

Perché programmare?

Capitolo 1



Python for Informatics: Exploring Information
www.pythonlearn.com

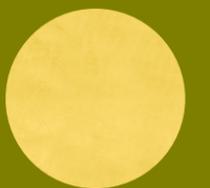


I computer vogliono essere d'aiuto...

- I computer sono fatti per uno scopo - fare cose per noi
- Dobbiamo parlare il loro linguaggio per comunicargli quello che vogliamo che facciano
- Per gli Utenti è facile - qualcuno ha già inserito molti programmi diversi (e istruzioni) nei computer e l'utente deve solo scegliere quello che vuole usare

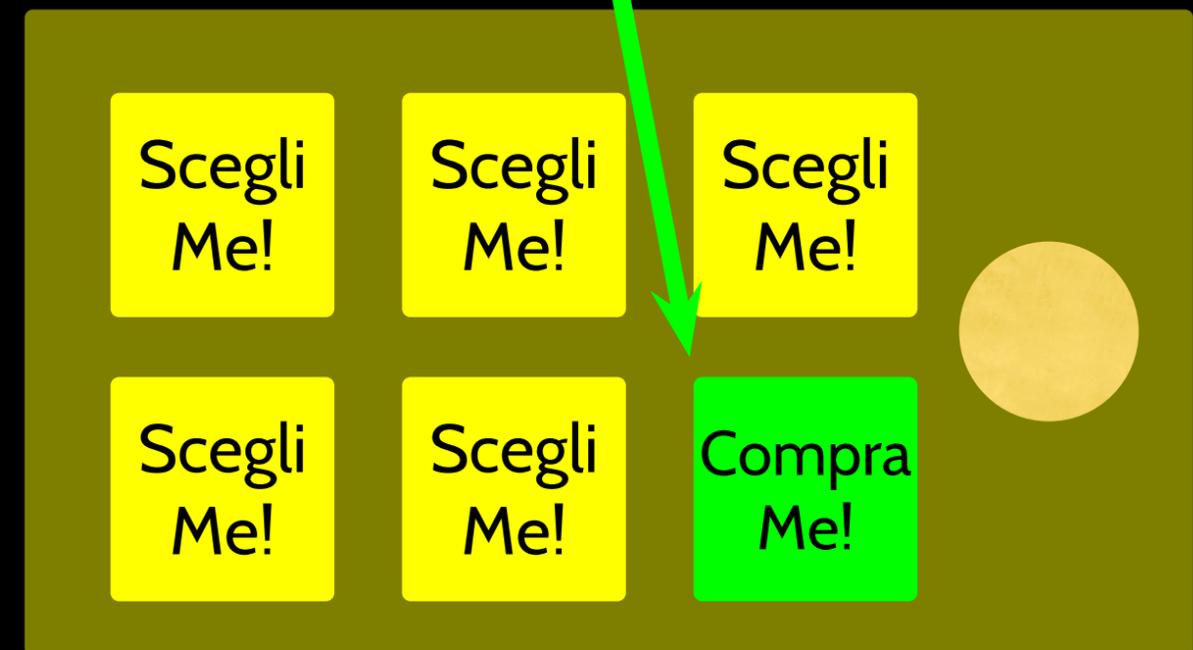
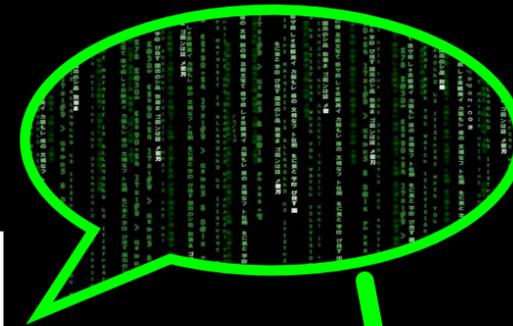


Cosa
faccio
ora?



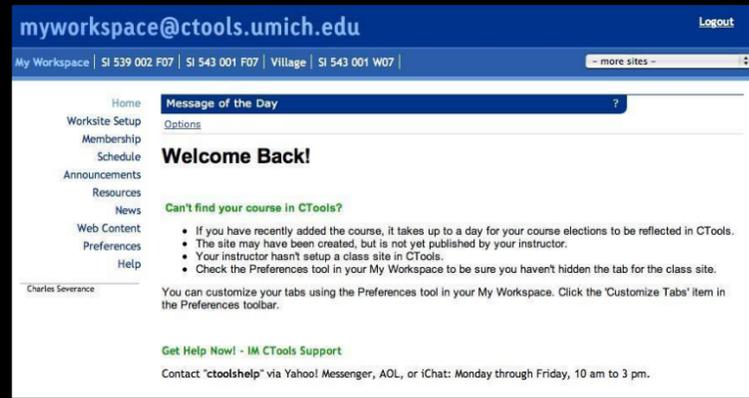
I Programmatori anticipano le necessità

- Le applicazioni per iPhone sono un mercato
- Le applicazioni per iPhone hanno oltre 3 download
- Alcuni programmatori hanno lasciato i loro impieghi per diventare sviluppatori iPhone a tempo pieno.
- I programmatori sanno **come programmare**

