

Activité – Vrai/Faux

Exercice 1

a.

• x = 0	A	B	
	$x < 10$	$x > -10$	A and B
	True	True	True

b.

• x = 0	A	B	
	$x < -10$	$x > 10$	A or B
	False	False	False

c.

• x = 0	A	B	
	$x \leq 10$	$x * x \geq 100$	A and B
	True	False	False

• x = 10	A	B	
	$x < 10$	$x > -10$	A and B
	False	True	False

• x = 10	A	B	
	$x < -10$	$x > 10$	A or B
	False	False	False

• x = 10	A	B	
	$x \leq 10$	$x * x \geq 100$	A and B
	True	True	True

• x = -20	A	B	
	$x < 10$	$x > -10$	A and B
	True	False	False

• x = -20	A	B	
	$x < -10$	$x > 10$	A or B
	True	False	True

• x = -20	A	B	
	$x \leq 10$	$x * x \geq 100$	A and B
	True	True	True

d.

• x = 0	A	B	C	D	X	Y	
	$x > -25$	$x < -5$	$x > 5$	$x < 25$	A and B	C and D	X or Y
	True	False	False	True	False	False	False

• x = 10	A	B	C	D	X	Y	
	$x > -25$	$x < -5$	$x > 5$	$x < 25$	A and B	C and D	X or Y
	True	False	True	True	False	True	True

• x = - 20	A	B	C	D	X	Y	
	$x > -25$	$x < -5$	$x > 5$	$x < 25$	A and B	C and D	X or Y
	True	True	False	True	True	False	True

Exercice 2

A	B	not B	A or not B	not(A or not B)	not(A or not B) and B
True	True	False	True	False	False
True	False	True	True	False	False
False	True	False	False	True	True
False	False	True	True	False	False

Les valeurs de A et B sont donc A = False et B = True pour que l'expression est pour valeur True.

Exercice 3

a.	A	B	A and B	not(A and B)	not A	not B	(not A) or (not B)
	True	True	True	False	False	False	False
	True	False	False	True	False	True	True
	False	True	False	True	True	False	True
	False	False	False	True	True	True	True

L'égalité est bien vérifiée : $\text{not}(A \text{ and } B) == (\text{not } A) \text{ or } (\text{not } B)$

b.	A	B	A or B	not(A or B)	not A	not B	(not A) and (not B)
	True	True	True	False	False	False	False
	True	False	True	False	False	True	False
	False	True	True	False	True	False	False
	False	False	False	True	True	True	True

L'égalité est bien vérifiée : $\text{not}(A \text{ or } B) == (\text{not } A) \text{ and } (\text{not } B)$

TP – Introduction au langage Python

Exercice 4

a. $(5 + 2) * 3 + 4 == 25$

b. $4 + 5 * (2 + 3) == 29$

c. $(2 + 3) * (5 + 4) == 45$

Exercice 5

Ecrire un code qui, étant donné deux nombres a et b , calcule leur moyenne.

```
1 #On demande 2 nombres à l'utilisateur et on les enregistre sous format entier (int)
2 a = int(input("Donner un nombre a: "))
3 b = int(input("Donner un nombre a: "))
4 print((a+b)/2)          #On calcule et affiche la moyenne
```

Exercice 6

La commande $a = b$ va effacer la valeur contenue à l'origine dans la variable a pour y placer la valeur contenue dans la variable b . A ce moment-là, les deux variables a et b contiennent la valeur d'origine de b , donc la commande $b = a$ ne changera rien à la valeur contenue dans b (puisque a contient la même valeur maintenant).

Pour corriger le problème, il faut « sauvegarder » la valeur d'origine de a dans une autre variable (qu'on peut nommer $temp$ par exemple) avant de faire $a = b$, puis utiliser $b = temp$ pour mettre la valeur initialement contenue dans a dans la variable b .

