

If value of count is above 1, then its value is incremented by 1 otherwise it is set to 0. It is also possible to combine multiple ternary operators to make a complex conditional logic. Normal if else statement can be used in place of ternary operator.

यदि काउंट की वैल्यू 1 से ऊपर है, तो उसका वैल्यू 1 बढ़ा दिया जाता है अन्यथा इसे 0 के रूप में सेट किया जाता है। कॉम्प्लेक्स कंडीशनल लाजिक बनाने के लिए कई टर्नरी ऑपरेटरों को संयोजित करना भी संभव है। सामान्य if else statement का उपयोग टर्नरी ऑपरेटर के स्थान पर किया जा सकता है।

6. typeof Operator: JavaScript type of operator returns a string which indicates the type of an operand. typeof operator always returns string value. JavaScript typeof operator returns six possible values from "number", "string", "boolean," "object", "function" and "undefined."

टाइपऑफ ऑपरेटर : जावास्क्रिप्ट टाइपऑफ ऑपरेटर एक स्ट्रिंग देता है जो एक ऑपरेंड के प्रकार को इंगित करता है। टाइपऑफ ऑपरेटर हमेशा स्ट्रिंग वैल्यू लौटाता है। जावास्क्रिप्ट टाइपऑफ ऑपरेटर छह संभावित वैल्यू "नंबर", "स्ट्रिंग", "बूलियन", "ऑब्जेक्ट", "फंक्शन", और "अनडिफाइंड" रिटर्न करता है।

typeof operator is mostly used with condition statements to check the type of variable before going further in execution to stop any exceptions.

टाइपऑफ ऑपरेटर का ज्यादातर प्रयोग कंडीशनल स्टेटमेंट के साथ किसी भी अपवाद को रोकने एवं वैरियेबल के प्रकार की जाँच के लिए किया जाता है।

It would be a good practice to check for type of variable is not undefined before doing any mathematical operation, otherwise it produces unexpected results.

किसी भी गणितीय कार्य को करने से पहले वैरियेबल के प्रकार की जांच करना एक अच्छा अभ्यास होगा, अन्यथा यह अप्रत्याशित परिणाम देता है।

Syntax: typeof (operand)

Below table shows the possible return values of typeof operator.

नीचे दी गई तालिका टाइपऑफ ऑपरेटर के संभावित रिटर्न वैल्यूज को दिखाती है।

Type	Result
Undefined	"undefined"
Null	"object" (see below)
Boolean	"Boolean"
Number	"number"
String	"string"
Function object	"function"
Any other object	"object"

See the example given here: नीचे दिए गए उदाहरण को देखें-

h1 h2 h3 p b i u span link img ul li
PREVIEW

```

1 <!DOCTYPE html>
2 <html>
3 <body>
4 Default Value is not set for locationName variable
5 <p id="demo"></p>
6 <script>
7 var locationName;
8 if(typeof locationName == 'undefined') {
9   alert("Variable locationName is undefined");
10 }
11 </script>
12 </body>
13 </html>
```

Default Value is not set for locationName variable

An embedded page on this page says

Variable locationName is undefined

Open your browser and check it. Immediately browser shows one alert box (in upcoming section will discuss about alert box or popup box). After this assign some value to the variable locationName="Sita rama" and save the file and refresh your browser.

अपना ब्राउजर खोलें और इसे जांचें। तुरंत ब्राउजर एक अलर्ट बॉक्स दिखाता है (आगामी अनुभाग में अलर्ट बॉक्स या पॉपअप बॉक्स के बारे में चर्चा करेंगे)। इसके बाद वैरिएबल लोकेशन पर कुछ वैल्यू असाइन करें = "Sita rama" और फाइल को सेव करें और अपने ब्राउजर को रिफ्रेश करें।

6.4 Conditions Statements in Java Script - जावा स्क्रिप्ट में कंडीशनल आपरेटर

Conditions are the expressions which are used to evaluate statements as per the condition. Conditions are the comparison statements to execute the statements condition wise and provide at least two options to execute the statements means there will be two options in between one has to be executed, if the condition is true then execute this otherwise execute that. We place different operators and conditional statements to establish condition in the program to execute the program statements.