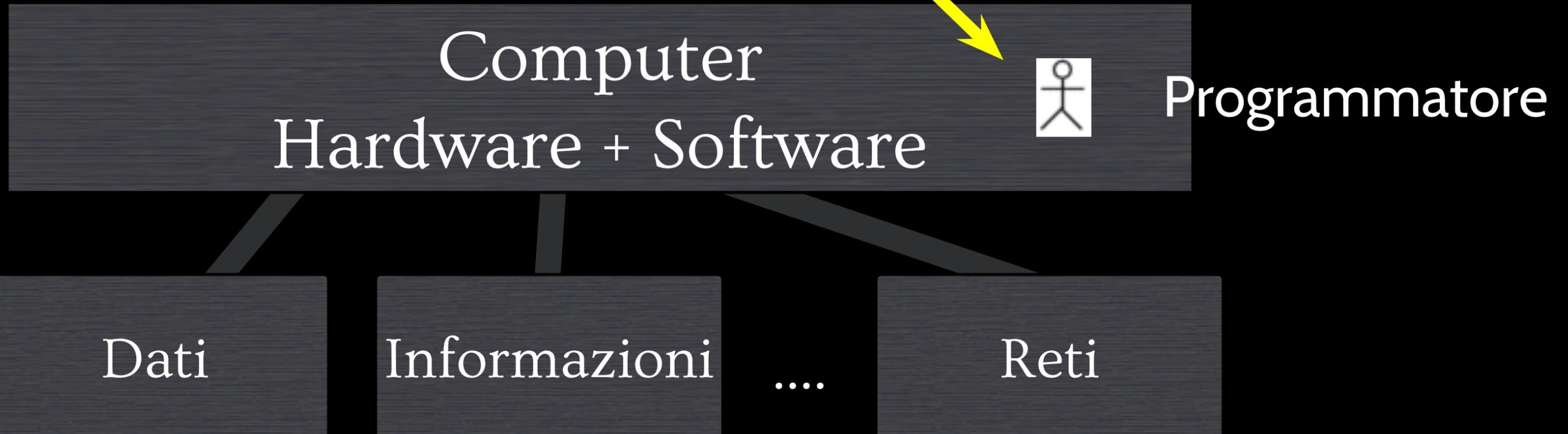


Utenti contro programmatori

- Gli utenti vedono i computer come degli strumenti - word processor, spreadsheet, mappe, todo list, ecc.
- I programmatori imparano la struttura e i linguaggi del computer
- I programmatori hanno a disposizione degli strumenti che gli permettono di svilupparne di nuovi
- I programmatori sviluppano sia strumenti per molte persone che piccoli programmi di utilità per se stessi per automatizzare alcune operazioni



Utente



Dal punto di vista dello sviluppatore di software, noi creiamo il software. Gli utenti finali (chi commissiona il software/utilizzatori) sono i nostri datori di lavoro - coloro che dobbiamo accontentare - spesso ci pagano quando sono soddisfatti. Ma dati, informazioni e reti sono il nostro problema da risolvere per loro. Hardware e software sono nostri alleati in questa sfida.

Perché programmare?

- Per eseguire alcune operazioni - siamo sia utenti che programmatori
 - › Ripulire dati derivanti da questionari
- Per produrre qualcosa che altri useranno - il lavoro del programmatore
 - › Sistemare un problema nel software Sakai
 - › Aggiungere un libro degli ospiti in un sito web

Cos'è un codice? Software? Un programma?

- Una sequenza di istruzioni memorizzate
 - › È una piccola porzione della nostra intelligenza dentro al computer
 - › È una piccola porzione della nostra intelligenza che possiamo condividere con altri - capiamo qualcosa e quindi la codifichiamo e la rendiamo disponibile ad altri che risparmieranno tempo ed energie
- Un pezzo di arte creativa - particolarmente quando facciamo un buon lavoro sulla interfaccia utente (User Experience)

Programmi per esseri umani...



<http://www.youtube.com/watch?v=vlzwuFkn88U>
<http://www.youtube.com/watch?v=sN62PAKoBfE>

Mentre suona la musica:

Mano sx fuori e su

Mano dx fuori e su

Ruota mano sx

Ruota mano dx

Mano sx su spalla dx

Mano dx su spalla sx

Mano sx dietro la testa

Mano dx dietro la testa

Mano sx sul fianco dx

Mano dx sul fianco sx

Mano sx dietro a sx

Mano dx dietro a dx

Ondeggiare

Ondeggiare

Salto

Programmi per esseri umani...



<http://www.youtube.com/watch?v=vlzwwFkn88U>

<http://www.youtube.com/watch?v=sN62PAKoBfE>

Mentre suona la musica:

Mano sx fuori e su

Mano dx fuori e su

Ruota mano sx

Ruota mano dx

Mano sx su spalla dx

Mano dx su spalla sx

Mano sx dietro la testa

Mano dx dietro la **fe**sta

Mano sx **colpisce** a dx

Mano dx **colpisce** a sx

Mano sx dietro a sx

Mano dx dietro a dx

Ondeggiare

Ondeggiare

Salto

Programmi per esseri umani...



<http://www.youtube.com/watch?v=vlzwuFkn88U>

<http://www.youtube.com/watch?v=sN62PAKoBfE>

Mentre suona la musica:

Mano sx fuori e su

Mano dx fuori e su

Ruota mano sx

Ruota mano dx

Mano sx su spalla dx

Mano dx su spalla sx

Mano sx dietro la testa

Mano dx dietro la testa

Mano sx sul fianco dx

Mano dx sul fianco sx

Mano sx dietro a sx

Mano dx dietro a dx

Ondeggiare

Ondeggiare

Salto

Programmi per esseri umani...



<http://www.youtube.com/watch?v=vlzwwFkn88U>

<http://www.youtube.com/watch?v=sN62PAKoBfE>



il clown corse dietro la macchina e la macchina si schianto sulla
tenda e la tenda cadde sul clown e la macchina

Programmi per Python...



