Lesson Activities Autumn Term Year 10

GCSE (9-1) Computer Science

Pearson Edexcel Level 1/Level 2 GCSE (9-1) in Computer Science (1CP1)

**LESSON ACTIVITIES FOR AUTUMN TERM YEAR 10**

# Week 1

## Lesson 1 activities

### Activity 1.1.1

* Write a program to print out ‘Hello World’.

### Activity 1.1.2

* Write a program that writes your name on the screen.

### Activity 1.1.3

* Copy and run this program:

print("hello " \* 10)

* Write a program that prints your name six times.

# Week 1

## Lesson 2 activities

### Activity 1.2.1

* Are the following statements true or false?

|  |  |
| --- | --- |
| Python was released in 2010. | True or false? |
| Python was named after the TV series *Monty Python’s Flying Circus*. | True or false? |
| Python is proprietary software, which is expensive to buy. | True or false? |
| Python was written by Mark Zuckerberg. | True or false? |
| YouTube is written in Python. | True or false? |

### Activity 1.2.2

* Make a note of what these useful keyboard shortcuts do.

|  |  |
| --- | --- |
| **Keyboard shortcut** | **What does the keyboard shortcut do?** |
| Control c | Copy |
| Control v | Paste |
| Control a | Select all |
| Control x | Cut |
| Control f | Find |
| Control n | New |
| Control p | Print |
| Control s | Save |
| Control z | undo |

### Activity 1.2.3

* Copy and run this program:

print("This is the end")

print("Hold your breath and count to ten”)

print("Feel the earth move and then”)

print("Hear my heart burst again”)

* Write a program that writes four lines of the lyrics of your favourite song on the screen.

### Activity 1.2.4

* Co y and run this program:

print("the rain falls","from the sky")

* Now answer these questions:
  + What happens?
  + What effect does the comma , have?

### Activity 1.2.5

Escape sequences can be used to alter how the information is displayed on the screen. An escape sequence is a back slash “\”.

* Experiment with these escape sequences and complete the table:

print(I’m \\ very \\ hungry.\*)

print("\tQuestion \”what goes woof”\t\tdogs\t\t\trabbits")

print("\n\nwhat kind of snake is good at maths?\n\nAn adder\n\n")

print("\n\n\’Goodbye’”)

*Hint: Use alt p to display the last command entered in the IDLE shell.*

|  |  |
| --- | --- |
| **Escape sequence** | **Effect** |
| \t | tab |
| \n | New line |
| \\ |  |
| \’ |  |
| \” |  |

### Activity 1.2.6

* Print this text using just one line of code:

Help, I need somebody

Help, not just anybody

Help, you know, I need someone

### Activity 1.2.7

* Write a program using print commands to display your initials five characters high on the screen.

For example, NH:

X x x x

Xx x x X

X x x Xxxxxx

X xx x x

X x x x

# Week 2

## Lesson 1 activities

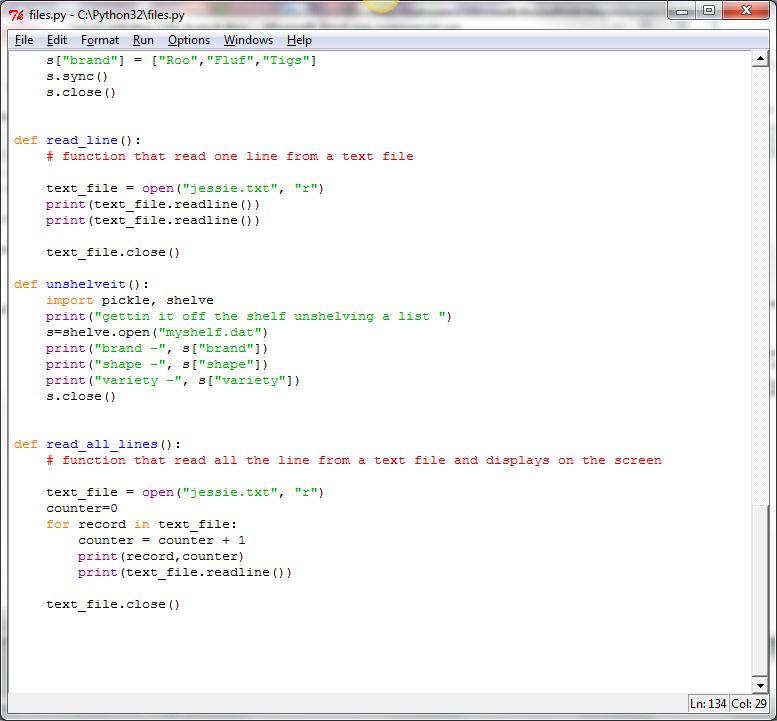
### Activity 2.1.1

An algorithm is a sequence of steps needed to perform a particular task. Algorithms can be represented in different ways.

* Match the words to the representations.

**Flowchart-D pseudo-code-B written description-C program code - A**

**A**



**B**

RECEIVE myName FROM (STRING) KEYBOARD

RECEIVE myAge FROM (INTEGER) KEYBOARD

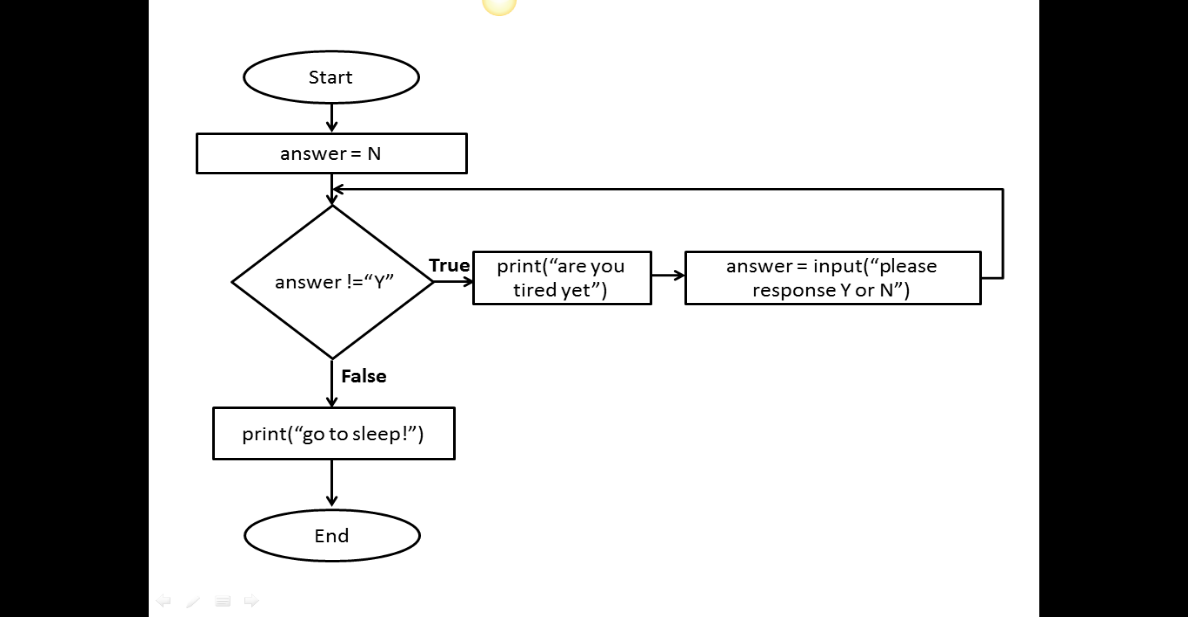
SET AgeInTen TOmyAge + 10

SEND myName “will be” AgeInTen “in 10 years time” TO DISPLAY

**C**

Write a program that prompts the user to enter their name and age. The program then adds 10 onto the age and displays the text “<name entered> will be <age + 10> in ten years time.”

**D**

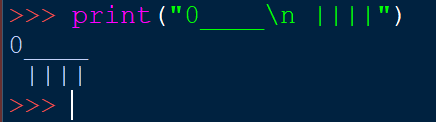


### Activity 2.1.2 (homework)

* Explore ASCII art (Wikipedia) and write a Python program to display your ASCII art creation.

.--. /\ \_\_\_\_

'--' /\_\_\ (^.\_.^)~ <(o.o)>

 Picture of a dog**Week 2**

## Lesson 2 activities

### Activity 2.2.1

* Copy and run these lines of code:

>>> 60/5

>>> 987+34

>>> 564\*89

>>> 2\*\*5

>>> 43-5

>>> 11//2

>>> 11%2

* Complete the table to explain what the mathematical operators do.

|  |  |
| --- | --- |
| **Mathematical operator symbol** | **Operation** |
| / | Division |
| + | Addition |
| \* | Multiplication |
| \*\* | Exponent |
| - | Subtraction |
| // | Whole number (not rounded) after division |
| % | remainder |

### Activity 2.2.2

* Make up some mathematical calculations of your own and add an example to the table for each mathematical operator.

|  |  |  |  |
| --- | --- | --- | --- |
| **Mathematical operator symbol** | **Operation** | **Example** | **Answer** |
| / | divide | 14 / 7 | 2.0 |
| + | add | 150+72 | 222 |
| \* | multiply | 80\*15 | 1200 |
| \*\* | exponential | 5\*\*3 | 125 |
| - | subtract | 95-34 | 61 |
| // | integer division | 29//5 | 5 |
| % | modulus (remainder after the division) | 32%7 | 4 |

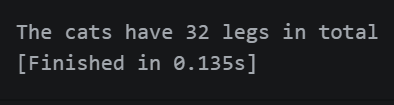
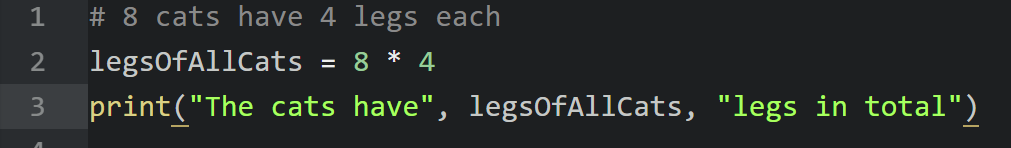
### 

### Activity 2.2.3

* Write a program to display this text on the screen and fill in the missing number.

8 cats have 4 legs each

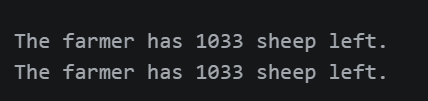
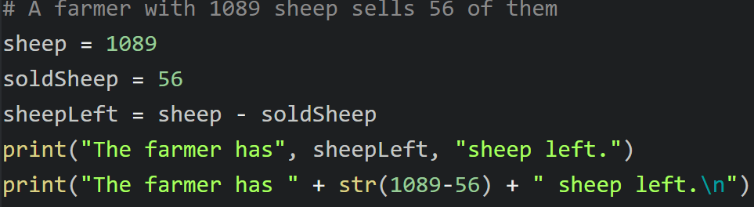
The cats have \_\_\_ legs in total



* Write a program to display this text on the screen and fill in the missing number.

A farmer with 1089 sheep sells 56 of them

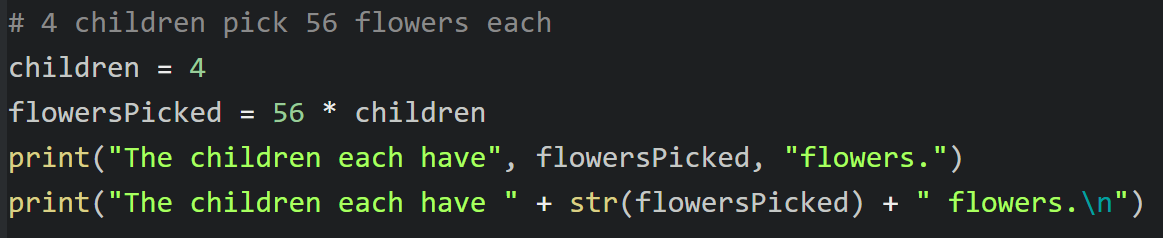
The farmer has \_\_\_\_\_ sheep left



* Write a program to display this text on the screen and fill in the missing number.

4 children pick 56 flowers each

The children each have \_\_\_\_ flowers



### 

### Activity 2.2.4

* Copy and run these lines of code:

>>> 5 \* 3 / 6 + 4

>>>(5 \* 3) / (6 + 4)

* What effect do the parentheses () have?

### Activity 2.2.5

* Predict what you think will be displayed on the screen when this line of code is executed.

15 / 2 \* 3 + 2

* Now copy and run the code. Explain the answer you get. Is it what you predicted?

### Activity 2.2.6 (homework)

* Make up some multiple-choice questions on precedence. Each question must have four possible answers, one of which is correct.

Here are two examples:

What is the correct answer to the following expression?

>>> 7 + 4 \* 5

1. 55
2. 27
3. 16
4. 33

What is the correct answer to the following expression?

