**Javascript**

**Introduction to Javascript**

* JavaScript is a programming language, which is used to create functionality in the web page.

Functionality means, “receiving inputs from the user and providing output to the user”.

* It can perform tasks such as calculations, decision making, repetitive tasks, dynamically displaying the output, reading inputs from the user dynamically, updating content on the web page based on the inputs, interacting with server, validations etc.
* It’s operators and control statements are similar to “C” language.
* JavaScript is client-side (browser-side) language. That means it executes on the browser. It can also be used in server by using NodeJS.
* JavaScript is a case sensitive language.
* JavaScript is “interpreter-based” language. That means the code will be converted into machine language, line-by-line.
* JavaScript was developed by “Netscape Corporation” in 1996.
* JavaScript is the implementation of "EcmaScript". "EcmaScript" is the specification of "JavaScript".
* "EcmaScript" is designed by "Ecma International".

**Syntax of Javascript**

<script>

Javascript code>

</script>

**Note:** You can write the <script> tag either in <head> tag or <body> tag also; however, writing

<script> tag at the end of <body> tag is a best practice.

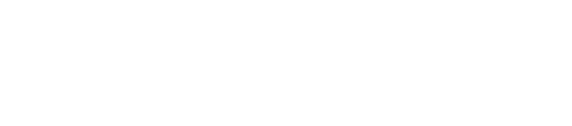
* The type="text/javascript" attribute specifies that you are using javascript language; It is optional.

**console.log()**

* The console.log() statement is used to display value in the browser console.
* To see console, first run the program in the browser and press "F12" or right click and choose "Inspect Element" – "Console" option in the browser.

**Syntax:** console.log(value);

* Variable is a “named memory location” in RAM, to store a value temporarily, while executing the



program.

* In JavaScript, the variables will be persisted (stored), while the web page is running in the browser.
* The value of variable can be changed any no. of times during the web page execution.
* The data type of the variable can be changed any no. of times during the web page execution, in JavaScript.

# Steps for development of variables

* Declare (create) the variable - optional: var variablename;
* Set value into the variable: variablename = value;
* Get value from the variable: variablename
* Operator is a symbol, which represents an operation.



* JavaScript supports the following types of operators.
  1. Arithmetical Operators
  2. Assignment Operators
  3. Increment and Decrement Operators
  4. Relational Operators
  5. Logical Operators
  6. Concatenation Operator

**Arithmetical Operators**

+ Addition

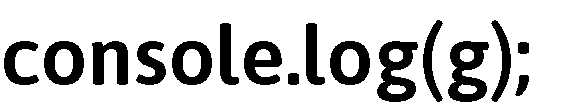
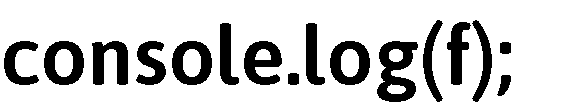
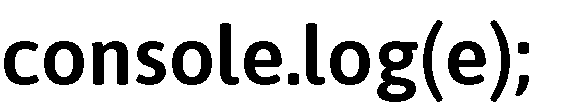
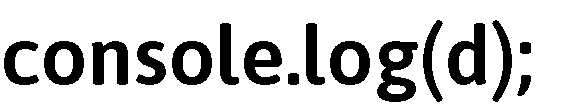
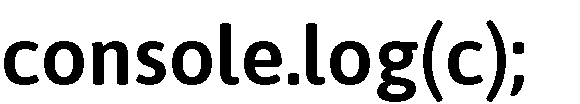
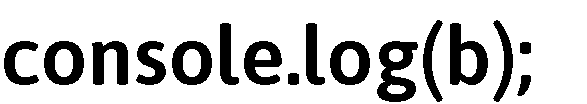
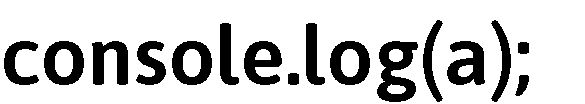
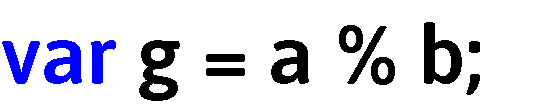
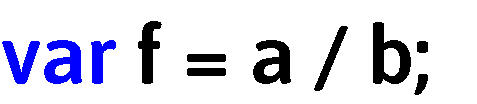
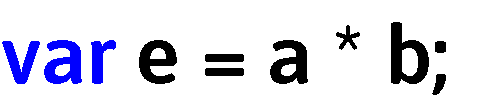
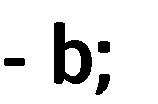
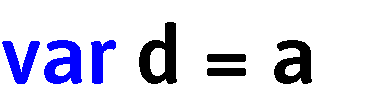
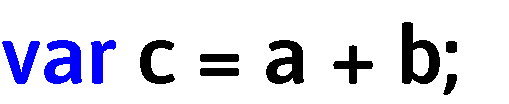
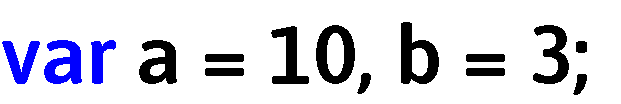
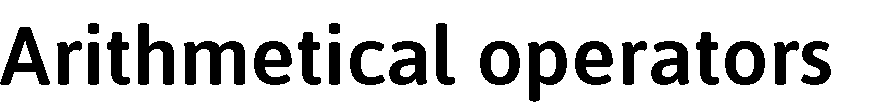
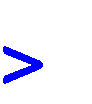
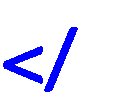
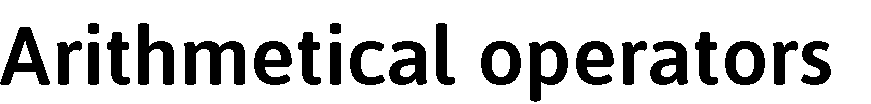
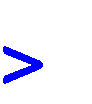
- Subtraction

