

# Python – Corso base

## Lezione 1

### **ConoscereLinux**

Luca Zomparelli



**CONOSCERE  
LINUX**

# Scopo del corso



- Cenni storici
- Perché Python
- Quale versione
- Come partire
- Filosofia
- Sintassi
- Struttura di un programma
- Cenni sulle librerie base e non
- Realizzare un semplice videogioco

# Cenni storici

- Ideatore Guido Van Rossum (Olanda 1982)
- Nato alla fine degli anni 90
- Il nome si ispira ai Monty Python
- Guido ha lavorato alla Google dal 2012 lavora alla Dropbox (sviluppato in gran parte in Python)
- Obbiettivi di Python:
  - Semplice
  - Open source
  - Linguaggio naturale
  - Tempi di sviluppo brevi



# Perché Python

The background features two interlocking puzzle pieces. The piece on the left is blue and the one on the right is yellow. Both pieces have a cutout in the shape of the Python logo, which is a stylized snake with its head raised. The pieces are slightly offset from each other, creating a 3D effect.

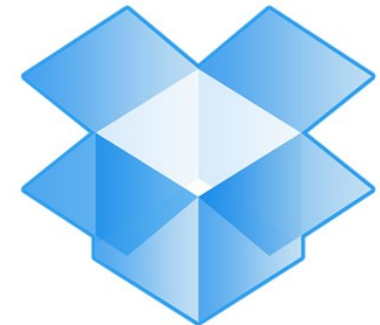
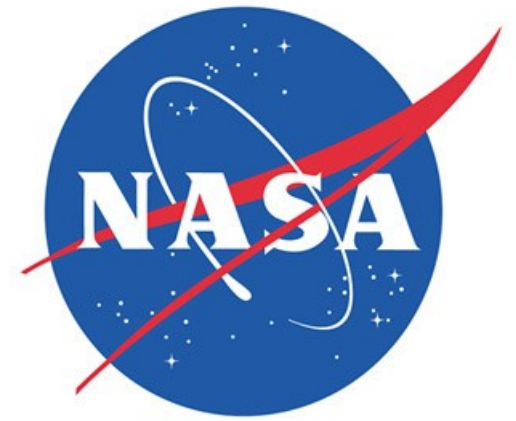
- E' Open Source
- Multi piattaforma
- Sintassi semplice
- Linguaggio di alto livello
- È indirizzato agli oggetti
- Estensibile / incorporabile
- Comunità vastissima

# Perché Python (segue)

- Numerosissime librerie
- Varietà di utilizzi
- Elevate prestazioni
- È usato da aziende importanti



# Chi usa Python



**Dropbox**



# Versioni di Python



- Python 2.5
  - Versione più vecchia
  - Alcune librerie vecchie supportano solo questa
- Python 2.6 / Python 2.7
  - Versioni bridge
- Python 3.x
  - Versione nuova, il futuro
  - Molte librerie storiche sono state portate
  - Supporto migliorato per l'UNICODE
  - Molte gestioni migliorate
- Python 2.7 vs Python 3.x

# Installare Python

- Su Linux 99% è già installato
  - `sudo apt-get install python-xx`
- Su Windows c'è l'installer (interprete + IDLE)
- Mac è presintallato (documentarsi se serve)
- Altri dispositivi: spesso si trova sui vari market place



# Filosofia di Python



- Programmazione indirizzata agli oggetti
- Ambiente aperto (si può accedere a tutto)
- Sintassi semplice e ordinata
- Ricco di convenzioni e con poche restrizioni
- Batterie incluse
- Moduli per tutto super ottimizzati

# Python Shell

- E' uno strumento comodissimo!

```
>>> print('Hallo world!')
```

```
Hallo world!
```

- Si può usare come calcolatrice

```
>>> 27 + (3 * 3)
```

```
36
```

- Lo si può lanciare anche da riga di comando  
ubuntu@ubuntu:~\$ python

# Creare un modulo

- Che cos'è un modulo (nome\_file.py)
- Come si crea (consigli per altre piattaforme)
- Eseguire un modulo da IDLE
- Eseguire un modulo da shell

# Sintassi

- Assegnazione variabili
- `raw_input(...)` \*
- `input(...)` \*\*
- `print(...)`

\*Python 2.xx  
\*\*Python 3.xx

# Sintassi (segue)

- `if` `<condition>`:  
    `<some code>`
- `elif` `<condition>`:  
    `<some code>`
- `else`:  
    `<some code>`
- Formattazione delle stringhe



# Fine

- Conclusioni
- Domande