1. Introduction

Python is a high level language that is used in many areas: YouTube, NASA and Google. It is an interpreted language (meaning that mistakes can be spotted very quickly). Guido Van Rossum (the developer) was a great fan of Monty Python's Flying Circus and so this is where the name, "Python" originated. It is open source meaning that it is free to download and use.

2. Python: Loading, Saving, Running

To Load

Start | Programs | Computing | Python 2.7 rm 12b only | IDLE (Python GUI)

File | New Window

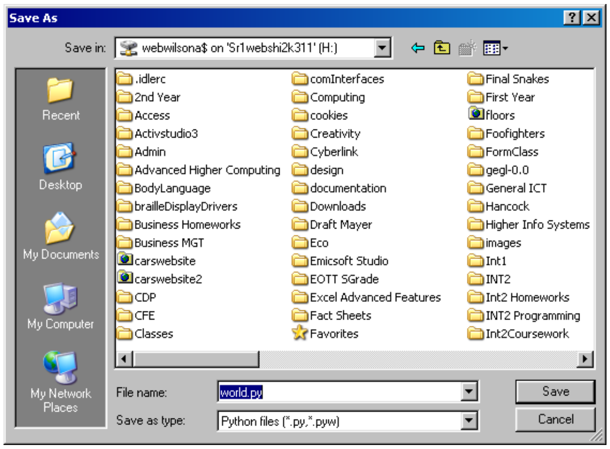
Type in the following line of code:

```
print "Hello, world!"
```

To Save

Now go to **File | Save as...**

The following window appears:


Save As

Save in: webwilsona$ on 'Sr1webshi2k311' (H:)

File name: world.py

Save as type: Python files (*.py,*.pyw)

Give the program a name followed by the extension **.py**

Make sure you save it in a sensible place

To Run

Run | Run Module (or press **F5**)

The program will play in a separate window.

3. Python: Displaying

Type in:

```
print "Hello world!" * 2

print "Goodbye!"
```

Add the space:

```
print "Hello world! " * 2

print "Goodbye!"
```

Now type in:

```
print "Hello world!\n" * 2
```

So what does the **\n** do?

4. Python: Numbers

Numbers in quotes are treated like text. Without the quotes you can do maths.
Type in this:

```
print "1" + "1"
```

now type:

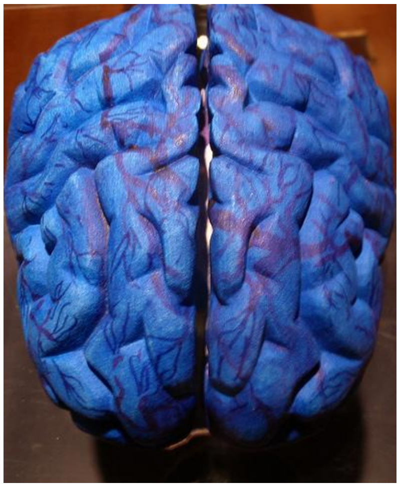
```
print 1 + 1
```

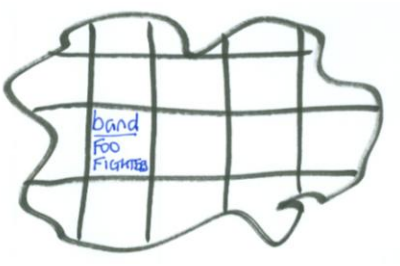
Type in this program:

```
print "2 plus 2 =", 2 + 2

print "30 divided by 3 =", 30 / 3

print "80 subtract 12 =", 80-12

print "100 times 4 =", 100 * 4
```

5. Python: Variables

When someone tells us something we remember it! Every fact we are told is kept in a separate brain cell. We have millions of brain cells so how do we know what each brain cell contains? Each one is named.





Friend: What is your favourite band?

Me: My favourite band is the, Foo Fighters

How does the computer do it?

```
band = raw_input (“What is your favourite band?”)
```

That’s text going in to memory dealt with – how do we deal with numbers and memory?

```
number1 = input("enter your first number")

number2 = input("enter your second number")

print number1+number2
```

If we ran this with 8 and 7 we would get 15

Int is short for integer. Integers are numbers that do not have decimal points in them.

34 is an integer (a whole number)

34.0 is a float – a number with a decimal point even if it just has 0 after it. They are more precise.

Tasks

1. A baseball team statistician wants a program that will ask for a player’s team, player’s name and points scored in each of the first three games of the season. It should then calculate the player’s average score and display a summary of the details as follows:

```
NORTH AMERICAN BASEBALL LEAGUE
******************************

PLAYER TEAM: New York Mets

PLAYER NAME:  Mike Nickeas

GAME 1: 1

GAME 2: 2

GAME 3: 1

AVERAGE: 1.3
```

6. Python: Commenting

Comments are notes included in a program. They are useful for programmers. The computer ignores them.

