

The other nice thing about this tab is that you can go through your running applications and shut them down there.

Adding Rich Content

Jupyter Notebook supports adding rich content to its cells. In this section, you will get an overview of just some of the things you can do with your cells using Markup and Code.

Cell Types

There are technically four cell types: Code, Text, Raw NBConvert, and Heading.

The Heading cell type is no longer supported. You can use Markdown for your Headings.

The Raw NBConvert cell type is only intended for advanced users. Basically it allows you to control the format of the output.

The primary cell types that you will use are Code and Text cells. Let's see how these cells work, so let's learn how to style your text.

```
1 # How to merge two dicts
2 # in Python 3.5+
3
4 >>> x = {'a': 1, 'b': 2}
5 >>> y = {'b': 3, 'c': 4}
6
7 >>> z = {**x, **y}
8
9 >>> z
10 {'c': 4, 'a': 1, 'b': 3}
```

Improve Your Python

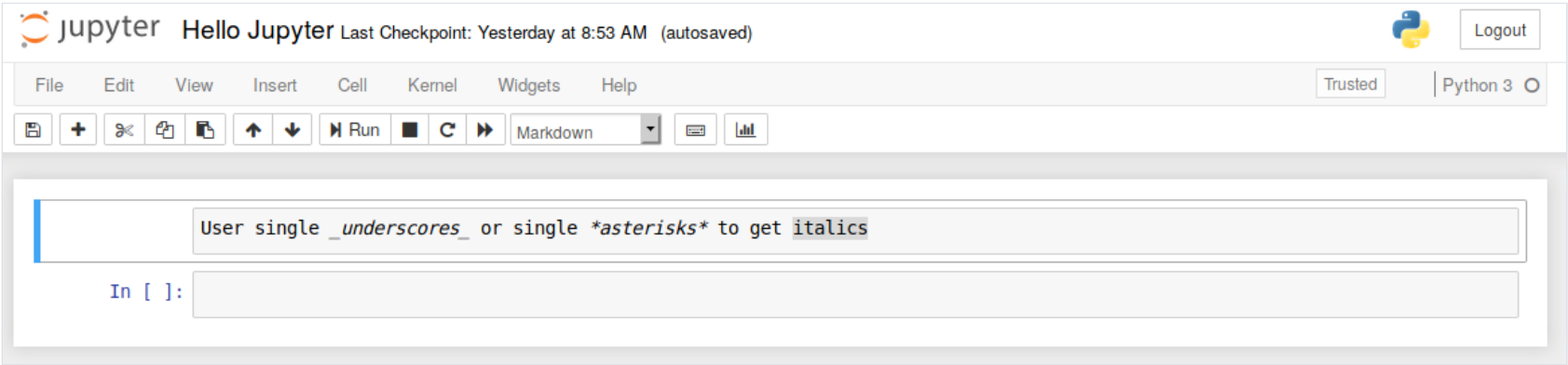
...with a fresh  **Python Trick**  code snippet every couple of days:

[Send Python Tricks »](#)

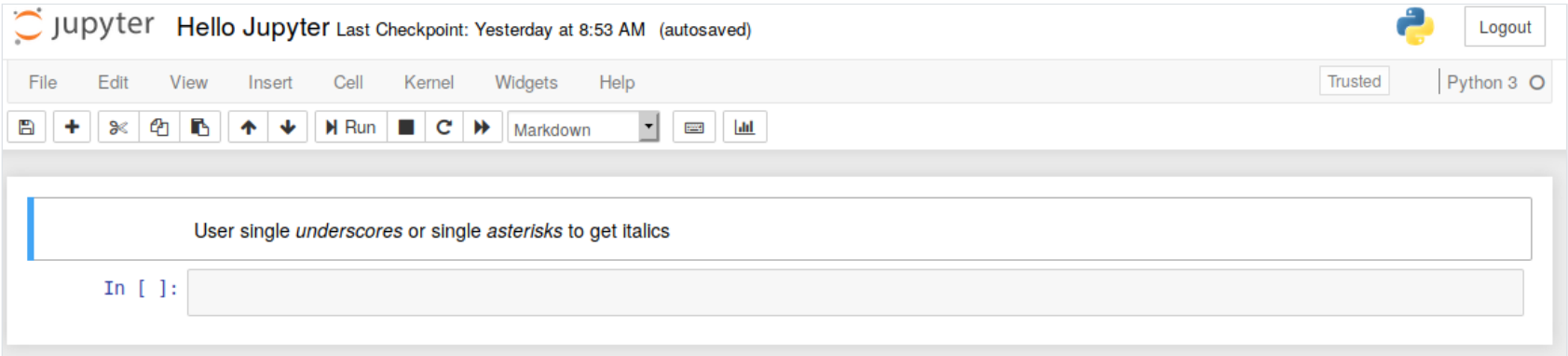
Styling Your Text

Jupyter Notebook supports Markdown, which is a markup language that is a superset of HTML. This tutorial will cover some of the basics of what you can do with Markdown.

Set a new cell to Markdown and then add the following text to the cell:



When you run the cell, the output should look like this:



If you would prefer to bold your text, use a double underscore or double asterisk.

Headers

Creating headers in Markdown is also quite simple. You just have to use the humble pound sign. The more pound signs you use, the smaller the header. Jupyter Notebook even kind of previews it for you: