



Università degli Studi di Cagliari
Corsi di Laurea in Ingegneria Chimica e Ingegneria Meccanica

FONDAMENTI DI INFORMATICA

`http://people.unica.it/gianlucamarcialis`

A.A. 2019/2020

Docente: **Gian Luca Marcialis**

ALGORITMI E LINGUAGGIO PYTHON
Introduzione

Sommario

- Algoritmi, linguaggi ed ambienti di programmazione
- Il Linguaggio Python
- Il mio primo programma in Python
- Variabili
- Lettura di dati da tastiera

Informatica e algoritmi

- “L’informatica è lo studio sistematico degli **algoritmi** che descrivono e trasformano l’informazione: la loro teoria, analisi, progetto, efficienza, realizzazione e applicazione”

Association for Computing Machinery (ACM)

➤ Algoritmo

- sequenza **precisa** (comprensibile) di **passi elementari** che consentono di realizzare un compito, ovvero risolvere un problema
- passi elementari: eseguibili dall’esecutore dell’algoritmo
- es.: istruzioni di montaggio di un mobile, prelievamento di denaro da un terminale Bancomat, calcolo del massimo comune divisore di due numeri naturali...

Definizione formale di algoritmo

- Un algoritmo è un insieme **ordinato** di operazioni **non ambigue** ed **effettivamente computabili** che, quando eseguito, **produce un risultato** osservabile e si arresta **in un tempo finito**.
- Le proprietà di un algoritmo sono richiamate dai seguenti termini:
 - “Ordinato”, “non ambigue”: precisione
 - “Effettivamente computabili”, “produce un risultato”: correttezza
 - “In un tempo finito”: efficienza

Quindi...

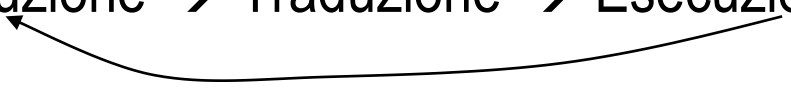
- Informatica come **studio degli algoritmi**, dei quali il calcolatore è un ottimo **esecutore**
- Un algoritmo comprensibile al calcolatore è espresso mediante **sequenze di bit**
- Tuttavia è complesso:
 - esprimersi in linguaggio macchina,
 - controllare la macchina della stessa (v. Sistemi Operativi)
- Dall'altra parte il linguaggio naturale non è ideale per scrivere algoritmi rispettandone le proprietà
- Esistono dei **linguaggi** che permettono la scrittura di programmi in una forma accessibile sia a noi che al calcolatore
- Per scrivere un programma esistono **gli ambienti di programmazione**

Interpreti e compilatori


- Per poter eseguire istruzioni di linguaggi di alto livello sulla *macchina fisica sottostante* esistono due possibilità:
 - **Interpretazione**, cioè si utilizza un programma chiamato *interprete* in grado di eseguire direttamente le istruzioni di alto livello
 - **Traduzione**, cioè la sequenza di istruzioni di alto livello viene prima tradotta, utilizzando un programma chiamato *compilatore*, in una corrispondente sequenza di istruzioni della macchina fisica

Linguaggi interpretati e compilati

➤ Interpretati

- Basic, Python, Matlab, Lisp, Smalltalk...
 - Programma → Istruzione → Traduzione → Esecuzione
- 

