



POLITECNICO
MILANO 1863



Fondamenti di Internet e Reti

Antonio Capone, Matteo Cesana,
Ilario Filippini, Guido Maier



POLITECNICO
MILANO 1863



Scripting con Python

**Antonio Capone, Matteo Cesana,
Ilario Filippini, Guido Maier**

Attività di laboratorio

Versioni software



python™



Gli esempi mostrati a lezione utilizzano:

- Python versione 2.7
 - python 3 ha alcune differenze sintattiche che richiedono modifiche agli esempi
- PyCharm IDE education edition
 - include python 2.7

Sono entrambi pre-installati nella macchina virtuale.

In alternativa potete installare Python e PyCharm sul vostro computer (+ libreria matplotlib)

Tempo di risposta di un server HTTP

- Valutare il tempo di risposta del server di Google per scaricare l'home page

```
01. import requests
02.
03. r = requests.get('http://www.google.com')
04. print r.elapsed.microseconds / 1000, 'ms'
```

- Ripetere la misura 10 volte

```
01. import requests
02.
03. for _ in range(10):
04.     r = requests.get('http://www.google.com')
05.     print r.elapsed.microseconds / 1000, 'ms'
```

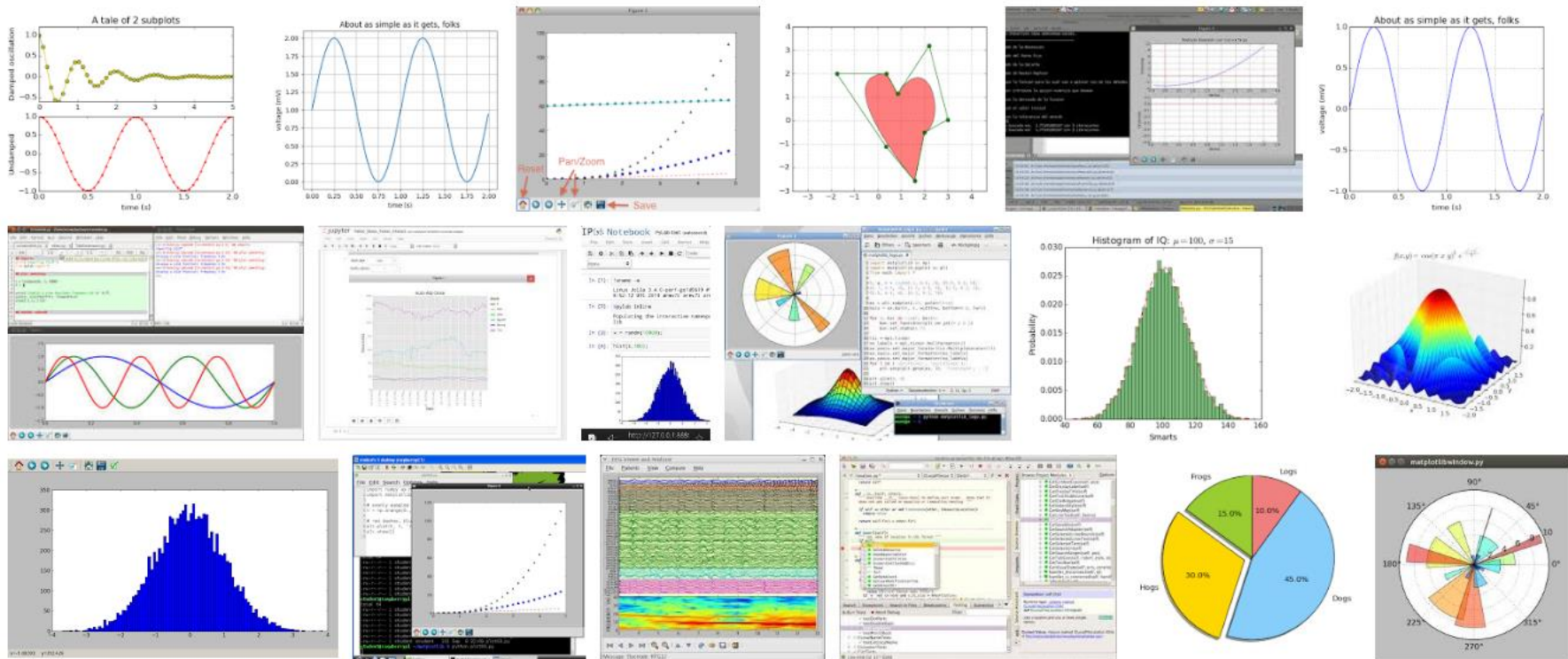
- Come possiamo calcolare minimo, media e massimo?