Programmi per Python...

```
name = raw_input('Inserire il file:')
handle = open(name, 'r')
text = handle.read()
words = text.split()

counts = dict()
for word in words:
    counts[word] = counts.get(word,0) + 1
bigcount = None
bigword = None

for word,count in counts.items():
    if bigcount is None or count >
bigcount:
        bigword = word
        bigcount = count
print bigword, bigcount
```

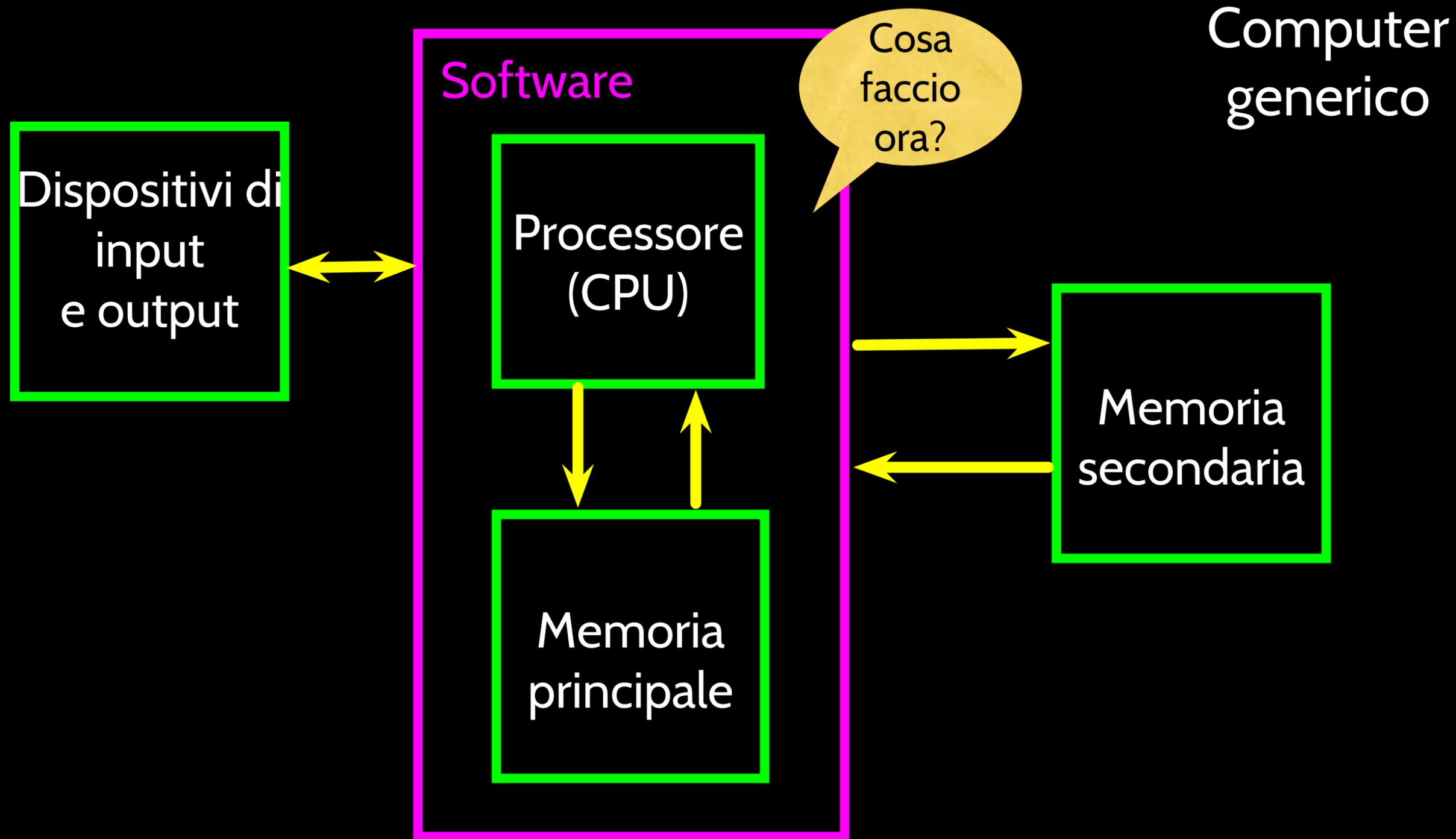
python words.py
Inserire il file: words.txt
to 16

