



A spasso con Python 2006

Eleganza, potenza, semplicità.

Data: 28 ottobre 2006

Relatore: Massimo Masson

BLUG - Belluno Linux User Group

<http://belluno.linux.it/>

<http://collaboraplone.homelinux.net/>





The flying circus

Ovvero: non-indice della presentazione

- Non è un corso di Python
 - non c'è il tempo...
- Non è un elenco esaustivo di features
 - non avrebbe significato...
- Non è un approfondimento di un tema specifico
 - non interesserebbe a nessuno in questa sede...
- Ma allora, cos'è?
 - Una chiacchierata “a ruota libera”, con la speranza di suscitare interesse e destare curiosità...

And now for something completely different..



- Software libero
- Multiplatforma
 - Linux, *BSD, Windows, Macintosh, ...
- Elegante
 - Leggibile
 - Mantenibile
- Potente, scalabile, modulare
- Semplice
- Programmazione procedurale, OOP, funzionale
- “Batterie incluse”
- **Non ha nulla a che fare con i rettili** (*Monty Python's Flying Circus*)

Ministry of silly walks...

Ovvero: molti modi di trarre vantaggi da Python

- Alcune note distribuzioni GNU/Linux approfittano di Python...
- UBUNTU <http://www.ubuntu.com/developers/bounties>
 - *Ubuntu prefer the community to contribute work in Python. We develop our own tools and scripts in Python and it's much easier for us to integrate your work if you use the same platform. Here are some Python-related projects that are up for bounties;*
 - *Python scripting everywhere: Python is a great glue language. Ubuntu is open to requests for funding work that needs to be done to make Python the most widespread common scripting language on the net. Please contact us with your ideas and proposals. Bounties will be offered on Python scripting interfaces for the following tools.*
- GENTOO
 - **Portage**, il cuore di Gentoo Linux, un sistema avanzato di gestione dei pacchetti che richiama il sistema dei ports in stile *BSD



Nobody expects...

- L'indentazione fa parte della sintassi
 - Migliora la leggibilità del codice
 - Rende il codice omogeneo tra gli sviluppatori
- Tutto è “oggetto” (nota sulle *new-style classes...*)
 - Ad esempio, le funzioni sono oggetti. Non hanno “tipo” di ritorno, ritornano *oggetti* (che hanno implicitamente un tipo)
- Moduli e Packages (collezioni di moduli)
- Gestione eccezioni (*try/except/finally - raise*)
- Ottimo per lo sviluppo rapido di applicazioni e per prototipi
- Ottima integrazione con C/C++ (e Java con Jython)

Principali “tipi di dato”

- Numeri (interi, long in precisione arbitraria, complessi)
- Stringhe (escape, multilinea, unicode, ...)
- Liste e tuple:
`['a', 'b', 3] (1, 2, 'c')`
- Dizionari (hash, array associativi) e insiemi
`{'a': 1, 'b': 2}`
- Classi
- In verità le variabili sono dei riferimenti ad oggetti. Vanno inizializzate ma non hanno proprio “tipo” in quanto dipende dall'oggetto cui si riferiscono. La tipizzazione è:
 - *dinamica*: il tipo è scoperto in esecuzione;
 - *forte*: i tipi sono imposti (no conversioni automatiche di tipo)

Qualcosa di Python: list comprehension

[map-expr for element in src-list if felter-expr]

- Esempi tratti da “*dive into python*”:

```
>>> li = ["a", "mpilgrim", "foo", "b", "c", "b", "d", "d"]
```

```
>>> [elem for elem in li if len(elem) > 1]
```

```
['mpilgrim', 'foo']
```

```
>>> [elem for elem in li if elem != "b"]
```

```
['a', 'mpilgrim', 'foo', 'c', 'd', 'd']
```

- Connessione a dbms...

```
params = {"server":"mpilgrim", "database":"master", "uid":"sa",  
         "pwd":"secret"}
```

```
return ";".join(["%s=%s" % (k, v) for k, v in params.items()])
```

Qualcosa di Python: duck typing...

