**Boucles et dictionnaires**

**Boucles For et dictionnaires**

Nous pouvons parcourir à la fois les clés et les valeurs d’un dictionnaire.

my\_books = {

"title": "Harry Potter",

"author": "JK Rowling",

}

for x, y in my\_books.items():

print("the" + x + "is" + y)

>> the title is Harry Potter

the author is JK Rowling

**Opérateur de boucles**

* range(start, stop[, step]) : itérateur dans les boucles.

print(list(range(1, 10, 2)))

>> [1, 3, 5, 7, 9]

* enumerate(iterable) : énumérer chaque élément de l'itérable

for item in enumerate('abcd'):

print(item)

>>

(0, 'a') # Syntax : (index , value)

(1, 'b')

(2, 'c')

(3, 'd')

for (index\_count, letter) in enumerate('abcd'):

print('At index {} the letter is {}'.format(index\_count, letter))

>>

At index 0 the letter is a

At index 1 the letter is b

At index 2 the letter is c

At index 3 the letter is d

* zip(itérable,..) : concat [itérables, …] dans un tuple.

list1 = [1,2,3]

list2 = ['a','b','c']

list3 = [1.1, 2.2, 3.3, 4.4, 5.5]

for item in zip(list1, list2, list3): # only go as far it is possible

print(item)

>>

(1, 'a', 1.1)

(2, 'b', 2.2)

(3, 'c', 3.3)

**Pour le reste**

La partie else est facultative. Lorsqu'elle est incluse, elle est toujours exécutée une fois la boucle for terminée, sauf si une instruction break est rencontrée.

for i in range(1, 3):

print(i)

else:

print('The for loop is over')

>>

1

2

The for loop is over

**Tandis que d'autre part**

while some\_condition:  
# faire  
autre chose:  
# faire une autre chose

x = 0

while x < 2:

print(f'x is {x}')

x += 1

else:

print('x is bigger than 2')

>>

x is 0

x is 1

x is bigger than 2

**Pause, Continuer, Passer**

* break : interrompre la boucle (arrêter l'exécution d'une instruction en boucle)  
  si vous sortez d'une boucle for ou while, tout autre bloc de boucle correspondant n'est pas exécuté.

for letter in 'Leonardo':

if letter == 'a':

break

print(letter, end='') # end='' renders each letter next to the other

>> Leon

while True:

s = input('Enter something : ')

if s == 'quit':

break

print('Length of the string is', len(s))

print('Done')

* continuer : revenir au début de la boucle (continuer à l'itération suivante de la boucle)

for letter in 'Leonardo':

if letter == 'o':

continue

print(letter, end='') # dont execute for 'o' letter

>> Lenard

while True:

s = input('Enter something : ')

if s == 'quit':

break

if len(s) < 3:

print('Too small')

continue

print('Input is of sufficient length')

* passer : ne rien faire

for item in [1,2,3]:

# comment

pass # to avoid the error

print('Finish the script')

>> Finish the script

**🚀 Défi**

Implémentez une boucle demandant à l'utilisateur d'ajouter des entrées au dictionnaire jusqu'à ce que « quit » soit saisi. Affichez le dictionnaire final.

**Conclusion**

Grâce à cette leçon, vous avez appris à combiner efficacement boucles et dictionnaires pour gérer et manipuler des données de manière dynamique. Ces compétences vous aideront à créer des applications Python plus robustes et interactives, capables de gérer efficacement diverses opérations de données.