

# Control Statements

---

- [Download PDF](#)
- To access the updated lecture notes, please click on the following link:  
<https://yasirbhutta.github.io/matlab/docs/flow-control.html>

## Introduction

## Loops

### For Loop

- A for loop in MATLAB is a programming statement that repeats a block of code a certain number of times.
- They are used in a wide variety of applications, such as mathematical computations, data processing, and graphical plotting.

#### Syntax - for loop

```
for variable = expression
    statements
end
```

- **variable** is a loop counter variable that is initialized to the value of expression at the start of the loop.
- **expression** is a mathematical expression that evaluates to a scalar value.
- **statements** are the statements that are executed within the loop body.

The loop counter variable is incremented by 1 after each iteration of the loop. The loop continues to iterate until the loop counter variable is greater than the value of expression.

#### MATLAB Example: Printing "Hello, World!" Ten Times Using a for Loop

```
for i = 1:10
    disp('Hello, world!');
end
```

#### MATLAB Example: Print Numbers from 1 to 5

```
for i = 1:5
    disp(i);
end
```

**MATLAB Example: Print the numbers from 1 to 10**

```
% Print the numbers from 1 to 10 to the console.
for i = 1:10
    fprintf('The number is %d\n', i);
end
```

**MATLAB Example: Sum of Numbers from 1 to N**

```
N = 10;
sum = 0;
for i = 1:N
    sum = sum + i;
end
disp(sum);
```

**MATLAB Example: Calculate the sum of the numbers from 1 to 100**

```
sum = 0;
for i = 1:100
    sum = sum + i;
end
disp(sum);
```

**MATLAB Example: Print Even Numbers from 1 to 10**

```
for i = 2:2:10
    disp(i);
end
```

**MATLAB Example: Print Sum of Even Numbers from 0 to 20**

```
sum = 0;
for k = 0:2:20,
    sum = sum + k;
end;
fprintf('sum %d', sum);
```

**MATLAB Example: Calculating the Sum of Elements in an Array**

```
sum = 0;
for arr1 = [1 5 7 6],
    sum = sum + arr1;
end;
fprintf('sum = %d',sum)
```

### MATLAB Example: Displaying Elements of an Array Using a for Loop

```
A = [10, 20, 30, 40, 50];
for i = 1:length(A)
    disp(A(i));
end
```

### MATLAB Example: Infinite loop

```
for k=1:inf
    disp(k)
end
```

- A **nested loop** is a loop inside another loop. It is a powerful programming technique that can be used to solve a wide variety of problems.
- We use nested loops when we need to iterate over multiple dimensions of data. For example, we might use a nested loop to print a two-dimensional array, or to search through a list of lists.

### Example: Nested Loops - Multiplication Table

```
for i = 1:5
    for j = 1:5
        fprintf('%d x %d = %d\n', i, j, i * j);
    end
end
```

## while loop

- A while loop in MATLAB is a control flow statement that repeatedly executes a block of code until a specified condition is met. [^1]
- While loops can be used to implement a variety of algorithms, such as finding the sum of a series of numbers, searching for a specific element in a list, or performing some task until a certain condition is met.

## Syntax

The syntax for a while loop is as follows:

```
while expression
    statements
end
```

The **expression** is a logical expression that evaluates to **true** or **false**. If the expression evaluates to true, the statements in the loop body are executed. The loop then repeats, and the expression is evaluated again. This process continues until the expression evaluates to false, at which point the loop terminates.

#### Example 1: Print "Hello, world!" to the console 10 times

```
i = 1;
while i <= 10
    disp('Hello, world!');
    i = i + 1;
end
```

#### Example 2: Print the numbers from 1 to 10

```
% Initialize a variable
i = 1;

% While the variable i is less than or equal to 10, print the value of i to the console
while i <= 10
    fprintf('The value of i is: %d\n', i);

    % Increment the variable i
    i = i + 1;
end
```

#### Example 3: Sum the numbers from 1 to 100.

```
i = 1;
sum = 0;
while i <= 100
    sum = sum + i;
    i = i + 1;
end
disp(sum);
```

#### Example 4: Sum of Even Numbers

```
sum = 0; % Initialize a variable to store the sum
number = 2; % Start with the first even number

while number <= 20
    sum = sum + number; % Add the current number to the sum
    number = number + 2; % Increment to the next even number
end

fprintf('The sum of even numbers from 2 to 20 is: %d', sum);
```