TP – Si, sinon si, sinon

Exercice 7

a. La condition $12 * 2 == 24$ étant vraie, le corps du if sera exécuté. Le code affichera "Logique !".

b. La condition $12 * 2 == False$ compare un entier et un booléen, ce qui est toujours faux (le booléen étant le résultat d'une comparaison, il ne peut pas être égal à un entier). Le corps du if n'est pas exécuté : le code n'affiche rien.

c. La condition $(12 * 2 == 23) == False$ étant vraie, le corps du if sera exécuté. Quant à la ligne 3, elle est en dehors de l'instruction conditionnelle et sera donc exécutée que la condition soit vraie ou fausse (elle ne fait pas partie du if). Le code affichera donc "Logique !" puis "Ou pas..".

d. La condition $12 * 2 == 23$ étant fausse, le corps du if ne sera pas exécuté et le code passera à l'instruction conditionnelle suivante, else. Comme ce dernier n'est exécuté que si la condition du if est fausse, le code affichera "Pas logique"

Exercice 8

```
1 T = int(input("Quelle est la température? "))
2 #On utilise if/elif/else
3 if T < -5:
4     print("Il fait très froid")
5 elif T <= -5 and T < 5:
6     print("Il fait froid")
7 elif T <= 5 and T < 15:
8     print("Il fait frais")
9 elif T <= 15 and T < 25:
10    print("Il fait bon")
11 elif T <= 25 and T < 30:
12    print("Il fait un peu chaud")
13 elif T <= 30 and T < 35:
14    print("Il fait chaud")
15 else: #pour toutes les températures non inclus dans les conditions précédentes (donc T >= 35)
16    print("Il fait très chaud")
```

TP – Encore, et encore, et encore...

Exercice 9

a. Le compteur i prendra les valeurs 0, 1, 2, 3, 4, 5, 6, 7, 8 et 9 : le corps de la boucle sera donc exécuté 10 fois. A chaque fois, la valeur de s est modifiée : on lui ajoute la valeur de i .

i	0	1	2	3	4	5	6	7	8	9
S en fin de corps	0	1	3	6	10	15	21	28	36	45

A la fin de l'exécution, on a donc $s = 45$.

b. Le compteur i prendra les valeurs 1, 2, 3, 4 et 5 : le corps de la boucle sera donc exécuté 5 fois. A chaque fois, la valeur de s est réinitialisée à 1 puis on lui multiplie la valeur de i .

i	1	2	3	4	5
S en fin de corps	1	2	3	4	5

A la fin de l'exécution, on a donc $s = 5$.

c. La boucle continue de tourner tant que $s < 20$ est vrai (s est strictement inférieur à 20), donc :

S en début de corps	0	5	10	15	20
Condition remplie ?	True	True	True	True	False
S en fin de corps	5	10	15	20	X

Le corps de la boucle est donc exécuté 4 fois et à la fin de l'exécution, on a $s = 20$.

d. La boucle continue de tourner tant que $s \neq 100$ est vrai (s est différent de 100), donc :

S en début de corps	1	2	4	8	16	32	64	128
Condition remplie ?	True							
S en fin de corps	2	4	8	16	32	64	128	256

Comme s est toujours différent de 100, le corps de la boucle est donc exécuté à l'infini (boucle infinie) et l'exécution ne s'arrête jamais.

Exercice 10

a.

a en début de corps	10	3	1
b en début de corps	3	1	0
Condition remplie ?	True	True	False
r en fin de corps	1	0	X
a en fin de corps	3	1	X
b en fin de corps	1	0	X

A la fin de l'exécution, $a = 1$.

b.

a en début de corps	10	15	10	5
b en début de corps	15	10	5	0
Condition remplie ?	True	True	True	False
r en fin de corps	10	5	0	X
a en fin de corps	15	10	5	X
b en fin de corps	10	5	0	X

A la fin de l'exécution, $a = 5$.

