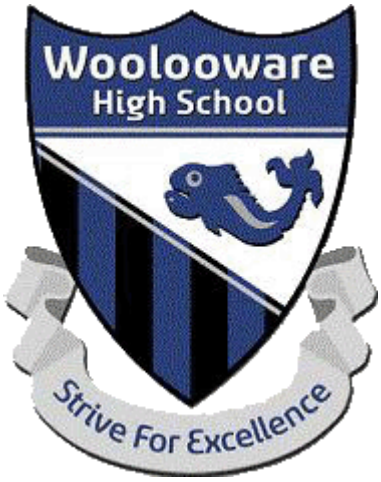


Stage 3 to 4
Technology
Transition
Program
2016-2017

Python 3 Workbook

Student Work Booklet

Contents

1.1 Meeting Python

1.1a What and where is Python?

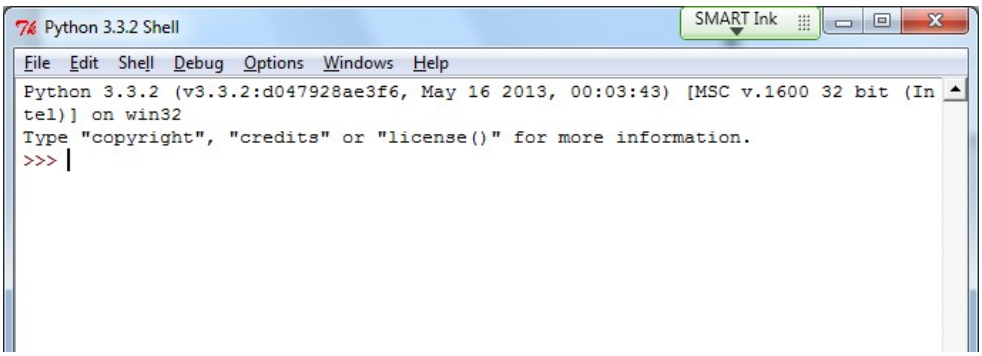
Welcome to your study of the computer science with Python. Python is a freely available programming language. You will need to install the Python compiler at home in order to make your own programs, although you don't need any special software to WRITE Python programs. You can do this at home if you wish to, from the website www.python.org. In school Python can be found in the start menu. If you own a Mac, then Python is already installed.

Code samples and Github

A collection of code sample is called a code repository or "repo". The "repo" for this course is called "**p3w**" (short for "Python 3 Workbook"). The repo author is davidhawkes11 so you can search by author too.
This can be found at www.github.com.
Create an account for yourself at github with your education email address.
Use the account name "firstnamesurname" (no dot) plus a number if there is a number after your name.
The Python code is all there if you need to check it later.

1.1b – What is IDLE? (Interactive Development and Learning Environment)

You can write Python code in notepad, or any other simple text editors - but it is easier if you use a program like IDLE (which comes with the standard Python installation) because it helps with things like indentation and debugging (things we will cover later). We will be using IDLE throughout most of the course. IDLE Looks a little like this. If you see the three 'greater than' signs, then you are in the interactive shell prompt (we will call this the 'shell'). You can type Python instructions at the interactive prompt. The computer will interpret these and follow your instructions.



You can type simple commands at the interactive prompt (>>>). Try typing:

```
2+2
17-9
16/4
3*7
```

Check that these answers are correct! You have just discovered how to use simple mathematics with Python. Well done!

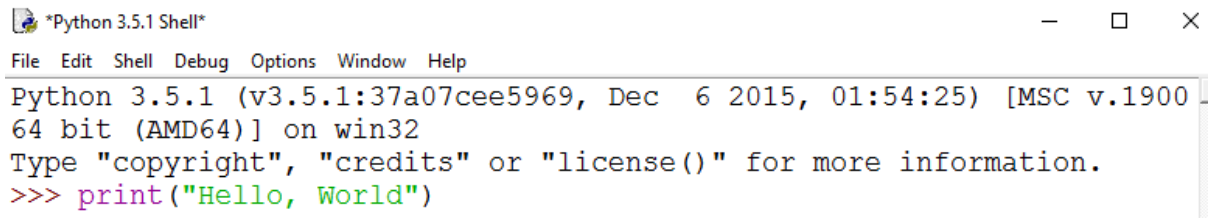
1.2 Writing Your First Program

Quick note: you are going to be making lots of Python programs. Create a Python folder in your ICT Folder in your S://StudentFolder.

