

Top 10 Magic Commands in Python to Boost your Productivity

Implementation of important IPython magic commands in jupyter notebook



Siddhesh Jadhav

Apr 12 · 6 min read



Photo by Artem Maltsev on Unsplash

Python is not only the most versatile programming language but also most flexible when it comes to integrating new features. With this said, Magic commands are one of the important features added to the python shell.

What exactly is a Magic Command in python?

Magic commands are enhancements added over the normal python code and these commands are provided by the IPython kernel.

These magic commands are usually prefixed by a “%” character

These commands are basically added to solve common problems we face and also provide few shortcuts to your code.

There are 2 flavours of magic commands available — % prefix and %% prefix

% prefix represents that the command operates over a single line of code whereas %% prefix allows the command to operate over an entire cell.

Following are the list of magic commands along with their implementation which are all executed in jupyter notebook.

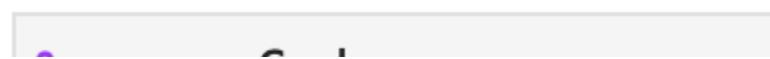
Running External File

As we try running few snippets in jupyter notebooks, we wish to run an external code file located in some directory.

%run allows you to run any external python file from jupyter notebook

```
myCode.py
1 def myprint():
2     print("The World Makes Sense!")
3
4 myprint()
```

The above file myCode.py contains a simple script that outputs the mentioned statement.



%run myCode.py

The World Makes Sense!

If we specify the filename including the path to the %run command, it will execute the file.

Note: %run also allows the execute of external jupyter notebooks.

Code Execution Time

Ever wondered how much exact time does your cell takes to run?



Photo by Luke Chesser on Unsplash

The time magic command allows to track the total execution of your cell.

Since we will be dealing with an entire cell here, we will use %% as prefix before time keyword.



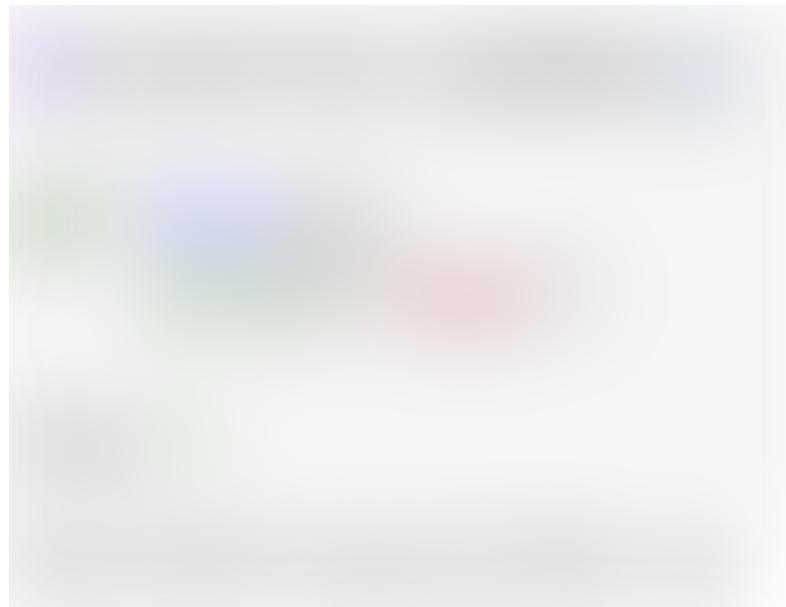
The above cell includes a for loop with random calculation. %%time helps in getting the execution time required for running the for loop.

Copy content to external file

Most of the times, you feel the need to add your content into python script or a text file directly from your jupyter notebook.

Instead of copying everything and creating a new file, you can directly export your cell content by adding writefile command before the code.

