

Fondamenti di Informatica I

Corso di Laurea in
Ingegneria Informatica e Automatica



SAPIENZA
UNIVERSITÀ DI ROMA

Aspetti Organizzativi e Introduzione

Professori:

Marco Schaerf (canale 1: A-MA)

Domenico Lembo (canale 2: ME-Z)

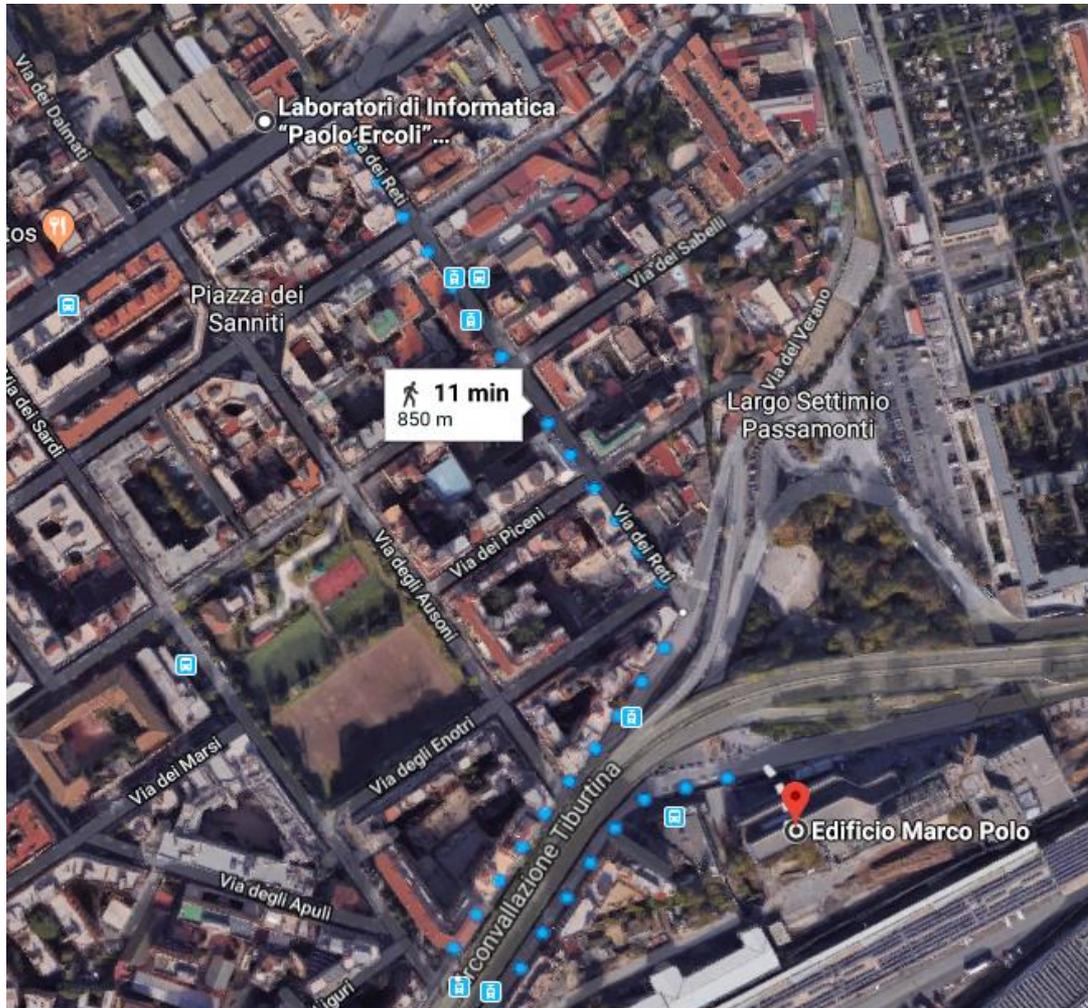
Prima lezione

- Aspetti organizzativi
- Descrizione del corso
- Introduzione ai linguaggi di programmazione ed a Python
- Prime istruzioni in Python
- L'editor IDLE e l'uso della Shell dei comandi

Informazioni generali sul corso

- 9 CFU (a partire dall'AA 2017-18, prima era da 12 CFU)
- Orario delle lezioni (23 Settembre 2019 – 20 Dicembre 2019)
 - Lunedì 10:30–13:00 – in Aula
 - Martedì 08:00–10:30 – in Aula (saltuariamente, previo preavviso)
 - Martedì 11:00–15:00 – in Laboratorio
 - Giovedì 10:30–13:00 – in Aula
- Aule
 - Presso l'edificio Marco Polo (Circonvallazione Tiburtina 4)
 - Canale 1 (prof. M. Schaerf): Aula 204
 - Canale 2 (prof. D. Lembo): Aula 108
- Laboratori
 - Presso il laboratorio Paolo Ercoli (via Tiburtina 205)
 - Laboratorio 15 e 16 (indipendentemente dal canale)

Esercitazioni presso il laboratorio Paolo Ercoli



Via Tiburtina, 205

*Le esercitazioni in laboratorio si svolgeranno ogni martedì **a partire da domani (24/9)**.*

Saranno organizzate in due turni da 1h e 45m ciascuno.

Turni del 24/9:

Matr pari: 11:00-12:45

Matr dispari: 13:00-14:45

Il corso in estrema sintesi

- Prerequisiti: Nessuno (veramente)
- I contenuti del corso sono divisi in due parti:
 - **Programmazione** (circa 2/3 del corso): semplici programmi scritti in un linguaggio comprensibile da un calcolatore. In questo corso studiamo il linguaggio **Python**.
 - **Teoria: “Modelli dell’informatica”** (circa 1/3 del corso): l’informatica non è solo programmazione

Il corso non è facile:

- ***Impegnarsi e studiare fin dall’inizio*** (anche se non ci sono interrogazioni in classe)
- *Esercitarsi nella programmazione (Laboratorio)*

Programma: Programmazione in Python

- **Nozioni introduttive:** Il calcolatore. Algoritmi e Programmi. Linguaggi di programmazione. La compilazione.
- **Nozioni elementari sulla programmazione in Python:** I/O di base. Uso dell'ambiente di sviluppo.
- **Aspetti di base della programmazione in Python:** Espressioni aritmetiche e tipi di dati elementari. Variabili e istruzioni di assegnazione. Il tipo di dato stringa.
- **Decisioni:** costrutti if ed else-if (elif).
- **Cicli:** Ciclo while. Ciclo for. Cicli annidati.
- **Funzioni e moduli:** Introduzione alla programmazione Python con funzioni. Moduli e loro uso. Esecuzione delle funzioni.
- **Liste:** Proprietà di base. Operazioni sulle liste. Algoritmi elementari che fanno uso di liste. Rappresentazione di tabelle e matrici.
- **Dizionari:** Proprietà di base. Accesso e manipolazione di dizionari.
- **File e file system:** Apertura, chiusura e manipolazione di file di testo. Funzioni di base per l'accesso al file system.

