



Lesson 11 Practice 3

سنقوم بعرض بعض الأمثلة على **if** و **for**
if تؤثر على أمر واحد بعدها طالما لا يوجد أقواس { }

if (الشرط) {

في حالة وجود هذه الأقواس يتم تنفيذ جميع الأوامر التي بداخل هذا القوس إذا تحقق الشرط //

}

أما إذا لم نضع الأقواس { } ستؤثر على أمر واحد فقط بعدها

مثال:

```
int x = 5;
if (x == 1);
printf("Hello");
```

هنا ستقوم بتنفيذ أمر الطباعة لأن

if تؤثر على أمر واحد فقط هنا ستؤثر على ; فقط وبالتالي سيتم تنفيذ أمر الطباعة

.....

- ماذا تعني **x == 5**

قارن بينهما هل قيمة المتغير x تساوي 5

- وماذا تعني **x = 5**

أن نقوم بنقل قيمة 5 للمتغير x

if (رقم)

```
printf("true");
```



- أي رقم غير الصفر يكون الشرط **true** وسيقوم بأمر الطباعة

.....

- أمر **for** يؤثر على أمر واحد فقط طالما لا يوجد أقواس **{ }**
- يمكنني وضع أمر **for** في هذه الهيئة وهذا معناه أن يستمر **loops** الا ما لانهاية.

```
for ( ; ; ) {
    الكود المراد تنفيذه //
}
```

.....

```
int i = 0;
for ( ; i < 2 ; i++);
printf("Hello");
```

- هذا معناه انه سيتم تنفيذ أمر الطباعة مرة واحدة فقط لأن هناك لامة (;) بعد أقواس ال **for** ما يجعلها تدور بدون تنفيذ أى أوامر عدا زيادة ال **i** إلى أن يتوقف الشرط عن التحقق عندما تصل قيمة ال **i** إلى 2 فيخرج البرنامج من ال **for** و يقوم بتنفيذ الأوامر التالية بطريقة مباشرة

.....

```
int i = 0;
for ( ; i; i++)
printf("Hello");
```

هنا لن يتم تنفيذ أمر الطباعة لأن الشرط داخل ال **for** عبارة عن **i** و في البداية قيمة **i** كانت تساوى 0 فلن يتحقق الشرط و لن تعمل ال **for** .
لفهم الموضوع يجب علينا معرفة ترتيب الخطوات التي يعمل بها أمر **for** :

تتكون جملة ال **for** من 3 خانات :

```
for (1; 2; 3) {
```



الأوامر التي ستتكرر //

}

- **الخانة رقم 1:** هي أول خانة يتم فحصها عند تشغيل أمر ال **for** و عادة ما يتم فيها إعطاء قيمة للمتغير المستخدم ك **counter** داخل ال **for** أو إعطاء قيم لأكثر من متغير و يفصل بينهم علامة (,) مثل (**for (i=0 , j=2 ; ;**) يمكن ترك هذه الخانة فارغة حسب إحتياج البرنامج، و ما هو موجود فى هذه الخانة هو أول ما ينفذ عند بدء ال **for** ولا يتم العودة إلى هذه الخانة مرة أخرى فهي يتم تنفيذها مرة واحدة فقط فى بداية عمل امر **for**

- **الخانة رقم 2:** هي ثانى خانة يتم فحصها عند تشغيل أمر ال **for** و هي خانة الشرط، فيها يتم فحص الشرط هل هو **true** أم **false** و يمكن أيضا كتابة أكثر من شرط و لكن باستخدام ال **operators** مثل **&&** , **||** , يمكن أيضا أن تترك هذه الخانة فارغة و لكن فى هذه الحالة ستكون ال **loop** عبارة عن **infinity** أى **loop** أى أنها لن تتوقف نهائيا إلا فى حالة أن كان من ضمن الأوامر التي يتم تنفيذها داخل ال **for** يوجد أمر **break** لاذى يجعل البرنامج يخرج من ال **loop**. يتم العودة إلى هذه الخانة بعد كل دورة حيث يتم التأكد من أن الشرط متحقق (**valid**) قبل بدء تنفيذ الأوامر التي بداخل ال **for**

- **الخانة رقم 3:** بعد التأكد من صحة الشرط و تنفيذ الدورة الأولى داخل ال **for** يبدأ البرنامج فى الاستعداد للبدء فى الدورة الثانية و الثالثة و هكذا إلى أن تتوقف ال **for**، و لكن قبل البدء فى أى دورة بعد الدورة الأولى، يتم تنفيذ الأوامر فى الخانة الثالثة و التي تكون عادة أوامر متعلقة بزيادة أو نقصان ال **counters** المستخدمة داخل ال **for loop**، ننفذ ما بداخلها من أوامر مثل **i++** أو **i--** ثم بعدها نفحص الشرط فى الخانة رقم 2 هل هو **true** أم **false** وتستمر الدورات حتى يصبح الشرط **false** فتتوقف

فى المثال السابق : لن يتم تنفيذ الأوامر بداخل ال **for** لأن الشرط فى المرحلة رقم 2 قيمته **false** (لأن ال 0 يعنى **false** و أى قيمة أخرى تعنى **true**)



.....

```
int i = 1;
```

```
for( ; i ; i++)
```

```
printf("Hello");
```

هنا يستمر الكود الا مالا نهاية لأن الشرط الأوسط **i** ليس بصفر ومعناه **true** يتم تنفيذ أمر الطباعة إلى مالا نهاية

.....

يمكنك استخدام متغيرين داخل أمر **for** لكن لابد من وضع , بين المتغيرين مثال :

```
int i = 5, j = 1;
```

```
for ( ; i ; i--, j-- )
```

```
printf("%d %d \n", i, j);
```

```
int i, j;
```

```
for (i = 5, j = 1; i ; i--, j--)
```

```
printf("%d %d \n", i, j);
```

output :

5 1

4 0

3 -1

2 -2

1 -3

(قم بتجربة الكود بنفسك واضغط هنا)

لا يمكنني وضع شرطين في الخانة الوسطى ووضع بينهم , لكن لعمل ذلك نضع &&



ومعناها **and** وظيفتها أن تجعل البرنامج يفحص الشرطين معاً وينفذهم إذا تحققوا سوياً
 أى كانت قيمة كل منهما **true**
 أو يستخدم **||** وهذا معناه أن يفحص الشرطين وإذا كان أحدهما **true** ينفذ الكود الموجود
 فى امر **for** وإما إذا كان الاثنين **false** فلن يتم تنفيذ الكود و تتوقف ال loop

مثال :

```
int i, j;
for (i = 5, j = 1; i > 0 && j < 6; i--, j++)
    printf("%d %d \n", i, j);
```

output :

```
5 1
4 2
3 3
2 4
1 5
```

(قم بتجربة الكود بنفسك واضغط هنا)

عند استخدام **&&** لابد أن يكون الشرطين **true** إما إذا كان أحدهما **false** لن ينفذ

.....

```
int i, j;
for (i = 5, j = 1; i > 0 || j > 6; i--, j++)
    printf("%d %d \n", i, j);
```

output :

```
5 1
4 2
```



3 3

2 4

1 5

عند استخدام || إذا كان أحد الشرطين **true** ينفذ الكود إما إذا كان الإثنين **false** لن ينفذ