

Laboratorio di programmazione con Python e Scratch

Mini corso "Informatica e Matematica nella Scuola Primaria"

Marco Liverani

liverani@mat.uniroma3.it

15 gennaio 2015

Python: utilizzo del linguaggio

- Per poter sviluppare esempi ed esercizi con il linguaggio Python è necessario predisporre l'ambiente sul computer
- È necessario:
 - scaricare da Internet il pacchetto di installazione di Python 3.x dal sito <http://www.python.it>
 - installare il programma interprete e il programma IDLE sul proprio computer (esistono versioni di Python per i principali sistemi operativi, Microsoft Windows, Apple OS X, Linux: dovete solo scegliere quello adatto al vostro computer)
 - eseguire il programma IDLE per scrivere programmi in Python ed eseguirli (usare IDLE è il modo più semplice, anche se non è il solo!)

Python: utilizzo del linguaggio

python.it

python - sito ufficiale della comunità italiana

Python è un linguaggio di programmazione dinamico orientato agli oggetti utilizzabile per molti tipi di sviluppo software. Offre un forte supporto all'integrazione con altri linguaggi e programmi, è fornito di una estesa libreria standard e può essere imparato in pochi giorni. I programmatori Python sostanzialmente aiutano che il loro sviluppo di codice sia di qualità superiore.

Python gira su Windows, Linux, OS/2, Amiga, palmari Palm e cellulare, ed è stato anche portato sulle macchine e .NET.

Python è distribuito con licenza approvata dalla OSI: il suo utilizzo è libero anche per prodotti commerciali. Leggi altre informazioni o prova.

Appunti di programmazione

python.it

Scarica Python

Le versioni correnti di Python sono Python 2.7.8 e Python 3.4.2.

Se non sai quale versione usare esiste un documento [in inglese], che potrebbe aiutarti nella scelta. In linea di massima se devi iniziare da zero, ti consigliamo di scegliere Python 3, mentre se devi utilizzare pacchetti software di terze parti che potrebbero non essere ancora compatibili con Python 3, ti consigliamo di utilizzare Python 2.7.

In sintesi: Python 2.x è l'eredità, Python 3.x è il presente ed il futuro del linguaggio Python 3.4.2

Vedi anche la pagina dettagliata su Python 3.4.2:

- Python 3.4.2: installer MSI per Windows x86 (Pacchetto binario per Windows, non include i sorgenti)
- Python 3.4.2: installer MSI per Windows x86-64 (Pacchetto binario per Windows AMD64 / intel 64 / x86-64, non include i sorgenti)
- Python 3.4.2: installer per Mac OS X 64-bit/32-bit x86-64/386 (per Mac OS X 10.6 e successivi)
- Python 3.4.2: installer per Mac OS X 32-bit/386/PPC (per Mac OS X 10.5 e successivi)
- Python 3.4.2: sorgenti compressi con XZ

Python 2.7.8

Python: utilizzo del linguaggio

```
Python 3.4.2 Shell
Python 3.4.2 (v3.4.2:ab2c023a9432, Oct 5 2014, 20:42:22)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>> WARNING: The version of Tcl/Tk (8.5.9) in use may be unstable.
Visit http://www.python.org/download/mac/tcltk/ for current information.

>>> ===== RESTART =====
>>>
Come ti chiami? Marco
Ciao Marco
Quanti anni hai? 46
Sei maggiorenne!
>>>
```

```
saluti.py - /Users/marco/src/python/saluti.py (3.4.2)
nome = input("Come ti chiami? ")
print("Ciao", nome)
eta = int(input("Quanti anni hai? "))
if eta < 18:
    print("Sei ancora minorenne!")
else:
    print("Sei maggiorenne!")
```

Finestra per eseguire il programma

Finestra per scrivere il programma in Python

Ln: 8 Col: 0

Il programma **IDLE** per scrivere ed eseguire programmi in Python

Python: alcuni esempi (1)