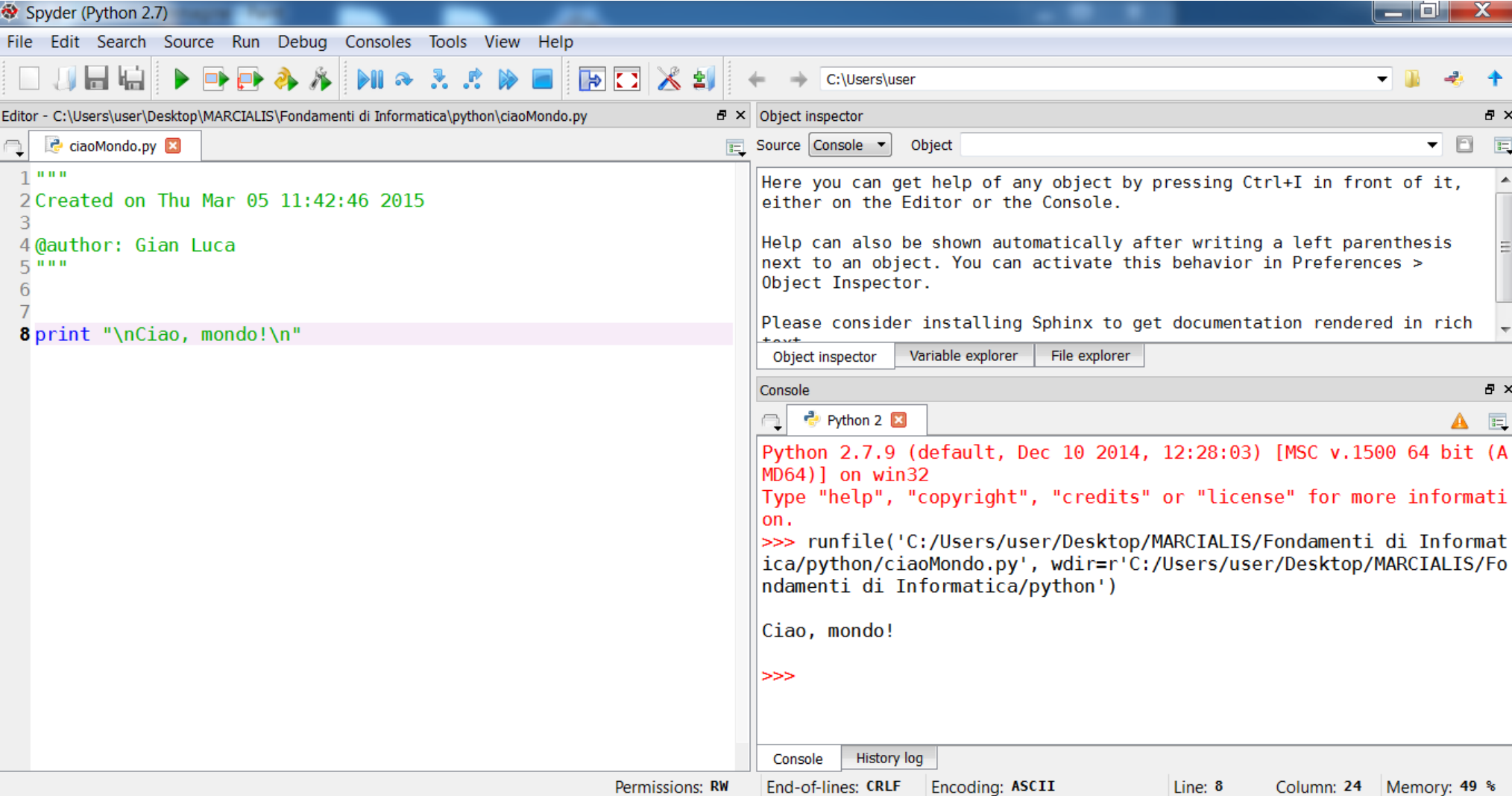
➤ Compilati

- C, C#, C++, Java, Pascal...
 - Programma → Traduzione → Istruzione → Esecuzione
- 

Il nucleo del linguaggio Python

- Come tutti i linguaggi, esso è dotato di:
 - Sintassi
 - Semantica
- La sintassi è l'insieme di regole per la costruzione corretta di una “frase”, ovvero di una istruzione
- La semantica è il significato che si dà alla “frase”
- Poiché il linguaggio Python deve essere comprensibile al calcolatore, non ci possono essere ambiguità semantiche
 - Una “frase” in Python ha sempre lo stesso significato

Un tipico ambiente Python



The image shows the Spyder Python IDE interface. The main window is titled "Spyder (Python 2.7)". The menu bar includes File, Edit, Search, Source, Run, Debug, Consoles, Tools, View, and Help. The toolbar contains various icons for file operations and execution. The editor window shows a file named "ciaoMondo.py" with the following code:

```
1 """
2 Created on Thu Mar 05 11:42:46 2015
3
4 @author: Gian Luca
5 """
6
7
8 print "\nCiao, mondo!\n"
```

The Object Inspector panel on the right shows help text for the selected object:

Here you can get help of any object by pressing Ctrl+I in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in Preferences > Object Inspector.

