



POLITECNICO
MILANO 1863

Coding in Python

Gianluca Palermo – gianluca.palermo@polimi.it



Obiettivo

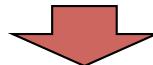
- Imparare i concetti di base della programmazione
- Imparare a usare Python come linguaggio di programmazione
- Imparare a risolvere piccoli problemi di programmazione

Informatica

- Scienza che studia la rappresentazione e l'elaborazione dell'informazione (mediante macchine dette calcolatori elettronici)
- non solo la tecnologia dei calcolatori, ma anche il modo in cui l'informazione viene strutturata ed elaborata automaticamente

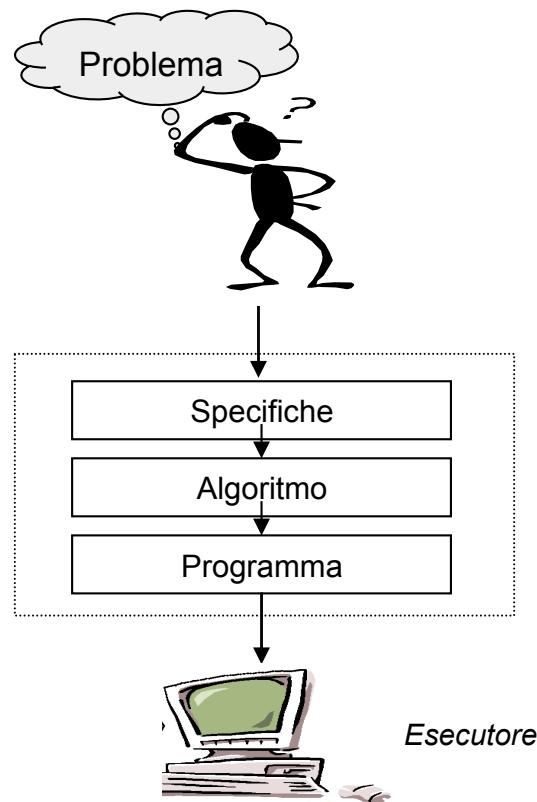
Informatica

- Scienza che studia la rappresentazione e l'elaborazione dell'informazione (mediante macchine dette calcolatori elettronici)
- non solo la tecnologia dei calcolatori, ma anche il modo in cui l'informazione viene strutturata ed elaborata automaticamente



- **Scienza**: approccio rigoroso e sistematico
- **Informazione**: è parte di qualsiasi attività umana
- **Rappresentazione**: il problema di astrarre i concetti importanti da quelli trascurabili per poter **modellare** la realtà di interesse. Dobbiamo studiare metodi di rappresentazione appropriati all'elaborazione da parte di una macchina digitale
- **Elaborazione**: uso e trasformazione dell'informazione in maniera funzionale agli obiettivi

Che cosa è la programmazione (o coding)?



- *Uso del calcolatore per agevolare e potenziare le possibilità di trattamento dell'informazione*
- *Soluzione del Problema in forma adatta alla risoluzione automatica*
- *Descrizione dell'Algoritmo in forma comprensibile dal calcolatore*

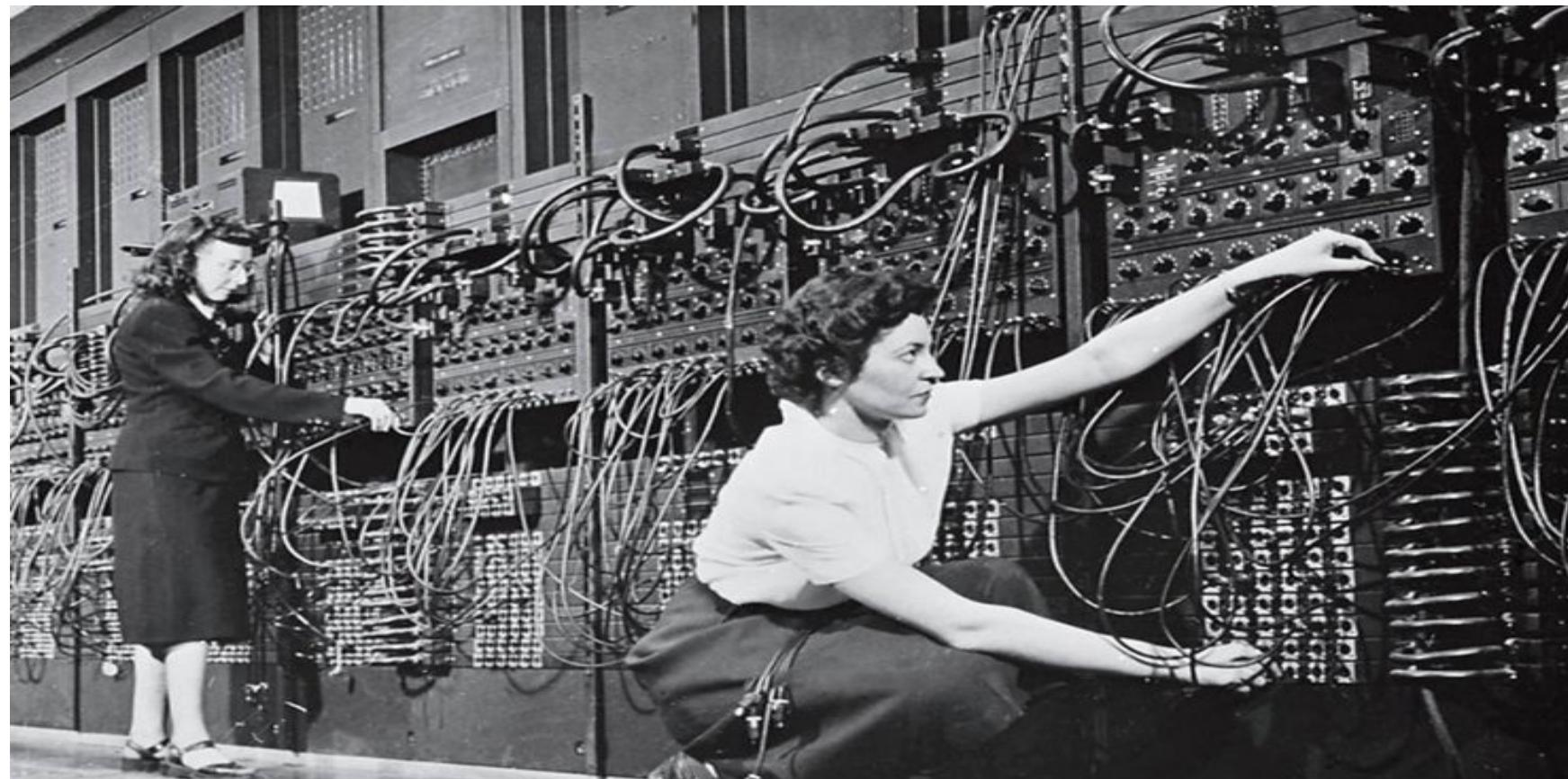
Un po' di storia: da macchine per svolgere un solo compito...



