

Loops in MATLAB

Loops are used to repeat a set of commands until a condition is met or until a certain number of iterations have been completed.

MATLAB provides different types of loops to handle looping requirements, including:

- While loops.
- For loops.
- Nested loops.

تستخدم التكرارات لتكرار ايعاز او مجموعة من الايعازات لغاية ان يتحقق الشرط او لغاية لحين ان ينتهي عدد التكرارات المقررة سلفا.

ما تلاب يوفر عدد من أنواع التكرارات المختلفة للتعامل مع حالات تكرار مطلوبة، هذه تشمل:

1. التكرار باستخدام While

2. التكرار باستخدام For

3. التكرارات المتداخلة

While Loop

A while loop is a **programming language** control structure. It executes a statement or group of statements repeatedly as long as a specified condition is proper.

Syntax:

while condition

% Commands

End

condition is a logical expression that must be true to continue the loop.

The commands inside the loop are executed until the condition is false.

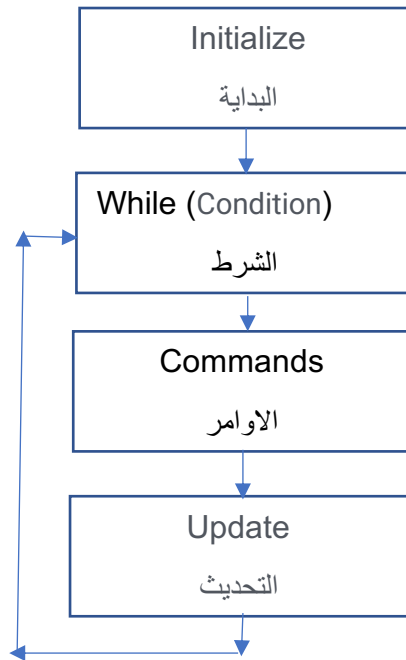
تكرار while هو هيكل سيطرة في لغة البرمجة. فهو يقوم بتنفيذ عبارة او مجموعة عبارات (أوامر او ايعازات) بشكل متكرر طالما الشرط المحدد يتحقق. ويكتب قواعديا كما يلي

While (شرط)

ايعازات

End (يجب ان تنتهي بكلمة النهاية)

المخطط العام لكتابة while هو



الان لنشرح هذا المخطط

1. أولا البداية، يجب ان يكون لدينا متغير له قيمة ابتدائية لكي نستخدمه بالشرط.. لاحظ اننا لا نستطيع استخدام متغير بالشرط دون ان تكون له قيمة ابتدائية لكي يستطيع البرنامج الدخول الى حلقة التكرار، لذلك يجب ان يكون المتغير محدد وله قيمة ابتدائية قبل حلقة التكرار.
2. الشرط.. ويجب ان نفكر بالشرط جيدا بحيث عندما يتحقق الشرط (يكون True) يتم التنفيذ وعندما لا يتحقق الشرط (يكون False) يتوقف التنفيذ.
3. تكتب الأوامر المراد تنفيذها عند تحقق الشرط.

4. التحديث موضوع مهم وهنا المقصود تحديث المتغير المستخدم بالشرط كان تزداد قيمته او تقل او تثبت الخ. ان عملية التحديث تضمن عدم استمرار تنفيذ التكرار الى ما لانهاية.
5. بعد عملية التحديث سيعود التكرار الى الشرط لفحص الشرط فاذا كان الشرط متحقق ينفذ وهكذا تستمر العملية لحين ان يكون الشرط غير متحقق فيتوقف التكرار وينتقل الى الابعاز الذي بعد نهاية التكرار.

مثال // اطبع الأرقام من 20 الى 30

لاحظ هنا عملية الطباعة ستكون متكررة لذلك نستخدم ايعاز التكرار. وطالما المطلوب من 20 فعليه ممكن ان نضع قيمة ابتدائية للمتغير هي 20.. لاحظ التحديث هنا هو إضافة واحد للمتغير كل مرة وبالتالي فانه سيصل الى 30 التي لا تحقق الشرط ويتوقف التكرار.

```
num = 20;
```

```
% while loop execution
```

```
while( a < 30 )
```

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

```
    a = a + 1;
```

```
end
```

النتيجة بعد التنفيذ

```
value of num: 20
value of num: 21
value of num: 22
value of num: 23
value of num: 24
value of num: 25
value of num: 26
value of num: 27
value of num: 28
value of num: 29
```

example:

find the square of values till 5. Here var is the variable name. The value assigned to var is 1, varying from 1 to 5.

Sol:

```
var = 1;  
while(var <=5)  
op=var*var;  
disp(op)  
var = var + 1;  
end
```

example: find the sum of set of numbers from 21 to 98.