कंडीशन एक बहुत ही खास घटक है जो कंडीशनल ब्रांचिंग एवं लूपिंग में प्रयोग होता है। कंडीशन का मतलब साधारण भाषा में शर्त होता है जो ब्रांचिंग एवं लूपिंग में कण्ट्रोल को मोनिटर करने के लिए प्रयोग किया जाता है। कंडीशन में किन्हीं दो ऑपरेंड के बीच ज्यादातर रिलेशनल ऑपरेटर का प्रयोग कर कंडीशन बनाते हैं जिसके सही या गलत होने की स्थिति में कण्ट्रोल का ब्रांचिंग एवं लूपिंग निर्भर करता है।

Here we see...

2<3

At some point in programming it is necessary to perform special actions based on different conditions to complete the decision. So that conditional statements are available in JavaScript.

प्रोग्रामिंग में कुछ बिंदु पर अक्सर निर्णय पूरा करने के लिए विभिन्न कंडीशंस के आधार पर विशेष क्रियाएं करना आवश्यक होता है। ताकि जावास्क्रिप्ट में कंडीशनल आपरेटर उपलब्ध हों।

- if.. statement
- if..else statement
- if..else if.. statement
- switch statement

If statement: The if statement is the fundamental control statement that allows JavaScript to make decisions and execute statements conditionally. The syntax for a basic if statement is as follows -

if statement एक फंडामेंटल कंट्रोल स्टेटमेंट है, जो जावास्क्रिप्ट को निर्णय लेने और कंडीशनल स्टेटमेंट एक्सजीक्यूट करने की अनुमति देता है। बेसिक इफ स्टेटमेंट के लिए सिन्टेक्स इस प्रकार है-

Syntax:

```
if (expression) {
    Statement(s);
}
```

Here a JavaScript expression is evaluated. If the condition is true, the given statement(s) are executed otherwise statement block would be not executed. Most of the times, you will use comparison operators while making decisions. You can understand how if statement works.

यहां एक जावास्क्रिप्ट एक्सप्रेशन का मूल्यांकन किया जाता है। यदि कंडीशन ट्रू है, तो दिए गए statement(s) को क्रियान्वित किया जाता है अन्यथा statement ब्लॉक क्रियान्वित (एक्सजीक्यूट) नहीं किया जाएगा। कई बार, आप निर्णय लेते समय कम्परीजन ऑपरेटरों का उपयोग करेंगे। if statement कैसे काम करता है, निम्नलिखित उदाहरण के द्वारा आप समझ सकते हैं।

```

1 <html>
2 <body>
3 <h2> Enter Age </h2> <br>
4 <input id="age"><br><br>
5 <button onclick="myFun()">Check my age</button>
6 <br><br>
7 <script type = "text/javascript">
8 function myFun()
9 {
10     var age = document.getElementById('age').value;
11     if( age > 18 ){
12         document.write("<b>Qualifies for driving</b>");
13     }
14 }
15 </script>
16 <p>Set the variable to different value and then try...</p>
17 </body>
18 </html>
    
```

PREVIEW

Enter Age

Check my age

Set the variable to different value and then try...

Enter the different values and then try. विभिन्न वैल्यू दर्ज करें और फिर प्रयास करें।

if-else statement: It is another most used conditional statement in if groups what provides two ways access for the statements to be executed. The **if** statement by itself will execute a single statement, or a group of statements, when the expression following **if** evaluates to true. It does nothing when the expression evaluates to false. Can we execute one group of statements if the expression evaluates to true and another group of statements if the expression evaluates to false?

if-else स्टेटमेंट सबसे ज्यादा प्रयोग होने वाला कंडीशनल स्टेटमेंट है जिसमें एक इफ स्टेटमेंट होता है और इसके पैरेलल एल्स स्टेटमेंट होता है जिसकी मदद से आप कंडीशन के टू होने की स्थिति में या फलसे होने दोनों की स्थिति में कम से कम एक स्टेटमेंट का एग्जीक्यूट होना अनिवार्य है।

```

if (expression)
    statement;

else
    statement;
    
```