- **Problema:** dato in input un numero intero n stabilire se n è **pari** o **dispari**

- **Esempio:**

sia $n = 7$, allora visto che $5 - 2 - 2 - 2 = 1$ allora n è dispari

sia $n = 8$, allora visto che $8 - 2 - 2 - 2 - 2 = 0$ allora n è pari

- **Programma in Python:**

```
n = int(input("Inserisci il numero n:"))
a = n
while n>1:
    n = n-2
if n==1:
    print("Il numero", a, "è dispari")
else:
    print("Il numero", a, "è pari")
```

Python: alcuni esempi (2)

- **Problema:** dati in input un numero intero n e un numero intero x stampare i primi n **multipli** di x

- **Esempio:** sia $n = 5$ e $x = 7$, allora si deve stampare **7, 14, 21, 28, 35**

- **Programma in Python:**

```
n = int(input("Inserisci il numero n:"))
x = int(input("Inserisci il numero x:"))
i = 0
m = 0
while i<n:
    m = m + x
    i = i+1
    print(m)
```

Python: alcuni esempi (3)

- **Problema:** dati in input n numeri interi, stampa in output il **minimo** e il **massimo**
- **Esempio:** sia $n = 5$ e siano $42, 12, 24, 51, 37$ i numeri della sequenza inserita in input; allora il minimo è 12 e il massimo 51
- **Programma in Python:**

```
n = int(input("Inserisci il numero n:"))
i = 1
x = int(input("inserisci un numero:"))
min = x
max = x
while i < n:
    x = int(input("inserisci un numero:"))
    i = i+1
    if min > x:
        min = x
    if max < x:
        max = x
print("Minimo:", min, "Massimo:", max)
```

Python: alcuni esempi (4)

- **Problema:** dati in input due numeri interi x e y , stampa in output il **minimo comune multiplo** (m.c.m.)
- **Esempio:** sia $x = 8$ e $y = 14$ allora il mcm è 56 perché:
 $8 + 8 + 8 + 8 + 8 + 8 + 8 = 56$
 $14 + 14 + 14 + 14 = 35$
- **Programma in Python:**

```
x = int(input("Inserisci il numero x:"))
y = int(input("Inserisci il numero y:"))
a = x
b = y
while a != b:
    if a < b:
        a = a + x
    else:
        b = b + y
print("Minimo comune multiplo:", a)
```

Python: alcuni esempi (5)

- **Problema:** dati in input n numeri interi, stampa in output la **media aritmetica**
- **Esempio:** sia $n = 4$ e siano 12, 3, 22, 18 i numeri inseriti in input; allora la media aritmetica è $(12+3+22+18)/4 = 13,75$

- **Programma in Python:**

```
n = int(input("Numero di elementi:"))
s = 0
i = 0
while i < n:
    x = int(input("Inserisci un numero:"))
    s = s + x
    i = i + 1
m = s/n
print("La media aritmetica è", m)
```

Scratch: un linguaggio per la didattica

- Il linguaggio **Scratch** è un linguaggio sviluppato dal MIT (Massachusetts Institute of Technology) per avvicinare i bambini delle scuole elementari alla codifica di algoritmi (*coding*)
- Può ricordare quello che in passato era stato il linguaggio **Logo**
- Sul sito web dedicato a Scratch è presente un **ambiente di sviluppo e di esecuzione dei programmi**, completamente **on-line**: non è quindi necessario installare nulla, ma occorre una connessione alla rete Internet
- Il linguaggio Scratch è costituito da una serie di istruzioni complesse, che riprendono il paradigma dei linguaggi imperativi, ma rispondente alle competenze e alle abilità dell'esecutore
- In questo caso l'esecutore è un gattino (Scratch) che si muove sullo schermo del computer, emette dei suoni e può disegnare (lasciare una traccia grafica) mentre si muove