python words.py
Inserire il file: clown.txt
the 7

Architettura hardware



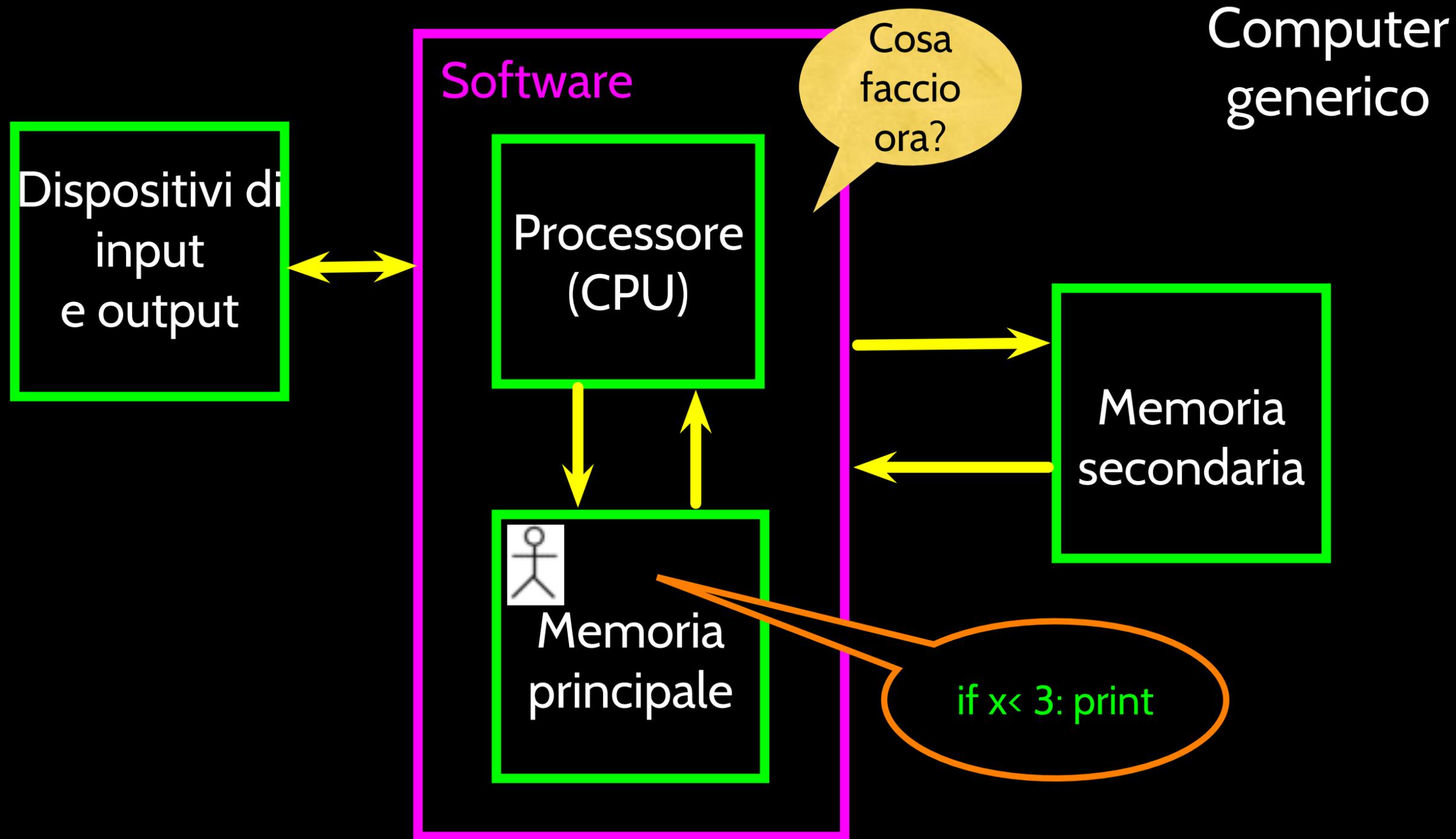
<http://upload.wikimedia.org/wikipedia/commons/3/3d/RaspberryPi.jpg>