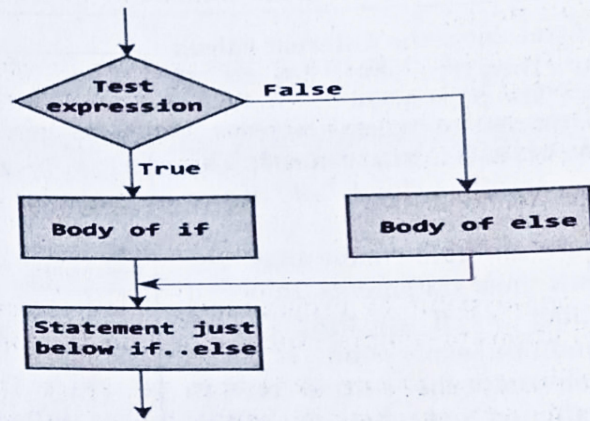


Figure: Flowchart of if...else Statement

It is the general form or syntax of if-else statement where the expression in the parenthesis of **if** is evaluated first and then executed the statement as per the condition means if the condition is true then the statements within block if is executed otherwise the statements of else block is executed.

Syntax:

```

If (boolean_expression)
{
    statement (s);
} else
{
    statement (s);
}
    
```

Example:

```

1 <!DOCTYPE html>
2 <html>
3 <body>
4 <h3>Enter Student Marks</h3>
5 <input id="marks1"><br><br>
6 <button onClick="checkResult()">Check</button>
7 <br><br>
8 <h1><span id="text"></span></h1>
9 <script>
10 function checkResult()
11 {
12   var sMarks = document.getElementById('marks1').value
13   if(sMarks >= 35)
14   {
15     document.getElementById('text').innerHTML = 'Pass';
16   } else {
17     document.getElementById('text').innerHTML = 'Fail';
18   }
19 }
20 </script></body></html>

```

output

Enter Student Marks

50

Check

Pass

output

Enter Student Marks

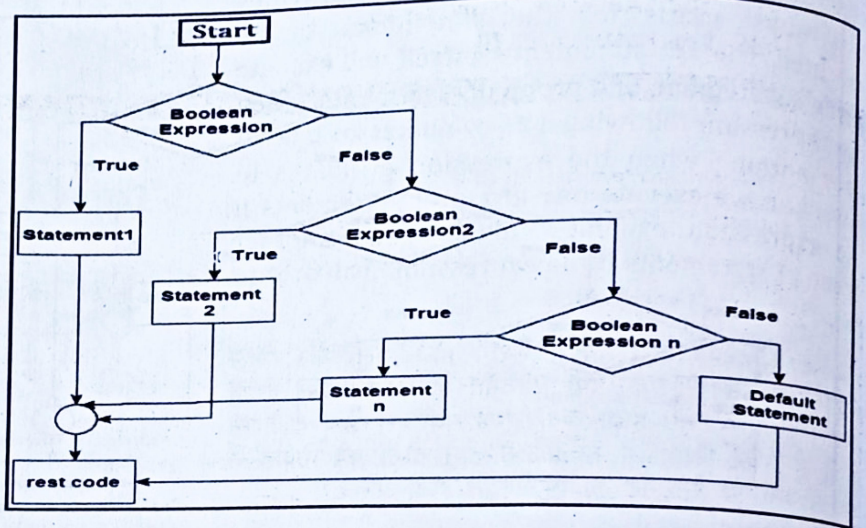
30

Check

Fail

Note: Enter the different values and then try. - विभिन्न वैल्यू दर्ज करें और फिर प्रयास करें।

if...else if... statement: This is another kind of if-else statement which is used to place multiple conditions along with their statements. In else-if ladder if the first condition/expression is false then only the second one is evaluated and when any of the condition found true then block of statements is executed associated with and the control terminates the else-if structure and moves ahead. In else-if ladder we can place multiple conditions.



if...else if... कंडीशनल स्टेटमेंट के अंतर्गत एक अन्य भाग है जिसमें एक if स्टेटमेंट होता है और बाकि सभी else स्टेटमेंट के अंतर्गत if स्टेटमेंट होते हैं। इस कंडीशनल स्टेटमेंट में यदि if स्टेटमेंट के अंतर्गत कंडीशन अगर ट्रू (सही) होती है तो उसका ब्लॉक ऑफ कोड एग्जीक्यूट होता है अन्यथा else के अंतर्गत वाले if स्टेटमेंट के कंडीशन की जांच होती है और सही पाए जाने पर उसका ब्लॉक ऑफ कोड एग्जीक्यूट होता है अन्यथा अगला else-if कंडीशन का सत्यापन होता है।

The if...else if... statement is an advanced form of if...else that allows JavaScript to make a correct decision out of several conditions.