Calcolo minimo, media e massimo

```
01. import requests
02.
03. tempi = []
04. for _ in range(10):
05.     r = requests.get('http://www.google.com')
06.     tempi.append(r.elapsed.microseconds / 1000)
07.
08. print 'min', min(tempi)
09. print 'avg', sum(tempi)/len(tempi)
10. print 'max', max(tempi)
```

- Come possiamo rappresentare graficamente i risultati?

matplotlib



<http://matplotlib.org/examples/>

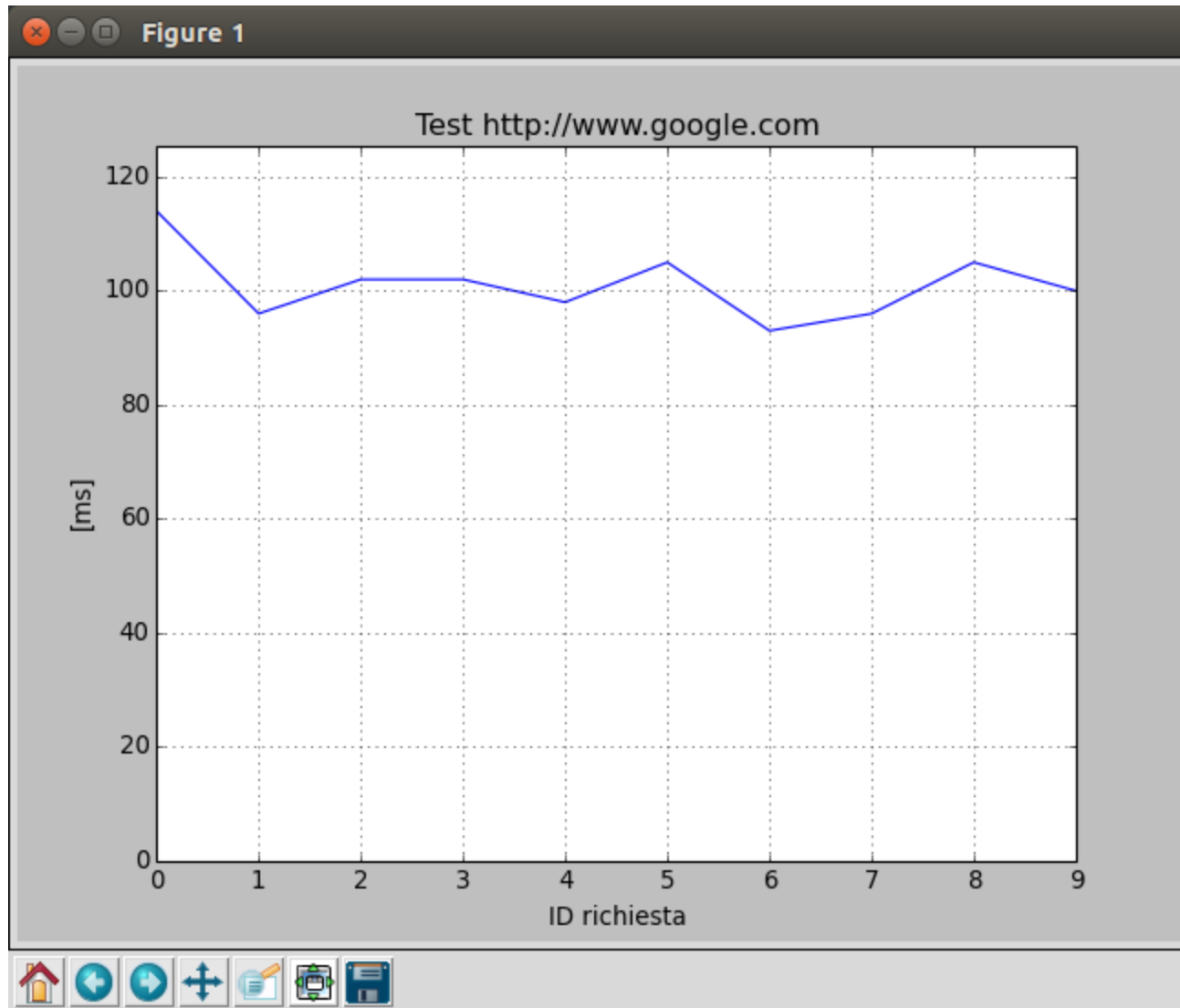
Grafici con Python

```
01. import requests
02. import matplotlib.pyplot as plt
03.
04. tempi = []
05. for _ in range(10):
06.     r = requests.get('http://www.google.com')
07.     tempi.append(r.elapsed.microseconds / 1000)
08.
09. print 'min', min(tempi)
10. print 'avg', sum(tempi)/len(tempi)
11. print 'max', max(tempi)
12.
13. plt.figure()
14. plt.plot(tempi)
15. plt.ylim([0, max(tempi)])
16. plt.show()
```