\* Multiplication

/ Division

% Remainder

# Example on Arithmetical Operators



**Assignment Operators**

= Assigns to

+= Add and assigns to

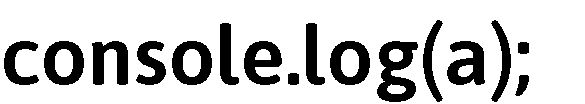
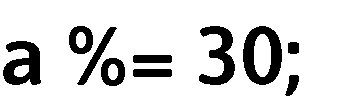
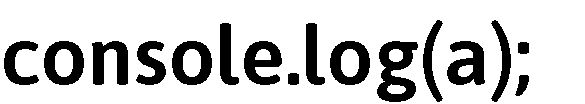
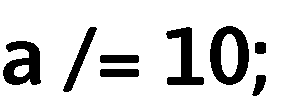
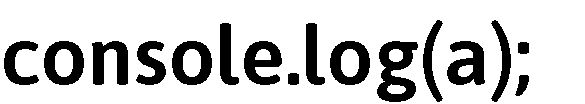
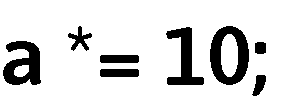
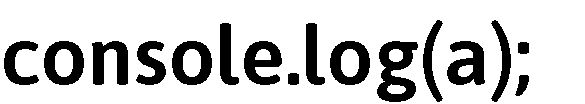
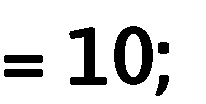
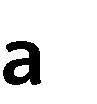
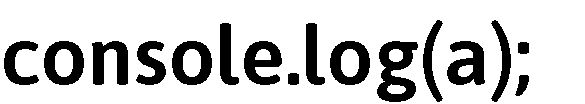
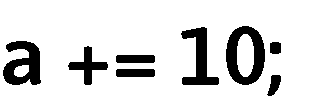
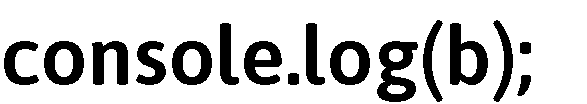
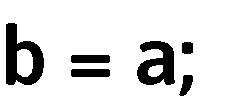
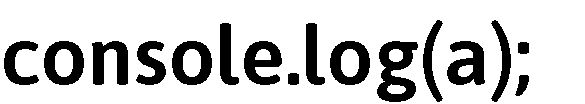
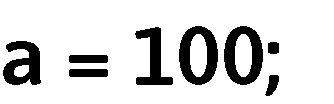
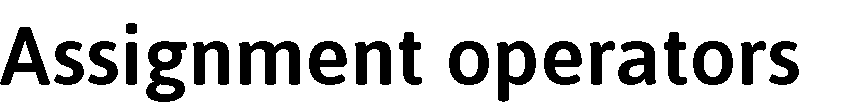
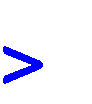
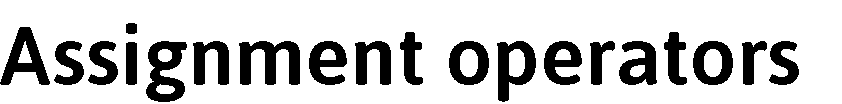
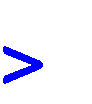
-= Subtract and assigns to

\*= Multiply and assigns to

/= Divide and assigns to

%= Remainder and assigns to

# Example on Assignment Operators

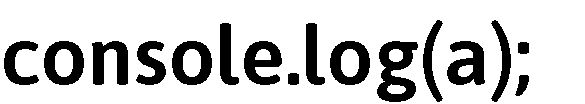
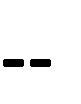
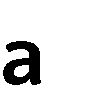
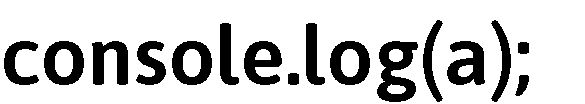
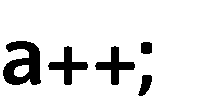
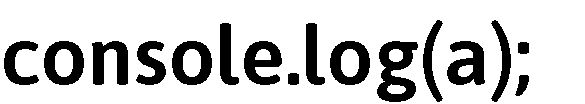
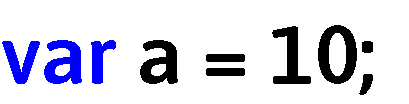
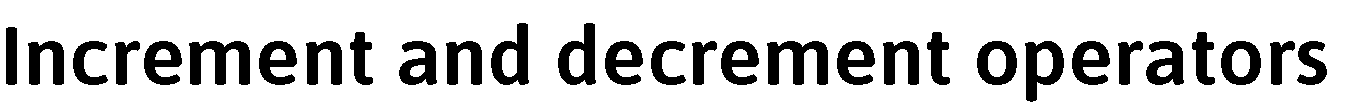
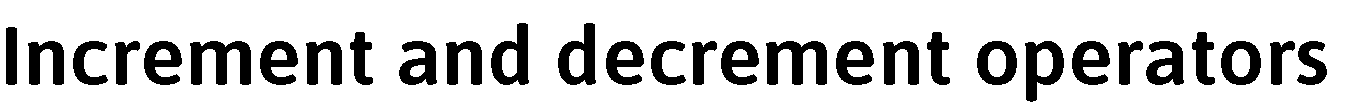


**Increment and Decrement Operators**

++ Increment (+=1)

-- Decrement (-=1)

