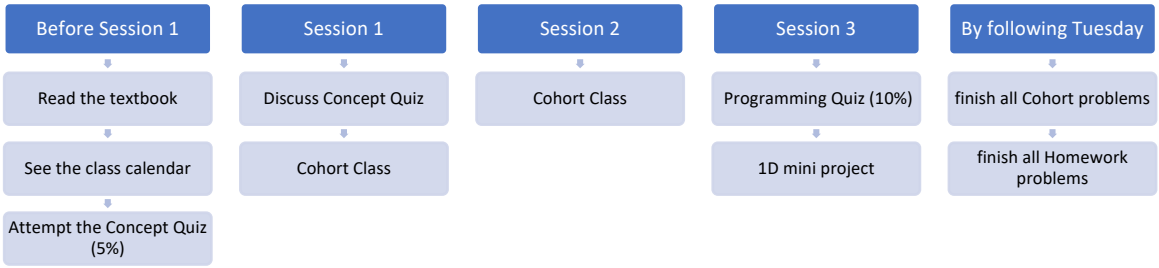
Lesson 2

How it works every week (in general)



Refer to our class calendar

Expectations

Students must read materials before coming to class.

Students will come early for class.

Students will participate actively in class.

Students will submit their own works for all assignments.

If you have questions

Posting on Piazza is preferred as everyone will get to see your question

Click on Information->10.009 Piazza

Three sets of help session are available!

Admin Matters

Homework

- Cohort / Homework problems begin this week and submission is via eDimension.
- We'll demonstrate how to do it.
- You press "Run" to check if your code meets the test cases. You can click this as many times as you like.
- You only have **one chance** to submit.
- Note the deadlines please.

Digital World + Chemistry Combined Assignment

- Please access the wikispaces page.
- Solve the problems progressively.
- You can work in groups but you have to submit individually.
- Please see our course webpage for more information.

1D Project

- IF you haven't done so, please form groups of five and enter your details at the 1D project page by 30 January 2018.

Pre-Reading

- Read Week 3 materials before Session 1. I encourage you to post your queries on Piazza.

Recall Last Week's Lesson

True/False

1. In Python 3, the result of `10/3` is `3.333`.

2. The result of `10%3` is `1`.

3. `o_O` is a valid variable name.

4. `lambda` is a valid variable name.

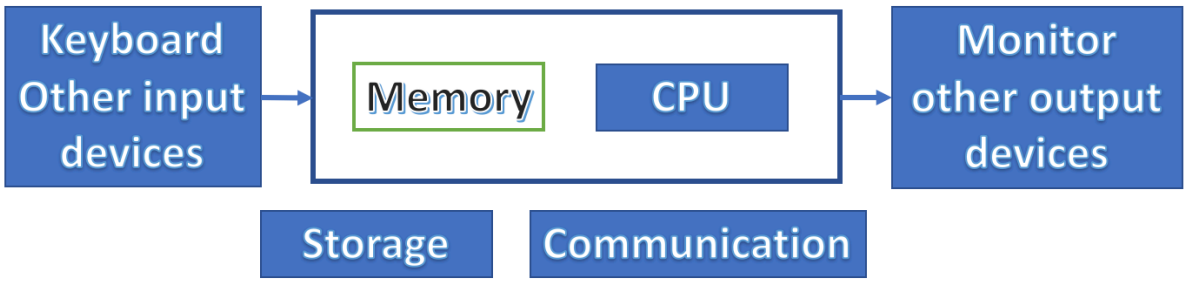
5. `dy/dx` is a valid variable name.

6. The result of the following is `12.0`.

```
foo = 10
foo += 2 # foo =foo + 2
print foo
```

Memory

Model of a computer



The Python Memory Story (Part 1)

The `id()` function tells you the address in memory where the object resides at. Consider the following code. What has happened in the memory?

```
x = 12.3
y = x
print( id(x), id(y) )
```

Now add the following statements. What has happened in the memory?

```
y = 495
print( id(x), id(y) )
```

Further reading at this link.

The meaning of Object

Throughout the lessons, we'll slowly unfold the idea of **Object.**

It refers to any datum/data in memory. Each object has an **address** in memory. As with the custom data-types that you created, it can have more than one variable associated with it.

Python Keywords

A list of python keywords is given here at this link.

To make sense of this list, you can put some of them in the following categories:

Functions	Logical	Decision Making	Loops

Python Functions

Why functions?

- Break a programming task up into smaller components
- Allow each component to be reused in other programs
- Easier for testing and to create mistakes

The Python Memory Story (Part 2)

When you define a function, a separate “zone” in memory is created.

When you call (i.e. execute) the function, the variable names defined in the function belong only to the **scope** of the function.

In other words, you cannot access them outside the function.

They are destroyed when the function terminates at the return statement.

Clicker Question 1 – What is the output? (Try without typing in)

```
19 a = 20
20 print ( my_function(a) )
21
22 def my_function(b):
23     b *= 3
24     return b
```

A. 3
B. 20
C. 60
D. Error
E. None

Clicker Question 2 – what is the output? (try without typing in)

```
27 def next_function(a):
28     a = 8
29     return a
30
31 a = 20
32 print( next_function(a) )
```

A. 8
B. 20
C. Error, because a cannot be assigned in two places
D. None
E. 0

Correct the mistakes (at least 4 mistakes)

```
1
2
3 def calculate_hypotenuse_mistakes(a,b)
4     ''' takes in two sides of a right-angled triangle a and b
5     and returns the hypotenuse
6     '''
7
8 hypotenuse = sqrt(a*a + b*b)
9
10
11
12 side1 = 3
13 side2 = 4
14
15 print calculate_hypotenuse_mistakes(side1,side2)
16
17
```

