



di.unito.it

DIPARTIMENTO DI INFORMATICA
Università degli Studi di Torino

Introduzione ed installazione di Python

Zhongli Filippo Hu

Dipartimento di Informatica

Università degli Studi di Torino

Python

- Linguaggio di programmazione orientato agli oggetti
- Si usa un editor di testo per scrivere i file sorgenti dei programmi
- Si usa un interprete di python per eseguirli
- Esistono due versioni dell'interprete di Python: Python2 e Python3.
- Esistono alcuni strumenti per semplificare la scrittura del codice:
 - PyCharm: un IDE che implementa un debugger, un software che permette di vedere in tempo reale i valori delle variabili e seguire il codice del programma
 - Jupyter Notebook: un ambiente interattivo che consente di scrivere ed eseguire codice Python da browser e vedere nella pagina gli output.

Strumenti utilizzati nel corso

- Nel corso utilizzeremo la combinazione editor di testo + interprete di Python3 e alcune volte il debugger di PyCharm.
- Esistono molti editor di testo come Blocco Note, Notepad++, Atom, Visual Studio Code o Sublime Text.

Installazione

- Su Windows
 - Andare su python.org
 - Scaricare il setup e seguire i passi indicati
- Su MacOS
 - Andare su python.org
 - Scaricare il setup e seguire i passi indicati

Installazione

- Su Linux
- Controllare se Python3 è installato scrivendo sul terminale
- `# python3 --version`
- Se non è installato scrivere
- `# sudo apt install python3.7`

Eseguire un programma

- Aprire un terminale nella cartella dove c'è il sorgente del programma
- Su Windows si può aprire un terminale nella cartella cliccando tasto destro + MAIUSC nella cartella e poi cliccando su Apri finestra PowerShell qui
- Su MacOS e Linux basta cliccare nella cartella con il tasto destro e poi Apri terminale qui
- Scrivere sul terminale il seguente comando:
- `# python3 «nomedelfilesorgente.py»`

Stampare un output

- Per stampare in output esiste l'istruzione `print()`.
- Viene stampato ciò che si scrive nelle parentesi.
- È un esempio di funzione

Commenti

- Il commento è un'annotazione che aiuta il programmatore a leggere il programma
- L'interprete ignora i commenti
- Ogni commento è preceduto da un #

Indentazione

- In Python non ci sono parentesi graffe che racchiudono le istruzioni
- L'indentazione ha un significato e raggruppa le istruzioni. Spiegheremo nelle prossime lezioni quando usarla.



di.unito.it

DIPARTIMENTO DI INFORMATICA
Università degli Studi di Torino

Variabili e operazioni di base sui numeri e sulle stringhe

Zhongli Filippo Hu

Dipartimento di Informatica

Università degli Studi di Torino

Valori

- I valori sono degli elementi base che un programma può elaborare. Possono essere per esempio un numero o una sequenza di caratteri
- Ogni valore ha un tipo. Alcuni esempi di tipi sono:
 - Numero intero (int): per esempio 5
 - Numero con la virgola (float): per esempio 3.0 o 4.5
 - Stringa cioè una sequenza di caratteri (str): per esempio 'ciao Mondo!'
- Con l'istruzione `type()` viene restituito il nome del tipo:
 - `type(5): int`
 - `type('5'): str`
 - Ciò che è tra apici è una stringa!

Valori e funzioni

- I valori possono essere utilizzati nelle espressioni o essere passati come argomenti alle funzioni.
- Le funzioni raggruppano una sequenza di istruzioni
- Vedremo nei prossimi video come definire delle funzioni, per ora ci limiteremo ad utilizzarne alcune.
- Una funzione si utilizza scrivendo il nome della funzione e mettendo tra le parentesi gli argomenti che si vogliono passare.
- Per esempio `print()` è una funzione e tra parentesi andiamo a inserire ciò che vogliamo stampare.
- Anche `type()` è una funzione.
- Per passare più argomenti, se la funzione lo consente, separare i valori da una virgola.

Assegnamento di una variabile

- L'istruzione di assegnazione serve per creare una nuova variabile e per assegnare un valore.
- Le variabili in Python hanno il tipo implicito, cioè durante la dichiarazione non dobbiamo specificare che tipo hanno.
- La sintassi è:
 - `[nomevariabile] = [valore]`
- Il valore può essere un numero, una stringa oppure il nome di una variabile.
- Il nome di una variabile non può iniziare con un numero

Operazioni matematiche

- Si segue l'ordine matematico: prima le parentesi, poi l'elevazione a potenza, poi le moltiplicazioni e divisioni ed infine le addizioni e sottrazioni.
- Si elenca la sintassi delle operazioni più comuni:
 - Somma: $[\text{num}] + [\text{num}]$
 - Sottrazione: $[\text{num}] - [\text{num}]$
 - Moltiplicazione: $[\text{num}] * [\text{num}]$
 - Divisione: $[\text{num}] / [\text{num}]$
 - Elevamento a potenza: $[\text{num}] ** [\text{num}]$
 - Radice quadrata: $\text{sqrt}([\text{num}])$

Operazioni sulle stringhe

- Per concatenare due stringhe, cioè aggiungere a una certa stringa una seconda stringa, si utilizza l'operatore +
- La sintassi è seguente: `[str] + [str]`



di.unito.it

DIPARTIMENTO DI INFORMATICA
Università degli Studi di Torino

Operazioni booleane ed if then else

Zhongli Filippo Hu

Dipartimento di Informatica

Università degli Studi di Torino

Valori booleani

- Oltre ai numeri e alle stringhe esistono i valori booleani:
 - True
 - False
- Una condizione è un'espressione che valutata ha un valore booleano
- Come nelle espressioni matematiche abbiamo delle operazioni matematiche e valutandole abbiamo un numero, nelle condizioni abbiamo delle operazioni booleane e valutandole abbiamo Vero o Falso

