

# Laboratorio di Informatica

## Introduzione a Python

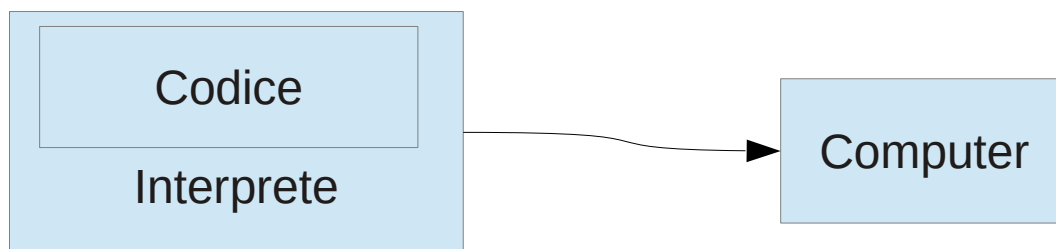
Dottore Paolo Parisen Toldin - [parisent@cs.unibo.it](mailto:parisent@cs.unibo.it)

# Argomenti trattati

- Che cosa è python
- Variabili
- Assegnazione
- Condizionale
- Iterazione in una lista di dati

# Cosa è python

- Linguaggio di programmazione
  - Attraverso delle **parole (lessico)** e **regole (sintassi)** diciamo **cosa fare** al computer (**semantica**).
- Python è un linguaggio interpretato
  - Il codice che scriveremo non verrà trasformato in 0/1 (linguaggio macchina) ma viene “letto” da un programma esterno chiamato “interprete python”



# Python console / script python

- Potete far eseguire i vostri comandi python attraverso la console “python console”
- Oppure scrivendo un file .py ed eseguendolo.
- La seconda scelta è utile se si vogliono salvare i programmi che si fanno in modo da poterli richiamare successivamente.

# Stampare a schermo un valore

- Una dei principali comandi di python è “print”
  - “print” permette di stampare a video quello che segue.

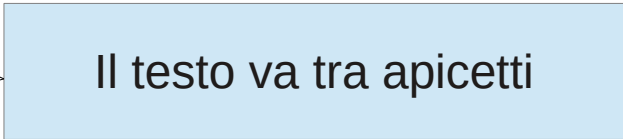
– >>> print “ciao”

ciao

– >>> print 24

24

– >>> print ciao



Il testo va tra apicetti

**NameError: name 'ciao' is not defined**

# Sintassi

- Variabili: nome inventato che descrive una locazione di memoria ove sarà possibile salvare valori.

...	...
X	54
...	...

- $X = 54$

- Parole riservate: “=”, “while”, “def” “if”, “else”, “elif”, “for”, ...
  - Nell'editor che useremo, le parole riservate vi verranno colorate in viola.
  - Non si possono usare parole riservate come nomi di variabili

# Assegnamento

- Ogni qual volta eseguiamo un assegnamento, scriviamo nella relativa cella di memoria, il valore associato.

Sintassi: `<nome_variabile> = <valore>`

- `>>> mia_variabile=34`

`>>> variabile2 = "hello"`

`>>> print mia_variabile`

34

`>>> mia_variabile = 54`

`>>> print mia_variabile`

54

`>>> mia_variabile = (mia_variabile + 54) / 11`

`>>> print mia_variabile`

8

Nota: la divisione "/" restituisce la divisione esatta se uno degli operandi è un numero reale, altrimenti effettua la divisione tra interi.  
 $5/3 = 1$  mentre  $5.0/3=1.666666$ .  
Il resto della divisione si ottiene con l'operatore "%"

# Operatori aritmetici

- “\*”: l'asterisco definisce la moltiplicazione
- “+”: simbolo per la somma
- “-”: simbolo per la sottrazione
- “%”: resto della divisione intera
- “/”: simbolo per la divisione. La divisione è intera se entrambi gli operandi sono interi, altrimenti effettua la divisione tra reali.
- `abs(espressione)`: restituisce il valore assoluto dell'espressione tra parentesi tonde.
- `a**b`: effettua l'elevamento a potenza di `a` per l'esponente `b`.
- `math.sqrt(espressione)`: effettua l'estrazione di radice quadrata del risultato dell'espressione tra parentesi

Nota bene: per usare `math.sqrt()`  
devi scrivere all'inizio del file  
**`import math`**