Exercice 11

```

1 | Q = float(input("Quelle est la quantité initiale ? "))
2 | i = 0
3 | while i < 5 :
4 |     Q = Q * 1,33
5 |     i = i + 1
6 | print(Q)
    
```

Exercice 12

a.

```
1 | for k in range(2, 21, 2) :
2 |     print(k)
```

b.

```
1 | k = 2
2 | while k <= 20 :
3 |     print(k)
4 |     k += 2
```

c.

```
1 | for k in range(20, 1, -2) :
2 |     print(k)
```

```
1 | k = 20
2 | while k >= 2 :
3 |     print(k)
4 |     k = k - 2
```

Exercice 13

Note : Il existe plusieurs façons d'écrire cette fonction. Ceci n'est qu'un exemple.

```
1 | N = int(input("Donner un nombre: "))
2 | #Tant que N est positif, on calcule le reste de la division euclidienne de N par 10
3 | #puis on retire ce reste de N puis on divise par 10 pour avoir le chiffre suivant
4 | while N > 0:
5 |     val = N%10
6 |     N = (N - val) // 10
7 |     print(val)
```

Avec ce code, les valeurs de N et val sont les suivantes pour l'exemple $N = 1234$:

N en début de corps	1234	123	12	1	0
Condition remplie ?	True	True	True	True	False
val	4	3	2	1	
N en fin de corps	123	12	1	0	

Exercice 14

Note : Il existe plusieurs façons d'écrire cette fonction. Ceci n'est qu'un exemple.

```
1 | Nmax = int(input("Nombre maximum d'insectes: "))
2 | N = 10
3 | jour = 1
4 | #Tant qu'on est en dessous de la limite, on passe au jour suivant et la population augmente
5 | while N <= Nmax:
6 |     N = 3 * N
7 |     jour += 1
8 | #A la fin de la boucle, N > Nmax, il faut donc afficher le résultat un jour avant
9 | print("jour #",jour - 1, ", populations = ",N/3)
```

TP – Les fonctions

Exercice 15

a. La fonction ne renvoie pas de résultat, donc l'appelle de la fonction renvoie un type None : a aura pour valeur None.

b. Le programme affichera : 42

Exercice 16

a. Le code est incorrect : la fonction est nommée aire_cercle, pas aire_cercle.

b. Le code est incorrect : la variable a n'est ici définie **que** dans la fonction f , elle n'existe pas en dehors de la fonction (et le `print(f(5), a)` est en dehors de la fonction).

c. Le code est correct. Attention cependant : le nom de variable a est utilisé à l'intérieur **et** à l'extérieur de la fonction, mais ce sont deux variables **différentes** (car les variables utilisées dans les fonctions sont des variables locales, qui disparaissent en dehors de la fonction). Le code affichera alors : 10, 100, 200

Exercice 17

Note : Il existe plusieurs façons d'écrire ces fonctions. Ceci n'est qu'un exemple.

```
1 def anterieur(j1,m1,a1,j2,m2,a2):
2     #Précondition: jours entre 1 et 30, mois entre 1 et 12
3     if j1 >= 1 and j1 <= 30 and j2 >= 1 and j2 <= 30 and m1 >= 1 and m1 <= 12 and m2 >= 1 and m2 <= 12:
4         #False si l'année 1 est supérieure à l'année 2
5         if a2 < a1:
6             return False
7         #Sinon, False si, pour la même année, le mois 1 est supérieure au mois 2
8         elif a2 == a1 and m2 < m1:
9             return False
10        #Sinon, False si, pour la même année et le même mois, le jour 1 est supérieure au jour 2
11        elif a2 == a1 and m2 == m1 and j2 < j1:
12            return False
13        #Sinon, False si les deux dates sont identiques
14        elif a2 == a1 and m2 == m1 and j2 == j1:
15            return False
16        #Dans tous les autres cas, la date 1 est antérieure à la date 2
17        else:
18            return True
19
20 def age(j1,m1,a1,j2,m2,a2):
21     #précondition: on ne peut calculer l'âge que si la date de naissance est antérieure à
22     #la date de calcule
23     if anterieur(j1,m1,a1,j2,m2,a2) == True:
24         #Nombre de jours depuis J.-C. pour chaque date
25         date1 = a1*360 + m1*30 + j1
26         date2 = a2*360 + m2*30 + j2
27         #L'age est la différence entre les deux nombres précédents, divisée par 360 pour avoir
28         #le résultat en années.
29         print("age: ", (date2 - date1)/360)
```

