

File editing and Reload Demo

Outline

1. Editing Files on Datahub
2. Reloading Module in Jupyter Notebook
3. `%autoreload`
4. What about IDE?

You can edit files directly on datahub! (just click on

⚙️



Files

Running

Clusters

Nbextensions

Select items to perform actions on them.



 `utils.py`

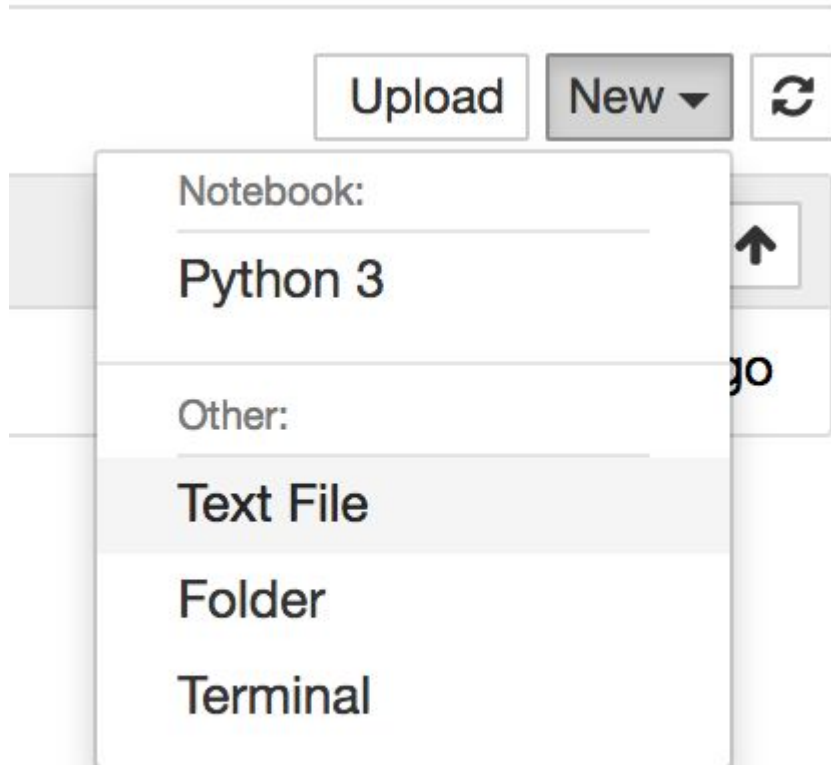
Here we have a functional minimalist code editor

 jupyter utils.py ✓ a minute ago

File Edit View Language

```
1 def helper_function():  
2     return "Hello World"
```

It can be created directly on datahub (via Text File)



You can then import the function from notebook

- The notebook and utils.py need to in the same directory.

```
In [1]: from utils import helper_function
```

```
In [2]: helper_function()
```

```
Out[2]: 'Hello World'
```

Complex Cases (Optional)

- What if they are not in the same directory?
- You can do relative import like “from ..utils.plotting import barplot”: go to the parent directory, then go to the utils module, in a py file called plotting, import the barplot function.

root

|— my_notebooks

| └─ demo.ipynb (current notebook)