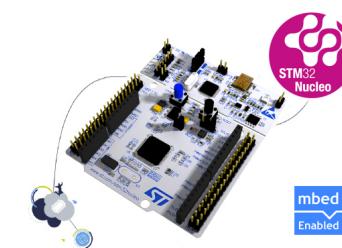
By Alessandro Nassiri - Museo della Scienza e della Tecnologia "Leonardo da Vinci",
CC BY-SA 4.0, <https://commons.wikimedia.org/w/index.php?curid=47910919>

Enigma fu una macchina elettro-meccanica per cifrare e decifrare messaggi utilizzata dal servizio delle forze armate tedesche durante il periodo nazista e della seconda guerra mondiale.

...a macchine programmabili...



...a macchine programmabili un po' più maneggevoli...



... ma anche meno maneggevoli ☺



<https://www.youtube.com/watch?v=avP5d16wEp0>

Utenti, computer, reti, programmatori

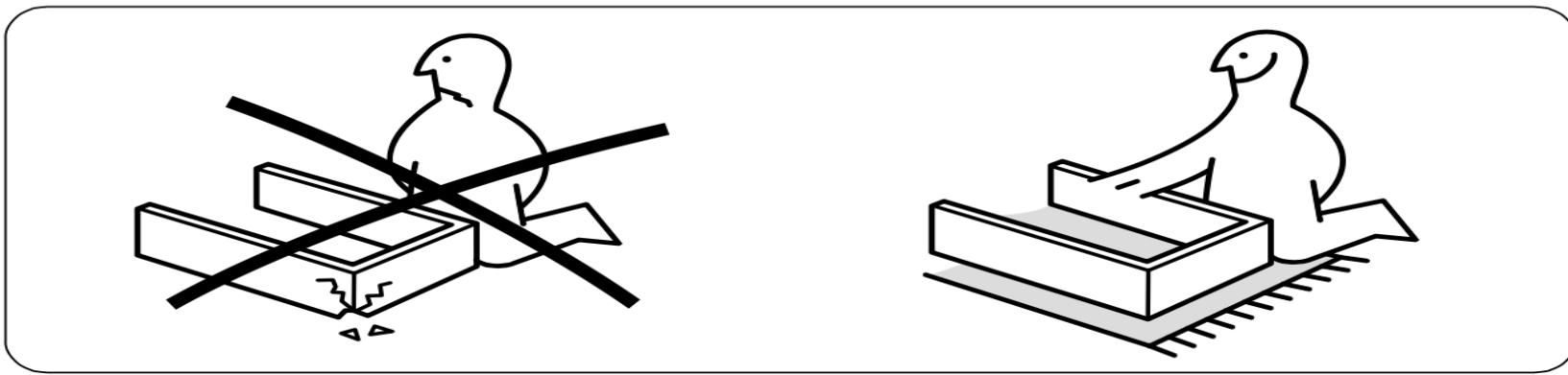
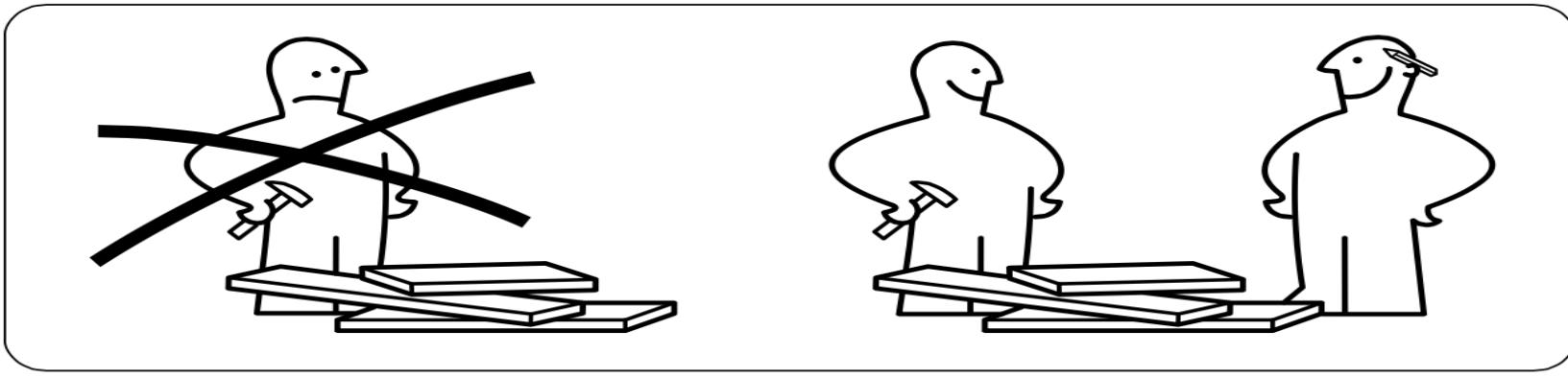


