

Pourquoi Python ?

FELD Boris - 28 novembre 2011
La Cantine

Qui suis-je ?

- * Étudiant en école d'ingénieur (UTBM)
- * En stage Assurance-Qualité chez Dailymotion
- * Développe en Python depuis 2 ans
- * Twitter: @lothiraldan

Python késako ?

- * Créé en 1990 par Guido Von Rossum
- * Nommé en hommage aux Monty Python
- * 8ème langage le plus populaire selon le Tiobe Index
- * 13 versions majeures de nos jours

Caractéristiques

- * Typage fort
- * Typage dynamique
- * Byte-compilé

Python est utilisé

- * Disqus
- * Eve Online
- * Mozilla (addons.mozilla.org)
- * Inkspase
- * De nombreux outils que vous utilisez tous les jours.



SIMPLICITY REQUIRES
COURAGE

1) La rapidité

Hello World JAVA

```
public class HelloWorld
{
    public static void main(String[] args)
    {
        System.out.println("Goodbye, World!");
    }
}
```

Hello World C

```
#include <stdlib.h>
#include <stdio.h>

int main(void)
{
    printf("Goodbye, World!\n");
    return EXIT_SUCCESS;
}
```


Hello World Python

```
print "Goodbye, Wor1d!"
```

Lancer les exemples

* JAVA :

- * `$> javac HelloWorld.java`
- * `$> java HelloWorld`

* C :

- * `$> gcc hello_world.c -o hello_world`
- * `$> ./hello_world`

* Python :

- * `$> python hello_world.py`

Simple...

- * Langage concis
- * Permet un développement rapide
- * Multi-paradigme

...mais pas simpliste

- * Fonctionnalités avancées:

- * Décorateurs

- * Générateurs

- * Listes compréhensives

- * Descripteurs

Les truc vraiment cool

- * Pas de gestion « à la main » de la mémoire
- * Des structures de données faciles à utiliser
- * Introspection avancée

and	assert	break	class	continue	def	False
del	elif	else	except	exec	finally	None
for	from	global	if	import	in	
is	lambda	not	or	pass	print	
raise	return	try	while	yield	True	

2) La syntaxe

Les types de base

```
an_int = 42
```

```
a_float = 3.14
```

```
a_list = [1, 2, 3]
```

```
a_dict = {'key1': 'value1', 'key2': 'value3'}
```

```
a_complex = complex(1, 2) #Ou complex('1+2j')
```

L'indentation compte

```
/* Warning: confusing C code! */
```

```
if (some condition)
    if (another condition)
        do_something(fancy);
else
    this_sucks(badluck);
```

```
# Warning: awesome python
```

```
if some condition:
    if another condition:
        do_something(fancy)
    else:
        this_dont_sucks(goodluck)
```


Fonctions

```
def fib(n):  
    if n < 2:  
        return n  
    else:  
        return fib(n-1) + fib(n-2)
```

```
fib(10)
```

Classes

```
class Message(object):  
    def __init__(self, message = ''):  
        self.message = message  
    def print_message(self):  
        print self.message
```

```
m = Message("Message example")  
m.print_message()
```

Import

```
import math  
math.sqrt(9)
```

Structure de données

```
a_list = []  
a_number = 5
```

```
a_list.append(a_number)  
another_number = a_list[0]
```


Syntaxe

- * Une syntaxe simple à écrire et à **LIRE**
- * L'indentation obligatoire rend le code clair à écrire et surtout à lire
- * Pas de points-virgules ni d'accolades
- * Très proche du pseudo-code



3) La librairie
standard

Types de données

Structures de données



- * Dates
- * Chaînes et les chaînes en unicode
- * Les ensembles (au sens mathématique)
- * Nombres complexes

Formats de fichiers



- * La librairie standard permet de lire et écrire des fichiers dans ces formats:
- * JSON
- * INI
- * CSV
- * XML
- * HTML

Réseau/Internet



- * Socket BSD
- * Serveur HTML/CGI simple
- * Gérer/Parser des requêtes HTTP
- * SMTP
- * Client/Serveur XML-RPC

