



Lesson 10 Switch

أمر **switch** شبيه بأمر **if** نستخدم أمر **switch** إذا كنا نريد فحص متغير **variable** مع مجموعة من الاحتمالات، أي أنه إذا تساوت هذه القيمة مع الشرط سينفذ الأوامر الذي وضعناها في هذا الشرط

```
switch(variable){
case constant :
    // الكود المراد تنفيذه
    break; // معناه إذا تحقق الشرط توقف هنا
default:
    // معناه إذا لم يتم تحقيق أي من الحالات يتم تنفيذ الكود الموجود هنا
}
```

مثال :

```
#include <stdio.h>
int main( ) {
    int day;
    // قمنا بحجز مكان في الذاكرة
    printf("Enter number of day : ");
    scanf("%d", &day);
    // قام المستخدم بإدخال الرقم
    switch (day) {
    case 1:
        // إذا كان واحد ، إذاً اطبع
        printf("Sunday \n");
        break;
    case 2:
        // إذا كان 2، إذاً اطبع
```



```
    printf("Monday \n");
    break;
case 3:
    // إذا كان 3، إذاً اطبع //
    printf("Tuesday \n");
    break;
case 4:
    // إذا كان 4، إذاً اطبع //
    printf("Wednesday \n");
    break;
case 5:
    // إذا كان 5، إذاً اطبع //
    printf("Thursday \n");
    break;
case 6:
    // إذا كان 6، إذاً اطبع //
    printf("Friday \n");
    break;
case 7:
    // إذا كان 7، إذاً اطبع //
    printf("Saturday \n");
    break;
default:
    // إذا لم يكن أى من هذه الحالات نفذ الكود الموجود هنا //
    printf("Wrong Enter");
}
}
```



(قم بتجربة الكود بنفسك واضغط هنا)

لاحظ أن أمر switch شبيه بأمر if في حالة المقارنى باستخدام علامة (==) لكن لا يمكن استخدام مقارنات من نوع > او < و هكذا، فعندما نكتب case 5 مثلاً

```
switch (day) {
    case 1:
        // إذا كان واحد ، إذاً اطبع //
        printf("Sunday \n");
        break;
}
```

هذا يعنى إذا كان المتغير day قيمته تساوى 1 سينفذ هذه ال case أى سينفذ الأوامر بداخلها و سيقوم بطباعة " Sunday " .

لاحظ أن يجب استخدام أمر break بعد انتهاء كل case و ذلك لأن عند تحقق case معينة و يبدأ البرنامج فى تنفيذ الأوامر التى بداخلها، لا يتوقف البرنامج عندما تنتهى ال case و يتم تنفيذ الأوامر الموجودة فى ال case التالية حتى لو شرطها غير متحقق. مثال

```
#include <stdio.h>
int main( ) {
    int day;
    printf("Enter number of day : ");
    scanf("%d", &day);
    switch (day) {
        case 1:
            printf("Sunday \n");
        case 2:
```



```

printf("Monday \n");
case 3:
printf("Tuesday \n");
break;
default:
printf("Wrong Enter \n");
}
}

```

فى هذه الحالة إذا كان الرقم الذى سيدخل هو 1، سيقوم البرنامج بطباعة

Sunday

Monday

Tuesday

سنجد أنه لم يتوقف بعد طباعة Sunday فقط، على الرقم من أن الأوامر بداخل case 1 انتهت و case 2 و case 3 غير صحيحين، لكن هذا حدث بسبب عدم وجود أمر break الذى يخرج من ال switch .

ملحوظة أمر break يمكن استخدامه فى مواضع كثيرة غير ال switch و يكثر استخدامه مع ال loops، حيث أن break تجعل البرنامج يخرج من أقرب loop هو فيها .

مثال :

```

#include <stdio.h>
int main( ) {
int i ;
for ( i = 0 ; i < 10 ; i++ ){
if ( i ==5 )
break ;
printf("%d ", i );
}
}

```



```
    }
}
```

هنا الناتج سيكون 0 1 2 3 4

على الرغم من أن ال loop كان يجب أن تستمر إلى أن تصل ال i إلى رقم 10 لكن بسبب وجود أمر break الذى سيتم تنفيذه عندما وصلت ال i لقيمة 5 فاضطر البرنامج إلى الخروج فى هذه اللحظة من ال loop دون أن يكمل الطباعة.

لاحظ أن أمر break فى المثال السابق هو الأمر الوحيد داخل ال if و أمر الطباعة غير محكوم بشرط ال if و ذلك لأن ال if تسيطر على أمر واحد فقط بعدها مادام ليس هناك أقواس بعد ال if { } تحتوى على أكثر من أمر مرتبطين بنفس شرط ال if .

هناك أمر شبيه بأمر break و هو أمر **continue** و لكن يختلف فى آلية عمله .
أمر **continue** فى ال loops تجعل البرنامج يتخطى الدورة التى هو فيها و يذهب للدورة التالية دون تنفيذ باقى الأوامر فى الدورة الأولى .
مثال :

```
#include <stdio.h>
int main( ) {
    int i ;
    for ( i = 0 ; i < 10 ; i++ ){
        if ( i == 5 )
            continue;
        printf("%d ", i);
    }
}
```

هنا الناتج سيكون 0 1 2 3 4 6 7 8 9



سنجد أنه عندما كانت ال i تساوى 5 لم يتم طباعتها و استمر البرنامج فى تنفيذ الدورات التالية .