Syntax:

```

if (expression 1)
    Statement (s) to be executed if expression 1 is true;
else if(expression 2)
    Statement(s) to be executed if expression 2 is true;
else if(expression 3)
    Statement(s) to be executed if expression 3 is true;
else
    Statement(s) to be executed if no expression is true;

```

Here we saw the syntax of else-if ladder where the first condition is placed within if parenthesis when found true then statement is executed associated with and when found false then only another condition is evaluated.

It is just a series of if statements, where each if is a part of the else clause of the previous statement. Statement(s) are executed based on the true condition, if none of the conditions is true, then the else block is executed.

यह सिर्फ स्टेटमेंट्स की एक श्रृंखला है, जहां प्रत्येक if पिछले स्टेटमेंट के दूसरे खंड का एक हिस्सा है। Statement(s) को सही स्थिति के आधार पर क्रियान्वित किया जाता है, यदि कोई भी स्थिति सही नहीं है, तो अन्य ब्लॉक क्रियान्वित किया जाता है।

Example: Write a program in javascript and enter valid marks (conditions apply)?
जावास्क्रिप्ट में एक प्रोग्राम लिखें, कुछ वैलिड मार्क्स एंटर करें (कंडीशन भी अप्लाई करें)।

Use the following condition:

- >=80 then Result = "First Grade"
- >=60 then Result = "Second Grade"
- >=35 then Result = "Third Grade"
- <35 then Result = "Dubkee"

```

1 <!DOCTYPE html>
2 <html>
3 <body>
4 <h3>Enter Student Marks</h3>
5 <input id="m"><br><br>
6 <button onClick="checkResult()">Check</button>
7 <br><br>
8 <h1><span id="text"></span></h1>
9 <script>
10 function checkResult()
11 {
12     var marks = document.getElementById('m').value
13     if(marks >= 80) {
14         document.getElementById('text').innerHTML = '1st Grade';
15     } else if(marks >= 60) {
16         document.getElementById('text').innerHTML = '2nd Grade';
17     } else if(marks >= 35) {
18         document.getElementById('text').innerHTML = '3rd Grade';
19     } else {
20         document.getElementById('text').innerHTML = 'Dubkee';
21     }
22 }
23 </script></body></html>
                
```

PREVIEW

Enter Student Marks

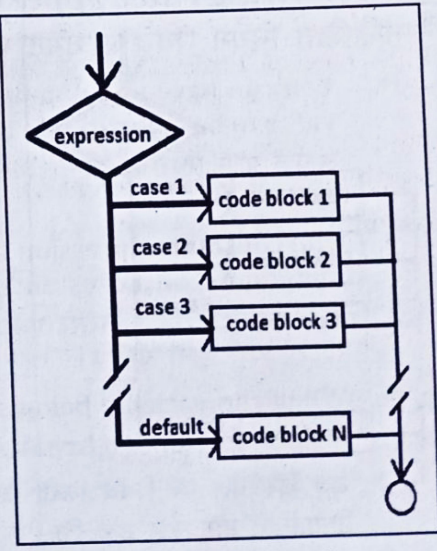
Dubkee

Run the program using different values and test your result. अलग-अलग वैल्यूज को एंटर करके ऊपर दिए गए प्रोग्राम को चलाएं और अपने परिणाम का परीक्षण करें।

Switch Statement - स्विच स्टेटमेंट

A **switch statement** allows a variable to be tested for equality against a list of values. Each value is called a case, and the variable being switched on is checked for each switch case. Instead of using the if-else-if ladder, the switch statement is available in JavaScript.