Grafici con Python (2)

```
01. import requests
02. import matplotlib.pyplot as plt
03.
04. tempi = []
05. for _ in range(10):
06.     r = requests.get('http://www.google.com')
07.     tempi.append(r.elapsed.microseconds / 1000)
08.
09. print 'min', min(tempi)
10. print 'avg', sum(tempi)/len(tempi)
11. print 'max', max(tempi)
12.
13. plt.figure()
14. plt.plot(tempi)
15. plt.ylim([0, 1.1*max(tempi)])
16. plt.xlabel('ID richiesta')
17. plt.ylabel('[ms]')
18. plt.title('Test http://www.google.com')
19. plt.grid()
20. plt.show()
```


Grafici con Python (3)



Tempo di risposta di server HTTP multipli

```
01. import requests
02.
03. siti = ['http://www.gazzetta.it', 'http://www.netflix.com', 'http://www.facebook.com']
04. for url in siti:
05.     print 'Test', url
06.     tempi = []
07.     for _ in range(10):
08.         r = requests.get(url)
09.         tempi.append(r.elapsed.microseconds/1000)
10.     print 'min', min(tempi)
11.     print 'avg', sum(tempi)/len(tempi)
12.     print 'max', max(tempi)
```

```
01. for url in siti:
02.     r = requests.get(url)
```

```
01. for ID_url in range(len(siti)):
02.     print 'Test sito #', ID_url
03.     r = requests.get(siti[ID_url])
```

