

Introduction a la Programmation avec le language Python

November 15, 2016

Introduction la programmation avec le language Python

1 Installer Python sur son ordinateur

- Il y a beaucoup d'outils, qui permettent de gnrer des programmes en Python: Microsoft Visual studio, ..etc
- Dans ce cours nous allons utiliser **Jupyter-notebook** comme diteur de texte pour construire nos programmes.
- Download [Anconda](#)

2 les variables

```
In [7]: age = 20
        print(age)

        name = "Vincent"
        print(name)

20
Vincent

In [14]: width = 15
        height = 45

        area = width * height
        perimeter = 2* ( width + height )
```

3 Les branchements conditionnels

3.1 Commande if – exprimer une conditions

```
answer=input("Wouldyoulikeexpressshipping? ")
ifanswer=="yes":
    print("Thatwillbeanextra$10") print("Haveaniceday")
```

Indentation aprs la commande “if” est important, le code indent est excut seulement si la condition est vrais.

```
In [20]: flag = True
        if flag:
            print("I am in the if statement")
        else:
            print("Else")

I am in the if statement
```

3.2 La commande “elif” permet de dclarer une srie de branchement conditionnel

```
country=input("Whereareyoufrom?")

ifcountry=="CANADA":
    print("Hello")
elifcountry=="GERMANY":
    print("GutenTag")
elifcountry=="FRANCE":
    print("Bonjour")
```

“elif” est l’abrviation de Else if

4 Les boucles

4.1 La boucle for

```
In [22]: for step in range(5):
         print(step)
```

```
0
1
2
3
4
```

4.2 la boucle while

```
answer="0"

whileanswer!="4":
    answer=input("What is 2+2")

print("Yes! 2+2=4")
```

5 Les fonctions

1. utiliser le mot cl: **def**
 - Recousis pour define
2. donner un nom sa fonction
3. donner un nom aux paramtres de la fonction
4. crire le corps de sa fonction

```
In [26]: def printMsg(name):
         print("hello", name)
         return

         printMsg("Vincent")
```

hello Vincent

6 Les structures de donne en Python

6.1 les listes

```
In [33]: l = [1, 2, 3, 4, "test"]
```

```
    for elem in l:
        print(elem)

    l1 = [1, 2, 3, 4]
    for i in range(len(l1)):
        l1[i] = l1[i] + 1

    print(l1)
```

```
1
2
3
4
test
[2, 3, 4, 5]
```

6.2 les dictionnaires

```
In [1]: d = {"nom": "Gauthier", "prenom": "Vincent"}
        d["age"] = "x"
```

```
    print(d["prenom"])

    for k, v in d.items():
        print(k,v)
```

```
Vincent
prenom Vincent
nom Gauthier
age x
```

7 La bibliotheque Numpy

Plus d'information sur la librairie Numpy l'adresse suivante : [Scipy Lecture Notes](#)

```
In [42]: import numpy as np
```

```
    V = np.array([1, 2, 3, 4])

    print(2*V)
```

```
[2 4 6 8]
```

```
In [46]: A = np.array([[1, 1], [1, 1]])
        V = np.array([1,1])
```

```
    # Produit vectoriel
    np.dot(A,V)
```

```
Out[46]: array([2, 2])
```

```
In [47]: # Transpose
         A = np.array([[0, 1], [0, 0]])
         A.T
```

```
Out[47]: array([[0, 0],
               [1, 0]])
```

8 Afficher des graphique avec Matplotlib

```
In [37]: import pylab as plt
         %matplotlib nbagg

         x = np.linspace(-3, 3, 100)

         def f(x):
             y = x**2 + 1
             return y

         plt.figure()
         plt.plot(x, f(x), lw=2)
```

```
<IPython.core.display.Javascript object>
```

```
<IPython.core.display.HTML object>
```

```
Out[37]: [<matplotlib.lines.Line2D at 0x11ac68518>]
```

9 La librairie Pandas

```
In [16]: import pandas as pd
         import numpy as np
         df = pd.DataFrame({'A' : ['one', 'one', 'two', 'three'] * 3,
                           'B' : ['A', 'B', 'C'] * 4,
                           'C' : ['foo', 'foo', 'foo', 'bar', 'bar', 'bar'] * 2,
                           'D' : np.random.randn(12),
                           'E' : np.random.randn(12)})
```

```
df.head()
```

```
Out[16]:
```

	A	B	C	D	E
0	one	A	foo	-0.268448	-1.151544
1	one	B	foo	-0.380286	0.566613
2	two	C	foo	1.086837	-0.008116
3	three	A	bar	-0.890874	0.006468
4	one	B	bar	-0.265532	-0.711051

```
In [17]: print(df.loc[0]['A'])
         print("-----")
         print(df.loc[0])
```

```
one
```

```
-----
```

```

A      one
B      A
C      foo
D    -0.268448
E    -1.15154
Name: 0, dtype: object

```

```
In [18]: df['F'] = 2 * df['E'] + 3
```

```
df.head()
```

```

Out[18]:
   A B  C      D      E      F
0  one A  foo -0.268448 -1.151544  0.696911
1  one B  foo -0.380286  0.566613  4.133225
2  two C  foo  1.086837 -0.008116  2.983768
3 three A  bar -0.890874  0.006468  3.012936
4  one B  bar -0.265532 -0.711051  1.577898

```

```
In [21]: print(df['F'].mean(), df['F'].max(), df['F'].min(), df['F'].std())
```

```
3.03146567221228 5.31327151975 0.696911047668 1.3708945454410075
```

```

In [23]: #filtre les valeur positives de la colone E
df[df['E'] > 0]

```

```

Out[23]:
   A B  C      D      E      F
1  one B  foo -0.380286  0.566613  4.133225
3 three A  bar -0.890874  0.006468  3.012936
5  one C  bar  0.561156  0.028698  3.057396
8  one C  foo  0.092873  1.156636  5.313272
9  one A  bar -0.748160  0.029093  3.058187
10 two B  bar -1.681138  0.975054  4.950107
11 three C  bar  0.453790  0.334250  3.668499

```

```

In [34]: #filtre avec la chaine de caractere 'one' de la colone A et trie avec la colone F
df[df['A'] == "one"].sort_values(['F'], ascending=[False])

```

```

Out[34]:
   A B  C      D      E      F
8  one C  foo  0.092873  1.156636  5.313272
1  one B  foo -0.380286  0.566613  4.133225
9  one A  bar -0.748160  0.029093  3.058187
5  one C  bar  0.561156  0.028698  3.057396
4  one B  bar -0.265532 -0.711051  1.577898
0  one A  foo -0.268448 -1.151544  0.696911

```