Programma: Modelli dell'Informatica

- **Architettura dei calcolatori:** Architettura di von Neumann. Esempi di Linguaggio Macchina. La legge di Moore.
- **Rappresentazione dell'Informazione:** Rappresentazione di caratteri e stringhe, Rappresentazione dei numeri positivi. Rappresentazione in complemento a due. Rappresentazione in virgola mobile.
- **Logica Proposizionale:** Tavole di verità. Connettivi logici. Soddisfacibilità, validità, equivalenza logica. Conseguenza Logica. Forme normali. Deduzione automatica.
- **Calcolabilità e complessità:** Problemi di decisione. Problemi indecidibili. Tempo di esecuzione di un programma e la notazione O-grande. Le classi di complessità P ed NP.
- **Espressioni regolari:** Funzione di ricerca. Esempi di utilizzo di espressioni regolari. Sintassi e semantica delle espressioni regolari.
- **Automati a Stati finiti:** Espressioni regolari vs. automi. Funzionamento degli automi. Progettazione di automi. Automi non deterministici.

Appelli d'esame

- 2 Appelli fra il 7 gennaio ed il 20 febbraio 2020
- 2 Appelli fra il 1 giugno ed il 24 luglio 2020
- 1 Appello fra il 1 settembre ed il 17 settembre 2019

Appelli straordinari:

- 1 Appello fra il 16 marzo ed il 16 aprile 2020, riservato a studenti part-time, fuori corso nell'A.A. 2019-2020, studenti con disabilità o con D.S.A.
- 1 Appello fra il 5 ottobre ed il 5 novembre 2020, riservato a studenti part-time, fuori corso nell'A.A. 2019-2020, studenti con disabilità o con D.S.A. *ed agli studenti iscritti al terzo anno nell'A.A. 2019-2020.*

Modalità d'esame

- La prova d'esame consiste di:
 - **Una prova scritta**, durante la quale lo studente dovrà rispondere ad alcuni quesiti ed esercizi relativi alla parte “**Modelli dell'Informatica**”. Generalmente vengono proposti 4 quesiti o esercizi e lo studente ha a disposizione circa 40 minuti per rispondere.
 - **Una prova pratica al calcolatore**, durante la quale lo studente dovrà realizzare alcune funzioni in **Python**. Generalmente vengono proposti 4 esercizi di questo tipo e lo studente ha a disposizione circa 1h e 30m per risolverli.
- La prova è identica e si svolge nello stesso giorno per i due canali
- Il voto finale è stabilito tramite una media pesata dei voti delle due parti (che quindi contribuiscono rispettivamente per 1/3 e 2/3 alla definizione del voto) ⁹

Prova Intermedia

- Per questo anno accademico, in via sperimentale, è stata introdotta una prova valutativa intermedia che sarà possibile sostenere **il 5 novembre**.
- La prova intermedia conterrà esercizi sulla **prima metà del corso**, che comprende tutto quanto sarà presentato a lezione fino alla settimana precedente la prova stessa.
- il superamento della prova intermedia equivale al **superamento del 50% dell'esame**. Questo vuol dire che lo studente dovrà sostenere un'ulteriore prova d'esame che verterà solo sugli argomenti presentati durante la seconda parte del corso.
- La **seconda prova** potrà essere sostenuta **durante un qualunque appello d'esame**.
- Maggiori dettagli sulla Prova Intermedia verranno dati nelle prossime lezioni

Materiale Didattico

- Per la parte di Programmazione
 - Libro di Testo:

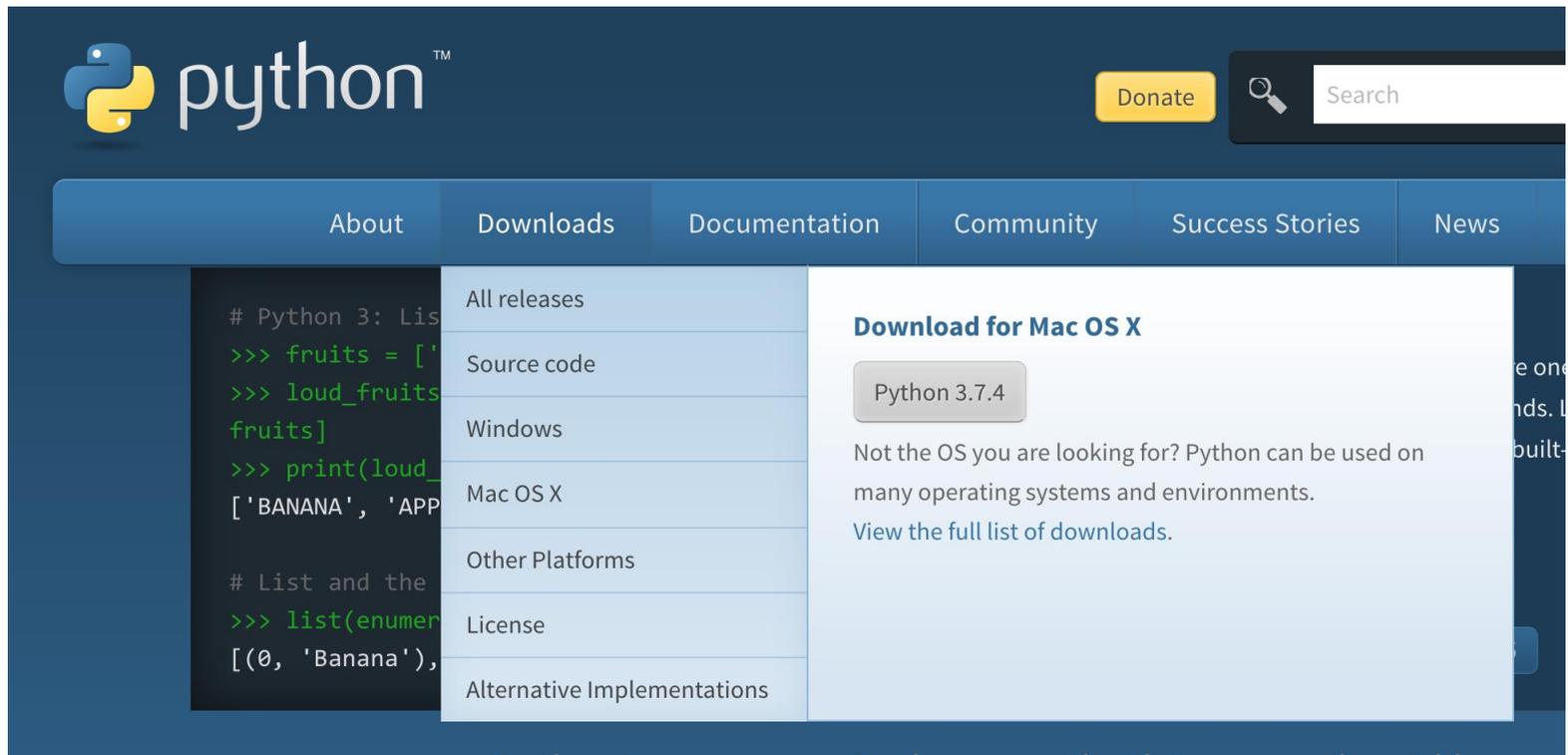
C. Horstmann, R. D. Necaie.
Concetti di Informatica e fondamenti di Python.
Maggioli Editore
(qualunque edizione)
 - **Slide e dispense del corso**
(distribuite sul sito)
 - **Testi e soluzioni delle esercitazioni**
- Per la parte di Modelli
 - **Slide e dispense** (distribuite sul sito)



Ambiente di Lavoro per la Programmazione

– Python 3.7.4 – disponibile al sito

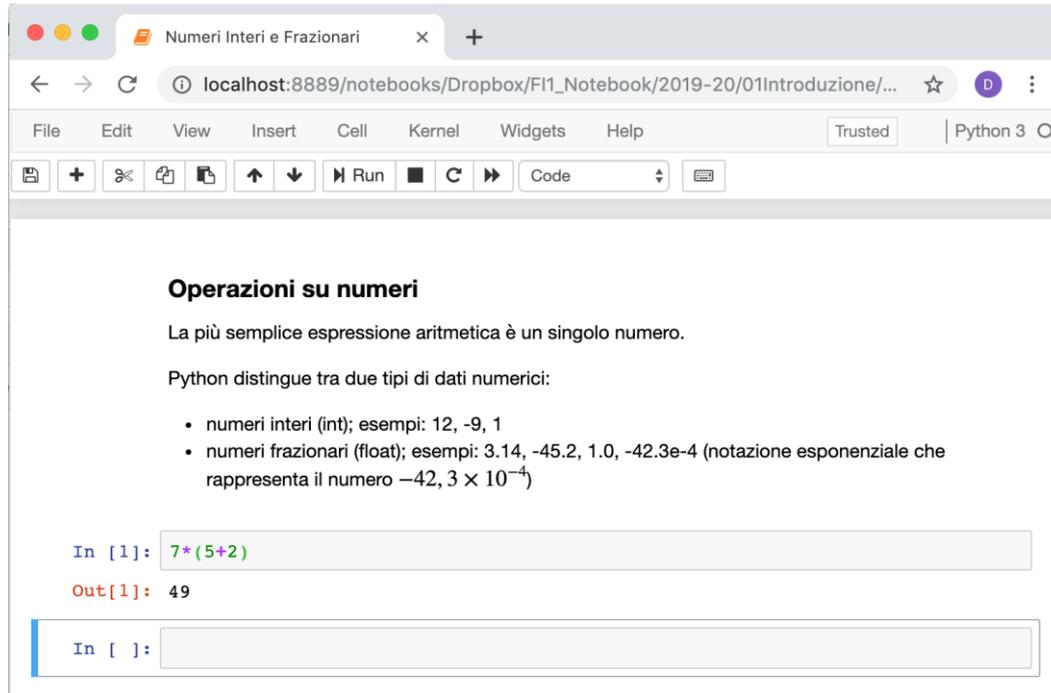
<https://www.python.org/>



– Editor **IDLE**, incluso nell'installazione di Python

Consultazione file Notebook su Python (distribuiti sul sito del corso)

Le note del corso su Python saranno distribuite nella forma di file Notebook (file .ipynb) che possono essere consultati Jupyter Notebook (o Jupyter Lab)  jupyter



Operazioni su numeri

La più semplice espressione aritmetica è un singolo numero.

Python distingue tra due tipi di dati numerici:

- numeri interi (int); esempi: 12, -9, 1
- numeri frazionari (float); esempi: 3.14, -45.2, 1.0, -42.3e-4 (notazione esponenziale che rappresenta il numero $-42,3 \times 10^{-4}$)

```
In [1]: 7*(5+2)
```

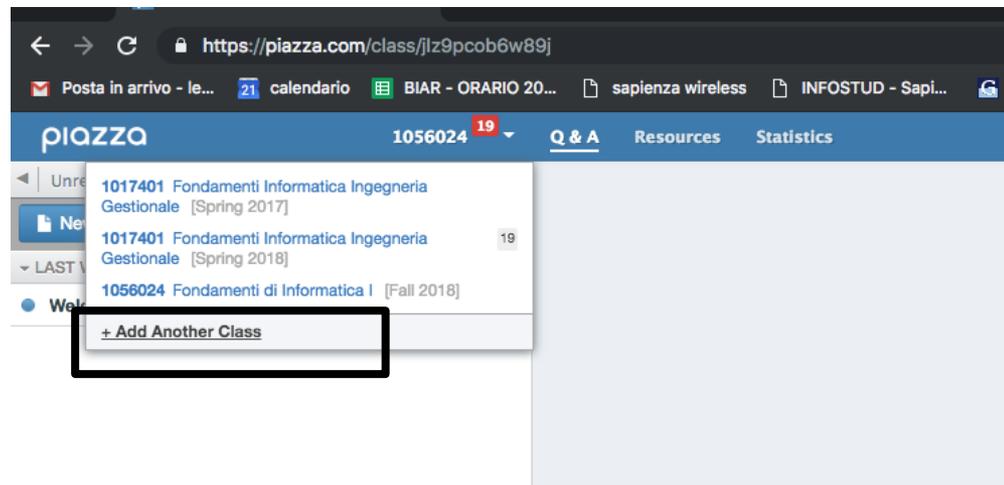
```
Out[1]: 49
```

```
In [ ]:
```

Istruzioni relative all'istallazione degli ambienti di lavoro sono disponibili sul sito del corso

Sito Web del Corso

- Sulla piattaforma **Piazza**
<https://piazza.com/uniroma1.it/fall2019/1056024/home>
- Sul sito verranno pubblicate tutte le informazioni relative al corso, gli avvisi, il materiale didattico.
- **Si raccomanda la registrazione al sito.** Questa consente l'accesso a tutte le risorse ed al **forum** del corso.
- Chi è già registrato al sito Piazza, può accedere con le sue credenziali e selezionare 'Add Another Class' dal menu in alto a sinistra



