेंट्स कहते हैं। इवेंट हैंडलर को जावास्क्रिप्ट में लिखा जा सकता है और इन इवेंट हैंडलर को इवेंट टैग ऐट्रिब्यूट्स की वैल्यू के रूप में निर्दिष्ट कर सकते हैं। HTML 4.01 स्पेसिफिकेशन नीचे सूचीबद्ध 19 इवेंट्स ऐट्रिब्यूट्स को परिभाषित करता है -

8.1 <body >और <frameset > level Events

केवल दो ऐट्रिब्यूट्स हैं जिनका उपयोग किसी भी जावास्क्रिप्ट कोड को ट्रिगर करने के लिए किया जा सकता है जब कोई इवेंट डॉक्यूमेंट लेवल पर होता है।

| ऐट्रिब्यूट्स | वैल्यू | डिस्क्रिप्शन |
|--------------|-----------|---|
| ऑनलोड | स्क्रिप्ट | जब HTML डॉक्यूमेंट लोड होता है तब स्क्रिप्ट चलती है। |
| ऑनअनलोड | स्क्रिप्ट | जब HTML डॉक्यूमेंट अनलोड होता है, तो स्क्रिप्ट चलती है। |

नोट - यहाँ स्क्रिप्ट किसी भी जावास्क्रिप्ट फ़ंक्शन या कोड के भाग का संदर्भ देती है।

8.2 <form > लेवल इवेंट्स

निम्नलिखित छह ऐट्रिब्यूट्स हैं, जिनका उपयोग किसी भी जावास्क्रिप्ट कोड को ट्रिगर करने के लिए किया जा सकता है, जब कोई इवेंट फॉर्म लेवल पर होता है।

| ऐट्रिब्यूट्स | वैल्यू | डिस्क्रिप्शन |
|--------------|-----------|--|
| ऑनचेंज | स्क्रिप्ट | जब एलिमेंट बदलता है तब स्क्रिप्ट चलती है। |
| ऑनसबमिट | स्क्रिप्ट | फॉर्म सबमिट होने पर स्क्रिप्ट चलती है। |
| ऑनरिसेट | स्क्रिप्ट | फॉर्म रीसेट होने पर स्क्रिप्ट चलती है। |
| ऑनसेलेक्ट | स्क्रिप्ट | एलिमेंट सेलेक्ट होता है, तब स्क्रिप्ट चलती है। |
| ऑनब्लर | स्क्रिप्ट | जब एलिमेंट फोकस खो देता है, तब स्क्रिप्ट चलती है। |
| ऑनफोकस | स्क्रिप्ट | जब एलिमेंट को फोकस मिलता है, तब स्क्रिप्ट चलती है। |

8.3 कीबोर्ड इवेंट

कीबोर्ड द्वारा उत्पन्न तीन इवेंट्स का अनुसरण किया जाता है। ये इवेंट base , br, frame , frameset , head , html, iframe, meta , param , script , style और title एलिमेंट्स में मान्य नहीं हैं।

| ऐट्रिब्यूट्स | वैल्यू | डिस्क्रिप्शन |
|--------------|-----------|--|
| ऑनकीडाउन | स्क्रिप्ट | जब की (key) को दबाया जाता है तो स्क्रिप्ट चलती है |
| ऑनकीप्रेस | स्क्रिप्ट | जब की(Key) को दबाया और रिलीज़ किया जाता है तब स्क्रिप्ट चलती है। |
| ऑनकीअप | स्क्रिप्ट | जब की (key) रिलीस होती है, तब स्क्रिप्ट चलती है। |

8.4 अन्य इवेंट्स

अन्य 7 इवेंट्स जो किसी HTML टैग के संपर्क में आने पर माउस द्वारा उत्पन्न होती हैं। ये इवेंट base , br, frame , frameset , head , html, iframe, meta , param , script , style और title एलिमेंट्स में मान्य नहीं हैं।

| एट्रिब्यूट्स | वैल्यू | डिस्क्रिप्शन |
|--------------|-----------|--|
| ऑनक्लिक | स्क्रिप्ट | जब माउस क्लिक होता है, तो स्क्रिप्ट चलती है। |
| ऑनडीब्लिक | स्क्रिप्ट | जब माउस डबल-क्लिक करता है, तो स्क्रिप्ट चलती है। |
| ऑनमाउसडाउन | स्क्रिप्ट | जब माउस बटन दबाया जाता है तो स्क्रिप्ट चलती है। |
| ऑनमाउसमूव | स्क्रिप्ट | जब माउस पॉइंटर चलता है तो स्क्रिप्ट चलती है। |
| ऑनमाउसआउट | स्क्रिप्ट | जब माउस पॉइंटर किसी एलिमेंट से बाहर जाता है, तो स्क्रिप्ट चलती है। |
| ऑनमाउसओवर | स्क्रिप्ट | जब माउस पॉइंटर किसी एलिमेंट पर चलता है, तो स्क्रिप्ट चलती है। |
| ऑनमाउसअप | स्क्रिप्ट | जब माउस बटन रिलीस किया जाता है, तो स्क्रिप्ट चलती है। |

