

Nom :	Prénom :	Classe :
-------	----------	----------

## SNT — Premiers pas en Python : calculs, commentaire, variable

Objectifs :

- Savoir reconnaître l'ordre d'exécution d'un programme Python.
- Savoir identifier un commentaire Python.
- Savoir réaliser des calculs mathématiques élémentaires en Python.
- Savoir affecter (stocker) une valeur à une variable.
- Savoir afficher la valeur stockée dans une variable.

### Recette de cuisine (ordre d'exécution d'un programme Python)

Dans une recette de cuisine, on trouve un certain nombre d'étapes, à suivre dans un **ordre précis**.

Exemple de recette :

- Tamiser 250g de farine.
- Ajouter 2 cuillères à soupe de sucre.
- Ajouter 5 pincées de sel.
- Incorporer progressivement 50cl de lait en mélangeant doucement.
- Etc.

Dans une recette, **l'ordre d'exécution**, c'est la suite des étapes que l'on suit dans le "bon sens" : tamiser la farine, ajouter le sucre, le sel, le lait, etc. Si l'on change cet ordre sans réfléchir (mettre le lait avant la farine), la recette risque d'être ratée, *même si chaque étape prise isolément est correcte*.

### Ordre d'exécution

Notre recette de cuisine possède un ordre d'exécution dans lequel chaque ligne est exécutée de haut en bas.

Un programme informatique écrit en langage Python<sup>1</sup> fonctionne de la même manière, exemple :

```
1 print('Bonjour')
2 print('Je suis un programme Python,')
3 print('constitué de 3 lignes de code.')
```

Que ce soit dans une recette ou un **programme, l'ordre d'exécution** est crucial, car la même liste d'actions faite dans un autre ordre ne donne pas forcément le même résultat.

En code, un mauvais ordre peut provoquer des erreurs, tandis qu'en cuisine il peut simplement rendre la réalisation impossible ou de mauvaise qualité...



<sup>1</sup> Python est un langage de programmation simple, lisible, facile à écrire. Il permet aux développeuses et développeurs de se concentrer sur la **logique** du code, plutôt que sur des détails syntaxiques complexes.

Python est aussi le **langage le plus utilisé au monde**, et il bénéficie d'une grande quantité de livres, documentations, tutoriaux et bibliothèques qui lui sont consacrés. On retrouve Python dans l'univers du jeu vidéo, du streaming, de la finance, de l'intelligence artificielle, de l'aérospatial, de la médecine, etc. Python est *partout* !

## ► Exercice 1 — prédire

Lisez le code Python suivant :

```
1 print('Je')
2 print('suis')
3 print('un')
4 print('programme')
5 print('informatique')
```

Prédisez ce que va afficher le code Python ci-dessus :

## ► Exercice 2 — exécuter

Il est temps d'exécuter le programme de l'exercice 1, afin de vérifier votre prédiction.

Pour exécuter le programme de l'exercice 1, scannez le code QR ci-dessous puis cliquez sur le bouton "Exécute".



*Alors, votre prédiction était-elle juste ?*

## Pourquoi pouvoir écrire des commentaires, en programmation ?

---

En programmation, un **commentaire** est une **ligne de code** qui ne sera **pas exécutée** par l'ordinateur.

Un commentaire n'est pas écrit *pour l'ordinateur* mais *pour les être humains* qui sont en train de lire le code, car il sert à donner une **information**.

En langage Python, une **ligne de commentaire** est précédée du **symbole** `#`

Voici un exemple de code Python contenant des commentaires :

```
1 # On donne la valeur 10 à la variable bob
2 bob = 10
3 # On donne la valeur 18 à la variable alice
4 alice = 18
```

### ► Exercice 3 — prédire

Lisez le code Python suivant :

```
1 # Affiche le mot « Bonjour »
2 print('Bonjour')
3 # Affiche la phrase « la planète »
4 print('la planète')
```

Prédisez ce que va **afficher** le code Python ci-dessus :

Selon vous, quelles sont les **numéros de lignes** qui ne seront pas exécutées par l'ordinateur ?

### ► Exercice 4 — exécuter

Il est temps d'exécuter le programme de l'exercice 3, afin de vérifier votre prédiction.

Scannez le code QR ci-dessous puis cliquez sur le bouton "Execute" :



*Alors, votre prédiction était-elle juste ?*

## Calculs mathématiques en Python

---

Le langage de programmation Python est également une formidable « calculatrice » qui permet de réaliser des opérations arithmétiques élémentaires telles que l'**addition**, la **soustraction**, la **multiplication**, la **division**.