# Example on Increment and Decrement Operators



**Relational Operators**

== Equal to

!= Not Equal to

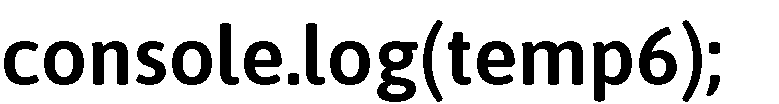
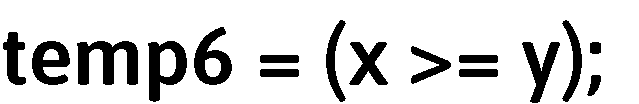
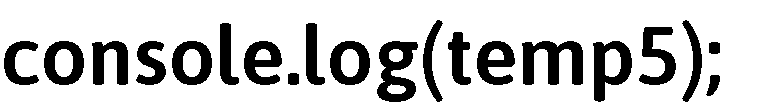
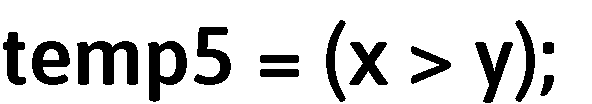
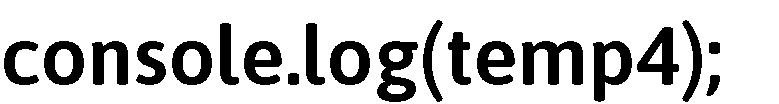
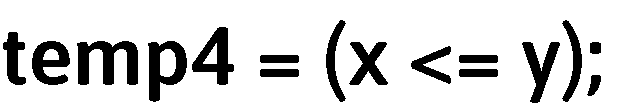
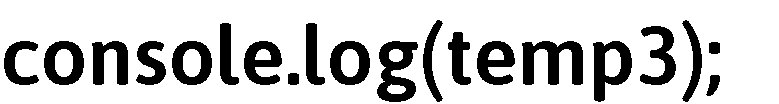
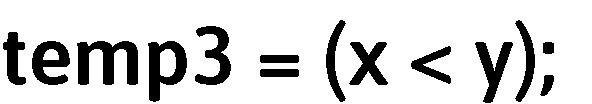
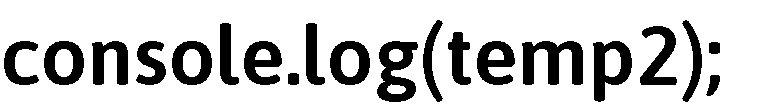
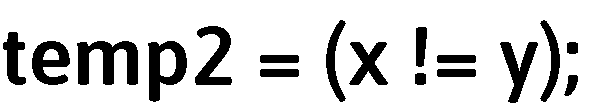
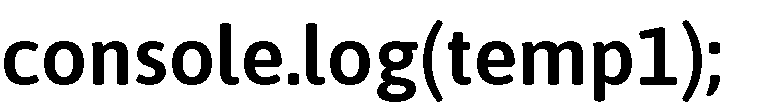
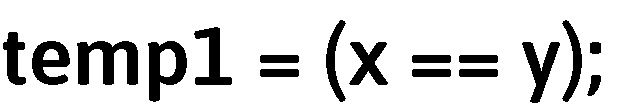
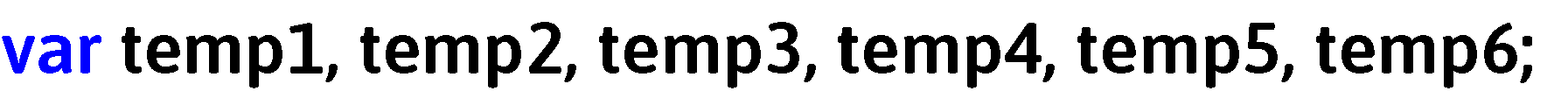
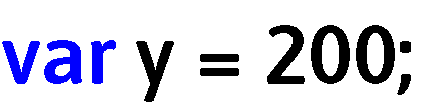
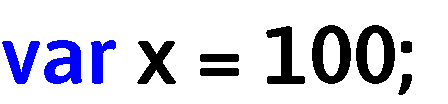
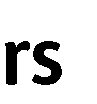
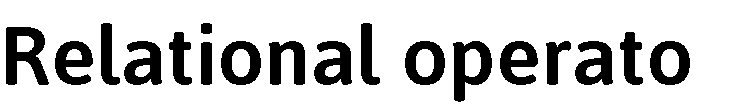
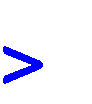
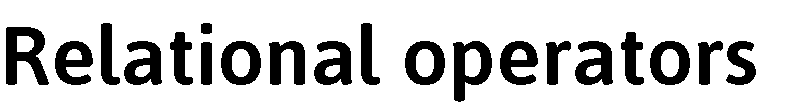
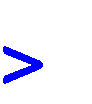
< Less than

> Greater than

<= Less than or equal to

>= Greater than or equal to

# Example on Relational Operators



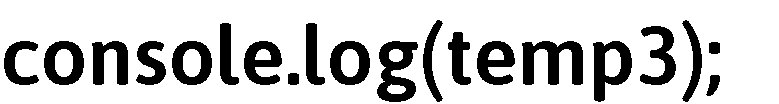
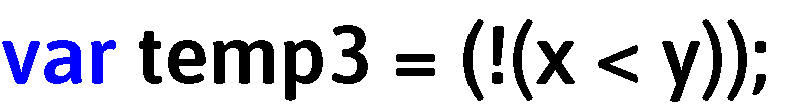
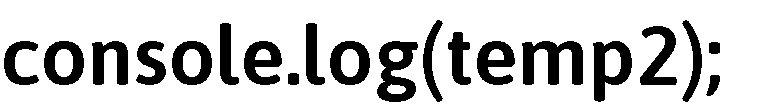
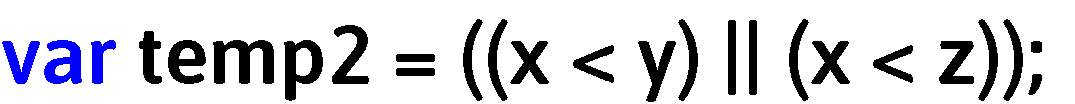
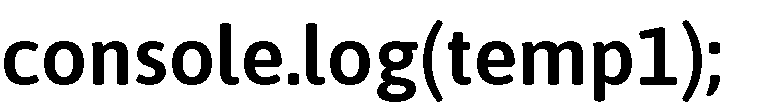
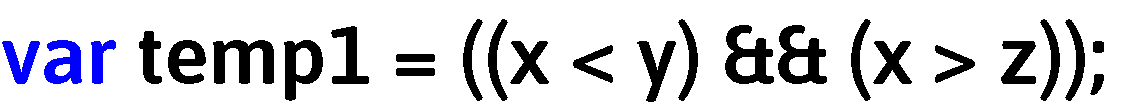
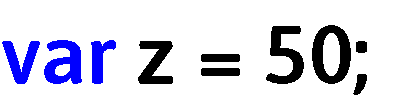
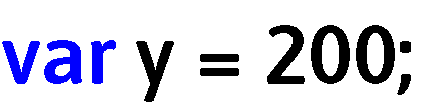
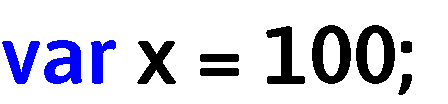
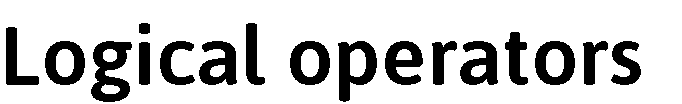
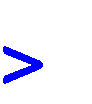
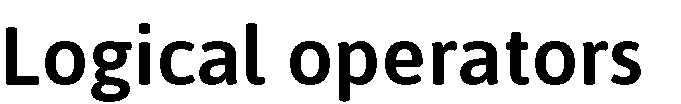
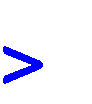
**Logical Operators**