8.5 ऑनक्लिक इवेंट: यह एक माउस इवेंट है और किसी डिफाइंड लॉजिक को उत्पन्न करता है, यदि उपयोगकर्ता उस तत्व पर क्लिक करता है, जिसके लिए वह बाध्य है।

```
<html>
<head>
  <script type = "text/javascript">
    function hi() {
      alert('Don't press again!');
    }
  </script>
</head>
<body>
```

```
<button type="button" onclick="hi()">Click me event</button>
</body>
</html>
```

जब बटन प्रेस किया जाता है, तो फ़ंक्शन hi() कॉल होता है ।

8.6 ऑनमाउसओवर इवेंट -यह इवेंट तब होता है, जब माउस पॉइंटर को एलिमेंट या उसके किसी एक बच्चे पर ले जाया जाता है, जिसके लिए वह बाध्य होता है।

```
<html>
<head>
  <script type = "text/javascript">
    function hov(e) {
      e.style.display = 'none';
    }
  </script>
</head>
<body>
  
  style="background-color:green;height:200px;width:200px;"
</body>
</html>
```

आउटपुट:

माउस को इमेज पर ले जाने से पहले इमेज की स्टाइल दिखाई देती है।

जब माउस को इमेज पर ले जाते हैं ,तो इमेज की स्टाइल गायब हो जाती है।

8.7 ऑनसबमिट इवेंट: यह इवेंट तब होती है, जब कोई फॉर्म सबमिट किया जाता है।

```
<html>
<head>
  <script type = "text/javascript">
    function hov() {
      alert("The form was submitted");
    }
  </script>
</head>
<body>
```

<p> When form is submitted , a function is triggered which alerts some text.</p>

```
<form action = "/action_page.php" onsubmit = "hov()">
  Enter name : <input type = "text" name = "fname">
  <input type = "submit" value = "Submit">
</form>
</body>
</html>
```

आउटपुट:

जब आप फॉर्म सबमिट करते हैं, तो एक फंक्शन ट्रिगर होता है, जो अलर्ट में कुछ टेक्स्ट देता है।

नाम दर्ज करे -

8.8 ऑनफोकस ईवेंट - इस ईवेंट को सूचीबद्ध करने वाला एलिमेंट निर्देशों को निष्पादित करता है ,जब भी वह फोकस को प्राप्त करता है। इस ईवेंट का उपयोग अक्सर <input>, <select> के साथ किया जाता है।

ऑनफोकस इवेंट ऑनब्लर इवेंट के विपरीत है।

```
<html>
<head>
  <script type = "text/javascript">
    function focused(x) {
      x.style.background = "green";
    }
  </script>
</head>
<body>
  <p >Take the focus into the input box below:</p>
  Enter your name : <input id="inp" type = "text" onfocus="focused(this)">
</body>
</html>
```

नाम दर्ज करे -

8.9 ऑनचेंज इवेंट

जब एलिमेंट्स की वैल्यू बदलती है और सूची से नए वैल्यू का चयन किया जाता है, तो ऑनचेंज इवेंट ऐट्रिब्यूट्स काम करता है। यह इवेंट ऐट्रिब्यूट्स का एक हिस्सा है। यह ऑनइनपुट इवेंट ऐट्रिब्यूट के समान है। लेकिन अंतर यह है कि ऑनइनपुट ऐट्रिब्यूट इवेंट एलिमेंट परिवर्तन के तुरंत बाद होता है, जबकि ऑनचेंज इवेंट तब होता है, जब एलिमेंट फोकस खो देता है। यह ऐट्रिब्यूट `<select>` एलिमेंट से संबद्ध है।

```
<html>
<head>
<title>onchange event attribute</title>
<style>
body {
text-align:center;
  }
  h1 {
color:green;
  }
</style>
<script>
function selection() {
var x = document.getElementById("GFG").value;
document.getElementById("sd").innerHTML =
"Selected Subject: " + x;
  }
</script>
</head>
<body>
<h2>onchange Event Attribute</h2>
<p>Choose Subject:</p>
<select id="GFG"onchange="selection()">
<option value="Data Structure">Data Structure
```

```
<option value="Java">Java
<option value="Computer Network">Computer Network
<option value="Operating System">Operating System
<option value="HTML">HTML
</select>
<p id="sd"></p>
</body>
</html>
```