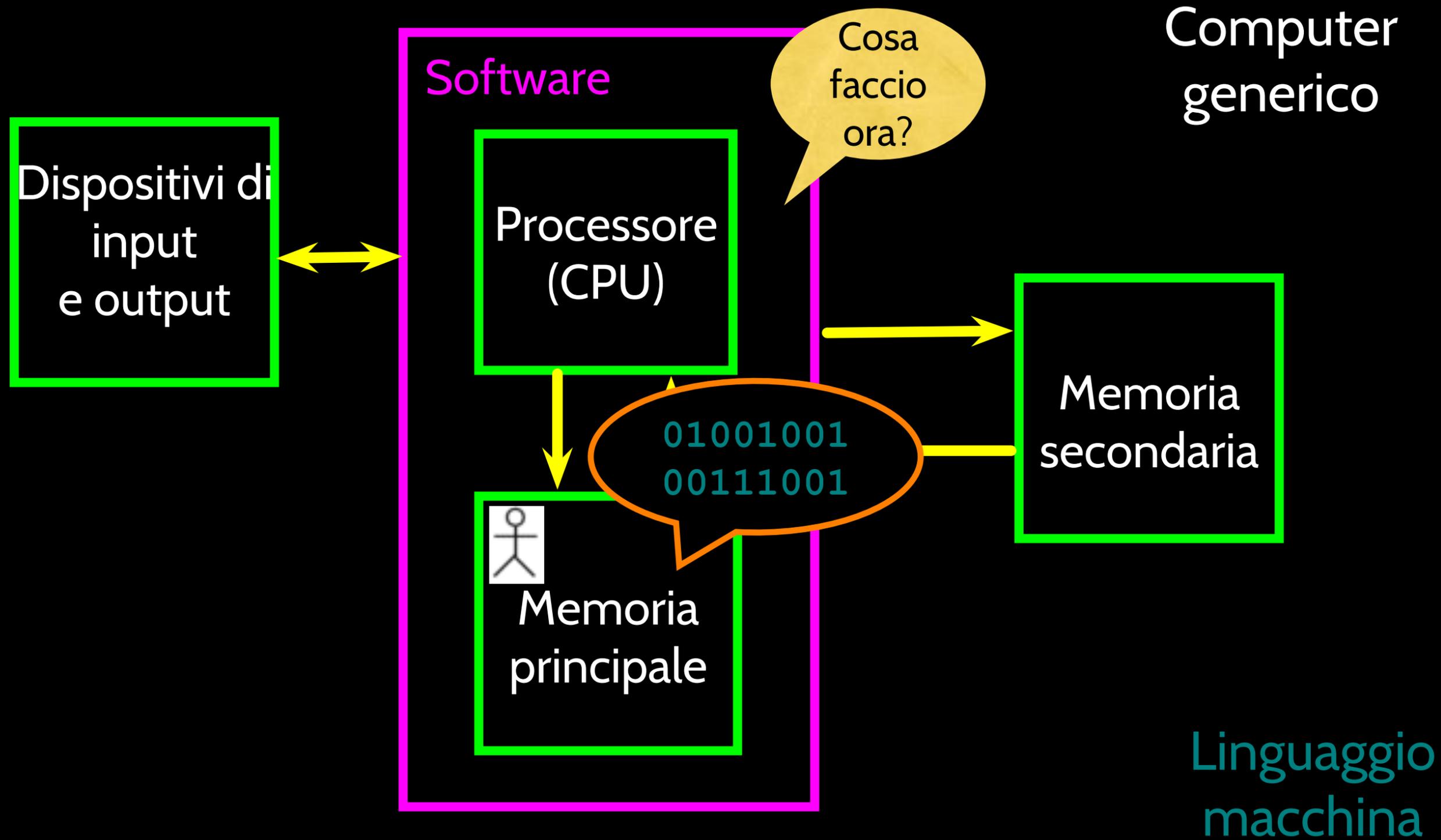


Definizioni

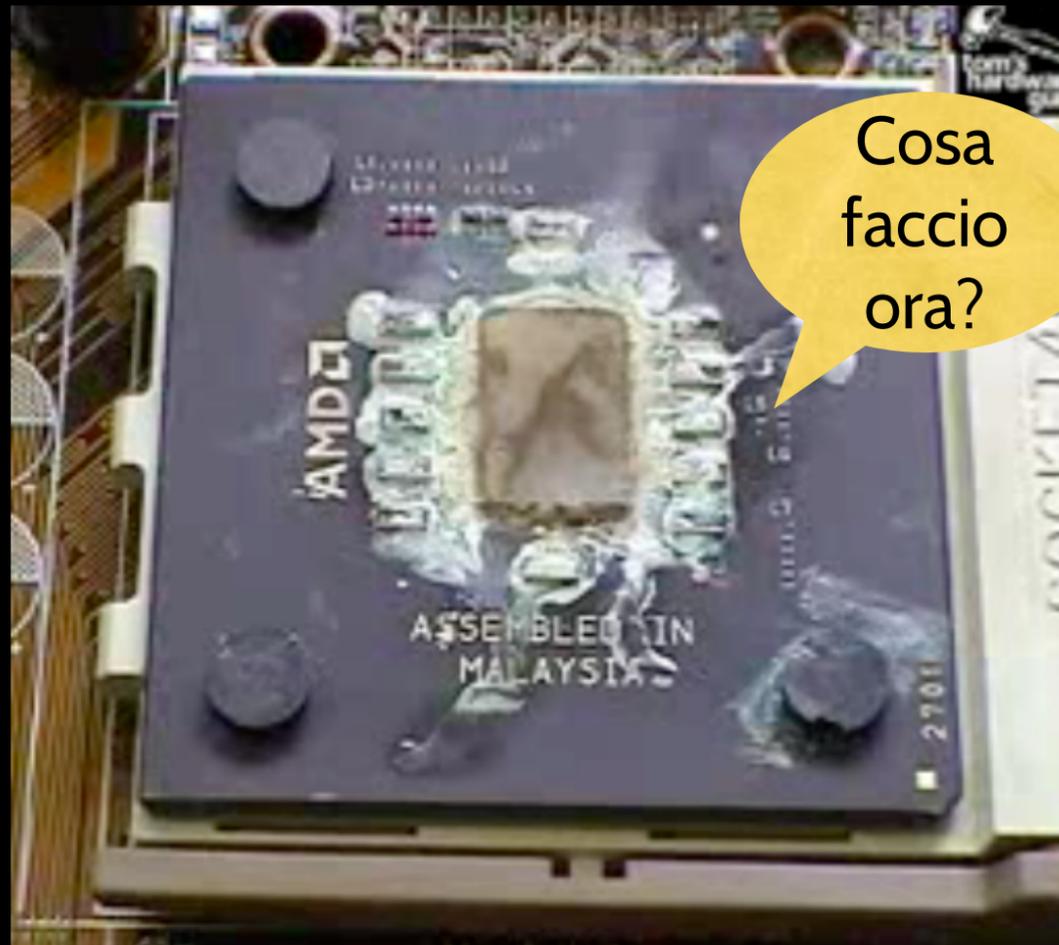
- **Processore:** esegue i programmi - la CPU è sempre in ascolto, chiedendo “Cosa faccio ora?”. Non è esattamente un “cervello” - piuttosto stupido ma molto, molto veloce
- **Dispositivi di input:** tastiera, mouse, touch screen
- **Dispositivi di output:** monitor, altoparlanti, stampanti, masterizzatori DVD
- **Memoria principale:** memoria veloce, di piccole dimensioni e temporanea - si perde al riavvio - conosciuta come RAM
- **Memoria secondaria:** memoria lenta, di grandi dimensioni e permanente - persiste sino a che non la si cancella - disk drive / memory stick (USB)





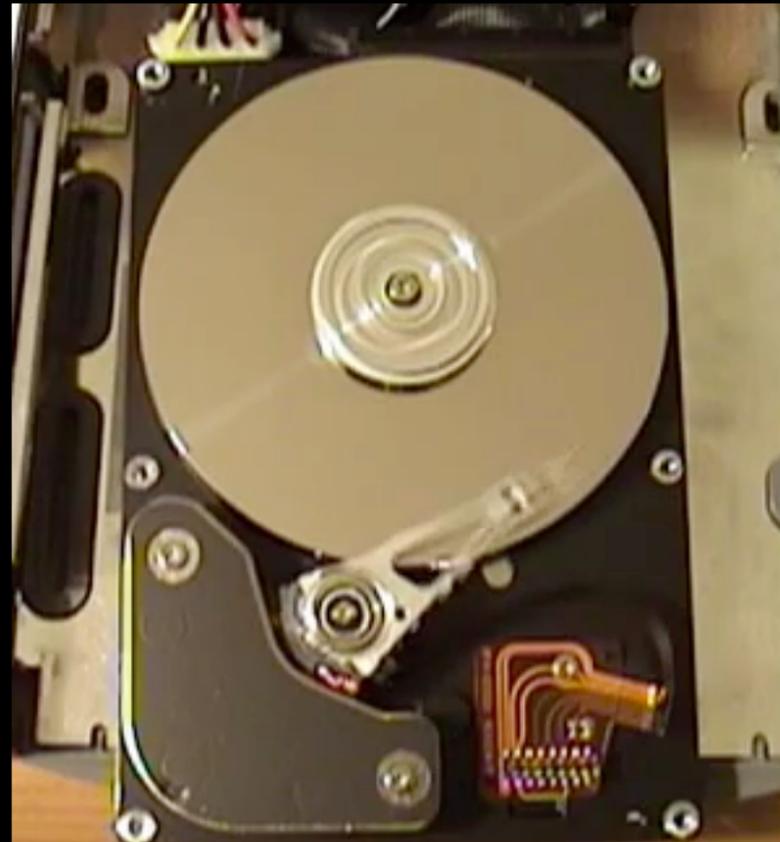


CPU “rovente”