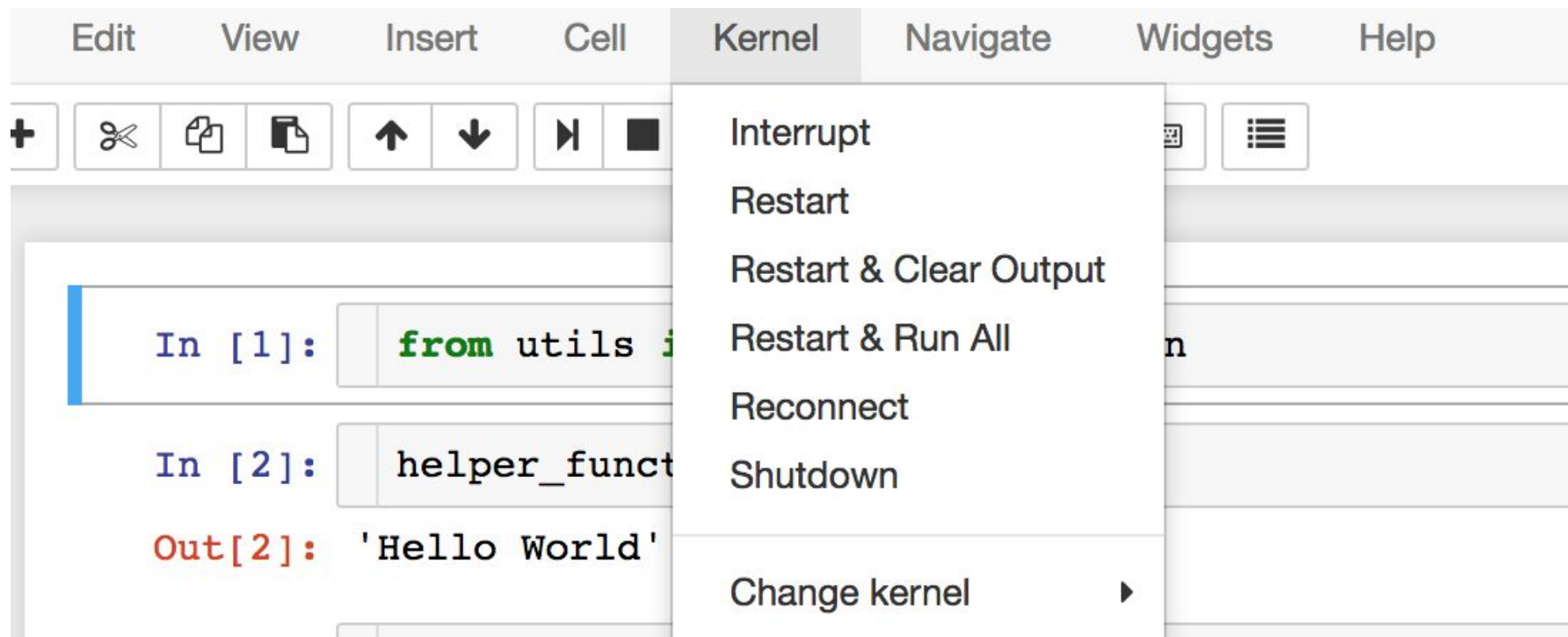
└─ utils

└─ plotting.py (has barplot function)

Reload

- Let's say you edited the py file, saved it, how can the changes show up in the notebook? There are 3 options:
 1. "Restart Kernel"
 2. "from importlib import reload"
 3. "%autoreload"

Restart Kernel



The image shows a Jupyter Notebook interface with the 'Kernel' menu open. The menu options are:

- Interrupt
- Restart
- Restart & Clear Output
- Restart & Run All
- Reconnect
- Shutdown
- Change kernel ▶

The notebook content shows the following code and output:

```
In [1]: from utils import
```

```
In [2]: helper_func
```

```
Out[2]: 'Hello World'
```

But I don't want to run the restart kernel 100 times!

- You can “reload” the module
- Be careful about the specific ordering.

1. `import some_module`
2. `reload(some_module)`
3. `from some_module import some_function`
4. `# Change the code`
5. `reload(some_module)`
6. `from some_module import some_function`

```
In [1]: from importlib import reload
```

```
In [2]: import utils  
        reload(utils)
```

```
Out[2]: <module 'utils' from '/Users/simonmo,
```

```
In [3]: from utils import helper_function
```

```
In [4]: helper_function()
```

```
Out[4]: 'Hello World'
```

Now we make some changes in utils.py

```
In [5]: reload(utils)  
        from utils import helper_function
```

```
In [6]: helper_function()
```

```
Out[6]: 'Hello World - V2'
```

1. import some_module
2. reload(some_module)
3. from some_module import
 some_function
4. # Change the code
5. reload(some_module)
6. from some_module import
 some_function

There must be a better way!

