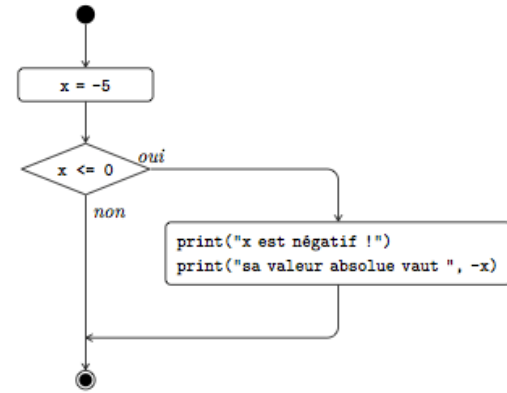


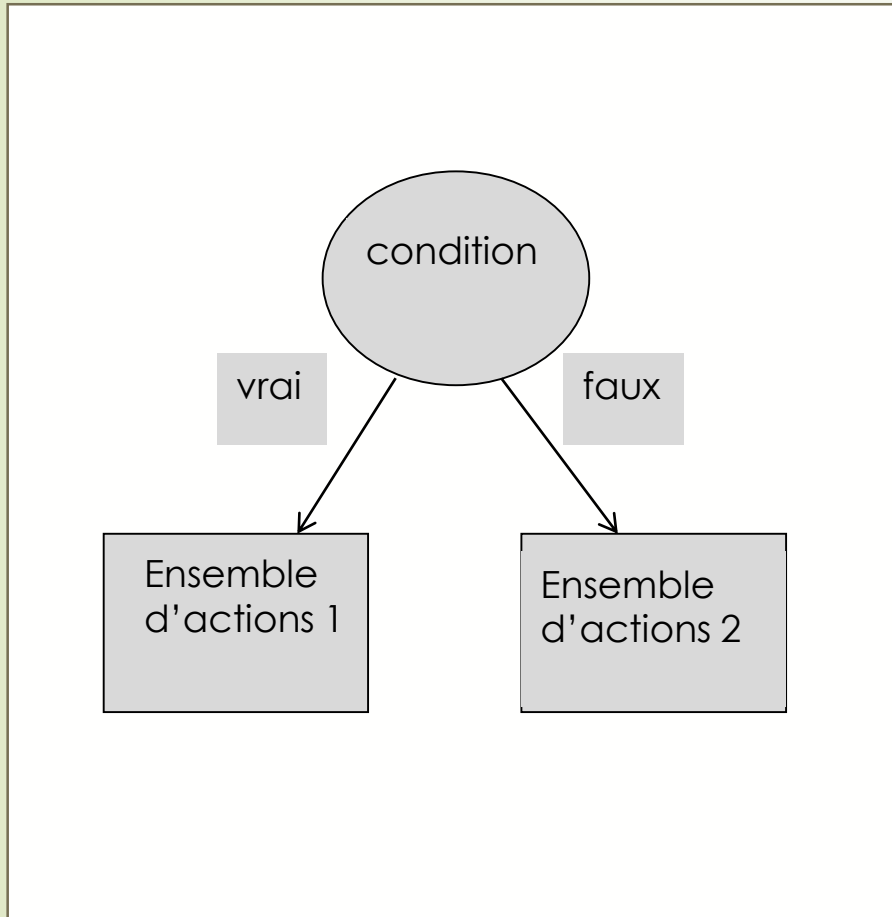
# Structure de contrôles Conditionnelles et boucles

```
PS C:\Users\Florian> $i = 0
$j = 5
Do{
    $i++
    $j++
    Write-Host "i = $i et j = $j"
}Until(($i -eq 10) -or ($j -eq 10))
Write-Host "Une des deux variables est arrivée à 10 donc on sort de la boucle !"
i = 1 et j = 6
i = 2 et j = 7
i = 3 et j = 8
i = 4 et j = 9
i = 5 et j = 10
Une des deux variables est arrivée à 10 donc on sort de la boucle !
```



# Structures de contrôles

- Un programme est une suite d'instructions qui décrit les actions à effectuer pour résoudre un problème. L'exécution du programme consiste à exécuter les instructions les unes après les autres dans l'ordre où elles apparaissent dans le programme.
- Il existe des instructions qui ont pour but d'enrichir ce schéma d'exécution en permettant de ne pas exécuter certaines instructions ou au contraire de les exécuter plusieurs fois : **ce sont les structures de contrôles..**
  - La conditionnelle : introduit un aiguillage, un choix entre des ensembles de instructions
  - La boucle : introduit une itération, un ensemble de instructions est exécuté plusieurs fois.