<http://www.youtube.com/watch?v=y39D4529FM4>

Hard Disk in Azione



<http://www.youtube.com/watch?v=9eMWG3fwiEU>

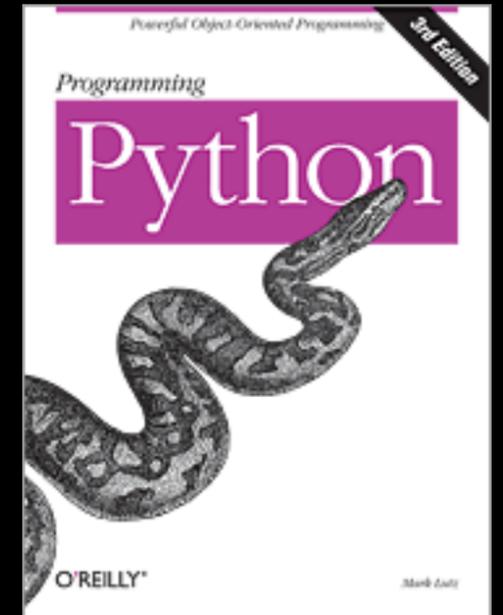
Il linguaggio Python

Il **Serpentese** è il linguaggio dei serpenti e di coloro che possono conversarvi. Un individuo che può parlare il **Serpentese** è conosciuto come **Rettilofono**. E' una capacità poco comune che può essere ereditaria. Praticamente tutti i **Rettilofoni** conosciuti sono discendenti di [Salazar Serpeverde](#).



<http://harrypotter.wikia.com/wiki/Parseltongue>

Python è il linguaggio del Python Interpreter e di coloro che possono conversarvi. Un individuo che può parlare in Python è conosciuto come Pythonista. È una capacità poco comune che può essere ereditaria. Praticamente tutti i Pythonisti usano il software sviluppato da Guido van Rossum.



Prima lezione: Syntax Errors

- Impariamo il **linguaggio Python** così da poter comunicare le nostre istruzioni a Python. All'inizio faremo molti errori e parleremo in modo "confuso" come i bambini piccoli.
- Quando fate degli errori il computer no pensa "che carino". Dice solo "**syntax error**" - dato che "conosce" il linguaggio che state imparando. Può sembrare che Python sia crudele e senza sentimenti.
- Dovete ricordarvi che **voi** siete intelligenti e **potete** imparare - il computer è semplicemente molto veloce - ma non può imparare - quindi **è più facile per voi imparare Python che per il computer imparare l'Italiano (Inglese)...**

Parlare con Python

```
csev$ python
```

```
Python 2.5 (r25:51918, Sep 19 2006, 08:49:13)
```

```
[GCC 4.0.1 (Apple Computer, Inc. build 5341)] on darwin
```

```
Type "help", "copyright", "credits" or "license" for more  
information.
```

```
>>>
```

← Cosa faccio adesso?

```
csev$ python
```

```
Python 2.5 (r25:51918, Sep 19 2006, 08:49:13)
```

```
[GCC 4.0.1 (Apple Computer, Inc. build 5341)] on darwin
```

```
Type "help", "copyright", "credits" or "license" for more  
information.
```

```
>>> x = 1
```

```
>>> print x
```

```
1
```

```
>>> x = x + 1
```

```
>>> print x
```

```
2
```

```
>>> exit()
```

Questo è un ottimo modo per assicurarvi di aver installato Python correttamente. Nota: quit() funziona anche in modalità interattiva

Parliano a Python...

```
Default
dr-chuck2:~ csev$ python
Python 2.6.1 (r261:67515, Jun 24 2010, 21:47:49)
[GCC 4.2.1 (Apple Inc. build 5646)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> print "hello world"
hello world
>>> █
```

```
Administrator: C:\Windows\system32\cmd.exe - C:\Python27\python.exe
Microsoft Windows [Version 6.0.6001]
Copyright (c) 2006 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>C:\Python27\python.exe
Python 2.7.2 (default, Jun 12 2011, 15:08:59) [MSC v.1500 32 bit (Intel)] on win
32
Type "help", "copyright", "credits" or "license" for more information.
>>> print "hello world"
hello world
>>> _
```

Cosa dobbiamo dire?

Elementi di Python

- **Vocabolario / Parole** - Variabili e parole riservate (Capitolo 2)
- **Struttura delle frasi** - Modelli di sintassi (Capitoli 3-5)
- **Struttura della Storia** - Costruire un programma per uno scopo

```
name = raw_input('Inserire il file:')
handle = open(name, 'r')
text = handle.read()
words = text.split()

counts = dict()
for word in words:
    counts[word] = counts.get(word,0) + 1
bigcount = None
bigword = None

for word,count in counts.items():
    if bigcount is None or count > bigcount:
        bigword = word
        bigcount = count
print bigword, bigcount
```

Una breve “storia”
su come contare le
parole in un file
usando Python

python words.py
Inserire il file: words.txt
to 16

Parole riservate

- NON potete usare le **parole riservate** come nomi di variabili/identificatori

and del for is raise assert elif from
lambda return break else global
not try class except if or while
continue exec import pass yield
def finally in print as with

Frasi o linee

`x = 2` ← Istruzione di assegnazione
`x = x + 2` ← Assegnazione con espressione
`print x` ← Istruzione di stampa (a video)

