

INTRODUCTION & INSTALL

Python est un langage de programmation informatique se servant d'un vocabulaire et d'un ensemble de règles grammaticales. Simple et rapide à apprendre, ce langage permet d'ordonner à un ordinateur d'effectuer des tâches. Nous pouvons l'utiliser pour la réalisation d'application, de site web, de jeux, d'IA, etc. C'est un langage simple qui permet de se concentrer sur ce qu'on fait plutôt que de la façon dont on le fait.

Pour l'installer, suivez le lien : <https://www.python.org/downloads/>

COMMENTAIRE

Il est important de commenter votre code !

Cela permet aux autres de le comprendre (et vous aussi)

Il y a deux manières :

- sur une ligne :

```
#comme ntaire
```

- sur plusieurs lignes :

```
lignes 1
```

```
lignes n..." " "
```

VARIABLES

Déclaration de variables :

```
x = nombre, boolean, strg, etc...
```

```
Integer -2, 18 #nombre entier
```

```
Float -2.1, 18.2 #nombre à virgule
```

```
String " hello world** #texte
```

```
Boolean True, False #1 ou 0
```

```
List[variableur,...]#liste
```

```
Tuple( variableur ,...)#immuable
```

```
Dictio nary{clé: variableur,..} #dicti onnaire
```

CONVERSION DE TYPE

```
int(exp) #converti exp en Integer
```

```
float(exp) #converti exp en Float
```

```
str(exp) #converti exp en String
```

```
chr(num) #converti num en ASCII
```

OPERATIONS

OPERATEURS DE COMPARAISON

```
x < y #inférieur x <= y #inf. ou égale
```

```
x > y #supérieur x >= y #sup. ou égale
```

```
x == y #égale x != y #non égale
```

OPERATEURS ARITHM ETIQUES

```
x + y #addition x - y #soust -raction
```

```
x * y #multi pli cation x / y #division
```

```
x % y #modulo : retourne le reste de la division
```

```
x ** y #puissance
```

OPERATEURS BOOLEENS

```
not x x and y x or y
```

CONDITIONNELLE

```
if expression:
```

```
    instru ction.s
```

```
elif expres sion:#optio nnelle
```

```
    instru ction.s
```

```
else: #optio nnelle
```

```
    instru ction.s
```

BOUCLES

BOUCLE WHILE

```
while expres sion:
```

```
    instru ction.s
```

BOUCLE FOR

```
for var in expres sions:
```

```
    instru ction.s
```

BOUCLE FOR AVEC UNE RANGE

```
for i in range (début, fin [,étape]):
```

```
    instru ction.s
```

#ATTENTION le début est inclus mais pas la fin

FONCTIONS

DEFINITIONS D'UNE FONCTION

```
def name(args1, args2, ...):
```

```
    instru ction.s
```

```
    return expr
```

FONCTIONS UTILES

- afficher un texte

```
print('texte')
```

- demander à l'util isateur une entrée

```
input('phrase')
```

LISTE / TUPLE

list = [valeur1, ...] créer une liste

s[i] ième objet de la liste s (1er élément = 0)

print(list) afficher une liste

print(list[a]) afficher valeur a de list

s[début : fin] coupe s du début à fin (exclus)

list.index(a) afficher l'indice de la valeur a

len(list) connaître le nombre d'élément dans la liste

Fonctions ne marchant pas sur Tuple

list.append() ajouter un élément à la fin de list

list.remove() enlever un objet dans la liste

list.sort() trier la liste

DICTIONNAIRE

dico = {} : créer un dictionnaire

dico.update(list2) ajouter un dico dans un dico

dico.keys() : retourner les clés

dico.values() : retourner les valeurs

dico.items() : retourner les objets

dico.get() : accéder à une valeur si elle existe.

dico.get(key, default) si la clé n'est pas trouvée = "None".

DICTIONNAIRE (cont)

dico.pop() : retirer une clé, valeur, ou objet.

OPERATIONS SUR CHAÎNES

**** : sortir un caractère spécial

"blabla \ "blabla\"."

str + str : concaténer plusieurs chaînes

print("mot" in variable) savoir si le caractère existe

len() connaître la longueur d'une chaîne

s.lower() remplace les majuscules par minuscules

s.upper() remplace les minuscules par majuscules

s.format() remplacer des {} dans une chaîne

s.split(sep) séparer par des valeurs ou des mots.

s.title() transforme une chaîne en format titre

s.split() divise une chaîne pour en faire une liste

s.find() trouve la première apparition de l'indice

s.join() joindre des chaînes entre elles

s.split(sep) séparer par des -

LIBRAIRIES

Pour apporter de nouvelles fonctionnalités à Python, nous utilisons des bibliothèques ou librairies.

Importer une librairie

import nom_librairie

Importer une fonction présente dans une librairie

from nom_librairie import fonction_voulue

numpy #Manipulation de matrices et tab. multi D.

pandas #Manipulation et analyse de données

matplotlib #Création et Visualisation de graphique

seaborn #Data Visualisation (matplotlib++)

scikit learn #Machine Learning

tensorflow #Deep Learning



By Grp1CheatSheet

cheatography.com/grp1cheatsheet/

Not published yet.

Last updated 22nd October, 2022.

Page 3 of 2.

Sponsored by [Readable.com](https://readable.com)

Measure your website readability!

<https://readable.com>