Outils

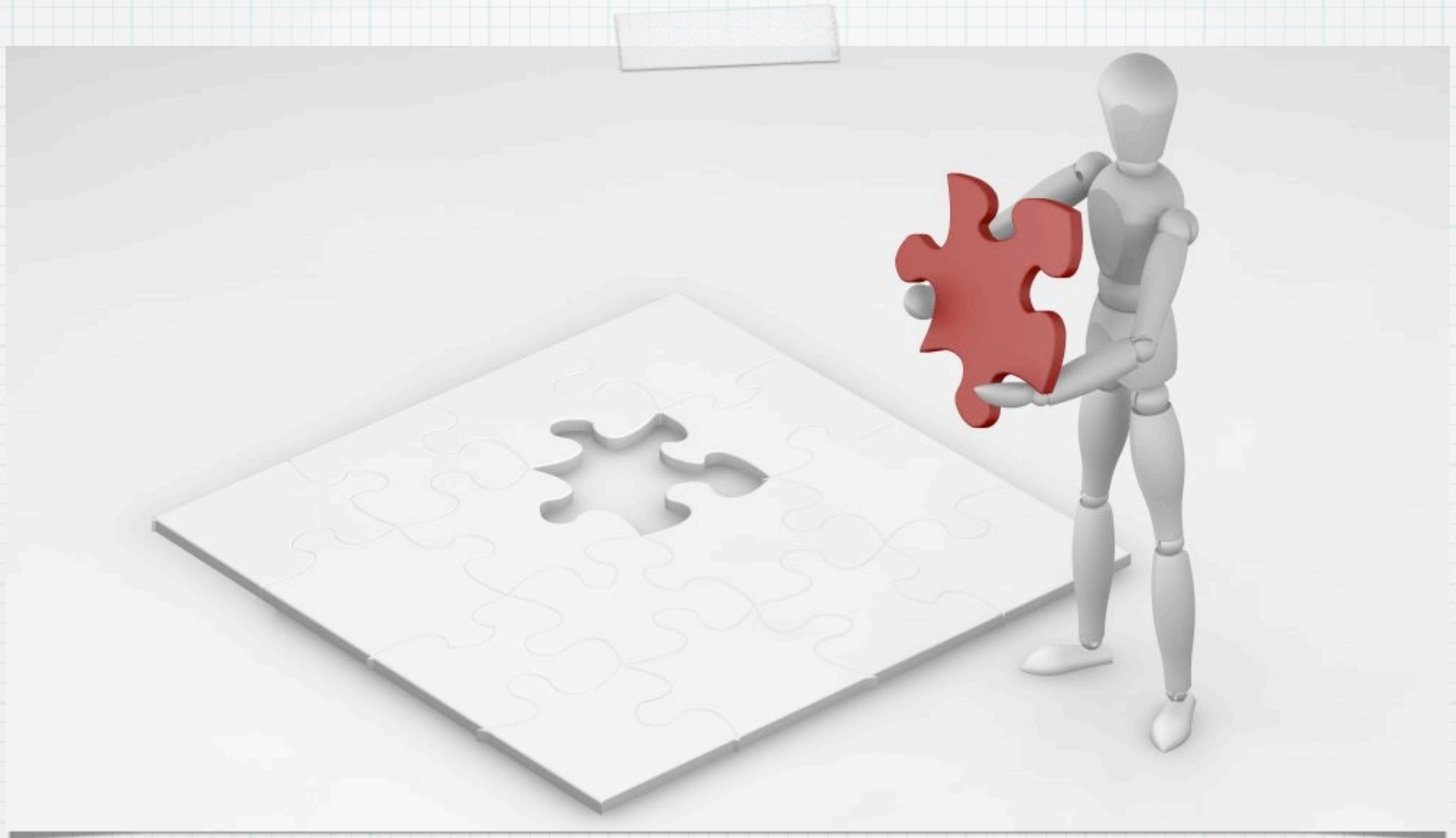


- * Débogueur
- * Profiling
- * Librairie de Test Unitaires

Autre



- * Cryptographie
- * Compression
- * Regex
- * Sérialisation
- * Encore plus ? <http://docs.python.org/library/>



4) Intuitif

Python est intuitif

- * En python tout est objet...
- * On encourage une seule façon de faire...
- * Et cela permet d'acquérir rapidement des réflexes de programmation.

Un langage objet

- * `file.readline()`
- * `file.read(size)`
- * `" abc ".strip()`

Parcours d'une structure de données

- * Liste:

- * `for elem in [1, 2, 3]`

- * Dictionnaire:

- * `for key in {'key1': 'val1', 'key2': 'val2'}`

- * Chaîne:

- * `for letter in 'mystring'`

Longueur d'une structure de donnée

- * Liste:

- * `len([1, 2, 3])`

- * Dictionnaire:

- * `len({'key1': 'value1', 'key2': 'value'})`

- * Chaîne:

- * `len('mystring')`

Vérifier si une valeur est présente

- * Liste:

- * `4 in [1, 2, 3]`

- * Dictionnaire:

- * `'key3' in {'key1': 'val1', 'key2': 'val'}`

- * Chaîne:

- * `'x' in 'string'`



5) Extensibilité

Interpréteurs

- * CPython, l'interpréteur de référence
- * Jython
- * IronPython
- * PYPY compilateur JIT

Extensibilité

- * Problèmes de performances:
 - * Écrivez des modules en C
- * Besoin d'interfacer python:
 - * Avec Java, utilisez Jython
 - * Avec .NET, utilisez IronPython

Conclusion

- * C'est un véritable plaisir de coder en Python
- * Donnez lui sa chance
- * Python vous rendra de grands services même en tant que langage de script

Demo time !

Récupérer le dernier tweet