Software, codice e programmi

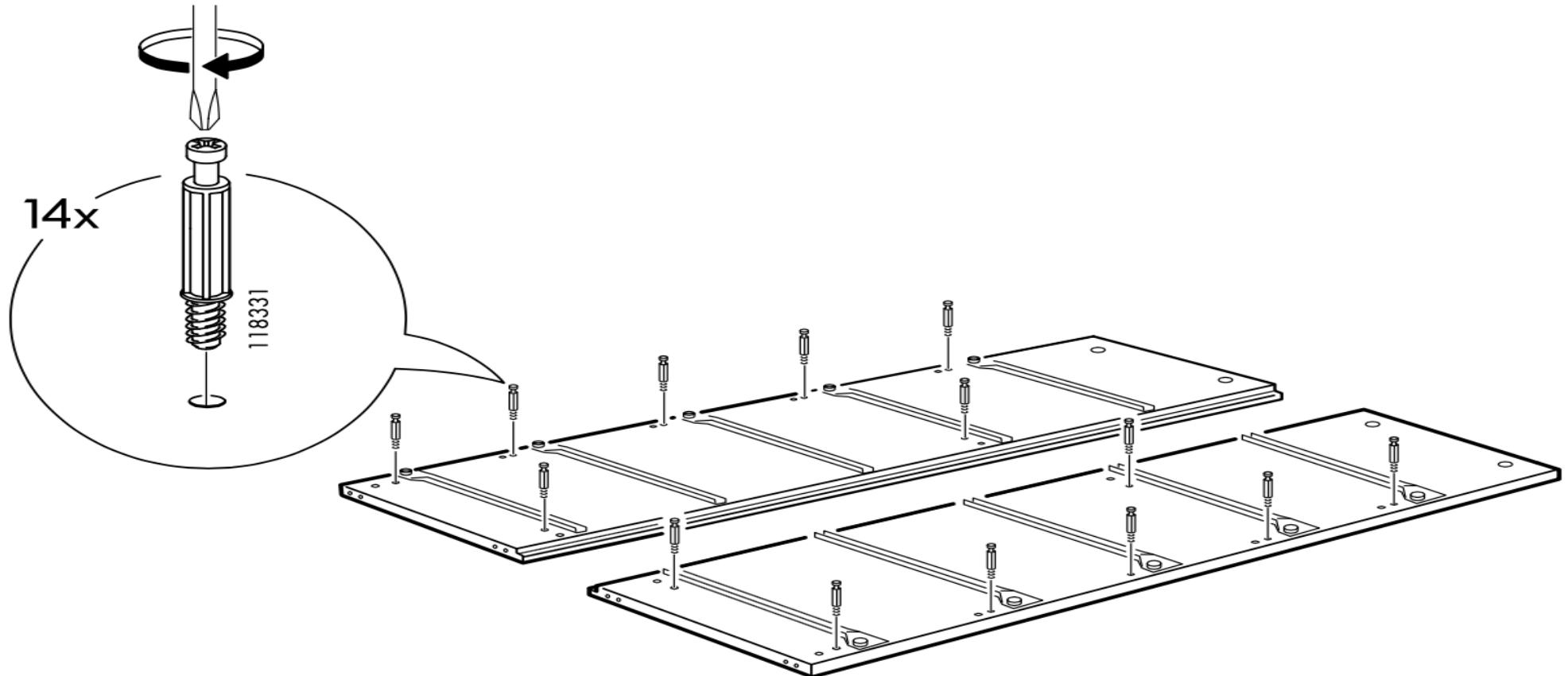
Una sequenza di istruzioni

- Può essere usato più volte per svolgere un certo compito
- Può essere un pezzo d'arte creativa!

Programmi per umani



Programmi per umani



Programmi per umani

Leggi w e y

Controlla e ricorda se $y < 0$

Inizializza la somma parziale a zero ($sp=0$)

Ripeti y volte

$sp=sp+w$

se y era < 0

cambia segno a sp

Scrivi sp

Moltiplicazione tramite
somme successive W^*Y

Programmi per umani

Divisione tra polinomi

Vediamo delle regole generali per effettuare la divisione tra due polinomi.

