Lezioni di Python Day 1

Dr. Giuliano Taffoni – Dr. Giuseppe Murante

Cosa è Python

• Linguaggio di programmazione e scripting.

"Object Oriented"

 Procedurale, funzionale > possiamo scegliere il nostro stile di programmazione

- Versioni: 2.x and 3.x
 - 3.x is **not** backwards compatible with 2.x

Caratteristiche principali

Linguaggio interpretato (o semi interpretato)

- Tipizzato e dinamico
- Gestione automatica della memoria

- Ampio set di librerie standard e prodotte da terzi (moduli)
 - E.g. numpy, matplotlib, astropy etc

Caratteristiche principali

• Sintassi semplice e struttura molto leggibile

Portabile (Unix/Linux, Mac OS X, Windows, Android)

Interattivo

Open source

Tipi di Linguaggi di programmazione

 Un programma viene sviluppato scrivendo il codice sorgente in un opportuno linguaggio definito dalla sintassi del linguaggio stesso.

 Linguaggio compilato: codice sorgente viene trasformato (e ottimizzato) dal compilatore in linguaggio macchina che può essere eseguito dal processore;

```
void store(double *a, double *b, double *c)
{
    *c = *a + *b;
}

.text
.p2align 4,,15
.globl store
.type store, @function
store:
    movsd (%rdi), %xmm0 # Load *a to
%xmm0
```

Linguaggi interpretati

• Linguaggio Interpretato: il codice sorgente viene interpretato al volo e vengono eseguite le istruzioni così come descritte nel codice sorgente.

• Linguaggio semi interpretato: il codice sorgente viene compilato in un formato intermedio (chiamato bytecode), il quale a sua volta viene interpretato dall'interprete python, che ha il compito di interpretare "al volo" le istruzioni bytecode in istruzioni per il processore.

Le variabili

- "la programmazione si si basa sui dati (variabili) e le istruzioni (codice e funzioni).[...] Le istruzioni dicono al computer cosa fare con le variabili"
- "We store values in variables. Each variable is identified by a variable name. [...] Each variable has a variable type"
- "a variable must be defined in a declaration statement"
 - Serve a definire il nome e il tipo
- Variables are given a value through the use of an assignment statement.

```
La_mia_variabile = il_suo_valore
```

Le variabili in Python

 Tutti gli elementi in python sono oggetti: variabili, funzioni, classi etc.

• Il linguaggio è tipizzato: oggetti tipizzati, identificatori non tipizzati.

 Il linguaggio e' dinamico: gli oggetti non sono dichiarati all'inizio ma solo quando vengono usati (assignement statement)

Python interattivo

ipython

Commenti e documentazione

```
Commento su piu' righe
Utile per documentare un programma
```

commento su riga singola

Alcuni esempi

$$a=3$$

$$p=[1, 2, 3, 4]$$

- Crea l'oggetto numero intero di valore 3 e gli assegna il nome a
- Crea l'oggetto lista di quattro elementi e gli assegna il nome p
- Assegna alla lista di nome p un secondo nome k. In altri termini, crea un nuovo riferimento all'oggetto
- Crea l'oggetto stringa e gli assegna il nome a. L'oggetto 3 di cui sopra viene distrutto

Tipi numerici

```
int, long, float, complex
```

- a = 3
- b=3.1415926
- c=15000L
- d=a+b
- e=7/2
- f=float(333)
- g=complex(1.5,2)
- h=None
- i=True
- print f * g
- print e

Le operazioni

- $\chi + y$
- x − y
- x * y
- x / y
- x // y
- long(x)
- float(x)

- complex(re,im)
- c.conjugate()
- divmod(x, y)
- pow(x, y) x ** y
- abs(x)
- int(x)

Operazioni sui bit

```
X
             >>> x=3
             >>> y=10
X^{\wedge} y
             >>> print bin(x), bin(y)
x & y
             >>> print bin(x|y)
x << n
             >>> print bin(x^y)
x >> n
             >>> print bin(x&y)
 ~X
             >>> print bin(x<<3)</pre>
             >>> print bin(x>>4)
             >>> print bin(~x)
```

Stringhe

Sono oggetti immutabili che definiscono un insieme di caratteri.