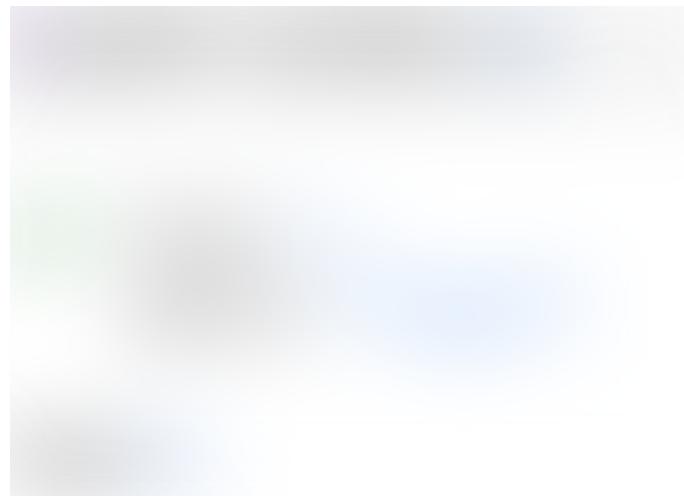
Notice the double % before the command which indicates the whole content of the cell will be exported.



Since I had the file already created with some content, it displays ‘Overwriting myCode.py’ specifying that it will overwrite my original content with the content shown in the above image.

Display content of external file

Usually you feel the need to copy few lines of code from external file into your code. Instead of the long procedure to fetch the file and opening it up to copy, %pycat allows you to display the content of any file in any directory.



It displays all the content of the external file as its output. It can be considered as the reverse of %writefile in terms of its application.