Registrazione al sito

- Andate sul sito <https://piazza.com>

piazza

Product

In Professors' Words

Support

About Us

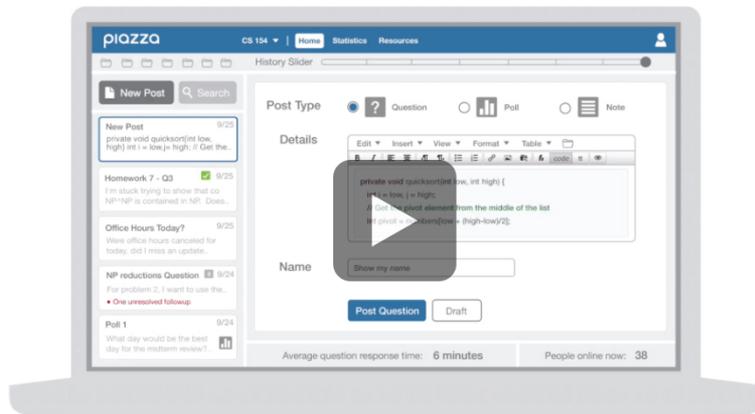
Piazza For Companies

Sign Up

Login

The incredibly easy, completely free Q&A platform

Save time and help students learn using the power of community



- Wiki style format enables collaboration in a single space
- Features LaTeX editor, highlighted syntax and code blocking
- Questions and posts needing immediate action are highlighted
- Instructors endorse answers to keep the class on track
- Anonymous posting encourages every student to participate
- Highly customizable online polls
- Integrates with every major LMS and is FERPA compliant

Students Get Started

Professors and TAs Get Started

View a Real Class

Registrazione al sito

- Andate sul sito <https://piazza.com>
- Selezionate **Sign Up**

piazza

Product

In Professors' Words

Support

About Us

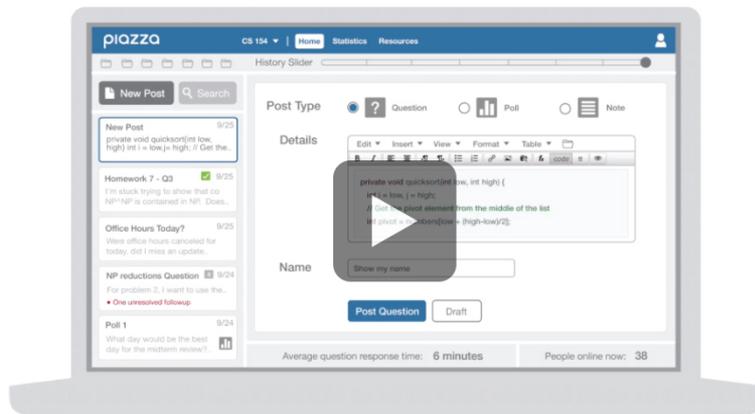
Piazza For Companies

Sign Up

Login

The incredibly easy, completely free Q&A platform

Save time and help students learn using the power of community



- Wiki style format enables collaboration in a single space
- Features LaTeX editor, highlighted syntax and code blocking
- Questions and posts needing immediate action are highlighted
- Instructors endorse answers to keep the class on track
- Anonymous posting encourages every student to participate
- Highly customizable online polls
- Integrates with every major LMS and is FERPA compliant

Students Get Started

Professors and TAs Get Started

View a Real Class

Registrazione al sito

- Andate sul sito <https://piazza.com>
- Selezionate **Sign Up**
- Selezionate **Student Get Started**

piazza

Product

In Professors' Words

Support

About Us

Piazza For Companies

Sign Up

Login

The incredibly easy, completely free Q

Save time and help students learn using the power

Instructors Get Started

Students Get Started

Registrazione al sito

- Andate sul sito <https://piazza.com>
- Selezionate Sign Up
- Selezionate Student Get Started
- Inserite Sapienza
- Selezionate Sapienza University of Rome

Welcome to Piazza!

Piazza is a free platform for instructors to efficiently manage class Q&A. Students can post questions and collaborate to edit responses to these questions. Instructors can also answer questions, endorse student answers, and edit or delete any posted content.

Piazza is designed to simulate real class discussion. It aims to get high quality answers to difficult questions, fast!

The name Piazza comes from the Italian word for plaza—a common city square where people can come together to share knowledge and ideas. We strive to recreate that communal atmosphere among students and instructors.

Are you a professor?

Click here to create & join classes

Search Schools:

Sap	
Searching for "Sap"	
Sapienza-University of Rome (uniroma1.it)	54 Classes
Ingegneria Informatica e Automatica - La Sapienza	3 Classes
Apopka High School (https://sapopkahs.ocps.net)	1 Classes
Chesapeake Math & IT Academy	0 Classes
sapienza ingegneria gestionale	0 Classes
sapkal knowledge hub	0 Classes
iOS Apps Development (iosappsdev.org)	0 Classes
St Andrews Presbyterian College (sapc.edu)	0 Classes
Haliwa-Sapponi Tribal School	0 Classes

to add your classes...

Registrazione al sito

- Selezionate il Term: **Fall 2019**

Welcome to Piazza!

Piazza is a free platform for instructors to efficiently manage class Q&A. Students can post questions and collaborate to edit responses to these questions. Instructors can also answer questions, endorse student answers, and edit or delete any posted content.

Piazza is designed to simulate real class discussion. It aims to get high quality answers to difficult questions, fast!

The name Piazza comes from the Italian word for plaza--a common city square where people can come together to share knowledge and ideas. We strive to recreate that communal atmosphere among students and instructors.

Are you a professor?
Click here to create & join classes

Sapienza University of Rome

Select a Term
Summer 2018
Fall 2018
Winter 2019
Spring 2019
Summer 2019
✓ **Fall 2019**
Winter 2020
Spring 2020
Summer 2020
Fall 2020
Other

Selected Term: Fall 2019

Class 1:

Class 2: ×

Class 3: ×

Class 4: ×

Class 5: ×

[Add Another Class](#)

[Join Classes](#)

Registrazione

- Selezionate il Term: Fall 2019
- Inserite il codice del corso: 1056024
- Selezionate il corso

Sapienza-University of Rome

(change school)

Welcome to Piazza!

Piazza is a free platform for instructors to efficiently manage class Q&A. Students can post questions and collaborate to edit responses to these questions. Instructors can also answer questions, endorse student answers, and edit or delete any posted content.

Piazza is designed to simulate real class discussion. It aims to get high quality answers to difficult questions, fast!

The name Piazza comes from the Italian word for plaza--a common city square where people can come together to share knowledge and ideas. We strive to recreate that communal atmosphere among students and instructors.

Are you a professor?
Click here to create & join classes

Selected Term:

Fall 2019

Class 1:	<input type="text" value="1056024"/>	
Class 2:	<input type="text" value='Searching for "1056024"'/>	×
Class 3:	1056024: Fondamenti di Informatica I	5 Enrolled ×
Class 4:	<input type="text"/>	×
Class 5:	<input type="text"/>	×

[Add Another Class](#)

[Join Classes](#)

Registrazione

- Selezionate il Term: Fall 2019
- Inserite il codice del corso: 1056024
- Selezionate il corso
- Selezionate **Join as: Student** e **Join classes**

Sapienza-University of Rome

(change school)

Welcome to Piazza!

Piazza is a free platform for instructors to efficiently manage class Q&A. Students can post questions and collaborate to edit responses to these questions. Instructors can also answer questions, endorse student answers, and edit or delete any posted content.

Piazza is designed to simulate real class discussion. It aims to get high quality answers to difficult questions, fast!