&& And (both conditions should be true)

|| Or (At least any one condition should be true)

! Not (given condition will be reverse)

# Example on Logical Operators



**Concatenation Operator**

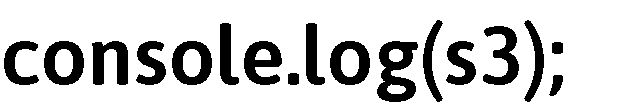
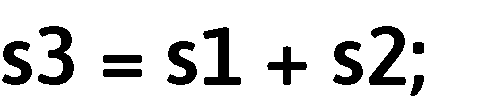
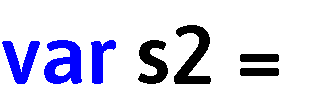
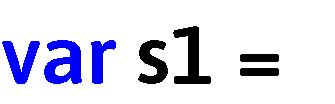
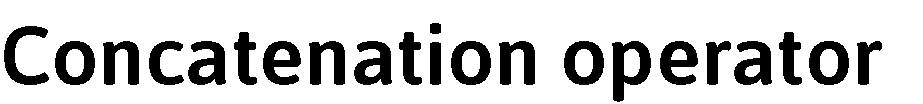
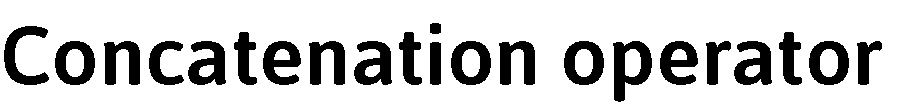
+ Attaches two strings and returns a single string.

Ex: “new” + “delhi” = “newdelhi”

Number + Number = addition String + String = concatenation String + Number = concatenation

Number + String = concatenation

# Example on Concatenation Operator



# Concatenation Operator

* Control statements are used to control (change) the program execution flow.



* These are used to make the execution flow jump forward / jump backward.
* JavaScript supports two types of control statements:
  + 1. Conditional Control Statements: Used to jump forward.
    2. Looping Control Statements: Used to jump backward.

# Conditional Control Statements

* + If
  + Switch-case

# Looping Control Statements

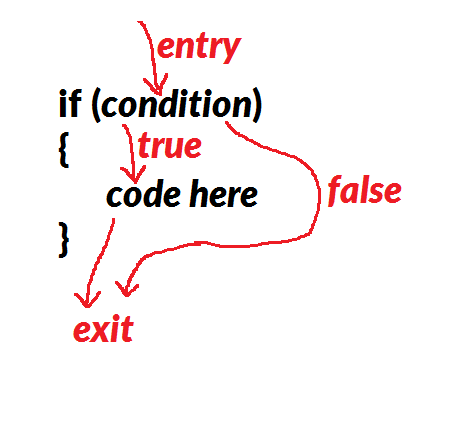
* + While
  + Do-while
  + For

**if**

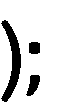
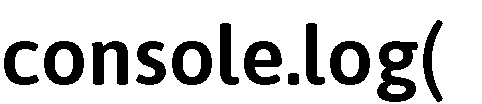
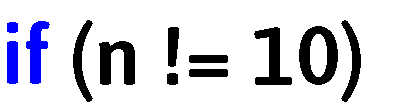
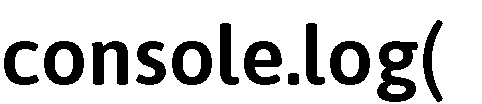
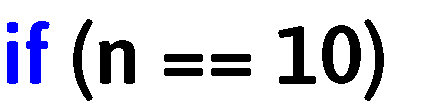
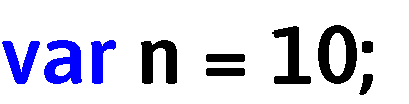
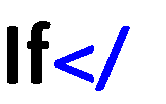
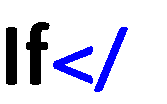
* “If” statement is used to check a condition, and execute the code only if the condition is TRUE.
* Types of “if”

1. If
2. If-else
3. Else-if
4. Nested if

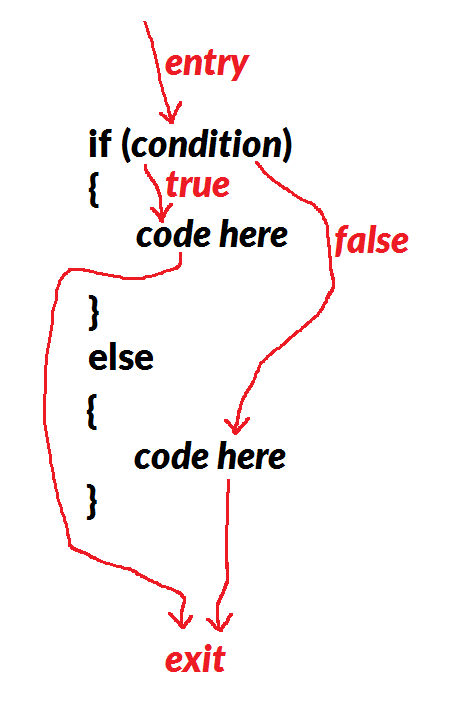
# Simple If



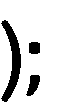
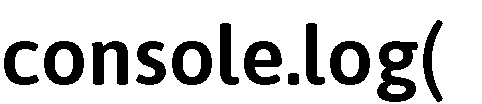
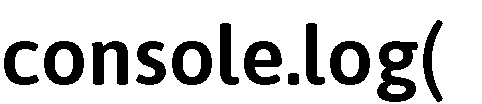
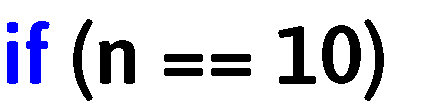
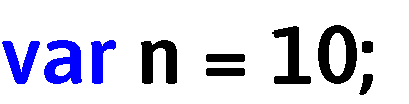
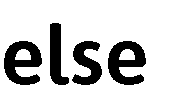
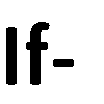
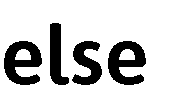
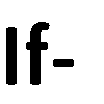
**Example of “Simple if”**