Scratch: un linguaggio per la didattica

- Scrivere un programma con Scratch significa **comporre una sorta di diagramma di flusso** utilizzando gli strumenti (i "tasselli" corrispondenti alle istruzioni del linguaggio) messi a disposizione dall'ambiente di sviluppo
- Le istruzioni sono suddivise nelle seguenti categorie:
 - **Movimento**: comandi per il movimento del personaggio
 - **Aspetto**: comandi per modificare l'aspetto del personaggio
 - **Suono**: comandi per far emettere suoni o parole al personaggio
 - **Penna**: comandi per utilizzare la penna con cui il gatto lascia una traccia sullo schermo mentre si muove
 - **Variabili e liste**: comandi per creare variabili di memoria, assegnare valori alle variabili e modificare il valore precedentemente assegnato ad una variabile
 - **Situazioni**: eventi che possono essere rilevati durante l'esecuzione del programma (la pressione di un tasto, ecc.)
 - **Controllo**: le istruzioni per il controllo dell'algoritmo (condizioni, loop, ecc.)
 - **Sensori**: comandi per utilizzare ciò che è stato rilevato da uno dei sensori del personaggio
 - **Operatori**: istruzioni per confrontare numeri o eseguire operazioni

Scratch: un linguaggio per la didattica

The screenshot shows the Scratch web interface. The main stage displays a cat sprite and a red hexagon. The script area on the right contains a sequence of blocks: a 'quando si clicca su' block, a 'pulisci' block, an 'usa penna di colore' block, a 'penna giù' block, a 'ripeti 3 volte' loop containing 'fai 100 passi', 'ruota di 60 gradi', and 'attendi 1 secondi', followed by 'produci suono meow'. The 'Script' category is selected in the left sidebar.

Annotations in blue speech bubbles point to specific parts of the interface:

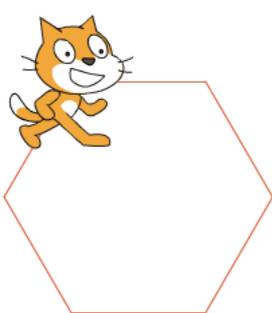
- "Selezione categoria di comandi" points to the 'Script' category in the left sidebar.
- "Esecuzione del programma Scratch" points to the cat sprite on the stage.
- "Programma Scratch" points to the script area.
- "Scelta dei comandi per comporre il programma" points to the 'ripeti' loop block.

Scratch: alcuni esempi

- Disegnare un esagono di 100 passi per ogni lato

```

quando si clicca su
pulisci
usa penna di colore
penna giù
ripeti 6 volte
  fai 100 passi
  ruota di 60 gradi
  attendi 1 secondi
  produci suono meow
            
```

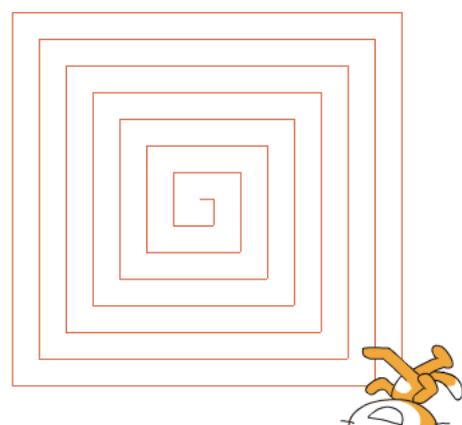


Scratch: alcuni esempi

- Disegnare una "spirale quadrata"

```

quando si clicca su
pulisci
usa penna di colore
penna giù
porta x a 10
ripeti 30 volte
  fai x passi
  cambia x di 10
  ruota di 90 gradi
  attendi 1 secondi
  produci suono meow
            
```