The name Piazza comes from the Italian word for plaza--a common city square where people can come together to share knowledge and ideas. We strive to recreate that communal atmosphere among students and instructors.

Are you a professor?
Click here to create & join classes

Selected Term:

Fall 2019

Class 1: 1056024: Fondamenti di Informatica I (edit)
Instructors: Marco Schaerf · 5 Enrolled
✓ **Join as:** Student *Instructor self-enrollment has been disabled for this class.*

Class 2: ×

Class 3: ×

Class 4: ×

Class 5: ×

[Add Another Class](#)

Registrazione al sito

- Inserire la mail istituzionale (studenti.uniroma.it)
- Selezionare **Submit Email**

Sapienza-University of Rome

(change school)

Selected Term: [\(go back & edit classes\)](#)

Fall 2019

1. 1056024: Fondamenti di Informatica I
Instructors: Marco Schaerf · 5 Enrolled
✓ **Joining as Student**

Please enter your school email address

Please enter the **uniroma1.it** or **studenti.uniroma1.it** email address to which you would like to add your classes.

Email:

Confirm Email:

Submit Email

Registrazione al sito

- Riceverete una mail contenente un link
- Selezionare il link (copiate ed incollate sul browser se non venite ridirezionati automaticamente alla pagina)
- Riempite la form come Major (corso di laurea)
- Scrivete Ingegneria Informatica e Automatica
- In Graduation Date (data di laurea) selezionare la data in cui pensate di laurearvi (alla triennale!)



Introduzione:

L'elaborazione delle Informazioni

Elaborazione delle Informazioni

Un sistema per la rappresentazione e l'elaborazione di informazioni può

- raccogliere impressionanti quantità di dati
- elaborare i dati secondo le istruzioni fornite producendo nuovi dati
- rendere disponibili questi dati in modo istantaneo e con prospettive diverse a utenti diversi e in parti diverse del mondo

Esempi: social networks, banca via internet, calcolo scientifico, video giochi

Elaborazione delle Informazioni

- Computer (calcolatore, microprocessore...)
 - Esegue semplici operazioni aritmetiche e logiche ad una velocità molto maggiore degli esseri umani (1 nanosecondo per operazione = 1 milionesimo di millesimo di secondo)
- Programma
 - Insieme di istruzioni eseguibili da un computer
- Hardware
 - Dispositivi fisici di un sistema di elaborazione
- Software
 - Programmi eseguiti su un elaboratore

Cos'è l'informatica?

L'informatica NON è lo studio dei computer come artefatti. Non studieremo come si costruisce un computer (o come lo si ripara).

Siamo interessati a USARE i Computer

- L'informatica ha a che fare con lo studio di risoluzione di problemi (“problem solving”) per la cui soluzione si scrive un programma che viene eseguito da un computer
- L'informatica si basa su una serie di tradizioni intellettuali che comprende aspetti della matematica e dell'ingegneria

Cos'è l'informatica?

L'informatica svolge un ruolo sempre più importante in altre discipline:

- Biologia: analizzare il genoma umano.
- Economia: la creazione di migliori modelli economici e finanziari.
- Ambiente: i modelli climatici richiedono la moderna tecnologia informatica.
- Letteratura: Analisi computerizzata aiuta a risolvere paternità contestata.
- Cinema e intrattenimento: utilizzo di informatica grafica nei film e nei videogiochi.

Programma e dati di ingresso

Un ***programma*** è una sequenza di istruzioni scritte in un linguaggio di programmazione (e comprensibili da un calcolatore) che realizzano un algoritmo

Dati di ingresso: di solito un problema non deve essere eseguito una sola volta ma molte volte con dati (di ingresso) diversi

Esempi:

- Trovare il percorso più breve in una rete stradale:
dati di ingresso: rete stradale in esame, punto di partenza e punto di arrivo
- Eseguire una ricerca nel web: dati di ingresso: web (in quel momento), parole date nella ricerca

Elaborazione dell'informazione

Risoluzione di un problema

1. Dato un problema **si trova un algoritmo** di soluzione
2. Si scrive l'algoritmo di soluzione in un **programma** comprensibile da un calcolatore
3. Dato un programma, un calcolatore e dei dati di ingresso, **l'esecuzione del programma con i dati di ingresso** sul calcolatore risolve il problema su quei dati

Elaborazione dell'informazione

Come viene risolto un problema :



Algoritmi e Programmi

L'informatica studia la risoluzione di problemi (“problem solving”) realizzando un **programma** di soluzione che viene eseguito da un computer

Un **algoritmo** è una “ricetta”, ovvero un procedimento, composto da una sequenza di istruzioni elementari, che consente di risolvere un problema.

Esempio ordinare alfabeticamente una serie di nomi, trovare le pagine più interessanti in una ricerca su un motore di ricerca (google), trovare il percorso più breve (tom tom)

Algoritmo

In modo informale, si può pensare a un algoritmo come ad una procedura per risolvere un problema

Una definizione più formale richiede che un algoritmo deve verificare:

- I passi (le **istruzioni**) dell'algoritmo devono essere definiti in modo chiaro e **non ambiguo**
- Efficace, nel senso che i suoi passi siano **eseguibili da una macchina**
- Finito, nel senso che **termina dopo un numero fissato di operazioni**

I calcolatori e la programmazione

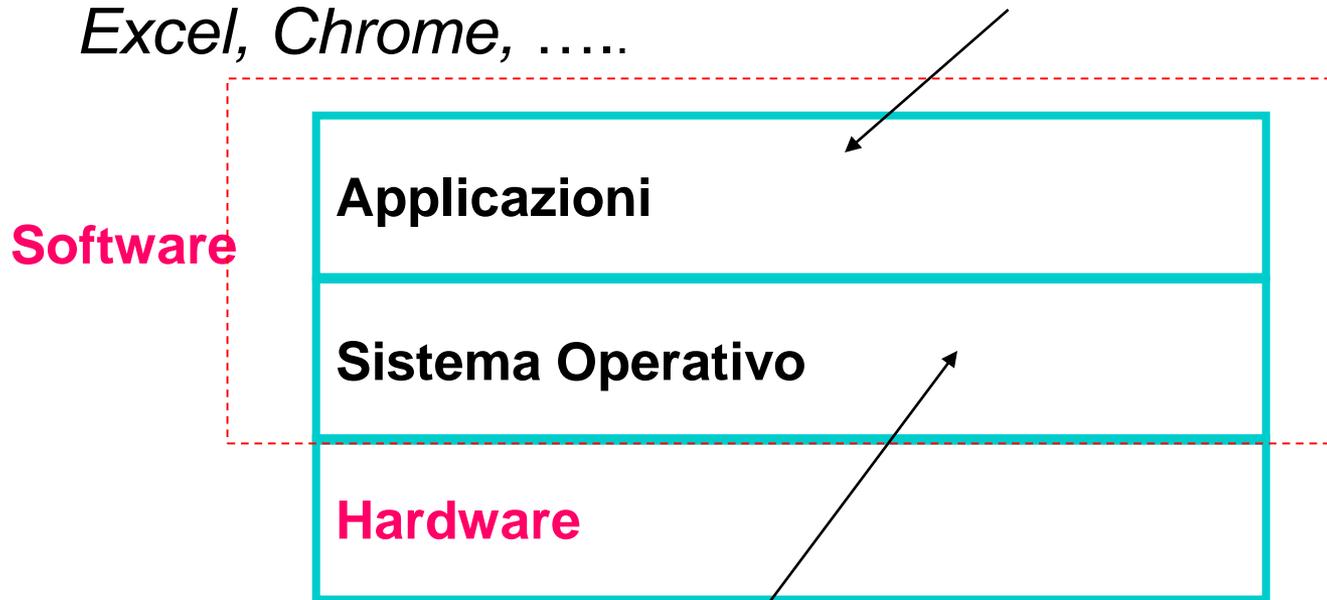
Programmare: capacità di specializzare il dispositivo per attività complesse di elaborazione dell'informazione