Consideriamo un polinomio A di grado m (dividendo) e un polinomio B di grado n (divisore), con $m \leq m$;

1. Si ordinano i polinomi secondo le potenze decrescenti della lettera x;
2. Si divide il primo termine del dividendo per il primo termine del divisore: il quoziente ottenuto è il primo termine del quoziente dei due polinomi;
3. Si moltiplica il termine in questione per il divisore e si somma il prodotto, cambiato di segno, con il dividendo; il polinomio ottenuto è il primo resto parziale;
4. Si divide il primo termine del resto parziale per il primo termine del divisore e si ottiene il secondo termine del quoziente dei polinomi;
5. Si moltiplica questo termine per il divisore e si somma il prodotto, cambiato di segno, con il precedente resto, ottenendo il secondo resto parziale;
6. Si procede in questo modo finché non si ottiene un resto parziale di grado inferiore al grado del divisore (questo ultimo resto sarà il resto della divisione).

Programmi per umani

<https://www.youtube.com/watch?v=smYstwutAvA>



Programmi per il computer ... un esempio in Python

```
text = input('Write a sentence: ')  
  
Alphabet = [ 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i',  
             'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r',  
             's', 't', 'u', 'v', 'w', 'x', 'y', 'z' ]  
  
notAPangram = False  
  
for letter in Alphabet:  
    if letter not in text:  
        notAPangram = True  
  
if notAPangram:  
    print('the string is not a pangram')  
else:  
    print('the string is a pangram')
```

Il Pangram è una frase di senso compiuto che contiene tutte le lettere dell'alfabeto

Programmi, esecutori e linguaggi di programmazione

L'esecutore di un programma deve essere capace di capire il programma e di eseguirne le istruzioni

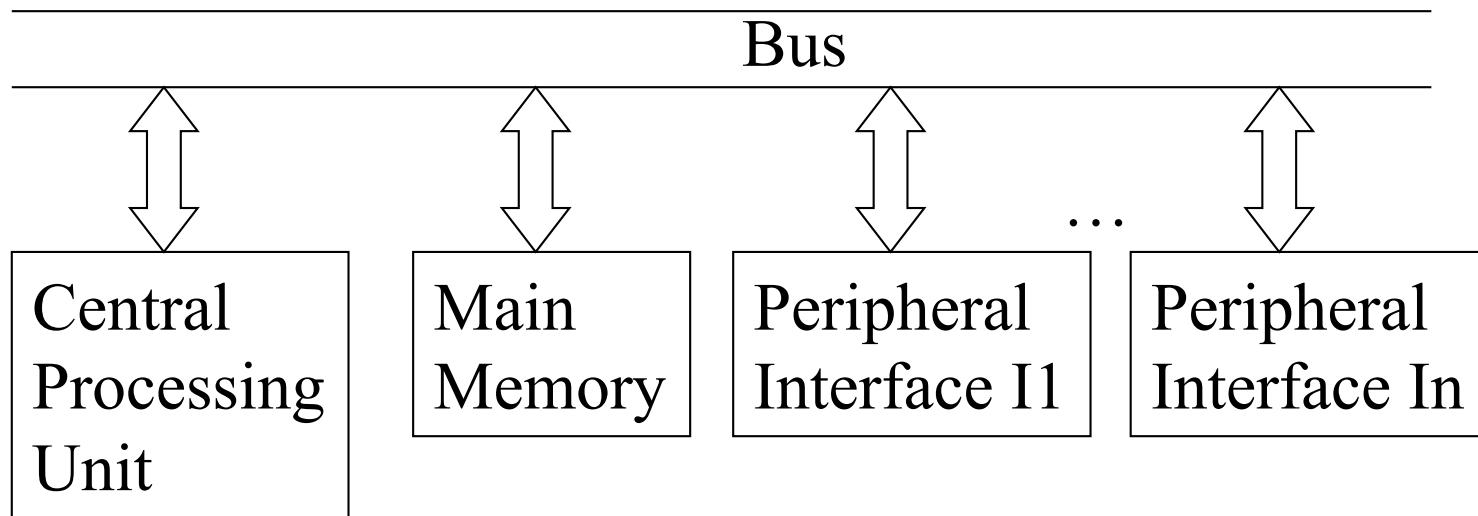
Il programma deve essere definito in un linguaggio che l'esecutore capisce e di cui sa eseguire le istruzioni

Qual è il linguaggio compreso da un computer?

Che cosa è un computer?



Il modello di Von Neumann



Rappresentazione dell' informazione

- Codifica in forma binaria con sequenze finite di 1 e 0
- Unità minima di informazione: bit (binary digit)
- Byte: 8 bit
 - Posso usarlo per rappresentare 2^8 valori diversi (00000000, 00000001, 00000010, ..., 11111111)
- Esempi
 - Numeri naturali: intervallo [0, 255]. $255 = 2^8 - 1$
 - Numeri interi: posso usare un bit per il segno e 7 per rappresentare il numero. Intervallo $[-127 \text{ } (-(2^{(8-1)}-1)), +127 \text{ } (2^{(8-1)}-1)]$
 - Caratteri: codifica ASCII (American Standard Code for Information Interchange)
 - “a”: 01100001
 - “0”: 00110000

Esempio di un programma scritto nel linguaggio del calcolatore (versione semplificata)

0000	0100000000001000
0001	0100000000001001
0010	0000000000001000
0011	0001000000001001
0100	0110000000000000
0101	0010000000001010
0110	0101000000001010
0111	1101000000000000
1000	
1001	
1010	

Acquisisci il primo numero e conservalo in 1000
Acquisisci il secondo numero e conservalo in 1001
Carica in A il numero conservato in 1000
Carica in B il numero conservato in 1001
Effettua la somma
Conserva in 1010 il valore che si trova in A
Stampa il contenuto di 1010
Termina l' esecuzione

