

I- Edupython**A) Calculs**

Compléter le tableau suivant en tapant les instructions au fur et à mesure.

>>>	5+3	2-9	7+3*4	(7+3)*4	3**2	20/3	20.5/3	20,5/3	20//3	20%3
résultat	8	-7	19	40	9	6.666	6.833	(20,1.66)	6	2

En effet, pour savoir si :

- un nombre entier a est pair, il suffira de voir si le reste $a\%2$ vaut 0.
- un nombre entier a est impair, il suffira de voir si le reste $a\%2$ vaut 1.

Explication pour les non matheux : Le reste est toujours strictement inférieur au diviseur donc si on effectue une division euclidienne par 2, le reste sera soit 0 soit 1. Si a est pair, la division par 2 va tomber juste donc le reste est 0.

B) Variables, affectations

Tapez les trois instructions ci-contre.

- 1) n , msg et pi sont des **variables**.
- 2) Que signifie le signe égal ici ?

C'est une affectation qui signifie « prend la valeur »

```
>>> n=7
>>> msg="Quoi de neuf ?"
>>> pi=3.14159
```

>>>	n	msg	pi
Résultat avec la fonction print()	7	Quoi de neuf ?	3.14159
Résultat avec la fonction type()	<class 'int'>	<class 'str'>	<class 'float'>

Tester les deux instructions suivantes :

```
>>> point_de_vie=15
>>> print(point_de_vie)
```

Qu'affiche Python ? 15

Écrire à présent l'instruction suivante :

```
>>> print("point_de_vie")
```

Qu'affiche Python ? point_de_vie

```
>>> print("point_de_vie =",point_de_vie)
```

Qu'affiche Python ? point_de_vie = 15

C) La fonction input()

Tester l'instruction ci-dessous. La variable *nom* va contenir la réponse entrée au clavier par l'utilisateur.

```
>>> nom=input("Quel est votre nom ?")
Quel est votre nom ?
```

Afin de vérifier que la variable *nom* contient bien ce qui a été rentré au clavier, écrire l'instruction :

```
>>> print(nom)
```

Qu'affiche Python ? Eric (c'est le nom qu'a rentré l'élève au clavier)

Attention : si l'élève n'obtient rien après l'instruction « print(nom) » c'est qu'il n'a pas compris qu'il fallait saisir le nom ! Vous verrez, ça m'est arrivé assez souvent ☹

Cette fois on souhaite rentrer un entier. Tester l'instruction ci-dessous en tapant au clavier un entier.

```
>>> age=input("Quel est votre âge ?")
Quel est votre âge ?
```

Afin de vérifier que la variable *age* contient bien ce qui a été rentré au clavier, écrire l'instruction :

```
>>> print(age)
```

Qu'affiche Python ? 15 (c'est ce qu'a rentré l'élève au clavier)

Maintenant, tester l'instruction :

```
>>> print(age+age)
```

Qu'affiche Python ? 1515 (si l'âge rentré était 15)

Pour comprendre ce qu'il s'est passé, taper l'instruction :

```
>>> type(age)
```

Qu'affiche Python ? <class 'str'>

```
>>> a="Une conca"
>>> b="téna"
>>> c="tion"
>>> print(a+b+c)
```

Qu'affiche Python ?
Une concaténation

```
>>> age=int(input("Quel est votre âge ?"))
Quel est votre âge ?
```

Puis tester les instructions suivantes :

```
>>> type(age)
```

 Qu'affiche Python ? 15

Puis tester les instructions suivantes :

```
>>> print(age+age)
```

 Qu'affiche Python ? 30

II- Expressions booléennes (vrai/faux)

Tester l'instruction suivante :

```
>>> print(4==5)
```

 Qu'affiche Python ? False

Tester maintenant :

```
>>> print(7==7)
```

 Qu'affiche Python ? True

Attention la suite est à faire dans la zone de script (en haut à droite):

Écrire donc ce programme puis l'exécuter.

<pre>a=4 b=7 print(a==b) a=7 print(a==b)</pre>	Affichage après exécution : False True
--	--

Modifier le programme précédent et le tester.

<pre>a=4 b=7 print(a!=b) a=7 print(a!=b)</pre>	L'opérateur « != » signifie « différent de ». Affichage après exécution : True False
--	--

Modifier le programme précédent et le tester.

<pre>a=4 b=7 print(a<b) a=7 print(a<b)</pre>	L'opérateur « < » signifie « strictement inférieur à ». Affichage après exécution : True False
--	---

Modifier le programme précédent et le tester.

<pre>a=4 b=7 print(a<=b) a=7 print(a<=b)</pre>	L'opérateur « <= » signifie « inférieur ou égal à ». Affichage après exécution : True True
--	---

III- Instructions conditionnelles

Exercice 1 : Répondre aux questions suivantes sans taper le programme ! Après avoir répondu aux questions, on tapera le programme et on testera les réponses.

1) Selon vous, que va afficher le programme suivant ?

<pre>note=int(input("rentrez une note:")) if note<8: print("ajourné") else: if note<10: print("repêchage") else: print("reçu")</pre>	si note=7 : ajourné. si note=8 : repêchage si note=9 : repêchage si note=10 : reçu si note=11 : reçu
--	---

Attention : beaucoup d'élèves n'alignent pas correctement les if et les else, et d'autres oublient de fermer la 2^{ème} parenthèses.

<pre>a=5 b=2 if a<b: print(a) print("a")</pre>	Que va afficher le programme suivant selon-vous ? Vérifier votre réponse en l'exécutant : Rien ne s'affiche
---	--

<pre>a=5 b=2 if a<b: print(a) print("a")</pre>	Que va afficher le programme suivant selon-vous ? Vérifier votre réponse en l'exécutant : a
---	--

Explication : dans le 1^{er} cas, rien ne s'affiche car a n'est pas strictement inférieur à b, donc Python n'exécute pas les instructions qui se trouvent dans le bloc du « if ». Dans le 2^{ème} cas, il exécute la dernière instruction car elle n'est pas dans le bloc du « if ».

Exercice 2 : créer un programme qui demande à l'utilisateur deux valeurs entières a et b, et qui donne la plus petite de ces deux valeurs. Par exemple si a=3 et b=2, votre programme devra afficher « La valeur la plus petite est 2 » et non pas « La valeur la plus petite est b ».

Tester bien votre programme en prenant également des nombres égaux.

```

a=int(input("a= "))
b=int(input("b= "))
if (a<b):
    print("La plus petite valeur est a")
else:
    print("La plus petite valeur est b")

```

Exercice 3 : créer un programme qui demande à l'utilisateur trois valeurs décimales x , y , z et qui affiche la plus grande valeur (on pourra utiliser une variable max qui prendra la plus grande valeur entre x et y , puis il suffira de comparer max à z). Par exemple si $x=3$, $y=2.5$ et $z=7$, votre programme devra afficher « La valeur la plus grande est 7.0 ».

Tester bien votre programme en prenant également des nombres égaux.

```

x=float(input("x= "))
y=float(input("y= "))
z=float(input("z= "))
if (x<y):
    max=y
else:
    max=x
if (max<z):
    max=z
print("La plus grande valeur est",max)

```

Exercice 4 : créer un programme qui demande à l'utilisateur deux valeurs décimales a et b et qui affiche les solutions de l'équation $ax + b = 0$.

```

a=float(input("a= "))
b=float(input("b= "))
if (a==0):
    if (b==0):
        print("Tous les réels sont solutions de l'équation")
    else:
        print("L'équation n'a aucune solution")
else:
    print("La solution est",-b/a)

```