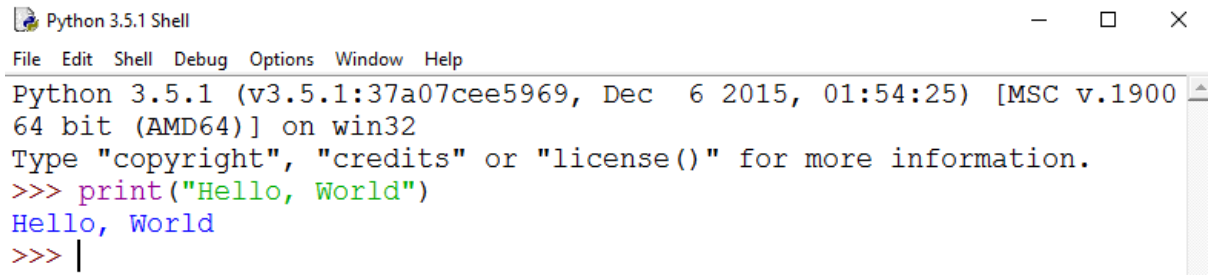
1.2a - Hello World

We can put simple instructions into the IDLE window, but we can't write and save complex programs. Open IDLE and click 'File >> New' - this will give you a new window that you can use to write and save your programs!
In this new window, type the following (exactly, lower case print):

```
print ("Hello, World")
```



Press enter after typing this. The words you have selected should be displayed.



Congratulations! You have written your first Python program! You can run your program on any computer with Python installed, no matter what kind of computer it is.

1.2b – Student Activity (Python Basics)

The following questions should help you get a basic understanding of the shell.

When you have got the code working, write down the answer (the code) under the questions below.

1. Write a short program that prints your name to the python shell.

2. Create code that will show you know how to perform the following mathematical operators in Python.

    a. Addition

    b. Subtraction

    c. Multiplication

    d. Division

3. Try using the rules of BIDMAS to create a complex mathematical code.

CHALLENGE – Try and create code that prints both words and numbers on the Python shell at the same time, on the same line. Write your code below.

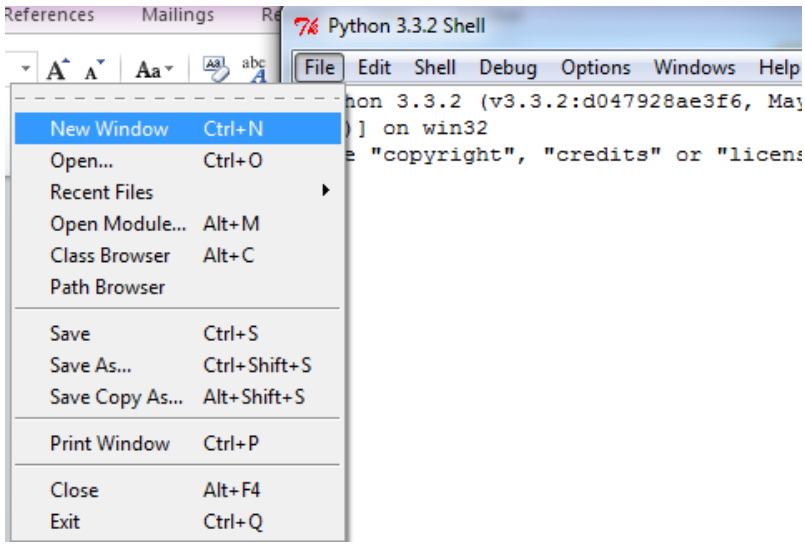
Congratulations, you have just created your very own code, from scratch. Well done you!

2.0 Naming and Using Variables

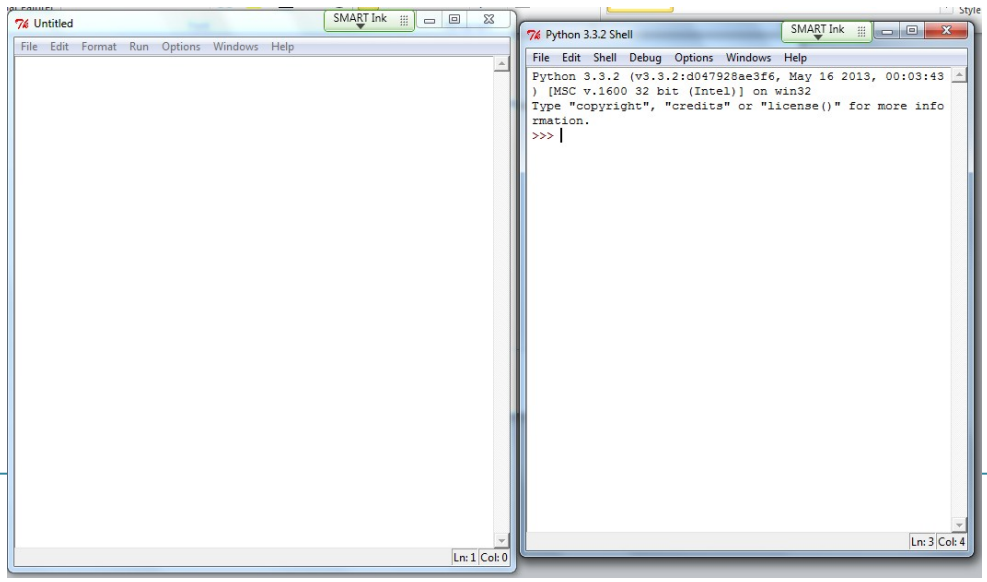
Variables are a really useful part of programming and they allow us create a really important element, user input.

