



# PROGRAMMATION WEB AVANCÉE

---

# PYTHON



- Qu'est-ce que Python ?
- Installer Python
- Interpréteur de commandes Python
- Variables
- Structures conditionnelles
- Boucles
- Exceptions
- Listes, Tuples et Dictionnaires
- Programmation Orientée Objet
- Travaux Pratiques

# QU'EST-CE QUE PYTHON ?

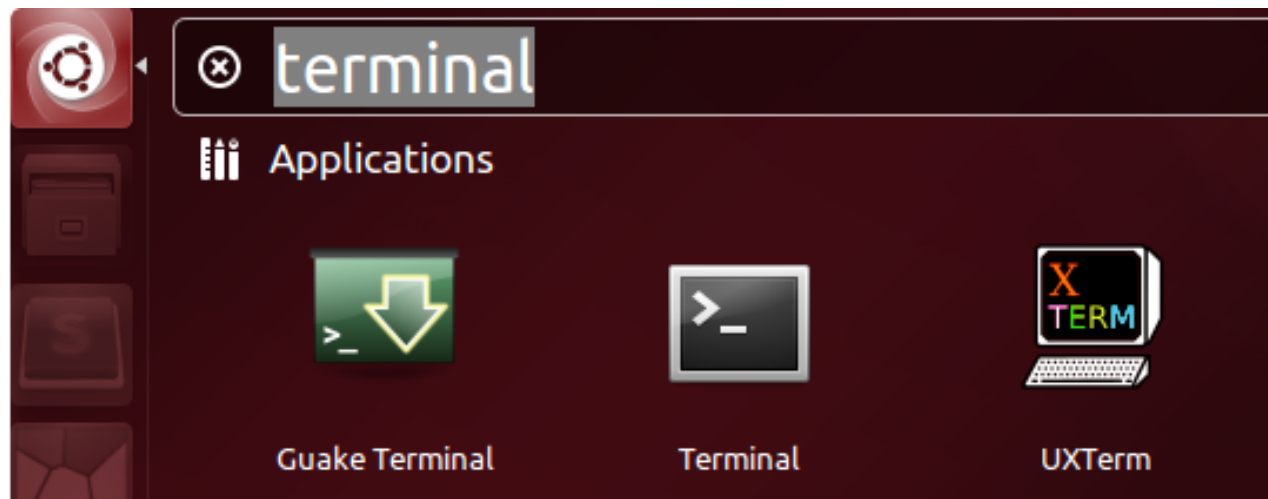
- ▶ Langage de programmation objet interprété
- ▶ Guido Van Russom, 1991
- ▶ Python 2.x, Python 3.x
- ▶ Langage de script pour automatiser des tâches
- ▶ Calcul numérique
- ▶ Développement web
- ▶ Instagram, YouTube, Dropbox, Spotify, Pinterest, ...

## INSTALLER PYTHON

- ▶ <https://www.python.org/>
- ▶ Windows : Downloads
- ▶ Linux : « python -V »
- ▶ Mac OS X

## INTERPRETEUR DE COMMANDES PYTHON

- ▶ Lancer le terminal, puis lancer la commande « python »



```
olivier@bigone: ~  
olivier@bigone:~$ python  
Python 2.7.6 (default, Mar 22 2014, 22:59:56)  
[GCC 4.8.2] on linux2  
Type "help", "copyright", "credits" or "license" for more information.  
>>> 
```

```
olivier@bigone: ~  
olivier@bigone:~$ python  
Python 2.7.6 (default, Mar 22 2014, 22:59:56)  
[GCC 4.8.2] on linux2  
Type "help", "copyright", "credits" or "license" for more information.  
>>> 3+1  
4  
>>> 
```

# VARIABLES

- ▶ Une variable est une donnée de votre programme, stockée dans votre ordinateur.
- ▶ En python, il suffit d'écrire « `nom_variable = valeur` »
- ▶ Règles de syntaxes incontournables :
  - ▶ Le nom de la variable ne peut être composée que de lettres, chiffres, majuscules, minuscules et du chiffre souligné
  - ▶ Le nom de la variable ne peut pas commencer par un chiffre
  - ▶ Python est sensible à la casse

## VARIABLES – TYPES DE DONNEES

- ▶ int, float, str, ...
- ▶ Fonction : Elle exécute un certain nombre d'instructions déjà enregistrées. « `ma_fonction(param1, param2, ..., param3)`
- ▶ Fonction **type**
- ▶ Fonction **print**

# STRUCTURES CONDITIONNELLES

- ▶ L'une des notions les plus importantes en programmation
- ▶ Syntaxe :

    If (condition):

        ...

    elif (condition):

        ...

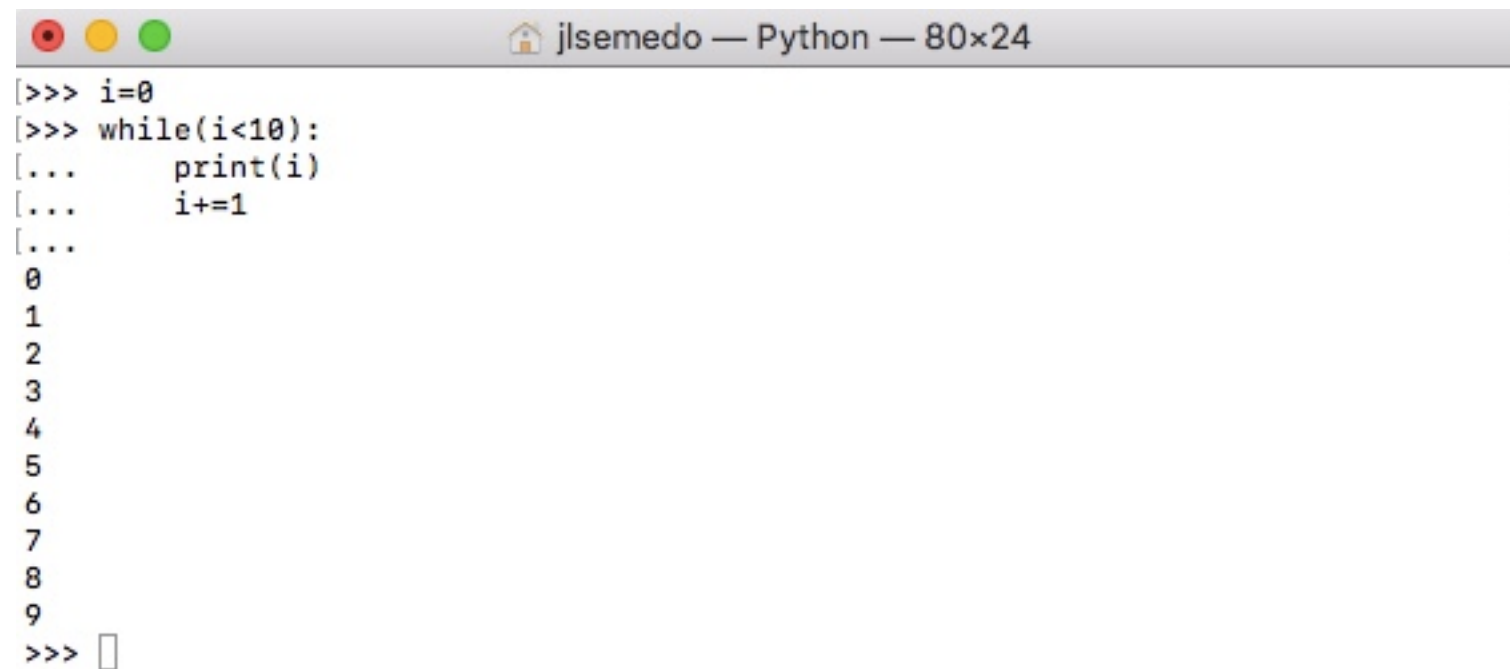
    else:

        ...



## BOUCLES

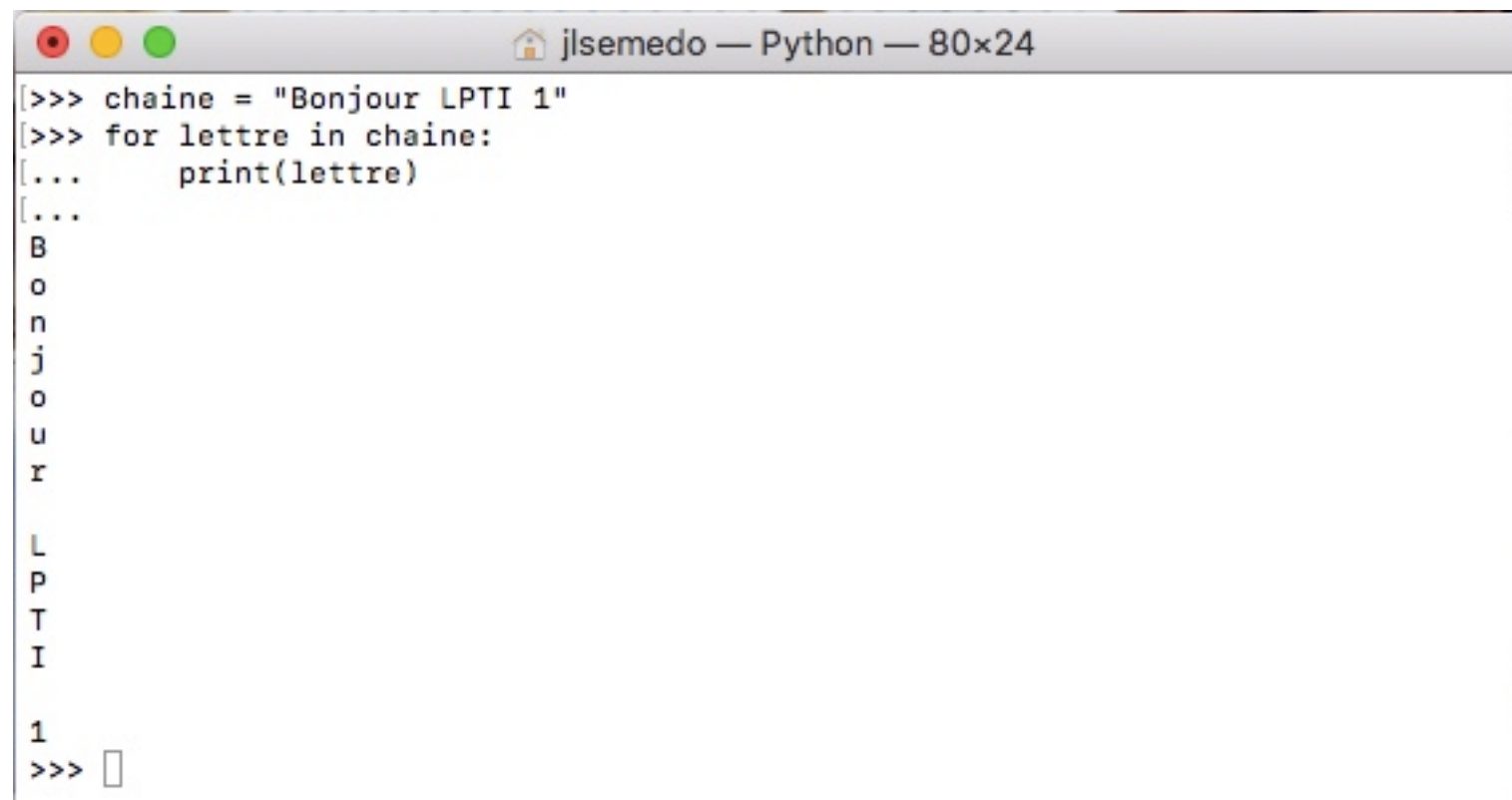
- ▶ Une boucle permet de répéter à l'infini des instructions.
- ▶ Boucle while



```
jlsemedo — Python — 80x24
[>>> i=0
[>>> while(i<10):
[...     print(i)
[...     i+=1
[...
0
1
2
3
4
5
6
7
8
9
>>> ]
```