Valori booleani nelle variabili

- Possiamo mettere un valore booleano in una variabile
- Possiamo mettere la valutazione di un'espressione booleana in una variabile

Esempio:

- `piove = True`
- `piove = stradaBagnata and nuvoloso` # `stradaBagnata` e `nuvoloso` sono variabili booleane

Operatori booleani

- Abbiamo tre operatori booleani:
 - not
 - and
 - or

Comparazione

- Abbiamo:

- ==
- !=
- <
- >
- <=
- >=

Operatori booleani

x	not x
True	False
False	True

Not

x	not x
True	False
False	True

- Esempio:
 - Se abbiamo una variabile chiamata piove che contiene il valore di falso, seguendo questa tabella not piove è vero

piove falso  not piove vale vero

And

x	y	x and y
True	True	True
True	False	False
False	True	False
False	False	False

- Esempio:
 - Se abbiamo una variabile chiamata stradaBagnata che contiene il valore di falso e una variabile nuvoloso che contiene vero. Seguendo questa tabella stradaBagnata and nuvoloso è falso



Or

x	y	x and y
True	True	True
True	False	True
False	True	True
False	False	False

- Esempio:
 - Se abbiamo una variabile chiamata stradaBagnata che contiene il valore di falso e una variabile nuvoloso che contiene vero. Seguendo questa tabella stradaBagnata or nuvoloso è vero



Operazioni in python

- Le operazioni in python sono scritte in questo modo:
 - not [Booleano]
 - [Booleano] and [Booleano]
 - [Booleano] or [Booleano]

Operazioni in python

- Le condizioni possono essere:
 - [Booleano]
 - not [Booleano]
 - [Booleano] and [Booleano]
 - [Booleano] or [Booleano]

Dove [Booleano] può essere:

- un valore booleano
- una variabile booleana
- una condizione

if

- La selezione di azioni (costrutto in cui certe istruzioni vengono eseguite se una certa condizione è vera) ha la seguente sintassi:

if [condizione]:

 istruzione1

 istruzione2

- Come abbiamo detto la prima lezione l'indentazione raggruppa una sequenza di istruzioni.
- Tutte le istruzioni che devono essere eseguite se la condizione è vera hanno una tabulazione in più rispetto all'if.
- Quando scriveremo un'istruzione con le stesse tabulazioni che contiene la riga dell'if avremo terminato le istruzioni contenute nell'if.

If - else

- Per inserire una serie di azioni nel caso la condizione sia falsa bisogna usare la selezione d'azioni con l'else. La sintassi è la seguente:

if [condizione]:

 [istruzione1]

 [istruzione2]...

else:

 [istruzione3]

 [istruzione4]

If - elif

- Per inserire degli else if (cioè istruzioni che vengono eseguite se la prima condizione nell'if è falsa e la seconda condizione è vera) bisogna utilizzare la seguente sintassi:

if [condizione]:

 [istruzione1]

 [istruzione2]...

elif [condizione]:

 [istruzione3]

 [istruzione4]



di.unito.it

DIPARTIMENTO DI INFORMATICA
Università degli Studi di Torino

Liste, cicli while e for each

Zhongli Filippo Hu

Dipartimento di Informatica

Università degli Studi di Torino

Liste

- Oltre ai tipi di dati numerici (int e float), stringhe e boolean esiste anche il tipo lista.
- La lista è un tipo di dato che memorizza una collezione ordinata di elementi.
- Puoi anche avere liste di liste!

Sintassi per creare una lista:

- `nome_lista = ['elem1', 'elem2', 'elem3']` # lista con 3 stringhe
- `nome_lista = []` # lista vuota

Liste

- Sintassi per usare il valore dell'i-esimo elemento:
 - `nome_lista[i]`
- Ogni elemento della lista è memorizzata in una posizione.
- Si parte a contare da 0, cioè la prima posizione è 0.
- La lista `x = ['elem1', 'elem2', 'elem3']` contiene 3 elementi.
- `x[0]` contiene 'elem1'
- `x[1]` contiene 'elem2'
- Se vogliamo modificare il valore di una posizione possiamo semplicemente usare l'operatore di assegnamento (es: `x[2] = 'ciao'`)

Operazioni sulle liste

- Append: aggiunge un elemento alla lista alla fine
- Remove: tolgo un elemento dalla lista (il primo se ci sono multiple occorrenze)
- Insert: inserisce un elemento in una determinata posizione
- pop: tolgo un elemento dalla list (anche con l'indice) e lo ritorno
- In: operazione per vedere se un elemento è nella lista

Iterazione

- In python abbiamo due costrutti per iterare (ripetere) le azioni:
 - While
 - For each
- Il costrutto while ripete le azioni finché una certa condizione è vera (smette quando diventa falsa)
- In costrutto for each ripete una certa azione per ogni elemento di una lista

while

- La sintassi del while è la seguente

while [condizione]:

istruzione1

istruzione2

- Tutte le istruzioni che devono essere ripetute se la condizione è vera hanno una tabulazione in più rispetto al while.
- Quando scriveremo un'istruzione con le stesse tabulazioni che contiene la riga del while avremo terminato le istruzioni contenute nel while.