> **Variable:** a name given to a piece of data that can be accessed or changed by the program. In computer science, a variable is a data item that may take on more than one value during the runtime of a program.

If you want to store a number or some text for use later on, you do this using a variable. Using Python, we can this very simply. To do this, we need to be in "Script Mode". You will need to click on File in the Python shell and then "New Window".



This will bring up a brand new window alongside your shell window. You will need BOTH of these windows to work in script mode.



2.0a – Experiment

In the scripting window (the one that's not the shell), enter the following code. Save it in your Python code folder as 1.2a Experiment.

```
name = input("What is your name?\n")
print ("Hello", name)
```

Be sure to click "Run > Run Module" or Press the F5 key to run it from the script window.

What happens when you run the code? HINT – Look at your shell window!

What is the **VARIABLE** in this element of code?

What do you think the \n does in the code above?

Congratulations, you have just created your very own variable, from scratch. Well done you! I'm very impressed!

2.1b – Student Activity (Using Variables)

Using the scripting window carry out the following tasks. Be sure to SAVE each task in your Python coding folder.

1. Create code to accept two variables. First name and last name. You then need to print these two variables together on the same line. HINT – firstname & lastname could be your two variables.

2. From your previous naming code. Try to add the users age to the program, display this after the name

We now need to see what you can do using some variables. Let's see how well you can do on your own.

3. Allow the user to enter two numbers, then times these numbers together and show the result. HINT – Remember * preforms the times function.

4. Using two or more numbers, create code to work out how to convert Celsius into Fahrenheit. HINT – The formula for converting Celsius to Fahrenheit
   is °C x $9/5 + 32$ = °F

CHALLENGE – Try to write code the will allow you work out the percentage of the cost of a given product. You will need to enter the price for the product as well as how much percentage discount you want to remove. Write your code below.

Congratulations, you are now a master at using variables and controlling code while using them. Well done you!

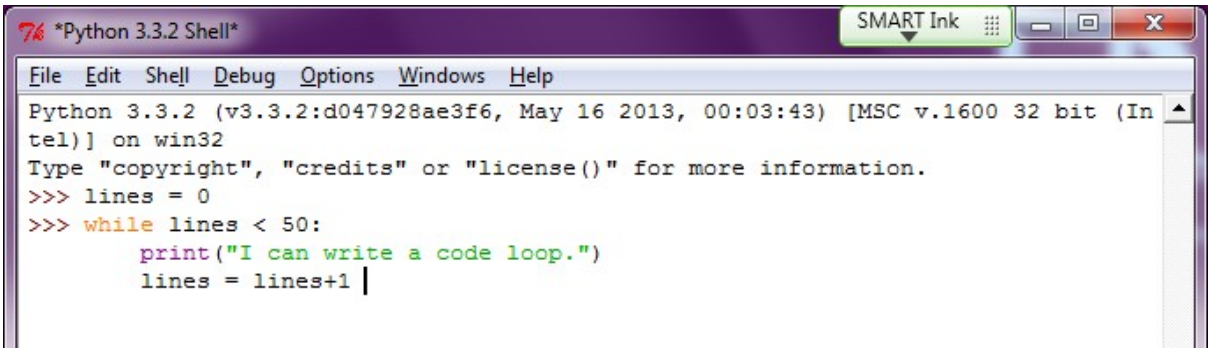
3.0 Getting Loopy

3.0a What is a loop and what does it do?

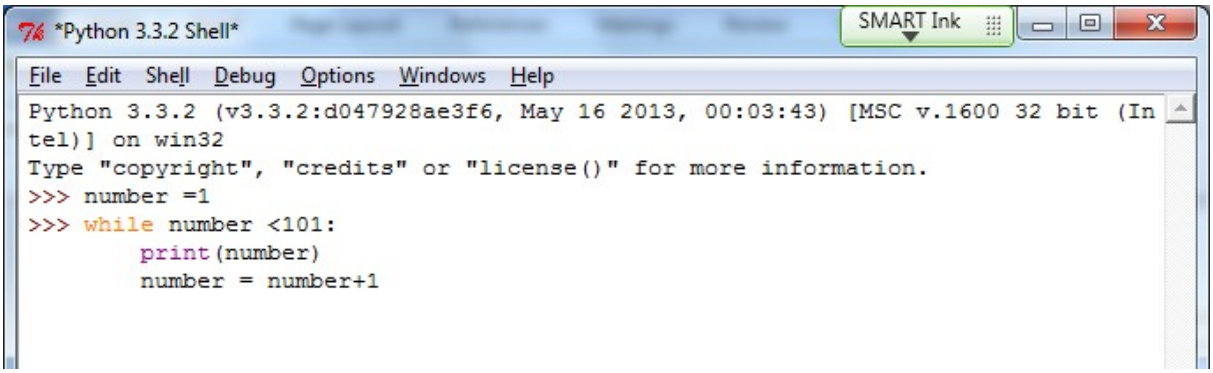
Computers are really good at repetitive tasks. The computer word for repetition is 'iteration'. Computers are not only good at these tasks, but they are very fast and they don't get bored very quickly. If we want to tell a computer to do something over and over as well as how to do it, we do this using a WHILE LOOP. This runs the code while something in the program is true, it then stops when its false.

WHILE Loop – A statement in coding that allow the repetition of something for a given condition. E.g. While it is true or While it is false. Upon meeting the condition, it stops.

Let's try some things in the shell mode first of all and see what a while loop can do. Type out the following code. When you have completed it, press enter.

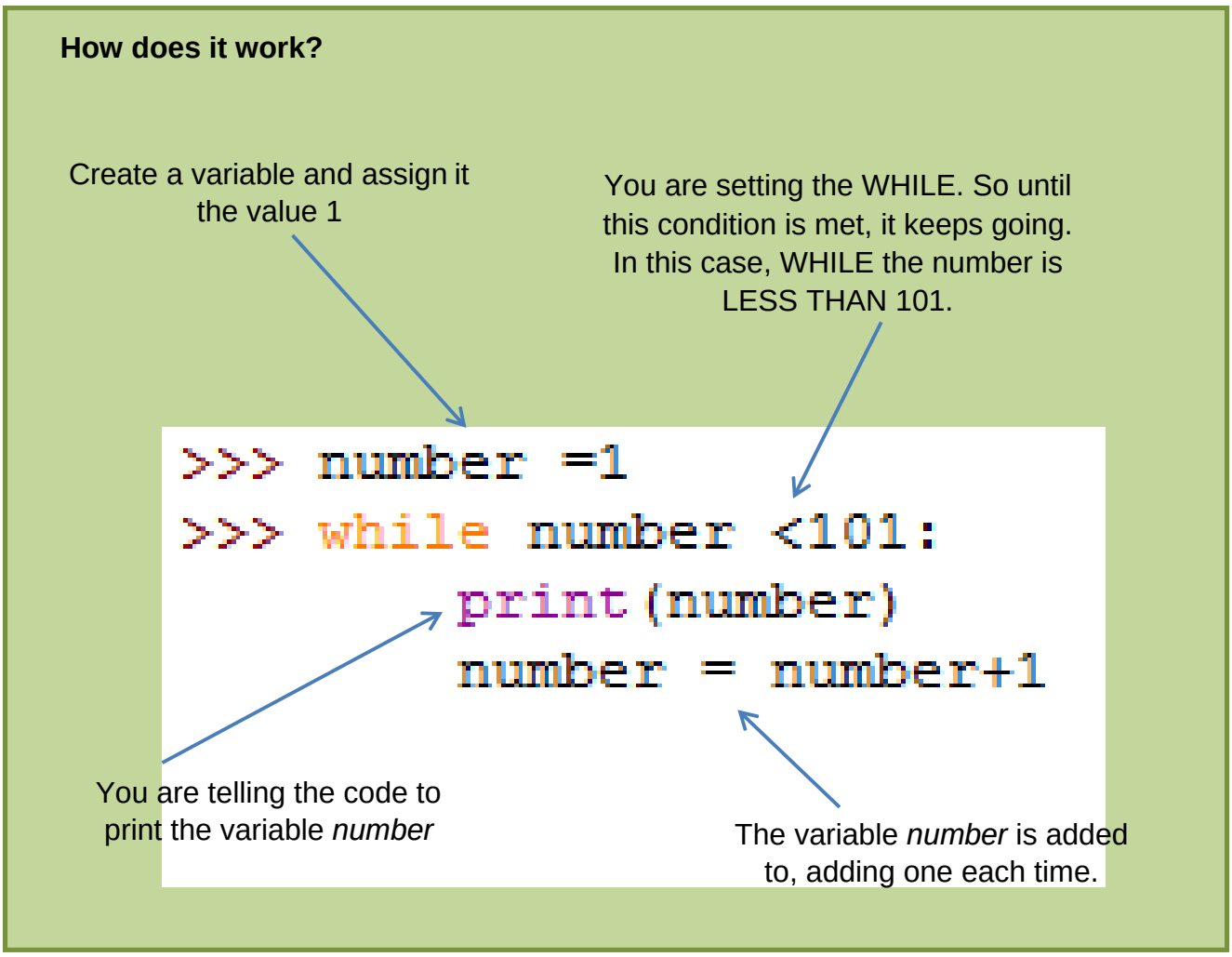


Let's try something else. Type this into Python's shell.