>>> 6 -2 / 2 + 5

1. 0.57
2. 10
3. 7
4. 11

* Check your answers are correct by using the interactive Python shell.
* Try your questions out on other people. You must be able to explain to them why the answers are correct.

### Activity 2.2.7 (homework)

* Devise your own mnemonic for remembering the order of precedence.

|  |  |
| --- | --- |
| **Python order of precedence** | **Mnemonic** |
| Parenthesis (brackets) |  |
| Exponential |  |
| Division and multiplication |  |
| Addition and subtraction |  |

# Week 3

## Lesson 1 activities

### Activity 3.1.1

* Copy and run this code and explain the result:

>>>print hello world

Syntax error, missing parentheses and speech marks

### Activity 3.1.2

* Copy and run this code and explain the result:

>>>print("hello world")

Result: hello world

### Activity 3.1.3

* Python produces different types of error. Match up the type of error with its description.

|  |
| --- |
| Type of error |
| TypeError |
| RuntimeError |
| NameError |
| ZeroDivisionError |
| KeyboardInterrupt |
| Description of the error | |
| Dividing a number by zero. | |
| When a program is interrupted from the keyboard by pressing control+c. | |
| When a name is used that is not known about (often a variable name that is misspelt). | |
| When an operation is attempt that is invalid for that type of data. | |
| An error occurs when the program is running. | |

* Experiment with Python to see if you can create each type of error.

*Hint: A full list of Python errors can be found here:* [*www.tutorialspoint.com/python/standard\_exceptions.htm*](http://www.tutorialspoint.com/python/standard_exceptions.htm)

### Activity 3.1.4

* Copy and run this code and explain the result:

# Programmer Amy Smith 6 September 2016

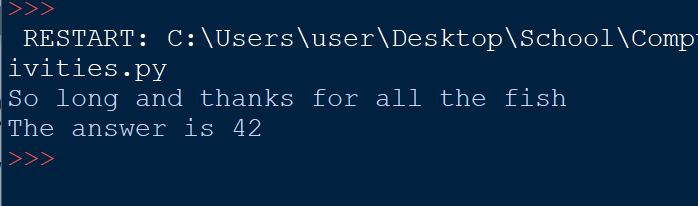
**No result because the line with ‘#’**

### Activity 3.1.5

* Use *file/new window* to open a script file in the IDLE.
* Copy this code into the file:

print("So long and thanks for all the fish")

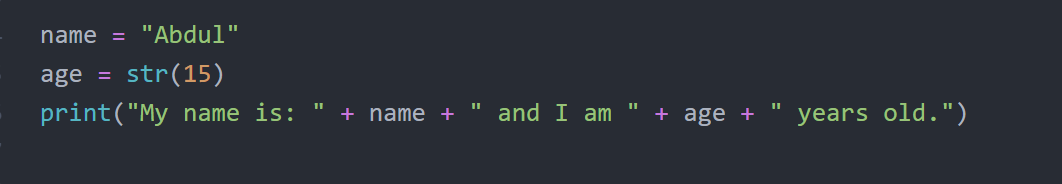
print("The answer is",6\*7)

* Save the file using *file/save*. Always give the filename an extension of .py. Use a meaningful filename and store in a suitable folder.
* Run the commands by using *run/run module*.
* Close Python and then restart Python, open the file and run the program again.
* 

### Activity 3.1.6

* Following the same steps as in Activity 3.1.5, write a program that displays your name and then your age.

*Hint: Use* file/new window *and save the file using* file/save*. Don’t forget to use a meaningful filename.*



### Activity 3.1.7 (homework)

#### Using the Python IDLE

* Write your own summary sheet to describe how to run Python programs, including how to save and open .py files. Include any other commands that you have found useful.

|  |  |
| --- | --- |
| **Using the Python IDLE** |  |
| How to open a new window | Ctrl n |
| How to save a file | Ctrl s |
| How to open a file | Ctrl o |
| How to run a program | F5 |
|  |  |
|  |  |
|  |  |
| **Useful tips** |  |
| alt p | Displays the last line you entered |
| Commands in the file are not colour coded | Save as a .py file |
|  |  |
|  |  |

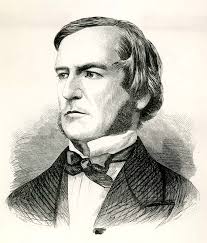
# Week 3

## Lesson 2 activities

### Activity 3.2.1

#### Where does the name ‘Boolean’ come from?

Boolean variables are named after George Boole who invented the mathematics of digital logic.

* Find a picture of George Boole and insert it here.
* 
* When was he born? November 2, 1815

### Activity 3.2.2

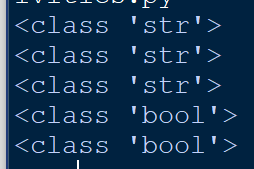
#### Data types

* Use the ‘type’ function to find out the data types for these values:

>>>type("Fred")

>>>type(198)

>>>type(88.9)

>>>type(True)

>>>type(False)

*Hint: Remember that True and False must start with a capital letter.*

### Activity 3.2.3

The type function returns the data type of an expression.

* For each of these expressions, first predict the data type then use the type command to see if your prediction was correct.

|  |  |  |  |
| --- | --- | --- | --- |
| **Expression** | **Predicted data type** | **Type command** | **Result** |
| “hello world” | String | type("hello world") |  |
| False | **Bool** | type(False) |
| 15 | **Integer** | type(15) |
| 35.6 | **Float** | type(35.6) |
| -999 | **Integer** | type(-999) |
| “15” | **String** | type(“15”) |
| “False” | **String** | type(“False”) |
| True | **Bool** | type(True) |
| 0.001 | **Float** | type(0.001) |

### Activity 3.2.4

The interactive shell is a useful place to explore variables.

* Copy and run this code in the interactive shell:

myName="Fred Smith"

myAge=14

print(myName,myAge)

* Create a variable called myName and assign it to your name.
* Create a variable called myAge and assign it to your age
* Create a variable called myEyes and assign it to your eye colour.
* Create a variable called myHeight and assign it to your height.
* Write the commands to display on the screen your name, age, eye colour and height.

*Hint: Remember that a single ‘=’ is used to assign a value to a variable.*

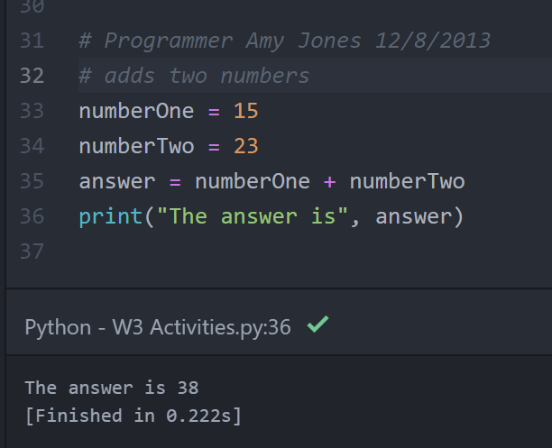
### Activity 3.2.5

#### Variable names

A programmer is trying to decide on a valid name for a variable that represents a house number.

* Which of the following variable assignments are valid? Why are the others not valid?

|  |  |  |
| --- | --- | --- |
|  | **Valid or invalid variable name?** | **Reason why not valid** |
| 8HouseNumber = 288 | Invalid | Starts with a number |
| houseNumber = 288 | Valid |  |
| house Number = 288 | Invalid | Space between the two words |
| house\_number = 288 | Valid |  |
| import = 288 | invalid | Uses a python keyword |

* What type of error do you get when using an invalid variable name?
* NameError

### Activity 3.2.6

* Copy and run this code and explain the result:

# Programmer Amy Jones 12/8/2013

# adds two numbers

numberOne=15

numberTwo=23

answer=numberOne + numberTwo

print("The answer is ",answer)

* Amend the program to add another variable called numberThree. Assign the value 76 to this variable. The program should add up all three numbers and print the result.

### Activity 3.2.7

#### Data types in Python

Complete this table to describe the four data types.

|  |  |  |  |
| --- | --- | --- | --- |
| **Data type** | **Python abbreviation** | **Explanation** | **Example** |
| integer | int | Whole numbers | 3 |
| string | str | Text | “string” |
| float | float | Numbers with decimals | 0.3 |
| boolean | bool | True or false | True |

*Hint: Quotation marks are used to show the start and end point of a string, e.g. “this is a string”.*

# Week 4

## Lesson 1 activities

### Activity 4.1.1

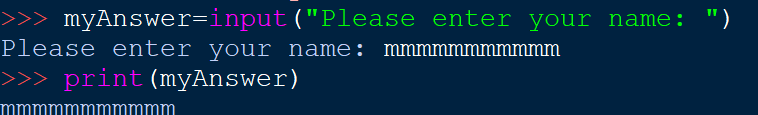
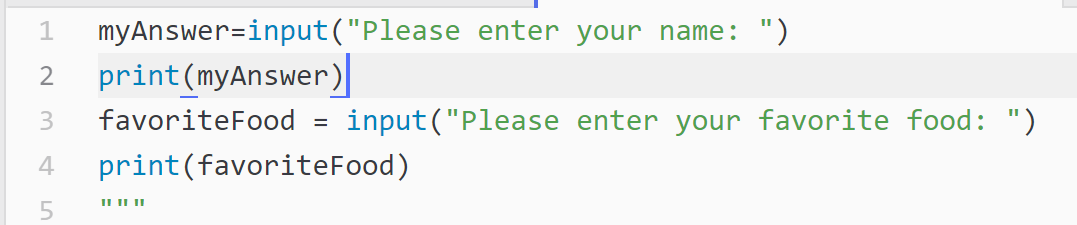
#### Input function

The input function allows the user to input a value and assign it to a variable.

* Copy and run this program:

myAnswer=input("Please enter your name: ")

print(myAnswer)

* Re-run the program several times, entering different names each time.
* 
* Write a program that asks the user for their name and favourite food and displays these on the screen.
* 

### Activity 4.1.2

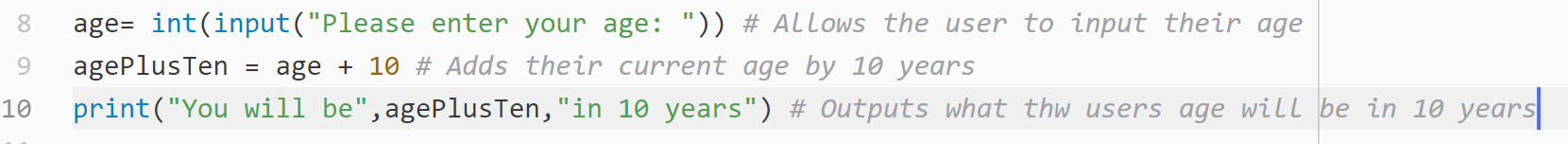
#### Int function

* Copy and run this program:

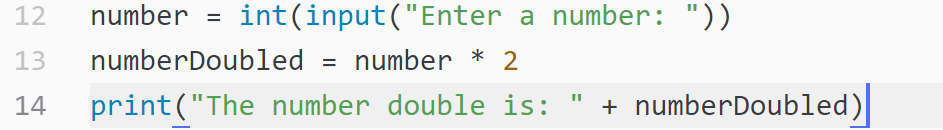
age=input("Please enter your age: ")

agePlusTen = age + 10

print("You will be",agePlusTen,"in 10 years")

* Add comments at the start of the program to explain briefly what **Int** does. Check with a neighbour that your comments are helpful and make sense.
* 
* Explain why the program does not work.
* It works
* Correct the program.

### Activity 4.1.3

* Write a program that asks you to enter a number then displays the number doubled.
* 

# Week 4

## Lesson 2 activities

**Activity 4.2.1**

#### String formatting: text

The string method .format gives you more control over the formatting of your output than printing using space-separated values. The string formatting commands are given in curly brackets {}.