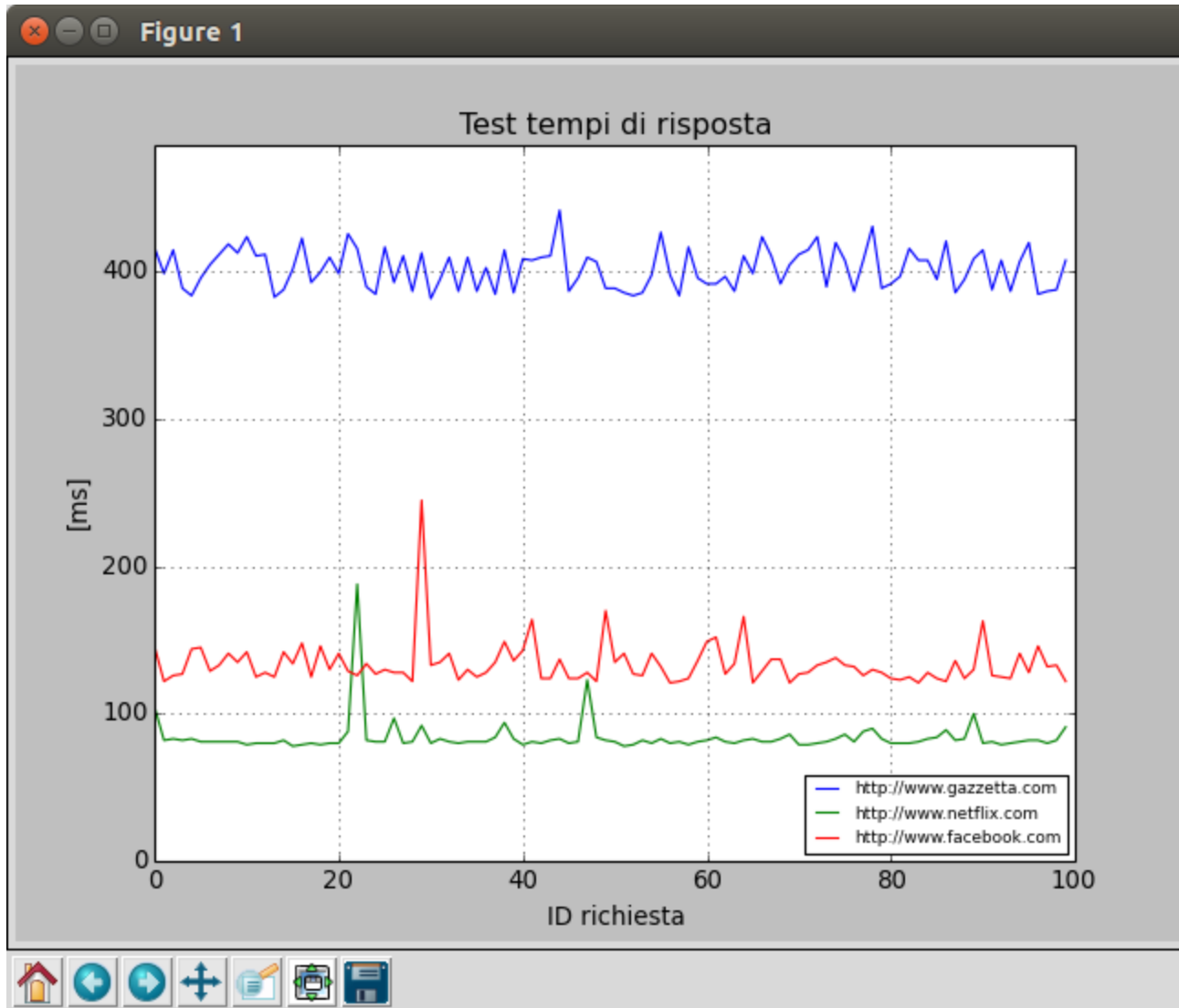
VS

```
01. for ID_url, url in enumerate(siti):
02.     print 'Test sito #', ID_url
03.     r = requests.get(url)
```

Grafici con server HTTP multipli

```
01. import requests
02. import matplotlib.pyplot as plt
03.
04. m = 0 # massimo tra i massimi
05. plt.figure()
06. siti = ['http://www.gazzetta.it', 'http://www.netflix.com', 'http://www.facebook.com']
07. for url in siti:
08.     print 'Test', url
09.     tempi = []
10.     for _ in range(10):
11.         r = requests.get(url)
12.         tempi.append(r.elapsed.microseconds/1000)
13.     plt.plot(tempi, label=url)
14.     print 'min', min(tempi)
15.     print 'avg', sum(tempi)/len(tempi)
16.     print 'max', max(tempi)
17.     m = max([m, max(tempi)]) # ricalcolo il massimo tra i massimi
18.
19. plt.ylim([0, 1.1*m])
20. plt.xlabel('ID richiesta')
21. plt.ylabel('[ms]')
22. plt.title('Test tempi di risposta')
23. plt.legend(loc='lower right', fontsize=8)
24. plt.grid()
25. plt.show()
```

Grafici con server HTTP multipli (2)



Esercizio

Scrivere uno script che stampi il nome della pagina col miglior tempo di risposta medio tra **2** siti Internet. Ripetere con **6** siti Internet.

- Numero di richieste = 10
- Siti internet:

A {
http://www.google.com
http://www.youtube.com
http://www.polimi.it
http://www.wikipedia.org
http://www.amazon.com
http://www.twitter.com
B }

Hint (per la parte B):

LISTA.index(x) ritorna la posizione dell'elemento x nella lista LIST.

