



Lesson 6 for loop

الكمبيوتر له مميزات كثيرة جداً
أحد هذه المميزات :

هو التكرار إذا كان لدى 100 ورقة أريد الإمضاء عليهم سيكون من المستحيل أن يكون أول امضاء فى الورقة كأخر امضاء فى الورقة الأخيرة كإنسان سيتعب ولن يستطيع تكرار نفس الشئ بنفس الدقة مرات عديدة، لكن الكمبيوتر ليس كذلك ، باستطاعته تكرار أى عملية مرات عديدة وبنفس الدقة، وجدنا ذلك في "الطابعة" (printer) لنفترض أننا قمنا بطباعة ورقة منها 200 نسخة أو 300 نسخة أو ألف نسخة سيكون جميع النسخ مثل بعضهم تماماً كيف سنعمل ذلك بالبرمجة؟ أريد أن أطبع شئ بالبرمجة! كيف أقوم ببرمجة الكمبيوتر لتكرار هذه العملية عدد معين من المرات.

هناك أمر يسمى **for**

إذا كنا نريد تنفيذ الكود عدة مرات محددة، فمثلاً إذا كنا نريد تنفيذ كود معين 10 مرات، نضعه داخل **حلقة (loop)** تعيد نفسها 10 دورات.

```
for (initialisation; condition; increment أو decrement){
    // statements
    هنا يتم وضع الكود المراد تنفيذه لعدد معين من المرات //
}
```

الخطوات التى يعمل بها امر **for** بالترتيب :

1 - **التهيئة (initialisation)** : هي أول خطوة تنفذ لمرة واحدة، لابد بعدها فاصلة منقوطة ; Semicolon

2 - **الشرط (condition)** : هي ثاني خطوة تنفذ، هنا يكون شرط يحدد متى تتوقف **(loop)** وهنا فى كل دورة يتم فحص الشرط هل هو **true** أو **false**



إذا كان الشرط **true** فسوف يكمل الدوران
أما إذا كان **false** فتتوقف (**loop**) ، لابد بعدها فاصلة منقوطة ; Semicolon .

3 - الكود الذى بداخل الأقواس (**statements**): الخطوة الثالثة يتم تنفيذ جميع الاكواد التى بداخل الأقواس { }

4 - الزيادة او النقصان (**increment** أو **decrement**) : الخطوة الرابعة تنفذ فى كل (**loop**) سنقوم بزيادة قيمة المتغير (**variable**) او ننقص منها.

دعونا نرى هذا الأمر:

```
#include <stdio.h>
int main( ) {
    int i;
    for ( i = 0; i < 5; i++) {
        printf("Gammal ");
        printf("Tech\n");
    }
}
```

output:

```
Gammal Tech
Gammal Tech
Gammal Tech
Gammal Tech
Gammal Tech
```

(قم بتجربة الكود بنفسك واضغط هنا)

ملحوظة إذا نسيت كتابة أقواس ال for { } فى المثال السابق كانت النتيجة ستكون :



Gammal Gammal Gammal Gammal Gammal Tech

فتذكر دائما إذا كان هناك أكثر من أمر يتم تنفيذهم في حيز معين يجب وضعهم في أقواس { } ليتم تنفيذهم في نفس الحيز

ال **for loop** سوف تقوم بتكرار جميع الأوامر التي بداخل الأقواس عدد معين من المرات

لاحظ عندما كتبنا في الخانة الثالثة ($i++$) داخل أقواس ال **for** ما معنى ذلك ؟ هناك اختصار في لغات البرمجة لزيادة أى رقم بقيمة 1 أو نقصانه بقيمة 1 و هو كتابة اسم المتغير ثم بعده نكتب ++ للزيادة أو -- للنقصان إذا كنت تتذكر من الدرس السابق تكلمنا عن أن علامة = معناها ضع قيمة ما على يمين العلامة في المتغير الذى على اليسار .

توقع ماذا سيكون الناتج لهذا الكود :

```
#include <stdio.h>
int main( ) {
    int i = 5;
    i = i + 2;
    printf("%d \n", i);
}
```

الناتج سيكون 7

هل عرفت لماذا ؟

لأن في السطر الرابع توجد علامة (=) ، إذا سيذهب البرنامج إلى ما على يمين العلامة و يقوم بحساب قيمته و هو $5 + 2$ يساوى 7 ثم سيضع هذه القيمة في المتغير الذى على يسار العلامة و هو ال i ، إذا ستصبح قيمة ال i تساوى 7 .



لكن هناك إختصار قاموا به عند عمل لغة البرمجة ، تم الإتفاق على أن عندما يكون المتغير الذى سيتم تغير قيمته (الموجود على يسار علامة =) عبارة عن طرف مستقل فى الطرف الأيمن ، يمكن أن يتم حذفه مع كتابة العملية التى كان يقوم بها فى الطرف الأيمن .
مثل :

```
#include <stdio.h>
int main( ) {
    int i = 5;
    i += 2;
    printf("%d \n", i);
}
```

سيكون الناتج أيضا 7 .
حيث أن $i = i + 2$ لا تختلف عن $i += 2$

إذا فى ال for loop ليس بالضرورة أن تكون الخانة الأخيرة مسؤولة عن زيادة القيمة بمقدار 1 أو نقصانها بمقدار 1 ، لكن يمكن تغيير القيمة كيفما تشاء .
لنفترض أنك تريد كتابة برنامج لطباعة الأرقام الزوجية من 0 إلى 10 يمكنك كتابته على النحو التالى :

```
#include <stdio.h>
int main( ) {
    int i;
    for ( i = 0; i <= 10; i+=2 )
        printf("%d\n ", i);
}
```

مثال آخر لتوضيح كيفية عمل ال for loop بالتفصيل :



```
#include <stdio.h>
```

```
int main( ) {
```

```
    int i;
```

```
    // variable هنا نقوم بتعريف
```

```
    for (i = 0; i < 10; i++) {
```

```
        printf("%d ", i);
```

```
    }
```

```
}
```

```
output: 0 1 2 3 4 5 6 7 8 9
```

(قم بتجربة الكود بنفسك واضغط هنا)