8.10 ऑनब्लर इवेंट- ऑनब्लर इवेंट तब होती है जब कोई ऑब्जेक्ट फोकस खो देता है। यह इवेंट अक्सर फॉर्म वेलिडेशन कोड के साथ उपयोग की जाती है (जब उपयोगकर्ता फॉर्म फ़िल्ड छोड़ता है)। यह इवेंट ऑनफोकस इवेंट के विपरीत है।

```
<html>
<head>
  <script type = "text/javascript">
    function valid() {
      alert("Don't leave it empty");
    }
  </script>
</head>
<body>
  <p> Write something in the input box and then click elsewhere in the
  document body </p>

  Enter your name <input type = "text" id = "fname" onblur="valid()">
</body>
</html>
```

पहला नाम दर्ज करें -

यदि टेक्स्ट बॉक्स में कुछ नहीं लिखा गया है , तो माउस के हटाने ही अलर्ट आ जायेगा।

8.11 ऑनलोड इवेंट - जब कोई HTML डॉक्यूमेंट लोड किया जाता है, तो ऑनलोड इवेंट होता है। वेब पेज पूरी तरह लोड होने के बाद स्क्रिप्ट को निष्पादित करने के लिए इस इवेंट का उपयोग <body> एलिमेंट के भीतर किया जाता है।

```
<html>
<head>
  <script type = "text/javascript">
    function fload() {
      alert('Page is loaded');
    }
  </script>
</head>
<body onload = "fload()">
  <h1> JavaScript</h1>
</body>
</html>
```

8.12 ऑनअनलोड इवेंट - ऑनअनलोड इवेंट तब होता है, जब कोई HTML दस्तावेज़ अनलोड किया जाता है। वेब पेज पूरी तरह से अनलोड होने के बाद स्क्रिप्ट को निष्पादित करने के लिए इस इवेंट का उपयोग <body> एलिमेंट के भीतर किया जाता है।

```
<html>
<head>
  <script type = "text/javascript">
    function uload() {
      alert('ThankYou for visiting this page');
    }
  </script>
</head>
<body onunload = "uload()">
  <p> JavaScript page</p>
</body>
</html>
```

9. पॉपअप बॉक्स

एक पॉप-अप एक ग्राफिकल यूजर इंटरफेस (जीयूआई) डिस्प्ले (क्षेत्र) है, आमतौर पर एक छोटी विंडो , जो विसुअल इंटरफ़ेस के फोरग्राउंड में अचानक दिखाई देती है, जिसे "पॉप अप" कहते हैं।

जावास्क्रिप्ट में तीन तरह के पॉपअप बॉक्स होते हैं: अलर्ट बॉक्स, कन्फर्म बॉक्स और प्रॉम्प्ट बॉक्स

9.1 अलर्ट बॉक्स

एक अलर्ट () विधि एक निर्दिष्ट संदेश (वैकल्पिक) और एक OK बटन के साथ एक अलर्ट बॉक्स प्रदर्शित करती है। अलर्ट के आगे जाने से पहले उपयोगकर्ताओं को संदेश की पुष्टि करने की आवश्यकता होती है । अलर्ट बॉक्स वर्तमान विंडो से ध्यान हटाता है और वर्तमान ब्राउज़र को संदेश पढ़ने के लिए मजबूर करता है।

```
<html>
<head>
  <script type = "text/javascript">
    function msg() {
      alert("Welcome to our site");
    }
  </script>

</head>
<body>
  <p>Alert Message</p>
  <button onclick = "msg()"> Press the Button </button>
</body>
</html>
```

9.2 कन्फर्म बॉक्स

कन्फर्म () विधि एक निर्दिष्ट संदेश के साथ एक "Ok" और एक "Cancel" बटन के साथ एक डायलॉग बॉक्स प्रदर्शित करता है। यदि उपयोगकर्ता "OK" पर क्लिक करता है तो कन्फर्म () विधि सही है और अन्यथा गलत है । उपयोगकर्ता द्वारा किसी भी चीज़ को सत्यापित करने या स्वीकार करने के लिए एक कन्फर्म बॉक्स का उपयोग अक्सर किया जाता है। कन्फर्म बॉक्स फोकस को वर्तमान विंडो से दूर ले जाता है और वर्तमान ब्राउज़र को संदेश पढ़ने के लिए मजबूर करता है।

```
<html>
<head>
  <script type = "text/javascript">
    function getmsg() {
      varretval = confirm("Do you want to continue ?");
      if ( retval == true){
        document.write("Continue with the process");
        return true; }
      else{
        document.write("Stop the process");
        return false; }
    }
  </script>
</head>
<body>
  <p>Confirmation Message</p>
  <input type ="button" value = "Press the Button" onclick = "msg()">
</body>
</html>
```