**La  
conditionnelle :  
choisir une  
action en  
fonction d'une  
condition  
vérifiée ou pas**

# Un exemple

Nom élève	Html	Javascript
christophe	15.5	11
Meryem	5	9

On écrit un programme qui :

- Demande les notes de HTML et de Javascript pour Christophe
- Calcule la moyenne et l'affiche
- Si la moyenne  $\geq 10$  affiche « c'est bien » sinon affiche « c'est pas terrible ! »
- Demande les notes de HTML et de Javascript pour Meryem
- Calcule la moyenne et l'affiche
- Si la moyenne  $\geq 10$  affiche « c'est bien » sinon affiche « c'est pas terrible ! »

# Un exemple : algorithme

# La conditionnelle

La conditionnelle est une structure de contrôle qui permet de lier l'exécution d'une séquence de commandes au résultat d'un test. Le test est appelé **condition** ou **prédicat**.

Si (**condition**)

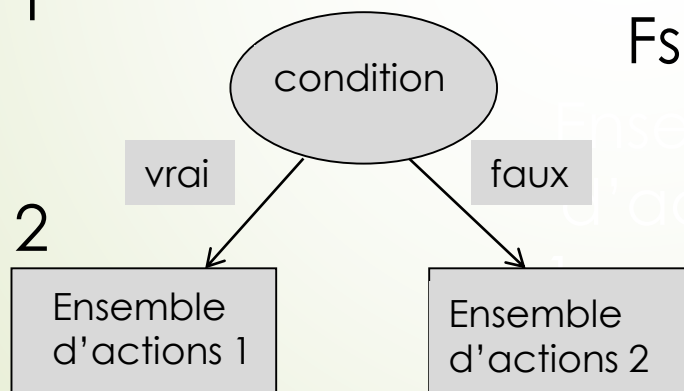
Alors

Ensemble  
d'actions 1

Sinon

Ensemble  
d'actions 2

Fsi

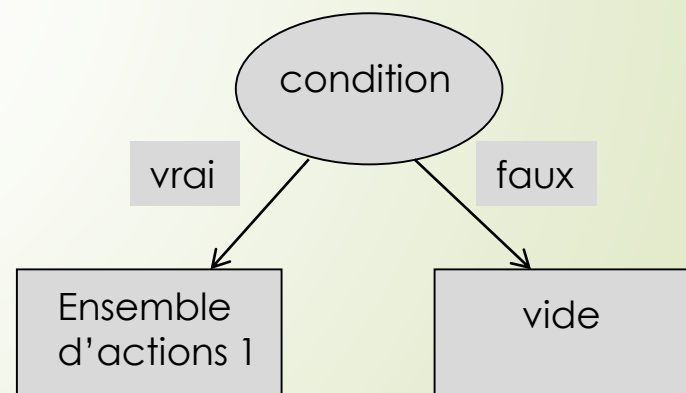


Si (**condition**)

Alors

Ensemble  
d'actions 1

Fsi



# La conditionnelle

Si (condition)  
Alors  
    ensemble  
    d'actions 1  
Sinon  
    ensemble  
    d'actions 2  
Fsi

## Syntaxe en C

```
if (condition)
{
    ensemble d'instructions
}
else
{
    ensemble d'instructions
}
```

# Condition : opérateurs de relation

<, <=	Inférieur Inférieur ou égal	<code>int a = 10; int b = 6;</code> <code>a &lt; b; a &lt;= b</code>
>, >=	Supérieur Supérieur ou égal	<code>a &gt; b; a &gt;= b</code>
==	égalité	<code>a == b;</code>
!=	différent	<code>a != b</code>