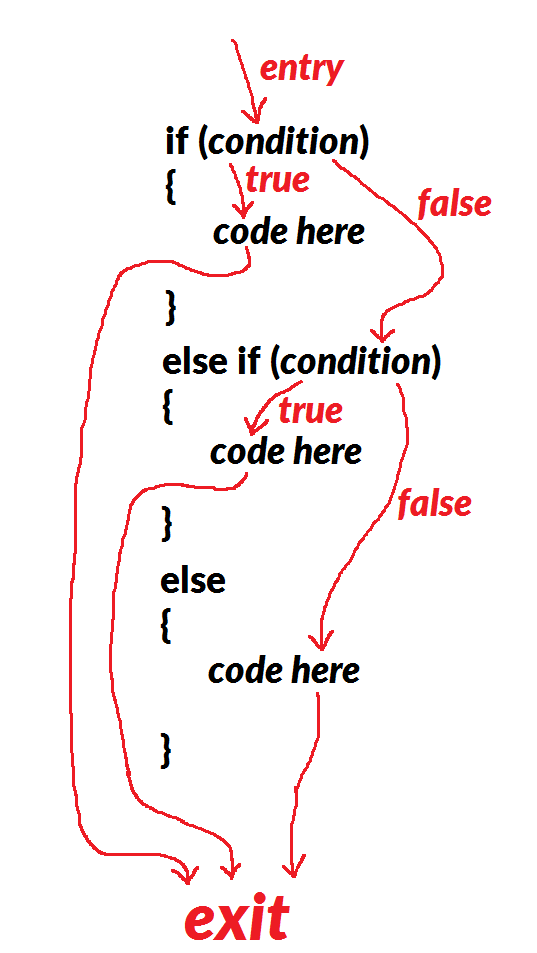
# If Else



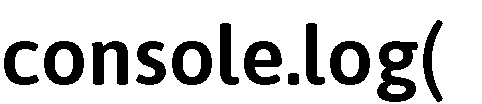
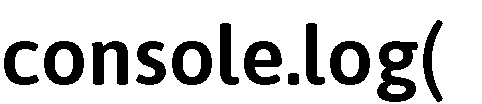
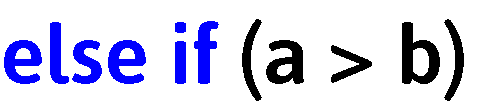
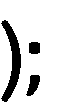
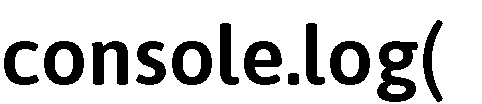
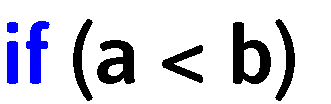
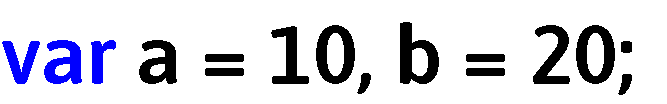
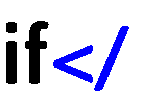
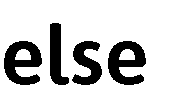
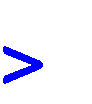
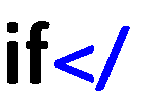
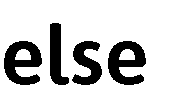
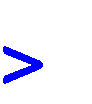
**Example of “if-else”**



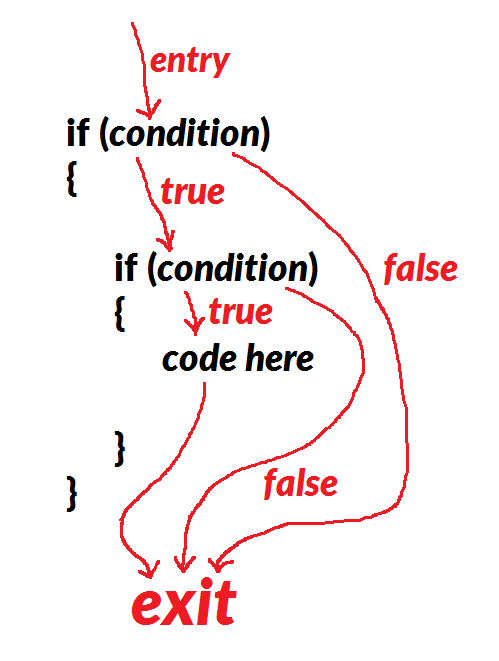
# Else if



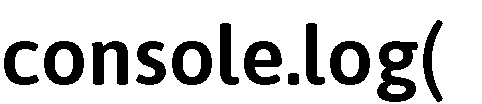
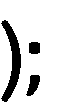
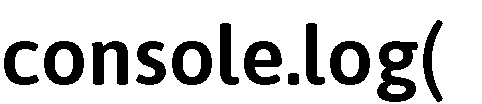
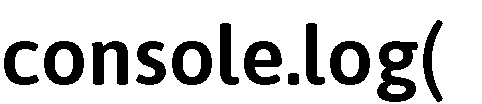
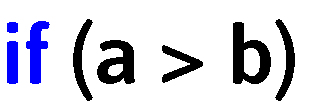
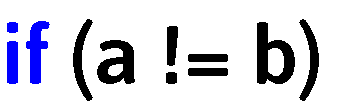
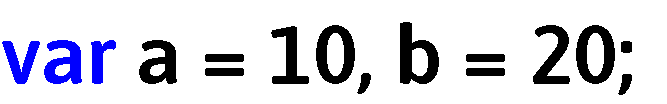
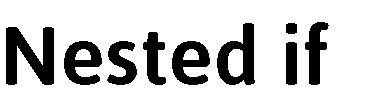
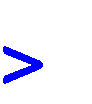
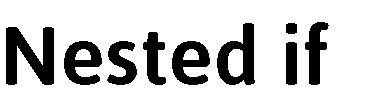
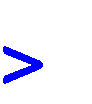
**Example of “else-if”**