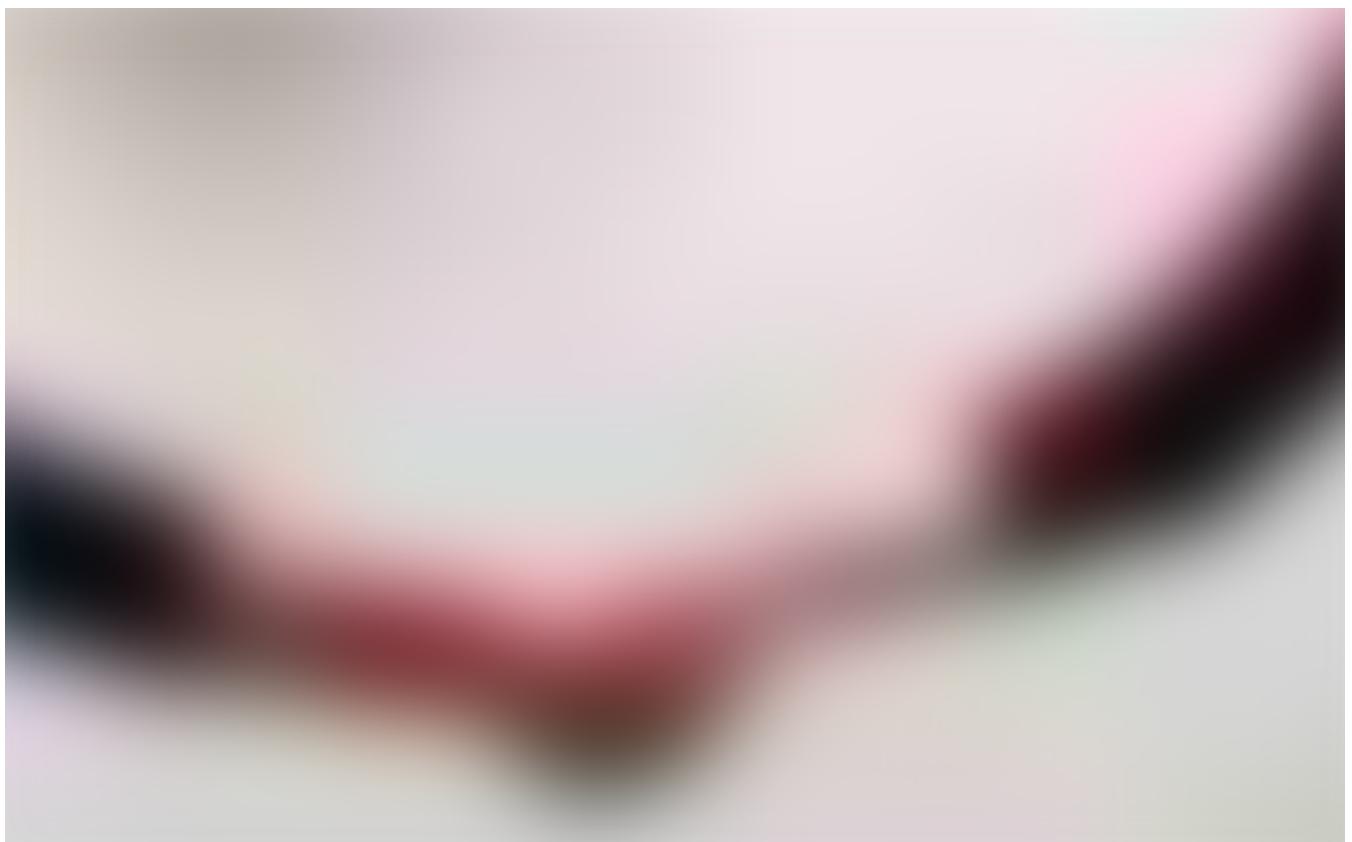


Photo by Chris Liverani on Unsplash

Hold on tight! Plenty of amazing commands are still left to explore.

List all variables

This magic command displays all the variables used in the whole notebook.

Following are 3 variables — 2 strings and 1 integer. If we run `%who`, it will list down all the 3 variables which we have defined.

```
a = "hello"  
b = "Good Morning"  
c = 1
```



The above code displays all the variables irrespective of their data types.



In order to display specific data type variables, we need to pass the data type after the magic command. The above code displays all the string data type variables as its output.