Consider:

1. Does the function `calculate_hypotenuse_mistakes` display the result of the calculation on the screen?

2. How is data passed from `calculate_hypotenuse_mistakes` to `print`?

Writing Readable Code

Use comment statements to describe parts of your code

Give variables and functions meaningful names: `area_of_rectangle` and not `xx`

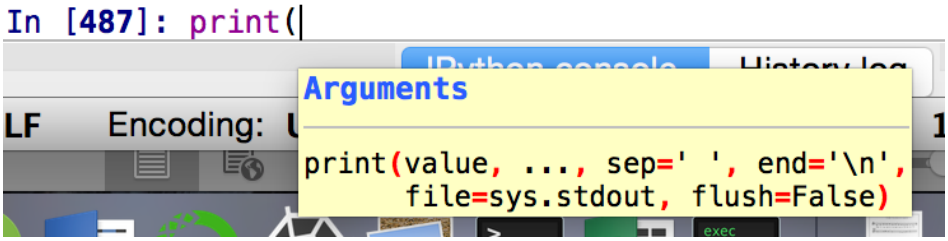
Python programmers prefer using underscore to separate words.

Java programmers prefer using Camel Case: `areaOfRectangle`

Print Statements

The print statement helps you to output messages to the computer's screen.

Here, you notice that Spyder gives you hints on what inputs a function should contain.



The **escape sequences:** `\n` and `\t`. What do they do?

Please read the python documentation for details!

Example 1. You may treat separate parts of your string as different inputs to the print function. Recall that a comma separates the different inputs.

```
number_animals = 3

type_animals = "chickens"

location = "zoo"

print("there are" , number_animals, type_animals, "in the",
location)
```

Example 2. You may use placeholders in your string and use the format operator to specify their contents.
Try yourself: change the positions of 0, 1 and 2 and see what happens.

```
#Example 2

print("there are {0} {1} in the
{2}".format(number_animals,type_animals,location))
```

Example 3. In the placeholders, you may use keywords instead.

```
print("there are {number} {animal} in the
{venue}".format(number=3,animal="chickens",venue=location))
```

Example 4. You may format your string separately and then print the string as a separate command. This example also shows you what the end does.

```
my_string = "there are {0} {1} in the
{2}".format(number_animals,type_animals,location)

print(my_string, end=" ")

print("next string")
```

Example 5. For each placeholder, you could specify the formatting and the data type.
What does the > operator do (change it to <)?
What do d, s and f mean?

```
unit = 100

currency = "JPY"

rate = 1.183

money_changer = "we sell {0:6d} {1:>10s} at {2:6.3f}
SGD".format(unit, currency, rate)

print(money_changer)
```

Example 6. Following example 5, if you are used to C programming, you may use the old style as well. Change the + to – and see what happens.

```
change_alley = "we sell %6d %+10s at %3.4f SGD \n" %(unit,
currency, rate)

print(change_alley)
```

Try the following on your own

(a) Explain the result of the following:

```
print(1e200*1e120)
```

(b) Explain the result of the following:

```
1/3 + 1/3 + 1/3 + 1/3 + 1/3 + 1/3
```

(c) List the different ways to specify a string.

Introduction to for-loops

You would like the following set of statements to help you with your currency conversion in your next trip.

```
"100 JPY is 1.19 SGD”
“200 JPY is 2.38 SGD”
“300 JPY is 3.57 SGD”
```

With what you know, you can write the following statements.

```
yen = 100
sgd = 1.19
print("{0} JPY is {1} SGD".format(yen, sgd)
print("{0} JPY is {1} SGD".format(2*yen, 2*sgd)
print("{0} JPY is {1} SGD".format(3*yen, 3*sgd)
```

If you are going to print up to 2000 yen, would you like to have 20 lines?
You can improve your productivity by using a for-loop with the range function.

```
yen = 100
sgd = 1.19
for i in range(1,21):
     print("{0} JPY is {1} SGD".format(i*yen, i*sgd)
```

`range(1,21)` gives you a sequence of integers beginning with 1 and ending with **20**. (It’s actually more complicated, but this is good enough for now).

`for i in range(1,21)` thus assigns the integers given by the range function sequentially to i at each iteration

Problem Solving

Write a program that takes in a three-digit integer from the keyboard and returns its individual digits.

PCDIT Framework

Problem Statement: Input – Output – Process

Test **C**ases - Generate different inputs and work out the outputs.

Design Algorithm – write down the steps in English

Implement

Testing – test your code with the test cases that you wrote down.

Something to think about …

Write a program that takes in parameters a b c of a quadratic equation and returns the two roots of the equation. $y = ax^2 + bx + c$

- **First Stage**. You can assume that inputs a, b and c will only give real roots.
- **Second stage**. Suppose a,b,c are such that you get complex roots, return None and print a message.

1D Mini Project

Some unix terminal commands

List the files in the current directory: `ls -l`

Change directory to /home/pi/Desktop: `cd /home/pi/Desktop`
(Press Tab after typing `cd /h`)

Print the current working directory: `pwd`

Python3

To get the Python3 console:
Raspberry Icon->Programming->Python3(IDLE)

To get the Python code editor:
File->New File or Open

The documentation for Thymio is found on our Wikispaces site, under courseware.

Having problems with the single apostrophe key?

Launch the terminal and type

```
sudo nano /etc/default/keyboard
```

Modify what you see and chances are you have to add

```
XKBMODEL="pc105"
XKBLAYOUT="us"
XKBVARIANT="intl,nodeadkeys"
XKBOPTIONS=""
```

Press Ctrl+X to exit.