X

Y

Z

Linguaggio macchina vs altri linguaggi di programmazione

Il linguaggio macchina è poco espressivo

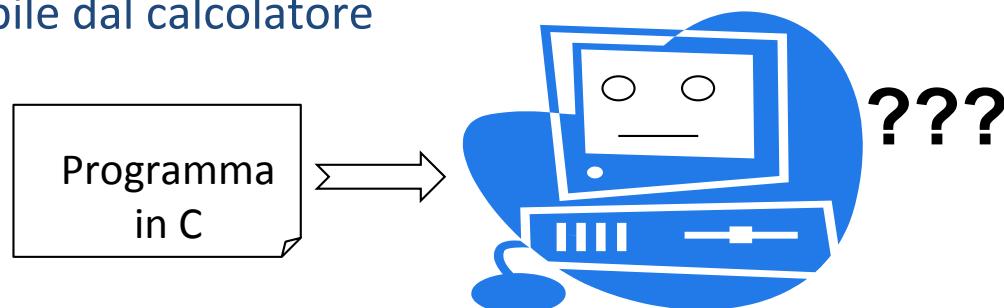
- Un compito semplice deve essere decomposto in più istruzioni macchina

I linguaggi di programmazione di “alto livello” offrono una maggiore espressività e semplicità

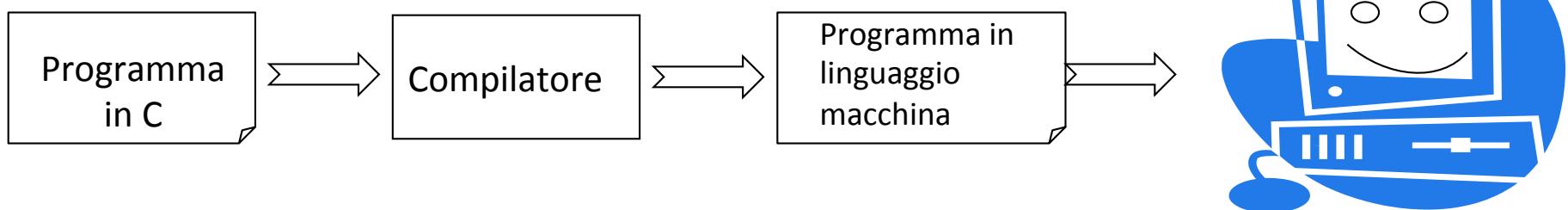
- Python, C, Java, C++, MATLAB...

Come fa la nostra macchina ad eseguire programmi in linguaggio di alto livello?

- Un programma scritto in un linguaggio di alto livello NON è direttamente eseguibile dal calcolatore



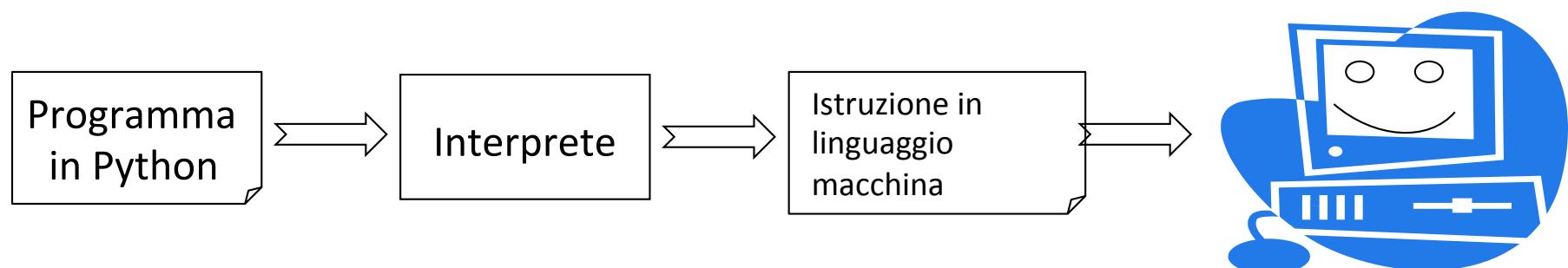
- Ci vuole un traduttore, il *compilatore*



- NB: il compilatore di solito è esso stesso un programma eseguito dal calcolatore

Come fa la nostra macchina ad eseguire programmi in linguaggio di alto livello?

Seconda possibilità: non operiamo una trasformazione a priori prima dell'esecuzione, usiamo i servizi di un interprete



L'interprete legge e traduce al volo durante l'esecuzione

Ambienti di programmazione

Forniscono strumenti a supporto della programmazione

In genere offrono

- Editor per scrivere i programmi
- Compilatore: crea un *codice oggetto* per ogni parte del programma
 - Linker: collega le varie parti del programma creando un *unico eseguibile*
 - Debugger: consente il controllo del programma durante la sua esecuzione
- Oppure interprete

Python come linguaggio di programmazione



WHICH PROGRAMMING LANGUAGE

SHOULD I LEARN FIRST?



WHAT IS PROGRAMMING?

Writing very specific
instructions to a very dumb,
yet obedient machine.