* Copy and run these lines of code:

>>>foodOne="fish"

>>>foodTwo="chips"

>>>print("{0} and {1}".format(foodOne,foodTwo))

fish and chips

>>>print("{1} and {0}".format(foodOne,foodTwo))

chips and fish

>>>print("{1} {1} {1} and {0} {0} {0}".format(foodOne,foodTwo))

chipschipschips and fishfishfish

* Create these variables:

one = “cheese”

two =”onion”

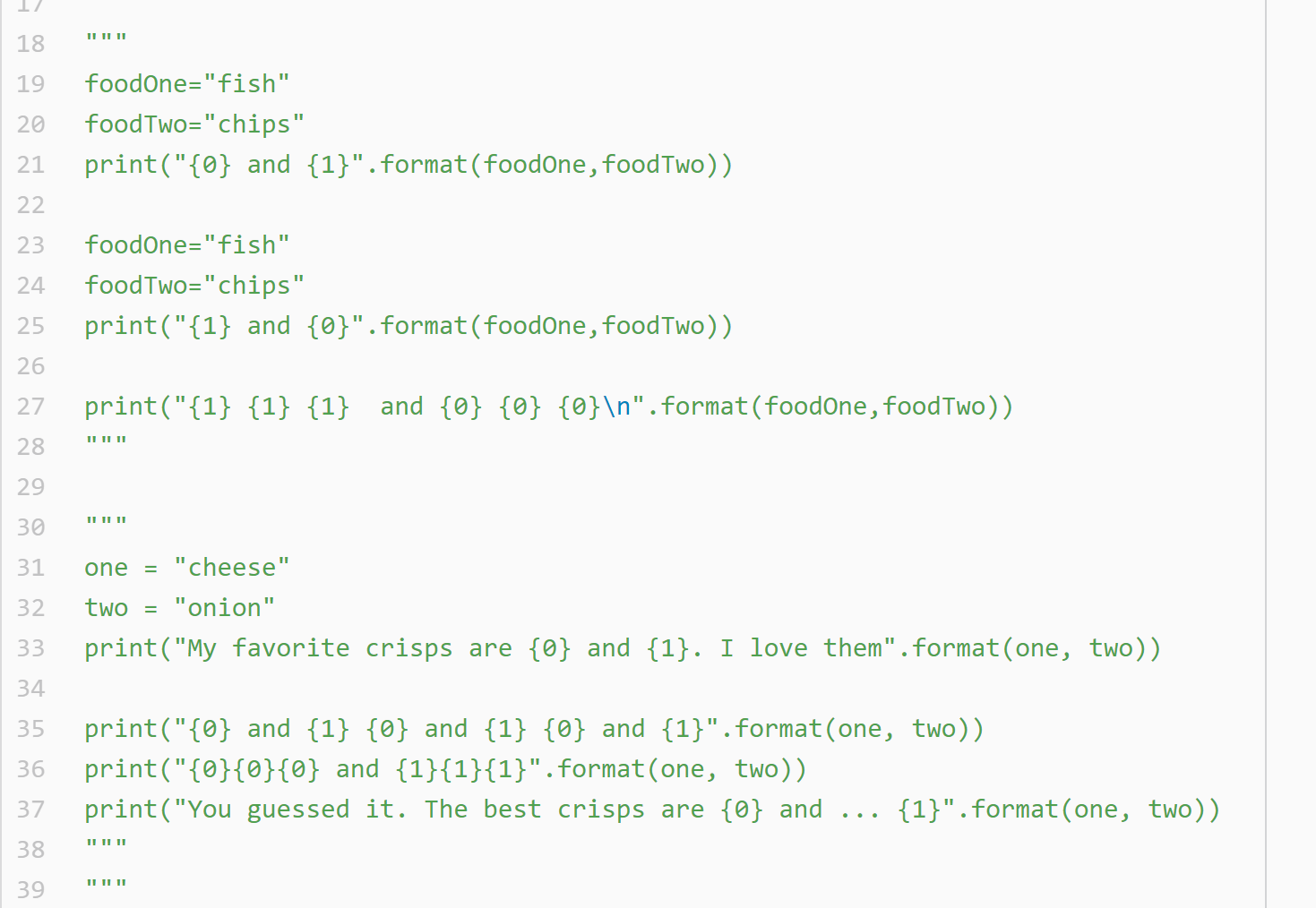
* Use the .format command to display:

My favourite crisps are cheese and onion. I love them!

cheese and onion and cheese and onion and cheese and onion

cheesecheesecheese and oniononiononion

You guessed it. The best crisps are onion and … cheese.

* Try altering the flavours assigned to the variables to your favourite flavour! Enjoy.
* 

### Activity 4.2.2

#### Formatting: numbers

You can use the .format method to format the number of decimal places.

* Copy and run these lines of code:

>>>number = 98.1468297645

>>>print("The answer is {0:.5f}".format(number))

The answer is 98.14683

>>>print("The answer is {0:.4f}".format(number))

The answer is 98.1468

>>>print("The answer is {0:.3f}".format(number))

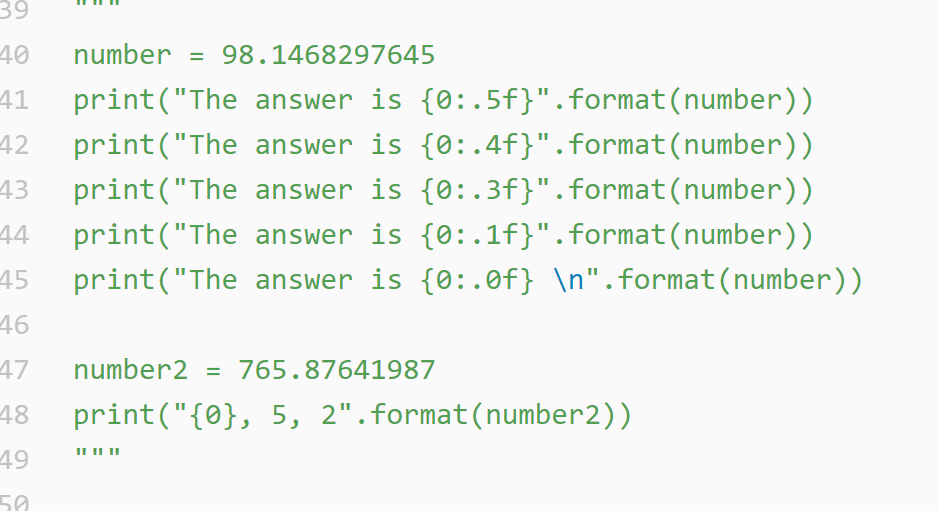
The answer is 98.147

>>>print("The answer is {0:.1f}".format(number))

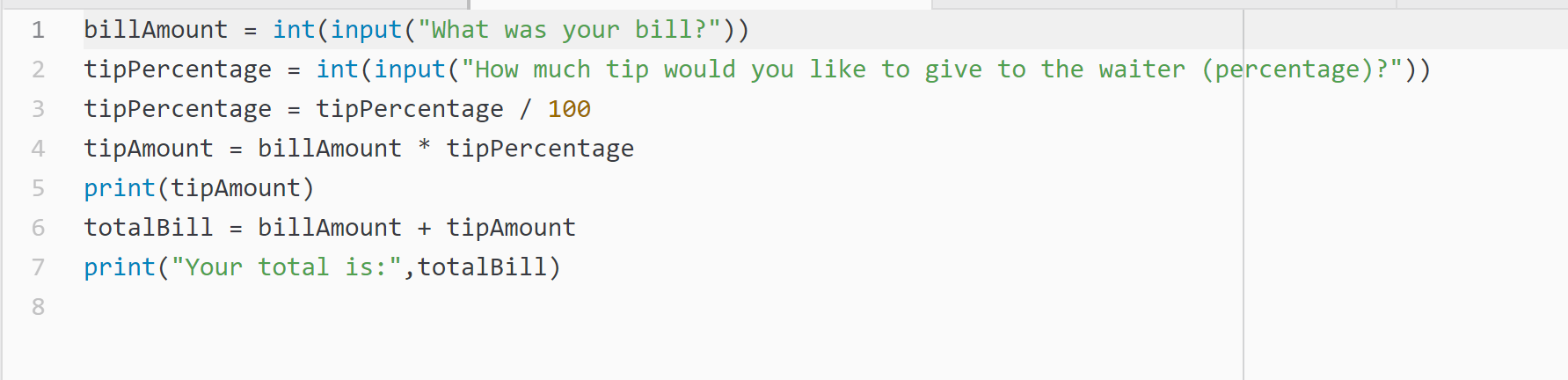
The answer is 98.1

>>>print("The answer is {0:.0f}".format(number))

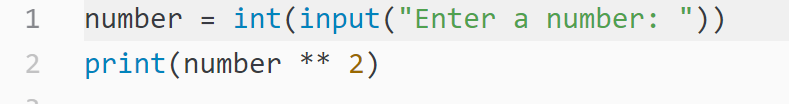
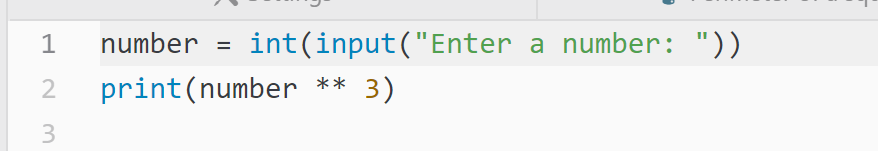
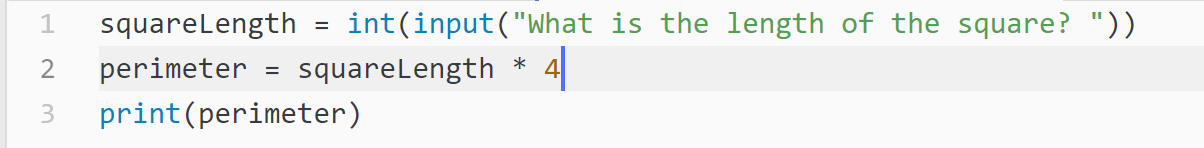
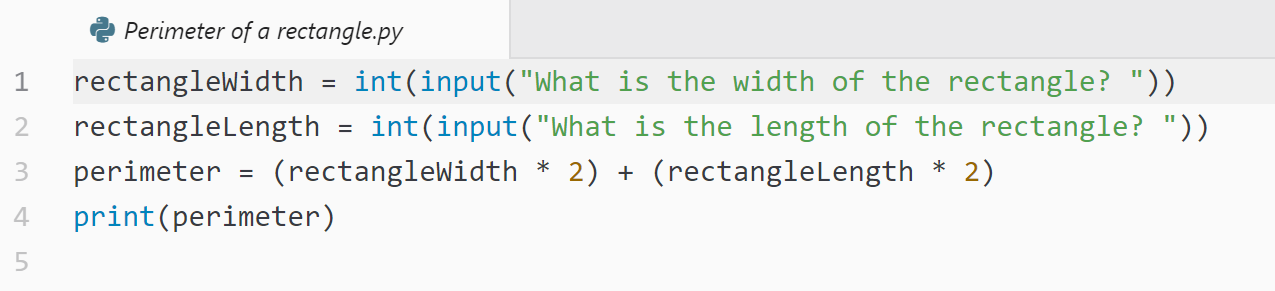
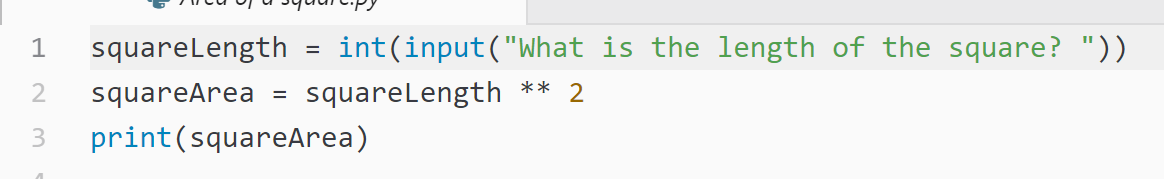
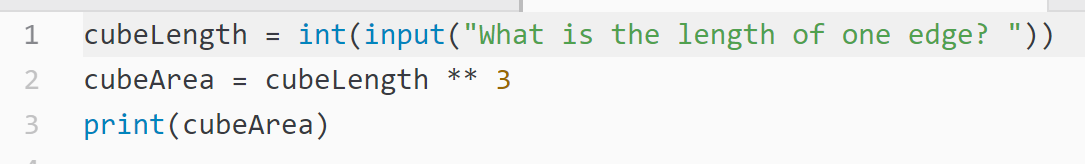
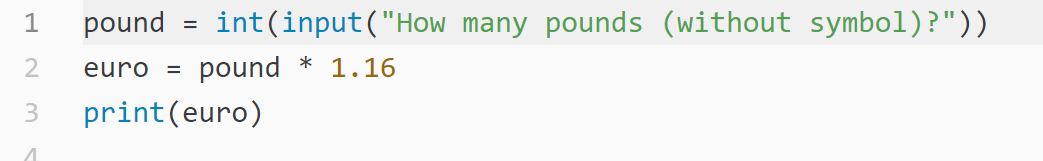
The answer is 98

* Assign the number 765.87641987 to a variable and display the number with 5, 2 and no decimal places using the .format command.
* 

### Activity 4.2.3

* Write a program that asks how much your bill is at a restaurant and then asks you to enter the % you want to give in a tip. The program should display the amount of tip to give to the waiter.
* Amend the program so that it also displays the total cost of the meal including the tip.
* 

### Activity 4.2.4 (homework)

* Write a program that displays the square of a number.
* 
* Write a program that prompts for a number and then displays the cube of a number.
* 
* Write a program to find the perimeter of a square.
* 
* Write a program to find the perimeter of a rectangle.
* 
* Write a program that finds the area of a square.
* 
* Write a program that finds the area of a cube. 
* Write a program to convert from pounds to euros.
* 

*Hint: All these programs should prompt for the input information and display the result appropriately.*

# Week 5

## Lesson 2 activities

### Activity 5.2.1

#### Relational operators

Password checking is an example of a relational operator. If the password entered is the same as the password stored then the condition is true. The operator is ‘is equal to’.