- “*If it looks like a duck and quacks like a duck, it must be a duck.*”
(se assomiglia ad un papero, e stanrazza come un papero, dev'essere un papero...)
- Stile di programmazione basato sulla determinazione del tipo di un oggetto tramite ispezione dinamica dei suoi metodi ed attributi piuttosto che sulle relazioni predefinite tra tipi.
 - Interfacce
 - Sostituzioni polimorfiche
- EAFP: *Easier to Ask Forgiveness than Permission*
(è più facile scusarsi che chiedere permesso)
- Esempio “standard”: classi *file like*
 - Implementazione di metodi che l'oggetto è in grado di gestire;
 - Utilizzo dell'introspezione per determinare se/cosa un oggetto può fare.

```
hasattr()
```


Alcuni esempi di Software...

- **BitTorrent**, l'originale ed alcuni derivati
- **Blender**, modellatore 3D open source
- **Chandler**, personal information manager con calendario, email, compiti, note
- **Civilization IV**, videogioco
- **Django**, framework per il web stile "MTV" (model, template, view)
- **EVE Online**, gioco di ruolo fantascientifico on line
- **Indian Ocean Tsunami Detector**, software per prevenzione tsunami via telefono
- **Mailman**, un popolare pacchetto per gestire le mailing list
- **MoinMoin**, un motore "wiki"
- **OpenRPG**, tavolo virtuale per giochi di ruolo su internet
- **Plone**, Content Management System basato su *Zope*
- **Trac**, bug/issue tracking database, integrato con *MoinMoin* e *Subversion*
- **TurboGears**, framework applicativo composto da *CherryPy*, *SQLObject*, *MochiKit*, *Kid*
- **ViewVC**, interfaccia basata su web per consultare repository CVS
- **Zope**, piattaforma per applicazioni web
- ... molto altro ...

Alcuni noti packages...

- **Matplotlib**, funzioni matematiche e grafici stile MATLAB
- **Numeric Python**: supporto a vettori e matrici multidimensionali
- **SciPy**, libreria di funzioni scientifiche e numeriche
- **Biopython**, biologia molecolare
- **PyOpenGL**, gestione 3d (OpenGL) con Python
- **Soya 3D**, motore per giochi 3D
- **Pygame**, per sviluppare giochi
- **PIL** (Python Imaging Library), gestione immagini in vari formati
- **PyGTK**, libreria GUI basata su GTK+ multiplatforma (rif. anche librerie GNOME)
- **PyQt**, libreria GUI multiplatforma basata su Qt (rif. anche librerie KDE)
- **wxPython**, libreria GUI multiplatforma, potente e popolare, basata su wxWidgets
- **py2exe**, compilatore per generare applicazioni autonome sotto Windows
- **CherryPy**, Django, Quixote, Topsite Templating System: web application frameworks
- **TurboGears**, web framework che combina CherryPy, SQLAlchemy, Kid
- **ZODB**, database object-oriented specifico per Python, fornito di default con Zope
- **Cheetah**, generazione di codice da template
- **mod_python**, modulo per il web server Apache
- **Twisted**, networking framework
- **...molto altro...**

Principali implementazioni

- **Python:** implementazione di riferimento, nota anche come CPython
- **Jython:** Python codificato in Java
- **IronPython:** Python per le piattaforme Mono e .NET
- **Stackless Python:** sperimentale, non usa lo stack C
- **Psyco:** JIT (Just In Time) compiler per CPython
- **PyPy:** Python scritto in Python
- **Parrot:** Macchina virtuale sviluppata come "runtime" per perl 6, dovrebbe supportare anche altri linguaggi dinamici come Python, Ruby, Tcl, etc. Attualmente esegue un sottoinsieme di codice Python
- **Movable Python:** distribuzione alternativa per Windows, può funzionare da una penna USB fornendo un ambiente di programmazione portabile

Domande? Don't Panic!



**The Ultimate answer to Life,
the Universe, and Everything is...**

(Your're not going to like it...)

Is...

42

(Deep Thought)

Douglas Adams
The Hitchhiker's Guide to the Galaxy