

Instructions__de__controle

November 28, 2020

1 Instructions de Contrôle

1.1 Notes de cours

1.1.1 Réaliser les commandes suivantes sous Python

```
a= float(input('donner a'))
b= float(input('donner b'))
c=a/b
print(c)
```

1.1.2 Remarque

On remarque que python déclare une erreur **division par 0** , si on donne 0 comme valeur de b, la question qui se pose est comment on peut contrôler l'exécution avant de faire **a/b** on doit s'assurer que **b** est non nulle.

1.2 Instructions de contrôle

- Une *instruction de contrôle* permet de contrôler l'exécution de quelque block d'instructions.

1.2.1 Algorithme

```
si cdt alors
    block A
sinon
    block B
finsi
```

1.2.2 Python

```
if cdt:
    block A
else:
    block B
```

1.2.3 Déroulement

- le block A sera exécuté si la condition **cdt** est vrai si non le block b qui sera exécuté.
- donc il faut savoir comment écrire une condition sous *Python*

1.3 Expression logique

1.3.1 Une expression logique:

- Est une expression qui peut être:
 - vrai sous python **True**
 - faux sous python **False**

1.3.2 Les principaux tests avec Python sont:

Opérateur	Syntaxe
égalité	==
différence	!=
strictement inférieur	<
strictement supérieur	>
inférieur ou égale	<=
supérieur ou égale	>=

Aussi on utilise les opérateurs logiques:

Opérateur	Syntaxe
et	and
ou	or
non	not

1.4 Exemple:

- Un programme qui calcule la valeur absolue d'un nombre réel

1.4.1 Algorithme

```
variable  x,v: réel
début
    écrire("donner x")
    lire(x)
    si x>=0 alors
        v=x
    sinon
        v=-x
    finsi
    écrire(v)
fin
```

1.4.2 Python

```
x= float(input("donner x"))
if x >= 0:
```

```

    v = x
else:
    v = -x
print(v)

```

1.5 Exemple 2

- Programme qui affiche le maximum de deux nombres x et y. (python dispose d'une fonction qui calcule le maximum **max(x,y)**).

1.5.1 Solution

```

x= float(input('donner un nombre x'))
y= float(input('donner un nombre y'))
if x>=y :
    print(x)
else:
    print(y)

```

2 Exemples:

2.1 Cas1: une seule variable avec deux résultats:

- Dans ce cas on a une seule variable avec deux résultats. pour réaliser ce traitement on ajoute une seule instruction if avec else
- on commence avec la condition qui donne un cas et l'autre cas sera dans le bloc sinon (else)

2.1.1 version 1

```

if cdt1:
    Résultat 1
else :
    Résultat 2

```

2.1.2 version 2

```

if cdt2:
    Résultat 2
else :
    Résultat 1

```

2.1.3 Exemple:Moyenne

Un programme qui permet de calculer la moyenne de 3 notes d'un étudiant, en suite il affiche ou selon les deux cas suivants:

- **validé** si moyenne ≥ 10
- **non valide** si moyenne < 10

2.1.4 Solution

```
note1 = float(input("donner note1"))
note2 = float(input("donner note2"))
note3 = float(input("donner note3"))
somme = note1+note2+note3
moyenne = somme/3
# on compare la mmoyenne avec 10
if moyenne >= 10:
    print("valide")
else:
    print("non valide")
```

2.2 Cas2: une seule variable avec plusieurs résultats

- Pour résoudre ce cas, on réalise un classement des résultats selon la variable d'entrée.

2.2.1 Exemple:

- si $var > b1$ alors res1
- si $var > b2$ et $var \leq b1$ alors res2
- si $var > b3$ et $var \leq b2$ alors res3
- si $var \leq b3$ alors res4

2.2.2 Version 1

```
if var > b1:
    res1
else:
    if var > b2:
        res2
    else:
        if var > b3:
            res3
        else:
            res4
```

2.2.3 Version 2

```
if var > b1:
    res1
elif var > b2:
    res2
elif var > b3:
    res3
else:
    res4
```

2.3 Cas2 Exemple : Appréciations des étudiants

Les enseignants affectent des appréciations aux étudiants selon leurs moyennes:

- 'Très bien', 'Bien', 'Passable', 'Rattrapage' ou 'Non valider'.