Variabile

Operatore

Costante

Parola Riservata

Paragrafi di programmazione

Script in Python

- Python interattivo va bene per esperimenti e programmi di 3-4 linee
- La maggior parte dei programmi sono molto più lunghi, quindi li scriviamo in un file e diciamo a python di eseguire i comandi scritti nel file
- In un certo senso stiamo “dando uno script a Python”
- Per convenzione daremo al file il suffisso “.py” per indicare che contengono istruzioni Python

Scrivere un semplice programma

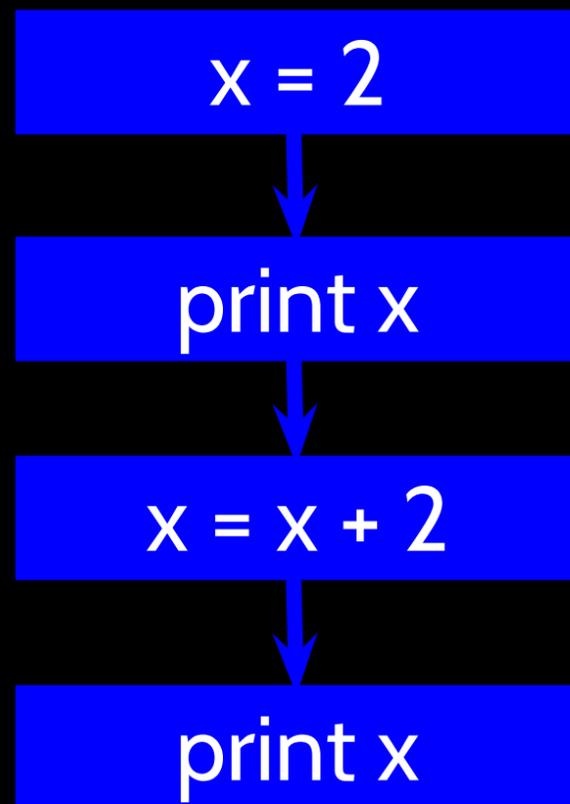
Interattivo contro script

- Interattivo
 - Scrivete direttamente in Python una linea per volta e ottenete la risposta
- Script
 - Scrivete una serie di istruzioni (linee) in un file usando un editor di testo e dite a Python di eseguire le istruzioni nel file

Passi di programma o Flusso di programma

- Come una ricetta o delle istruzioni di installazione, un programma è una sequenza di passi da fare in ordine
- Alcuni passi sono condizionali - possono essere “saltati”
- A volte un passo o un gruppo di passi possono essere ripetuti
- A volte memorizziamo un gruppo di passi che saranno ripetuti più volte i diversi punti del programma.(Capitolo 4)

Passi Sequenziali



Programma:

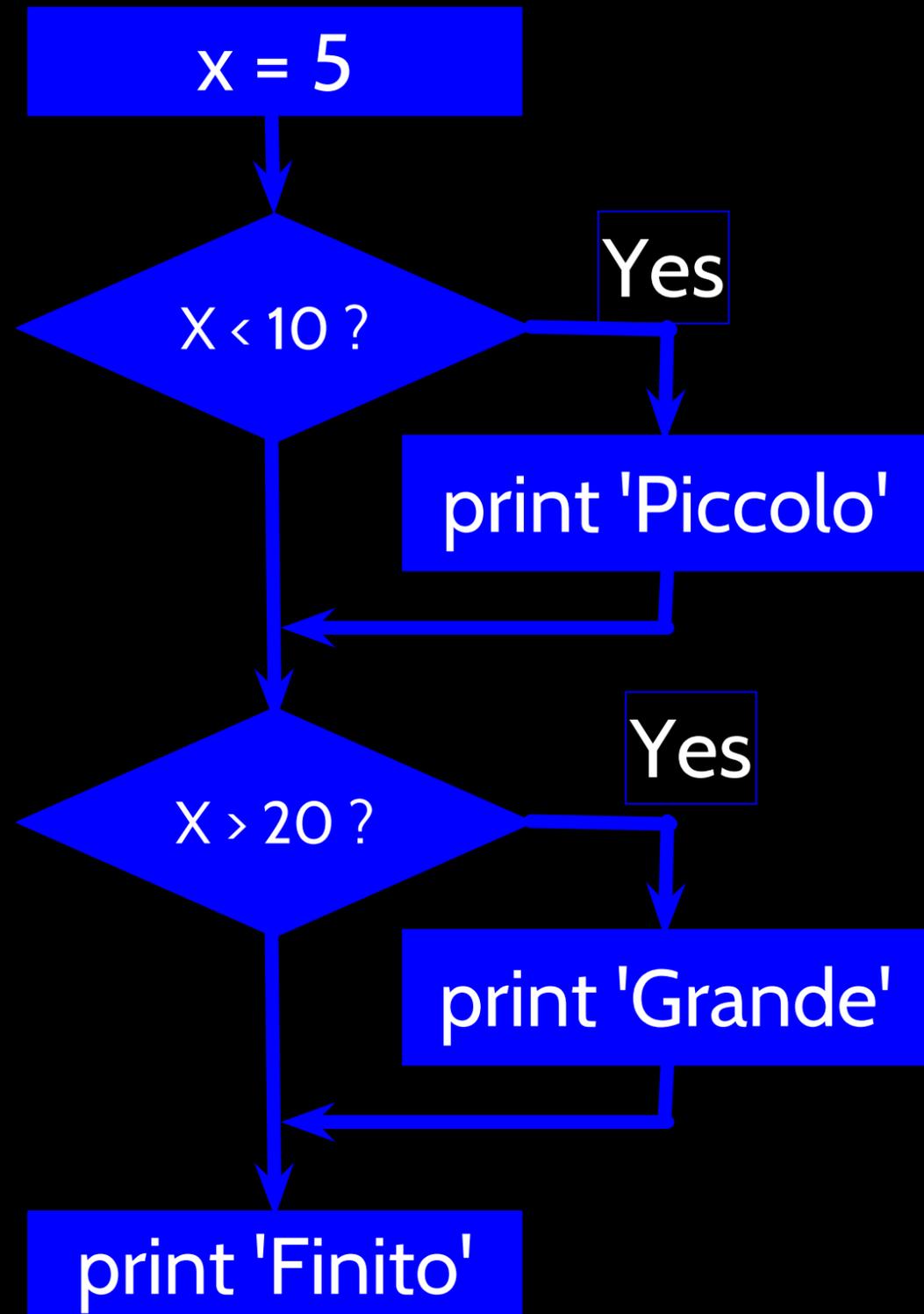
```
x = 2  
print x  
x = x + 2  
print x
```

Output
(risultato):

2
4

Quando un programma viene lanciato, esegue un passo alla volta in sequenza. Come programmatori, definiamo il “percorso” che il programma deve seguire.