LANGUAGES



10 of 10



10 of 10



10 of 10



100



10 of 10



3



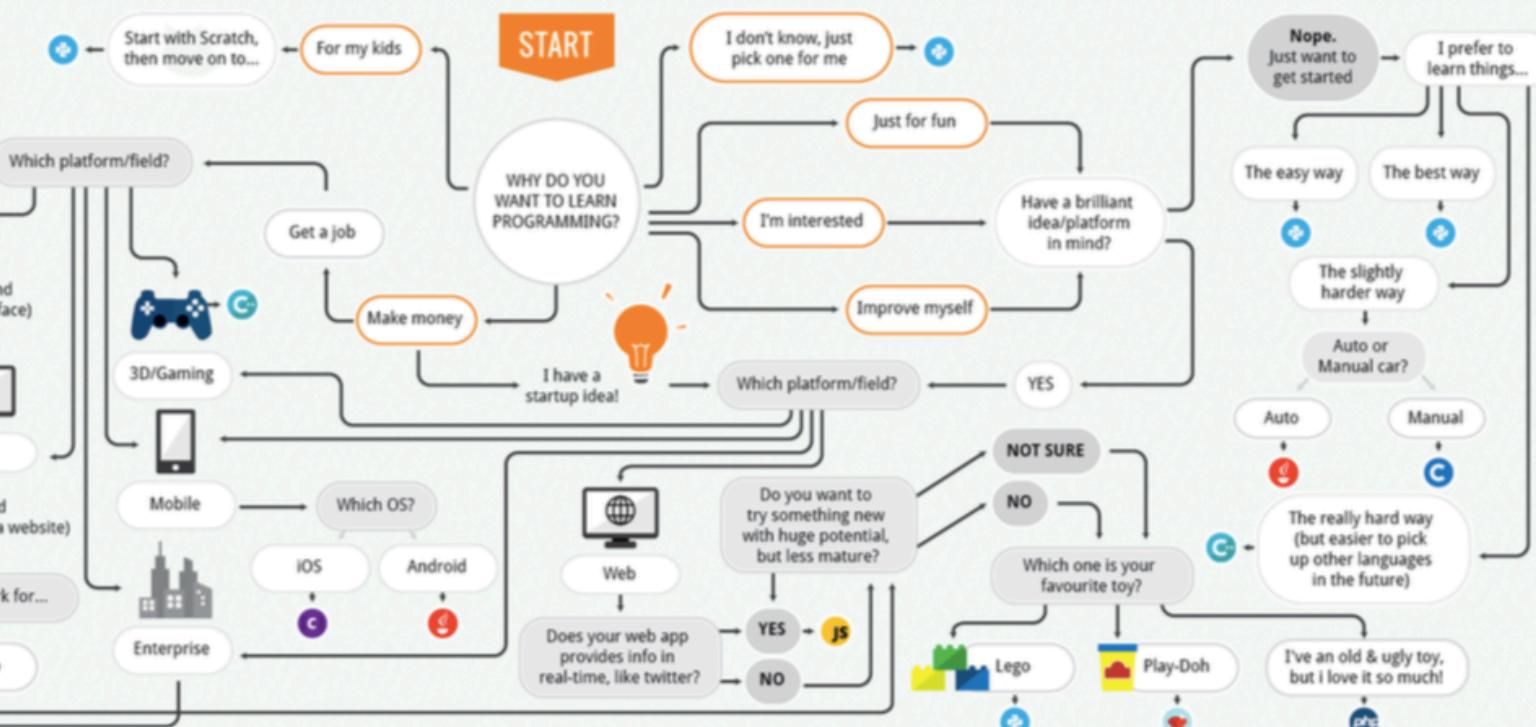
1



1



1



Perchè Python per il corso di CODING...

- Linguaggio di programmazione facile da apprendere per un “programmatore” alle prime armi
- Il codice è facile da leggere confrontato con altri linguaggi di programmazione
- Ha la possibilità di interagire in maniera interattiva con un interprete in modo da vedere l'effetto di ogni istruzione di un programma
- Esistono diverse librerie Python che rendono facile fare cose complesse nascondendone parte della complessità
- È uno dei linguaggi di programmazione con una rampa più ripida degli ultimi anni

Errori sintattici: noi e il computer

- Dobbiamo imparare un nuovo linguaggio
- Faremo tanti errori
- Il computer non sarà comprensivo
 - Ci dirà “syntax error”. Ci sembrerà crudele perchè ci “sgriderà” 😊
- Ricordiamo:
 - è più semplice per noi imparare Python che per il computer imparare l’Italiano (o l’Inglese)
 - noi siamo quelli intelligenti e pazienti!

Strumenti per usare Python

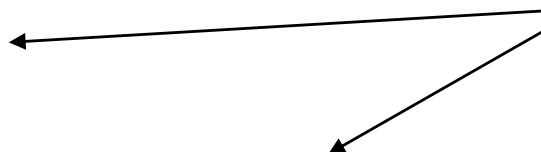
Interprete Python

- Può essere scaricato e installato da qui
<https://www.python.org>

Vari ambienti di programmazione

- Spyder
- Jupyter
- Anaconda

Useremo qualcuno
tra questi



<https://www.pythontanywhere.com>

Elementi di Python

Vocabolario: parole riservate e variabili

Struttura delle frasi: istruzioni

Struttura della storia: costruire un programma che ha uno scopo

Parole riservate

Parole che hanno un significato specifico

```
False  class  return  is      finally  
None   if     for      lambda continue  
True   def    from     while   nonlocal  
and    del    global   not    with  
as     elif   try     or     yield  
assert else   import  pass  
break  except  in     raise
```

Variabili

Una variabile corrisponde a una posizione nella memoria del calcolatore

Può essere usata dai programmati per immagazzinare dati e poi recuperarli

I programmati attribuiscono un nome a ciascuna variabile (non possono usare le parole riservate come nomi di variabile)

