

Conditions & loops

C programming
session 8



★ Conditions

- 1 if statement
- 2 Switch

★ Loops

- 1 While
- 2 For



★ if :

if (الشرط)

{



;

}

- عندما يتحقق الشرط الموجود داخل if يدخل ال compiler لينفذ الأمر بتاعي ()

- Ex:

```
#include <stdio.h>
int temp = 0;
void main()
{
    printf("please enter your temp: ");
    scanf("%i",&temp);
    if (temp == 30)
    {
        printf("turn on the fan");
    }
}
```

*إذا أدخل المستخدم الرقم 30 سيتم طباعه turn on the fan علي الشاشة ,
لأن بدخول رقم 30 يكون تحقق الشرط.

logical operators

- يجب ان يكون الشرط كله متحقق عشان ينفذ الأمر (&&):
- بوجود شرط واحد متحقق كافي لينفذ الأمر (OR): ||

• &&

```
#include <stdio.h>
int temp = 0;
char x = 'o';
void main()
{
    printf("please enter your temp: ");
    scanf("%i", &temp);
    if (temp == 30 && x == 'f')
    {
        printf("turn on the fan");
    }
    if (temp == 30 && x == 'o')
    {
        printf("no change");
    }
}
```

please enter your temp: 30
no change

*لم يتم تحقيق الشرط الأول ف تم الدخول علي if الثانية و تنفيذ أمرها لأن شرطها كله متحقق .



```
#include <stdio.h>
int temp = 0;
char x = 'o';
void main()
{
    printf("please enter your temp: ");
    scanf("%i", &temp);
    if (temp == 30 || x == 'o')
    {
        printf("turn on the fan");
    }
}
```

```
please enter your temp: 20
turn on the fan
```

*الأمر هنا اتنفذ عادي رغم إن درجة الحرارة
مش 30 لان or تستكفي بوجود شرط واحد
متحقق مش لازم الاتنين .

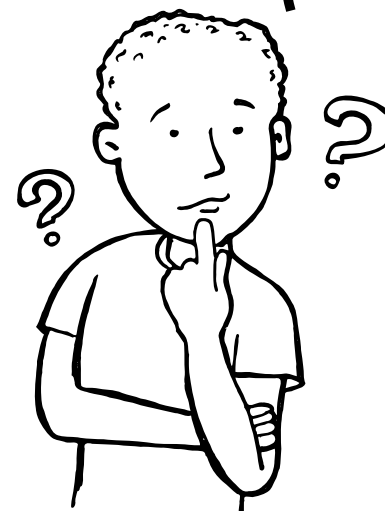
- Ex :

```
#include <stdio.h>
int day = 0;
void main()
{
    printf("please enter number of the day:\n ");
    scanf("%i", &day);
    if (day == 1)
    {
        printf("saturday");
    }
    else if (day == 2)
    {
        printf("sunday");
    }
    else if (day == 3)
    {
        printf("monday");
    }
    else if (day == 4)
    {
        printf("tuesday");
    }
    else if (day == 5)
    {
        printf("wednesday");
    }
    else if (day == 6)
    {
        printf("thursday");
    }
    else
    {
        printf("friday");
    }
}
```

```
please enter number of the day:
3
monday
```

*هندخل علي كل if لحد إلي الشرط بتاعها
اتحقق فننفذ الأمر وخلاص كدا.

لو عندنا أكثر من حالة يفضل إستخدام؟ 



☆ Switch :

```
switch(المتغير بتاعي)
{
    case :
        [ ] ;
    break;
    case :
        [ ] ;
    break;
    case :
        [ ] ;
    break;
    case :
        [ ] ;
    break;
}
```

• أهمية break :

عشان لما الشرط بتاع الحالة يتحقق
ينفذ الأمر بتاع الحالة ويخرج علي طول
ملوش دعوة بباقي الأوامر .