Attention : ne pas confondre test d'égalité et affectation !



```
1 #include <stdio.h>
2 #include "util.h"
3
4
5 main() {
6     float notehtml, noteJS;
7     float moyenne;
8     char chaine[20];
9
10
11     affiche_message("Donnez la note de html de Christophe\n ");
12     notehtml = lire_flottant();
13     affiche_message("Donnez la note de JS de Christophe\n ");
14     noteJS = lire_flottant();
15     moyenne = (notehtml + noteJS)/2.0;
16     affiche_message("Voici la moyenne de Christophe\n");
17     affiche_flottant_sautligne(moyenne);
18     if (moyenne >= 10.0)
19     {
20         affiche_message("c'est bien\n ");
21     }
22     else
23     {
24         affiche_message("c'est pas terrible\n ");
25     }
26     affiche_message("Donnez la note de html de Meryem\n ");
27     notehtml = lire_flottant();
28     affiche_message("Donnez la note de JS de Meryem\n ");
29     noteJS = lire_flottant();
30     moyenne = (notehtml + noteJS)/2.0;
31     affiche_message("Voici la moyenne de Meryem\n");
32     affiche_flottant_sautligne(moyenne);
33     if (moyenne >= 10.0)
34     {
35         affiche_message("c'est bien\n ");
36     }
37     else
38     {
39         affiche_message("c'est pas terrible\n ");
40     }
41 }
42
```

# Des exercices

- Ecrivez un programme qui demande une somme en francs, la convertit en euros ou en dollar selon le souhait de l'utilisateur puis il affiche le résultat
- Soient deux variables de type entier A et B. Vous écrivez un programme qui réalise les opérations suivantes :
  - Demandez à l'utilisateur une valeur et l'affecter à A ;
  - Demandez à l'utilisateur une valeur et l'affecter à B ;
  - Affichez la plus petite des deux valeurs.
- Ce sont les soldes. Le programme calcule la remise et le prix final d'un article et affichent ces deux valeurs.

Si le prix de l'article soldé est supérieur à 100 euros, la réduction est de 40 % sinon la remise est de 20%.

- Écrire un programme qui calcule le tarif d'entrée d'une piscine sachant que les mineurs paient 2 euros et les majeurs 4. Le programme doit demander à l'utilisateur son âge et afficher le tarif correspondant.

# Des exercices

- Ce sont les soldes. Le programme calcule la remise et le prix final d'un article et affichent ces deux valeurs.

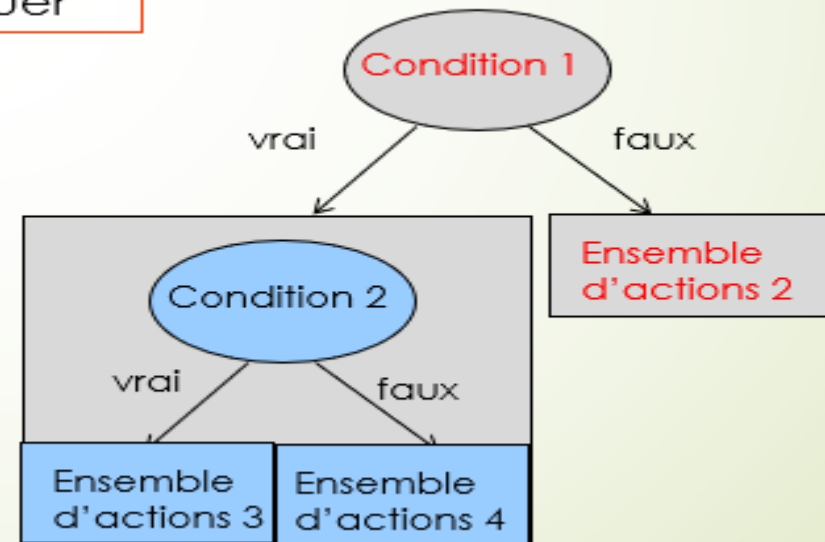
Si le prix de l'article soldé est supérieur à 100 euros, la réduction est de 40 %.

