TP1

CALCULS, VARIABLES ET FONCTIONS

I Réaliser un calcul, définir une variable

Python permet de réaliser simplement tous les calculs de base.

Opération	Syntaxe Python	Exemple
Addition	`+`	`2 + 5`
Soustraction	`-`	`2 - 5`
Multiplication	`*`	`2 * 5`
Division	`/`	`2 / 5`
Puissance	`**`	`2 ** 5`

La **virgule** des nombres **décimaux** s'écrira avec un **point** (par exemple, `2.3`).

Comme sur une calculatrice, faites attention aux parenthèses ! Par ailleurs, toutes les opérations doivent êtres explicites.

Il est toujours utile de pouvoir donner un nom à une valeur pour pouvoir la réutiliser plus tard. On dit que l'on affecte une valeur à une variable avec la commande :

```
nom = valeur
```

Attention à respecter cet ordre (ce n'est pas "valeur = nom" mais bien "nom = valeur" !)

Exercice 1

1. Écrire le calcul $\dfrac{3,1+10}{8\times 10^{-6}}$ en langage python.

2. Créer trois variables A, B et C et affecter à la variable A la valeur 10, à la variable B la valeur 12 et à C la valeur 3.

3. Effectuer les opérations suivantes : $2A-3B$, $C(A+2B)$, $A^3-(-1)^C+A^{B+C^2}$.

Exercice 2 Dans chaque cas, écrire le résultat attendu (de tête) puis, dans un second temps, vérifier.

1.

```
s = 0
s = s+1
s = s+2
```

Tableau des valeurs successives de s

	s
ligne 1	
ligne 2	
ligne 3	

2.

```
a = 1
b = 2
a = a-b
b = a+b
```

Valeurs successives de a et b

	a	b
ligne 1		
ligne 2		
ligne 3		
ligne 4		

3.

```
a,b,c = 1,2,1
a = 2*c + 3*b
c = b
b = a
```

Valeurs successives de a, b et c

	a	b	c
ligne 1			
ligne 2			
ligne 3			
ligne 4			

4.

```
a,b,c = 1,2,1
a = 2*c + 3*b
b = c
a = b
```

Valeurs successives de a, b et c

	a	b	c
ligne 1			
ligne 2			
ligne 3			
ligne 4			

Exercice 3 **Division euclidienne**

La division euclidienne entre deux entiers est la division avec reste.
Exemple : La division euclidienne de 17 par 3 est

$$17 = \underbrace{5}_{\text{quotient}} \times 3 + \underbrace{2}_{\text{reste}}.$$

Le **quotient** s'obtient avec `//`.
Le **reste** s'obtient avec `%`.

```
[In]: 17 // 3
```

```
[Out]: 5
```

```
[In]: 17 % 3
```

```
[Out]: 2
```

Calculer avec Python le reste et le quotient de la division euclidienne de 123456789 par 1234.

II Afficher une valeur

Première méthode

On crée une variable contenant la valeur à afficher, puis on appelle cette variable dans la console.

[In]:
```
A = 10
A
```

[Out]:
```
10
```

Cette méthode fonctionne si on a une seule valeur à afficher et qu'elle est demandée en dernière ligne d'une cellule. Dans l'exemple suivant, ça me marche pas :

[In]:
```
A = 10
A
B = 2
```

[Out]:
```
```

Affichage automatique : print

La fonction `print` permet de demander l'affichage d'une valeur et/ou d'un message (entre guillemets).

[In]:
```
A = 10
print('A =', A) # on affiche un message (entre guillemets) puis A
B = 2
print('B =', B)
```

[Out]:
```
A = 10
B = 2
```

Exercice 4 **Classique : échange de deux valeurs**

On considère deux variables a et b et on souhaite échanger leurs valeurs.
Si $a = 2$ et $b = 5$ au départ, on doit avoir $a = 5$ et $b = 2$ à la fin.
Votre programme doit fonctionner pour toutes valeurs de a et b.

```
a = 2 # exemple, tester aussi d'autres valeurs
b = 5 # exemple, tester aussi d'autres valeurs









print("a =", a)
print("b =", b)
```

Tester votre programme avec plusieurs valeurs de a et b.

III Définir et utiliser une fonction

Pour définir une **fonction** on utilise le format suivant

```
def nom_fonction(argument1, argument2, etc.):
    ''' Description de la fonction '''
    bloc_instructions
    return valeur_renvoyée_1, valeur_renvoyée_2, etc.
```

- Attention à l'indentation et aux deux points à la fin de la première ligne.
- `nom_fonction` est le nom de la fonction qui sera utilisé ensuite. Pas d'espace, pas de caractère spécial. On utilisera des noms de fonctions explicites.
- `argument1, argument2,...` sont les paramètres d'entrées. Ce sont les variables de la fonctions. Ne pas mettre de valeur ici, on garde des noms génériques.
- La description est affichée dans l'aide.
- Le mot clé `return` est essentiel. Il permet de définir le **résultat final**. L'exécution d'une fonction s'arrête dès que l'on tombe sur return et la fonction renvoie ce résultat.

Exemple

```
def fonction_f(x) :
    y = 3*x**2 - 2*x + 1
    return y
```

Pour faire appel à cette fonction, on donnera son nom et la valeur des paramètres d'entrée (ici x).

```
[In]: fonction_f(2)
```

```
[Out]: 9
```

Exercice 5 Définir une fonction renvoyant le discriminant de l'équation $ax^2 + bx + c = 0$. Les paramètres d'entrée seront a,b et c.

Tester votre fonction en calculant les discriminants de : $2x^2 - 4x + 5 = 0$ et de $x^2 + 6x + 9 = 0$