Please consider installing Sphinx to get documentation rendered in rich text.

The Console panel shows the output of the code execution:

```
Python 2.7.9 (default, Dec 10 2014, 12:28:03) [MSC v.1500 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> runfile('C:/Users/user/Desktop/MARCIALIS/Fondamenti di Informatica/python/ciaoMondo.py', wdir=r'C:/Users/user/Desktop/MARCIALIS/Fondamenti di Informatica/python')

Ciao, mondo!

>>>
```

The status bar at the bottom displays: Permissions: RW, End-of-lines: CRLF, Encoding: ASCII, Line: 8, Column: 24, Memory: 49 %.

Ambienti di programmazione

➤ Editor

- serve per scrivere il *programma sorgente*, cioè il **testo** che contiene le istruzioni nel linguaggio prescelto

➤ Compilatore

- traduce un programma sorgente in *programma oggetto*, cioè in un programma con un formato molto vicino a quello del linguaggio macchina. Se vi sono errori nella stesura viene avvisato il programmatore e il programma oggetto non viene generato.

➤ Interprete

- traduce ed esegue il programma istruzione per istruzione
- non usa il compilatore, ma un interprete che esegue direttamente il codice sorgente

Ambienti di Programmazione

➤ Linker

- collega insieme vari programmi oggetto che fanno parte di un unico programma suddiviso in *moduli* coordinati fra loro, generando il *programma eseguibile*.

➤ Debugger

- consente di eseguire il programma passo passo verificando l'esecuzione delle istruzioni del programma sorgente e individuando eventuali errori

➤ Gli ambienti di programmazione forniscono in genere anche un insieme di *funzioni di libreria*, cioè di algoritmi comuni a molti programmi (es. ordinamento di un vettore di numeri)

Installazione dell'ambiente «Spyder»

- Anche se un programma Python si può scrivere con carta e penna, meglio fare esperienza diretta sull'elaboratore
- Passaggi:
 - Andare nel sito <https://www.continuum.io/downloads>
 - Scaricare e installare «Anaconda» con **Python 3** per il relativo Sistema Operativo
 - Cliccare sull'icona sottostante per avviare l'ambiente Spyder o selezionarlo dal menu programmi :



Il mio primo programma Python

- Si scriva in linguaggio Python un programma che stampi a video il seguente messaggio:

```
Ciao, mondo!
```

Il mio primo programma in Python

```
#Il mio primo programma in Python
#Questo programma stampa a video
#la parola 'Ciao, mondo!'

print("\nCiao, mondo!\n")
```

Il mio primo programma in P.

```
#Il mio primo programma in Python  
#Questo programma stampa a video  
#la parola 'Ciao, mondo!'
```

```
print("\nCiao, mondo!\n")
```

Questo è un **commento**: non contiene istruzioni dell'algoritmo ma in genere serve per spiegare cosa fa una certa parte di programma (codice)

Un **commento** può anche essere inserito fra i simboli :

""" → inizio commento

""" → fine commento

Il mio primo programma in Python

```
#Il mio primo programma in Python
#Questo programma stampa a video
#la parola 'Ciao, mondo!'
```