9.3 प्रॉम्प्ट बॉक्स

प्रॉम्प्ट () विधि एक डायलॉग बॉक्स प्रदर्शित करती है ,जो विजिटर को इनपुट के लिए प्रेरित करती है। इसका उपयोग तब किया जाता है, जब पेज में प्रवेश करने से पहले इनपुट के लिए एक वैल्यू आवश्यक होती हैं। जब एक प्रॉम्प्ट बॉक्स पॉप

अप होता है, तो उपयोगकर्ता को इनपुट वैल्यू दर्ज करने के बाद आगे बढ़ने के लिए या तो "OK" या "Cancel" पर क्लिक करना होता है। यदि उपयोगकर्ता "OK" पर क्लिक करता है, तो प्रॉम्प्ट () पद्धति इनपुट वैल्यू लौटाती है। यदि उपयोगकर्ता "Cancel" पर क्लिक करता है, तो विधि नल हो जाती है।

```
<html>
<head>
  <script type = "text/javascript">
    function getmsg() {
      varretval;
      var txt = prompt("Please enter your message :","Anil Dubey");
      if(txt ==null || txt == ""){
        retval = "Cancelled"; }
      else{
        retval = "hello" + txt +"!" }
      document.getElementById("demo"),innerHTML = retval;
    }
  </script>
</head>
<body>
  <p>Prompt Message</p>
  <input type ="button" value = "Press the Button" onclick = "getmsg()">
<p id = "demo"></p>
</body>
</html>
```

10. डॉक्यूमेंट ऑब्जेक्ट मॉडल

हर वेब पेज एक ब्राउज़र विंडो के अंदर रहता है, जिसे एक ऑब्जेक्ट माना जा सकता है। एक डॉक्यूमेंट ऑब्जेक्ट HTML डॉक्यूमेंट को दर्शाता है, जो उस विंडो में प्रदर्शित होता है। डॉक्यूमेंट ऑब्जेक्ट में विभिन्न गुण होते हैं, जो अन्य वस्तुओं को संदर्भित करते हैं, जो डॉक्यूमेंट सामग्री तक पहुँचने की और संशोधन करने की अनुमति देते हैं। जिस तरह से एक डॉक्यूमेंट सामग्री तक पहुँचा जाता है और संशोधित किया जाता है, उसे डॉक्यूमेंट ऑब्जेक्ट मॉडल या DOM कहा जाता है। DOM डॉक्यूमेंट्स तक पहुँचने के लिए एक स्टैंडर्ड को परिभाषित करता है।

"डब्लू 3 सी डॉक्यूमेंट ऑब्जेक्ट मॉडल (डीओएम) एक प्लेटफार्म और भाषा-तटस्थ (language-neutral) इंटरफ़ेस है, जो प्रोग्राम और स्क्रिप्ट को डॉक्यूमेंट की सामग्री, संरचना और शैली को गतिशील रूप से एक्सेस और अपडेट करने की अनुमति देता है।" W3C DOM स्टैंडर्ड को 3 अलग-अलग भागों में विभाजित किया गया है-

कोर डोम - सभी डॉक्यूमेंट प्रकारों के लिए स्टैंडर्ड मॉडल

एक्सएमएल डोम - एक्सएमएल डॉक्यूमेंट्स के लिए स्टैंडर्ड मॉडल

HTML DOM - HTML डॉक्यूमेंट्स के लिए स्टैंडर्ड मॉडल

ऑब्जेक्ट को एक पदानुक्रम में व्यवस्थित किया जाता है। यह पदानुक्रमित संरचना एक वेब डॉक्यूमेंट में वस्तुओं के संगठन पर लागू होती है। विंडो ऑब्जेक्ट - शीर्ष पदानुक्रम। यह ऑब्जेक्ट पदानुक्रम का सबसे बाहरी एलिमेंट है।

डॉक्यूमेंट ऑब्जेक्ट - प्रत्येक HTML डॉक्यूमेंट जो एक विंडो में लोड होता है, डॉक्यूमेंट ऑब्जेक्ट बन जाता है। डॉक्यूमेंट में पेज की सामग्री शामिल है।

फॉर्म ऑब्जेक्ट - <फॉर्म> ... </ फॉर्म> टैग्स में संलग्न सब कुछ फॉर्म ऑब्जेक्ट सेट करता है।

फॉर्म कंट्रोल एलिमेंट - फॉर्म ऑब्जेक्ट में उस ऑब्जेक्ट के लिए परिभाषित सभी तत्व शामिल होते हैं, जैसे कि टेक्स्ट फ़ील्ड, बटन, रेडियो बटन और चेकबॉक्स।

10.1 HTML DOM

HTML DOM, HTML के लिए एक स्टैंडर्ड ऑब्जेक्ट मॉडल और प्रोग्रामिंग इंटरफ़ेस है। यह परिभाषित करता है-

- HTML एलिमेंट के रूप में ऑब्जेक्ट ।
- सभी HTML एलिमेंट्स की प्रॉपर्टीज ।
- सभी HTML एलिमेंट्स को एक्सेस करने के तरीके।
- सभी HTML एलिमेंट्स के लिए ईवेंट।

- दूसरे शब्दों में: HTML DOM , HTML एलिमेंट्स को प्राप्त करने, बदलने, जोड़ने या हटाने के लिए एक स्टैण्डर्ड है।

10.2 DOM HTML ट्री

ऑब्जेक्ट मॉडल के साथ, जावास्क्रिप्ट को गतिशील HTML बनाने के लिए आवश्यक सभी शक्ति मिलती है:

- जावास्क्रिप्ट पेज में सभी HTML एलिमेंट्स को बदल सकता है।
- जावास्क्रिप्ट पेज में सभी HTML ऐट्रिब्यूट्स को बदल सकता है।
- जावास्क्रिप्ट पेज में सभी सीएसएस शैलियों को बदल सकते हैं।
- जावास्क्रिप्ट मौजूदा HTML एलिमेंट्स और ऐट्रिब्यूट्स को हटा सकता है।
- जावास्क्रिप्ट नए HTML एलिमेंट्स और ऐट्रिब्यूट्स को जोड़ सकता है।
- जावास्क्रिप्ट पेज की सभी मौजूदा HTML इवेंट्स पर प्रतिक्रिया कर सकता है।
- जावास्क्रिप्ट पेज में नई HTML ईवेंट बना सकता है।

अस्तित्व में कई DOM हैं। निम्नलिखित अनुभाग इन DOMs में से प्रत्येक को विस्तार से वर्णन करते हैं और बताते हैं कि आप डॉक्यूमेंट कंटेंट तक पहुंचने और संशोधित करने के लिए उनका उपयोग कैसे कर सकते हैं।

10.2.1 द लिगेसी डोम - यह वह मॉडल है, जिसे जावास्क्रिप्ट भाषा के शुरुआती संस्करणों में पेश किया गया था। यह सभी ब्राउज़रों द्वारा अच्छी तरह से समर्थित है, लेकिन डॉक्यूमेंट्स के कुछ प्रमुख भागों, जैसे कि फॉर्म, फॉर्म एलिमेंट्स, और इमेज तक ही पहुँच प्रदान करता है।

10.2.2 W3C DOM - यह डॉक्यूमेंट ऑब्जेक्ट मॉडल सभी डॉक्यूमेंट कंटेंट के एक्सेस और संशोधन की अनुमति देता है और वर्ल्ड वाइड वेब कंसोर्टियम (W3C) द्वारा मानकीकृत है। यह मॉडल लगभग सभी आधुनिक ब्राउज़रों द्वारा समर्थित है।

10.2.3 E4 DOM - यह डॉक्यूमेंट ऑब्जेक्ट मॉडल Microsoft के इंटरनेट एक्सप्लोरर ब्राउज़र के वर्जन 4 में पेश किया गया था। IE 5 और बाद के संस्करणों में अधिकांश बुनियादी W3C DOM सुविधाओं के लिए समर्थन शामिल है।

10.3 डोम कम्पेटिबिलिटी

यदि आप अपनी उपलब्धता के आधार पर W3C DOM या IE 4 DOM का उपयोग करने के लचीलेपन के साथ एक स्क्रिप्ट लिखना चाहते हैं, तो आप कैपबिलिटी -टेस्टिंग एप्रोच का उपयोग कर सकते हैं, जो पहले यह निर्धारित करने के लिए कि विधि या प्रॉपर्टी के अस्तित्व के लिए जाँच करता है, कि क्या ब्राउज़र में क्षमता है, जो आप चाहते हैं। उदाहरण के लिए -

```
if (document.getElementById) {  
    // If the W3C method exists, use it  
} else if (document.all) {  
    // If the all[] array exists, use it  
} else {  
    // Otherwise use the legacy DOM  
}
```