* Brainstorm other examples of condition statements and decide what the operator is.

|  |  |
| --- | --- |
| **Relation statement** | **Operator** |
| > | More than |
| < | Less than |
| == | Is equal to |

### Activity 5.2.2

* Complete this table of the Python relational operators. Give an example of each and say whether it will evaluate to true or false.

|  |  |  |  |
| --- | --- | --- | --- |
| **Relational operator** | **Operator** | **Example** | **Evaluates to** |
| Equal to | == | Answer = 6  If answer == 6  Print(“123”) | 123 |
| Not equal to | != | Answer = 8  If answer != 6  Print(“123”) | 123 |
| Greater than | > | Answer = 75  If answer >= 70 & answer < 80  Print(“123”) | 123 |
| Greater than or equal to | >= | Answer = 6  If answer >= 6  Print(“123”) | 123 |
| Less than | < | Answer = 6  If answer < 6  Print(“123”) |  |
| Less than or equal to | <= | Answer = 6  If answer == 6  Print(“123”) | 123 |

* Try out your expressions by typing them into the Python interactive shell.

### Activity 5.2.3

#### Greater than and less than

* Find a way that works for you of remembering the difference between ‘less than’ < and ‘greater than’ >.

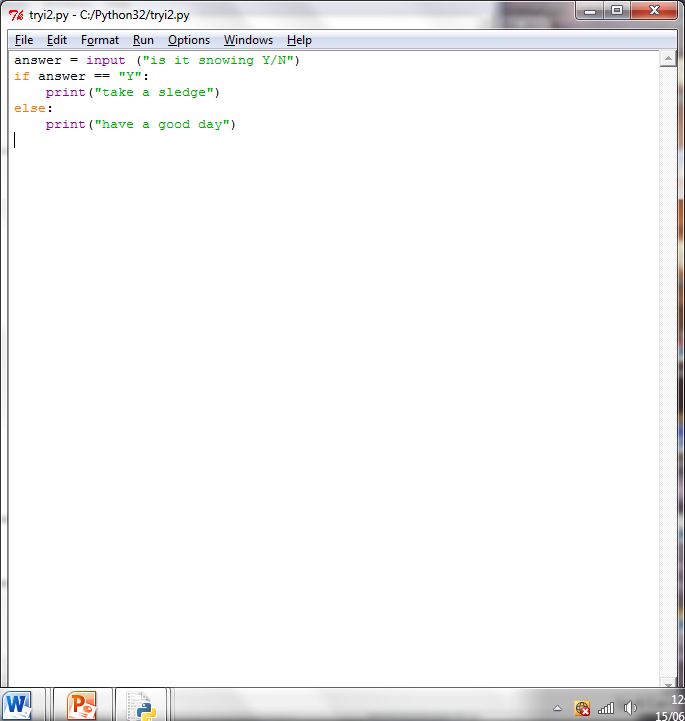
*Hint: As there are only two options, you only need to learn one!*

|  |  |  |
| --- | --- | --- |
| **Operator** | > | < |
| **Operator meaning** | Greater than | Less than |
| **How I remember this** | **Crocodile opens its mouth for a bigger number**  **Number to the left is bigger** | **Crocodile opens its mouth for a bigger number**  **So number to the left is smaller** |

### Activity 5.2.4

This program asks if it is snowing and if so tells you to take a sledge, otherwise to have a good day. The condition is missing.

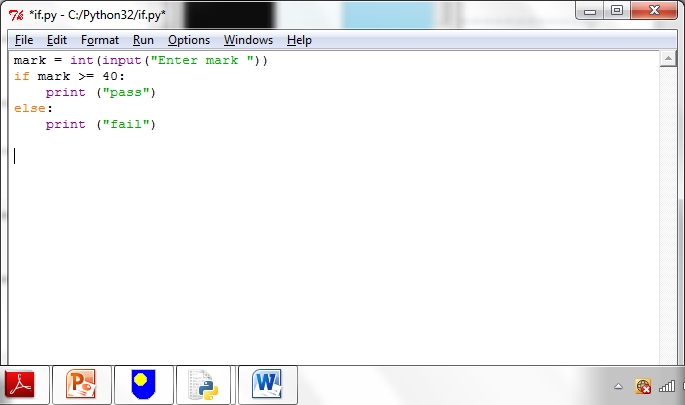
* Copy and complete the condition to make the program work correctly:



### Activity 5.2.5

What condition is needed in this program to display ‘pass’ if the exam mark is 40 or more?

* Copy and complete the condition to make the program work correctly:

****

### Activity 5.2.6

* Write a program that asks you to enter the colour of the light at a pedestrian crossing. If the light is green it tells you it is safe to cross, otherwise it tells you to stop.

**Activity 5.2.7**

* Write a program that asks you for your password. It then asks you to re-enter your password. If they are the same the message ‘access granted’ is displayed. If the passwords are not the same the message ‘access denied’ is displayed.

# Week 6

## Lesson 1 activities

### Activity 6.1.1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| * List as many different types of digital computers as you can. Look for computers both in the past as well as the present, and for objects that contain embedded computers.  |  |  | | --- | --- | | Phone | Tablet | | PC | Laptop | | Smartwatch | Smart TV | | Smart fridge | Robot vacuum cleaner | | Calculator | Microbit | | Consoles | Supercomputers | | Cars with an ECU | Smart glasses | | VR Headsets | DVD Player | | Dishwashers | Microwaves | | Remote controls | Wireless Headsets | |

### Activity 6.1.2

* Sort the things in this list according to whether they are hardware or software.
* Mouse
* Computer program
* Web browser
* Microphone
* CPU
* Hard disk drive
* Digital camera
* Windows operating system
* DVD drive
* Touch screen
* Word processor
* Speakers
* Spreadsheet
* Printer
* Mobile phone app
* Android operating system
* SD memory card

|  |  |
| --- | --- |
| **Hardware** | **Software** |
| *Mouse* | *Computer program* |
| *Microphone* | *Web browser* |
| *Cpu* | *Windows operating system* |
| *Hard disk drive* | *Word processor* |
| *Digital camera* | *spreadsheat* |
| *Dvd drive* | *Mobile phone app* |
| *Touch screen* | *Android operating system* |
| *Speakers* | *Presentation software* |
| *printer* | *Photo editing software* |
| *SD Memory card* | *Video editing app* |

### Activity 6.1.3 (homework)

* Answer these questions to summarise what you have learned in this lesson.

|  |  |
| --- | --- |
| What is a digital computer? | A computer with two-state transistors (on/off) to perform calculations |
| Define computer hardware. | Physical, tangible parts that make up a computer system |
| Give an example of computer hardware. | Speaker |
| Define computer software. | Collection of code that when run, get the computer to carry out a specific task, sequence by sequence. |
| Give an example of computer software. | Word processor |

# Week 6

## Lesson 2 activities

### Activity 6.2.1

* Write a computer program that asks the user to type in three numbers, adds them up and then displays the answer on the screen.
* Explain your program using the input-process-output model.

INPUT

Numbers are entered by the user

PROCESS

The computer adds up the numbers

OUTPUT

The result is printed on srcreen

### Activity 6.2.2

#### Using elif

This program simulates a fortune cookie. A random number is used to decide your ‘fortune’.

* Copy and run this program:

# a random number is given by the randint() function

import random

answer= random.randint(1,6)

if answer == 1:

print("You will make a new friend this week")

elif answer == 2:

print("You will do well in your GCSEs")

elif answer == 3:

print("You will find something you thought you’d lost")

* The program is not yet complete. Include your own ‘fortunes’ for the numbers 4, 5 and 6.

Note: random.randint(1,6) is a function that returns a random number between 1 and 6. The ‘import random’ command allows the program to access the random.randint() function.

### Activity 6.2.3 (homework)

* Select two actions you complete regularly and describe them in terms of the input-process-output model.

Examples could be:

* + running a mobile phone app
  + uploading an image to a social networking profile.

INPUT

Entering account Password

PROCESS

Checking the password is right

OUTPUT

Access to the account

INPUT

Entering a web address

PROCESS

Displaying the webpage

OUTPUT

webpage

# Week 7

## Lesson 1 activities

### Activity 7.1.1

* Use Python to experiment with how the computer handles strings, floats and bools. What is the outcome of running each of these code snippets? Why does this happen?

**#Snippet 1**

Snippet1

|  |  |
| --- | --- |
| Outcome |  |
| Why did this happen? | What is displayed and entered is a string.  They are both strings.  You cannot perform arithmetic operations with two strings together. |

**#Snippet 2**

Snippet2

|  |  |
| --- | --- |
| Outcome |  |
| Why did this happen? | Number1 is printed the number of times of number2 |

**#Snippet 3**

Snippet3

|  |  |
| --- | --- |
| Outcome |  |
| Why did this happen? | Both data values entered are integers |

**#Snippet 4**

Snippet4

|  |  |
| --- | --- |
| Outcome |  |
| Why did this happen? |  |

**#Snippet 5**

Snippet5

|  |  |
| --- | --- |
| Outcome |  |
| Why did this happen? | 5 \* 3.0 = 15.0 |

**#Snippet 6**

Snippet6

|  |  |
| --- | --- |
| Outcome |  |
| Why did this happen? | Cant multiply floats and strings |

### Activity 7.1.2

|  |  |
| --- | --- |
| * What are these binary numbers in denary? | |
| **Binary** | **Denary** |
| **0001** | 1 |
| **0010** | 2 |
| **0101** | 5 |
| **1000** | 8 |
| **1001** | 9 |
| **1100** | 12 |
| **1011** | 11 |
| **1111** | 15 |

|  |  |
| --- | --- |
| * What are these denary numbers in binary? | |
| **Denary** | **Binary** |
| **4** | 0100 |
| **6** | 0110 |
| **10** | 1010 |
| **0** | 0000 |
| **3** | 0011 |
| **7** | 0111 |
| **14** | 1110 |
| **13** | 1101 |

|  |  |
| --- | --- |
| What range of numbers can be represented by 8 bits? | 255 |

|  |  |
| --- | --- |
| Underline the most significant bit in this binary number. | **1**011 0011 |
| What weighting does it have? | 7 |
| What is its value in denary? | 128 |

**Activity 7.1.3**

|  |  |
| --- | --- |
| Answer these questions in binary. | |
| How many days in a week? | 7 – **0111** |
| How many months in a year? | 12 – **1100** |
| How many fingers (including the thumb) on one hand? | 5 – **0101** |
| How many toes on two feet? | 10 – **1010** |
| How many lives does a cat have? | 9 - **1001** |

# Week 7

## Lesson 2 activities

### Activity 7.2.1

**Writing readable code: a style guide**

Program code is read more often tha**n** it is written. Here are some guidelines on how to write Python code so that it is easy to read.

A style guide for Python code

* Indent by four spaces for each indentation level.
* Use blank lines to separate different parts of the program.
* Use two blank lines to separate functions.
* Choose meaningful names for variables, using camelCase or with words separated by underscores.
* Put imports at the top of the file.
* Include one space around each side of an assignment and other operators.
* Use complete sentences with the first word capitalised for comments.
* Write comments that add clarity and explain what the program does. Do not simply repeat what the code already says.
* Write function names in lowercase, with words separated by underscores.
* Use meaningful function names that describe the purpose of the function.
* Write constants in CAPITAL\_LETTERS.
* Use meaningful constant names, which describe the purpose of the constant.

For more details, see <http://www.python.org/dev/peps/pep-0008/>

Note: Functions are subprograms which start with the “deffunction\_name():. Constants are variables whose value never changes. You will be covering functions in later lessons.

* Apply the rules in the style guide above for this Python code:

def a(s):

if s<50:

print("You have lost")

else:

print("You have won")

def a(s):

if s<50:

print("You have lost")

else:

print("You have won")

### Activity 7.2.2

* Select some of your programs and make them more readable using the rules in the style guide.

### Activity 7.2.3

* Swap a program that you have updated to make more readable with two other people, so that you each have two other programs. Check through what has been done and comment on what has been missed or could be improved further.