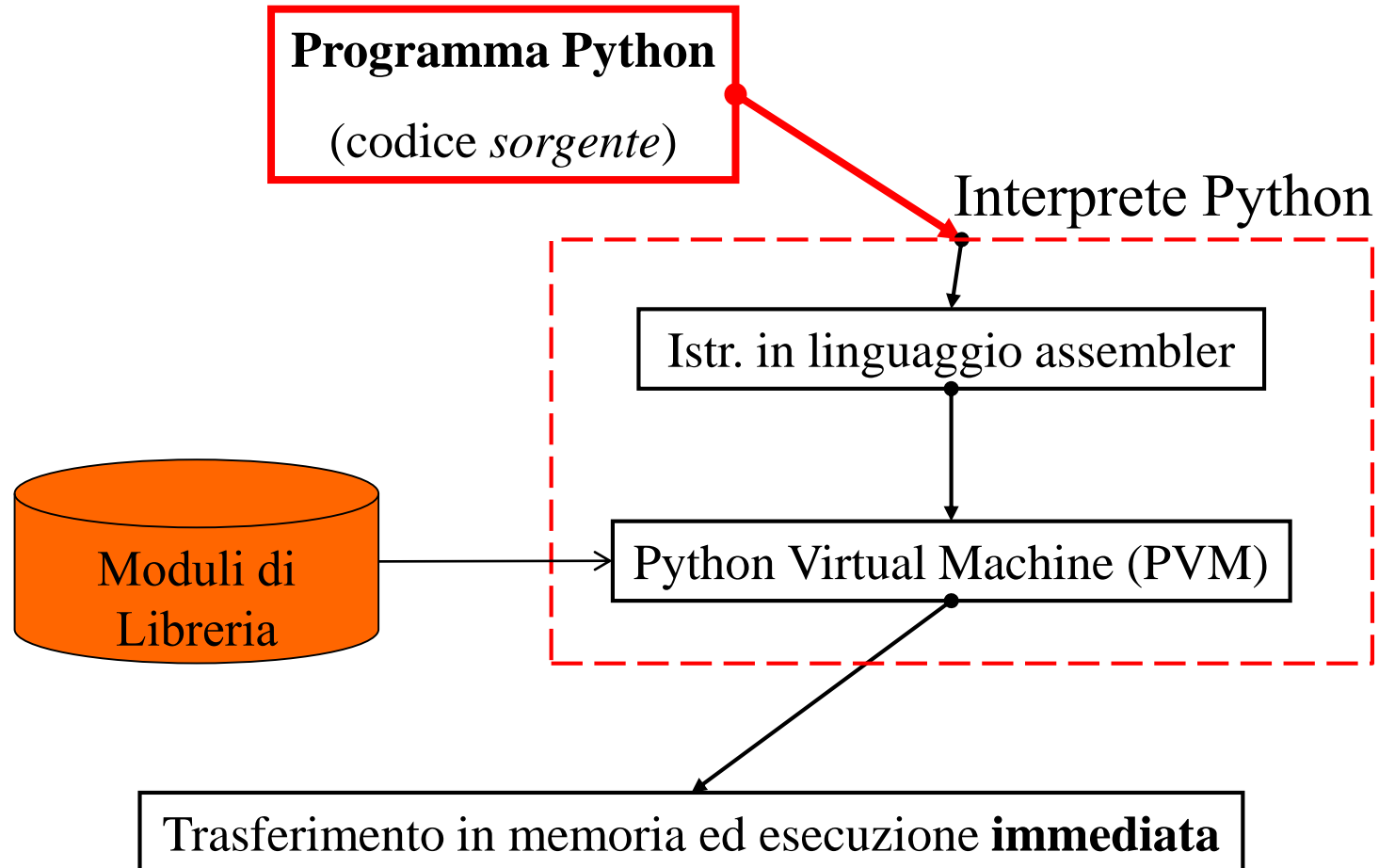
```
print("\nCiao, mondo!\n")
```

E' un'istruzione composta da una **funzione** presente nella libreria base di Python.