# Stampare più valori

```
>>> x = 34
```

```
>>> y = 12
```

```
>>> miavar = "stringa"
```

```
>>> miavar2 = "seconda stringa"
```

```
>>> print x,y
```

```
34 12
```

```
>>> print miavar,miavar2
```

```
stringa secondastringa
```

```
>>> print x+y
```

```
46
```

```
>>> print miavar+miavar2
```

```
stringasecondastringa
```

La virgola serve a "creare" una coppia di valori. nota che lo spazio tra i due valori stampati serve solo per discretizzare i due (o più) valori

Notare l'utilizzo del "+". con in numeri somma, con le stringhe, concatena

# Esercizio

- Definire delle variabili assegnandoli il valore corretto.
  - NOME, COGNOME, ANNI
- Stampare i valori uno di seguito all'altro.
- Definiti X e Y, calcolare la seguente espressione:

$$\sqrt{|(-X)^4| + Y(X/3)}$$

Effettuate la  
divisione  
vera e non quella  
tra numeri interi

- Eseguire gli esercizi sia in modalità scripting che da console python.

Se X=11 e Y=2, il  
Risultato è 121.052464968  
Funziona?

# Esercizio

- Scrivere un programma che, definite due variabili a e b, inverte i valori delle due variabili. Quindi:

a=

b=

print a,b



<vostro codice>

print a,b



Le due istruzioni devono restituire i valori invertiti.

Se a=0 e b=1, la prima volta deve stampare "0,1"  
La seconda volta, deve stampare "1,0"

# Espressioni booleane

- Una espressione booleana è una espressione che da come risultato o “vero” o “falso”.
- Esse si formano con i costrutti “<”, “>”, “=”, “<=”, “>=” (quando si confrontano espressioni matematiche)
- Oppure si formano utilizzando gli operatori “and”, “or”, “not” (quando si mettono assieme più espressioni booleane tra loro)

Esempio:  $(X=32 \text{ and } Y>0) \text{ or } X \geq Y$

# Condizionale

- Talvolta si vogliono eseguire delle operazioni dipendentemente dallo stato del sistema (valori di certe variabili)
- A questo serve il costrutto If-else. Due possibilità:

- if <condizione>:

- istruzione

- if <condizione>:

- istruzione

- else:

- istruzione

Il codice dentro il ramo IF  
e dentro il ramo ELSE  
VA INDENTATO.  
L'indentazione si fa con il  
tasto tab



# Condizionale

```
mia_variabile = 32
```

```
if mia_variabile == 32:
```

```
    print "la mia variabile vale 32"
```

```
else:
```

```
    print "il valore della variabile non è 32"
```

- Provate ad eseguire il codice. Cosa otteniamo?
- Viene eseguito il primo ramo del costrutto IF. Cosa succede se assegnamo a `mia_variabile` il valore 18?

# Esercizio

- Prendete l'esercizio precedente e aggiungete una espressione condizionale tale che controlli la vostra età.
  - Se la vostra età è maggiore di 22, allora stamperà una frase a vostra scelta
  - Altrimenti, vi deve stampare a video il numero di anni che vi servono per arrivare ad avere 22 anni.
- Testate l'esercizio con differenti età in modo da controllare che il risultato ottenuto sia corretto.

# Esercizio

- Definire due variabili intere X ed Y

Scrivere un programma che dopo aver controllato quale delle due possiede il valore più grande, lo stampa.

- Definire tre variabili. Scrivere un programma che scambia il valore più piccolo con il valore in a.

a=

b=

c=

print a,b,c

<vostro codice>

print a,b,c

Se a=11 e b=1 e c=4, la prima volta  
deve stampare "11,1,4"  
La seconda volta,  
deve stampare "1,11,4"



# Esercizio

- Scrivere il programma in python che risolva l'equazione  $ax^2 + bx + c = 0$ 
  - Ovviamente l'esercizio deve dare la soluzione corretta per qualunque a,b,c.

