



TP1 : exercices



A l'aide de la fiche de synthèse établir l'algorithme des différents exercices ci-dessous puis valider votre programme en langage python.

Exercice 1

Ecrire un programme qui calcule le carré d'un nombre et affiche le carré de ce nombre.

Algorithme pseudo-code	Python
$a \leftarrow 4$ $\text{carre} \leftarrow a^{**2}$ afficher("La valeur de a=",a," au carré est de ",carre)	<pre>a=4 carre=a**2 print("La valeur de a=",a," au carré est de ",carre)</pre>

Exercice 2

Ecrire un programme qui convertit les kilomètres en miles (1 mile = 1,609 km)

Algorithme pseudo-code	Python
$\text{kilometre} \leftarrow 100$ $\text{miles} \leftarrow \text{kilometre} / 1.609$ afficher("La distance de ",kilometre," kilomètre est de ",miles," miles")	<pre>kilometre=100 miles=kilometre/1.609 print("La distance de ",kilometre," kilomètre est de ",miles," miles")</pre>

Exercice 3

Ecrire un programme qui échange la valeur de deux variables.

Exemple : si a = 2 et b = 5, le programme donnera a = 5 et b = 2.

Algorithme pseudo-code	Python
<pre>a ← 2 b ← 3 afficher("Avant échange a=",a," et b=",b) temp ← a a ← b b ← temp afficher("Après échange a=",a," et b=",b)</pre>	<pre>a=2 b=3 print("Avant échange a=",a," et b=",b) temp=a a=b b=temp print("Après échange a=",a," et b=",b)</pre>

Exercice 4

Ecrire un programme qui à partir d'un prix HT d'un article et le taux de TVA, calcul le prix total TTC correspondant. Faire en sorte que des variables apparaissent clairement.

Algorithme pseudo-code	Python
<pre>valeur_tva ← 20 prix_ht ← 40 tva ← prix_ht * valeur_tva / 100; prix_ttc ← prix_ht + tva; afficher("Prix HT du produit: ",prix_ht," €") afficher ("Indice de la TVA en 2015: ",valeur_tva," %") afficher ("Valeur de la TVA: ",tva," €") afficher ("Prix TTC: ",prix_ttc," €")</pre>	<pre>valeur_tva=20 prix_ht=40 tva=prix_ht*valeur_tva/100 prix_ttc=prix_ht+tva print("Prix HT du produit: ",prix_ht," €") print("Indice de la TVA en 2015: ",valeur_tva," %") print("Valeur de la TVA: ",tva," €") print("Prix TTC: ",prix_ttc," €")</pre>

Exercice 5

A partir de deux nombres a et b, écrire un algorithme qui informe l'utilisateur si leur produit est négatif ou positif.

Algorithme pseudo-code	Python
<pre> a←-3 b←-15 r←a*b si r<0 print("Le résultat du produit est négatif") sinon print("Le résultat du produit est positif") finsi </pre>	<pre> a=3 b=-15 r=a*b if r<0: print("Le résultat du produit est négatif") else: print("Le résultat du produit est positif") </pre>

Exercice 6

Ecrire un algorithme, qui à partir d'un âge affiche la catégorie.

Poussin de 6 à 7 ans

Pupille de 8 à 9 ans

Minime de 10 à 11 ans

Cadet de 12 à 15 ans

Hors catégorie après 15 ans

Algorithme pseudo-code	Python
<pre> age←14 si (age>=6) et (age<=7): afficher("Poussin") sinon si (age>=8) et (age<=9): afficher ("Pupille") sinon si (age>=10) et (age<=11) afficher ("Minime") sinon si (age>=12) et (age<=15) afficher ("Cadet") sinon si (age>15) afficher ("Hors catégorie") fin si fin si finsi </pre>	<pre> age=14 if (age>=6) and (age<=7): print("Poussin") elif (age>=8) and (age<=9): print("Pupille") elif (age>=10) and (age<=11): print("Minime") elif (age>=12) and (age<=15): print("Cadet") elif (age>15): print("Hors catégorie") </pre>

Exercice 7

Ecrire un algorithme qui calcule la somme des entiers jusqu'à un nombre nb_valeur.

Par exemple, si le nb_valeur est 5, le programme doit calculer : $1 + 2 + 3 + 4 + 5 = 15$

Remarque : on souhaite afficher uniquement le résultat, pas la décomposition du calcul.

Algorithme pseudo-code	Python
<pre> nb_valeur ← 10 result ← 0 pour n de 1 à nb_valeur par pas de 1 faire result ← result + n finpour afficher("Somme des ",nb_valeur," premiers nombres entiers est: ",result) </pre>	<pre> nb_valeur=10 result=0 for n in range(1,nb_valeur+1): result=result+n print("Somme des ",nb_valeur," premiers nombres entiers est: ",result) </pre>

Exercice 8

Écrire un programme qui affiche la table de Pythagore de la multiplication.

```

0 0 0 0 0 0 0 0 0 0 0
0 1 2 3 4 5 6 7 8 9 10
0 2 4 6 8 10 12 14 16 18 20
0 3 6 9 12 15 18 21 24 27 30
0 4 8 12 16 20 24 28 32 36 40
0 5 10 15 20 25 30 35 40 45 50
0 6 12 18 24 30 36 42 48 54 60
0 7 14 21 28 35 42 49 56 63 70
0 8 16 24 32 40 48 56 64 72 80
0 9 18 27 36 45 54 63 72 81 90
0 10 20 30 40 50 60 70 80 90 100

```

Afin de produire un tableau sans décalage de chiffres, utiliser le formatage des nombres à l'aide de l'annexe page suivante.

Algorithme pseudo-code	Python
<pre> pour ligne de 0 à 10 par pas de 1 faire pour colonne de 0 à 10 par pas de 1 faire afficher(ligne*colonne) sans retour à la ligne fin pour afficher un retour à la ligne fin pour </pre>	<pre> for ligne in range(0,11): for colonne in range(0,11): print("{:3d}".format(ligne*colonne),end=' ') print() </pre>

Annexe

Formatage des nombres

Instructions :	Résultat :
<pre>a=42 print('{:d}'.format(a))</pre>	<pre>4 2</pre>
<pre>a=42 print('{:4d}'.format(a))</pre>	<pre> 4 2</pre>
<pre>a=42 print('{:04d}'.format(a))</pre>	<pre>0 0 4 2</pre>
<pre>pi=3.141592653589793 print('{:06.2f}'.format(pi))</pre>	<pre>0 0 3 . 1 4</pre>

Pour plus d'informations voir <https://pyformat.info/>

Gestion des sauts de lignes

Instructions :	Résultat :
<pre>a=42 print('hello') print(a, 'bonjour')</pre>	<pre>hello 42 bonjour</pre>
<pre>a=42 print('hello', end='') print(a, 'bonjour')</pre>	<pre>hello42 bonjour</pre>