एक स्विच स्टेटमेंट एक वैरिएबल को वैल्यू की सूची के विपरीत समानता के लिए परीक्षण करने की अनुमति देता है। प्रत्येक वैल्यू को एक केस कहा जाता है, और स्विच किए जा रहे वैरिएबल को प्रत्येक स्विच केस के लिए चेक किया जाता है। if-else-if लैडर का उपयोग करने के बजाय, स्विच स्टेटमेंट जावास्क्रिप्ट में उपलब्ध है।



You can use multiple if...else...if statements, as in the previous section, to perform a multi way branch. However, this is not always the best solution, especially when all of the branches depend on the value of a single variable. So that we need a best solution i.e **switch case statement**.

आप मल्टीपल ब्रांच के लिए if...else...if statements, पिछले सेक्शन में इस्तेमाल कर सकते हैं। हालांकि, यह हमेशा सबसे अच्छा समाधान नहीं होता है, खासकर जब सभी ब्रांच एक एकल वैरियेबल के वैल्यू पर निर्भर करती है। एक बेहतरीन समाधान की आवश्यकता हेतु **switch case statement** का प्रयोग करें।

The syntax for a switch statement as follows: एक स्विच स्टेटमेंट के लिए सिंटैक्स निम्नानुसार है:

Syntax:

```
switch(expression)
{
    case constant-expression :
        statement(s);
        break; /* optional */
    case constant-expression :
        statement(s);
        break; /* optional */
    /* you can have any number of case statements */
    default : /* Optional */
        statement(s);
}
```

This is how it works: -

यह इस तरह कार्य करता है-

- The switch expression is evaluated once. The value of the expression is compared with the values of each case.
स्विच एक्सप्रेशन का मूल्यांकन एक बार किया जाता है। प्रत्येक केश के वैल्यू के साथ एक्सप्रेशन के वैल्यू की तुलना की जाती है।
- If there is a match, the associated block of code is executed.
यदि कोई मैच होता है, तो कोड का संबंधित ब्लॉक क्रियान्वित किया जाता है।

The following rules apply to a switch statement:

निम्नलिखित नियम स्विच स्टेटमेंट पर लागू होते हैं

- You can have any number of case statements within a switch. Each case is followed by the value to be compared to and a colon.
आपके पास स्विच के अन्दर किसी भी प्रकार के केस स्टेटमेंट हो सकते हैं। प्रत्येक मामले की वैल्यू एक कोलन की तुलना में होता है।
- The constant-expression for a case must be the same data type as the variable in the switch, and it must be a constant or a literal.
एक मामले के लिए निरंतर एक्सप्रेशन स्विच में वैरियेबल के समान डेटा प्रकार होना चाहिए, और यह एक निरंतर या शाब्दिक होना चाहिए।
- When the variable being switched on is equal to a case, the statements following that case will execute until a **break** statement is reached.
जब वैरियेबल पर स्विच किया जा रहा है यह एक मामले के बराबर है, तो उस मामले के बाद के बयान तब तक क्रियान्वित होंगे जब तक कि एक ब्रेक स्टेटमेंट नहीं आता है।

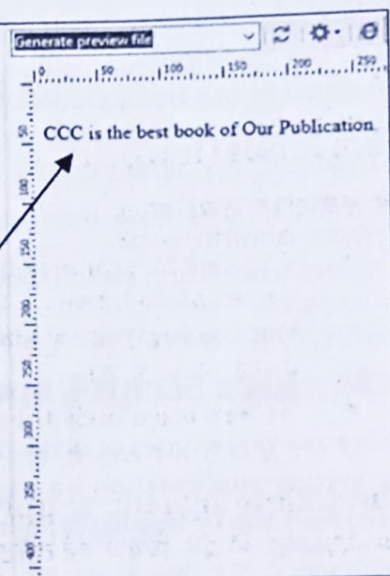
- When a **break** statement is reached, the switch terminates, and the flow of control jumps to the next line following the switch statement.
जब एक ब्रेक स्टेटमेंट आ जाता है, तो स्विच समाप्त हो जाता है, और नियंत्रण का प्रवाह स्विच स्टेटमेंट के बाद अगली पंक्ति में जाता है।
- Not every case needs to contain a **break**. If no **break** appears, the flow of control will fall through to subsequent cases until a **break** is reached.
हर केश में ब्रेक की आवश्यकता नहीं होती है। यदि कोई ब्रेक दिखाई नहीं देता है, तो नियंत्रण का प्रवाह बाद के मामलों तक गिर जाएगा जब तक कि एक ब्रेक नहीं मिल जाता है।
- A switch statement can have an optional **default** case, which must appear at the end of the **switch**. The **default** case can be used for performing a task when none of the cases is **true**. No **break** is needed in the **default** case.
एक स्विच स्टेटमेंट में एक वैकल्पिक डिफॉल्ट केस हो सकता है, जिसे स्विच के अंत में प्रदर्शित होना चाहिए। डिफॉल्ट केस का उपयोग किसी कार्य के निष्पादन के लिए किया जा सकता है जब कोई भी केश सत्य नहीं होता है। डिफॉल्ट मामले में किसी भी ब्रेक की आवश्यकता नहीं है।

