TP1 Bilan

EXPRESSION	VALEUR	TYPE	COMMENTAIRE
3+2	5	int (entire)	
3.0+2	5.0	float (flottant)	conversion automatique
int(2.0)	2	int	conversion
float(2)	2.0	float	conversion
str(2)	“2”	str (chaîne de caractères)	conversion
3//2	1	int (ici quotient euclidien modulo 2)	
3/2	1.5	float	
7%2	1	int (ici reste modulo 2)	
2==3	False car 2≠3	bool (booléen)	test d’égalité
3 !=2	True car (2≠3)	bool	test de différence
3≤3	True	bool	test d’inégalité
True and False	False	bool	operation booléenne and est vrai que si les 2 opérandes sont vrais
True or False	True	bool	operation booléenne or est fauxi que si les 2 opérandes sont faux
not(True)	False	bool	négation

```
x=2
print(x)
  x=x+1
```
Erreur d’indentation : dans un corps d’instructions les instructions doivent être alignées verticalement :

```
x=2
print(x) (affiche la valeur de la variable x)
x=x+1  (affecte à x la valeur courante de x et lui ajoute 2)
```

```
print(x," ", x+1)
```
Affichage de la valeur de x puis de la chaîne de caractères constituée d’un espace puis la valeur de l’expression x+1

```
y=int(input(' Donner un entier '))
print(y)
y=y+2
```
Erreur car y est du type str et on ne peut lui ajouter un entier
Rectification
y=int(y)
y=y+2 (y change et est du type int)
mais on pourrait aussi faire
y=y+str(2) (y change et est du type str : le + entre 2 chaînes est la concaténation : « a2c »+ «f8 » donne « a2cf8 »)

```
if y==3 :
   print('y=3')
else :
    print(' y est différent de 3 ')
while  y!=3 :
     y=int(input(' Donner un entier '))
print(y)
```
y est la valeur entrée par l'utilisateur et on lui a ajouté 2
L'instruction du while s'exécute si cette valeur est différente de 3 et alors l'utilisateur donne une nouvelle valeur pour y qui est l'instruction du while
Pour sortir du while y doit être égal à 3 et on passe à l'instruction « print(y) »
On sort donc du while soit directement si l'utilisateur a tapé 1 la première fois ou s'il n'a pas tapé 1 la première fois et 3 ensuite
3 sera alors affiché

instruction conditionnelle

$$\begin{array}{l} if\ bool1: \\ \quad \boxed{bloc1} \\ else: \\ \quad \boxed{bloc2} \\ instructionsuivante \end{array}$$

Si bool1 est vraie bloc1 est exécuté et on passe à instructionsuivante, sinon bloc2 est exécuté et on passe à instructionsuivante
Le else n'est pas obligatoire.

boucle conditionnelle

$$\begin{array}{l} while\ bool1: \\ \quad \boxed{bloc1} \\ instructionsuivante \end{array}$$

Si bool1 est vraie bloc1 est exécuté et on recommence : si bool1 est vraie bloc1 est exécuté etc...
Pour passer à instructionsuivante il faut que bool1 soit fausse

boucle itérative

$$\begin{array}{l} for\ i\ in\ ensemble: \\ \quad \boxed{bloc1} \\ instructionsuivante \end{array}$$

i est une variable qui va prendre toutes les valeurs de « ensemble » et qui à chaque fois exécute bloc1.
quand tout est terminé instructionsuivante est exécutée.