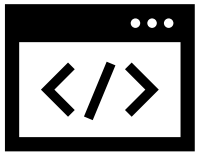
Imperial College London



BIOE40002 – Computer Fundamentals and Programming 1

Part II – Programming 1, Lab 3

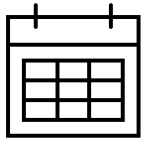
Binghuan Webster Li | *Department of Bioengineering*

binghuan.li19@imperial.ac.uk

March 17, 2022

Meme of the day…

Imperial College London



Today's Schedule

Imperial College London

- **Recap (~ 10 mins)**
  - *Functions*
  - *Recursion*
  - *Useful libraries*

- **Lab work**

Functions

Imperial College London

'Define' *Function name* *Arguments*

```
def function_name(arg1, arg2, arg3, …argN):

    statement1
    statement2
    …
    statementN

    return variable1, variable2, …, variableN
```

Your statements

Return variables from the function

- ***Argument:*** data send into the function. It is optional.
- ***Return:*** data send out the from function. It is optional.

Q: Why do we need arguments to pass data into the function, and use 'return' to get the data from the function? Scopes!

Recursion

Imperial College London

- A recursion / recursive function is a function that calls itself.

```
def message(times):

    if times > 0 :
        print("this is a recursive function.")
        message(times-1)
```

You need a condition to 'limit' your recursion, otherwise, this will repeat forever!

call function within the function definition itself

- Sometimes, an alternative to recursion is **looping**. Generally, recursion is less efficient than loops.

```
for i in range(0,4):
    print("this is a recursive function.")
```

Libraries and modules

Imperial College London

- A Library (or, module) is a file containing a set of functions you want to include in your application.
- `import` statement:

```
import math

x = math.sqrt(25)

print(x)
```

Usage	Related modules
Machine learning	PyTorch, TensorFlow, Karas
Scientific Computing	NumPy, SciPy, SimPy
Data science	NumPy, Pandas, Matplotlib
Web Development	Django, Flask
App Development	tkinter, PyQt
Game Development	Archade, PyGame
Your need?	**Your module!**

`import` statement

Imperial College London

1. Regular `import` statement

```
import math
```

2. Importing **a specific function** or class

```
from math import sqrt
```

3. Wildcard imports

```
from math import *
```

   - Load the entire contents of the module
   - Does not require you to use qualifies name of the items in the module

```
x = math.sqrt(25)
```

```
x = sqrt(25)
```

4. Using an **alias**

```
import math as mt
```

Installing modules with `pip`

Imperial College London

- Python Package Index (*PyPI*): A large collection of third-party modules exists at the website pypi.python.org.
- The easiest way to download and install a package is with the `pip` utility.

In your *Windows* **Command Prompt**	In your *Mac/Linux* **Terminal**
`pip install` *`package_name`*	**`sudo pip3 install`** *`package_name`*

- To check your library information: **`pip show`** *`package_name`*

Imperial College London

Questions?

That's it for now.

You can now proceed to the Lab 3 exercises.