

Langages et Compilateurs : Introduction

Matthieu Amiguet

2008 – 2009



1 Plan du cours

2 Pourquoi Python ?

3 Compilateurs

Plan

3

- Le langage Python
- Constructions avancées en Python
- Programmation fonctionnelle en Python
 - ↪ Test
- Compilateurs
 - ↪ Projet "Compilateur"
- Introduction à la théorie des langages et des automates
 - ↪ Test

1 Plan du cours

2 Pourquoi Python ?

3 Compilateurs

Qu'est-ce que python ?

5

- Langage de programmation
 - avec ramasse-miettes (*garbage collection*),
 - gestion des exceptions
- Multi-paradigme
 - Impératif structuré
 - Orienté objet
 - Fonctionnel (dans une certaine mesure)
- Langage qui semble interprété
 - Au fait, compilé à la volée en *bytecode* pour une machine virtuelle...
- Disponible sous Windows, Linux, MacOS, ...
 - Mais aussi implémenté en Java, .Net, et... Python!
- Licence libre et assez permissive

Pourquoi python pour ce cours ?

6

- Langage de haut niveau avec fonctionnalités avancées
 - Structures de données avancées
 - Itérateurs, générateurs, décorateurs
 - "Context managers"
 - Programmation fonctionnelle
 - ...
- ↔ Comparable à C# (ou Java) mais avec des choix de conception assez différents !
- Richesse des bibliothèques
- Excellent langage de prototypage rapide !

Python est "économique"

7

- "Hello, World" en python :

```
print "Hello , world!"
```

- Échanger la valeur de deux variables

```
a,b = b,a
```

- Retourner plusieurs valeurs

```
def pred_succ(x):
    return x-1, x+1

p, s = pred_succ(1)
print "%d is surrounded by %d and %d" % (1,p ← ,s)
```

La philosophie "Batteries included"

8

- Un serveur web en trois lignes :

```
import BaseHTTPServer, SimpleHTTPServer
server = BaseHTTPServer.HTTPServer(\
    ('',8080),\
    SimpleHTTPServer.SimpleHTTPRequestHandler\
)
server.serve_forever()
```

La philosophie "Batteries included" (Suite)

9

- Récupération et traitement d'un *feed RSS*

```
import urllib, sys, xml.dom.minidom
address = 'http://www.sebsauvage.net/rss/updates.xml'
document = \
    xml.dom.minidom.parse(urllib.urlopen(address))
for item in document.getElementsByTagName('item'):
    title = item.\
        getElementsByTagName('title')[0].\
            firstChild.data
    print "Title:", title.encode('latin-1', 'replace')
```

Python – pour quels usages ?

10

- Administration système
- Prototypage rapide
- Développement "agile"/orienté par les tests/...
- Développement web (Zope, Django, ...)
- Outils de recherche scientifique
- Langage d'extension
- Applications pédagogiques
- ...

Les limites de python

11

Python n'est pas forcément adapté lorsque...

- La vitesse pure est un critère prioritaire
- On veut un compilateur contraignant
- On ne veut pas diffuser le code source (quoique...)
- On ne veut/peut pas installer l'interpréteur
- ...

Qui utilise python ?

12

- Google (3e langage derrière C++ et Java)
 - YouTube est presque entièrement fait en python !
 - Yahoo
 - NASA
 - "Ubuntu prefers the community to contribute work in Python." (<http://www.ubuntu.com/community/developerzone/bounties>)
 - ...
- (<http://www.python.org/about/quotes/>)

Un mot sur la version...

13

- Python est au seuil d'une mutation importante : la version 3.0
 - beaucoup de changements et *rupture de la compatibilité arrière*
- En même temps sortira une version "de transition" 2.6
- Au moment du début du cours, ces versions sont encore en beta
- Le cours se basera donc sur python 2.5

1 Plan du cours

2 Pourquoi Python ?

3 **Compilateurs**

Compilateurs

15

- Tout programmeur utilise couramment un (des ?) compilateur(s)
 - Mais on ne sait pas toujours "comment c'est fait dedans"
- Une étude des techniques de compilation permet
 - D'écrire son propre compilateur/interpréteur/...
 - De mieux comprendre cet outil primordial...
 - ... et donc de mieux l'utiliser !
- L'étude des compilateurs nous donnera aussi l'occasion d'aborder
 - Les questions d'analyse de langages
 - La théorie des automates (dont la machine de Turing)