A="Corso di Laboratorio di Astronomia" B='Corso di Cosmologia Teorica"

Le stringhe si possono concatenare

C = A + B

• Posso accedere ai singoli caratteri o a una "sotto-stringa" (slice), ma non posso modificarli

A[6] A[3:7] A[:5] A[-1]

Linguaggio è tipizzato: verifichiamolo

• Il linguaggio è tipizzato: distingue interi da array da stringhe

A="Corso di Astronomia"

$$C = A + 7$$

TypeError: cannot concatenate 'str' and 'int' objects

 $type(A) \rightarrow str$

Gli oggetti di tipo sequenza

- Stringhe "···", Liste [···] e Tuple (···)
- Liste e tuple sono collezioni non omogenee di oggetti (int, str, char, oggetti etc)
- Liste sono modificabili tuple sono immutabili

```
a='corso di Astronomia'
b=['a',3,5,3.14]
c=(3,'x',0,'I')
```

```
3 in b
'h' not in a
b + b
b + c
a * 3, 4 * c
a[4]
del(b[3])
len(a) min(b) max(b)
```

Operazioni su liste (ma non su tuple!)

lista.reverse()

lista.sort([cmp[,key[,reverse]]])

map(fun,lista)

Argomenti opzionali

• filter(fun, lista)

fun == Funzione generica

reduce(fun, lista)

Esempio

```
In [80]: import operator as op
In [81]: op.mul?
Docstring: mul(a, b) -- Same as a * b.
Type: builtin_function_or_method
In [82]:
```

Come sommo tutti gli elementi di una lista? Come moltiplico tutti gli elementi di una lista?

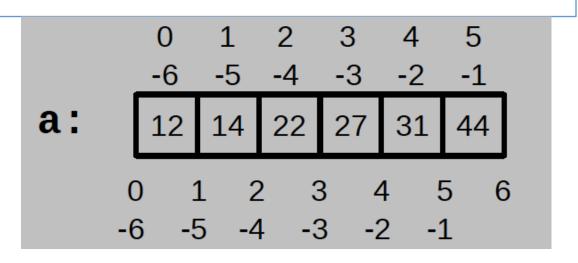
Liste: facciamole a fette

```
A[0:3] (A[:3])
```

B=range(1,10)

B[2:8:2]

B[-3:]



Copie di oggetti e referenze

- Quando faccio una assegnazione a=b non effettuo una copia ma solo duplico la referenza (puntatore)
- Esistono copie `shallow` e copie `deep`.

```
>>> M = range(10)
>>> C = M
>>> C[5]=56
>>> print M
>>> B = M[:]
>>> id(B)
```

- copy.copy() shallow copy
- copy.deepcopy() deepcopy

List comprehension

 Approccio "matematico" alla costruzione di liste (evoluzione della funzione map

```
Math style S = \{x^2 : x \text{ in } \{0 \dots 9\}\}

V = (1, 2, 4, 8, \dots, 2^{12})

M = \{x \mid x \text{ in } S \text{ and } x \text{ even}\}
```

In python si traduce

```
>>> S = [x**2 for x in range(10)]
>>> V = [2**i for i in range(13)]
>>> M = [x for x in S if x % 2 == 0]
```

Esempi complessi di list comprehension

Numeri primi

```
>>> N = [j for i in range(2, 8) for j in range(i*2, 50, i)]
>>> M = [x for x in range(2, 50) if x not in N]
>>> print M
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47]
```

Ma non solo numeri



```
>>> words = 'The quick brown fox jumps over the lazy dog'.split()
>>> stuff = [[w.upper(), w.lower(), len(w)] for w in words]
```

Dictionary: array associativi

Liste chiave:valore separate dal ':'

```
>>> dict = {'Name': 'M31', 'RA': 121.1743, 'DEC': -21.5733, 'REDSHIFT':-0.001001}
>>> dict.keys()
>>> dict.values()
>>> dict['Name']='NGC224'
```

Gli oggetti dictionary hanno vari metodi

```
dict.has_key('c')
dict.copy() [shallow copy]
```

Le prossime puntate

- Scrivere un programma in python
- Le funzioni
- Leggere argomenti dalla riga di comando
- Python 4 science: numpy e scipy
- Fare grafici scientifici in python
- Usare il python per data reduction