

Python le Radici

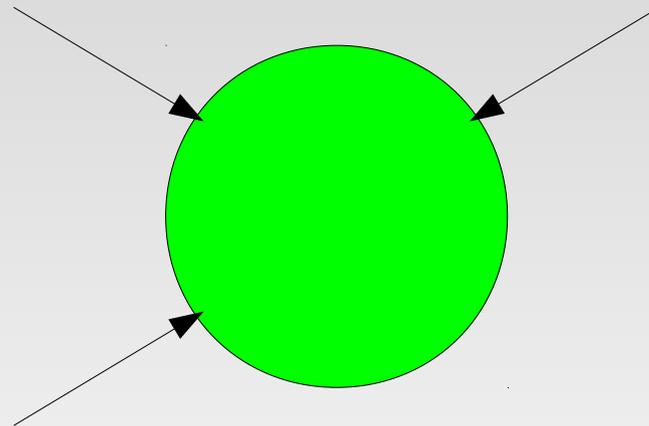
Ottobre 2006



Include le seguenti caratteristiche:

Programmazione ad Oggetti
(C++, Modula, ecc..)

Programmazione Funzionale
(Scheme, ecc..)



Linguaggio di Scripting
(Perl, Tcl, ecc..)



Venezia Free Software

Loris Michielutti
Loris Michielutti

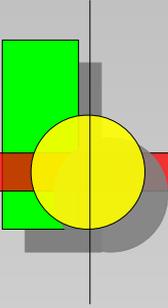
Users Group





perche' Python

- E' facile da imparare
- Porta con se un semplice approccio alla programmazione ad oggetti
- Ha una efficiente struttura dati
- Una elegante sintassi
- E' ideale per lo scripting
- Si possono sviluppare in modo rapido applicazioni anche complesse



perche' Python



Python e' stato creato da Guido Van Rossum nel 1990

- ricercatore di Amsterdam lavorava ad un un progetto di un linguaggio di programmazione a scopi didattici di nome ABC

Viene definito un linguaggio di scripting orientato agli oggetti

- porta con se la flessibilita' e la semplicita' di questi linguaggi, la potenza di elaborazione, la ricchezza di funzioni dei linguaggi di programmazione di sistema

E' Libero

- libero da licenze
condivide in pieno la filosofia di Open Source

E' Portabile

- e' stato scritto in ANSI C
questo ha permesso la facile trasportabilita' in diversi sistemi quali:
Unix, Linux, MS-DOS, MS-Windows, Macintosh, Amiga, BeOS, OS/2, VMS, QNX
recentemente e' stato scritto un interprete anche in java e anche sui Palmari



E' Veloce

- nonostante sia un linguaggio interpretato il bytecode e' molto efficiente. in realta' python e' sia interpretato che compilato.
- la prima volta che un modulo viene richiesto questo viene compilato e convertito in bytecode con lo stesso nome del file ma con estensione `.pyc`
- molte strutture dati e funzioni sono componenti intrinseche del linguaggio. chiamate "built-in types and tools" sono state sviluppate con molta accuratezza per ottenere il massimo dell'efficienza (es: gestione dei file)

Garbage collection

- la gestione della memoria e' automatica, come in java il programmatore non se ne deve piu' preoccupare

Sintassi chiara

- presenta una sintassi pulita e sintetica. l'indentazione diventa l'unico strumento per strutturare il codice questo permette un apprendimento piu' veloce e una maggiore facilitata' nel leggere il codice scritto da altri.



E' ricco di librerie

- la dotazione standard e quella che si trova in internet (HTML, PDF, XML, formati grafici, CGI e anche interi web servers)
- questo sta convincendo molti grandi attori del mercato informatico ad impiegarlo nelle loro applicazioni:
 - * Red Hat, Gentoo (tool di installazione).
 - * Infoseek (ricerca sul web).
 - * Yahoo! (servizi di internet).
 - * La NASA (sistemi di controllo delle proprie missioni).
 - * Zope (interi applicazioni)

Esistono compilatori Python che producono codice per altre macchine virtuali:

- Jython produce bytecode per una JVM (file .class)
- Python.NET produce il codice-intermedio MSIL tipico della piattaforma .NET
- C-Python e' la versione-base essendo stata scritta in C



L'evoluzione

- 2.2 (12/2001) ... 2.2.3 (05/2003) il riferimento
- 2.3 (07/2003) ... 2.3.5 (02/2005) ~ 30% piu' veloce
- 2.4 (11/2004) ... 2.4.4 (09/2006) ~ 5% piu' veloce
- 2.5 (09/2006) ... ~ 10% piu' veloce

Le novita' della 2.5

- pieno supporto a Resource Allocation Is Initialization
- import relativo e assoluto
- unificazione try/except/finally
- nuove builtins any/all, dict.__missing__
- nuove librerie standard: ctypes, xml.tree, ecc

esempio: "RAII"

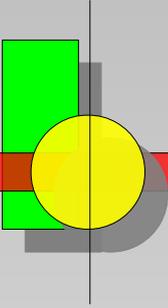
2.4

- resource: **allocazione della risorsa**
- try: **uso della risorsa**
- finally: **rilascio della risorsa**

2.5

- with : **allocazione della risorsa:**
uso della risorsa

(rilascio automatico all'uscita del blocco)



perche' Python



Per finire agli amici del Perl posso offrire queste definizioni:

Perl: "post-moderno"

- (come e' stato descritto dal suo stesso inventore Larry Wall)
ricco, gioioso, caotico nella sintassi e nella semantica, complesso, ridondante
- il suo motto ("C'e' piu' di un modo di farlo!") .

Python: "neo-classico"

- snello, limpido, cristallino
inteso ad offrire un modo (clamorosamente "giusto")
di svolgere un qualsiasi compito.

Link di riferimento:

- <http://www.python.org/> # sito ufficiale
- <http://www.planetpython.org/> # discussioni giornaliera
- <http://wiki.python.org/moin/Documentation> # documentazione
- <http://groups.google.com/group/it.comp.lang.python> # gruppo di discussione
- http://www.faqs.com/knowledge_base/index.phtml/fid/199 # faq
- <http://lists.python.it/pipermail/python/> # mail-list archivio