Il contenuto di una variabile può cambiare nel tempo

`x = 12.2`

`y = 14`

`x 12.2`

`y 14`

Variabili

Una variabile corrisponde a una posizione nella memoria del calcolatore

Può essere usata dai programmati per immagazzinare dati e poi recuperarli

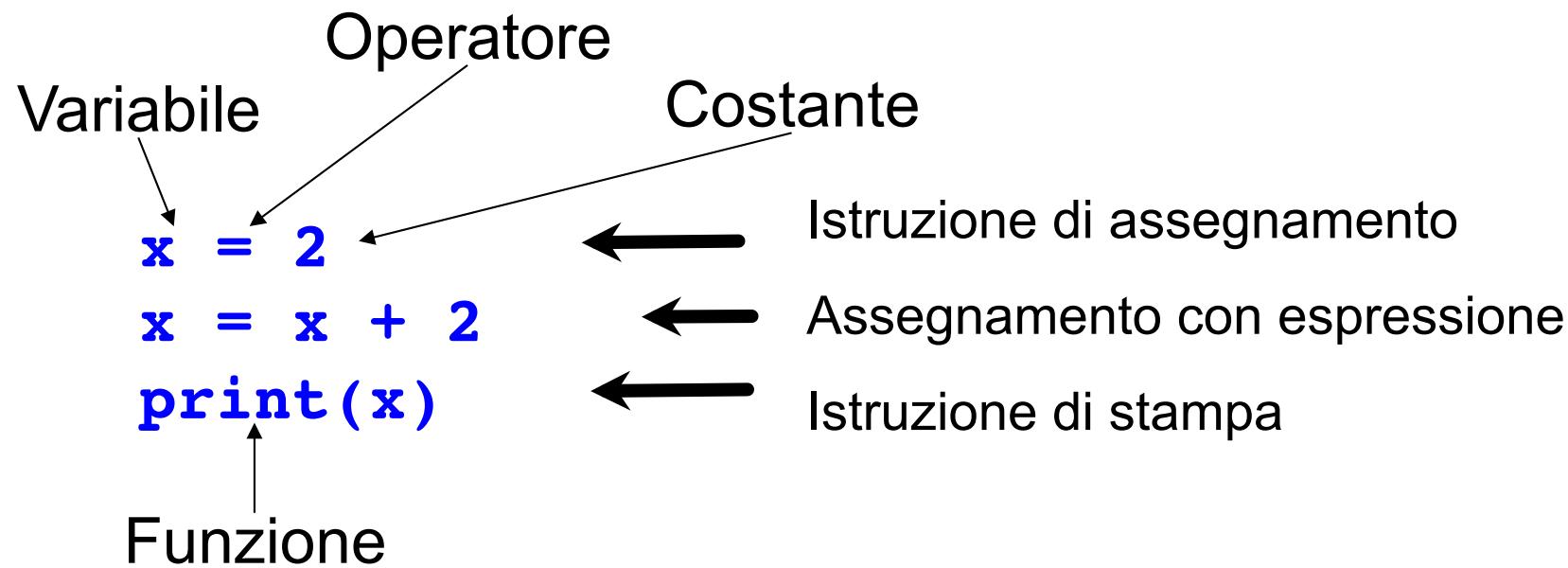
I programmati attribuiscono un nome a ciascuna variabile (non possono usare le parole riservate come nomi di variabile)

Il contenuto di una variabile può cambiare nel tempo

```
x = 12.2  
y = 14  
x = 100
```

x	12.2	100
y	14	

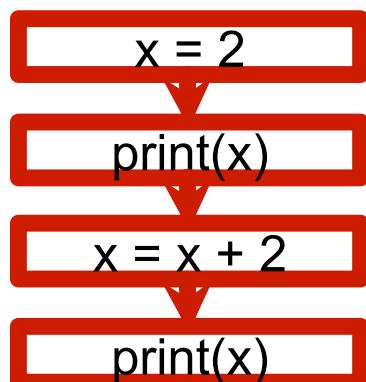
Frasi



Passi di un programma (istruzioni) e flusso di programma

- Come una ricetta o le istruzioni di installazione, un programma è una **sequenza** di passi da fare in ordine
- Qualche passo condizionato alla verifica di una **condizione** e a volte può essere saltato
- Qualche volta un passo o un gruppo di essi deve essere **ripetuto** più volte (cicli)
- Qualche volta alcuni passi sono raggruppati e richiamati per essere usati più volte laddove servono nel programma (queste sono le funzioni)

Passi (istruzioni) Sequenziali



Programma:

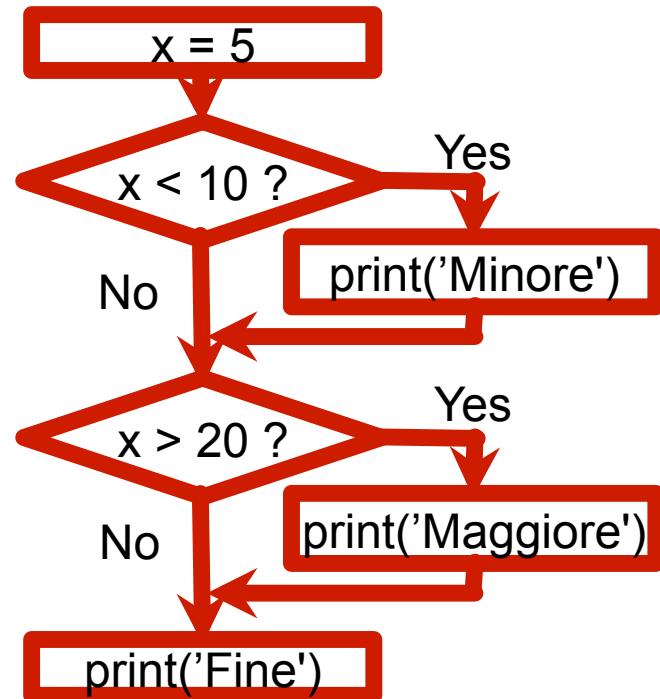
```
x = 2  
print(x)  
x = x + 2  
print(x)
```

