ACTIVITÉ 3: CALCULER AVEC PYTHON

Problématique: Comment utiliser Python pour effectuer et automatiser des calculs mathématiques ?

Objectifs: *À la fin de l'activité, vous serez capable de :*

- *traduire des calculs mathématiques en Python ;*
- *utiliser les principaux opérateurs et fonctions mathématiques ;*
- *utiliser la bibliothèque `math` dans un programme.*

Python permet d'**automatiser de nombreux calculs mathématiques et de résoudre rapidement des problèmes** avec différentes valeurs. Il possède les **opérations usuelles** ainsi que des instructions plus avancées, dont certaines sont regroupées dans des **bibliothèques**. Dans cette activité, nous allons découvrir les principales instructions mathématiques de Python et apprendre à les utiliser dans des programmes.

A FAIRE TOUT AU LONG DE L'ACTIVITÉ: COMPLÉTER LE TABLEAU À RETENIR

Le tableau « À RETENIR », situé en dernière page de l'activité, présente les principales instructions mathématiques utilisées en Python ainsi qu'un exemple de leur utilisation. Leur signification mathématique n'est pas indiquée : vous devrez la découvrir en traduisant et en testant les différents algorithmes proposés. Au fur et à mesure de l'activité, complétez la dernière colonne du tableau.

PARTIE 1 : OPÉRATIONS MATHÉMATIQUES

A. Opérations élémentaires

On considère **l'algorithme** ci-dessous et sa traduction en **langage python.**

demander(a) avec a un nombre entier
demander(b) avec b un nombre décimal
A ← a + b
B ← a - b
C ← a × b
D ← a ÷ b
E ← a^2
afficher(A)
afficher(B)
afficher(C)
afficher(D)
afficher(E)
	`a = int(input("a="))`
	`b = float(input("b="))`
	`A = a+b`
	`B = a-b`
	`C = a*b`
	`D = a/b`
	`E = a**2`
	`print(A)`
	`print(B)`
	`print(C)`
	`print(D)`
	`print(E)`

1. Pour a = 6 et b = 2, déterminer les valeurs des variables A, B, C, D et E

   A = 8 *B =* 4 *C =* 12 *D =* 3 *E =* 36

2. À l'aide des instructions proposées dans le tableau, compléter la traduction de cet algorithme en langage Python.

3. *Partie sur ordinateur*
   a) Saisir et compléter le programme dans Trinket.
   b) Exécuter le programme avec a = 6 et b = 2.
   c) Vérifier que les résultats correspondent à ceux obtenus à la question 1.
   d) Corriger si nécessaire les instructions choisies.

B. Division euclidienne

On considère **l'algorithme** ci-dessous et sa traduction en **langage python.**

demander(a) avec a un entier
demander(b) avec b un entier
quotient ← quotient de la division euclidienne de a par b
reste ← reste de la division euclidienne de a par b
afficher(quotient)
afficher(reste)
	`a = int(input("a="))`
	`b = int(input("b="))`
	`quotient = a//b`
	`reste = a%b`
	`print(quotient)`
	`print(reste)`

1.
   a) Pour a = 17 et b = 5, déterminer le quotient et le reste de la division euclidienne.
   On rappelle que : $17 = 5 \times quotient + reste$

   quotient = ….3…… *reste* = ….2……

   b) Pour a = 23 et b = 4, déterminer le quotient et le reste de la division euclidienne.

   quotient = ….5…… *reste* = ….3……

2. À l'aide des instructions proposées dans le tableau, compléter la traduction en Python.

3. *Partie sur ordinateur*
   a) Saisir et compléter le programme dans Trinket.
   b) L'exécuter avec a = 17 et b = 5.
   c) Vérifier que les valeurs obtenues correspondent à celles attendues.
   d) Tester ensuite le programme avec a = 23 et b = 4.

