



lesson 45 Boolean

هناك **data type** تسمى **bool** و هي اختصار لكلمة **Boolean** ، هذا النوع يقوم بتسجيل **true** أو **false** ، لا يعرف أى قيمة أخرى غير ال **true** و ال **false** .
لكي نستخدم هذا النوع يجب تضمين المكتبة التي يحتوى عليها هكذا :

```
#include <stdbool.h>
```

ثم نقوم باستخدامه بهذا الشكل :

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

```
#include <stdbool.h>
```

```
int main( ) {  
    bool x = true ;  
    if( x == true )  
        printf(" yes\n ");  
    else  
        printf(" No\n ");  
}
```

output:

yes

و كما نعرف أن الكمبيوتر يقرأ ال 0 على أنه **false** و ال 1 على أنه **true** فيمكن أن نسجل 0 بدل من **false** و 1 بدل من **true** :

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

```
#include <stdbool.h>
```



```
int main( ) {  
    bool x = 1 ;  
    if( x == true )  
        printf(" yes\n ");  
    else  
        printf(" No\n ");  
}
```

output:

yes

في الواقع الكمبيوتر يقرأ ال 0 فقط على أنه **false** و أي رقم آخر على أنه **true** ليس ال 1 فقط ، لذلك يمكن إعطاء أي قيمة لل bool غير الصفر و سيعتبرها true :

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

```
int main( ) {  
    bool x = 5;  
    if( x == true )  
        printf(" yes\n ");  
    else  
        printf(" No\n ");  
}
```

output:



yes

.....

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

```
int main( ) {
    bool x = -50;
    if( x == true )
        printf(" yes\n ");
    else
        printf(" No\n ");
}
```

output:

yes

.....

إذا وضعنا **صفر** هنا سيكون الناتج فقط **false**

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

```
int main( ) {
    bool x = 0;
```



```
if( x == true )
    printf(" yes\n ");
else
    printf(" No\n ");

}
```

output:

No

.....

يمكننا استخدام **boolean** في العديد من الاستخدامات، لكن أكثرها يكون داخل **if** او شئ يقوم بفحص الشرط .

مثال نقوم بعمل **bool** لفحص الرقم إذا كان الرقم **odd** أو **even** باستخدام
: function

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

```
#include <stdbool.h>
```

```
bool isEven(int x) {
```

```
    if (x % 2 == 0)
```

```
        return true;
```

```
    // الشرط هنا إذا كان الرقم زوجي
```

```
    // return true; سينفذ إذا تحقق الشرط
```

```
    return false;
```

```
    //return false; إذا لم يتحقق الشرط ينفذ
```

```
}
```



```
int main( ) {  
    int num;  
    printf("Enter num : ");  
    scanf("%d", &num);  
    if ( iseven(num) )  
        printf("Even \n");  
    else  
        printf("odd \n");  
}
```

input:

5

output:

odd

input:

6

output:

even

(قم بتجربة الكود بنفسك واضغط هنا)

لاحظ أن في ال function عندما تتنفيذ ال return فهي تجعل ال function توقف و لا تنفذ أي أوامر بعدها و تعود بالنتائج إلى المكان الذي تم استدعاؤها فيه.

نحن عندما نستخدم **function bool** يكون بها مجموعة من الأكواد ونريد أن نرجع **true** أو **false** كما يمكننا اختصار الكود هكذا :

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



```
#include <stdbool.h>
```

```
bool iseven(int x) {  
    return x % 2 == 0 ;  
}
```

```
int main( ) {  
    if (iseven(5))  
        printf("Even \n");  
    else  
        printf("odd \n");  
}
```

output:

odd

(قم بتجربة الكود بنفسك واضغط هنا)

الشرط هذا نفسه `0 == 2 % x` له قيمة إما `true` أو `false` و هكذا تم استخدامه مباشرة بدون الحاجة إلى `if`