

iPythonWHAT-2

February 19, 2020

```
[1]: 2+10
```

```
[1]: 12
```

```
[2]: _+10    # previous output by underscore
```

```
[2]: 22
```

```
[ ]: %env
```

```
[5]: In[1]
```

```
[5]: '2+10'
```

```
[ ]:
```

```
[18]: _1 # equivale ad Out[1]
```

```
[18]: 12
```

```
[20]: _i1 # equivale ad In[1]
```

```
[20]: '2+10'
```

```
[7]: %history -n 1-3
```

```
1: 2+10
2: _+10# previous output by underscore
3: %env
```

```
[9]: %pwd
```

```
[9]: '/home/andrea/andrea.prestini@gmail.com/pythonWorld/prove Coding'
```

```
[10]: !pwd
```

```
/home/andrea/andrea.prestini@gmail.com/pythonWorld/prove Coding
```

```
[2]: files = !ls
      print("nella mia directory ci sono questi files:\n", files)
      # il punto esclamativo esegue comandi della shell
```

nella mia directory ci sono questi files:

```
['attributo_classe_IF.py', 'calcolatrice.py', 'ciclo_For_and_Range.py', 'ciclo
While.py', 'classi_esempio_facile.py', 'collections_Python.ipynb',
'conversione_input_altro.py', 'crea_cancella_cartelle.py',
'crea_popola_file.py', 'csv_files.py', 'dizionari2.py', 'dizionari3.py',
'dizionari.py', 'elenco', 'Errori.py', 'file_esterni.py', 'flussi.py',
'fonte.json', 'funzioni.py', 'GUI.py', 'Indent_dict.ipynb', 'iPythonWHAT.ipynb',
'lambda_MAGIC.py', 'listComprehension.py', 'liste_1.py', 'liste2.py',
'liste3.py', 'logica_Bool_flusso.py', 'log.txt', 'Maps_Filters_Reduce.py',
'OUT_print', 'Panda_analisy.py', '__pycache__', 'ZerotoHero_1.ipynb',
'ZerotoHero_2.ipynb', 'ZIP_file.py']
```

Esportiamo la history in un file con l'opzione -f ~~~ %history -f storia.py %history -l 10 -f storia.py
~~~

```
[4]: for file in files: # files è una lista
      print(file)
```

```
attributo_classe_IF.py
calcolatrice.py
ciclo_For_and_Range.py
ciclo While.py
classi_esempio_facile.py
collections_Python.ipynb
conversione_input_altro.py
crea_cancella_cartelle.py
crea_popola_file.py
csv_files.py
dizionari2.py
dizionari3.py
dizionari.py
elenco
Errori.py
file_esterni.py
flussi.py
fonte.json
funzioni.py
GUI.py
Indent_dict.ipynb
iPythonWHAT.ipynb
lambda_MAGIC.py
listComprehension.py
liste_1.py
liste2.py
```

```

liste3.py
logica_Bool_flusso.py
log.txt
Maps_Filters_Reduce.py
OUT_print
Panda_analisis.py
__pycache__
ZerotoHero_1.ipynb
ZerotoHero_2.ipynb
ZIP_file.py

```

```
[5]: !echo $files # la variabile ipython è diventata una variabile di SHELL
```

```

[attributo_classe_IF.py, calcolatrice.py, ciclo_For_and_Range.py, ciclo
While.py, classi_esempio_facile.py, collections_Python.ipynb,
conversione_input_altro.py, crea_cancella_cartelle.py, crea_popola_file.py,
csv_files.py, dizionari2.py, dizionari3.py, dizionari.py, elenco, Errori.py,
file_esterni.py, flussi.py, fonte.json, funzioni.py, GUI.py, Indent_dict.ipynb,
iPythonWHAT.ipynb, lambda_MAGIC.py, listComprehension.py, liste_1.py, liste2.py,
liste3.py, logica_Bool_flusso.py, log.txt, Maps_Filters_Reduce.py, OUT_print,
Panda_analisis.py, __pycache__, ZerotoHero_1.ipynb, ZerotoHero_2.ipynb,
ZIP_file.py]

```

```
[13]: # comando shell con variabile python
!echo {files[0].upper()}
!echo {files[1].upper()}
!echo {files[2].upper()}
!echo {files[3].upper()}

```

```

ATTRIBUTO_CLASSE_IF.PY
CALCOLATRICE.PY
CICLO_FOR_AND_RANGE.PY
CICLO WHILE.PY

```

```
[27]: import os
for i, f in enumerate(files):
    if f.endswith(".ipynb"):
        !echo {"%02d" % i} - "{os.path.splitext(f)}"
        !echo {"%02d" % i} - "{os.path.splitext(f)[0]}"

```

