Découvrir Python avec Basthon

Activité en autonomie - Variables, print(), input(), if / else

Nom : _____ Prénom : _____ Classe : _____

Avant de commencer

1. Ouvre **basthon.fr** et choisis la **console Python** (éditeur à gauche, console à droite).
2. Écris ton programme dans l'éditeur, puis clique sur **Exécuter**.
3. Quand le programme utilise `input()`, une zone ou une fenêtre apparaît : tape ta réponse puis valide avec Entrée.
4. Fais les exercices **dans l'ordre** : chacun utilise ce que tu as appris dans le précédent.
5. Garde ton code de chaque exercice : tu le rendras à ton professeur (copie-colle ou capture d'écran).

Règles d'or

- Python distingue les majuscules des minuscules : `Age` et `age` sont deux variables différentes.
- Les textes s'écrivent entre guillemets : `"Bonjour"`.
- Un message d'erreur n'est pas une catastrophe : lis-le, il indique la ligne à corriger.
- Teste ton programme plusieurs fois avec des valeurs différentes.

Le cours en 5 minutes

Lis ce mini-cours avant de commencer, puis reviens-y quand tu bloques. Pas besoin de tout retenir : les exercices te feront manipuler chaque notion.

Un programme, c'est quoi ?

Un programme est une **suite d'instructions** que l'ordinateur exécute **dans l'ordre, ligne par ligne**, de haut en bas. L'ordinateur est très obéissant mais ne devine rien : il faut lui écrire exactement ce qu'il doit faire, dans un langage qu'il comprend. Ici, ce langage est **Python**.

Une variable : une boîte avec une étiquette

Une variable sert à **garder une information en mémoire**. Imagine une boîte avec une étiquette (le nom de la variable) et un contenu (sa valeur).

```
age = 15
prenom = "Lina"
```

- `age` est le nom de la boîte, `15` est ce qu'on met dedans.
- Le signe = veut dire « range dans », pas « est égal à ».
- On peut changer le contenu à tout moment : `age = 16` remplace `15`.

Il existe plusieurs sortes d'informations, qu'on appelle des **types** :

Type	Exemple	À retenir
Texte	`"Lina"`	toujours entre guillemets
Nombre entier	`15`	sans guillemets, on peut calculer avec
Nombre à virgule	`1.65`	on écrit un point, pas une virgule

print() : faire parler le programme

La fonction `print()` **affiche** quelque chose à l'écran : un texte, une variable, ou les deux séparés par des virgules.

```
print("Bonjour")
print("Tu as", age, "ans")
```

Résultat : `Bonjour` puis `Tu as 15 ans`.

input() : laisser l'utilisateur répondre

La fonction `input()` **pose une question** et attend que l'utilisateur tape une réponse. Cette réponse doit être rangée dans une variable.

```
prenom = input("Ton prénom : ")
print("Bonjour", prenom)
```

Attention : `input()` renvoie **toujours du texte**, même si l'utilisateur tape un nombre. Pour obtenir un vrai nombre, on convertit avec `int()` :

```
age = int(input("Ton âge : "))
print(age + 1)
```

if / else : faire un choix

Une **condition** permet au programme de choisir quoi faire selon une situation. Python teste la condition : si elle est vraie, il exécute le premier bloc, sinon il exécute le bloc après `else`.

```
if age >= 18:
    print("Tu es majeur.")
else:
    print("Tu es mineur.")
```

- Les lignes `if` et `else` se terminent par deux-points `:`.
- Les lignes qui dépendent du test sont **décalées vers la droite** (indentation de 4 espaces ou touche Tab). C'est ce décalage qui dit à Python ce qui appartient au `if` ou au `else`.
- Pour comparer : == (égal), `!=` (différent), < (plus petit), > (plus grand), <= et >=.

À retenir

Notion	Rôle	Exercices
Variable	mémoriser une information	1
`print()`	afficher	2
`input()`	demander une information	3 et 4
`int()`	transformer un texte en nombre	4
`if / else`	choisir selon un test	5, 6 et 7

Exercice 1 - Mes premières variables

Objectif : stocker une information dans une variable et afficher sa valeur.

Consigne : crée trois variables : `prenom` (ton prénom), `age` (ton âge) et `taille` (ta taille en mètres, par exemple 1.65). Affiche ensuite leur valeur avec `print()`.

```
prenom = "Lina"
age = 15
taille = 1.65
print(prenom)
print(age)
print(taille)
```

Pour exécuter ton programme :

1. Écris ces lignes dans l'**éditeur** (à gauche) puis clique sur **Exécuter**.
2. Les valeurs s'affichent dans la **console** (à droite).

Résultat attendu :

```
Lina
15
1.65
```

Astuce : le signe = ne veut pas dire « égal » mais « range la valeur dans la variable ». Lis `age = 15` comme « age reçoit 15 ». `print()` sert à afficher une valeur dans la console (on le découvre en détail à l'exercice 2) : sans `print()`, Python range la valeur mais n'affiche rien.

Conseil : un texte s'écrit entre guillemets, un nombre sans guillemets. Pour un nombre à virgule, utilise un **point** : `1.65`, pas `1,65`.

Pour aller plus loin : change `age = 16`, clique sur Exécuter et regarde la console. Que s'est-il passé pour l'ancienne valeur ?

