

COMP1531

2.4 - Python - Importing & Paths

importing and modules

- In python you're able to write code in one file, and import it into another file (just like C).

importing and modules

calmath.py

```
1 def daysIntoYear(month, day):
2     total = day
3     if month > 0:
4         total += 31
5     if month > 1:
6         total += 28
7     if month > 2:
8         total += 31
9     if month > 3:
10        total += 30
11    if month > 4:
12        total += 31
13    if month > 5:
14        total += 30
15    if month > 6:
16        total += 31
17    if month > 7:
18        total += 30
19    if month > 8:
20        total += 31
21    if month > 9:
22        total += 30
23    if month > 10:
24        total += 31
25    return total
26
27 def quickTest():
28     print(f"month 0, day 0 = {daysIntoYear(0,0)}")
29     print(f"month 11, day 31 = {daysIntoYear(11,31)}")
30
31 #if __name__ == '__main__':
32 #    quickTest()
33
34 quickTest()
```

importto.py

```
1 import sys
2
3 import calmath
4
5 if len(sys.argv) != 3:
6     print("Usage: importto.py month dayofmonth")
7 else:
8     print(calmath.daysIntoYear(int(sys.argv[1]), \
9                                int(sys.argv[2])))
```

What is this for??
(Live Demo)



Ways to import

use.py

```
1 import * from lib
2
3 # To do
4
5 from lib import one, two, three
6
7 # To do
8
9 import lib
10
11 # To do
```

lib.py

```
1 def one():
2     return 1
3
4 def two():
5     return 2
6
7 def three():
8     return 3
```

Which ways do we prefer and why?

pytest and paths

- Pathing with pytest is a big complicated. Because pytest is a program, it actually runs many python files in a container, which means that many different python files are run from different folders.
- This makes our assumptions around paths difficult...

"testpath" example

Live demo

Let's look at week 2 lecture code to learn more about importing, pytest, and paths

Python Path

This is something needed to make
pytest work

If your project is in ~/cs1531/project

```
1 export PYTHONPATH="$PYTHONPATH:~/cs1531/project"
```

You can add this line to your
~/.bashrc if you don't want to type it
in every time you open a terminal