Comments should tell what the program is about, who wrote it and the date it was last modified.

A comment always begins with the #

```
#Program that asks for a name
#Written by Mr Wilson
#Date Last Modified 07/07/11

name = raw_input("enter your name")
print name
```

7. Python: If Statements

An if statement is an example of a ***conditional*** statement. This gives the computer the ability to check what has been entered, decide if it is true or false and then run the appropriate program instructions.

```
guess = input("Choose a number:")
if guess == 8:
    print "You win"
else:
    print "You lose"
```

You could add more expressions to compare:

```
guess = input("Choose a number”)
if guess == 8:
    print "You win"
elif guess >8:
    print "Your guess is too high"
else:
    print "Your guess is too low"
```

== means if the value in guess is equal to 8. It is not an assignment statement ie give guess the value 8.

!= Not Equal to   > Greater than.   < Less than.

>= Greater than or Equal to.   <= Less than or Equal to.

Tasks

1. Type in the above
2. Write a program that asks for three shopping bills. If the shopping bill comes to over £150 a message should appear saying that you get a discount.
3. Draw a structure chart, write the steps and stepwise refinement.
4. Write a program that asks for two numbers. If the sum of the numbers is greater than 100, print "That is a big number."
5. Write a program that asks the user his/ or her name. When that name is entered say "That is a nice name".   If they enter "John Cleese" or "Michael Palin", tell them how you feel about them ;), otherwise tell them "You have a nice name."
6. Write a program that asks the user to enter a password. If the password is correct, a message saying “You have successfully logged in” should appear on the screen. Otherwise a message “Sorry the password is wrong should appear.”

8. Python: For Loops

This is a fixed loop.

Loops run a block of code a certain amount of times without you having to type it out time after time.

```
for counter in range(5):
    print(counter+1)
```

Another example:

```
for counter in range(5):
    counter = counter + 1
    if counter % 2:
        print(counter, "is odd")
    else:
        print(counter, "is even")
```

Tasks

1. Type in and understand this program:

```
for counter in range(10):
    print 3,"times", counter,"=",3*counter
```

9. Cumulative Totals

```
total = 0
for counter in range(5):
    score=input("enter the score")
    total=total+score
print total
```

10. Python: While Loops

These run until a defined condition is met (conditional loop). Look at this code:

```
secret_answer = "reddog"
secret_question = raw_input("Enter your answer: ")
if secret_answer != secret_question:
        secret_question = raw_input("Enter secret answer: ")
print "secret answer accepted"
```

This program will only work once. If we want it to keep asking until you get the right answer you would add a while loop

```
secret_answer = "s3cr3t"
secret_question = raw_input("Enter secret answer: ")
if secret_answer != secret_question:
    while secret_answer != secret_question:
        secret_question = raw_input("Enter secret answer: ")
print "secret answer accepted"
```

Tasks

1. Examine the code below:

```
choice = 1
while choice != 0:
        print "************MENU************"  #Make a menu
        print "1. Add Two Numbers"
        print "2. Find perimeter and area of a rectangle"
        print "0. Exit!"
        print "*" * 28
```

```
choice = input("Please make a selection: ") #Prompt user for a selection
if choice == 1: #If choice is 1, get input and calculate
    firstnumber = input("Enter 1st number: ")
    secondnumber = input("Enter 2nd number: ")
    add = firstnumber + secondnumber
    print firstnumber, "added to", secondnumber, "equals", add #show results
```

2. Enter and run the above program (save it as calculate.py)
3. What does **!=** mean?
4. What does the # symbol allow you to do?
5. Add a block of code that deals with the second option above.
6. Add a block of code that would allow you to work out the area of a triangle.
7. Add another block of code that would allow you to work out the area of circle – make sure it is commented.
8. Add another block of code that will take in a measurement in bits and convert it in to bytes, kilobytes and megabytes.
9. Do your stepwise refinement and structure charts for your entire system.

11. Arrays (Lists)

A list is simply a structure where lots of data can be held in one memory cell.

In order to tell the computer we are going to make a list, we give the list a name followed by the equals sign and a pair of square brackets. So for instance if we were going to make a list about trees we would say:

```
trees =[]
```

An example of an array with three elements would be:

```
Trees = [birch,beech,oak]
```

To identify an element, python behind the scenes, indexes the list, but the index begins at zero.

Trees = [birch,beech,oak]

0	1	2

To display the second element of the array (list)

```
cars=[]

for counter in range(3):

    make = raw_input('Enter car make ')

    cars.append(make)

print cars
```