- Thanks IPython! There is a better way.

```
In [1]: %load_ext autoreload
```

```
In [2]: %autoreload 2
```

```
In [3]: from foo import some_function
```

```
In [4]: some_function()
```

```
Out[4]: 42
```

```
In [5]: # open foo.py in an editor and change some_function to return 43
```

```
In [6]: some_function()
```

```
Out[6]: 43
```

But there comes a cost

- It's not perfect. If you see some odd behavior, what can you do?

But there comes a cost

- It's not perfect. If you see some odd behavior, what can you do?
- You know how to do **manual reload**.

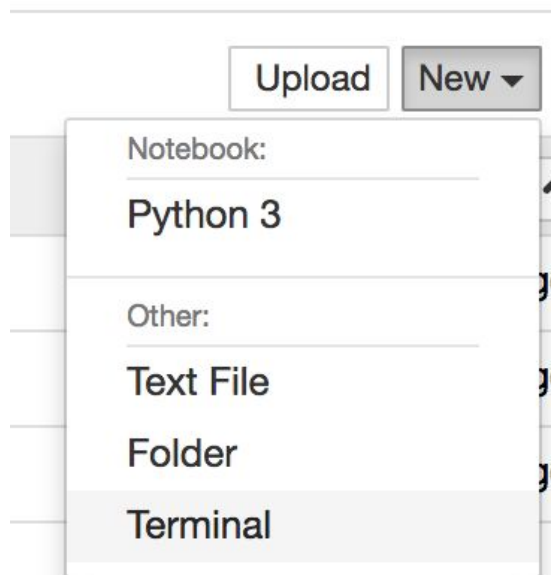
- But what if that won't work?

But there comes a cost

- It's not perfect. If you see some odd behavior, what can you do?
- You know how to do manual reload.
- But what if that won't work?
- Answer:
 - **Restart the Kernel**

Side note: wget or curl in terminal

- You can download files directly from the internet to datahub.
- Go to terminal:
- `wget link_to_file`
- `curl link_to_file -o filename`




```
bash-3.2$ wget https://github.com/DS-100/sp18/raw/gh-pages/assets/favicon/android-chrome-512x512.png
--2018-02-04 23:24:53-- https://github.com/DS-100/sp18/raw/gh-pages/assets/favicon/android-chrome-512x512.png
Resolving github.com (github.com)... 192.30.253.113, 192.30.253.112
Connecting to github.com (github.com)|192.30.253.113|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://raw.githubusercontent.com/DS-100/sp18/gh-pages/assets/favicon/android-chrome-512x512.png [following]
--2018-02-04 23:24:53-- https://raw.githubusercontent.com/DS-100/sp18/gh-pages/assets/favicon/android-chrome-512x512.png
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 151.101.64.133, 151.101.0.133, 151.101.128.133, ...
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|151.101.64.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 58060 (57K) [image/png]
Saving to: 'android-chrome-512x512.png'

android-chrome-512x512.png      100%[=====>] 56.70K  --.-KB/s   in 0.06s

2018-02-04 23:24:54 (966 KB/s) - 'android-chrome-512x512.png' saved [58060/58060]
```

```
bash-3.2$ ls
android-chrome-512x512.png
```

```
bash-3.2$ curl https://github.com/DS-100/sp18/raw/gh-pages/assets/favicon/android-chrome-512x512.png -o favicon.png
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100  162  100  162    0    0   162     0  0:00:01 --:--:--  0:00:01  419
```

```
bash-3.2$ ls
android-chrome-512x512.png  favicon.png
```

Finally, if you are working locally...

- Unless you have a text editor of your choice,
- we recommend visual studio code + python extension.