As you can see, we can loop carious thing, various times. It even counts for us.

How does it work?



In the above example, we can see that the code is set to count until we reach the number 101. There are other OPERATORS that we can use to achieve different results.

Operator	Meaning
==	Equal To
!=	Not Equal To
>	Greater Than
<	Less Than
>=	Greater Than Or Equal To
<=	Less Than Or Equal To

3.1b – Student Activity (WHILE Loops)

Using what you have just been taught, it is now your turn to try and create some simple while loops using Python.

You should be using Python in Scripting mode to complete the following questions.

1. Create code in Python that will instruct the computer to count up to 20.

2. Create code in Python that will allow you to count up to 20, this time in twos.

3. Using your knowledge of while loops, write some code that will output the 5 times table in a format like below.

   1x5=5
   2x5=10
   3x5=15
   4x5=20

   Hint – You will need a counter variable while you could call number. You should then run this 10 times one line at a time.

4. Write some code using the new operators provided to create a program that will allow you to count to 100. Write correct code below.

.

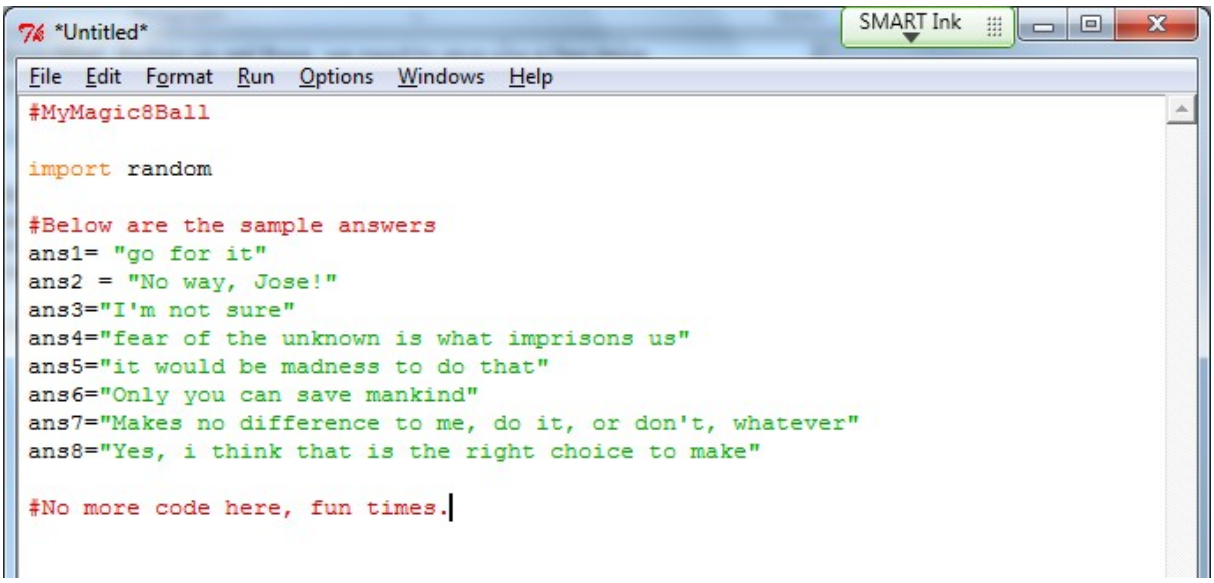
Congratulations, you have just learnt have to use while loops in Python. Go you!

4.0 Commenting

4.0a Why do we comment?

Sometimes in code, we need to add things that are for humans to read and for computers to ignore. We do this by commenting. The hashtag (#) symbol is used to comment in python scripting mode and can be used to write reminders of what each part of the code does for humans.

Below is an example of what commenting looks like in Python scripting mode. You will see them in #Red, Marked with a Hashtag.



It is good practice to always comment your code, so you know what you have done and also so your teacher and others can see what section of the code is for what purpose.

4.0b – Experiment with Comments

Open some of your previous Python scripts and have a go at adding some comments to them. Be sure to save them after you have finished.

