

# chsh\_pip\_install

November 22, 2019

## 1 Pip install from a notebook

How to install a module from a notebook.

```
[1]: from jupyterlab import add_notebook_menu
      add_notebook_menu()
```

```
[1]: <IPython.core.display.HTML object>
```

### 1.1 Update or install a module from the notebook

Running `pip install` from the command line requires to be in the right folder. And sometimes, several python installations interfere between each others. Why doing it from the notebook itself:

```
[2]: try:
      # pip >= 19.3
      from pip._internal.main import main as pip_main
    except Exception:
      try:
        # pip >= 10.0
        from pip._internal import main as pip_main
      except Exception:
        # pip < 10.0
        from pip import main as pip_main
```

```
[3]: pip_main("install -q qgrid".split())
```

```
[3]: 0
```

### 1.2 Interesting options

#### 1.2.1 Avoid installing dependencies

```
[4]: try:
      pip_main("install -q qgrid --no-deps".split())
    except Exception as e:
      print(e)
```

#### 1.2.2 Upgrade

```
[5]: try:
      pip_main("install -q qgrid --upgrade --no-deps".split())
      except Exception as e:
          print(e)
```

### 1.2.3 No cache

By default, `pip` uses cached version. So, if a module has just been updated, `pip` might choose to use the previous version. To tell it not to do so:

```
[6]: pip_main("install -q qgrid --upgrade --no-deps --no-cache-dir".split())
```

```
[6]: 0
```

## 1.3 For the hackathon...

```
pip_main("install pyquickhelper pyensae ensae_projects --upgrade --no-deps
--no-cache-dir".split())
```

```
[7]:
```

```
[8]:
```