Introduction à Python

Julien Grosjean

M1
Année 2016-2017 S2

Généralités

- Guido van Rossum : 1990
- Très simple, orienté néophytes
- Langage interprété ET objet
- Multi-plateforme
- Léger
- Performant
- Énorme communauté
- Libre

Pourquoi ? Pour qui ?

- Langage « lisible » facilement (épuré et rigoureux)
- Simple à apprendre et à comprendre : utilisable par des non informaticiens : mathématiciens, biologistes
- Inspiré des langages ABC, Icon, C, Modula-3, Perl, …
- Modulaire inter-connexions avec d'autres langages

- Utilisé par beaucoup de communautés scientifiques, web et notamment par : Google, Nokia, la NASA, …

Versions 2 et 3

- Elles cohabitent mais diffèrent
- Fin de la maintenance de la 2.7 en 2020
- Différences de fonctions et types de bases
- Différences dans les librairies
- De préférence, utiliser bien sûr la 3.x

Les bases : types

- `int, float, complex`
- `tuple, list, set, frozenset, dict, str, bytearray`

- Typage dynamique

Les bases : mots réservés

- `and - as - assert - break - class - continue - def - del - elif - else - except - exec - finally - for - from - global - if - import - is - lambda - not - or - pass - raise - return - try - while - with - yield`

Les bases : syntaxe

- Indentations et « `:` »
- Pas d'accolades
- « `;` » inutiles

- `# commentaires`
- POO : `self`
- Exceptions : `try, except, finally, raise`

- `import os`
- `from httplib import HTTPConnection`

Les bases : exemples

```
for i in range(5,10):
    print(i)

a = ""
while a!=None:
    a+="P"
    if len(a)==10:
        a = None
        p = {}
        z = []
print(p.keys())
z.append(1.)
print(z)
```

POO

```
class NeuronNetwork:
    def __init__(self,maxDist,nbSticks):
        self.neurons = []
        for i in range(1,nbSticks+1):
            self.neurons.append(Neuron(self,i))
        for neuron in self.neurons:
            neuron.makeConnections(maxDist,nbSticks,BASE_WEIGHT)
        self.initPath()
    def initPath(self):
        self.path = {}
    def getNeuron(self,index):
        if index-1>=0 and index<=len(self.neurons): return self.neurons[index-1]
        else: return None
    def activateNeuronPath(self,neuron1,neuron2):
        self.path[neuron1]=neuron2
    def recompenseConnections(self):
        for neuron1,neuron2 in self.path.items():
            neuron1.recompenseConnection(neuron2)
        self.initPath()
    def printAllConnections(self):
        for neuron in self.neurons: neuron.printConnections()
    def printScores(self):
        scores = {}
        for neuron in self.neurons:
            for n,s in neuron.connections.items():
                if n not in scores: scores[n]=s
                else: scores[n] = scores[n] + s
        for neuron,score in scores.items():
            print(neuron.asString(),score)
```

Références

- http://www.courspython.com/introduction-python.html
- https://docs.python.org/3/
- http://www.diveintopython3.net/