

# ENGG1003 - Monday Week 2

First steps: libraries & modules, printing and plotting

Steve Weller

University of Newcastle

February 26, 2021

# Lecture overview

- ① Python program with a library function §1.3
  - ▶ principles
  - ▶ live demo
- ② importing from modules and packages §1.4
  - ▶ principles
  - ▶ live demo
- ③ simple plotting §1.5
  - ▶ principles
  - ▶ live demo
- ④ plotting, printing and input data §1.6
  - ▶ principles
  - ▶ live demo

# 1) Python program with a library function

- describe the problem
- simple diagram:  $x, y, \theta$
- maybe a ball?
- algorithm is  $\tan^{-1}$

# The program

```
x = 10.0           # Horizontal position
y = 10.0           # Vertical position

angle = atan(y/x)

print((angle/pi)*180)
```

ball\_angle\_first\_try.py

# First use of a Python function

- first use of a *function*, in this case `atan`
- *argument*
- *return value*

# Math review: radians and degrees

- Python's `atan` returns value in radians
- $\times \frac{180}{\pi}$  to get answer in degrees

# Running the program

- screen grab from PyCharm – error message

# Python standard library and import

- Python has plenty of functionality “built-in”
- LOTS more can be *imported*
- `atan` and other trigonometric functions not built in
- to activate that functionality, must explicitly import
- `atan` function is grouped together with many other mathematical functions in a *library module* called `math`

```
from math import atan, pi
```



# The program: second attempt

```
from math import atan, pi

x = 10.0          # Horizontal position
y = 10.0          # Vertical position

angle = atan(y/x)

print((angle/pi)*180)
```

ball\_angle.py

- script correctly produces 45.0 as output
- live demo in PyCharm shortly

# Another way of importing

- use the import statement `import math`, but require `atan` and `pi` to be *prefixed* with `math`
- both techniques are commonly used and are the two basic ways of importing library code in Python

```
import math

x = 10.0           # Horizontal position
y = 10.0           # Vertical position

angle = math.atan(y/x)

print (angle/math.pi)*180
```

ball\_angle\_prefix.py

# Live demo of Python program with a library function

blah

## 2) Importing from modules and packages

motivation and context

- (a) importing for use **without** prefix
- (b) importing for use **with** prefix

# Importing for use *without* prefix

```
from math import atan, pi

x = 10.0           # Horizontal position
y = 10.0           # Vertical position

angle = atan(y/x)

print((angle/pi)*180)
```

- ✓ Python code is easier to read
- ✗ allows name conflicts!

# Name conflicts

- explain the basic idea
- do *not* explain example from text, which is too complicated
- will show an example shortly

# Importing for use *with* prefix

```
import math

x = 10.0           # Horizontal position
y = 10.0           # Vertical position

angle = math.atan(y/x)

print (angle/math.pi)*180
```

- ✗ Python code is a little harder for humans to read
- ✓✓ eliminates name conflicts!
- **standard and safer and preferred method of importing**

# Avoiding name conflict using prefixes

```
import numpy
import math

x = numpy.exp([0, 1, 2])          # do all 3 calculations
print(x)                         # print all 3 results

y = math.cos(0)
print(y)
```

- `numpy` library includes an `exp` function
    - ▶ math review: exponential function  $e^z = \exp(z)$
  - `math` library also includes an `exp` function—with a different implementation!
- ✓ **prefixes make clear which `exp` to use**



# Imports with name change

```
import numpy as np
import math as m

x = np.exp([0, 1, 2])          # do all 3 calculations
print(x)                      # print all 3 results

y = m.cos(0)
print(y)
```

- using **as**, numpy name becomes np
- similar for math and m
- ✓ Python code is easy to read
- ✓✓ eliminates name conflicts

# Main modules used in ENGG1003

- `math`—description
- `numpy`—description
- `matplotlib`—description

# Live demo of importing from modules and packages

blah

### 3) Simple plotting

blah

# Live demo of simple plotting

blah

## 4) Plotting, printing and input data

blah

# Live demo of plotting, printing and input data

blah