Example: The example of the switch statement is given below:

```

1 <!DOCTYPE html>
2 <html>
3 <body>
4 <p id="text">Our Books</p>
5 <script>
6 document.getElementById('text').innerHTML = '<br >' +getFruitColor('CCC');
7 function getFruitColor(BOOKS) {
8     var msg;
9     switch(BOOKS) {
10        case 'CCC':
11            msg = 'CCC is the best book of Our Publication.';
12            break;
13        case 'HINDI':
14            msg = 'HINDI is our upcoming book.';
15            break;
16        case 'BEO':
17            msg = 'BEO is the latest book of our publication.';
18            break;
19        default:
20            msg = 'Books is not available';
21    }
22    return msg;
23 }
24 </script></body></html>

```



Output of the above code is: CCC is the best book of Our Publication.

Note: Change the value and try again. मान बदलें और पुनः प्रयास करें।

When a JavaScript found a **break** statement, the switch statement execution stops from where it reached and returns next statement after switch. When a match is found from available cases, there is no need to check further cases for matches. It is not necessary to have a code block for every cases in switch statement. In such cases more than one switch cases can have same code blocks.

जब जावास्क्रिप्ट ब्रेक स्टेटमेंट पाता है, तो स्विच स्टेटमेंट क्रियान्वयन बंद हो जाता है, जहां से यह पहुंचा और स्विच के बाद अगला स्टेटमेंट देता है। जब उपलब्ध मामलों में से एक मिलान पाया जाता है, तो मिलान के लिए आगे के मामलों की जांच करने की कोई आवश्यकता नहीं है। स्विच स्टेटमेंट में प्रत्येक मामले के लिए एक कोड ब्लॉक होना आवश्यक नहीं है। ऐसे केशों में एक से अधिक स्विच स्टेटमेंट में समान कोड ब्लॉक हो सकते हैं।

Lets us see one example without break statement.

आइए हम बिना ब्रेक स्टेटमेंट के एक उदाहरण देखें।

```

1 <html>
2 <body>
3 <script type = "text/javascript">
4 <!--
5     var group = 'A';
6     switch (group) {
7         case 'A': document.write("CCC is a good book. <br />");
8         case 'B': document.write("It is good to use balaji publication books.<br />");
9         case 'C': document.write("OK you can use it for better result.<br />");
10        case 'D': document.write("Do not use any other book for CCC.<br />");
11        case 'F': document.write("You are Failed.<br />");
12        default: document.write("Choose another Book.<br />")
13    }
14    document.write("You are now exiting switch block");
15    //-->
16 </script>
17 <p>Set the variable to different value and then try again...</p>
18 </body>
19 </html>

```

Generate preview file

0 50 100 150 200 250

CCC is a good book.
It is good to use balaji publication books.
OK you can use it for better result.
Do not use any other book for CCC.
You are Failed.
Choose another Book.
You are now exiting switch block

Set the variable to different value and then try again...

Output of the code is here

Output: आउटपुट:

CCC is a good book.
It is good to use balaji publication books.
OK you can use it for better result.
Do not use any other book for CCC.
You are Failed.
Choose another Book.
You are now exiting switch block

Set the variable to different value and then try again...

6.5 Java Script Popup Boxes - जावा स्क्रिप्ट पॉपअप बॉक्स

There are three types of dialog box in JavaScript. These dialog boxes are used for raising and alerting, for getting confirmation on any input or having a kind of input from the users. Each dialog box is discussed one by one in the following.

जावास्क्रिप्ट में तीन तरह के डायलॉग बॉक्स होते हैं। इन डायलॉग बॉक्स का उपयोग किसी भी इनपुट पर पुष्टि प्राप्त करने और उपयोगकर्ताओं से एक प्रकार का इनपुट प्राप्त करने के लिए उपयोग और अलर्ट करने के लिए किया जाता है। प्रत्येक डायलॉग बॉक्स की चर्चा नीचे एक-एक करके करते हैं।

1. Alert Dialog Box: अलर्ट डायलॉग बॉक्स

Alert dialog box is usually used for giving a warning message to the users. For example if one input field requires entering some text but the user does not provide any input, then as a part of validation, you can use an alert box for giving a warning message. And on the other hand, the alert box can be used for friendlier messages. Alert box consists of only one "Ok" button to select and proceed.

एक अलर्ट डायलॉग बॉक्स का उपयोग ज्यादातर उपयोगकर्ताओं को वार्निंग मैसेज देने के लिए किया जाता है। उदाहरण के लिए, यदि किसी इनपुट फील्ड में कुछ टेक्स्ट दर्ज करने की आवश्यकता होती है, लेकिन उपयोगकर्ता कोई इनपुट प्रदान नहीं करता है, तो सत्यापन के एक भाग के रूप में, आप वार्निंग मैसेज देने के लिए अलर्ट बॉक्स का उपयोग कर सकते हैं। दूसरी

ओर, फ्रेंडलिएर मैसेजेस के लिए अभी भी एक अलर्ट बॉक्स का उपयोग किया जा सकता है। अलर्ट बॉक्स चयन और आगे बढ़ने के लिए केवल एक "ओके" बटन देता है।

Syntax: window.alert("Give Some Information");

Note: The window.alert() method can be written without the window prefix.

```

1 <!DOCTYPE html>
2 <html>
3   <body>
4     <h2>Example of Alert Box</h2>
5     <button onclick="myFun()">Click Me</button>
6     <script>
7       function myFun()
8       {
9         alert("Hello, I am an alert box!");
10      }
11    </script>
12  </body>
13 </html>
14

```

2. Confirmation Dialog Box - कंफर्मेशन डायलॉग बॉक्स

The confirmation dialog box is mostly used for taking user's approval for any option. It consists a dialog box with two buttons: 'Ok and Cancel'. If the user clicks on the Ok button, the window method **confirm()** will return true, and if the user clicks on the Cancel button, then **confirm()** returns false. The confirmation dialog box can be used as follows.

किसी भी विकल्प पर उपयोगकर्ता की स्वीकृति लेने के लिए आमतौर पर एक कंफर्मेशन डायलॉग बॉक्स का उपयोग किया जाता है। यह दो बटन के साथ एक डायलॉग बॉक्स प्रदर्शित करता है: पहला 'ओके' और दूसरा 'कैन्सल'। यदि उपयोगकर्ता ओके बटन पर क्लिक करता है, तो विंडो मेथड **कंफर्म()** ट्रू रिटर्न करेगी। यदि उपयोगकर्ता कैन्सल बटन पर क्लिक करता है, तो **कंफर्म()** फाल्स रिटर्न करेगा। आप निम्नानुसार एक कंफर्मेशन डायलॉग बॉक्स का उपयोग कर सकते हैं।