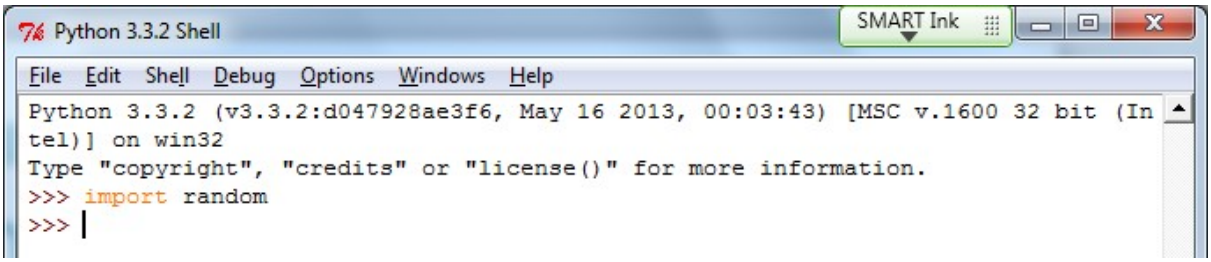
5.0 Modules

5.1a What is a Module?

A module is a section of codes that preforms a specific function in the code. In our case we are going to focus on the Random Module. This preforms a random selection of numbers, a bit like a dice does when you throw it.

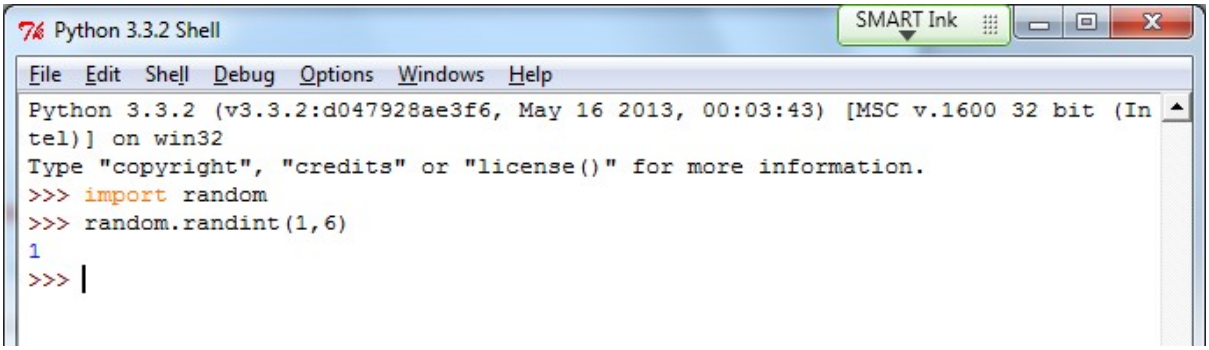
5.1b A Simple Dice

Before doing anything we must be sure to import the function random. By doing this we can tell the computer to expect that there will be a function called random in the program.



Below you can see that we have added the ***random.randint(1,6)*** command, this means we are randomising a number between the numbers 1 & 6.

You will see that the number that it randomised for me was the number 1.



The int part of ***randint*** means integer, which mean it's a whole number.

5.1c Student Activity (Modules)

So, now you know a little about modules. Let's see if you can work out the following. Be sure to save its task, giving it a name that you can look back on in case you need the code. Remember, you can always look back in the workbook or back at previous codes to help you.

1. Create the code as above that will allow the simulation of a 6 sided dice to be rolled and return a result.

2. Extend this code to allow a 12 sided dice to be rolled and return a result.

3. Using what you know about user input and variables. Try to allow user input of the amount of sides of the dice.

   Hint – You will need to use a user input to ask the user how many sides they want the dice to have before generating the random number.

CHALLENGE – Using what you know about generating random numbers, see if you can generate two random numbers. Print both of these numbers next to each other as if two people were rolling a dice. You might want to add names. Write your working code below.

Hint – You will need to use user inputs for the names of the players rolling the dice and the two random numbers.

Congratulations, you can now use the random module!

6.0 Selection

Selection means selecting (or choosing) what to do next. Should I cycle to school, or ask for a lift? If it's a sunny day I might cycle. If it's raining, I'll ask for a lift.

6.0a - IF ... ELSE (Student Experiment)

Create a new file and type in the following code into script mode:

```
x=int(input("How many hours a day do you play computer games? ")
if x < 2:
     print("That seems a fairly healthy balance. Well done!")
else:
     print("You're probably good enough by now with all that
practice.")
```

Save this file as "games.py", press F5 and test it.