Programma: sequenza di istruzioni, ciascuna istruzione specifica una operazione da eseguire sui dati

Un **programma** è la codifica di un algoritmo scritta in un linguaggio non ambiguo e direttamente comprensibile dal computer.

Componenti del computer

Applicazioni: È il livello di SW con cui interagisce l'utente e comprende programmi quali: *Word, PowerPoint, Excel, Chrome,*



È il livello di SW che interagisce direttamente con l'HW e che si occupa di un uso corretto ed efficiente delle risorse fisiche

Windows
Unix/Linux
Mac OS

Linguaggi di programmazione

Dato un problema, un algoritmo di soluzione deve essere scritto in un linguaggio comprensibile da una macchina

I calcolatori sono in grado di comprendere solo programmi scritti in **linguaggio macchina** un linguaggio specifico della CPU e complicato per gli uomini

I programmi sono scritti in un linguaggio più facile da utilizzare; ce ne sono tantissimi che sono detti **linguaggi ad alto livello**; Python è uno di questi, semplice da imparare e quindi adatto per iniziare; ce ne sono ovviamente molti altri..

Linguaggi di programmazione

Ci sono molti linguaggi di programmazione:

Pascal, C, C++, Java, Ada, Perl, Php, Python...

- Questi linguaggi condividono gli stessi principi
- Ognuno però ha caratteristiche specifiche che lo distinguono dagli altri

Nelle applicazioni di oggi

- Elaborazione complesse dei dati sono spesso fatte usando linguaggi diversi
- Imparando I concetti fondamentali sarete in grado di apprendere da soli nuovi linguaggi

Linguaggi di programmazione

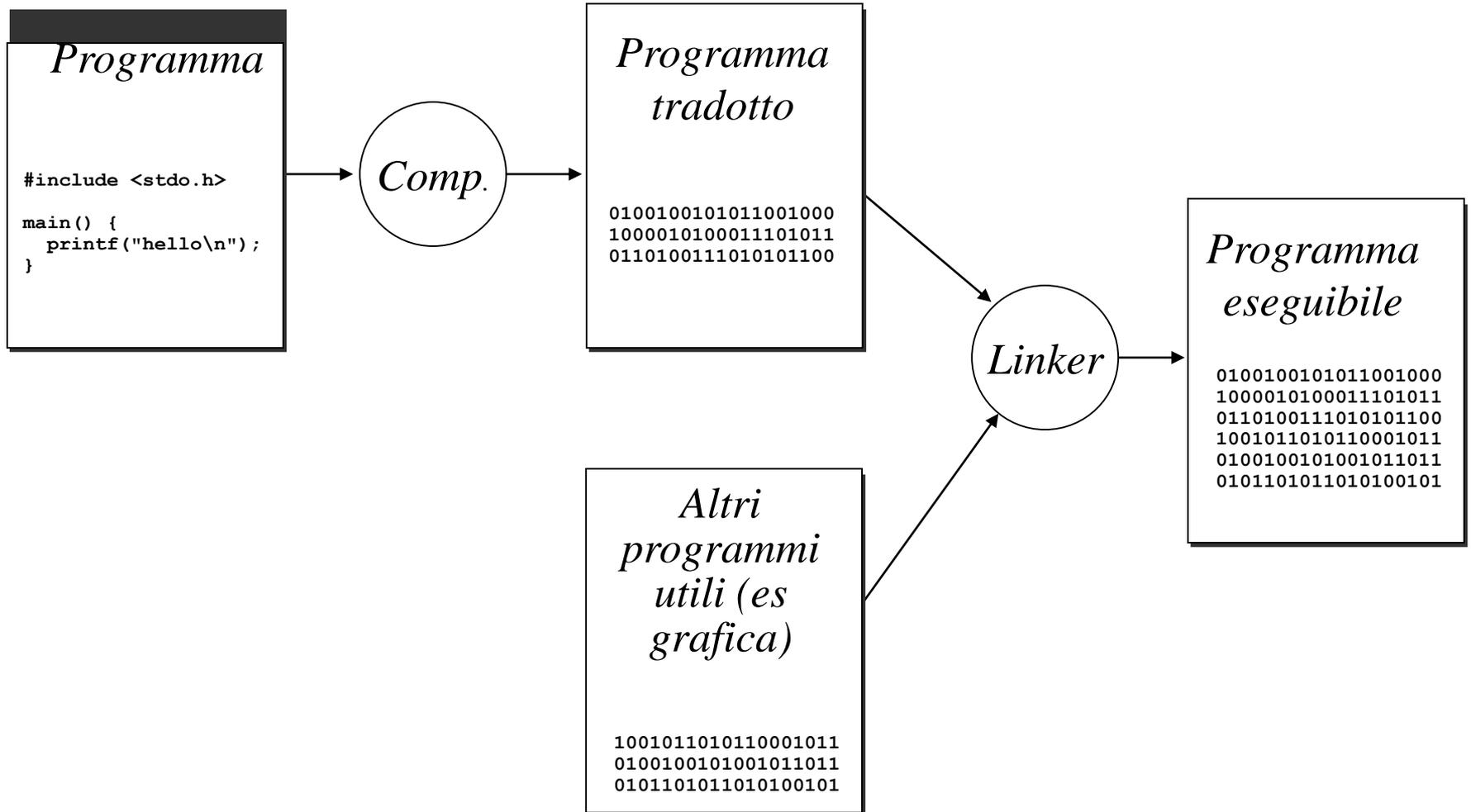
I linguaggi ad alto livello **non sono** direttamente eseguibili da un elaboratore.

Per eseguire un programma scritto in un linguaggio ad alto livello dobbiamo ***tradurlo*** nel linguaggio macchina

Due possibilità principali

- **Compilatore:** programma che traduce un linguaggio ad alto livello in un linguaggio macchina eseguibile dal sistema operativo
- **Interprete:** il programma scritto in linguaggio ad alto livello viene simulato senza tradurlo completamente.

