

04- Variables (Programmation)



I – Visualiser les variables dans Thonny

On crée des variables en saisissant un nom, suivi du signe =, suivi lui-même d'une valeur ou d'une expression.

Dans Thonny, on peut visualiser le contenu d'une variable en utilisant le menu VIEW – VARIABLES ou AFFICHAGE – VARIABLES.

II – Variables en Python

2.1 Déclaration et affectation

Déclarer une variable veut dire réserver un espace mémoire en lui attribuant un nom, un alias.

Affecter veut dire d'attribuer un contenu à une variable.

Exemple en Python :

```
>>> a = 5
>>> a
5
```

2.2 Version naïve de la boîte



On peut voir dans un premier temps, une variable comme une sorte de nom qu'on attribuerait à une boîte contenant des données.

2.3 Incrémentation

Incrémenter veut dire d'augmenter le contenu d'une variable préexistante. On prend son contenu actuel et on l'utilise pour lui attribuer une nouvelle valeur.

```
>>> a = 8
>>> a
8
>>> a = a + 1
>>> a
9
```

2.4 Type de variables en Python

La fonction native **type()** permet de connaître le type du contenu associé à une variable.

Les types de base en Python sont :

- * Booléen (bool) : True ou False
- * Integer (int) : les entiers
- * Float (float) : les nombres à virgules
- * String (str) : les chaînes de caractères

Attention à la contenance des strings :

```
>>> a = "5"
>>> b = a + a
>>> b
"55"

>>> a = "5"
>>> c = a * 2
>>> c
"55"
```

2.5 Transformer un type de contenu

Il existe plusieurs fonctions permettant de tenter de transformer un contenu d'un type en un autre type, si la transformation peut avoir un sens.

- **int()** pour tenter de transformer en entier.
- **float()** pour tenter de transformer en réel.
- **str()** pour tenter de transformer en chaîne de caractères.
- **bool()** pour transformer un contenu en booléen

```
>>> a = "5"
>>> d = int(a) * 2
>>> d
10
```

2.6 Noms des variables

- Commencez impérativement par une miniscule.
- Si nom complexe, deux façons de faire :
 - hauteur_mur
 - hauteurMur
- Evitez les noms trop peu explicites
- Evitez les noms vraiment trop longs
- Ne jamais prendre un mot-clé réservé
- Python est sensible à la casse : majuscule et minuscule sont distingués : **toto** et **toTo** sont deux noms de variables différents.

III – Annexe : Mots réservés

False
None
True
and
as
assert
except

class
continue
def
del
elif
else
in

finally
for
from
global
if
import
raise

is
lambda
nonlocal
not
or
pass

return
try
while
with
yield
break