Passi Condizionali



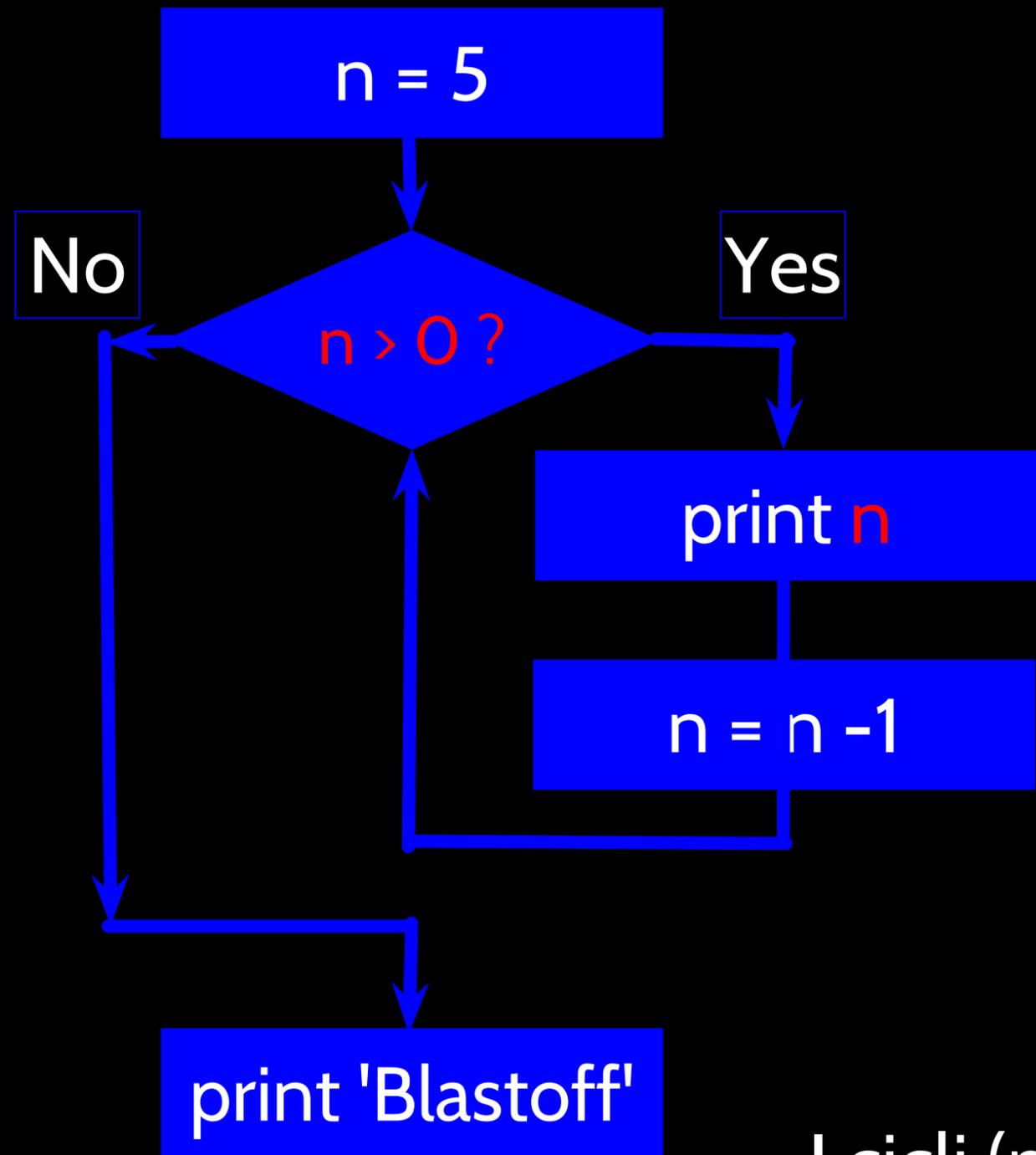
Programma:

```
x = 5
if x < 10:
    print 'Piccolo'
if x > 20:
    print 'Grande'
print 'Finito'
```

Output:

Piccolo
Finito

Passi Ripetuti



Program:

```
n = 5
while n > 0:
    print n
    n = n - 1
print 'Blastoff!'
```

Output:

5
4
3
2
1
Blastoff!

I cicli (passi ripetuti) hanno **variabili di iterazione** che cambiamo ad ogni ciclo. Spesso queste **variabili di iterazione** seguono una sequenza di numeri.

Sommario

- Questa è una veloce panoramica del Capitolo 1
- Rivedremo tutti questi concetti durante il corso
- Concentratevi sulla visione generale tralasciano, per ora, i dettagli



Acknowledgements / Contributions



These slides are Copyright 2010- Charles R. Severance (www.dr-chuck.com) of the University of Michigan School of Information and open.umich.edu and made available under a Creative Commons Attribution 4.0 License. Please maintain this last slide in all copies of the document to comply with the attribution requirements of the license. If you make a change, feel free to add your name and organization to the list of contributors on this page as you republish the materials.

Initial Development: Charles Severance, University of Michigan School Of Information