- Ex :

```
#include <stdio.h>
int day = 0;
void main()
{
    printf("please enter number of the day:\n ");
    scanf("%i", &day);
    switch (day)
    {
        case 1:
            printf("saturday");
            break;
        case 2:
            printf("sunday");
            break;
        case 3:
            printf("monday");
            break;
    }
}
```

```
please enter number of the day:
2
sunday
```

- إلى الحالة بتاعتو اتحققت بس هو الي بيتنفذ الأمر بتاعو .



Loops



while :

while (الشرط)

{

;

break;

}

- ال loop دا يعني حاجه بتتكرر .
- لو احنا محطناش break وكان الشرط متحقق هيفضل ينفذ قي الأمر .
- لذلك حطينا break عشان تطلع منو .



- Ex :

```
#include <stdio.h>
int id = 123;
int pass;
void main()
{
    while (id == 123)
    {
        printf("id is true \n");
        printf("enter your pass: \n");
        scanf("%i" ,& pass);
        while (pass == 456)
        {
            printf("pass is true");
            break;
        }
        break;
    }
}
```

1:

```
id is true
enter your pass:
456
pass is true
```

2:

```
id is true
enter your pass:
456
pass is true
id is true
enter your pass:
```

*بالكود الموجود دا رقم 1 هو ال output , لكن لو شيلنا ال break الأخيرة هيبقي output رقم 2

(لأن ال loop لسه مكمله موقفتش فالكود هيفضل مكمل عادي)

☆ For : Data Type

```
( العملية ; الشرط ; بداية العد )  
for(   
{  
     ;  
}
```

- لو عندي أمر وعاييز أنفذه عدد معين من المرات



- *أول حاجه هنشوف الشرط متحقق ولا لا , لو متحقق هيروح ينفذ الأمر , بعد كذا بروح اعمل العملية علي الرقم واثقق من الشرط لو متحقق هينفذ الأمر , واروح اعمل العمليه ثاني بس علي الرقم الجديد واشوف الشرط , وهفضل كذا لحد ما الشرط ميكونش متحقق , فأقوم خارجة من ال For loop .

- Ex:

```
#include <stdio.h>
void main()
{
    for (int i = 0; i < 5; i = i + 1)
    {
        printf("name \n");
    }
}
```

```
name
name
name
name
name
name
```

- أولاً الشرط متحقق فطبع اول مرة ،عمل العملية ف i بقت ب 1 ومازال الشرط متحقق فطبع للمرة الثانية ،عمل العمليه ف i بقت ب2 ومازال الشرط متحقق فتمت الطباعة للمرة الثالثة ،عمل العمليه ف i بقت ب3 ومازال الشرط متحقق فتمت الطباعة للمرة الرابعة ،عمل العمليه ف i بقت ب4 ومازال الشرط متحقق فتمت الطباعة للمرة الخامسة ، عمل العملية ف i اصبحت ب 5 فالشرط لم يعد متحقق فخرج برا ال for loop .

*Ex:

```
#include <stdio.h>
int n1 = 0;
int n2 = 0;
int result = 0;
char operator;
int main()
{
    while (1)
    {
        printf("please enter the operator : ");
        scanf("%c", &operator);
        printf("please enter two numbers: ");
        scanf("%i %i", &n1, &n2);

        switch (operator)
        {
            case '+':
                result = n1 + n2;
                printf("The result is %i", result);
                break;
            case '-':
                result = n1 - n2;
                printf("The result is %i \n", result);
                if (n1 < n2)
                {
                    printf("number1 is the smallest(%i)", n1);
                }
                break;
            case '*':
                result = n1 * n2;
                printf("The result is %i", result);
                break;
            case '!':
                result=1;
                for (int i = n1; i > 0; i = i - 1)
                {
                    result = result * i;
                }
                printf("The result is %i", result);
                break;
        }
        break;
    }

    return 0;
}
```

```
please enter the operator : +
please enter two numbers: 3 4
The result is 7
```

```
please enter the operator : -
please enter two numbers: 2 3
The result is -1
number1 is the smallest(2)
```

```
please enter the operator : *
please enter two numbers: 2 4
The result is 8
```

```
please enter the operator : !
please enter two numbers: 5
0
The result is 120
```

Thank you!

