Python Programming
Module Search Path

Mostafa S. Ibrahim
Teaching, Training and Coaching since more than a decade!

Artificial Intelligence & Computer Vision Researcher
PhD from Simon Fraser University - Canada
Bachelor / Msc from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)

The Module Search Path

- When the interpreter executes import **ourlib**, it searches in order one of these 3 locations:
  - 1) The script's directory (e.g. <>/pycharm/latest for running <>/pycharm/latest/**file1.py**
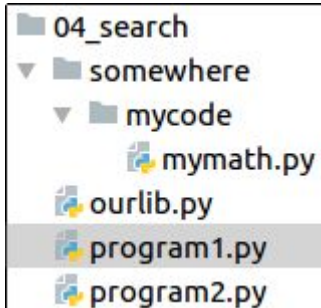   - Observe not the current directory (unless interactive session).

The Module Search Path

- When the interpreter executes import **ourlib**, it searches in order one of these 3 locations:
  - 2) list of directories contained in the PYTHONPATH
  - 3) installation-dependent list of directories
- Observe, these 3 things are actually what in **sys.path** *in order*

Missing module

- We have current directory: 04_search
- We have the following directory structure
  - 2 main modules: ourlib and and ourmath
    - Observe ourmath at: somewhere/mycode
  - 2 scripts program1, program2


```
ourlib.py ×
1    def sq(n):
2        return n*n
3
```



```
mymath.py ×
1    def sum1n(n):
2        return n * (n+1) // 2
3
```

```
04_search
▼  somewhere
  ▼  mycode
        mymath.py
    ourlib.py
    program1.py
    program2.py
```

ModuleNotFoundError

```python
import ourlib

print(ourlib.sq(5))        # 25


# ModuleNotFoundError: No module named 'mymath'
import mymath
print(mymath.sum1n(5))

# mymath is not visible
# neither on script directory (04_search)
# nor on pythonpath nor installation dirs!

# one way: let's add to sys.path
```
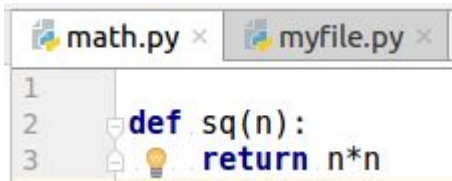
Now found!
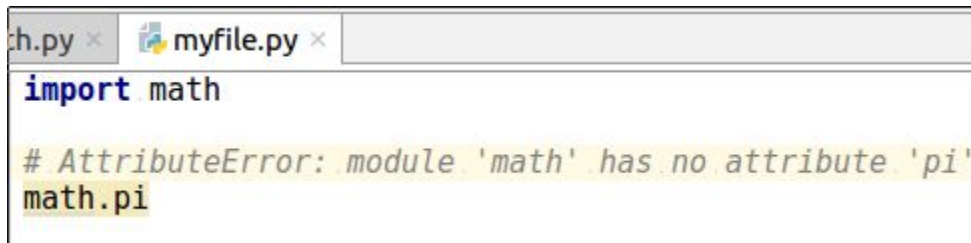
- We should construct path independent from OS (later)

```python
import ourlib

print(ourlib.sq(5))        # 25


import sys
sys.path.append('somewhere/mycode')     # bad / (linux) not windows

import mymath    # now is visible on the path
print(mymath.sum1n(5))        # 15


# Note: sometimes u want to add in the top of the list
```

Common mistake

- One of the very annoying common mistakes is to name your module **similar to built-in** or installed module.
- Then when you mean to import the built-in, it may actually importing yours (earlier on search path)
  - Tip: Think twice in your file naming.
- *Tip: Similar error if you merged several projects without caution*

```
math.py ×    myfile.py ×
1
2    def sq(n):
3        return n*n
```

```
h.py ×    myfile.py ×
import math

# AttributeError: module 'math' has no attribute 'pi'
math.pi
```

"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."