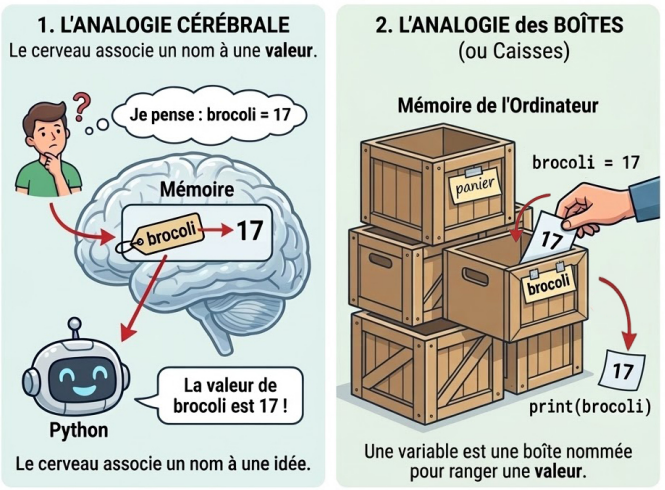
Les variables en Python

Les variables sont l'un des concepts fondamentaux en programmation. Elles permettent de **stocker des informations et de les utiliser dans un programme**.



Une variable est **un espace de mémoire temporaire où on stocke une valeur**.

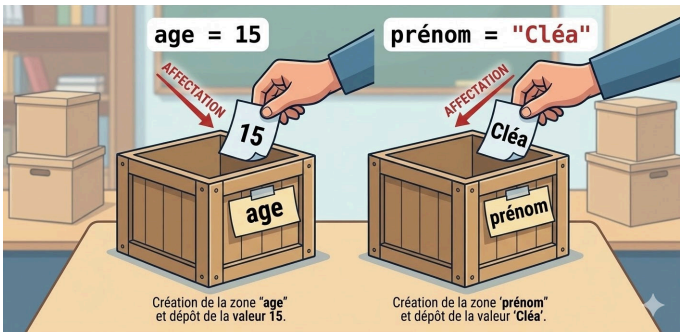
Premier exemple

Prenons un exemple pour illustrer ce concept en Python :

```
1 | nom = "Cléa"
2 | age = 15
```

Dans cet exemple :

- `nom` est une variable qui contient le texte `Cléa`,
- `age` est une variable qui contient le nombre `15`,

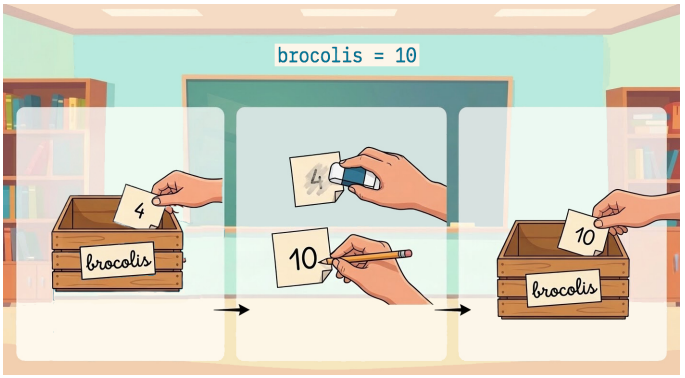


Ensuite, on pourra récupérer ces valeurs pour les afficher ou les utiliser dans un calcul par exemple.

Une variable est une zone **temporaire** dans la **mémoire** de l'ordinateur, à laquelle on donne un **nom** et dans laquelle on stocke une **valeur**.

On peut doncc changer la valeur d'une variable... autant de fois qu'on veut. Imagine une variable `brocolis` qui contient `4`. On peut lui affecter une nouvelle valeur:

```
1 | brocolis = 10
```



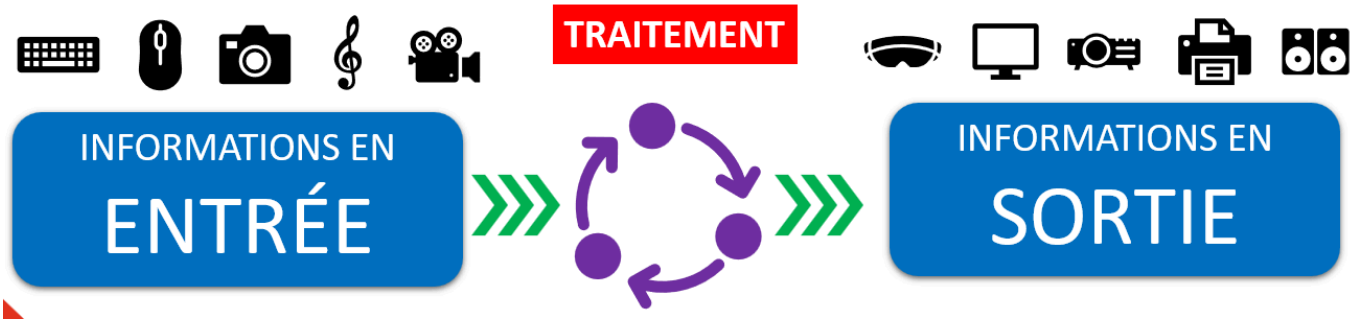
Importance des variables

Les variables servent à **stocker des informations** que le programme peut **utiliser** ou **modifier** au cours de son exécution. Elles agissent comme des "**boîtes**" dans lesquelles on peut ranger une valeur, par exemple un nombre, une chaîne de caractères, ou encore le résultat d'un calcul. **On utilise les variables lorsque l'on souhaite conserver une donnée pour s'en servir plus tard**, par exemple pour effectuer des **opérations**, **afficher un résultat**, ou gérer des interactions avec l'utilisateur.