C. Valeur absolue et arrondi

On considère **l'algorithme** ci-dessous et sa traduction en **langage python.**

demander(a) avec a un nombre décimal
demander(b) avec b un nombre décimal
ecart ← valeur absolue de a-b
quotient ← a ÷ b
approximation ← quotient arrondi à 2 chiffres après la virgule
afficher(ecart)
afficher(approximation)
	`a = float(input("a="))`
	`b = float(input("b="))`
	`ecart = abs(a-b)`
	`quotient = a/b`
	`approximation = round(quotient,2)`
	`print(ecart)`
	`print(approximation)`

1. Pour a = -10 et b = 3, déterminer les valeurs des variables *ecart, quotient, approximation*

   ecart = …13…… *quotient* = $-\frac{10}{3}$…… *approximation* = …-3,33….

2. À l'aide des instructions proposées dans le tableau, compléter la traduction en Python.

3. *Partie sur ordinateur*
   a) Saisir et compléter le programme dans Trinket.
   b) L'exécuter avec a = -10 et b = 3.
   c) Vérifier que les valeurs obtenues correspondent à celles attendues.
   d) Modifier le programme pour arrondir le quotient à 3 chiffres après la virgule.

PARTIE 2: UTILISER LA BIBLIOTHÈQUE MATH

Certaines opérations mathématiques ne sont pas directement disponibles en Python. Elles sont regroupées dans des bibliothèques. **Une bibliothèque est un ensemble d'instructions** que l'on peut ajouter à Python lorsque l'on en a besoin.

Pour **utiliser les instructions de la bibliothèque** mathématique math, on écrit au début du programme :

```
from math import *
```

A. Racine carrée

Rappel: La distance entre deux points $A(x_A; y_A)$ et $B(x_B; y_B)$ est donnée par : $AB = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}$

On considère **l'algorithme** ci-dessous et sa traduction en **langage python.**

demander(xA)
demander(yA)
demander(xB)
demander(yB)
distance ← $\sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}$
afficher(distance)
	`from math import *`
	`xA = input("xA = ")`
	`yA = input("yA = ")`
	`xB = input("xB = ")`
	`yB = input("yB = ")`
	`distance = sqrt((xB-xA)**2+(yB-yA)**2)`
	`print(distance)`

1. Pour *A(1; 2)* et *B(4;6)*, calculer la distance *AB*:

   $AB = \sqrt{(4-1)^2 + (6-2)^2} = \sqrt{3^2 + 4^2} = \sqrt{9+16} = \sqrt{25} = 5$

2. À l'aide des instructions proposées dans le tableau, compléter la traduction en Python.
3. *Partie sur ordinateur*
   a. Saisir et compléter le programme dans Trinket.
   b. L'exécuter avec *A(1; 2)* et *B(4;6).*
   c. Vérifier que le résultat obtenu correspond à celui de la question 1.

B. Calculs avec π

On souhaite calculer précisément le périmètre et l'aire d'un disque de rayon r.

demander(r) avec r un nombre décimal
perimetre ← 2 × π × r
aire ← π × r^2
afficher(perimetre)
afficher(aire)
	`from math import *`
	`r = float(input("rayon = "))`
	`perimetre = 2*pi*r`
	`aire= pi*r**2`
	`print(perimetre)`
	`print(aire)`

1. À l'aide des instructions déjà identifiées et de celles qui restent à découvrir, compléter la traduction de l'algorithme en Python.

2. *Partie sur ordinateur*
   a. Saisir et compléter le programme dans Trinket.
   b. L'exécuter avec *r=5.*
   c. Vérifier la cohérence des résultats obtenus.
   d. Modifier le programme pour afficher le périmètre et l'aire arrondis à deux chiffres après la virgule.

C. Fonctions trigonométriques

On souhaite calculer le cosinus d'un angle de 60°. On considère **l'algorithme** ci-dessous et sa traduction en **langage python.**

angle ← 60 degrés
resultat ← cosinus(angle)
afficher(resultat)
	`from math import *`
	`angle = radian(60)`
	`resultat = cos(angle)`
	`print(resultat)`

1. Quelle est la valeur de cos(60°) ? ..... $\cos 60° = \frac{1}{2}$ .....
2. On souhaite utiliser l'instruction python suivante : *cos(angle).* Peut-on directement remplacer angle par 60?