# Liste di valori

- Possiamo definire liste di valori utilizzando le parentesi quadrate “[ “ ]”

```
>>> a = ["a","b","c","d","e"]
```

```
>>> print a
```

```
[ "a" , "b" , "c" , "d" , "e" ]
```

# Liste di valori

- Possiamo accedere ai valori all'interno della lista tramite l'operatore “[numero]”

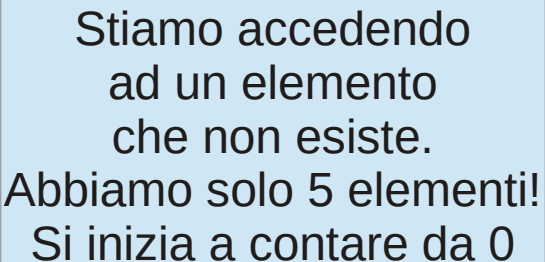
```
>> a[0]
```

```
“a”
```

```
>> a[1]
```

```
“b”
```

```
>> a[5]
```



Stiamo accedendo  
ad un elemento  
che non esiste.  
Abbiamo solo 5 elementi!  
Si inizia a contare da 0

**IndexError: list index out of range**

# Liste di valori

- Le liste di valori funzionano pressapoco come un contenitore di tante celle di memoria, variabili. Possiamo accedervi tramite il costrutto <lista>[<numero>]

```
>>> a=[1,2,3,4]
```

```
>>> print a
```

```
[1,2,3,4]
```

```
>>> a[1] = 34
```

```
>>> print a
```

```
[1,34,3,4]
```

# Generazione di liste di interi

- Tramite il comando `range(<index>,<index>)` possiamo generare una lista di valori interi.

```
>>>range(0,10)
```

```
[0,1,2,3,4,5,6,7,8,9]
```

```
>>> x = 11
```

```
>>> range(4,x)
```

```
[4,5,6,7,8,9,10]
```

```
>>> y=5
```

```
>>> range(y,x)
```

```
[5,6,7,8,9,10]
```

Nota come il **primo indice** indichi il **primo valore della lista generata** mentre il **secondo indice non rientra tra i valori della lista**

# Esercizio

- Scrivere una lista di valori e provare ad accedervi controllando che il valore ritornato sia quello voluto.
- Generare una lista di valori ed ri-eseguire il primo esercizio.

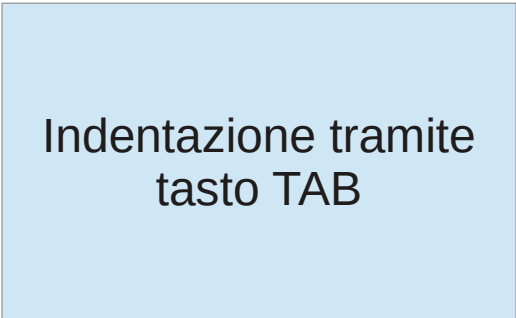
# Iterazione

- Come si può accedere a tutti gli elementi di una determinata lista in modo automatico?
- Si usa il costrutto for <variabile> in <lista>:

```
a=["a","b","c","d"]
```

```
for mio_elemento in a:
```

```
    print mio_elemento
```



Indentazione tramite  
tasto TAB

- Il programma restituirà in sequenza tutti i valori della lista.

# Esercizio

- Definita una lista di valori interi, scrivere un programma che trova il valore massimo all'interno della lista e infine stampa il valore.
- Data una lista **a=[1413,15,4,1,5,76,43,14,312,4,52,-2,6,-1,-31,54,-6,65,-987]**, scrivere un programma che, analizzando tutta la lista, restituisca una lista ordinata in modo crescente. Quindi, **a[0]** deve contenere il valore più piccolo.

Suggerimento: confrontare ogni elemento con tutti gli altri e spostare in testa i valori più piccoli.



# Esercizio

- Scrivere il programma in python  $\text{add} : (\mathbf{N} \times \mathbf{N}) \rightarrow \mathbf{N}$  definita come  $\text{add}(x, y) = x + y$  senza utilizzare direttamente la somma tra i due operandi ma utilizzando solamente la funzione successore  $\mathbf{S}(x) = x + 1$

– Alcuni esempi:

$$5 + 3 = \mathbf{S}(\mathbf{S}(\mathbf{S}(5)))$$

$$7 + 5 = \mathbf{S}(\mathbf{S}(\mathbf{S}(\mathbf{S}(\mathbf{S}(7)))))$$

La funzione  $\mathbf{S}$  non è definita in python; l'esercizio richiede dunque di eseguire una somma iterata "+1" un numero di volte pari al secondo operando.