1-Moduli

January 9, 2021

1 Moduli

Un modulo è uno script ovvero un file di linguaggio in cui sono presente una serie di istruzioni che è possibile usare importando semplicemente il file. L'utilizzo di moduli è una procedura molto utilizzata in programmazione per la suddivisione delle funzionalità e dei compiti, inoltre riduce la necessità di ridefinire delle funzionalità. Python è un ambiente molto ricco di moduli che permettono di usare molte funzionalità anche non presenti in python grazie al collegamento con altri linguaggi.

1.1 Come creare un modulo

La creazione di un modulo è abbastanza facile, il primo passo è creare un file denominato `.py` in cui viene definita una serie di funzioni, fatto ciò è necessario salvare il file.

[3]:
```
#attenzione questo comando crea un file nel vostro folder
#appena finita la lettura, consigliabile eliminare il file in seguito
f = open('prova.py', 'w')
f.write('def saluta():\n')
f.write('\tprint("Ciao a tutti!")')
f.close()
```

1.2 Importare il modulo

Per importare il modulo che abbiamo appena creato basta usare il comando `import nome_modulo` e per accedere poi alle sue funzionalità basta usare il comando `nome_modulo.funzione` vediamo la sua applicazione.

[4]:
```
import prova
prova.saluta()
```

```
Ciao a tutti!
```

Qualora volessimo solo importare delle funzioni è possibile usare anche il comando `from nome_modulo import nome_funzione` in tal caso solo la funzione specificata verrà messa in memoria.

[6]:
```
from prova import saluta
saluta()
```

```
Ciao a tutti!
```

Qualora dovessimo usare il nome del modulo numerose volte è possibile definire un acronimo per il suo utilizzo con `import nome_modulo as nuovo_nome`.

[7]:
```
import prova as pv
pv.saluta()
```

```
Ciao a tutti!
```

2 Librerie Python

2.1 pip

Python permette di importare ed installare delle librerie attraverso un gestore di pacchetti che possiede un registro delle librerie presenti, qualora stiate usando python da terminale è possibile usare `pip` per installare il pacchetto interessato usando il comando:

[ ]:
```
pip install nome_libreria
```

2.2 anaconda

Poiché però in alcuni casi potrebbero esserci dei problemi nell'installazione è consigliabile usare invece **Anaconda** che permette un'installazione quasi sempre stabile dei pacchetti, aprite **Anaconda prompt** e digitate:

[ ]:
```
conda install nome_libreria
```

3 Aggiornare librerie python

Nel futuro sarà necessario per voi aggiornare le librerie che state utilizzando ed in tal caso potete usare i seguenti comandi.

3.1 anaconda

Qualora utilizziate **Anaconda** cosa fortemente consigliata potete aprire **Anaconda prompt** e digitare:

[ ]:
```
conda update conda
```

Questo vi aggiornerà sia anaconda che i pacchetti python, qualora vogliate aggiornare solo i pacchetti dovete usare:

[ ]:
```
conda update --all
```

3.2 pip

Per pip la situazione è più complicata potete usare il seguente comando:

[ ]:
```
pip freeze --user | cut -d'=' -f1 | xargs -n1 pip install -U
```

La spiegazione del comando la potete trovare **qui**. In alternativa potete usare il pacchetto **pipupgrade** dopo averlo installato. Comando installazione:

[ ]:
```
pip install pipupgrade
```

Comando per aggiornare tutti i pacchetti:

[ ]:
```
pipupgrade --all
```

COMPLIMENTI AVETE COMPLETATO LA LEZIONE SUI MODULI!