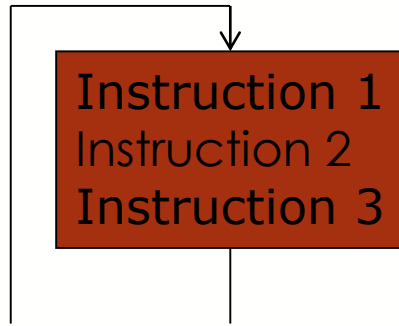
Si le prix de l'article soldé est inférieur à 50 euros, la réduction est de 10 %.

Si le prix de l'article est compris entre 50 et 100 euros, la réduction est de 20 %.

Les conditionnelles peuvent s'imbriquer

```
Si (condition1)
Alors
  Si (condition2)
  Alors
    Ensemble d'actions 3
  Sinon
    Ensemble d'actions 4
Fsi
Sinon
  Ensemble d'actions 2
Fsi
```





## L'iteration : répéter plusieurs fois un ensemble d'actions

Dans la majorité des programmes, il y a des tâches répétitives. Les langages de programmation nous offrent différentes instructions pour nous faciliter ce travail. Ce sont les instructions de boucles.

```
1 #include <stdio.h>
2 #include "util.h"
3
4
5 main() {
6     float notehtml, noteJS;
7     float moyenne;
8     char chaine[20];
9
10
11     affiche_message("Donnez la note de html de Christophe\n ");
12     notehtml = lire_flottant();
13     affiche_message("Donnez la note de JS de Christophe\n ");
14     noteJS = lire_flottant();
15     moyenne = (notehtml + noteJS)/2.0;
16     affiche_message("Voici la moyenne de Christophe\n");
17     affiche_flottant_sautligne(moyenne);
18     if (moyenne >= 10.0)
19     {
20         affiche_message("c'est bien\n ");
21     }
22     else
23     {
24         affiche_message("c'est pas terrible\n ");
25     }
26     affiche_message("Donnez la note de html de Mervem\n ");
27     notehtml = lire_flottant();
28     affiche_message("Donnez la note de JS de Mervem\n ");
29     noteJS = lire_flottant();
30     moyenne = (notehtml + noteJS)/2.0;
31     affiche_message("Voici la moyenne de Mervem\n");
32     affiche_flottant_sautligne(moyenne);
33     if (moyenne >= 10.0)
34     {
35         affiche_message("c'est bien\n ");
36     }
37     else
38     {
39         affiche_message("c'est pas terrible\n ");
40     }
41 }
42
```

1

2

# Un exemple : algorithme

# La boucle for

Boucle for : les instructions sont itérées en traitant la liste de valeurs fournies :

- val 1 à itération 1
- val 2 à itération 2
- val n à itération n

```
for (liste de valeurs val1,  
    val2, ... val n)  
{  
/* Instructions à répéter */  
}
```

```
i = 0; j = 1;  
for (i=0; i < 3; i =i+1)  
{  
    1      2      3  
j = i + 1;  
i = i + 1;  
}
```

1. initialisation
2. condition
3. itération

```
1 #include <stdio.h>
2 #include "util.h"
3
4
5 main(){
6     float notehtml, noteJS, moyenne;
7     int nbeleve, tour;
8     char nom[20];
9
10    affiche_message("combien d'élèves ? ?\n ");
11    lire_entier(nbeleve);
12    for (tour=0; tour < 2; tour= tour + 1)
13    {
14        affiche_message("quel est le nom de l'élève ?\n ");
15        lire_chaine(nom);
16
17        affiche_message("Donnez la note de html de ");
18        affiche_chaine_sautdeligne(nom);
19        notehtml = lire_flottant();
20        affiche_message("Donnez la note de JS de ");
21        affiche_chaine_sautdeligne(nom);
22        noteJS = lire_flottant();
23        moyenne = (notehtml + noteJS)/2.0;
24        affiche_message("Voici la moyenne de ");
25        affiche_chaine_sautdeligne(nom);
26        affiche_flottant_sautligne(moyenne);
27        if (moyenne >= 10.0)
28        {
29            affiche_message("c'est bien\n ");
30        }
31        else
32        {
33            affiche_message("c'est pas terrible\n ");
34        }
35    }
36 }
```