Command-line Python

One Step in Python Language

Built-in Functions

Functions of Downloaded Packages

My Functions (later)

Functions of My Defined Modules

How We Can Talk to Python

As a Developer

- Command-line UI
- Graphical UI

As a User

- Command-line UI
- Graphical UI

First UI: Command-line

- Arithmetic operators
  - +, -, *, %, /

First UI: Command-line

- Arithmetic operators
- Built-in functions
  - Functions associated with modules
  - How to list functions, given a module?
  - Ex)

```
print()
str()
split()
int()
input()
format()
```

First UI: Command-line

- Arithmetic operators
- Built-in functions
  - Functions associated with modules
  - How to list functions, given a module?
- A package contains sub-packages
  - A package has multiple modules
    - A module has multiple sub-modules
    - A module has multiple functions
  - Ex) Install a package

    pip install datetime
  - Ex) Import a module from the package

    import datetime

`datetime.date.today()`

First UI: Command-line

- Arithmetic operators
- Built-in functions
  - Functions associated with modules
  - How to list functions, given a module?
- A package contains sub-packages
  - A package has multiple modules
    - A module has multiple sub-modules
    - A module has multiple functions
- Get the detail information from help() or from google

Another Useful Fn: `format()`

- Built-in Function
  - format(*value*, *format*)
  - Ex) format(0.78, '%')
  - Ex) format(123456789, ',')

- Check out the following site:

https://docs.python.org/3/library/functions.html

Yet Another format(): `string.format()`

- Method of Class String
  - str.format(*format_string*, **args*, ***kwargs*)
  - Ex) "My friends, {}, {}, and {}".format("James", "Tommy", "Sam")

Summary: format() v.s. string.format()

- Built-in Function
  - format(*value*, *format*)
  - Ex) format(0.78, '%')
  - Ex) format(123456789, ',')
  - Ex) format(3.148976543577, ‘.2f')

- Method of Class String
  - str.format(*format_string*, **args*, ***kwargs*)
  - Ex) "My friends, {}, {}, and {}".format("James", "Tommy", "Sam")

HW2

Consider one variable called smallNumber that contains a floating point, 12.23456789.

These two variables, smallNumber and myName will be manipulated as follows.

1) Print two variables as they are.

2) Print smallNumber to 12 using the function int().

3) Print smallNumber to 12.34 using the built-in function format().

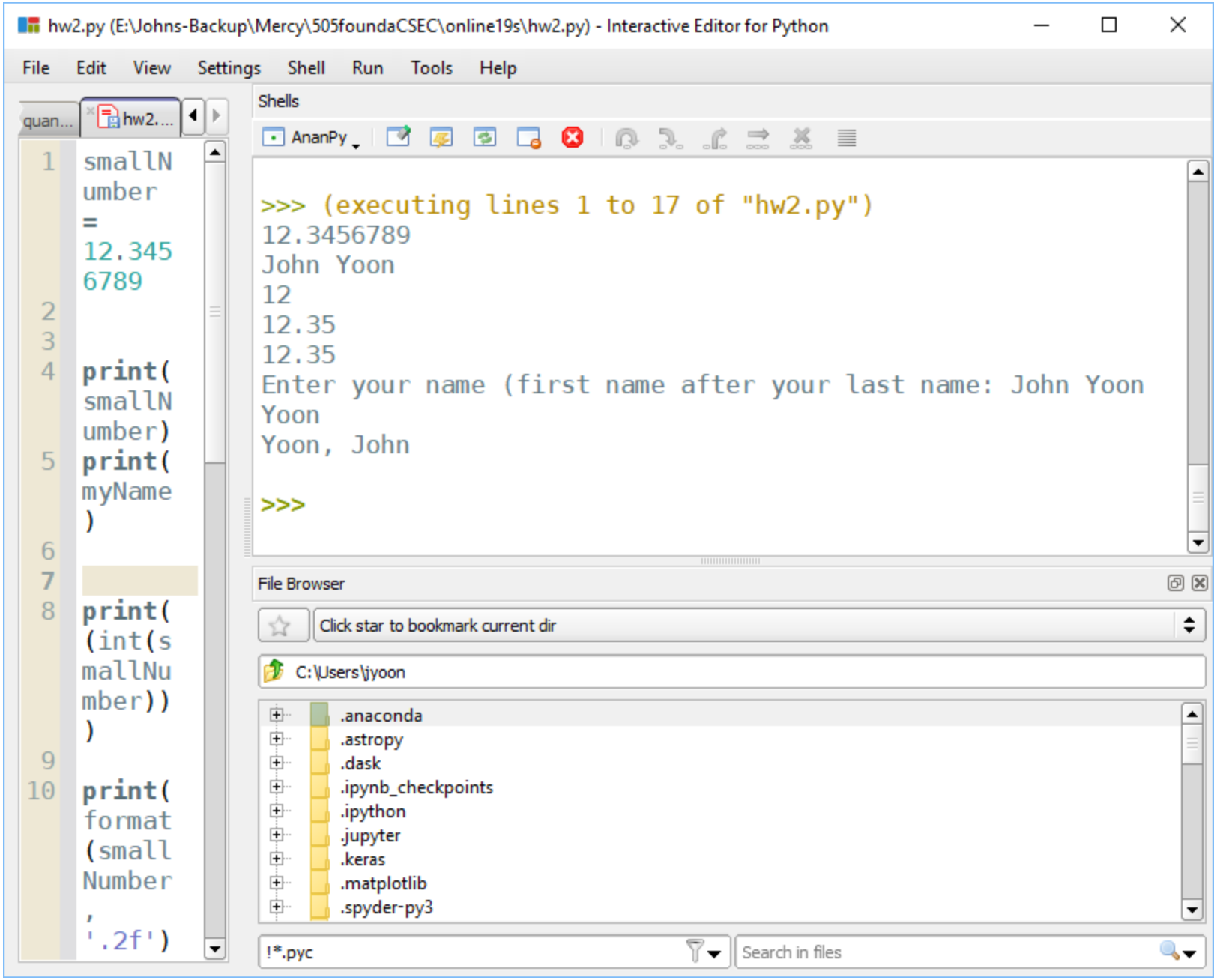
4) Print smallNumber to 12.34 using the string method format().

5) Prompt the question: “Enter your name (First name after the last name): “ Store the name in the variable called myName.

6) Split the variable myName to print the last name only using split().

5) Print

Note that any hard-coded print will receive 0 credit.


hw2.py (E:\Johns-Backup\Mercy\505foundaCSEC\online19s\hw2.py) - Interactive Editor for Python
File Edit View Settings Shell Run Tools Help
quan...
hw2....
1 smallNumber = 12.3456789
2
3
4 print(smallNumber)
5 print(myName)
6
7
8 print((int(smallNumber)))
9
10 print(format(smallNumber, '.2f'))
Shells
AnanPy
>>> (executing lines 1 to 17 of "hw2.py")
12.3456789
John Yoon
12
12.35
12.35
Enter your name (first name after your last name: John Yoon
Yoon
Yoon, John
>>>
File Browser
Click star to bookmark current dir
C:\Users\jyoon
.anaconda
.astropy
.dask
.ipynb_checkpoints
.ipython
.jupyter
.keras
.matplotlib
.spyder-py3
!*.pyc
Search in files