Notice how the colon (:) is used to say what should happen in each case.
Also notice that the indentation is VERY important. Python only knows when your IF statement is finished by looking at the indentation!
Try this program to see what I mean:

```
x = int(input("How many hours a day do you play
computer games? ")
if x < 2:
     print("That seems a fairly healthy balance. Well done!")
else:
     print("You're probably good enough by now with all
that practice.")
print("Xbox 360s are better than PS3s")
```

6.0b - IF ... ELIF ... ELSE (Student Experiment)

Sometimes there are more than two options. I could walk OR cycle OR get the bus OR get a lift. As well as IF and ELSE, we can stick an 'ELSE IF' (or ELIF) in the middle:

```
x = int(input("How many hours a day do you play
computer games? ") if x < 2:
    print("That seems a fairly
healthy balance. Well done!")
elif x < 4:
    print("You're probably good enough by now with
    all that practice.")
else:
    print("Put the controller down and get some
    fresh air once in a while!")
```

6.1c - IF ... ELIF ... ELIF ... ELIF ... ELSE

You can include an unlimited number of ELIFs if you need to. Try the following:

```
menu = "What would you like:\n\
   1. A complement?\n\
   2. An insult?\n\
   3. A proverb?\n\
   4. An idiom?\n\
   9. Quit\n"

x =

int(input(menu))

if x == 1:
   print("You look
lovely today!") elif
```

```
x == 2:
   print("You
smell funny.")
elif x == 3:
   print("Two wrongs don't make a right.
But three lefts do...") elif x == 4:
   print("The pen is mightier
than the sword.") elif x == 9:
```

```
print("Goodbye!!!")
```

There are a couple of important bits here:
You can put a line break in your string by using "\n".

You can continue a line of code by putting a "\" at the end.

If you are testing for equality, use a double equals (is 3x2 == 6?)

6.1d – Student Activity (Selection)

Let's try and use some selection statements in

python. You will need to open a script mode Python

console.

1. Create code in python that ask the user for two numbers. Use an if statement to tell the user which one is bigger.

2. Write a program that asks the user for their Password, IF their password is equal to "password" then let then print the message "welcome to the world", otherwise print "Access Denied"

   Hint – You will need only one variable.

3. Modify your existing program that asks for the user's 2 favorite numbers. Then ask the user for what they want to do with those numbers (+, -, *, /) then perform that operation using IF Statements

CHALLENGE – Using code from your Modules section. Create code to randomly select two numbers and then tell you which out of the two numbers is bigger. Write your code below.

7.0 MyMagic8Ball Project

So, it's time for you to use all you know now to finally create a program all of your own. We are going to call this program – MyMagic8Ball.

The concept is that you can run the program, ask it a question and it gives you a random piece of advice. Some good, some bad and some weird ones.

7.1a – Project (MyMagic8Ball – Student Project)

You will need to carry out your entire project on ONE script.

1. Print code that welcomes the user to the Program. Something like "Welcome to MyMagic8Ball"
2. Ask for user input and assign it to the variable QUESTION. Your user input should ask the
3. Add comments to your code so that you teacher knows what you are expecting the code to do. (#) adds comments.
4. Import the random function into your code at the start
5. Add in a list of answers that you would like your magic 8 ball to possible answer questions with.
6. Add in a module that will allow the dice to randomly select a number between 1 and 8, the total numbers of options on the Magic 8 ball.
7. Add in a selection statements to match up the random number to an answer

   HINT – You will need multiple else statements.

   ```
   #using the randint function to select a
   reply choice=random.randint(1,8)
   if choice == 1:
      answer=ans1
   elif choice == 2:
   ```

8. Print the answer you have selected
9. Check all your code works.

   Check with your teacher to see if there is any further activities you can do?

8.0 Extension Exercises

Here are some fairly straightforward exercises. They're not completely easy, but should be quite manageable. All of these programs should include the following comment block at the start (preferably completed) as well as appropriate comments within the code to explain what is going on:

"""

Filename:

Author:

Date:

Description:

"""

8.0a - Pythagoras' Calculator

Write a program that will do the following:

• Print a menu:

Pythagoras' Calculator 1 - Find a hypotenuse 2 - Find another side 9 - Exit

Enter an option:

• If '1' is entered, prompt for the two other sides, calculate the hypotenuse and print the answer. Reprint the menu.

•If '2' is entered, prompt for the hypotenuse and the other side, calculate the third side and print the answer. Reprint the menu.

•If '9' is entered, print a goodbye message and exit (break)

•If another value is entered, print an error message and print the menu again.

NB: Remember you will need to import the math module (import math) and use the sqrt() function.

8.0b - Primary Division

Write a program that will do the following:

- Prompt the user for two numbers.
- Calculate, and print the result of the division in the format x remainder y (e.g. 17 / 4 = 4 remainder 1).
- Ask the user if they would like to run the program again or quit.

8.0c - Random Arithmetic

The random module lets you generate random numbers (actually pseudo-random, but that's another story) using the function random.randint(x,y) where x and y are the lower and upper boundaries.

