

---

# IngeSUP - TD 01 - Opérateurs, variables et entrées sorties

---

*“L'homme doit savoir que son destin se trouve au bout du tunnel et non à son entrée.”*

Mokhtar Reguieg

---

## Exercice 01.1

---

1. Qu'est-ce qu'un mot-clé ? En connaissez-vous ?
2. Qu'est-ce qu'un identificateur ?
3. Listez quelques caractères spéciaux ? A quoi servent-ils ?
4. Combien de types peuvent être attribués à une variable? Lesquels ?

### Réponse:

1. Un mot-clé est un mot détecté par Python, qui sert à déclencher des fonctions interne au langage. Par exemple print, type, and, or, input, int, float, str...
2. Un identificateur est un mot clé qui sert au typage d'une variable : int, float, bool, str
3. +/\*-%//, ils servent à faire des opérations
4. On en connaît 4 pour l'instant : int, float, bool, str

---

## Exercice 01.2

---

Quel est le type approprié pour les variables suivantes :

1. Une variable qui contient le prénom de l'utilisateur ;
2. Une variable qui contient la distance entre deux villes ;
3. Une variable qui stocke au fur et à mesure le plus grand des nombres qu'un utilisateur tape au clavier ;
4. Une variable qui permet de traiter différemment un nombre selon qu'il est pair ou impair.

### Réponse:

1. str
2. float (ou int si le nombre est entier)
3. float (ou int si le nombre est entier)
4. bool pour la variable qui permet de traiter un nombre selon... mais int pour le nombre en question

---

## Exercice 01.3

---

1. Testez toutes les commandes suivantes, notez le résultat obtenu si le résultat diffère de celui que vous espériez, essayez d'en trouver la raison puis en prendre note :

$20+1$  ;  $20/3$  ;  $20\%3$  ;  $5.45*10$  ;  $2**5$  ;  $(3+2)*5$  ;  $3+2*5$  ; "bonjour" ; 'bonjour' ; "bonjour" \*5

Entrée [19]:



$20+1$

Out[19]:

21

Entrée [20]:



$20/3$

Out[20]:

6.666666666666667

Entrée [21]:



$20\%3$

Out[21]:

2

Entrée [22]:



$5.45*10$

Out[22]:

54.5

Entrée [23]:



```
2**5
```

Out[23]:

32

Entrée [24]:



```
(3+2)*5
```

Out[24]:

25

Entrée [25]:



```
3+2*5
```

Out[25]:

13

Entrée [26]:



```
"bonjour"
```

Out[26]:

```
'bonjour'
```

Entrée [27]:



```
'bonjour'
```

Out[27]:

```
'bonjour'
```

Entrée [28]:



```
"bonjour"*5
```

Out[28]:

```
'bonjourbonjourbonjourbonjourbonjour'
```

2. Certaines instructions ne sont pas acceptées par l'interpréteur python et provoquent l'affichage d'un message d'erreur. Il faut lire attentivement tous les messages d'erreur (même s'ils sont en anglais) car ces messages nous apprennent comment fonctionne le langage et nous aident à corriger nos instructions et nos programmes. Tapez les instructions suivantes, observez les erreurs obtenues et analysez-les :

20/0 ; 20@3; "bonjour"/3 ; "bonjour" + 5 ; (3+2))\*5 ; (3+2\*5

Entrée [ ]:



# VOTRE CODE ICI

3. On peut parfois transformer une donnée d'un certain type en une donnée d'un autre type. Cela s'appelle le **transtypage** (ou **cast** en anglais). Transtyper le résultat et observez le type de la variable en sortie et sa valeur :

Donnée de départ	De quel type est la donnée de départ?	Instruction à taper	Type du résultat obtenu et/ou commentaire	Valeur du résultat
3.0	float	int(3.0)	int	3
3.5	float	int(3.5)	int	3
3	int	float(3)	float	3.0
4	int	str(4)	str	"4"
"3"	str	int("3")	int	3
"3.5"	str	float("3.5")	float	3.5
"3.5"	str	int("3.5")	ERREUR	
"bonjour"	str	int("bonjour")	ERREUR	

Entrée [ ]:



# VOTRE CODE ICI

## Exercice 01.4

On considère l'algorithme ci-dessous écrit en langage Python :