### Example 5: User Input Validation

In this example, a while loop is used to repeatedly ask the user for a positive number until a valid input is provided.

```
userInput = 1; % Initialize the user input with an invalid value

while userInput >= 0
    userInput = input('Enter a positive number: ');

    if userInput <= 0
        disp('Invalid input. Please enter a positive number.');
```

### Example 6

This example shows how to use a while loop to read a line of input from the user until they enter a blank line.

```
inputStr = 'Start';
while ~isempty(inputStr)
    inputStr = input('Enter a line of text:', 's');
end

disp('You entered a blank line.')
```

## Conditional Statements / Branches

if statement

### Example

This example shows how to use a while loop to search for a specific element in an array.

```
array = [1, 3, 5, 7, 9];
elementToFind = 7;

index = 1;

while index <= length(array) && array(index) ~= elementToFind
    index = index + 1;
end

if index > length(array)
    disp('Element not found.')
else
    fprintf('Element found at index %d.', index);
end
```

line-by-line explanation of the above MATLAB code example:

```
array = [1, 3, 5, 7, 9];
```

This line creates an array named `array` containing the elements 1, 3, 5, 7, and 9. This array is used to search for a specific element.

```
elementToFind = 7;
```

This line creates a variable called `elementToFind` and assigns it the value 7. This is the element that the code will search for in the `array`.

```
index = 1;
```

This line initializes a variable `index` to 1. This variable will be used to keep track of the current index while searching for `elementToFind`.

```
while index <= length(array) && array(index) ~= elementToFind
    index = index + 1;
end
```

This is the start of a while loop. It continues to execute as long as two conditions are met:

1. `index` is less than or equal to the length of the `array`.
2. The element at the current index of `array` (given by `array(index)`) is not equal to `elementToFind`.

Inside the loop, the `index` is incremented by 1 in each iteration. This loop effectively searches for `elementToFind` in the `array`.

```
if index > length(array)
    disp('Element not found.')
else
    fprintf('Element found at index %d.', index);
end
```

This is an if statement. If the `index` variable is greater than the length of the `array`, then the code displays the message "Element not found." Otherwise, the code displays the message "Element found at index %d.", where %d is the index of the `elementToFind`.

To summarize, this code searches for `elementToFind` in the array using a while loop and reports whether the element was found or not. The result is displayed in the console.

Output:

```
Element found at index 4.
```

## Review Questions

1. What is the purpose of a for loop in MATLAB?
2. What is the syntax for creating a for loop in MATLAB?
3. What is the exit condition for a for loop in MATLAB, and how is it specified?
4. How can you create nested for loops in MATLAB, and what is their purpose?
5. How can you use the break statement in a for loop to prematurely exit the loop?
6. What is the purpose of the continue statement in a for loop?
7. How can you calculate the cumulative sum of elements in an array using a for loop?
8. What happens if you forget to increment the loop variable in a for loop?
9. What is a while loop and what is it used for?
10. What is the syntax for a while loop in MATLAB?
11. How can you use a while loop to perform a repetitive task until a certain condition is met?
12. Give an example of a while loop in MATLAB.
13. What are some tips for using while loops effectively? **Answer:** Here are some tips for using while loops effectively:
  - Use a while loop when you need to execute a block of code repeatedly as long as a condition is true.
  - Make sure to update the condition variable in the loop body.
  - Test your loops carefully to make sure that they are working as expected.
12. What are some common mistakes to avoid when using while loops? **Answer:** Some common mistakes that people make when writing while loops in MATLAB include:
  - **Infinite loops:** This occurs when the condition for the while loop is always true, which causes the loop to execute forever.

- **Unreachable code:** This occurs when the code inside the while loop is never executed because the condition for the loop is never met.

13. What is the difference between a for loop and a while loop?

**Answer:** A for loop is used to execute a block of code a fixed number of times. A while loop is used to execute a block of code repeatedly as long as a condition is true.