# Week 8

## Lesson 1 activities

### Activity 8.1.1

|  |  |
| --- | --- |
| * What are these binary numbers in denary? | |
| **Binary** | **Denary** |
| 0101 | 5 |
| 1010 | 20 |
| 0100 | 4 |
| 1000 | 8 |
| 0011 | 3 |
| 0011 | 3 |
| 1111 | 15 |
| 0010 | 2 |

|  |  |
| --- | --- |
| * What are these denary numbers in binary? | |
| **Denary** | **Binary** |
| 7 | 0111 |
| 3 | 0011 |
| 10 | 1010 |
| 0 | 0000 |
| 9 | 1001 |
| 12 | 1100 |
| 6 | 0110 |
| 13 | 1101 |

### Activity 8.1.2

|  |  |
| --- | --- |
| * Answer these questions. | |
| How many states does a transistor have? | 2 |
| How many bits in a nibble? | 4 |
| What range of unsigned integer numbers can be represented by 8 bits? | 255 |
| What is the position of the most significant bit in a byte? | 8th – mostleft’ |
| How many different values can be represented with 6 bits? | 63 |

### Activity 8.1.3

|  |  |
| --- | --- |
| * What comes next? | |
| 1111 | 00010000 - 16 |
| 0011 1111 | 0100 0000 |

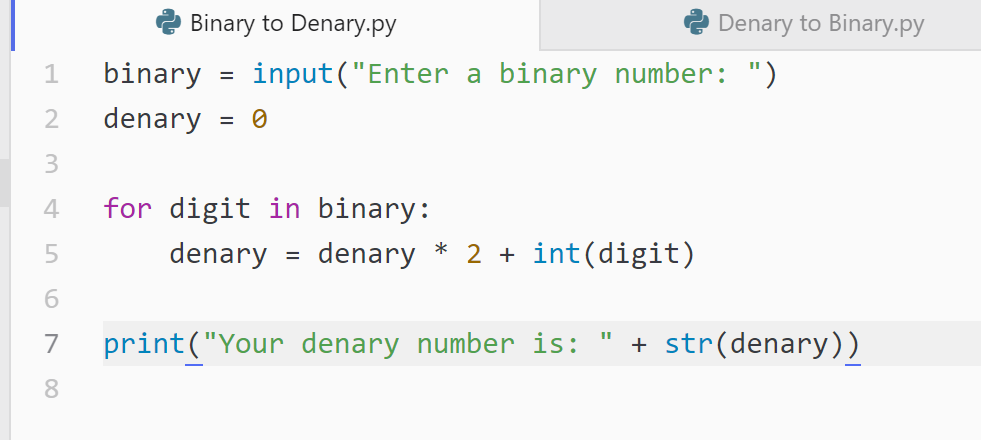
|  |  |
| --- | --- |
| * What comes before? | |
| 1000 | 7 – 0111 |
| 1111 | 15- 01111 |

### Activity 8.1.4

|  |  |
| --- | --- |
| * What are these binary numbers in denary? | |
| **Binary** | **Denary** |
| 00001011 | 11 |
| 11010010 | 210 |
| 01010000 | 80 |
| 01001000 | 72 |
| 10010111 | 151 |
| 11001100 | 204 |
| 10101010 | 170 |
| 11111111 | 255 |

|  |  |
| --- | --- |
| * What are these denary numbers in binary? | |
| **Denary** | **Binary** |
| 65 | 01000001 |
| 100 | 01100100 |
| 78 | 01001110 |
| 29 | 00011101 |
| 223 | 11101001 |
| 80 | 01010000 |
| 14 | 00001110 |
| 55 | 00110101 |

### Activity 8.1.4

* Working in pairs, design and write a simple binary to denary conversion program that inputs a 4-bit binary number and converts it to an unsigned integer.
* 

# Week 8

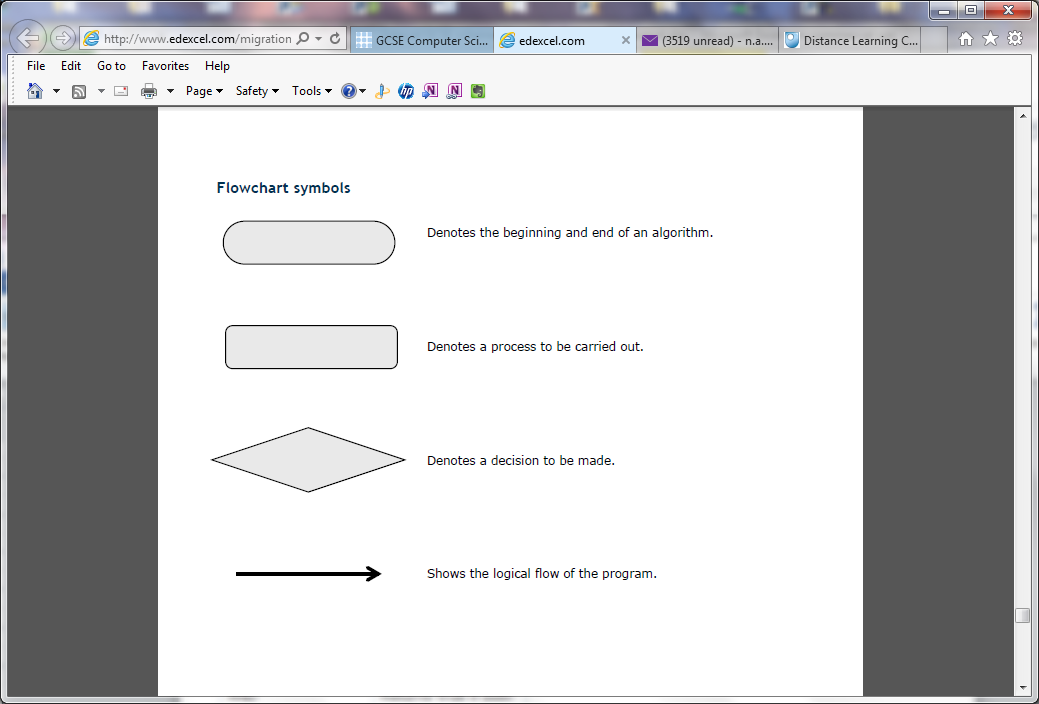
## Lesson 2 activities

### Activity 8.2.1

**Flowcharts**

Flowcharts can be used to represent algorithms.

* Identify the flowchart symbols below.

start/stop, process, decision, flow of code

### Activity 8.2.2

* Study this program:

# password checking

password=input("Please enter password: ")

newPassword=input("Please re-enter password: ")

if password == newPassword:

print("Access granted")

else:

print("Access denied")

* Draw the flowchart for this program.

START

### Activity 8.2.3

Print “access granted”

Print “access denied”

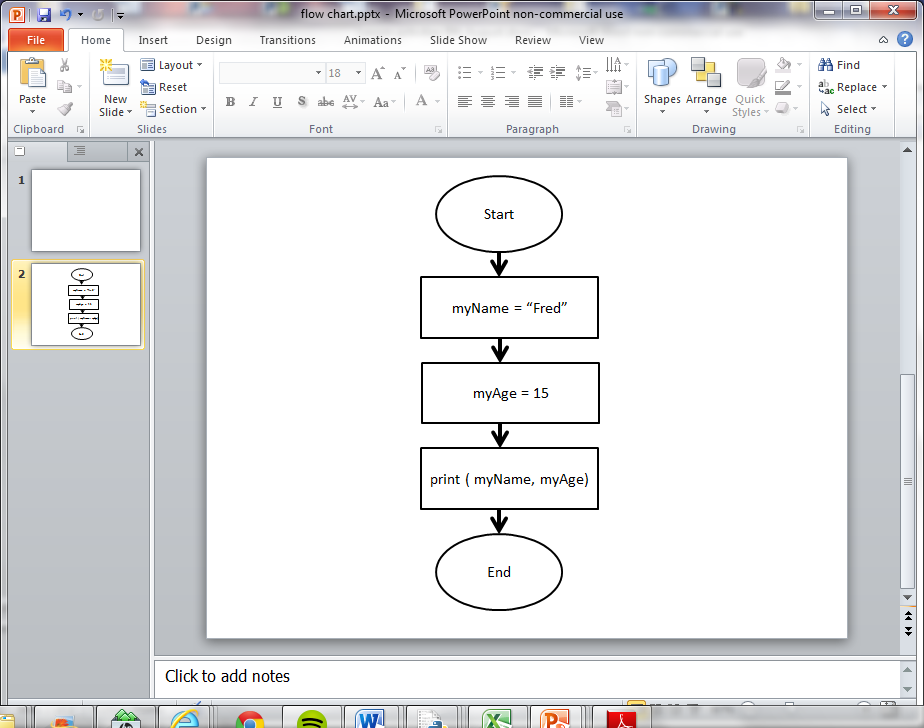
yes

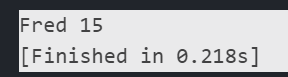
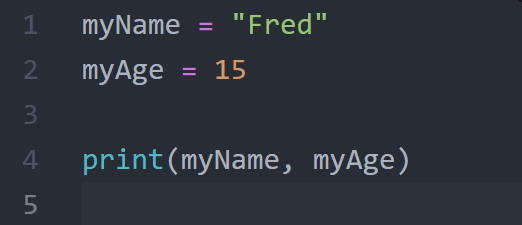
no

Password1 == password2

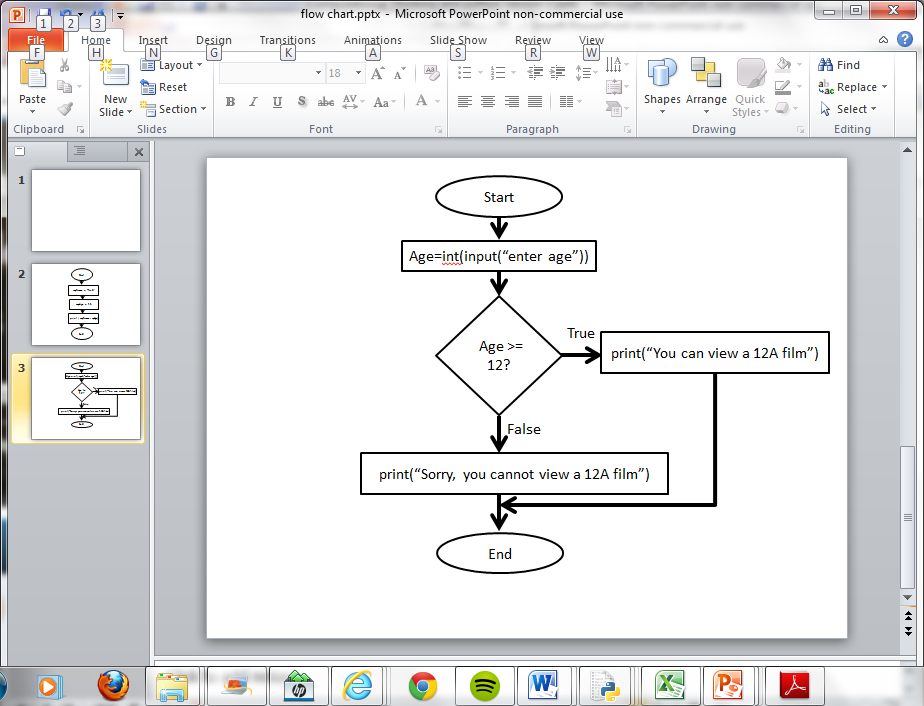
RECEIVE password again From user

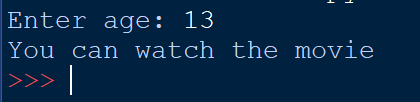
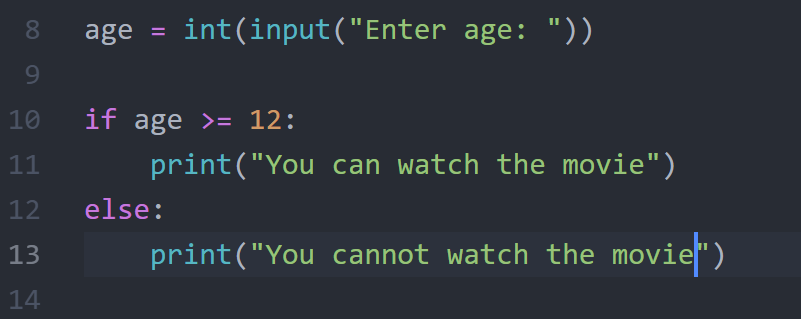
RECEIVE password From user

* Study the flowchart.
* Write a program to implement this flowchart.