Il processo di compilazione



Introduzione:
Il linguaggio Python

Python

- Sviluppato all' inizio degli Anni Novanta da Guido van Rossum.
 - Il suo nome fu scelto per via della passione dell' autore per i Monty Python e per la loro serie televisiva Monty Python' s Flying Circus (BBC, 1969 - 1974).
- L' intento era quello di progettare un linguaggio che consentisse di operare agevolmente con dati complessi e articolati:
 - Dinamicità
 - Semplicità
 - Flessibilità



Python

- Python è **free software**: il download dell'interprete e l'uso di Python è completamente gratuito, ma oltre a questo Python può essere liberamente modificato e ridistribuito, secondo le regole di una licenza pienamente *open-source*.
- Lo sviluppo di Python viene coordinato dall'organizzazione no-profit [Python Software Foundation](#).
- Guido van Rossum è rimasto per anni il “*Benevolo dittatore a vita*” di Python, colui che all’interno della comunità di sviluppatori avrebbe comunque avuto il peso maggiore nelle decisioni relative allo sviluppo del linguaggio.
- Dallo scorso anno van Rossum ha però lasciato il suo ruolo, per via delle crescenti difficoltà legate al suo svolgimento. Ha in particolare dichiarato:
"I don't ever want to have to fight so hard for a PEP (ndr: proposta di modifica al linguaggio) and find that so many people despise (sic) my decisions."

Python

- Supporta **diversi paradigmi di programmazione**:
 - **Imperativo** (enfasi sulle operazioni intese come **comandi** che cambiano lo stato dell'elaborazione)
 - **Funzionale** (enfasi sulle operazioni intese come **funzioni** che calcolano risultati)
 - **Orientato agli oggetti** (enfasi sugli **oggetti** che complessivamente rappresentano il dominio di interesse)
- Offre un **controllo dei tipi**
 - **Forte** (strong typing): Ogni elemento sintattico che denota un valore (ad es. variabile) è associato ad un determinato tipo durante l'esecuzione
 - **Eseguito a run-time** (dynamic typing): il controllo del tipo della variabile è effettuato a runtime

Python

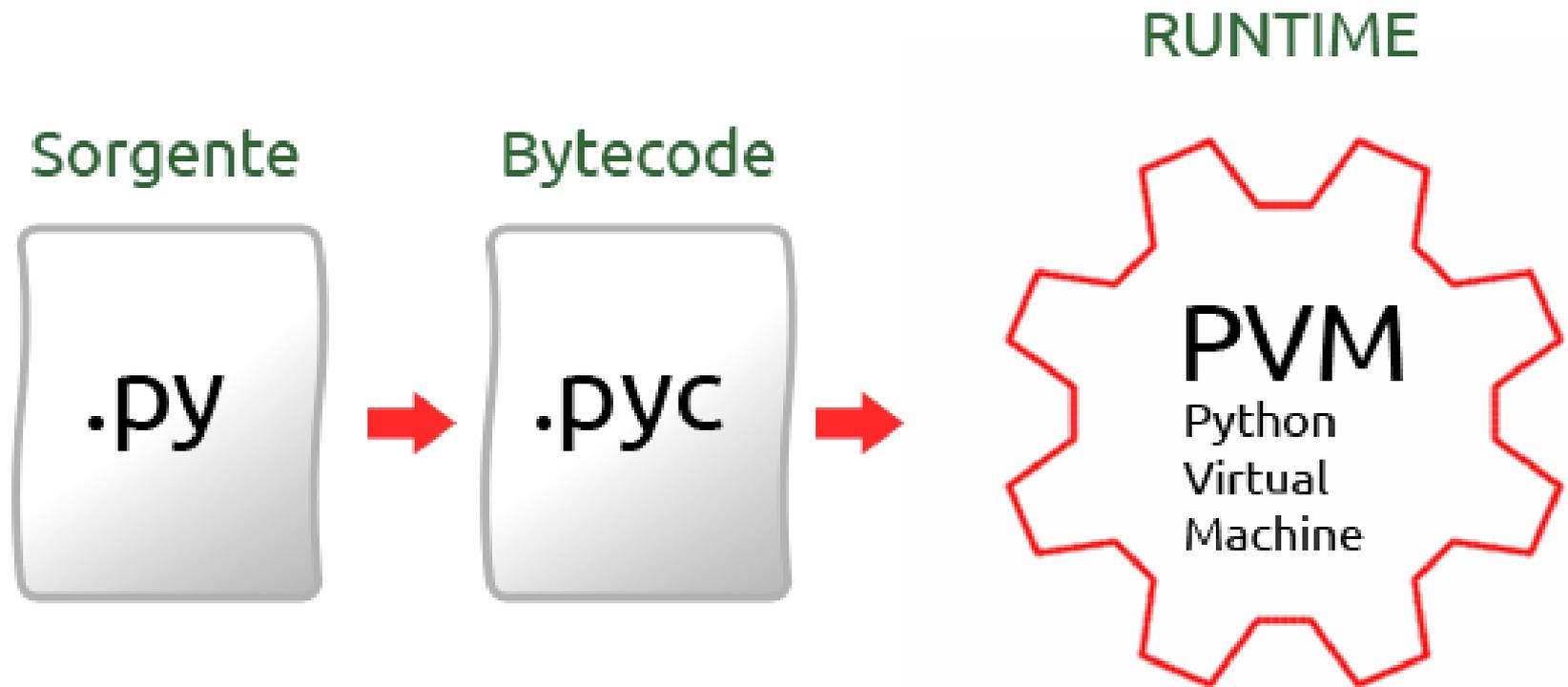
- Offre una gestione automatica della memoria.
 - Garbage collector.
- Fornito di una libreria built-in estremamente ricca.
- Offre costrutti robusti per la gestione delle eccezioni
 - come mezzo per segnalare e controllare eventuali condizioni di errore (incluse le eccezioni generate dagli errori di sintassi).