Exercice 18

Note : Il existe plusieurs façons d'écrire ces fonctions. Ceci n'est qu'un exemple.

```
1 from random import randint          #Appel de la bibliothèque random
2
3 def Choix(choixA, choixB):
4     #Egalité
5     if choixA == choixB:
6         return 0
7     #Cas où A choisi Pierre
8     elif choixA == 1 and choixB == 2:          #et B choisi Papier
9         return -1
10    elif choixA == 1 and choixB == 3:          #et B choisi Ciseaux
11        return 1
12    #Cas où A choisi Papier
13    elif choixA == 2 and choixB == 3:          #et B choisi Ciseaux
14        return -1
15    elif choixA == 2 and choixB == 1:          #et B choisi Pierre
16        return 1
17    #Cas où A choisi Ciseaux
18    elif choixA == 3 and choixB == 1:          #et B choisi Pierre
19        return -1
20    elif choixA == 3 and choixB == 2:          #et B choisi Papier
21        return 1
```

```

22
23 def Tour():
24     #On joue le tour tant qu'il y a égalité
25     val = 0
26     while val == 0:
27         #Choix aléatoire entre Pierre, Papier et Ciseaux
28         choixA = randint(1,3)
29         choixB = randint(1,3)
30         #Affichage des choix
31         if choixA == 1:
32             if choixB == 1:
33                 print("A: Pierre; B: Pierre")
34             elif choixB == 2:
35                 print("A: Pierre; B: Papier")
36             elif choixB == 3:
37                 print("A: Pierre; B: Ciseaux")
38         elif choixA == 2:
39             if choixB == 1:
40                 print("A: Papier; B: Pierre")
41             elif choixB == 2:
42                 print("A: Papier; B: Papier")
43             elif choixB == 3:
44                 print("A: Papier; B: Ciseaux")
45         elif choixA == 3:
46             if choixB == 1:
47                 print("A: Ciseaux; B: Pierre")
48             elif choixB == 2:
49                 print("A: Ciseaux; B: Papier")
50             elif choixB == 3:
51                 print("A: Ciseaux; B: Ciseaux")
52         val = Choix(choixA, choixB) #On joue le tour
53         if val == 0:
54             print("Egalité! On rejoue le tour")
55     return val
56
57 ##### Programme principal #####
58 scoreA = 0 #On initialise les scores
59 scoreB = 0
60 for tour in range(50): #On joue 50 tours
61     print("\n Tour #",tour+1) #Affichage du numéro de tour ; Le \n permet d'aller à la ligne
62     point = Tour()
63     #Calcul des scores
64     if point == 1:
65         scoreA += 1
66     else:
67         scoreB += 1
68     #Affichage des scores
69     print("Score A = ",scoreA,"; Score B = ",scoreB)
70 #Gagnant à la fin du jeu
71 if scoreA == scoreB:
72     print("\n Egalité!")
73 elif scoreA > scoreB:
74     print("\n Gagnant: A")
75 else:
76     print("\n Gagnant: B")

```

Exercice 19

Note : Il existe plusieurs façons d'écrire ces fonctions. Ceci n'est qu'un exemple.

```
1  from math import *
2
3  def f(x):      #Définition de la fonction f(x) à tester
4      return sin(x)
5
6  def ValeurX(min, max, pas, v):
7      x = min    #première valeur testée
8      #On parcourt les valeurs de x de pas en pas jusqu'à ce que f(x) < v < f(x+pas)
9      while f(x) < v:
10         x += pas
11     return x
12
13 def Estimation(min, max, pasini, v, ordre):
14     pas = pasini
15     #On utilise ValeurX pour des pas de plus en plus petits, jusqu'à ordre
16     while pas >= ordre:
17         x = ValeurX(min, max, pas, v)
18         #A chaque itération, on réduit l'intervalle de recherche et le pas
19         min = f(x)
20         max = f(x)+pas
21         pas = pas/10
22     return x    #Valeur finale
```