## Coding Questions

1. Write a MATLAB for loop that prints the numbers from 1 to 10 to the command window.
2. Write a MATLAB for loop that calculates the sum of the numbers from 1 to 10.
3. Write a while loop that prints the numbers from 1 to 100.
4. Write a while loop that calculates the sum of the first 100 even numbers.
5. Write a MATLAB program to get input from the user to display a table of a given number. The program should prompt the user to enter the number and then print a table showing the multiplication table for that number from 1 to 10.

### Example output:

```
Enter a number: 5
```

```
Multiplication table for 5
```

```
1 * 5 = 5
2 * 5 = 10
3 * 5 = 15
4 * 5 = 20
5 * 5 = 25
6 * 5 = 30
7 * 5 = 35
8 * 5 = 40
9 * 5 = 45
10 * 5 = 50
```

2. Write a MATLAB program that uses a for loop to print the numbers from 1 to 10.
3. Create a MATLAB program that calculates the sum of all even numbers from 1 to 50 using a for loop.
4. Write a for loop to print the even numbers from 1 to 100.
5. Write a for loop to find the factorial of a given number.
6. Write a for loop to find the prime numbers from 1 to 100.

## Multiple Choice

### True/False

1. A for loop can be used to execute a block of code multiple times. (True)
2. The increment expression in a for loop is evaluated before each iteration of the loop. (True)
3. A MATLAB for loop can only iterate over a range of numeric values. (False)



4. A for loop can be used to execute a block of code once. (False)
5. A MATLAB for loop can be nested inside of another for loop. (True)
6. The break statement can be used to exit a for loop early. (True) The continue statement can be used to skip the remaining code in the current iteration of a for loop and proceed to the next iteration. (True)

A for loop can be nested inside another for loop. (True)

## For loop

What is the default increment value for a MATLAB for loop?

1. ☒ 1
2. ☐ 2
3. ☐ 10
4. ☐ None of the above

Which of the following is the correct syntax for a for loop in MATLAB?

1. ☐ for i = 1:10 % code block endfor
2. ☐ for i = 1:10 do
3. ☒ for i = 1:10 % code block end
4. ☐ for i = 1:10

What is the purpose of the increment expression in a for loop?

1. ☐ To set the initial value of the loop counter
2. ☐ To specify the number of times the loop will iterate
3. ☒ To determine the step size between iterations
4. ☐ All of the above

In MATLAB, what is the primary purpose of a for loop?

1. ☐ To create a sequence of numbers
2. ☒ To execute a block of code repeatedly a specified number of times
3. ☐ To check if a condition is true
4. ☐ To perform mathematical calculations

## What is the structure of a for loop in MATLAB?

1. ☒ for i = 1:10
2. ☐ repeat 10 times
3. ☐ while i < 10
4. ☐ if i = 1 to 10

## How is the loop variable updated in a for loop?

1. ☐ Automatically by MATLAB
2. ☐ It is not updated
3. ☐ Manually within the loop
4. ☒ The loop variable cannot be changed

**What is the output of the following for loop?**

```
for i = 1:5
    disp(i);
end
```

1. ☒ 1 2 3 4 5
2. ☐ 5 4 3 2 1
3. ☐ 1 1 1 1 1
4. ☐ No output is generated.

**What is the expected output of the following MATLAB code?**

```
for i = 2:2:10
    disp(i);
end
```

1. ☐ 2 4 6 8 10
2. ☐ 1 3 5 7 9
3. ☐ 2 4 8
4. ☒ 2 4 6 10

**What is the expected output of the following MATLAB code?**

```
sum = 0;
for i = 1:5
    sum = sum + i;
end
disp(sum);
```

1. ☐ 0
2. ☐ 1
3. ☒ 15
4. ☐ 10

**What is the expected output of the following MATLAB code?**

```
for i = 1:4
    if i == 3
        continue;
    end
    disp(i);
end
```

1. ☐ 1 2
2. ☐ 1 2 3 4
3. ☒ 1 2 4
4. ☐ 4

What is the expected output of the following MATLAB code?

```
for i = 1:5
    if i == 3
        break;
    end
    disp(i);
end
```

1. ☐ 1 2 3 4 5
2. ☒ 1 2
3. ☐ 1 2 4 5
4. ☐ 3

## while loop

## References

[^1] [while loop to repeat when condition is true - MATLAB while](#)

## Muhammad Yasir Bhutta

- [Web](#)
- [Youtube](#)
- [Facebook](#)
- [Twitter](#)