Mais Python permet également de calculer la **puissance d'un nombre** ou encore de réaliser une division euclidienne, en calculant son **quotient** (le résultat entier de la division) et son **reste** !

Nous allons découvrir tout cela, à la page suivante ...

*quel suspense !*

## Tableau récapitulatif des calculs mathématiques en Python

Voici un tableau qui tente de récapituler quelques calculs mathématiques réalisables en Python :

Nom	Symbole Python	Exemple Python	Résultat	Vérification
Addition	<code>+</code>	<code>13 + 10</code>	23	$13 + 10 = 23$
Soustraction	<code>-</code>	<code>13 - 10</code>	3	$13 - 10 = 3$
Multiplication	<code>*</code>	<code>3 * 4</code>	12	$3 \times 4 = 12$
Division	<code>/</code>	<code>15 / 4</code>	3.75	$15 \div 4 = 3,75$
Puissance	<code>**</code>	<code>3 ** 3</code>	27	$3 \times 3 \times 3 = 27$
Quotient	<code>//</code>	<code>9 // 4</code>	2	Le <u>résultat entier</u> de $9 \div 4$ est 2
Reste	<code>%</code>	<code>9 % 4</code>	1	Le <u>reste</u> de la division $9 \div 4$ est 1

Notons que la **virgule** d'un *nombre à virgule* s'écrit avec un **point** en Python.

### ► Exercice 5 — prédire

Analysez chacun des calculs mathématiques Python ci-dessous, puis déterminez-en le résultat.

N°	Calcul Python	Résultat (à compléter)
1	<code>10 - 1</code>	
2	<code>8 * 4</code>	
3	<code>15 / 6</code>	
4	<code>2 ** 3</code>	
5	<code>15 // 6</code>	
6	<code>15 % 6</code>	
7	<code>9 + 2 * 2</code>	
8	<code>(9 + 2) * 2</code>	

## ► Exercice 6 — exécuter

Vérifiez vos résultats de l'exercice 5 en scannant le code QR ci-dessous.  
Une fois la page Web ouverte, cliquez sur le bouton "Execute" :



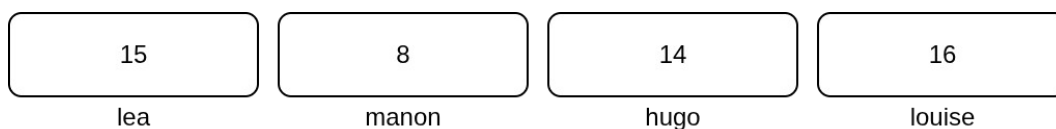
Alors, vos calculs étaient-ils justes ?

## Variable : à quoi ça sert ?

Une **variable** est un peu comme une **boîte de rangement étiquetée** dans laquelle on **stocke** des **informations**.

Prenons un exemple :

Nous souhaitons créer des variables qui serviraient à stocker l'âge de quatre élèves.



Le schéma ci-dessus illustre la création de quatre variables.

- Ces variables possèdent respectivement les **noms de variables** suivants : `lea`, `manon`, `hugo` et `louise`.
- Ces variables stockent respectivement les **valeurs** suivantes : `15`, `8`, `14` et `16`.

## Créer une variable en Python

En langage Python, pour **créer une variable** on utilise le signe `=`

En reprenant notre exemple, voici le code Python permettant de créer les variables `lea`, `manon`, `hugo` et `louise` en leur donnant les bonnes valeurs :

```
1 lea = 15
2 manon = 8
3 hugo = 14
4 louise = 16
```

Voilà, nous venons de créer (**déclarer**) quatre variables dont les noms sont `lea`, `manon`, `hugo` et `louise`, et dont les valeurs sont respectivement `15`, `8`, `14` et `16`.

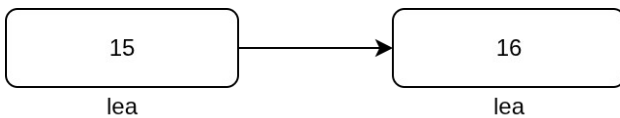
## ► Exercice 7 — créer

Déclarez trois variables dont les noms sont `tom`, `sarah` et `eva`, et dont les valeurs sont respectivement `23`, `16` et `14`.

## Changer la valeur d'une variable

Il est possible de **changer l'information stockée dans une variable**, tout en conservant la même étiquette.

En reprenant notre exemple, nous pouvons imaginer que l'une des élèves (Léa) vient de fêter son anniversaire :



Le schéma ci-dessus illustre le changement de valeur stockée dans la variable `lea`. Cette valeur passe de l'entier `15` à l'entier `16`.

