

Laurea Triennale in Ingegneria Gestionale

Corso di Fondamenti di Informatica A.A. 2015/2016

DIPARTIMENTO DI INGEGNERIA INFORMATICA
AUTOMATICA E GESTIONALE ANTONIO RUBERTI



SAPIENZA
UNIVERSITÀ DI ROMA

Introduzione a 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



- Supporta **diversi paradigmi di programmazione**:
 - Orientato agli oggetti.
 - Imperativo.
 - Funzionale.
 - Dichiarativo.
- Offre un controllo dei tipi
 - Forte (strong typing).
 - Eseguito al run-time (dynamic typing).
- 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.
 - 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.
 - Portabilità.
 - Indipendenza dalla piattaforma.



Ambiente di sviluppo Python

- Integrato
 - IDLE (<https://docs.python.org/3.4/library/idle.html>)
- Editor di testo
- Finestra di terminale
- Modalità
 - Scripting.
 - Interactive.



Ambiente di sviluppo Python

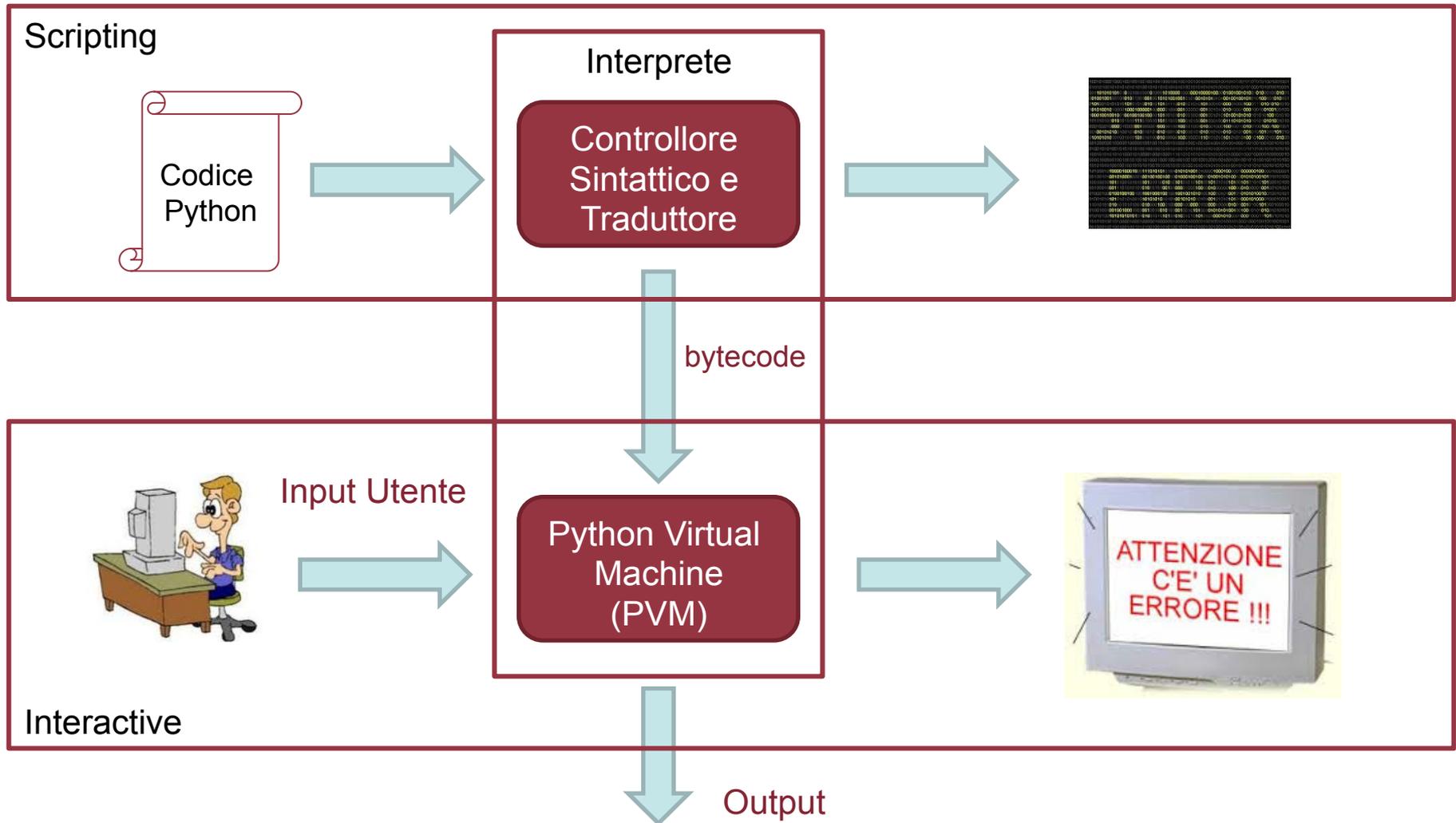
```
Python 3.3.2 Shell
File Edit Shell Debug Options Windows Help
>>>
>>> #questa è la shell di Python
>>> #ogni istruzione viene eseguita appena premete invio
>>> #fate caso ai colori
>>> print(3+5)
8
>>> |
```

```
Python 3.4.1: Esempio editor.py - C:\Dropbox\Doc_dida\CorsoPython\Lezioni_2014_2015\Prog...
File Edit Format Run Options Windows Help
#questo è l'editor di file di Python
#tutte le istruzioni vengono eseguite insieme
#tutte le righe che cominciano con # vengono ignorate
#NON valuta le espressioni --> per vedere il risultato
#fate caso ai colori

print("Ciao")
print('Ciao')
print(5+3)
```

```
Python 3.4.1 Shell
File Edit Shell Debug Options Windows Help
Python 3.4.1 (v3.4.1:c0e311e010fc, May 18 2014, 10:
45:13) [MSC v.1600 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more
information.
>>> ===== RESTART =====
>>>
Ciao
Ciao
8
>>> |
```

Evoluzione del Programma Python



Funzioni di base

Funzione	Descrizione
<code>print(<esp₁>, ... , <esp_n>)</code>	Valuta le espressioni e le visualizza nella shell separandole con uno spazio
<code>input(<stringa per il prompt>)</code>	Visualizza la stringa passata come parametro sul prompt e attende l'immissione del valore. Restituisce la stringa dei caratteri immessi dall'utente
<code>int(<stringa rappresentante in intero>)</code>	Trasforma una stringa di cifre in un numero intero