# Nested If



**Example of “nested if”**

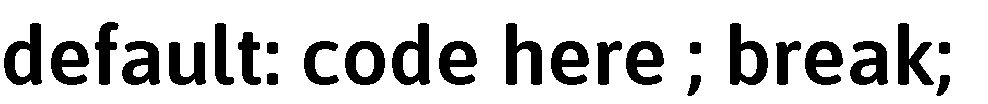
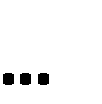
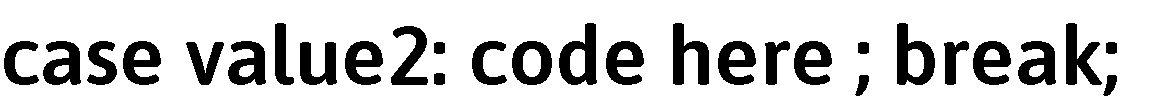
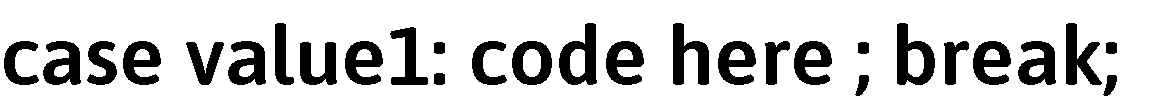
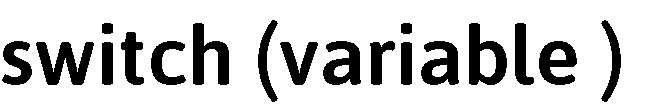


**Switch-case**

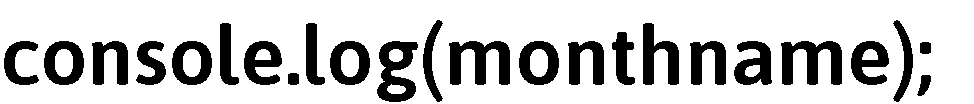
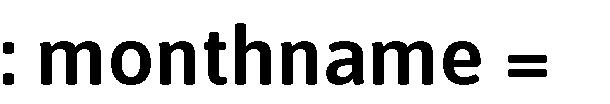
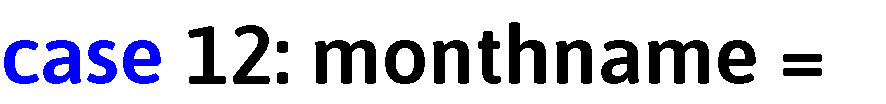
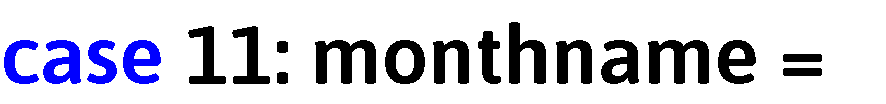
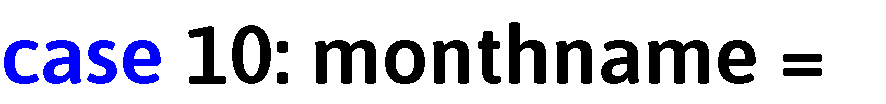
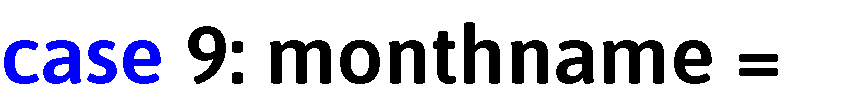
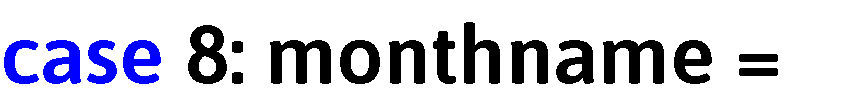
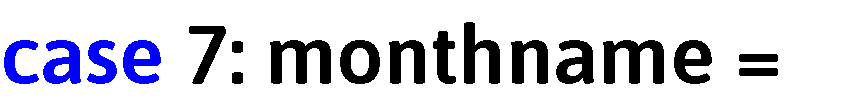
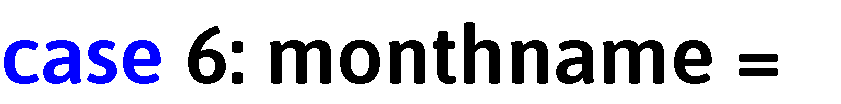
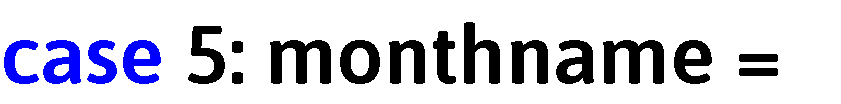
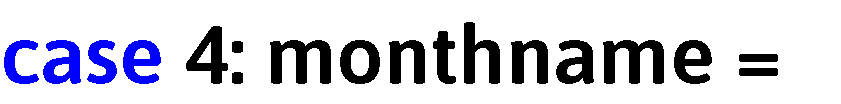
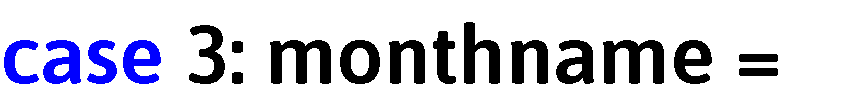
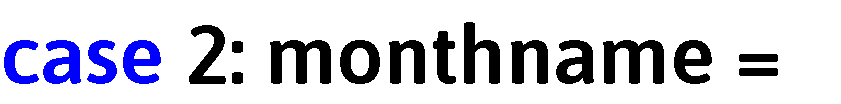
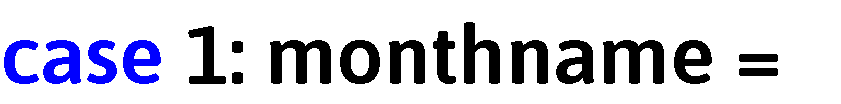
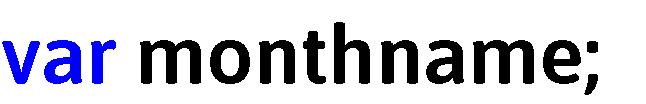
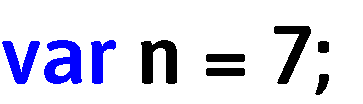
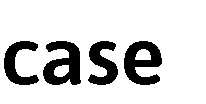
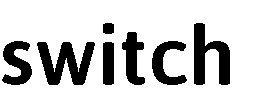
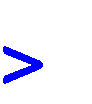
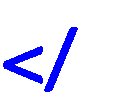
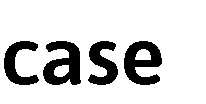
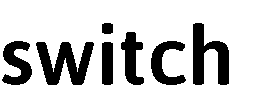
* It is used to check a variable’s value, whether it matches with any one of the set of cases, and execute

the code of the matched case.