Entrée [14]:



```
a = 2
b = -5
a = a+b
b = a-b
print("Maintenant a= ", a, " et b = ", b)
```

Maintenant a= -3 et b = 2

1. Complétez le tableau.

Ligne   a   b

Ligne	a	b
1	2	
2	2	-5
3	-3	-5
4	-3	2

Maintenant on considère cet autre algo écrit en langage Python :

Entrée [16]:



```
a=5
b=7
a = a+b
b = a-b
a = a-b
```

7 5

2. Complétez de nouveau le tableau.

Ligne	a	b
1	5	
2	5	7
3	12	7
4	12	5
5	7	5

3. Que constatez-vous concernant le contenu des variables a et b ?

**Réponse:**

Les valeurs de a et b ont été inversée

Note : on peut aussi utiliser une affectation double comme ceci :

Entrée [29]:



```
a=3
b=4
a,b=b,a
print(a,b)
```

4 3

## Exercice 01.5

---

Ecrivez un programme qui convertisse en degré un angle fourni au départ en radians.

Entrée [18]:



```
angleEnRadian=float(input("entrez un angle en radian"))
angleEnDegre=angleEnRadian*180/3.14

print("votre angle en degré vaut", angleEnDegre, "°")
```

```
entrez un angle en radian1
votre angle en degré vaut 57.324840764331206 °
```

---

## Exercice 01.6

---

Ecrivez un programme qui convertisse en degré Celsius une température exprimée initialement en degrés Fahrenheit.

La formule de conversion est la suivante :

$$TF=TC*1.8+32$$

Entrée [30]:



```
tempF=float(input("entrez une température en F"))
tempC=(tempF-32)/1.8

print("cela vaut", tempC, "°C")
```

```
entrez une température en F100
cela vaut 37.77777777777778 °C
```

---

## Exercice 01.7

---

Ecrivez un programme qui convertisse en centimètres une longueur exprimée initialement en pouces.

Entrée [31]:



```
pouces=float(input("entrez une longueur en pouce"))
cm=pouces*2.54

print("cela fait", cm, "cm")
```

entrez une longueur en pouce17

cela fait 43.18 cm

---

## Exercice 01.8

---

Complétez le programme suivant qui convertit un nombre de secondes fournit au départ (qui doit être assez grand) en durée composée d'années, de mois, de jours, d'heures, de minutes et de secondes.

On donne l'exemple ci-dessus pour illustrer ce qui est demandé : Nombre de secondes à convertir :  
10000000000

Cette durée correspond à 317 années de 365 jours, plus 1 mois de 30 jours, 5 jours, 17 heures, 46 minutes et 40 secondes.

Pour des facilités de calcul on ne tiendra pas compte des années bissextiles et on supposera que les mois comportent 30 jours.

Entrée [32]:



```
# Le nombre de secondes est fourni au départ :
nsd = 10000000000

# Nombre de Secondes Par Jour :
nspj = 3600 * 24

# Nombre de Secondes Par An (soit 365 jours) :
nspa = nspj * 365

# Nombre de Secondes Par Mois (soit 30 jours) :
nspm = nspj * 30

# Nombre d'Années contenues dans la durée fournie :
na = nsd // nspa      # division <entière>
nsr = nsd % nspa      # n. de sec. restantes

# Nombre de Mois restants :
nmo = nsr // nspm     # division <entière>
nsr = nsr % nspm      # n. de sec. restantes

# Nombre de jours restants :
nj = nsr // nspj      # division <entière>
nsr = nsr % nspj      # n. de sec. restantes

# Nombre d'heures restantes :
nh = nsr // 3600      # division <entière>
nsr = nsr % 3600      # n. de sec. restantes

# Nombre de minutes restantes :
nmi = nsr // 60       # division <entière>
nsr = nsr % 60        # n. de sec. restantes

print("Nombre de secondes à convertir :", nsd)
print("Cette durée correspond à", na, "années de 365 jours plus", nmo, "mois de 30 jours,".
```

Nombre de secondes à convertir : 10000000000

Cette durée correspond à 317 années de 365 jours plus 1 mois de 30 jours, 5 jours, 17 heures, 46 minutes et 40 secondes.