Output:

```
x = 2  
print(x) --> 2  
x = x + 2 --> 4  
print(x)
```

When a program is running, it flows from one step to the next. As programmers, we set up “paths” for the program to follow.

Passi Condizionali



Programma:

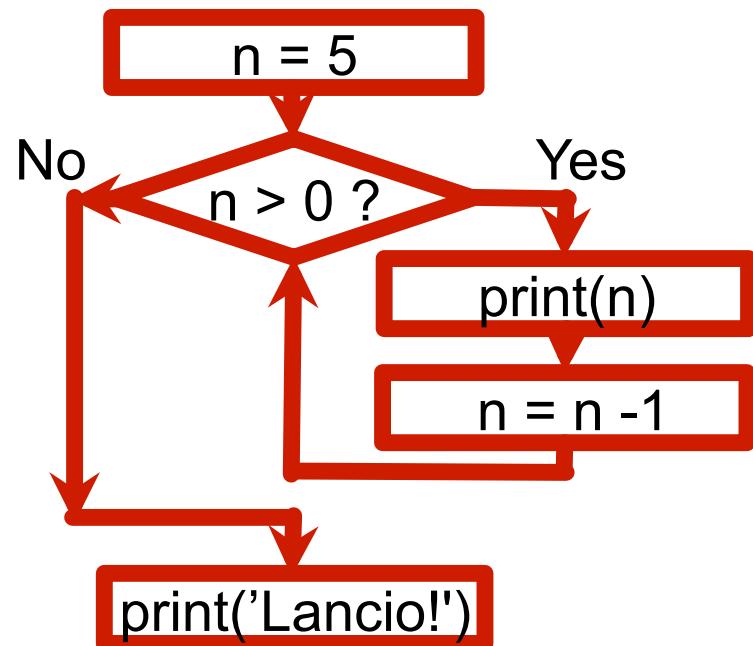
```
x = 5
if x < 10:
    print('Minore')
if x > 20:
    print('Maggiore')

print('Fine')
```

Output:

Minore
Fine

Passi ripetuti (Cicli)



Program:

```
n = 5  
while n > 0 :  
    print(n)  
    n = n - 1  
print('Lancio!')
```

Output:

```
5  
4  
3  
2  
1  
Lancio!
```

Loops (repeated steps) have iteration variables that change each time through a loop.

Classifichiamo le istruzioni del primo esempio...

```
text = input('Write a sentence: ')  
  
Alphabet = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i',  
            'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r',  
            's', 't', 'u', 'v', 'w', 'x', 'y', 'z']  
  
notAPangram = False  
  
for letter in Alphabet:  
    if letter not in text:  
        notAPangram = True  
  
if notAPangram:  
    print('the string is not a pangram')  
else:  
    print('the string is a pangram')
```

Sequential

Repeated

Conditional

Classifichiamo le istruzioni del primo esempio...

```
text = input('Write a sentence: ')  
  
Alphabet = [ 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i',  
             'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r',  
             's', 't', 'u', 'v', 'w', 'x', 'y', 'z']  
  
notAPangram = False  
  
for letter in Alphabet:  
    if letter not in text:  
        notAPangram = True  
  
if notAPangram:  
    print('the string is not a pangram')  
else:  
    print('the string is a pangram')
```

Sequential

Repeated

Conditional

“Parole, Frasi e Paragrafi”... Come scrivere una Python Story

```
text = input('Write a sentence: ')  
  
Alphabet = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i',  
            'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r',  
            's', 't', 'u', 'v', 'w', 'x', 'y', 'z']  
  
notAPangram = False  
  
for letter in Alphabet:  
    if letter not in text:  
        notAPangram = True  
  
if notAPangram:  
    print('the string is not a pangram')  
else:  
    print('the string is a pangram')
```

A short Python “Story” about how identifying pangrams

A word used to read data from a user

A sentence about considering all alphabet letters

A paragraph about informing the usage of the program outcome

Interagire con l'interprete Python

- Intettivo
 - Scriviamo direttamente all'interprete Python una linea per volta e lui risponde immediatamente;
- Script
 - Inseriamo una sequenza di istruzioni (linee) in un file usando un qualsiasi editor di testo e diciamo all'interprete Python di eseguirlo fino alla fine.
 - Come convenzione l'estensione di uno script Python è “.py” Questo ci fa capire che il file non contiene puro testo ma delle istruzioni Python

Acknowledgements / Contributions



Part of these slides are Copyright 2010- Charles R. Severance (www.dr-chuck.com) of the University of Michigan School of Information and made available under a Creative Commons Attribution 4.0 License.

Adaptation and extensions: Elisabetta Di Nitto e Gianluca Palermo, Politecnico di Milano