* **Syntax:**

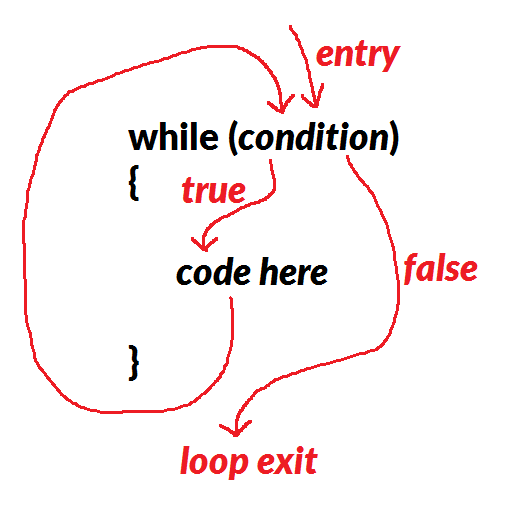


**Example on Switch-case**

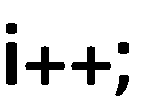
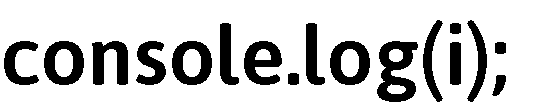
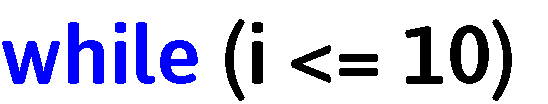
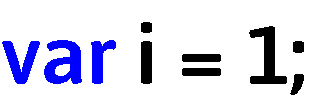
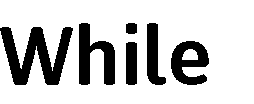
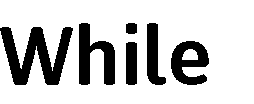


**While**

* “While” statement is used to execute the code repeatedly, while the condition is TRUE.

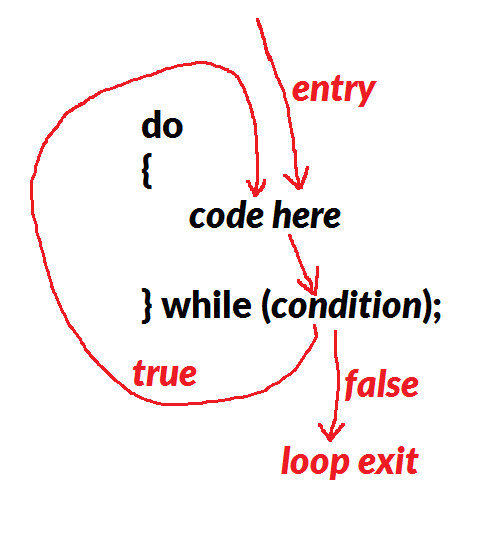


# Example on While

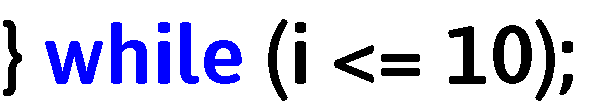
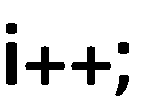
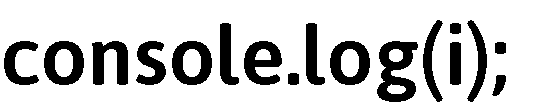
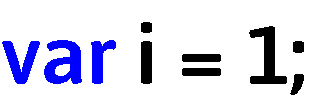
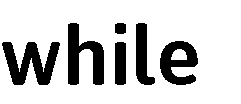
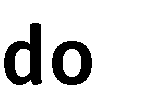
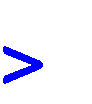
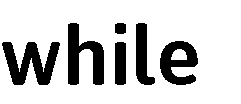
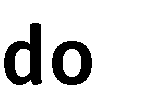
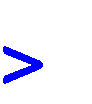


**Do-While**

* “Do-while” loop is mostly same as “while” loop.
* The difference is: “while” loop checks the condition for the first time also; but “do-while” loop doesn’t check the condition for the first time.

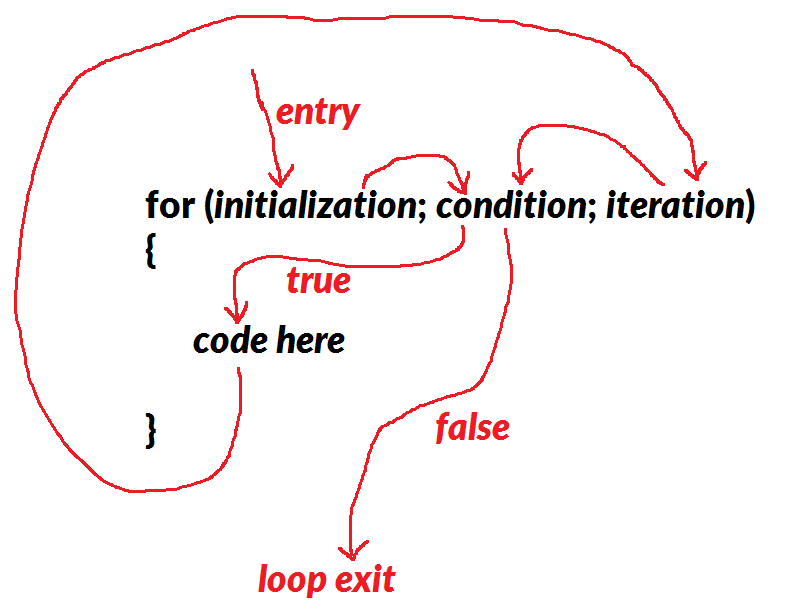


# Example on Do-While

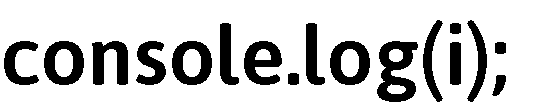
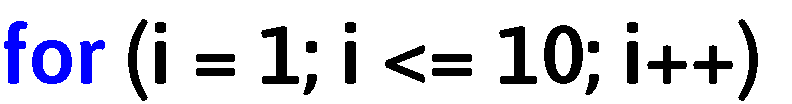
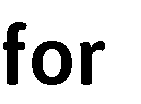
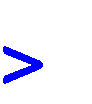
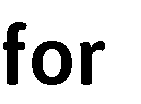
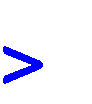