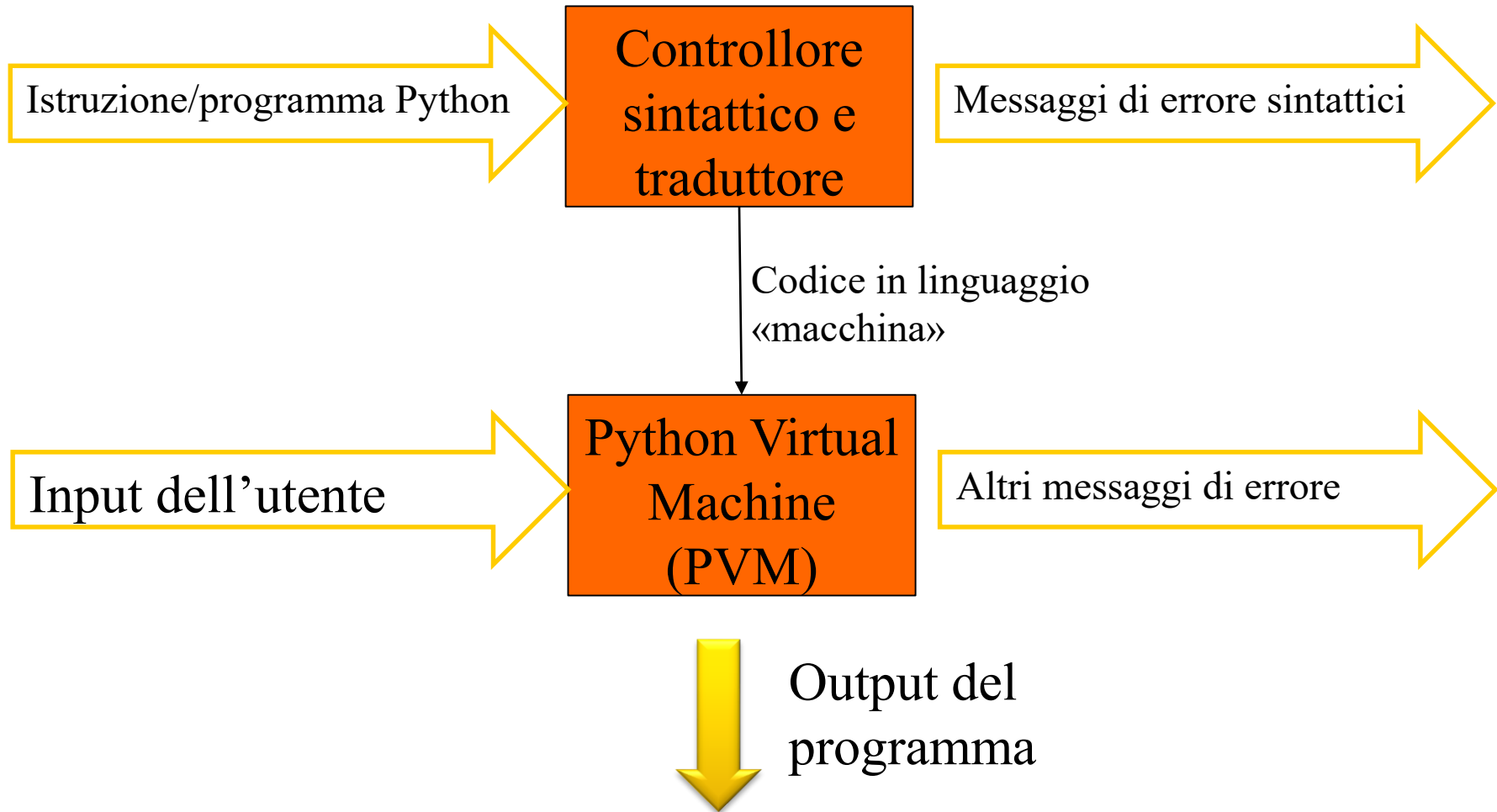
L'identificatore **print** è il nome della funzione (una sorta di codice operativo), e prevede che si stampi su SO la sequenza di caratteri fra virgolette "".

La sequenza **\n** significa «vai a capo».

Funzionamento dell'interprete Python



In altre parole...



Codice sorgente ed output

The image shows a screenshot of the Spyder Python IDE. The main editor window displays the source code of a Python script named `ciaoMondo.py`. The code is as follows:

```
1 """  
2 Created on Thu Mar 05 11:42:46 2015  
3  
4 @author: Gian Luca  
5 """  
6  
7  
8 print "\nCiao, mondo!\n"
```

A red oval highlights the source code in the editor. A red dashed box with a pointer contains the text "Codice sorgente".

The console window on the right shows the output of the program:

```
Python 2.7.9 (default, Dec 10 2014, 12:28:03) [MSC v.1500 64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license" for more information.  
>>> runfile('C:/Users/user/Desktop/MARCIALIS/Fondamenti di Informatica/python/ciaoMondo.py', wdir=r'C:/Users/user/Desktop/MARCIALIS/Fondamenti di Informatica/python')  
Ciao, mondo!  
>>>
```

A green oval highlights the output "Ciao, mondo!". A green dashed box with a pointer contains the text "Output del programma".

