

Python

Cours 2 - Syntaxe, Typage, et Variables

Steve Lévesque, Tous droits réservés © où applicables

Table des matières

- 1 Syntaxe (indentation)
- 2 Typage et casting
- 3 Variables et structures de données
 - Variables
 - Integer
 - String
 - Structures de données
 - Tableaux
 - Dictionnaires

Syntaxe (indentation)

Python fonctionne sur la base de l'indentation pour la structuration ainsi que la validation du code.

Contrairement aux autres langages populaires, une mauvaise indentation/structure du code va causer une erreur de compilation.

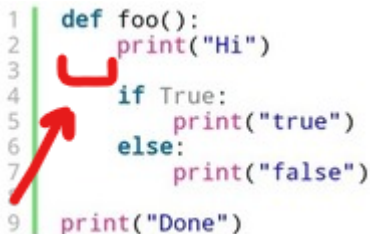
*S'applique aussi aux fonctions.

Syntaxe (indentation)

En général, une indentation en Python équivaut à 4 espaces.

Contrairement à JavaScript ou HTML, ce n'est pas standard dans la communauté d'avoir une indentation de 2 espaces.

Certains disent que c'est 4 espaces comme notion commune, puisque "def" et l'espace équivaut à 4 espaces, et le code est bien aligné avec une tabulation de 4 espaces.



```
1 def foo():
2     print("Hi")
3
4     if True:
5         print("true")
6     else:
7         print("false")
8
9     print("Done")
```

The image shows a Python code snippet with line numbers 1 through 9 on the left. The code is: `def foo():` (line 1), `print("Hi")` (line 2), (line 3 is blank), `if True:` (line 4), `print("true")` (line 5), `else:` (line 6), `print("false")` (line 7), (line 8 is blank), and `print("Done")` (line 9). A vertical green line is at the start of line 1. A red bracket highlights the indentation of line 2. A red arrow points to the indentation of line 4, which is 4 spaces from the start of the line.

Figure: Les indentations ont 4 espaces en général.

Syntax (indentation) - Exemple 1

Listing: https://www.w3schools.com/python/python_syntax.asp

```
1  # Steve Levesque, All rights reserved
2
3  # Incorrect
4  if 2 > 1:
5  print("2 is larger than 1 : True")
6
7  # Correct
8  if 2 > 1:
9      print("2 is larger than 1 : True")
```

Syntax (indentation) - Exemple 2

Listing: https://www.w3schools.com/python/python_syntax.asp

```
1  # Steve Levesque, All rights reserved
2
3  # Different invoking methods for a Python file.
4
5  print("Always executed")
6
7  if __name__ == "__main__":
8      print("Executed when invoked directly")
9  else:
10     print("Executed when imported")
```

Typage et casting

Le type est renforcé en Python. Alors il faut bien les gérer lorsque l'on programme.

Généralement, les langages sont typés lorsque leur but demande beaucoup de précision.

Par exemple, dans un langage typé, il n'est pas possible de concaténer un chiffre (int, float, etc.) avant de l'avoir converti en chaîne de caractères (string).

Typage et casting

Listing: https://www.w3schools.com/python/python_datatypes.asp

```
1  # Steve Levesque, All rights reserved
2
3  """
4      print("La note de passage est : " + 60)
5          ~~~~~
6  TypeError: can only concatenate str (not "int") to str
7  """
8  print("La note de passage est : " + 60)
```


Typage et casting

Listing: https://www.w3schools.com/python/python_datatypes.asp

```
1  # Steve Levesque, All rights reserved
2
3  """
4  La note de passage est : 60
5  """
6  print("La note de passage est : " + str(60))
```

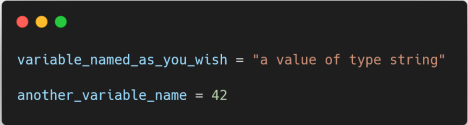
Variables - Définition

Les variables sont des conteneurs pour stocker des données (valeurs) pouvant être appelées de nom pour les réutiliser.

Variables - Définition

Python n'a pas de mot clé pour définir les variables.

Il suffit de nommer la variable, suivi de “=” et mettre la valeur par la suite.



```
variable_named_as_you_wish = "a value of type string"  
another_variable_name = 42
```

Figure: Exemple de déclarations (string en haut, int en bas).

Variables et structures de données - Integer

Python à deux types numériques courant :

- Int : Entiers
- Float : Décimales

Variables et structures de données - Integer

Listing: Int et Float sont une précision numérique différente

```
1  # Steve Levesque, All rights reserved
2
3  x = 42          # int
4  y = 123.456    # float
5
6  print(type(x)) # <class 'int'>
7  print(type(y)) # <class 'float'>
```

Variables et structures de données - String

Une chaîne de caractères est un mélange de lettres et de chiffres (alphanumérique) entourés par des guillemets simples ou doubles.

Variables et structures de données - String

Listing: String en simple ou double guillemets

```
1  # Steve Levesque, All rights reserved
2
3  print("This is a string double quote")
4
5  print('This is a string single quote')
6
7  print("Client001 - Alphanumeric string")
```

Structures de données - Définition

Une structure de données est utilisée pour stocker de multiples éléments dans une seule variable.

Nous allons voir les structures suivantes :

- Listes (List)
- Dictionnaires (Dictionary)

Variables et structures de données - Tableaux

Les listes sont utilisées pour stocker plusieurs éléments dans une seule variable.

Les listes sont l'un des 4 types de données intégrées en Python utilisées pour stocker des collections de données, les 3 autres sont **Tuple**, **Set** et **Dictionary**, tous avec des qualités et des utilisations différentes.

Les listes sont créées à l'aide de crochets carrés : “[]”

Variables et structures de données - Tableaux

Listing: https://www.w3schools.com/python/python_lists.asp

```
1  # Steve Levesque, All rights reserved
2
3  # Base declaration
4  grades = [60, 80, 90, 80]
5
6  # Print directly
7  print(grades)
8
9  # Convert to string and print with concat to another string sentence
10 print("The array itself as a string : " + str(grades))
11
12 # Iterative print for each element in array (in order)
13 print()
14 print("Print each element of the array with a special loop : ")
15 for grade in grades:
16     print(grade)
```

Variables et structures de données - Dictionnaires

Les dictionnaires sont utilisés pour stocker les valeurs de données dans des paires clé/valeur.

Un dictionnaire est une collection ordonnée*, modifiable et qui n'autorise pas les doublons.

Les dictionnaires sont écrits avec des accolades “{}”, et ont des clés et des valeurs.

*Depuis la version 3.7 de Python, les dictionnaires sont ordonnés. Dans Python 3.6 et les versions antérieures, les dictionnaires ne sont pas ordonnés.

Variables et structures de données - Dictionnaires

Listing: https://www.w3schools.com/python/python_dictionaries.asp

```
1  # Steve Levesque, All rights reserved
2
3  # "key": "value"
4  example = {
5      "brand": "Ford",
6      "electric": False,
7      "year": 1964,
8      "year": 2020, # Will replace 1964 with 2020 since duplicate
9      "colors": ["red", "white", "blue"]
10 }
11
12 # {'brand': 'Ford', 'electric': False, 'year': 2020,
13 # 'colors': ['red', 'white', 'blue']}
14 print(example)
```

Bibliographie

- `https://www.w3schools.com/python/`