**For**

* “For” loop is mostly same as “while” loop.
* The difference is: “While” loop has the initialization, condition and iteration in three different places, so that it will be difficult to understand, if the code increases. But “for” loop has the initialization, condition and iteration in the same line, so that it will be easy to understand.



# Example on For



# Simple if

# a=10

# b=5

# if a>b

# console.log(a)

# if..else

# a=5

# b=5

# if a>b

# console.log(a)

# else

# console.log(b)

# if..else…if

# a=5

# b=5

# if a>b

# console.log(a)

# else

# if b>a

# console.log(b)

# else

# console.log(“both are equal”)

# if..else…if

# a=5

# b=5

# if a==b

# console.log(“Both are equal”)

# else

# if a>b

# console.log(“a is bigger”)

# else

# console.log(“b is bigger”)

# Program to find even or odd number?

# n=5

# if n%2==0

# console.log(“Even number”);

# else

# console.log(“Odd number”);

# Program to find positive or negative or zero number?

# n=5

# if n>0

# console.log(“positive”)

# else

# if n<0

# console.log(“Negative number”)

# else

# console.log(“Zero number”)

# perform arithmetic operations on 2 numbers as per user’s choice? (Menu type program)

# console.log(“1.Addition”)

# console.log(“2.Subtraction”)

# console.log(“3.Multiplication”)

# console.log(“4.Division ”)

# a=10

# b=5

# ch=4

# if ch==1

# console.log(“Addition is”,a+b);

# else

# if ch==2

# console.log(“Subtraction is ”,a-b);

# else

# if ch==3

# console.log(“Multiplication is”,a\*b);

# else

# if ch==4

# console.log(“Division is”,a/b);

# else

# console.log(“Invalid choice”);

# perform arithmetic operations on 2 numbers by an arithmetic operator? (Menu type program)

# console.log(“+ Addition”)

# console.log(“- Subtraction”)

# console.log(“\* Multiplication”)

# console.log(“/ Division ”)

# a=10

# b=5

# opt=’/’

# if opt==’+’

# console.log(“Addition is”,a+b);

# else

# if opt==’-‘

# console.log(“Subtraction is ”,a-b);

# else

# if opt==’\*’

# console.log(“Multiplication is”,a\*b);

# else

# if opt==’/’

# console.log(“Division is”,a/b);

# else

# console.log(“Invalid arithmetic operator”);

# switch..case

# it executes only a part of the program which matches or satisfies the given condition

# perform arithmetic operations on 2 numbers as per user’s choice? (Menu type program)

# console.log(“1.Addition”)

# console.log(“2.Subtraction”)

# console.log(“3.Multiplication”)

# console.log(“4.Division ”)

# a=10

# b=5

# ch=4

# switch(ch)

# {

# case 1:

# console.log(“Addition is”,a+b);

# break;

# case 2:

# console.log(“Subtraction is”,a-b);

# break;

# case 3:

# console.log(“Multiplication is”,a\*b);

# break;

# case 4:

# console.log(“Division is”,a/b);

# break;

# default:

# console.log(“Invalid choice”);

# 

# perform arithmetic operations on 2 numbers by accepting an arithmetic operator? (Menu type program)

# console.log(“+ Addition”)

# console.log(“- Subtraction”)

# console.log(“\* Multiplication”)

# console.log(“/ Division ”)

# a=10

# b=5

# opt=’/’

# switch(opt)

# {

# case ‘+’:

# console.log(“Addition is”,a+b);

# break;

# case ‘-‘:

# console.log(“Subtraction is”,a-b);

# break;

# case ‘\*’:

# console.log(“Multiplication is”,a\*b);

# break;

# case ‘/’:

# console.log(“Division is”,a/b);

# break;

# default:

# console.log(“Invalid arithmetic operator”);

# 

# program to print day name of a week?

# daynum=5

# switch(daynum):

# {

# case 1:

# console.log(“Monday”)

# .

# .

# ..

# default:

# Console.log(“Invalid day number”)

# program to print month name?

# monthnum=5

# switch(monthnum):

# {

# case 1:

# console.log(“January”)

# program to print VIBGYOR colour name?

# colcode=’O’:

# switch(colcode):

# {

# case ‘V’:

# case ‘v’:

# console.log(“Violet colour”);

# break;

# program to find vowel or consonant?

# vc=’E’:

# switch(colcode):

# {

# case ‘A’:

# case ‘a’:

# case ‘E’:

# case ‘e’:

# case ‘I’:

# case ‘i’:

# case ‘O’:

# case ‘o’:

# case ‘U’:

# case ‘u’:

# console.log(‘Vowel’)

# default:

# console.log(“consonant”)

# program to find even or odd number

# n=5%2:

# switch(n):

# {

# case 0:

# console.log(“Even number”):

# break;

# case 1:

# console.log(“Odd number”):

# break;

# note:- instead of case 1: , use default

# Postive or negative or zero number?

# while

# it executes statements repeatedly until the given condition becomes false.

# Program to print 1 to 10 numbers?

# var i=1

# while (i<=10)

# {

# console.log(i);

# i++; (or) i+=1;

# }

# Program to print 10 to 1 numbers?

# Program to squares and cubes of numbers from 1 to 10?

# Program to print even numbers upto 20?

# Program to print odd numbers upto 19?

# Program to print your name 10 times?

# Program to print 1 to “n” numbers?