È l'inverso dell'accesso alla lista tramite posizione

```
01. | L = [1, 5, 20, 4]  
02. | print L.index(5) # stampa 1
```

```
01. | L = [1, 5, 20, 4]  
02. | print L[1] # stampa 5
```

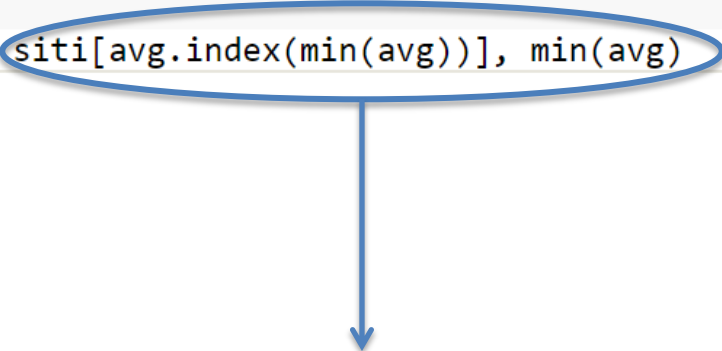
NB gli indici in Python iniziano da 0!

Soluzione parte A

```
01. import requests
02.
03. tempi1 = []
04. for _ in range(5):
05.     r = requests.get('http://www.google.com')
06.     tempi1.append(r.elapsed.microseconds/1000)
07. avg1 = sum(tempi1)/len(tempi1)
08.
09. tempi2 = []
10. for _ in range(5):
11.     r = requests.get('http://www.youtube.com')
12.     tempi2.append(r.elapsed.microseconds/1000)
13. avg2 = sum(tempi2)/len(tempi2)
14.
15. if avg1 < avg2:
16.     print 'http://www.google.com'
17. else:
18.     print 'http://www.youtube.com'
```

Soluzione parte B

```
01. import requests
02.
03. siti = ['http://www.google.com', 'http://www.youtube.com', 'http://www.polimi.it',
04.         'http://www.wikipedia.org', 'http://www.amazon.com', 'http://www.twitter.com']
05. avg = []
06. for url in siti:
07.     print 'Test', url
08.     tempi = []
09.     for _ in range(10):
10.         r = requests.get(url)
11.         tempi.append(r.elapsed.microseconds/1000)
12.     print 'avg', sum(tempi)/len(tempi)
13.     avg.append(sum(tempi)/len(tempi))
14.
15. print siti[avg.index(min(avg))], min(avg)
```



```
01. siti = ['SIT01', 'SIT02', 'SIT03']
02. avg = [150, 90, 200]
03. min(avg) → 90
04. avg.index(min(avg)) → avg.index(90) → 1
05. siti[avg.index(min(avg))] → siti[1] → 'SIT02'
```

Hostname, IP e geolocalizzazione

```
01. import socket
02. from geoip import geolite2
03.
04. # Ottiene l'indirizzo IP a partire dall'hostname
05. hostname = 'www.gazzetta.it'
06. addr = socket.gethostbyname(hostname)
07. print "L'indirizzo IP di", hostname, "e'", addr
08.
09.
10. # Ottiene lo stato a partire dall'indirizzo IP
11. match = geolite2.lookup(addr)
12. if match is not None:
13.     print hostname, 'si trova in ', match.country, match.continent, match.timezone
```

NB se non state utilizzando la VM è necessario installare i moduli:

- python-geoip
- python-geoip-geolite2
- geopy

Utilizzando Wireshark provare a verificare come viene effettuata la traduzione da hostname a indirizzo IP

Hostname, IP e geolocalizzazione (2)

```
01. import socket
02. from geoip import geolite2
03. from geopy.geocoders import Nominatim
04.
05. geolocator = Nominatim()
06.
07. siti = ['gazzetta.it', 'netflix.com', 'facebook.com', 'google.com', 'polimi.it']
08. for url in siti:
09.     addr = socket.gethostbyname(url)
10.     match = geolite2.lookup(addr)
11.     if match is not None:
12.         print "L'indirizzo IP di", url, "e'", addr
13.         print "Il fuso orario e'", match.timezone
14.         print "Le coordinate GPS sono", match.location
15.         print "L'indirizzo fisisco e'", geolocator.reverse(match.location)
16.         print
```