

# Fiche 2 : structures et boucles conditionnelles

## 1 Structure conditionnelle

### 1.1 Une seule condition

On veut exécuter  $n$  instructions sous une certaine condition  $C$ , et si cette condition n'est pas vérifiée, on exécute alors une autre série d'instructions.

Si la condition $C$ est vérifiée, alors :	<code>if C :</code>
Instruction 1	Instruction 1
Instruction 2	Instruction 2
.	.
.	.
Instruction $n$	Instruction $n$
Sinon :	<code>else :</code>
Instruction 1'	Instruction 1'
Instruction 2'	Instruction 2'
.	.
.	.
Instruction $m'$	Instruction $m'$

#### Exemple : affichage du maximum entre deux nombres

```
if x<y :
    print(y)
else :
    print(x)
```

Si  $x = 3$  et  $y = 5$ , alors l'interpréteur affiche 5.

### 1.2 Plusieurs conditions s'excluant mutuellement

On suppose dans ce contexte qu'on a une famille de conditions  $C_1, \dots, C_n$ , et une famille d'instructions  $I_1, \dots, I_n$ . On cherche à exécuter la famille d'instructions  $I_j$  dans le cas où  $C_j$  est vérifiée, mais aucune des conditions  $C_i$  avec  $i < j$  ne l'est.

Si la condition $C_1$ est vérifiée, alors :	<code>if C<sub>1</sub> :</code>
Instructions $I_1$	Instructions $I_1$
Sinon, si $C_2$ est vérifiée, alors :	<code>elif C<sub>2</sub> :</code>
Instructions $I_2$	Instructions $I_2$
.	.
Sinon, si $C_n$ est vérifiée, alors :	<code>elif C<sub>n</sub> :</code>
Instructions $I_n$	Instruction $I_n$
Sinon :	<code>else :</code>
Instructions ...	Instructions ...

### Exemple : trinôme du second degré

```
d=b**2-4*a*c
if d>0 :
    print("deux solutions réelles")
elif d<0 :
    print("pas de solution réelle")
else :
    print("une solution réelle")
```

Si  $a = 1$ ,  $b = 1$ , et  $c = -1$ , alors l'interpréteur affichera ‘deux solutions réelles’.

## 2 Boucle while

On cherche à répéter une série d'instructions  $I$  tant qu'une certaine condition  $C$  est vérifiée.

tant qu'une condition $C$ est vérifiée, répéter :	<code>while C :</code>
Instructions $I$	Instructions $I$

### Exemple : calcul du reste d'une division

```
while (r>=n) :
    r=r-n
print(r)
```

Si  $r = 18$  et  $n = 7$ , à l'issue de la boucle `while`, l'interpréteur affiche 4.

**Attention!** Si les instructions sont mal construites, on peut ne jamais sortir de la boucle, et avoir une boucle infinie, problème informatique courant.