### Activity 8.2.4

* Study the flowchart.
* **What does this flowchart do?**
* ask someone their age and decides whether the user is old enough to watch a 12A film
* **Write the program to implement this flowchart.**

**Extension:** Can you extend the program to display whether they can watch 15 films or 18 films?

# Week 9

## Lesson 1 activities

### Activity 9.1.1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| * Add up these binary numbers. * Check your answers by converting them to denary. | | | | | |
| 0111 |  | 0101 |  | 1111 |  |
| 0100 + |  | 0110 + |  | 0111 + |  |
|  |  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0011 |  | 1011 |  | 0111 |  |
| 0011 + |  | 1111 + |  | 0010 + |  |
|  |  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1110 |  | 01111 |  | 01110 |  |
| 0111 + |  | 10101 + |  | 10111 + |  |
|  |  |  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 10101010 |  | 10101100 |  |
| 00110010 + |  | 00010010 + |  |
|  |  |  |  |

**Activity 9.1.2**

Apply the following shifts:

|  |  |
| --- | --- |
|  | **Arithmetic shift 1 Left** |
| 00101100 |  |
| 10011100 |  |
| 00101000 |  |
| 00100111 |  |

|  |  |
| --- | --- |
|  | **Arithmetic shift 2 Right** |
| 00101100 |  |
| 10011100 |  |
| 00101000 |  |
| 00100111 |  |

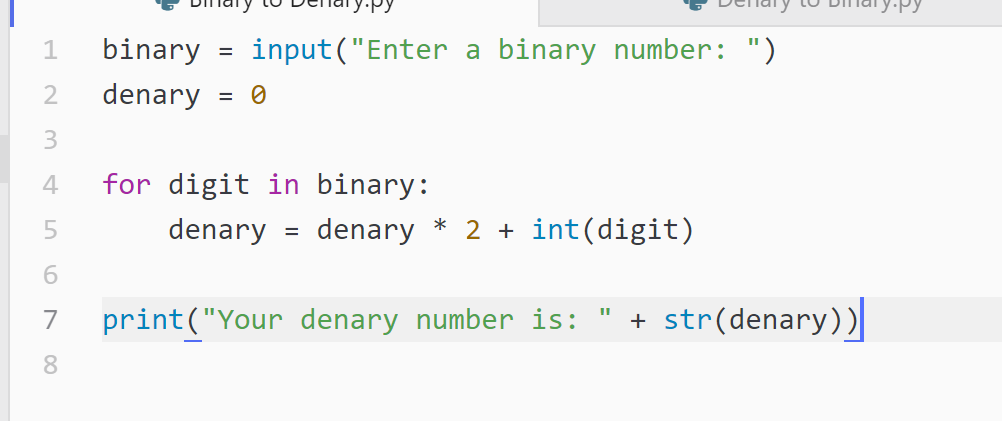
|  |  |
| --- | --- |
|  | **Logical shift 1 Right** |
| 00101100 |  |
| 10011100 |  |
| 00101000 |  |
| 00100111 |  |

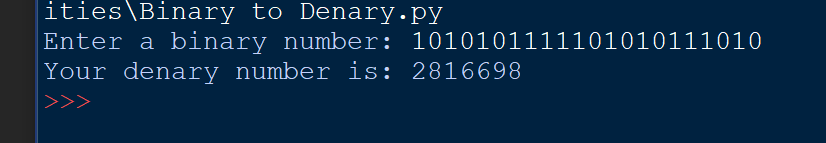
|  |  |
| --- | --- |
|  | **Logical shift 1 Left** |
| 00101100 |  |
| 10011100 |  |
| 00101000 |  |
| 00100111 |  |

**Activity 9.1.3 (homework)**

* Modify the program you created in Activity 8.1.4 so that it can convert a binary number of any length into denary.

*Hint: You will need to use the exponential operator.*

**

**

# Week 9

## Lesson 2 activities

### Activity 9.2.1

**Boolean Operators**

* Complete the table using the logical (Boolean) operators AND, OR and NOT.

|  |  |
| --- | --- |
| Symbol | Description |
| **And** | Returns true if both conditions are true. |
| **Or** | Returns true if any of the conditions are true. |
| **not** | Reverses the outcome of the expression; true becomes false, false becomes true. |

**Activity 9.2.2**

* Predict the answer to the conditions and then use the interactive shell to test your answer.

|  |  |  |
| --- | --- | --- |
| **Condition** | **Your prediction  (true or false?)** | **Result** |
| (78 == 10) **or** (6 == 7) |  |  |
| (78 == 10) **or** (6 == 6) |  |  |
| (78 == 10) **and** (6 == 6) |  |  |
| (1 < 10) **and** (2 < 10) |  |  |
| (1 < 10) **or** (2 < 10) |  |  |
| not ( 5 == 5) |  |  |
| not (6 < 4) |  |  |

*Hint: Try asking yourself the questions:*

*Is condition\_1 true* ***OR*** *condition\_2 true – if YES then the answer is true.*

*Is condition\_1 true* ***AND*** *condition\_2 true – if YES then the answer is true.*

### Activity 9.2.3

* Make up some Boolean operator questions of your own and check your answers are correct by using the interactive Python shell.

Here is an example question:

>>>answer = 50

>>> (answer<40) or (answer >80)

True or False?

* Try them out on other people. Make sure you can explain the right answer to them if they need you to.

### Activity 9.2.4

A truth table lists all the possible combinations of true and false outcomes for each condition.

* Complete the truth tables for AND, OR and NOT operators.

**Truth table showing true and false AND conditions**

|  |  |  |
| --- | --- | --- |
| **Input 1** | **Input 2** | **Output** |
| false | false |  |
| true | false |  |
| false | true |  |
| true | true |  |

**Truth table showing true and false OR conditions**

|  |  |  |
| --- | --- | --- |
| **Input 1** | **Input 2** | **Output** |
| false | false |  |
| true | false |  |
| false | true |  |
| true | true |  |

**Truth table showing true and false NOT conditions**

|  |  |
| --- | --- |
| **Input** | **Output** |
| false |  |
| true |  |
| false |  |
| true |  |

### Activity 9.2.5

* Write a program that asks the user to enter a year group and tells them which key stage that year group is in.

|  |  |
| --- | --- |
| **Key stage** | **Year group** |
| Key stage 1 | Years 1 and 2 |
| Key stage 2 | Years 3, 4, 5 and 6 |
| Key stage 3 | Years 7, 8 and 9 |
| Key stage 4 | Years 10 and 11 |

* **Extension:** What happens if the year group entered is greater than 11? Can you improve your program to cope with this?

### Activity 9.2.6

**Python commands colour coding**

The Python language is colour coded.

* Complete this table.

|  |  |  |
| --- | --- | --- |
| **Colour coding** | **What does it show** | **Example** |
| green |  |  |
| purple |  |  |
| black |  |  |
| orange |  |  |
| red |  |  |

# Week 10

## Lesson 1 activities

### Activity 10.1.1

|  |  |
| --- | --- |
| * Convert these sign and magnitude numbers into denary. The first one is done for you. | |
| **Sign and magnitude binary number** | **Denary** |
| 1100 1101 | -(64+8+4+1) = -77 |
| 0001 1111 |  |
| 1000 1010 |  |
| 0101 1100 |  |
| 1000 0000 |  |
| 1111 1111 |  |
| 0111 1111 |  |

### Activity 10.1.2

|  |  |
| --- | --- |
| * Convert these two’s complement numbers into denary. The first one is done for you. | |
| **Two’s complement binary number** | **Denary** |
| 1100 1101 | -128+64+8+4+1 = 51 |
| 0001 1111 |  |
| 1000 1010 |  |
| 0101 1100 |  |
| 1000 0000 |  |
| 1111 1111 |  |
| 0111 1111 |  |

*Hint: The most significant bit is -128.*

### Activity 10.1.3

|  |  |
| --- | --- |
| * Add your own numbers and work out the answers. | |
| **Sign and magnitude binary number** | **Denary** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

### Activity 10.1.4

|  |  |
| --- | --- |
| * Add your own numbers and work out the answers. | |
| **Two’s complement binary number** | **Denary** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Week 10

## Lesson 2 activities

### Activity 10.2.1

**Pseudo-code**

Pseudo-code, or ‘mock’ code, is another way of describing a program. You will need a copy of the pseudo-codes used in this course (*Appendix 1* of the specification) to answer these questions.

* What does this pseudo-code do?

RECEIVE myName FROM (STRING) KEYBOARD

RECEIVE myAge FROM (INTEGER) KEYBOARD

SET AgeInTen TOmyAge + 10

SEND myName “will be” AgeInTen “in 10 years’ time” TO DISPLAY

* Write the Python code for this program.

### Activity 10.2.2

* What does this pseudo-code do?

SET score TO 119

IF score < 50 THEN SEND “You have lost” TO DISPLAY

ELSE SEND “You have won” TO DISPLAY

END IF

* Write the Python code for this program.

### Activity 10.2.3

Your task is to create a program to ask a user to input 3 integer numbers, calculate the total and the average of the number and display the results.

* Create a flowchart for this task.
* Write pseudo-code for the same task.
* Write the program.

# Week 11

## Lesson 1 activities

### Activity 11.1.1

|  |  |
| --- | --- |
| * Convert these hexadecimal numbers into binary. The first one is done for you. | |
| **Hexadecimal** | **Binary** |
| 3B | 0011 1011  00111011  =59 |
| 26 |  |
| FA |  |
| 61 |  |
| AA |  |
| 8C |  |

### Activity 11.1.2

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | * Convert these binary numbers into hexadecimal. The first one is done for you. | | | Binary | Hexadecmial | | 10101010 | 10101010  1010 1010  A A  AA | | 01001111 |  | | 10000001 |  | | 01110001 |  | | 11110000 |  | | 01001100 |  | | 0111 1111 |  | |

### Activity 11.1.3

* Look at this website: [www.html-color-codes.com](http://html-color-codes.com/). You can flick between hex and decimal (denary) versions of the colours.
* What is the relationship between the two versions?
* Now create a word document and insert a shape. Find the RGB colours of the shape. Fill and change by typing in values from the decimal version.

# Week 11

## Lesson 2 activities

### Activity 11.2.1

**Strings**

* Write the index position for each character in the string.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| H | e | l | l | o |  | w | o | r | l | d | ! |
|  |  |  |  |  |  |  |  |  |  |  |  |

* What character is at index 3?
* What character is at index 5?
* What character is at index 0?
* What character is at index 11?
* How many characters are there in this string?

### Activity 11.2.2

Use the interactive shell to explore how strings can be concatenated.

* Make a **variable** called FirstName and set its value to the string “Fred”.
* Make a **variable** called SecondName and set its value to the string “Smith”.
* Make a **variable** called FullName and set its value to the value of the **variables** FirstName plus SecondName.
* Print the value of the **variable** FullName.

### Activity 11.2.3

**Slicing strings**

Parts of a string can be extracted (or sliced) by giving the index location in the format:

[start position: end position]

Start position is the index at which the slice starts. End position is the index AFTER the last index required.

Use the interactive shell to explore how strings can be sliced.

* Make a variable called word and set its value to the string ‘PIZZA’.

>>>word="PIZZA"

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| P | I | Z | Z | A |
| 0 | 1 | 2 | 3 | 4 |

* What do these commands do?
  + >>>word[2:5]
  + >>>word[1:2]
  + >>>word[0:3]
  + >>>word[0:5]

*Hint: Remember that the index starts at zero not one.*

### Activity 11.2.4

A variable has been created and assigned the string “watch #bbcclick today”

myVariable = “watch #bbcclick today”

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| w | a | t | c | h |  | # | b | b | c | c | l | i | c | k |  | t | o | d | a | y |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |

* What command would you use to extract the following slices?
  + #bbcclick
  + watch
  + today
* Check your answers using the interactive shell.

### Activity 11.2.5

**Using string methods**

Any variable with a string assigned is a member of the class called string. String methods can be used to manipulate that string.

* Explore string methods by working through these commands in the interactive shell and describing what happens.
  + >>>myVariable="A million shards of glass"
  + >>>myVariable.upper()
  + >>>myVariable.replace("a","x")
  + >>>myVariable.title()
  + >>>myVariable.swapcase()

### Activity 11.2.6

**The string .format method revisited: using the string .format method to format output**

There are two ways to output information to the screen. Using the print() function or by giving the expression.

* Try these now:

>>>myName=’Sam Smith’

>>>print(myName)

Sam Smith

>>>myName

'Sam Smith'

The string .format is used to give more control over formatting.

It allows you to use placeholders and then specify the variable you want to print in the string. The placeholders are shown using {}.

>>>first="Sam"

>>>second="Smith"

>>>print("The best music is from {0} {1}!".format(first,second))

The best music is from Sam Smith!

It also allows you to choose how to format the output.

