

When you are working with Jupyter Notebooks, you will find that you need to share your results with non-technical people. When that happens, you can use the `nbconvert` tool which comes with Jupyter Notebook to convert or export your Notebook into one of the following formats:

- HTML
- LaTeX
- PDF
- RevealJS
- Markdown
- ReStructured Text
- Executable script

The `nbconvert` tool uses Jinja templates formats.

[Jinja](#) is a template engine that was made able to export to all the formats above. I work. For more information, you should

```
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:

Email Address

Send Python Tricks »

## How to Use nbconvert

The `nbconvert` command does not take very many parameters, which makes learning how to use it easier. Open up a terminal and navigate to the folder that contains the Notebook you wish to convert. The basic conversion command looks like this:

Shell

```
$ jupyter nbconvert <input
notebook> --to <output format>
```

## Example Usage

Let’s pretend that you have a Notebook named `py_examples.ipynb` and you want to convert it to PDF. Here is the command you would use to do that:

Shell

```
$ jupyter nbconvert py_examples.ipynb --to pdf
```

When you run this command, you should see some output that tells you about the conversion process. `nbconvert` will display warnings and errors if there are any. Assuming everything goes according to plan, you will now have a `py_examples.pdf` file in your folder.

The conversion process for the other file types is quite similar. You just have to tell `nbconvert` what type to convert to (PDF, Markdown, HTML, and so on).

## Use the Menu

You can also export your currently running Notebook by going to the *File* menu and choosing the *Download as* option.

This option allows you to download in all the formats that `nbconvert` supports. The benefit of using the menu is that you don’t need to learn the `nbconvert` at all if you don’t want to. However I recommend doing so as you can use `nbconvert` to export multiple Notebooks at once, which is something that the menu does not support.

## Notebook Extensions

While Jupyter Notebooks have lots of functionality built in, you can add new functionality through extensions. Jupyter actually supports four types of extensions:

- Kernel

Improve Your Python