**MATLAB LOOPING STRUCTURES**

**FOR..LOOP , WHILE..LOOP, NESTED LOOPS**

**For Loop**

A **for loop** is a repetition control structure that allows you to efficiently write a

loop that needs to execute a specific number of times.

**Syntax**

The syntax of a **for loop** in MATLAB is −

for index = values

<program statements>

...

End

*values*  has one of the following forms –

|  |  |
| --- | --- |
| **Sr.No.** | **Format & Description** |
| 1 | *initval:endval*  increments the index variable from *initval* to *endval* by 1, and repeats execution of *program statements*until *index* is greater than *endval*. |
| 2 | *initval:step:endval*  increments *index* by the value step on each iteration, or decrements when step is negative. |
| 3 | *valArray*  creates a column vector *index* from subsequent columns of array *valArray* on each iteration. For example, on the first iteration, index = valArray(:,1). The loop executes for a maximum of n times, where n is the number of columns of *valArray*, given by numel(valArray, 1, :). The input *valArray* can be of any MATLAB data type, including a string, cell array, or struct. |

Example 1

Create a script file and type the following code −

[Live Demo](http://tpcg.io/gfjKHn)

for a = 10:20

fprintf('value of a: %d\n', a);

end

When you run the file, it displays the following result −

value of a: 10

value of a: 11

value of a: 12

value of a: 13

value of a: 14

value of a: 15

value of a: 16

value of a: 17

value of a: 18

value of a: 19

value of a: 20

for a = 10:2:20

fprintf('value of a: %d\n', a);

end

When you run the file, it displays the following result −

value of a: 10

value of a: 12

value of a: 14

value of a: 16

value of a: 18

value of a: 20

Example 2

Create a script file and type the following code −

[Live Demo](http://tpcg.io/trJKV0)

for a = 1.0: -0.1: 0.0

disp(a)

end

When you run the file, it displays the following result −

1

0.90000

0.80000

0.70000

0.60000

0.50000

0.40000

0.30000

0.20000

0.10000

0

Example 3

Create a script file and type the following code −

[Live Demo](http://tpcg.io/pXRiVe)

for a = [24,18,17,23,28]

disp(a)

end

When you run the file, it displays the following result −

24

18

17

23

28

# MATLAB - Loop Types

There may be a situation when you need to execute a block of code several number of times. In general, statements are executed sequentially. The first statement in a function is executed first, followed by the second, and so on.

Programming languages provide various control structures that allow for more complicated execution paths.

A loop statement allows us to execute a statement or group of statements multiple times and following is the general form of a loop statement in most of the programming languages −



MATLAB provides following types of loops to handle looping requirements. Click the following links to check their detail −

|  |  |
| --- | --- |
| **Sr.No.** | **Loop Type & Description** |
| 1 | [while loop](https://www.tutorialspoint.com/matlab/matlab_while_loop.htm)  Repeats a statement or group of statements while a given condition is true. It tests the condition before executing the loop body. |
| 2 | [for loop](https://www.tutorialspoint.com/matlab/matlab_for_loop.htm)  Executes a sequence of statements multiple times and abbreviates the code that manages the loop variable. |
| 3 | [nested loops](https://www.tutorialspoint.com/matlab/matlab_nested_loops.htm)  You can use one or more loops inside any another loop. |

## Loop Control Statements

Loop control statements change execution from its normal sequence. When execution leaves a scope, all automatic objects that were created in that scope are destroyed.

MATLAB supports the following control statements. Click the following links to check their detail.

|  |  |
| --- | --- |
| **Sr.No.** | **Control Statement & Description** |
| 1 | [break statement](https://www.tutorialspoint.com/matlab/matlab_break_statement.htm)  Terminates the **loop** statement and transfers execution to the statement immediately following the loop. |
| 2 | [continue statement](https://www.tutorialspoint.com/matlab/matlab_continue_statement.htm)  Causes the loop to skip the remainder of its body and immediately retest its condition prior to reiterating. |

# MATLAB - The while Loop

The while loop repeatedly executes statements while condition is true.

## Syntax

The syntax of a while loop in MATLAB is −

while <expression>

<statements>

end

The while loop repeatedly executes program statement(s) as long as the expression remains true.

An expression is true when the result is nonempty and contains all nonzero elements (logical or real numeric). Otherwise, the expression is false.

## Example

Create a script file and type the following code −

a = 10;

% while loop execution

while( a < 20 )

fprintf('value of a: %d\n', a);

a = a + 1;

end

When you run the file, it displays the following result −

value of a: 10

value of a: 11

value of a: 12

value of a: 13

value of a: 14

value of a: 15

value of a: 16

value of a: 17

value of a: 18

value of a: 19

# MATLAB - The Nested Loops

MATLAB allows to use one loop inside another loop. Following section shows few examples to illustrate the concept.

## Syntax

The syntax for a nested for loop statement in MATLAB is as follows −

for m = 1:j

for n = 1:k

<statements>;

end

end

The syntax for a nested while loop statement in MATLAB is as follows −

while <expression1>

while <expression2>

<statements>

end

end

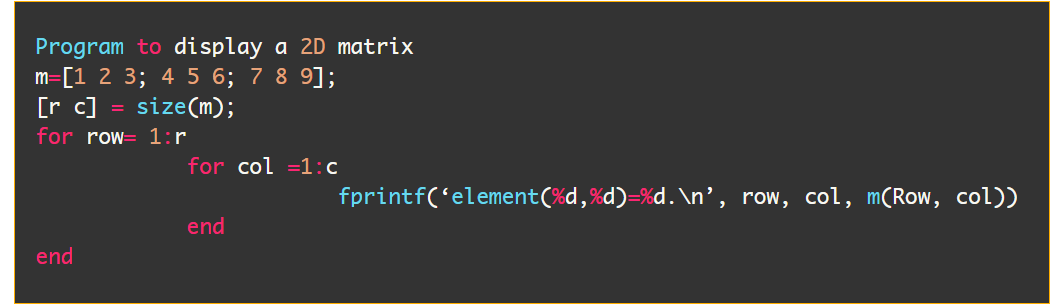
Let us write an m-file that calculates the sum: The sum to be calculated is: 1+1/22+1/32+…..+1/102.

|  |
| --- |
| the\_sum=0;  for i=1:10  the\_sum=the\_sum+1/(i^2);  end  fprintf('The sum of 10 elements gives: %f ', the\_sum) |

Nested For loops:

Example:

Program to display a 2D matrix

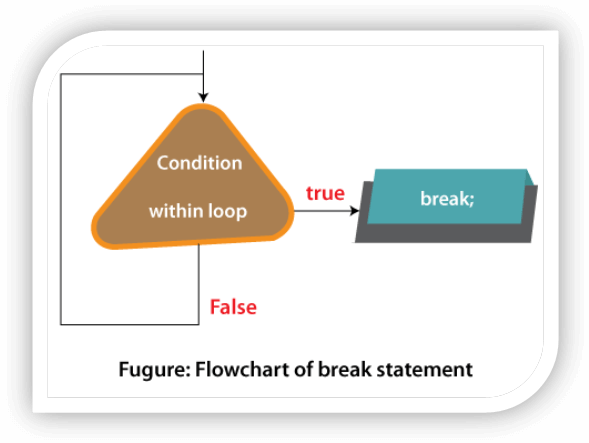
m



BBREAb

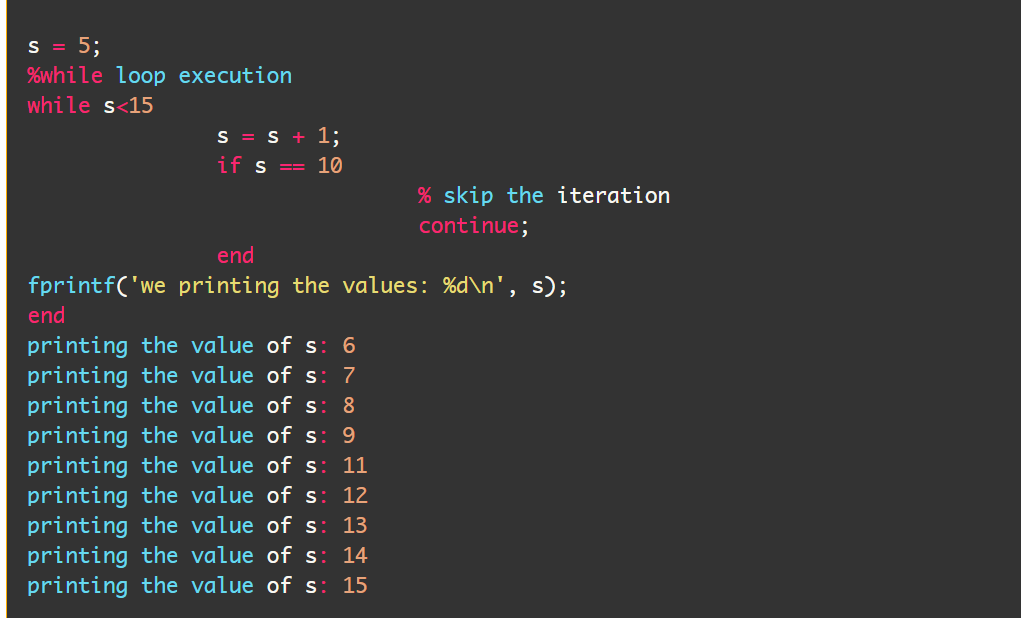
BREAK STATEMENT

Terminates the **loop** statement and transfers execution to the statement immediately following the loop. This statement is used inside loops.

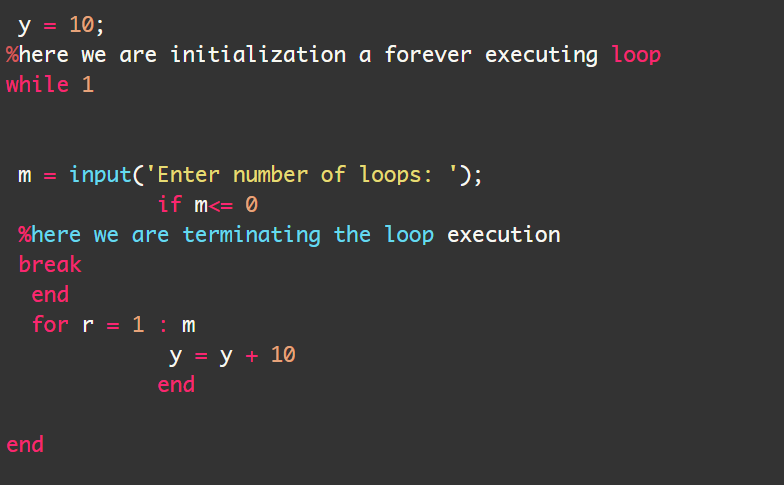


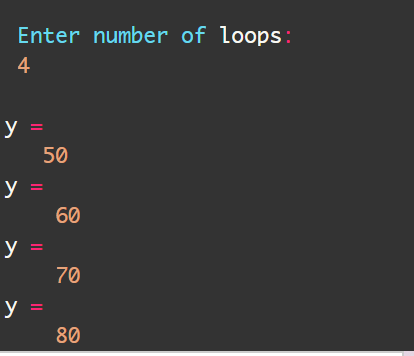
### MATLAB CONTINUE

This statement passes control to next iteration of the current loop (it maybe FOR or WHILE loop).



BREAK STATEMENT Example:





CONTINUE – FORLOOP : Example