Share variable between notebooks

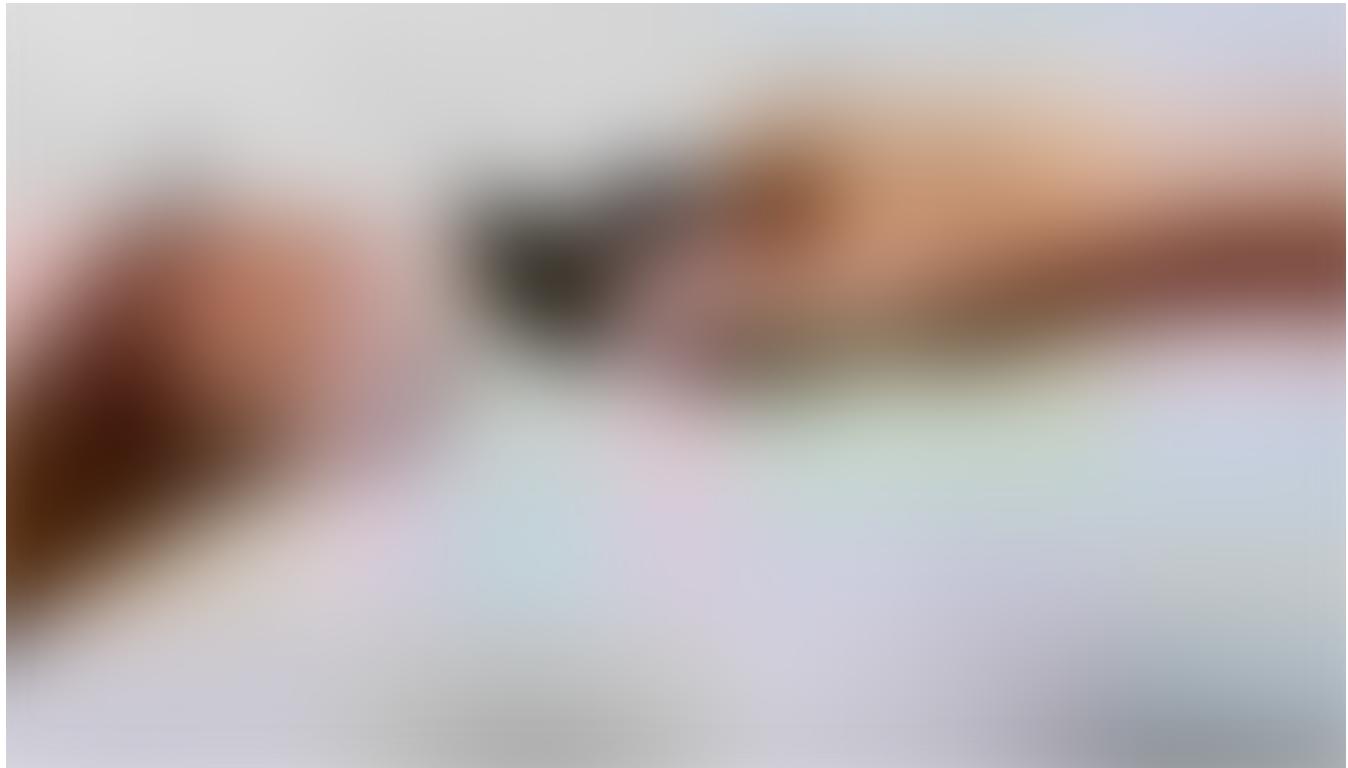


Photo by Kelly Sikkema on Unsplash

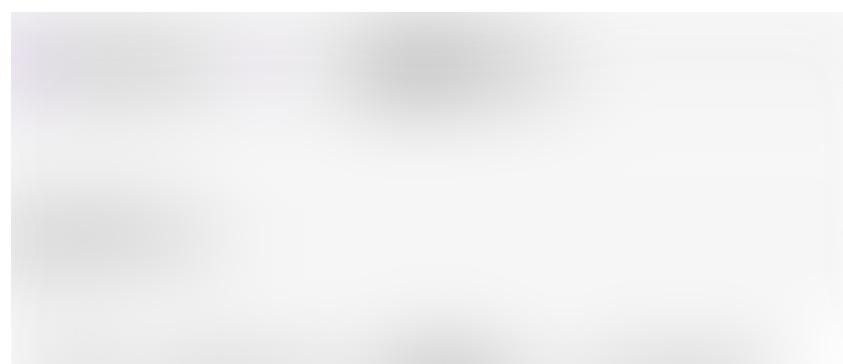
This magic command allows you to share any variable between different Jupyter notebooks. You need to pass the original variable with the magic command.

To retrieve the variable, you need to pass the same command with '-r' parameter.

This is how the first notebook looks like



The code needed to retrieve this data is written in another notebook.



This might be the simplest way to share data of any data type between different notebooks.

Execute html script

%%html allows us to write html code in the cell. The cell will now act as a html editor with html output of the cell.

Following code comprises of a simple table created in html. You can notice the html output displaying the expected table.

```
%%html
<html>
<body>
<table>
    <tr>
        <th>Name</th>
        <th>Country</th>
        <th>Age</th>
    </tr>
    <tr>
        <td>Sid</td>
        <td>India</td>
        <td>22</td>
    </tr>
    <tr>
        <td>Dave</td>
        <td>UK</td>
        <td>28</td>
    </tr>
</table>
</body>
</html>
```

Tip : You can run Javascript code in a cell using the %%js magic command similar to HTML magic command.

Display Matplotlib graphs

%matplotlib inline magic command is the most popular command. This command allows Jupyter notebook to display matplotlib graphs in notebooks. This command activates matplotlib interactive support for your jupyter notebook.

```
import random
import matplotlib.pyplot as plt
%matplotlib inline
```

We have imported few libraries we require to explain the functioning of the command.

