

```
1 %load_ext autoreload
2 %autoreload 2
```

```
1 from google.colab import drive
2 drive.mount('/content/drive')
```

➞ Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client_

Enter your authorization code:

.....

Mounted at /content/drive

```
1 import os, sys
2
3 DATAPATH = '/content/drive/My Drive/Coursera/EDHEC/investment-portfolio/data'
4 print(f"DATAPATH:{DATAPATH} contents:{os.listdir(DATAPATH)}")
5
6 MODULEPATH = '/content/drive/My Drive/Coursera/EDHEC/investment-portfolio/nb'
7 print(f"MODULEPATH:{MODULEPATH} contents:{os.listdir(MODULEPATH)}")
8
9 sys.path.append(MODULEPATH)
10 print(f"sys.path:{sys.path}")
```

➞ DATAPATH:/content/drive/My Drive/Coursera/EDHEC/investment-portfolio/data contents:
MODULEPATH:/content/drive/My Drive/Coursera/EDHEC/investment-portfolio/nb contents:
sys.path:['', '/env/python', '/usr/lib/python36.zip', '/usr/lib/python3.6', '/

▼ Building a Module

A module in Python is conceptually not much more than a file with python code in it that you can a module.

When you `import` a module, you are reading the file and making the code contained in it accessible.

Start by creating a new file called `hello.py` and add the following message in there:

```
message = 'Hello Jim'
```

and then try executing this code:

```
import hello as h
h.message
```

Great!

Now, let's change the message. Edit the file to now say:

```
message = 'Hello Jane'
```

and reexecute the code:

```
import hello as h
h.message
```

What went wrong?

Since the contents of modules don't usually change very much (except during module development), Python doesn't bother re-importing the module. Normally, this is fine. But when we are developing the module we want to force the module to get automatically reloaded every time we edit the file.

Fortunately, there is a *magic* command sequence that does exactly that. First, we have to load an extension module called `%load_ext autoreload`. Once you have loaded that extension, you can execute a *command* called `%autoreload` which supports different modes of autoreload. The mode we want is mode 2, which reloads any module that changes, which is mode 2. So, you need to execute:

```
%load_ext autoreload
%autoreload 2
```

Now try executing:

```
h.message
```

and you will see that future edits to the hello file will immediately be reloaded.

Try editing the `hello.py` file to:

```
message = 'Hello John and Jane'
```

and then execute

```
h.message
```

Now, let's create our new module, which we'll build on through the course, and put the `drawdown` function in it. Create a file called `edhec_risk_kit.py` and copy the following code into it.

```
import pandas as pd

def drawdown(return_series: pd.Series):
    """Takes a time series of asset returns.
    returns a DataFrame with columns for
    the wealth index,
    the previous peaks, and
    the percentage drawdown
    """
    wealth_index = 1000*(1+return_series).cumprod()
    previous_peaks = wealth_index.cummax()
    drawdowns = (wealth_index - previous_peaks)/previous_peaks
    return pd.DataFrame({"Wealth": wealth_index,
                        "Previous Peak": previous_peaks,
                        "Drawdown": drawdowns})

def get_ffme_returns():
    """
    Load the Fama-French Dataset for the returns of the Top and Bottom Deciles by Market
    """
    me_m = pd.read_csv("data/Portfolios_Formed_on_ME_monthly_EW.csv",
                      header=0, index_col=0, na_values=-99.99)
    rets = me_m[['Lo 10', 'Hi 10']]
    rets.columns = ['SmallCap', 'LargeCap']
    rets = rets/100
    rets.index = pd.to_datetime(rets.index, format="%Y%m").to_period('M')
    return rets
```

```

1 import edhec_risk_kit_104 as erk
2
3 returns = erk.get_ffme_returns(DATAPATH)
4 returns.head()

```



	SmallCap	LargeCap
1926-07	-0.0145	0.0329
1926-08	0.0512	0.0370
1926-09	0.0093	0.0067
1926-10	-0.0484	-0.0243
1926-11	-0.0078	0.0270

```

1 import sys
2 print(f"sys.path:{sys.path}")
3 #print(f"PYTHONPATH:{PYTHONPATH}")

```

```

sys.path:['', '/env/python', '/usr/lib/python36.zip', '/usr/lib/python3.6', '/u

```



1