2.3.1 Exemple :

Donner un programme Python qui calcule la moyenne de 3 notes et qui affiche les messages suivant (il faut utiliser des si imbriqués).

- Si moyenne ≥ 16 on affiche **Très bien**
- Si moyenne ≥ 12 et moyen < 16 **Bien**
- Si moyenne ≥ 10 et moyen < 12 **passable**
- Si moyenne ≥ 7 et moyen < 10 **rattrapage**
- Si moyenne < 7 **Non valider**

2.4 Exemple: solution

```
note1 = float(input("donner note1"))
note2 = float(input("donner note2"))
note3 = float(input("donner note3"))
moyenne = (note1+note2+note3)/3
if moyenne >= 16 :
    print('Très bien')
else:
    if moyenne >= 12 :
        print('Bien')
    else:
        if moyenne >= 10 :
            print('Passable')
        else:
            if moyenne >= 7 :
                print('rattrapage')
            else: print('Non valide')
```

2.5 Exemple solution avec if elif else

2.5.1

```
note1 = float(input("donner note1"))
note2 = float(input("donner note2"))
note3 = float(input("donner note3"))
moyenne = (note1+note2+note3)/3
if moyenne >= 16 :
    print('Très bien')
elif moyenne >= 12 :
    print('Bien')
elif moyenne >= 10 :
    print('Passable')
elif moyenne >= 7 :
```

```

    print('rattrapage')
else:
    print('Non valide')

```

2.6 Cas3: plusieurs variables d'entrées avec deux résultats

- si cdt1 et cdt2 ou cdt 2 alors **res1**
- sinon **res2**

Dans ce cas on utilise les opérateurs logiques **and**, **or** et **not** pour faciliter l'expression de la condition **

2.6.1 Syntaxe:

```

if cdt1 and cdt2 or cdt3 :
    res1
else:
    res2

```

2.7 Cas 3: Exemple

Un programme qui demande 3 notes et qui calcule la moyenne. puis il affiche:

- **Valide** Si moyenne ≥ 10 et toutes les notes sont supérieures ou égale à 7
- **non valide** sinon

2.8 Cas 3: Solution

```

note1 = float(input("donner note1"))
note2 = float(input("donner note2"))
note3 = float(input("donner note3"))
somme = note1+note2+note3
moyenne = somme/3
if moyenne >=10 and note1>=7 and note2>=7 and note3>=7:
    print("validé")
else:
    print("non valide")

```

2.9 Cas 4: plusieurs variables d'entrées et plusieurs résultats

Pas de solution extraordinaire, mais on peut procéder de la façon suivante:

- On examine les différents cas
- On choisit la variable la plus dominante c-à-d qui figure dans tous les cas
- On traite le cas de cette variable
- Après on passe à la variable suivante et ainsi de suite.

2.10 Syntaxe python

```
if cdtDominante1 :
    if cdtDominante2 :
        bloc #si cdt1 et cdt2 sont vrai
    else:
        bloc #si cdt1 est vrai et cdt2 est fausse
else:
    if cdtDominante3 :
        bloc #si cdt1 est fausse et cdt3 est vrai
    else:
        bloc #si cdt1 est fausse et cdt3 est fausse
```

2.11 Cas 4:

2.11.1 Exemple

Pour réaliser le suivi des étudiants, l'enseignant a besoin d'un programme qui permet d'afficher des messages selon les cas suivants :

- Si $\text{moyenne} \geq 10$ et $\text{note1} \leq \text{note2}$ et $\text{note2} \leq \text{note3}$ alors 'valider avec progrès'
- Si $\text{moyenne} \geq 10$ et $\text{note1} > \text{note2}$ ou $\text{note2} > \text{note3}$ alors 'valider mais avec recule'
- Si $\text{moyenne} < 10$ alors 'non valide'

2.12 Cas 4: exemple

2.12.1 Solution

```
note1 = float(input("donner note1"))
note1 = float(input("donner note1"))
note1 = float(input("donner note1"))
somme = note1+note1+note3
moyenne = somme/3
if moyenne >= 10 :
    if note1 <= note2 and note2 <= note3 :
        print("valider avec progrès")
    else :
        print("valider avec recule")
else:
    print("Non valide")
```