```

05 - ('collections_Python', '.ipynb')
05 - collections_Python
20 - ('Indent_dict', '.ipynb')
20 - Indent_dict
21 - ('iPythonWHAT', '.ipynb')
21 - iPythonWHAT
33 - ('ZerotoHero_1', '.ipynb')

```

```
33 - ZerotoHero_1
34 - ('ZerotoHero_2', '.ipynb')
34 - ZerotoHero_2
```

## 1 Magic Functions

```
[1]: %%timeit list(range(1000)) # in line %%timeit più linee di codice della cella
```

20.4  $\mu$ s  $\pm$  389 ns per loop (mean  $\pm$  std. dev. of 7 runs, 10000 loops each)

```
[6]: %%bash
echo "ciao amico mio"
echo "come stai"
```

```
ciao amico mio
come stai
```

```
[9]: %%writefile amico.txt
questo è un file di test per magic functions in ipython
```

```
Writing amico.txt
```

```
[12]: !cat amico.txt
```

```
questo è un file di test per magic functions in ipython
```

```
[13]: %lsmagic
```

```
[13]: Available line magics:
```

```
%alias %alias_magic %autoawait %autocall %automagic %autosave %bookmark
%cat %cd %clear %colors %conda %config %connect_info %cp %debug %dhist
%dirs %doctest_mode %ed %edit %env %gui %hist %history %killbgscripts
%ldir %less %lf %lk %ll %load %load_ext %loadpy %logoff %logon
%logstart %logstate %logstop %ls %lsmagic %lx %macro %magic %man
%matplotlib %mkdir %more %mv %notebook %page %pastebin %pdb %pdef %pdoc
%pfile %pinfo %pinfo2 %pip %popd %pprint %precision %prun %psearch
%psource %pushd %pwd %pycat %pylab %qtconsole %quickref %recall %rehashx
%reload_ext %rep %rerun %reset %reset_selective %rm %rmdir %run %save
%sc %set_env %store %sx %system %tb %time %timeit %unalias %unload_ext
%who %who_ls %whos %xdel %xmode
```

```
Available cell magics:
```

```
%%! %%HTML %%SVG %%bash %%capture %%debug %%file %%html %%javascript
%%js %%latex %%markdown %%perl %%prun %%pypy %%python %%python2
%%python3 %%ruby %%script %%sh %%svg %%sx %%system %%time %%timeit
%%writefile
```

Automagic is ON, % prefix IS NOT needed for line magics.

### 1.0.1 eseguiamo codice in altri linguaggi

```
[ ]: %%ruby
name = "world"
puts "hello #{name.capitalize}!"
```

```
[17]: %%perl
@months = ("july", "august", "september");
print $months[0];
```

july

```
[23]: %connect_info
```

```
{
  "shell_port": 56035,
  "iopub_port": 48265,
  "stdin_port": 50773,
  "control_port": 60577,
  "hb_port": 54763,
  "ip": "127.0.0.1",
  "key": "11d4e84c-b40d75ca1d14330c4181e3ff",
  "transport": "tcp",
  "signature_scheme": "hmac-sha256",
  "kernel_name": ""
}
```

Paste the above JSON into a file, and connect with:

```
$> jupyter <app> --existing <file>
```

or, if you are local, you can connect with just:

```
$> jupyter <app> --existing kernel-df2d3ba8-259d-48db-b1d9-229c705c89b5.json
```

or even just:

```
$> jupyter <app> --existing
```

if this is the most recent Jupyter kernel you have started.

## 2 Rich output

### 2.1 Basic display import

```
[30]: from IPython.display import Image
```

```
[33]: !mv ./OUT_print/asa ./OUT_print/asa.jpg
```

```
[35]: i = Image(filename="./OUT_print/asa.jpg")
```

[37]: `display(i)`



```
[38]: Image(url="http://python.org/images/python-logo.gif")
```

```
[38]: <IPython.core.display.Image object>
```

```
[2]: from IPython.display import Math
```

```
[3]: Math(r'F(k) = \int_{-\infty}^{\infty} f(x) e^{2\pi i k} dx')
```

```
[3]:
```

$$F(k) = \int_{-\infty}^{\infty} f(x) e^{2\pi i k} dx$$

```
[8]: Math('\sum_{i=0}^{i=100} x_i')
```

```
[8]:
```

$$\sum_{i=0}^{i=100} x_i$$

```
[24]: Math('\dfrac{x+y_i}{3z}')
```

```
[24]:
```

$$\frac{x + y_i}{3z}$$

```
[25]: from IPython.display import Audio  
Audio(url="http://www.nch.com.au/acm/8k16bitpcm.wav")
```

```
[25]: <IPython.lib.display.Audio object>
```