

```
$ 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
- IPython kernel
- Notebook
- Notebook server

This tutorial will focus on Notebook extensions.

## What Are Extensions?

A Notebook extension (`nbextension`) is a JavaScript module that you load in most of the views in the Notebook's frontend. If you are handy with JavaScript, you can even write your own extension. An extension can access the page's DOM and the Jupyter JavaScript API.

## Where Do I Get Extensions?

You can use Google or search for Jupyter Notebook extensions. There are actually quite a few out there. One of the most popular extension sets is called **jupyter\_contrib\_nbextensions**, which you can get from [GitHub](#). This is actually a collection of extensions that is provided by the Jupyter community and installed with `pip`.

## How Do I Install Them?

Most Jupyter Notebook extensions can be installed using Python's `pip` tool. If you find an extension that can't be installed with `pip`, then you will likely have to use the following command:

Shell

```
$ jupyter nbextension install EXTENSION_NAME
```