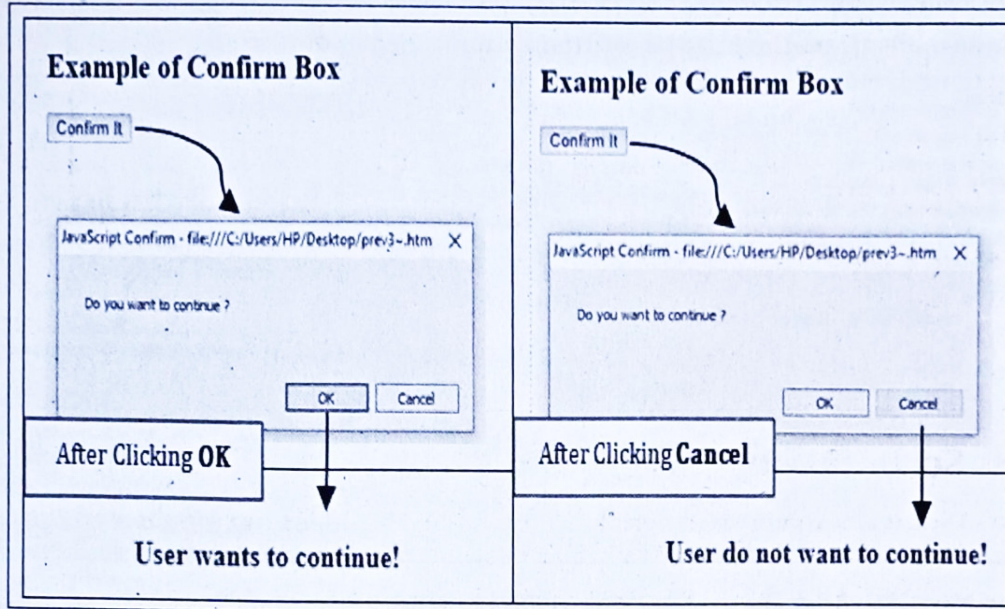
Run the following code and see the result:

```

1 <!DOCTYPE html>
2 <html>
3   <body>
4     <h2>Example of Confirm Box</h2>
5     <button onclick="myConfirmation()">Confirm It</button>
6     <br><br><br>
7     <h4><p id="try"></p></h4>
8     <script>
9       function myConfirmation()
10      {
11        var txt;
12        if(confirm("Do you want to continue ?"))
13          txt="User wants to continue!";
14        else
15          txt="User do not want to continue!";
16
17        document.getElementById("try").innerHTML= txt;
18      }
19    </script>
20  </body>
21 </html>

```

Output of the above code after clicking on "Confirm It" button: 'कन्फर्म इट' बटन पर क्लिक करने के बाद नीचे दिए गए आउटपुट आपको दीखेंगे—



3. Prompt Dialog Box - प्रॉम्प्ट डायलॉग बॉक्स:

It is a useful dialog box for you to pop-up a text box to get user input and also enables to interact with user. The user needs to fill in the field and then click **OK**. This dialog box is displayed using a method called **prompt()** which takes two parameters.

यह टेक्स्ट को पाप-अप करने के लिए एक उपयोगी डायलाग बॉक्स है जो यूजर द्वारा इनपुट लेता है एवं यूजर को इंटरैक्ट भी कराता है। उपयोगकर्ता को फील्ड को भरना होगा और फिर ओके पर क्लिक करना होगा। यह डायलॉग बॉक्स प्रॉम्प्ट() नामक मेथड का उपयोग करके प्रदर्शित किया जाता है जो दो मापदंडों (पैरामीटर) को एक्सेप्ट करता है।

The parameters are as follows: पैरामीटर नीचे दिए गए है—

- A label which you want to display in the text box and - एक लेबल जिसे आप टेक्स्ट बॉक्स में दिखाना चाहते हैं, और
- A default string to display in the text box. टेक्स्ट बॉक्स में प्रदर्शित करने के लिए एक डिफॉल्ट स्ट्रिंग।

This dialog box has two buttons, **OK** and **Cancel**. If the user clicks the **OK button**, the window method **prompt()** will return the **entered value** from the text box. If the user clicks the **Cancel button**, the window method **prompt()** returns **null**.

इस डायलॉग बॉक्स में दो बटन हैं, **OK** और **Cancel**। यदि उपयोगकर्ता ओके बटन पर क्लिक करता है, तो विंडो विधि **prompt()** टेक्स्ट बॉक्स से दर्ज वैल्यू वापस कर देगा। यदि उपयोगकर्ता **Cancel** बटन पर क्लिक करता है, तो विंडो विधि **prompt()** वापस आ जाता है।