Day 5 assignment

Q)

Ans:-

Decision Making in programming is similar to decision making in real life. In programming also we face some situations where we want a certain block of code to be executed when some condition is fulfilled.  
A programming language uses control statements to control the flow of execution of the program based on certain conditions. These are used to cause the flow of execution to advance and branch based on changes to the state of a program.

JavaScript’s conditional statements:

* [if](https://www.geeksforgeeks.org/if-else-statement-in-javascript/#if)
* [if-else](https://www.geeksforgeeks.org/if-else-statement-in-javascript/#if-else)
* [nested-if](https://www.geeksforgeeks.org/if-else-statement-in-javascript/#nested-if)
* [if-else-if](https://www.geeksforgeeks.org/if-else-statement-in-javascript/#if-else-if)

These statements allow you to control the flow of your program’s execution based upon conditions known only during run time.

* if: if statement is the most simple decision making statement. It is used to decide whether a certain statement or block of statements will be executed or not i.e if a certain condition is true then a block of statement is executed otherwise not.  
  **Syntax**:
* if(condition)
* {
* // Statements to execute if
* // condition is true
* }

Here, **condition** after evaluation will be either true or false. if statement accepts boolean values – if the value is true then it will execute the block of statements under it.  
If we do not provide the curly braces ‘{‘ and ‘}’ after **if( condition )** then by default if statement will consider the immediate one statement to be inside its block. For example,

if(condition)

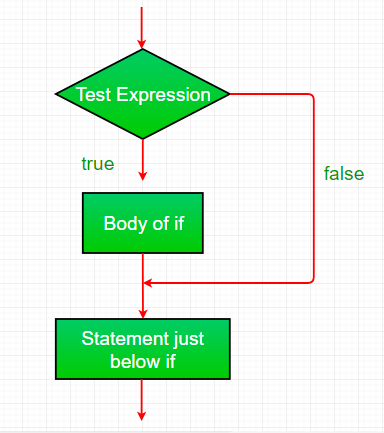
statement1;

statement2;

// Here if the condition is true, if block

// will consider only statement1 to be inside

// its block.

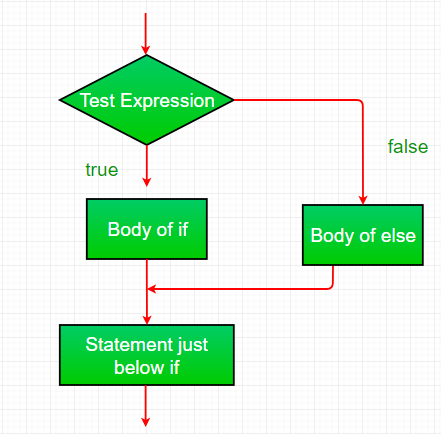
Flow chart:  
  
Example:

|  |
| --- |
| <script type = "text/javaScript">    // JavaScript program to illustrate If statement    **var** i = 10;    **if** (i > 15)    document.write("10 is less than 15");    // This statement will be executed  // as if considers one statement by default  document.write("I am Not in if");    < /script> |

Output:

I am Not in if

* if-else: The if statement alone tells us that if a condition is true it will execute a block of statements and if the condition is false it won’t. But what if we want to do something else if the condition is false. Here comes the else statement. We can use the else statement with if statement to execute a block of code when the condition is false.  
  **Syntax**:
* if (condition)
* {
* // Executes this block if
* // condition is true
* }
* else
* {
* // Executes this block if
* // condition is false
* }

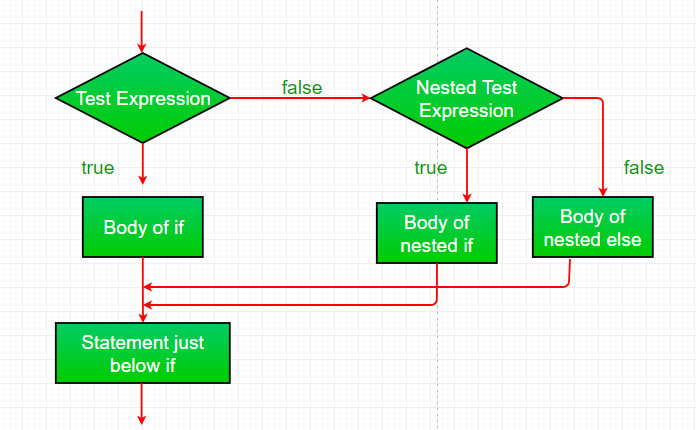
  
Example:

|  |
| --- |
| <script type = "text/javaScript">    // JavaScript program to illustrate If-else statement    **var** i = 10;    **if** (i < 15)    document.write("10 is less than 15");  **else**    document.write("I am Not in if");    < /script> |

Output:

i is smaller than 15

* nested-if: A nested if is an if statement that is the target of another if or else. Nested if statements means an if statement inside an if statement. Yes, JavaScript allows us to nest if statements within if statements. i.e, we can place an if statement inside another if statement.  
  Syntax:
* if (condition1)
* {
* // Executes when condition1 is true
* if (condition2)
* {
* // Executes when condition2 is true
* }
* }



Example:

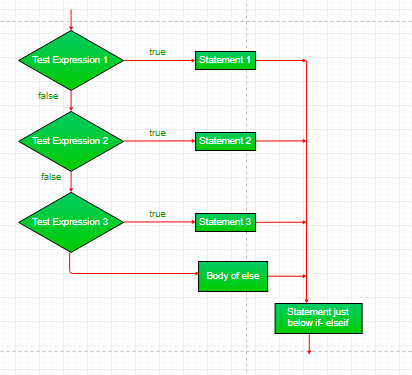
|  |
| --- |
| <script type = "text/javaScript">    // JavaScript program to illustrate nested-if statement    **var** i = 10;    **if** (i == 10) {      // First if statement  **if** (i < 15)      document.write("i is smaller than 15");      // Nested - if statement    // Will only be executed if statement above    // it is true  **if** (i < 12)      document.write("i is smaller than 12 too");  **else**      document.write("i is greater than 15");  }  < /script> |

Output:

i is smaller than 15

i is smaller than 12 too

* if-else-if ladder: Here, a user can decide among multiple options.The if statements are executed from the top down. As soon as one of the conditions controlling the if is true, the statement associated with that if is executed, and the rest of the ladder is bypassed. If none of the conditions is true, then the final else statement will be executed.
* if (condition)
* statement;
* else if (condition)
* statement;
* .
* .
* else
* statement;

  
Example:

|  |
| --- |
| <script type = "text/javaScript">  // JavaScript program to illustrate nested-if statement    **var** i = 20;    **if** (i == 10)    document.write("i is 10");  **else** **if** (i == 15)    document.write("i is 15");  **else** **if** (i == 20)    document.write("i is 20");  **else**    document.write("i is not present");  < /script> |

Output:

i is 20

We use cookies to ensure you have the best browsing experience on our website. By using our site, you acknowledge that you have read and understood our [Cookie Policy](https://www.geeksforgeeks.org/cookie-policy/) & [Privacy Policy](https://www.geeksforgeeks.org/privacy-policy/)Got It !