To get you started, try this program:

```
import random #import the random

module for n in range(20): #repeat 20

times
    print(random.randint(1,100)) #print a random integer between 1 and 100
```

Write a program that will do the following:

- Generate two random numbers
- Calculate and print the results if you add, subtract, divide and multiply the two numbers verbosely (e.g. "2 + 3 = 5")
- Ask the user if they would like to run the program again or quit.

Notes Pages

Notes Pages

FAQ [Frequently Asked Questions]

The Becoming "Un-Stuck" Guide to Python

Why is my code not working?

There could be many reasons why your code isn't working.
Have you looked over it line by line for example?
Code is a little bit like a puzzle - there could be just one little bit that is slightly wrong that make the puzzle not work.
Looking at code line by line or "debugging" should help you figure out what's wrong with the code.

It is important to look at the code and read it out like a flowchart.

Many problems that you encounter will probably be the result of one small error. Check the following:

Capital letters? Do you need one?
Speech marks? Have you used them?
Greater or less than, is it the right way around?
Have you checked that you have use brackets where needed and closed them where needed?

Remember…. Half the fun is finding the error!

What is white space?

White space is the area that is white on the code screen. It is the way you indent and lay down the code. The white space in Python is really important, you must indent and lay things out correctly, or you may end up with issues.

How do I fix my code?

Be sure to check it line by line. Look at the steps and tips above to try and fix your code.

What if I get a "Syntax" Error, what does this mean?

Syntax means that you have put something in the wrong order or in the wrong place. You will need to try and fix this before the code will work. When you get a Syntax error you will get lots of red text telling you there has been a problem. This can be frustrating, but be sure that you read your code line by line.

Where can I find more coding examples?

You can use websites on the internet as well as the interactive websites provided by the school. These are Codecademy and Code Avengers. You may also wish to have a look at W3Schools and look at their coding examples and interactive solutions.

What if I get stuck?

Half of the fun with code is it breaking and you having to fix the error. You are never really STUCK. You can use the internet if you need to find snippets of code. You can ask your test busy; you can also ask your teacher as a last resort. Try and solve the problem yourself before asking for help from others.

Coding is hard, why are we learning it?

Coding languages re becoming ever more important in the world and they are becoming much more widely used. We believe that you should leave school with as much Computing and ICT knowledge as possible using as many apps and languages as possible. The world is looking for people who have an in depth computer knowledge and so we aim to this by equipping you with these skills.

What is a .py file?

.PY files are Python script files and they are used to store python scripts. You can run these from your computer and they run in the interactive shell.

Can I use Python at home?

Python is FREE and can be downloaded at home for you to practice on.
It's important to practice code. Practice makes perfect.

What if I finish the work quickly?

There are a number of extension tasks in this work booklet; I might be that you wish to have a god at one of these. You may wish to take some time to extend your own code, or explore new coding elements. There are also the options to use the interactive websites to gain even more skills for you to implement in your own Python code.

Student/Teacher – Feedback Section (not compulsory)

Section	Feedback	Level & Stamp
1 **My First Program / Meeting Python**	WWW EBI MRI	Level Achieved ___
2 **Variables and User Inputs**	WWW EBI MRI	Level Achieved ___
3 **While & For Loops**	WWW EBI MRI	Level Achieved ___
4 **Commenting**	WWW EBI MRI	Level Achieved ___

5 **Using Modules**	WWW EBI MRI	Level Achieved ___
6 **Selection Statements**	WWW EBI MRI	Level Achieved ___
7 **MyMagic8Ball Project**	WWW EBI MRI	Level Achieved ___
8 **Extension Activities**	WWW EBI MRI	Level Achieved ___

9 **Assessment Lesson**	WWW EBI MRI	Level Achieved ___

End of Unit Assessment Guide

Level	Descriptor (Sub-Level)
Level 4	A. Understand python and its applications B. Understand Python C. Know what Python is
Level 5	A. Apply Python basics to a given scenario and create a working solution B. Apply Python basics to create a solution that partly works. C. Apply Python to a situation with errors.
Level 6	A. Use variables and loops confidently in Python creating effective and effective code. B. Use variables and loops in Python to crate code that works mostly. C. Understand and use variables and loops in code. This may not be the most effective.
Level 7	A. Be able to create effective and efficient code in Python using a range of skills presented in the unit of work. B. Be able to create code in Python using an array of skills some of which won't be effective C. Create Python code that uses a number of skills but doesn't always work.
Level 8	A. Be able to critically evaluate the code they have written in Python adding comments, modules and sections of advanced code that work effectively and efficient and are fi for the task at hand. B. Be able to critically evaluate the code they have written in Python adding comments, modules C. Create and critically evaluate code in Python.