# Week 12

## Lesson 1 activities

### Activity 12.1.1

* Who has been called the first programmer? Note down some facts about this person below:

### Activity 12.1.2

* Research and complete the table below:

|  |  |  |  |
| --- | --- | --- | --- |
| **High level programming language** | **When was it created?** | **Who invented it?** | **Find out something special about the language (e.g. is it used for a particular purpose?)** |
| FORTRAN |  |  |  |
| Java |  |  |  |
| C |  |  |  |
| Visual Basic |  |  |  |
| PHP |  |  |  |
| Python |  |  |  |
| Ruby |  |  |  |

### Activity 12.1.3 (homework)

* Find three different websites that analyse the popularity of programming languages. Note down the top 10 from each site – you will use this in later lesson. Put the URL of each website in the top row.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 6 |  |  |  |
| 7 |  |  |  |
| 8 |  |  |  |
| 9 |  |  |  |
| 10 |  |  |  |

# Week 12

## Lesson 2 activities

### Activity 12.2.1

**Lists (arrays)**

* Complete this table to summarise the list commands.

|  |  |
| --- | --- |
| **Things to do with lists** | **Commands** |
| Create a list |  |
| Reference an item in a list |  |
| Delete an item in a list |  |
| Append an item to the end of a list |  |

### Activity 12.2.2

**Lists (arrays)**

A list is a data structure that stores a set of elements. Lists are assigned to a name using square brackets.

>>>mylist=["apple","oranges","lemon","pear","lime"]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| apples | oranges | lemon | pear | lime |
| 0 | 1 | 2 | 3 | 4 |

Each element in a list has an index location. The first element of the list is in position zero (0).

Elements of a list are referenced using their index location (an integer number).

List name[index]

A range of elements can be displayed using

[start index: end index]

Start index is the position to start at (remember that indexing starts at zero). End position is the index AFTER the index required.

* Make this list and experiment with the list commands.

>>>mylist=["apple","orange","lemon","pear","lime"]

* What does **mylist[1]** display?
* What does **mylist[1:3]** display?
* What does **mylist[-1]** display?
* What command will display just apple?
* What command will display lemon and pear?
* Make a new list called myfood containing your five favourite foods.
* Display the whole list.
* Display the food item at index position 3.
* Display the food item at index position 0.
* Display the food items at index position 1 to 4.

### Activity 12.2.3

**Using lists**

* Make the list that contains the class marks for Amy Jones.

Marks = ['Amy', 'Jones', 'English', 67, 'Maths', 76, 'Computer Science', 96]

* The English teacher has entered Amy’s mark incorrectly; it should be 72 not 67. Alter this item in the list.
* Add the mark for Physics to the end of the list. “Physics”, 65
* Remove “Maths” and the score 76 from the list.
* Write a program to find the average score for the three subjects (English, Computer Science and Physics).

### Activity 12.2.4 (homework)

**Using Python docs help**

* Select help/python docs then select the Python tutorial and go to 3.1.4 Lists.
* Read through the discussion of lists and try out the examples. Make a note of three more facts about lists to share in the next lesson.

Python is a very powerful programming language, which is used in universities and commercial organisations. You do not need to know all the details provided in the Python docs but, with practice, you should be able to find information about Python that can be very useful.

# Week 13

## Lesson 1 activities

### Activity 13.1.1

* Use the labels below to complete the diagram on the next page. The diagram shows how high level programs are translated into machine code so that they can be run by the CPU.

**Labels**

* Python (high level language)
* Machine code
* Compiler
* Assembly language (low level language)
* Assembler

### Activity 13.1.2



**MOV R1, #50**

**MOV R2, #60**

**ADDS R0, R1, R2**

**MOV R7, #1**

**1111000011110000**

**1011100011100010**

**1010010010010100**

* Write a mock translator program using the data in the table below.
  + Prompt the user to choose a high level or assembly language, and translate to the next lower level and display the result.

|  |  |  |
| --- | --- | --- |
| **High level language command** | **Assembly language** | **Machine code** |
| If x == y | CMP | 10110011 |
| Total = num1 + num2 | ADD | 10000101 |
| Minus = num2 – num1 | SUB | 10010000 |

* **Extension (homework):** enter a high level language command and translate it into machine code in two stages.

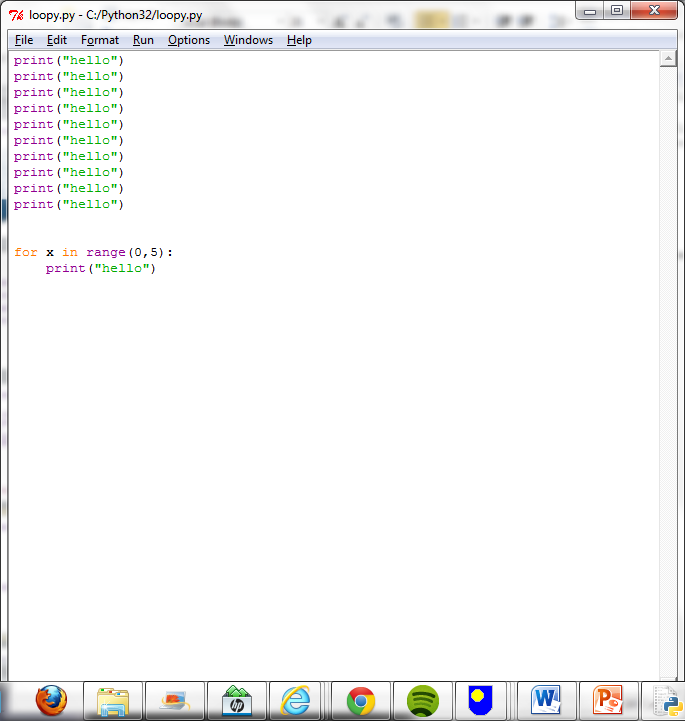
# Week 13

## Lesson 2 activities

### Activity 13.2.1

**For loops**

* Copy and run this program.



* Write a program that prints out ‘I like Ellie Goulding’ ten times.

**Activity 13.2.2**

* Copy and run this program:

for number in range(10):

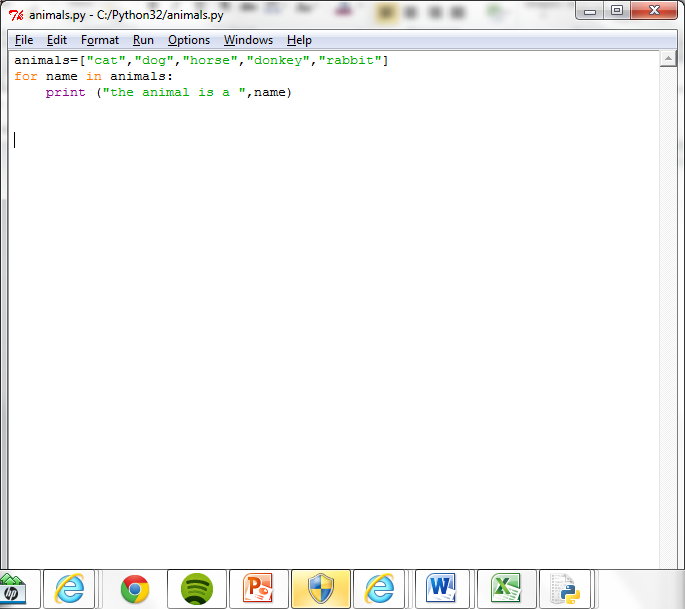
print(number)

* Explain what this program does.
* Write a program that prints the numbers from 1 to 15.
* Write a program that prints the numbers from 1 to 8, with the square of each number, both on the same line.
* Write a program that prints out the 9 times table (up to 20 x 9).
* Write a program that asks the user which times table they want and then displays that table.
* Write a program that asks for a start value and an end value and then counts in fives between those numbers.

*Hint: range(start value, end value, step)*

### Activity 13.2.3

* Copy this program into a file and run the program.



* Explain the purpose of the ‘name’ variable.
* Write a program that creates a list of things you would take to a desert island. The program should then display each item in the list a line at a time.

### Activity 13.2.4

* Re-arrange the program statements to write a program that will print out the names of the animals that begin with the character ‘c’.

print(next)

if next[0] == "c" :

myList=["cat","dog","cow","donkey","rabbit","canary"]

for next in myList:

# Week 14

## Lesson 1 activities

### Activity 14.1.1

|  |  |
| --- | --- |
| * Use the function to identify the hardware component. | |
| **Function** | **Hardware component** |
| A temporary storage area for data and program instructions while a program is running. |  |
| A microprocessor that carries out the instructions in computer programs by performing arithmetic and logic operations, and controls inputs and outputs. |  |
| A persistent data storage area for data and program instructions. |  |
| To provide connections to the input and output peripherals such as printers, mouse, keyboard, touch screen, speakers, networks, and so on. |  |

### Activity 14.1.2

* This diagram represents the way in which buses connect the main hardware components in a digital computer. Copy and complete the diagram using these labels:
* CPU
* memory
* input/output devices
* data, control and address bus

### Activity 14.1.3

* From the function given, identify and insert the name of the correct bus.

|  |  |
| --- | --- |
| **Function** | **Type of bus** |
| Sends and receives signals that control the CPU and other parts of the computer system |  |
| Carries the address of memory locations used to store data and program instructions |  |
| Transfers the binary data around the computer |  |

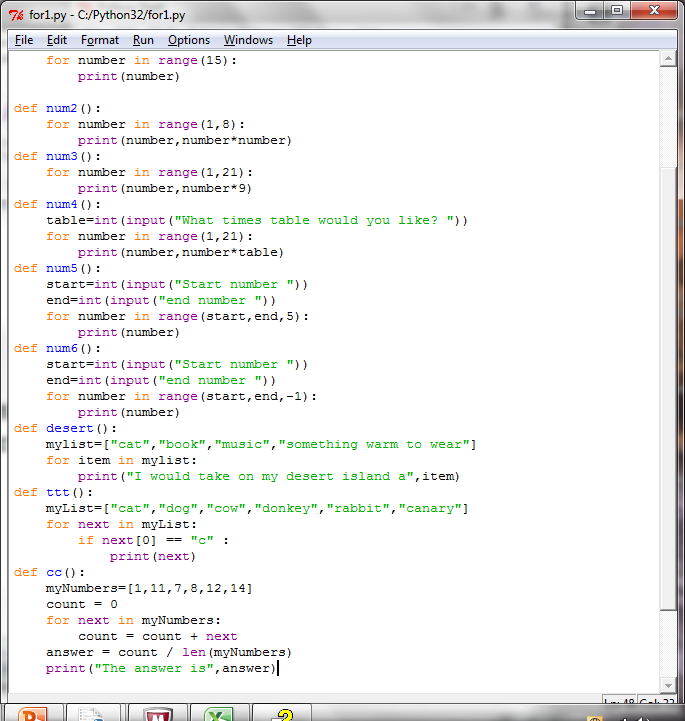
### Activity 14.1.4

* Label the following on the picture of the Raspberry Pi:
* SD card (memory)
* CPU processor and RAM memory
* ethernet network connection
* two USB connectors (mouse and keyboard)
* HDMI connector for monitor
* audio
* power supply

# Week 14

## Lesson 2 activities

### Activity 14.2.1

* Study this program:
* What will the program do?
* Explain what the variable next is used for?
* Explain what the variable count is used for?
* What does len(myNumbers) do?
* What happens if you add or append more numbers in the list?
* If a print statement was added to the for loop, as shown below, what would be displayed?

### Activity 14.2.2

* Write a program that finds the largest number in this list:

myNumbers=[19,6,7,9,2,25,16]

### Activity 14.2.3

* Write a program to calculate the average popularity position for the languages you identified in Activity 12.1.3. First create a flowchart to show how you will solve this problem.

# Week 15

## Lesson 1 activities

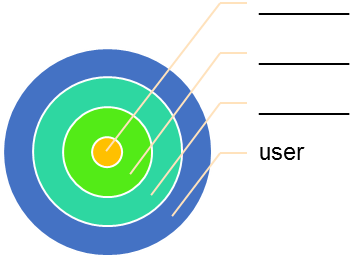
### Activity 15.1.1

* Windows is an example of an operating system. List as many other operating systems as you can. Aim for a list of at least five operating systems. Discuss with other class members or use an internet search if you need to.

|  |
| --- |
| *e.g. Windows* |
|  |
|  |
|  |
|  |
|  |

### Activity 15.1.2

* Assuming that the user is the outer circle, use the labels ‘hardware’, ‘application software’ and ‘operating system’ for the other circles to show the role of the operating system.



### Activity 15.1.3

* Using a non-networked computer and the operating system of your choice, do the following:
* make a hierarchy of directories and files
* remove files and directories
* rename files and directories
* change the current working directories
* view the current working directory.