Ton code :

Exercice 2 - Afficher avec print()

Objectif : afficher un texte et une variable.

Consigne : en réutilisant tes variables, écris un programme qui affiche exactement ces trois lignes (avec tes propres valeurs) :

Résultat attendu :

```
Bonjour tout le monde
J'ai 15 ans
Je m'appelle Lina et je mesure 1.65 m
```

Astuce : dans `print()`, sépare les morceaux par des virgules : `print("J'ai", age, "ans")`. Python ajoute lui-même les espaces.

Conseil : `print(age)` affiche la valeur de la variable, mais `print("age")` affiche le mot « age ». Les guillemets font toute la différence. Si ton texte contient une apostrophe, entoure-le de guillemets doubles : `"J'ai"`.

Ton code :

Exercice 3 - Demander son prénom avec input()

Objectif : lire un texte tapé au clavier.

Consigne : écris un programme qui demande le prénom de l'utilisateur, puis lui dit bonjour. Teste-le avec ton prénom, puis avec celui d'un(e) ami(e).

Résultat attendu :

```
Ton prénom : Samir
Bonjour Samir
```

Astuce : `input("question")` affiche la question et renvoie ce qui est tapé. Il faut donc **ranger** la réponse dans une variable : `prenom = input("Ton prénom : ")`.

Conseil : mets un espace à la fin de la question (`"Ton prénom : "`) pour que la réponse ne soit pas collée au texte. Si rien ne s'affiche, vérifie que tu as bien cliqué sur Exécuter et que la zone de saisie est ouverte.

Ton code :

Exercice 4 - Calculer avec un nombre saisi

Objectif : lire un nombre avec `int()` et faire un calcul.

Consigne : demande l'âge de l'utilisateur, puis affiche son âge dans 10 ans.

Résultat attendu :

```
Ton âge : 15
Dans 10 ans tu auras 25 ans
```

Astuce : `input()` renvoie toujours du **texte**. Pour faire un calcul, on convertit : `age = int(input("Ton âge : "))`. Lis-le de l'intérieur vers l'extérieur : on lit un texte, puis `int` le transforme en nombre.

Conseil : si tu vois l'erreur `can only concatenate str` ou `unsupported operand`, tu as oublié `int()`. Tu peux aussi essayer volontairement sans `int()` pour voir ce qui se passe : `"15" + 1` ne marche pas, mais `15 + 1` si.

Ton code :

Exercice 5 - Ma première condition : majeur ou mineur ?

Objectif : exécuter une instruction seulement si un test est vrai.

Consigne : demande l'âge, puis affiche « Tu es majeur. » si l'âge est supérieur ou égal à 18, et « Tu es mineur. » sinon. Teste avec 12, 18 et 40.

```
if age >= 18:
    print("Tu es majeur.")
else:
    print("Tu es mineur.")
```

Astuce : lis la condition comme une phrase : « si l'âge est supérieur ou égal à 18, alors... sinon... ». Teste toujours la valeur limite (ici 18) : c'est là que se cachent les erreurs.

Conseil : n'oublie pas les **deux-points** `:` à la fin des lignes `if` et `else`, et l'**indentation** (4 espaces, ou la touche Tab) devant les lignes qui en dépendent. Sans elle, Python affiche `IndentationError`.

Ton code :

Exercice 6 - Le mot de passe

Objectif : comparer un texte avec ==.

Consigne : le mot de passe secret est `"python"`. Demande à l'utilisateur de le saisir, puis affiche « Accès autorisé » s'il est correct, « Accès refusé » sinon. Teste avec le bon mot de passe, puis avec un mauvais, puis avec « Python » (avec une majuscule).

Résultat attendu :

```
Mot de passe : python
Accès autorisé
```

Astuce : pour tester si deux choses sont égales, on utilise **deux** signes égal : ==. Les autres comparateurs sont `!=` (différent), <, >, <= et >=.

Conseil : = range une valeur, == compare. Si tu écris `if mdp = "python":`, Python signale une `SyntaxError`. Observe aussi ce que donne « Python » avec une majuscule : pour Python, ce n'est pas le même texte.

Ton code :

Exercice 7 - Mini-projet : le prix avec réduction

Objectif : assembler variables, `input()`, `print()` et `if / else`.

Consigne : écris un programme qui demande le prénom du client et le prix d'un article en euros (nombre entier). Si le prix est supérieur ou égal à 50 euros, le client a 10 euros de réduction ; sinon il paie le prix normal. Le programme affiche le prix à payer.

Résultat attendu :

```
Ton prénom : Lina
Prix de l'article : 60
Bonjour Lina, tu paies 50 euros
```

Astuce : avant de coder, écris les étapes sur papier : (1) demander le prénom, (2) demander le prix, (3) tester si le prix est au moins 50, (4) calculer et afficher. Code ensuite une étape à la fois et exécute après chaque étape.

Conseil : teste avec 30, 50 et 60 pour vérifier les deux cas. Si ça ne marche pas, ajoute un `print(prix)` temporaire pour voir ce que contient ta variable à cet endroit.

Défi bonus : ajoute un troisième cas avec `elif` : à partir de 100 euros, la réduction passe à 25 euros.

Ton code :

Bravo, tu as terminé ! Envoie ton code (copie-colle ou capture d'écran) à ton professeur.