   ..... Non, car cos attend un angle en radians .....
3. La bibliothèque math contient une instruction permettant de convertir un angle exprimé en degrés en radians. À l'aide des instructions proposées dans le tableau Compléter le programme Python afin de calculer cos(60°) .
4. *Partie sur ordinateur*
   a) Saisir et compléter le programme dans Trinket.
   b) L'exécuter et vérifier que le résultat obtenu est cohérent avec la valeur attendue.
   c) Modifier ensuite le programme afin de calculer successivement : sin(30°), cos(45°), tan(45°)

APPEL → **Appeler le professeur pour vérification**

D. Bonus: Que font ces instructions ?

On considère les instructions Python suivantes :

```
randint(1, 6)
randint(1, 10)
randint(5, 10)
random()
```

Leur fonctionnement ne vous est pas donné. À vous de le découvrir en réalisant différents tests dans Trinket.
Pour pouvoir utiliser ces instructions, ajouter au début du programme :

```
from random import *
```

1. Tester plusieurs fois chacune des instructions précédentes à l'aide de `print()`. Par exemple :

```
from random import *
print(randint(1, 6))
```

2. Pour chaque instruction, exécuter plusieurs fois le programme et observer les résultats.
3. Compléter le tableau à partir de vos observations.

Instruction testée	Résultats obtenus	Type de nombre obtenu	Que semble faire l'instruction ?
`randint(1,6)`	1, 2, 6	Entier entre 1 et 6	Nombre entier aléatoire entre 1 et 6 .....
`randint(1,10)`	1, 7, 9, ...	Entier entre 1 et 10	Nombre entier aléatoire entre 1 et 10 .....
`randint(5,10)`	5, 6, 7, 8, 9, 10	Entier entre 5 et 10	Nombre entier aléatoire entre 5 et 10 .....
`random()`	0,952...	Nombre décimal à 16 chiffres après la virgule, entre 0 et 1	Nombre aléatoire entre 0 et 1 décimal

1. À partir de vos observations, compléter :

`randint(a,b)` permet de obtenir un nombre entier aléatoire entre a et b

`random()` permet de obtenir un nombre décimal aléatoire entre 0 et 1

2. On considère **l'algorithme** ci-dessous et sa traduction en **langage python.**

resultat ← choisir un entier entre 1 et 10
afficher(resultat)
	`from random import *`
	`resultat = randint(1, 10)`
	`print(resultat)`

À l'aide des instructions que vous venez d'étudier, traduire cet algorithme en Python.

1. *Partie sur ordinateur*
   a) Saisir votre programme dans Trinket.
   b) L'exécuter plusieurs fois.
   c) Comparer les résultats obtenus lors des différentes exécutions.
   d) Expliquer à quelle expérience réelle ce programme pourrait correspondre.

Un lancé de dé à 10 faces

BILAN DE L'ACTIVITÉ 1.3: CALCULER AVEC PYTHON

Tableau d'instructions mathématiques en Python

Instruction Python	Exemple d'utilisation	Signification / utilisation mathématique
`+`	`a + 2`	addition
`-`	`a - 2`	soustraction
`*`	`a * 2`	multiplication
`/`	`a / 2`	division
`**`	`a ** 2`	puissance
`//`	`a // 2`	quotient de la division euclidienne
`%`	`a % 2`	reste de la division euclidienne
`abs()`	`abs(a)`	valeur absolue
`round()`	`round(a, 2)`	valeur approchée à 2 chiffres après la virgule

Remarque:

Pour utiliser les instructions `sqrt()`, `pi`, `sin()`, `cos()`, `tan()` et `radians()`, il faut ajouter la ligne ci-dessous au début du programme :

```
from math import *
```

`sqrt()`	`sqrt(a)`	Racine carrée
`pi`	`2 * pi * a`	Nombre π
`sin()`	`sin(a)`	Sinus d'un angle en radian
`cos()`	`cos(a)`	Cosinus " " "
`tan()`	`tan(a)`	tangente " " "
`radians()`	`radians(a)`	Conversion degrés → radians

Remarque:

Pour utiliser les instructions `randint()`, `random()`, il faut ajouter la ligne ci-dessous au début du programme :

```
from random import *
```

`randint()`	`randint(1,6)`	Nombre entier aléatoire entre 1 et 6 (donné de dé à 6 faces)
`random()`	`random()`	Nombre décimal aléatoire dans l'intervalle [0;1]

APPEL → **Appeler le professeur pour vérification**