Here is a list of useful Raspbian (Unix) and Command Line Interpreter (LX Terminal) file and directory commands.

|  |  |
| --- | --- |
| **Commands** | **Function** |
| pwd | Print working directory – display the current directory |
| cd directory | Change directory |
| mkdir directory | Make a new directory |
| rm file | Remove a file |
| touch file | Create a new empty file |
| cat file | Catalogue a file – list the contents of a file |
| ls -l | List the files in the current working directory displaying all the details about the files |

### Activity 15.1.4

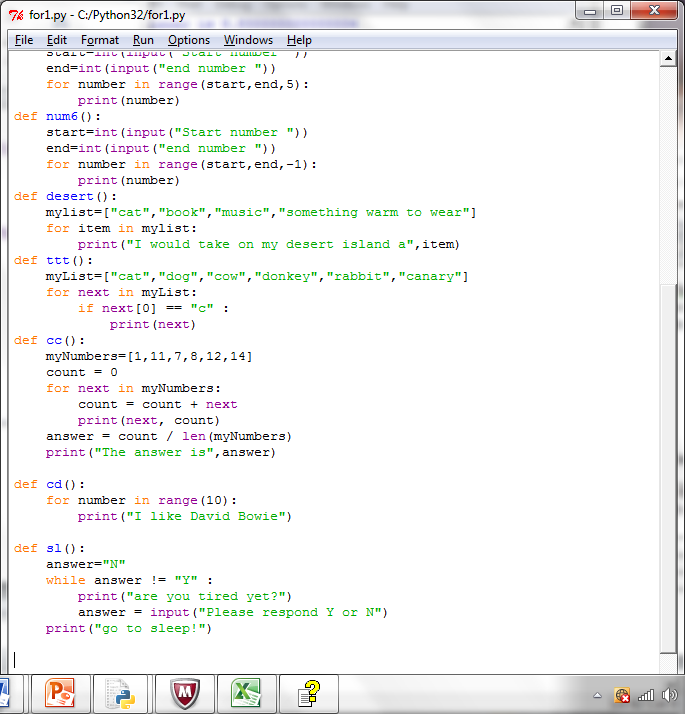
* Bring up the task manager in whatever operating system you are using (in Windows, control/alt/delete).
* Find the networking information within the task manager (in Windows, choose the networking tab).
* What happens when the user accesses the network, for example by opening a web page or accessing information on a server?

# Week 15

## Lesson 2 activities

### Activity 15.2.1

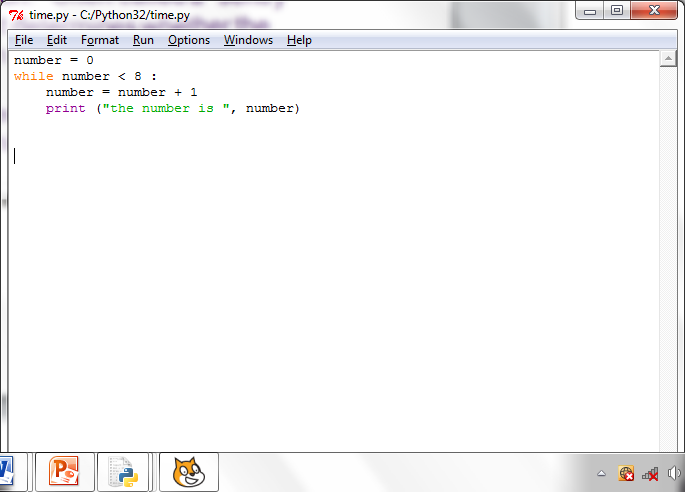
**While command**

* What does this while command do?
* What is the condition that is being tested and how does it change?

### Activity 15,2.2

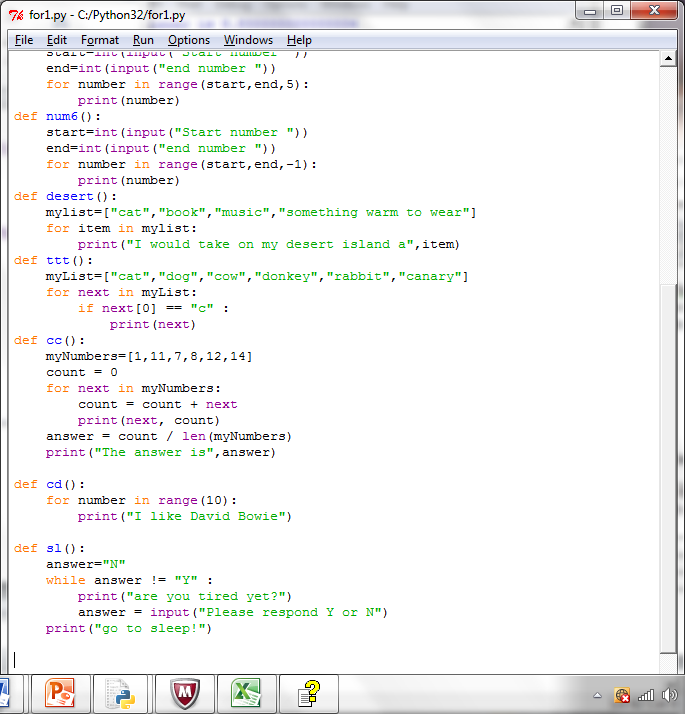
* Write a program that asks the user if they are hungry. While they reply ‘N’ the program repeats the question, showing how many times they have replied ‘N’. When they reply ‘Y’, the program tells them to get something to eat.

### Activity 15.2.3

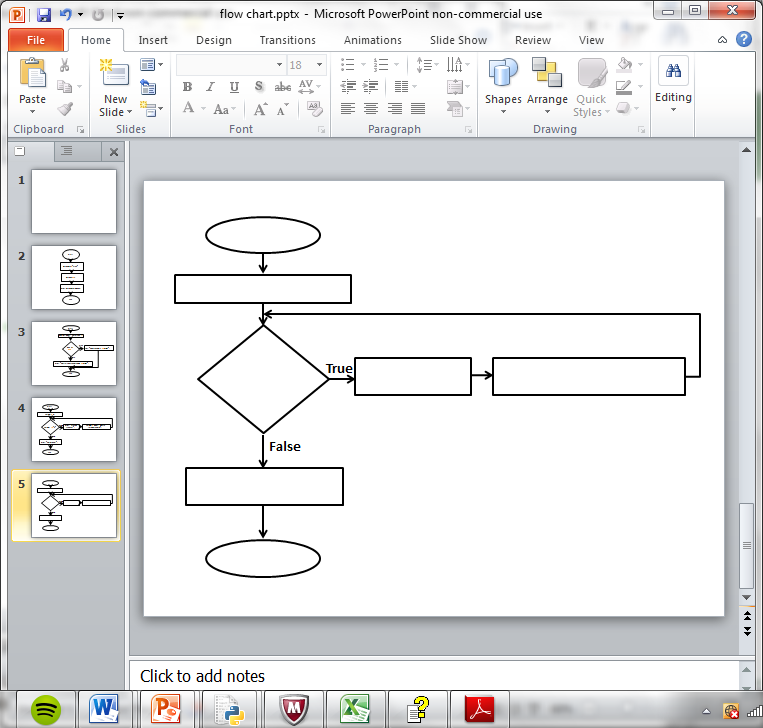
* What does this program do?
* What is the ‘sentry’ variable?
* What is the condition?

### Activity 15.2.4

**Flowcharts and while command**

Flowcharts can be used to represent programs.

* Complete the flow chart for this program:



# Week 16

## Lesson 1 activities

### Activity 16.1.1

* Open up the operating system’s task manager (in Windows, control/alt/delete) and find the list of applications running (in Windows, choose the Applications tab).
* Now find the information on the processes running (in Windows, choose the Processes tab). Scroll up and down the list to see how many processes it takes to run the operating system and any applications you currently have open.
* Now open up a few applications, e.g. a web browser, word processor, spreadsheet software, a graphics package.
* Answer these questions:
* What happens to the list of applications?
* What happens to the number of processes running on the computer?
* Try to explain why there are more processes running than applications running.

### Activity 16.1.2

Your teacher will run this activity. The instructions are here for information only.

Process scheduling: an ‘unplugged’ activity to simulate the scheduling of processes

*Requirements*

* Large stick notes
* White board
* Two tables
* Area for process queue
* Stop watch
* Pens
* Two students who represent the operating system (‘OS1’ will allocate the process ID and time slots, ‘OS2’ will send the next process to the ‘CPU’)
* One student who represents the ‘CPU’
* One student who represents the ‘user’ who is running the programs
* Sixteen other students who represent the programs and applications and who will be the ‘processes’ (just add more programs/applications depending of number of students in group)

*Starting positions*

* The CPU should sit at a table with a pen and stop watch.
* OS1 needs a pen and an empty process table drawn on a whiteboard (add more rows as required:

|  |  |
| --- | --- |
| **Process ID** | **Name of program or application** |
|  |  |
|  |  |
|  |  |
|  |  |

* Allocate a queue area where the processes will stand in order while they wait to be ‘executed’ after they have been allocated a process ID.
* The user sits at table with a pen and sticky notes.

*Running the programs*

* The user starts by writing on a sticky note the name of a program or application, in the order given in the table below. The user must write program and application names in the same order as they appear in the table below.

|  |
| --- |
| Word processor |
| Web browser |
| Sound program |
| Myprog.py |
| Sort.py |
| Word processor |
| Myprog.py |
| Spreadsheet |
| Sort.py |
| Word processor |
| Spreadsheet |
| Web browser |
| Sound program |
| Web browser |
| Sound program |
| Sort.py |

* The user hands the sticky note on which he/she has written a program or application name to a student who will be that process.
* The process goes to OS1 who allocates the next free process ID from the process table. OS1 also allocates the correct number of time slots needed to complete the process, according to the table below:

|  |  |
| --- | --- |
| **Type of process** | **Time slots** |
| word processor | 1 2 3 |
| myprog.py | 1 2 |
| sort.py | 1 |
| spreadsheet | 1 2 3 4 |
| web browser | 1 2 3 |
| sound program | 1 2 3 4 5 |

* So, if the process hands OS1 a sticky note that says ‘Word processor’, OS1 adds a process ID and three time slots ‘1, 2, 3’ to the sticky note and hands it back to the process. OS1 must also update the process table on the whiteboard.
* The process joins the back of the queue holding up their sticky note.
* OS2 is constantly looking at the queue waiting for a process with a process ID. They send the process at the top of the queue to the CPU table.
* Each process gets 10 seconds to execute with the CPU – when the process arrives at the table, CPU starts the stopwatch, crosses one time slot off the sticky note and then says, ‘Time’s up!’ after 10 seconds. At this point, the process must return to the back of the queue.
* The process should check their sticky note. If all their time slots have been crossed off, they should shout their process ID and ‘Finished!’ and then sit down.
* When OS1 hears that a process has finished, they should remove that process from the process table.
* If the process still has time slots left they go to the back of the queue.
* Repeat until all the processes have finished.

*Alternatives*

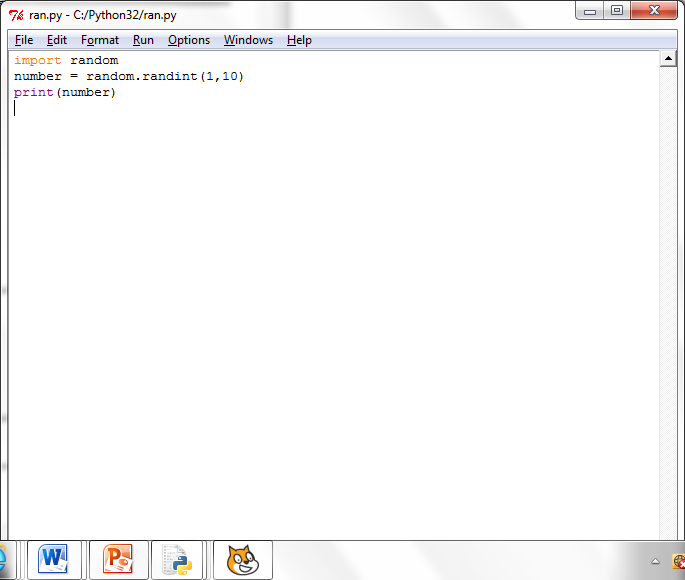
* Change the length of the time slice (make it more or less than 10 seconds).
* Add more or different programs/applications.
* Give the processes a priority that alters their position in the queue.

# Week 16

## Lesson 2 activities

### Activity 16.2.1

**random() function**

* Copy and run this program code:
* Run the program a few times. What does it do?
* What is meant by ‘import random’?
* What happens if you alter the values to 1, 100?

### Activity 16.2.2

* Write a program that acts like a dice. After each ‘throw’ of the dice it should ask if the player wishes to continue and stop when they enter ‘Y’.

### Activity 16.2.3

* Write a program that gives the user ten chances to guess a number between 1 and 10. It then compares their answer with a randomly generated number and stops the program when the number generated matches the number guessed or when the user is out of guesses.