## Changer la valeur d'une variable, en Python

En langage Python, pour changer la valeur d'une variable on utilise à nouveau le signe `=`

```
1 lea = 15
2 lea = 16
```

Dans le code ci-dessus, nous affectons la valeur `15` à la variable `lea` (ligne 1).

Puis nous affectons la valeur `16` à la variable `lea` (ligne 2).

Mais nous aurions pu faire cela autrement : en ajoutant (additionnant) la valeur 1 à la variable `lea` !

Voici comment faire cela en Python :

```
1 lea = 15
2 lea = lea + 1
3 print(lea)
```

Dans le code ci-dessus, nous affectons la valeur `15` à la variable `lea` (ligne 1).

Puis nous ajoutons `1` à la valeur stockée dans `lea` (ligne 2).

Puis nous affichons la valeur stockée dans `lea` (ligne 3), ce qui a pour effet d'afficher `16`.

Il est ainsi possible de réaliser tous les calculs mathématiques imaginables, en utilisant des variables (voir le chapitre « Tableau récapitulatif des calculs mathématiques en Python », vu précédemment.)



## ► Exercice 8 — modifier

Corrigez chacun des programmes ci-dessous, **en ne modifiant qu'une seule ligne d'affectation**, de manière à afficher le nombre 10. *La correction du premier programme (n°0) est donnée à titre d'exemple.*

N°	Code Python incorrect	Code Python corrigé (à compléter)
0	<pre>a = 11 b = 2 c = a - b print(c)</pre>	<pre>a = 11 b = 1 c = a - b print(c)</pre>
1	<pre>mael = 8 d = 1 mael = mael + d print(mael)</pre>	
2	<pre>a = 20 b = 3 c = a // b print(c)</pre>	
3	<pre>c = 60 b = 20 a = c - b print(a)</pre>	
4	<pre>clara = 3 clara = clara ** 2 + 3 print(clara)</pre>	
5	<pre>sushis = 39 amies = 4 partage = sushis // amies print(partage)</pre>	
6	<pre>pokemons = 10 copains = 11 reste = pokemons % pokemons print(reste)</pre>	

## ► Exercice 9 — créer

Créer le programme Python à partir de la description fournie :

Description	Python (à compléter)
<ul style="list-style-type: none"><li>- Déclarez une variable <code>a</code>, en lui affectant la valeur <code>23</code>.</li><li>- Déclarez une variable <code>b</code>, en lui affectant la valeur <code>10</code>.</li><li>- Déclarez une variable <code>c</code>, en lui affectant le résultat de <code>a</code> soustrait de <code>b</code>.</li><li>- Affichez la valeur de <code>c</code>.</li><li>- Soustraire <code>1</code> de la valeur stockée dans <code>c</code>.</li><li>- Affichez la valeur de <code>c</code>.</li></ul>	

correction:



## Ce qu'il faut retenir

Vous retrouverez ici l'essentiel du cours.

Astuce : transformez ces points en questions/réponses pour vos flashcards (Anki).

- En programmation, une **instruction** est une **ligne de code**.
- En programmation, un **programme** est composé d'une **suite d'instructions**.
- **L'ordre d'exécution** d'une suite d'instructions est crucial, car la même liste d'instructions, mais faite dans un autre ordre ne donne pas forcément le même résultat.
- En Python, la commande `print()` permet d'afficher une **valeur**.
- En programmation, un **commentaire** est une ligne de code qui n'est pas exécutée par l'ordinateur. Un commentaire n'est pas écrit *pour l'ordinateur* mais *pour les êtres humains* qui lisent le code. Un commentaire est précédé du symbole `#`.
- Python permet de réaliser des **opérations arithmétiques** telles que **l'addition**, la **soustraction**, la **multiplication** et la **division** dont les symboles sont respectivement `+`, `-`, `*` et `/`.
- Python permet de **calculer la puissance d'un nombre**, grâce au symbole `**`.
- Python permet de calculer le **quotient** (résultat entier) d'une division euclidienne, grâce au symbole `//`.
- Python permet de calculer le **reste** d'une division euclidienne, grâce au symbole `%`.
- En programmation, une **variable** est une sorte de boîte de rangement étiquetée dans laquelle on stocke une **information** (une **valeur**). Il est possible de changer la valeur stockée dans une variable.
- En programmation, donner une valeur à une variable s'appelle une **affectation**.
- Python permet **d'affecter une valeur à une variable** avec le symbole `=`.
- Python permet de réaliser des opérations arithmétiques en utilisant des variables.