We will now create two random list for plotting a graph

```
a = []
b = []
for i in range(10):
    a.append(random.randint(0,10))
    b.append(random.randint(0,10))
```

Now we will plot a scatter graph of the data.

```
plt.scatter(a,b)
```

The %matplotlib inline magic command allows you to visualize graph inside jupyter notebook.

Set Environment variables

This magic commands allows you to do 3 things — Lists all environment variables, get the value of a particular environment variable and set a value for a variable.

%env with no parameter will list down all the environment variables.

%env with a single parameter will return the value for the specified parameter.

'%env variable value' will set the value for the specified variable name.

Object detailed information

%pinfo provides a detailed information about the object which is passed along with it. It is similar to **object?** function.

In the following snippet, I have passed a simple string 'a' along with %pinfo to get detailed information about it.

```
a = "The World Makes Sense!"  
%pinfo a
```



From the above output, %pinfo provides all the information about the string object.

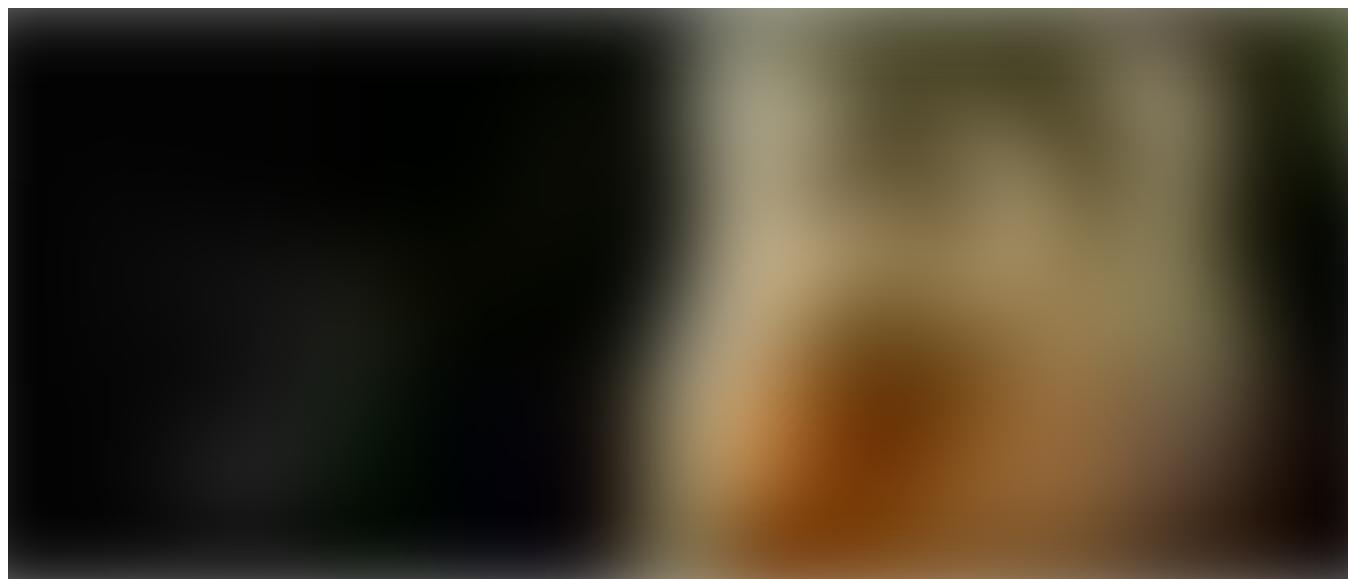


Photo by Alex Guillaume on Unsplash

You can find all the magic command list using the '`%lsmagic`' command.

```
%lsmagic
```



These were my top 10 magic commands which can help you boost your productivity and save your time.

Hope you like it!

Data Science Artificial Intelligence Programming Python Technology

About Help Legal

Get the Medium app