Python

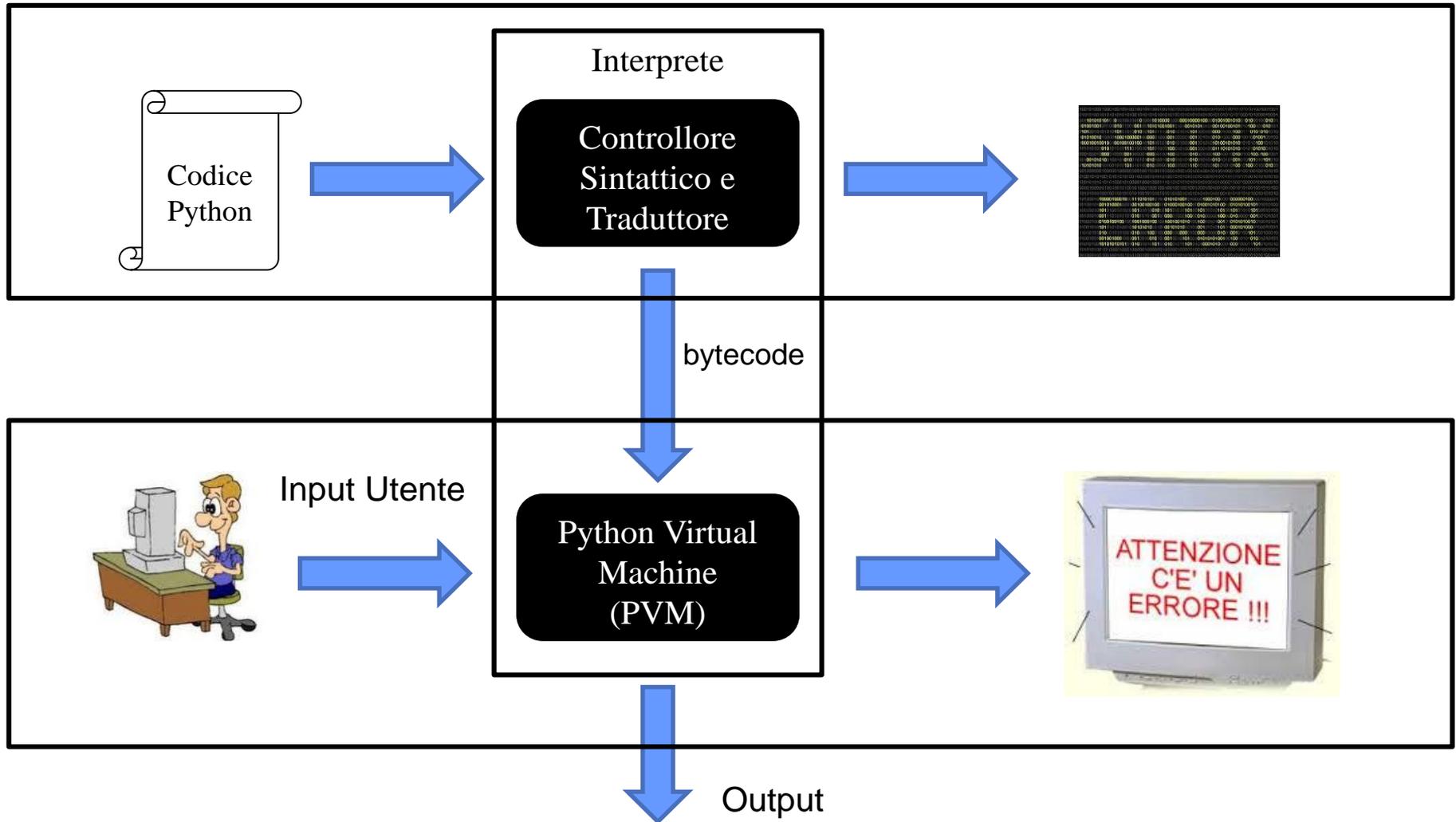
Linguaggio **interpretato** (o meglio pseudo-compilato):

- un interprete si occupa di analizzare il codice sorgente (semplici file testuali con estensione .py) e, se sintatticamente corretto, di eseguirlo.
- il codice sorgente passa prima da una fase di pre-compilazione in bytecode (riduce la dipendenza dall'hardware).
 - viene riutilizzato dopo la prima esecuzione del programma, evitando così di dover ogni volta interpretare il sorgente ed incrementando di conseguenza le prestazioni (in modo trasparente per l'utente).
- non esiste una fase di compilazione separata che generi un file eseguibile partendo dal sorgente.
 - il codice sorgente non viene convertito direttamente in linguaggio macchina, ma il bytecode prodotto viene eseguito da una **macchina virtuale Python (PVM)**
- Portabilità.
- Indipendenza dalla piattaforma.

Interpretazione di un programma Python (1\2)



Interpretazione di un programma Python (2\2)



Esecuzione di script .py

Per prima cosa bisogna generare un semplice file di testo* e salvarlo con estensione *.py*, per esempio, *helloworld.py*.

A questo punto possiamo aprire il file creato con un qualsiasi editor di testi (non Word, bensì Notepad, Notepad++, Sublime Text o simili) e scrivere:

```
print('Hello World!')
```

Un modo più agevole per creare gli script è usare un **Editor per programmi Python** (come ad esempio IDLE – vedi oltre).

*Nota: Si noti che, su Windows, potrebbe non essere immediato modificare l'estensione del file (che di default è *.txt*). In questo caso, occorre cambiare l'impostazione del sistema che nasconde le estensioni (oppure modificare l'estensione dall'editor di testo, se possibile).

Come primo esempio, abbiamo inserito nel file `helloworld.py` una sola riga di codice, costituita dalla funzione `print()`, che come risultato stamperà una stringa.

Eseguendo il file dalla shell dei comandi (anche detta terminale o console), otterremo quanto segue:

```
$ python helloworld.py  
Hello World!
```

Nota: Se usiamo un Editor per programmi Python (come IDLE), è possibile eseguire lo script aprendolo nell'editor e usando i comandi per l'esecuzione messi a disposizione dell'editor. Verrà automaticamente mostrata una finestra di shell con il risultato dell'esecuzione.

Modalità Interattiva

- In ogni installazione di Python è disponibile anche un interprete interattivo in grado di leggere e valutare man mano le espressioni inserite dall'utente.
- In modalità interprete interattivo digitando dei comandi Python si ottiene subito una risposta:

```
>>> 5 * 3
```

```
15
```

```
>>>
```

```
>>> a = 5
```

```
>>> b = 6
```

```
>>> 2 * (a+b) + 3*a
```

```
37
```



Ambiente di sviluppo Python

- Integrato
 - IDLE
- Editor di testo
- Finestra di terminale
- Modalità
 - Scripting.
 - Interactive.

Ambiente di sviluppo Python



```
Python 3.6.2 Shell
File Edit Shell Debug Options Window Help
Python 3.6.2 (default, Jul 29 2017, 00:00:00)
[GCC 4.8.4] on linux
Type "copyright", "credits" or "license()" for more information.
>>> # questa è la shell di Python
>>> # ogni istruzione viene eseguita appena premete INVIO
>>> 8 + 4
12
>>> |
```

```
Example.py - /home/biar/Desktop/Example.py (3.6.2)
File Edit Format Run Options Window Help
# Questo è l'editor dei file di Python
# tutte le istruzioni vengono eseguite insieme (Menu Run -> Run Module)
# le righe precedute da # vengono ignorate (indicano commenti)
# l'esecuzione NON valuta le espressioni singolarmente
# per vedere i risultati sulla shell usare il comando print()

print('Hello World')
print('The sum of 8 and 5 is')
print(8+5)
```

```
Python 3.6.2 Shell
File Edit Shell Debug Options Window Help
Python 3.6.2 (default, Jul 29 2017, 00:00:00)
[GCC 4.8.4] on linux
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: /home/biar/Desktop/Example.py =====
Hello World
The sum of 8 and 5 is
13
>>>
```