Les variables permettent également de **rendre le code plus clair** et réutilisable, car elles **remplacent des valeurs fixes par des noms compréhensibles**.

En résumé, **on utilise des variables pour stocker les valeur qui seront affichées, utilisées ou modifiées plus tard dans le programme**, souvent plusieurs fois.

Souviens-toi de ce schéma:



Un programme reçoit des données/informations en entrée, leur applique un traitement et fournit des données/informations en sortie. **Dans un programme informatique, il y a donc des données et des traitements**. Retiens ceci:

Les données sont stockées dans des variables.

Déclarer ses Variables

En Python, les variables sont **déclarées** en leur **affectant** une valeur.

En Python, **déclarer une variable** signifie **créer une variable** en lui donnant un **nom** et une **valeur**.

Par exemple, pour déclarer une variable nommée "`age`" avec une valeur de 25, nous pouvons utiliser le code suivant :

```
1 | age = 25
```

Cela veut dire qu'on va **stocker** la valeur `25` dans la variable `nom`. Cela se fait **de droite à gauche**.

💡 Il faut **toujours** déclarer une variable avant de l'utiliser. Si tu utilises une variable avant de l'avoir déclarée, Python va afficher l'erreur **NameError**:

```
>>> %Run -c $EDITOR_CONTENT
  Traceback (most recent call last):
    File "<string>", line 1, in <module>
  NameError: name 'age' is not defined

>>>
```

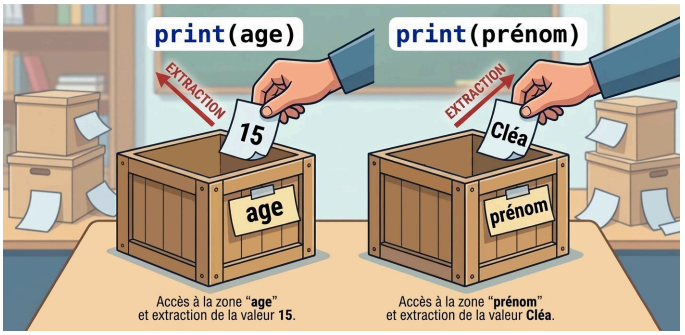
`"name 'age' is not defined"` signifie que la variable `age` n'a pas été déclarée au moment où elle est utilisée. Cela peut peut être dû au fait que:

- tu essaies d'utiliser une variable avant de l'avoir déclarée
- il y a une faute dans le nom (ex: `prenon` au lieu de `prenom` ou une différence minuscules/majuscules)
- tu as oublié les guillemets autours d'un texte (par exemple: `print(Sara)`)

Utiliser les Variables

Une fois la variable déclarée, on peut l'afficher ou l'utiliser dans des opérations et autres instructions.

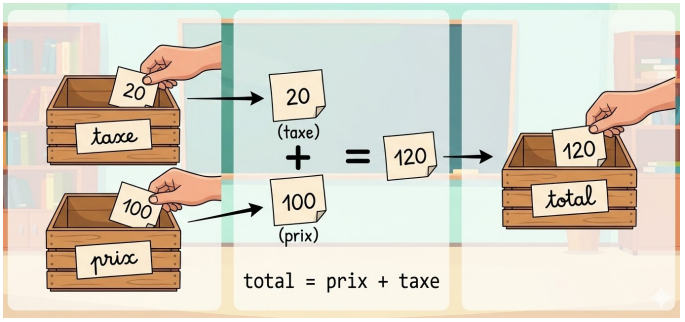
```
1 age = 15
2 prenom = "Cléa"
3 print (age) # Affichera 15
4 print (prenom) # Affichera Cléa
```



Explications

- `print` permet d'afficher quelque chose à l'écran.
- ligne 3: On affiche le **contenu** de la variable `age`: 15
- ligne 4: On affiche le **contenu** de la variable `prenom`: Cléa

```
1 prix = 100
2 taxe = 20
3 total = prix + taxe # total vaut maintenant 120
4
5 print (total) # Affiche 120
```



Explications

- ligne 3: On calcul le total en additionnant les **valeurs** contenues dans les variables `prix` et `taxe` et on stocke le résultat dans la variable ´total`.
- ligne 5: On affiche le **contenu** de la variable `total`, c'est à dire `120`

Ce code affiche:

```
120
```

Il est aussi possible, en une instruction, de mettre à jour la valeur d'une variable:

```
1 brocolis = 18
2
3 brocolis = brocolis + 4 # maintenant, brocolis vaut 22 (18+4)
```

Ligne 4: Python exécute d'abord les instructions à droite du `=`. Il récupère la valeur de `brocolis` (=18), et y ajoute 4 (=22) . Python stocke ensuite le résultat (22) dans la variable `brocolis`.

Conclusion

Les variables sont une partie essentielle de la programmation. Elles permettent de **stocker et manipuler des données**. En maîtrisant les variables, tu pourras écrire des programmes plus dynamiques et puissants.