## BOUCLES

### ► Boucle for



```
jlsemedo — Python — 80x24
[>>> chaine = "Bonjour LPTI 1"
[>>> for lettre in chaine:
[...     print(lettre)
[...
B
o
n
j
o
u
r

L
P
T
I

1
>>> ]
```

# EXCEPTIONS

- ▶ En python, on a **les erreurs de syntaxe et les exceptions**
- ▶ Quand Python rencontre une erreur dans le code, il **lève une exception**
- ▶ **Try...Except**
- ▶ **Exemples d'exceptions : NameError, TypeError, ZeroDivisionError**

## LISTES, TUPLES ET DICTIONNAIRES

- ▶ Les listes (ou list / array) sont des variables pouvant contenir plusieurs variables
- ▶ Créer une liste : « **liste = []** » ou « **liste = [1, 2, « toto »]** »
- ▶ Ajouter une valeur : « **liste.append(variable)** »
- ▶ Supprimer un item : « **del(liste[0])** »
- ▶ Inverser une liste : « **liste.reverse()** »
- ▶ Nombre d'items d'une liste : « **len(liste)** »

## LISTES, TUPLES ET DICTIONNAIRES

- ▶ Un tuple est une liste qui ne peut être modifiée
- ▶ Créer un tuple: « **t = ()** » ou « **t = (1, 2, « toto »)** »
- ▶ Afficher la valeur d'un tuple : « **t[position]** »
- ▶ Les tuples permettent les affectations multiples
- ▶ Il permet également à une fonction de renvoyer plusieurs valeurs

## LISTES, TUPLES ET DICTIONNAIRES

- ▶ Un dictionnaire est une liste mais au lieu d'utiliser des index on utilise des clés
- ▶ Créer un dictionnaire: « **d = {}** »
- ▶ Ajouter une valeur : « **d['classe'] = 'LPTI 1'** »
- ▶ Méthode **get(key, default)** permet de récupérer une valeur et si la clé est introuvable, renvoie une valeur par défaut

## P00 – CLASSES

```
Voiture
1  #!/usr/bin/env python3.4
2  # _ coding: utf-8 -*-
3
4  class Voiture:
5
6      def __init__(self):
7          self.nom = "Car rapide"
8
9      def get_nom(self):
10         return self.nom
11
12     def set_nom(self, nom):
13         self.nom = nom
```

- ▶ Classe « Voiture »
- ▶ La méthode « `__init__()` » est appelée à la création d'un objet
- ▶ « nom » est un attribut de classe
- ▶ « `get_nom` » et « `set_nom` » sont des méthodes
- ▶ « `self.nom` » est une manière de stocker une information dans la classe

# P00 – OBJETS

- ▶ Pour créer un objet, on fait « `mon_objet = Voiture()` »
- ▶ Pour donner une valeur à l'attribut « `nom` », on a 2 façons de procéder dans ce cas précis :
  - ▶ `mon_objet.nom = « LPTI 1 »`
  - ▶ `mon_objet.set_nom(« LPTI 1 »)`
- ▶ Fonction **dir** donne un aperçu des méthodes d'un objet



TEXTE

---