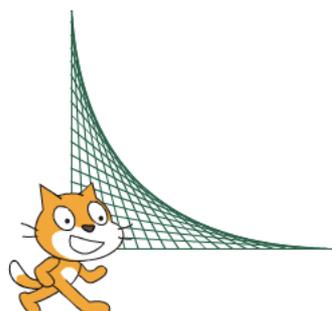
Scratch: alcuni esempi

- Disegnare un ramo di iperbole con dei segmenti di retta

```

quando si clicca su
pulisci
usa penna di colore
porta x a 0
porta y a 200
porta delta_x a 10
porta delta_y a 10
ripeti fino a quando x > 200
  vai a x: x y: 0
  penna giù
  vai a x: 0 y: y
  penna su
  attendi 1 secondi
  cambia x di 10
  cambia y di -10
  
```

x 210
y -10
delta_x 10
delta_y 10



Dash: un robot esecutore di algoritmi



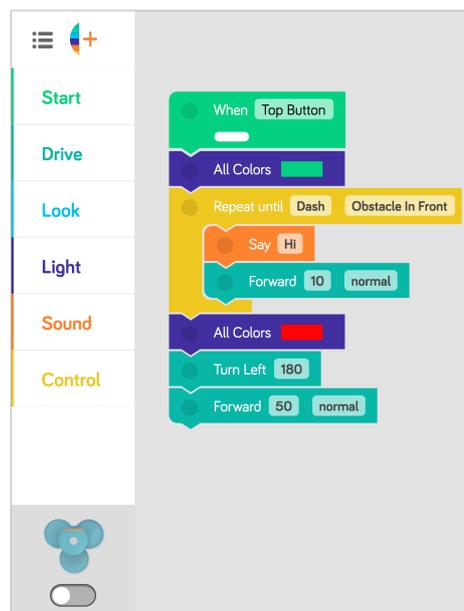
- Per uscire da un contesto un po' auto-referenziale, in cui l'ambiente di programmazione e l'esecutore coincidono con lo stesso computer, cercando di stimolare ulteriormente la fantasia dei bambini che si avvicinano al "coding", è possibile utilizzare un robot nato per finalità didattiche
- A titolo di esempio utilizziamo **Dash**, il robot prodotto da *Wonder Workshop*, che è possibile acquistare on-line sul sito <http://www.makewonder.com>

Dash: un robot esecutore di algoritmi

- Dash è un robot che è possibile programmare con il linguaggio **Blocky**, simile a Scratch (di fatto una versione personalizzata e un po' limitata di Scratch)
- Il linguaggio è stato adattato alle effettive caratteristiche e capacità del robot:
 - **movimento**: il robot ha delle ruote e può muoversi su una scrivania o sul pavimento di una stanza
 - **sensori**: il robot ha dei sensori con cui può rilevare la presenza di oggetti e ostacoli
 - **luci**: il robot ha delle luci colorate che può accendere o spegnere
 - **suoni**: il robot può emettere dei suoni e delle parole in inglese
 - **controllo dell'algoritmo**: anche il linguaggio Blocky dispone di strutture di controllo di tipo condizionale e iterativo

Dash: un robot esecutore di algoritmi

- L'ambiente di sviluppo di Dash è costituito al momento da una **App** disponibile per dispositivi Apple iOS (iPad) o Android (tablet)
- La App permette di **sviluppare un algoritmo**, come con Scratch, e **trasmettere il programma al robot** mediante una connessione wireless Bluetooth



Quale riferimento utile

- M. Beri, *Python*, Apogeo, 2010
- A. Colombi, *Immagina, programma e condividi con Scratch*, Centro Studi Erickson, 2010
- Siti Internet di interesse:
 - Comunità italiana di Python: <http://www.python.it>
 - Sito di Scratch presso il MIT: <http://scratch.mit.edu>
 - Sito di Dash presso Wonder Workshop: <https://www.makewonder.com>

Laboratorio di programmazione con Python e Scratch

... questa volta abbiamo veramente terminato.

Grazie per l'attenzione!