Sol:

N=21;

Sum = 0;

While (N<=98)

Sum = sum+ N;

N = N + 1;

End

disp (sum);

example:

find sum of the set of numbers where the last number in the set equal 30;

sol:

```
x= 0;  
while (x!=30)  
x=input (“enter any number”)  
sum = sum + x;  
end  
disp(sum);
```

example:

find (y) where,

$$y = 1 + x + x^2 / 2 + x^3 / 3 + x^4 / 4 \dots + x^n / n$$

sol:

```
sum = 1;  
  
x = input (“ enter value of x”);  
  
n = input (“ enter value of n”);  
  
counter = 1;  
  
while (counter < n)  
  
sum = sum + pow (x, n) / n ;  
  
count = count + 1;  
  
end  
  
disp (sum);
```

For Loop

A **for loop** is a repetition control structure that allows you to efficiently write a loop that needs to be executed a specific number of times.

It's similar to the while loop, but it has two significant differences:

- You can say how many times the loop should repeat instead of having the loop continue until some condition is true.
- The condition at the end of the loop is checked before each iteration instead of after each iteration with while loops.

Syntax:

```
for i = start: increment: end
```

```
% commands
```

```
end
```

i is a loop variable that takes on values from start to end with an increment of increment.

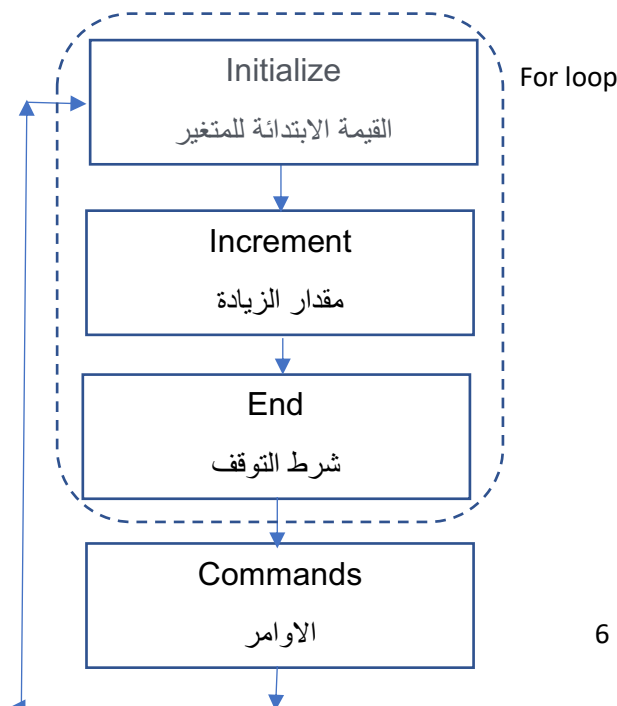
The commands inside the loop are executed for each value of *i*.

تكرار for هو تكرار هيكل سيطرة يسمح لك بكتابة تكرار بشكل كفوء لتنفيذ ايعاز او مجموعة ايعازات لعدد من المرات المحددة مسبقا.

هو مشابهة لتكرار while لكنه يختلف بما يلي:

هنا بتكرار for نحدد عدد مرات التكرار مسبقا بينما في while لا نحدد ولا نعرف عدد مرات التكرار بشكل مسبق لأنه يعتمد على تحقق الشرط.

المخطط التوضيحي لحلقة التكرار for هو



لنوضح مخطط التكرار for

1. ان حلقة التكرار for يجب ان تبدأ بمتغير (هذا المتغير يمثل عداد). المتغير عادة يكون متبوع بعلامة المساواة.
عدد القيم بعد علامة المساواة اما قيمتان او ثلاثة تفصل بين قيمة واخرى النقطتان المتعامدتان (:).
اذا كانت عدد القيم فقط قيمتان بعد المساواة فان الأولى تمثل القيمة الابتدائية (أي ان العداد يبدأ بهذه القيمة، والقيمة الثانية ستكون القيمة التي يصلها العداد ليتوقف). في هذه الحالة فان مقدار الزيادة في كل مرة هو واحد.
اما اذا كان عدد القيم ثلاثة فان اول رقم بعد علامة المساواة يمثل القيمة الابتدائية للعداد (بالإمكان استخدام أي قيمة ابتدائية وحسب السؤال). بعد اول رقم تاتي النقطتان المتعامدتان (:). والقيمة الثانية (الوسط) لتمثل مقدار الزيادة في كل مرة، والقيمة الثالثة تمثل القيمة التي سيصل لها العداد ليتوقف التكرار (أي شرط التوقف).
لاحظ هنا ان عدد مرات التكرار عادة تذكر بالسؤال بشكل مباشر او غير مباشر. ومقدار الزيادة أيضا ممكن استنتاجها من السؤال.
2. النقطة الأولى تمثل ايعاز البداية للتكرار يأتي بعدها الأوامر المطلوب تنفيذها.
3. بعد تنفيذ الأوامر المطلوب تنفيذها يعود المؤشر الى بداية حلقة التكرار (for) ويتم هنا مقارنة قيمة العداد بالقيمة النهائية فاذا لازال لم يصل له يستمر التنفيذ لحين ان يصل العداد الى قيمة التوقف المحددة. هنا عملية زيادة العداد تحدث اليا دون الحاجة الى تحديث داخل حلقة التكرار كما فعلنا في .while
4. دائما ينتهي التكرار باستخدام for بايعاز النهاية .end