

Python Basics

(adapted from Quantecon)

Basics

Exercise 1 Run the following code in the python interpreter:

```
def say_hello(name):  
    """This function prints morning greetings"""  
  
    print(f"Good morning {name}!\n")  
  
    # we can import libraries  
    import datetime  
    t = datetime.datetime.now()  
  
    # blocks are defined by indentation and colons  
    if (t.hour,t.min) <= (9,15):  
        print("All good?\n")  
    else:  
        print("Time to get started?\n")  
  
say_hello("Pablo")
```

Exercise 2 What do you think the value of `z` is after running the code below?

```
z = 3  
z = z + 4  
print("z is", z)
```

z is 7

```
# your response there
```

Exercise 3 Read about what the `len` function does (by writing `len?`).

What will it produce if we give it the variable `x`?

Check whether you were right by running the code `len(x)`.

```
# your code here
```

Exercise 4 We can use our introspection skills to investigate a package's contents.

In the cell below, use tab completion to find a function from the `time` module that will display the **local** time.

Use `time.FUNC_NAME?` (where `FUNC_NAME` is replaced with the function you found) to see information about that function and then call the function.

Look for something to do with the word `local`

```
import time  
# your code here
```

Exercise 5 The code below is invalid Python code (once uncommented)

```
# x = 'What's wrong with this string'
```

Collections

Exercise 6 In the first cell, try `y.append(z)`.

In the second cell try `y.extend(z)`.

Explain the behavior.

When you are trying to explain use `y.append?` and `y.extend?` to see a description of what these methods are supposed to do.

```
y = ["a", "b", "c"]  
z = [1, 2, 3]  
#
```

```
y = ["a", "b", "c"]
z = [1, 2, 3]
#
```

Exercise 7 Verify that tuples are indeed immutable by attempting the following:

- Changing the first element of `t` to be 100
- Appending a new element `!!!` to the end of `t` (remember with a list `x` we would use `x.append("!!!")` to do this
- Sorting `t`
- Reversing `t`

```
t = (1,2,3,4)
```

Exercise 8 Look at the [World Factbook for Australia](#) and create a dictionary with data containing the following types: float, string, integer, list, and dict. Choose any data you wish.

To confirm, you should have a dictionary that you identified via a key.

```
# your code here
```

Exercise 9 Use Jupyter's help facilities to learn how to use the `pop` method to remove the key `"irrigated_land"` (and its value) from the dict.

```
# uncomment and use the Inspector or ?
#china_data.pop()
```

Exercise 10 Explain what happens to the value you popped.

Experiment with calling `pop` twice.

```
# your code here
```

Control

Exercise 11 Run the following two variations on the code with only a single change in the indentation.

After, modify the x to print 3 and then 2, 3 instead.

```
x = 1

if x > 0:
    print("1")
    print("2")
print("3")
```

```
1
2
3
```

```
x = 1

if x > 0:
    print("1")
print("2") # changed the indentation
print("3")
```

```
1
2
3
```

Exercise 12 Write a for loop that uses the lists of cities and states below to print the same “{city} is in {state}” using a `zip` instead of an `enumerate`.

```
cities = ["Phoenix", "Austin", "San Diego", "New York"]
states = ["Arizona", "Texas", "California", "New York"]
```

```
for i,c in enumerate(cities):
    print(c, " : ", states[i])
```

```
Phoenix : Arizona
Austin : Texas
San Diego : California
New York : New York
```

```
# your code here
```