The bottom status bar shows: Permissions: RW, End-of-lines: CRLF, Encoding: ASCII, Line: 8, Column: 24, Memory: 49 %.

La funzione `print` • `print()`

- Solo la *sintassi* della versione 2.7 prevede l'uso o meno delle parentesi tonde a seconda della versione di Python
 - `print(sequenza)` #versione 2.7 e successive
 - `print sequenza` #solo versione 2.7
- Esempi con e senza parentesi
 - `print(3+4)`
 - `print("La risposta è: ", 6*7)`
 - `print 3+4`
 - `print "La risposta è: ", 6*7`

L'istruzione di assegnamento

➤ Sia dato il seguente problema:

Scrivere un programma in Python che riceva da tastiera due valori interi e stampi su video la loro somma

Soluzione

```
"""  
Programma per la stampa a video della somma di due  
numeri  
"""  
print("\\nInserire due valori da sommare\\n")  
a=input("Inserire il primo valore: ")  
b=input("Inserire il secondo valore: ")  
  
somma = a + b  
  
print (a, " + ", b, " = ", somma)
```

La soluzione presentata

- Il programma presenta in generale le stesse caratteristiche del precedente
- Cambia ovviamente la sequenza di istruzioni...
 - In particolare, viene chiamata la funzione `input` che serve a leggere dei valori immessi da tastiera
 - **I valori letti da tastiera vengono assegnati alle variabili `a`, `b`**
 - Infine la somma di `a` con `b` viene **assegnata** ad un'altra variabile `somma`...
 - ... che viene stampata a video tramite la funzione `print` che separa interi e parole attraverso l'uso della virgola `\, '`

Variabili in Python

- Cos'è una **variabile**?
 - E' un contenitore di valori (numeri o caratteri o dati complessi)
 - Vedremo studiando le architetture che a ciascuna variabile è associato un opportuno «spazio» nella memoria fisica del calcolatore
- Una variabile viene sempre creata nell'istante in cui serve al programmatore e può contenere, come vedremo, qualsiasi **tipo di dato**
- Ad essa può quindi essere assegnato un valore con apposita istruzione
 - `somma = a + b`

Assegnamento ed operazioni aritmetiche e booleane elementari

- L'istruzione di assegnamento prevede la seguente sintassi:
 - `Variabile Destinazione = F(V1, V2, ..., VN);`
 - Le variabili V1, ..., VN sono elaborate attraverso operatori:
 - Somma, sottrazione, prodotto, divisione: `+`, `-`, `*`, `/`
 - Uguaglianza e disuguaglianza: `==`, `!=`
 - Se i valori interi vengono espressi come valori booleani (0 per False, 1 per True):
 - `and`, `or`, `not`
 - Esempi:
 - `area_quadrato = lato*lato;`
 - `segue_informatica = (chimico or meccanico) and matricola`
 - `non_iscritto_chimica = not chimico #se chimico=True`

Esercizio

- Siano date due variabili a , b , da considerarsi intere
- Si supponga $a=15$, $b=-20$
- Dire il valore della seguente variabile c in funzione delle espressioni date:
 - $c = a - b$
 - $c = a \neq b$
 - $c = \text{not } a$
 - $c = (a == b) \text{ or } ((a + b) == -5)$
 - $c = (a \text{ or } b) + b$
 - $c = a \text{ and } b$

Struttura generale di un programma Python

- Commenti
- Importazione di funzioni (anche dichiarazione di costanti)
- **Modulo principale**
 - ALGORITMO VERO E PROPRIO

Esercizi

- Scrivere un programma Python che riceva da tastiera un valore (intero) del lato di un quadrato e ne stampi a video l'area ed il perimetro
- Scrivere un programma Python che riceva da tastiera tre bit indicanti i valori di verità di `iscritto_in_chimica`, `iscritto_in_meccanica`, `primo_anno`, e stampi a video il valore di verità `segue_informatica`

Per saperne di più

- Lambert, *Programmazione in Python*, Cap.1, Apogeo