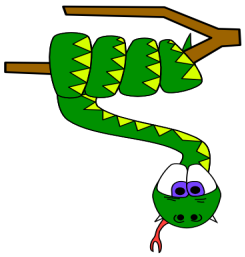


Python Lesson 4

Debugging, strings, modules, while and functions

Suffolk One, Ipswich, 4:30 to 6:00
Tuesday Feb 11
Nicky Hughes

NETWORK OF EXCELLENCE COMPUTER SCIENCE TEACHING

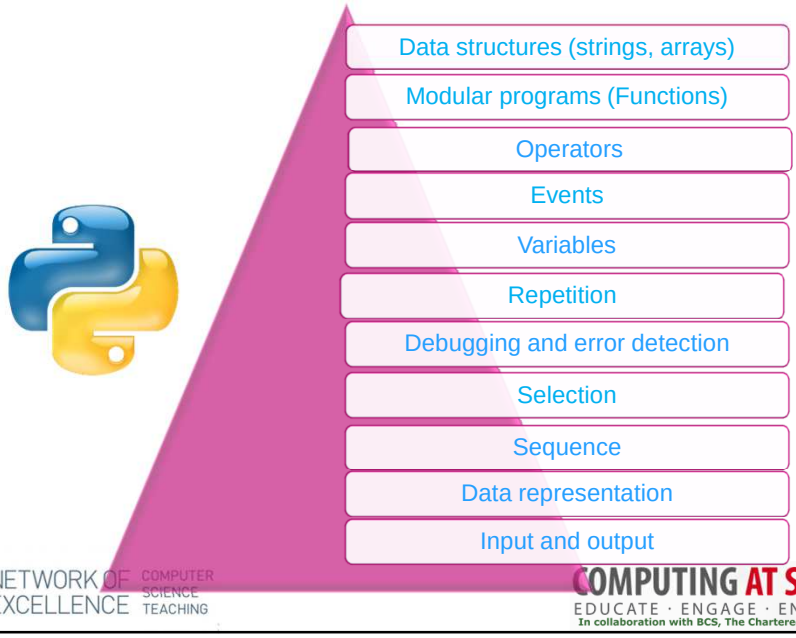
COMPUTING AT SCHOOL
EDUCATE · ENGAGE · ENCOURAGE
In collaboration with BCS, The Chartered Institute for IT

All programs can be represented in terms of sequence, selection and iteration.

NETWORK OF EXCELLENCE COMPUTER SCIENCE TEACHING

COMPUTING AT SCHOOL
EDUCATE · ENGAGE · ENCOURAGE
In collaboration with BCS, The Chartered Institute for IT

Computational thinking – programming building blocks



NETWORK OF EXCELLENCE COMPUTER SCIENCE TEACHING

COMPUTING AT SCHOOL
EDUCATE · ENGAGE · ENCOURAGE
In collaboration with BCS, The Chartered Institute for IT

Debugging, strings, modules, while and functions

Learning outcomes

- Review last week lesson
- Be able to use the IDLE debugger
- Be able to use **strings** at the interactive prompt
- Be able to use **strings** in scripts
- Be able to import modules and use pre-written functions (random, maths, webbrowser)
- Be able to use **while** loops
- Be able to write and run your own **function**

NETWORK OF EXCELLENCE COMPUTER SCIENCE TEACHING

COMPUTING AT SCHOOL
EDUCATE · ENGAGE · ENCOURAGE
In collaboration with BCS, The Chartered Institute for IT

Debug this program…
When do I wear which colour socks?

```
myList= {"red","green","blue","yellow","black"}

days = [Monday","Tuesday","Wednesday","Thursday","Friday"]

for item in range(1,5):
     print("I wear","myList[item]","socks","on", days[item])
```

NETWORK OF EXCELLENCE COMPUTER SCIENCE TEACHING

COMPUTING AT SCHOOL
EDUCATE · ENGAGE · ENCOURAGE
In collaboration with BCS, The Chartered Institute for IT

Debug this program…
When do I wear which colour socks?

```
myList= {"red","green","blue","yellow","black"}

days = [Monday","Tuesday","Wednesday","Thursday","Friday"]

for item in range(1,5):
     print("I wear","myList[item]","socks","on", days[item])
```

```
myList= ["red","green","blue","yellow","black"]

days = ["Monday","Tuesday","Wednesday","Thursday","Friday"]

for item in range(5):
     print("I wear",myList[item],"socks","on", days[item])
```

NETWORK OF EXCELLENCE COMPUTER SCIENCE TEACHING

COMPUTING AT SCHOOL
EDUCATE · ENGAGE · ENCOURAGE
In collaboration with BCS, The Chartered Institute for IT

Complete the trace table for this code.

```
for value in range(1,5):
    number1 = value
    number2 = value * value
    number3 = value / 2
    print(number1, number2, number3)
```

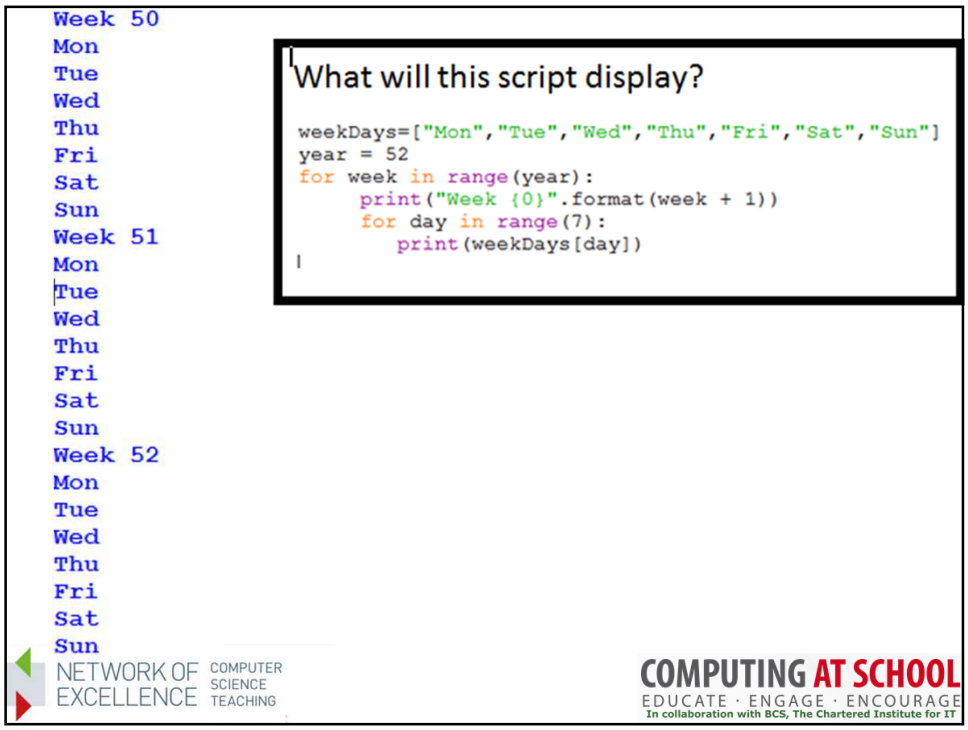
value	number1	number2	number3

Lists and loops

What will this script display?

```
weekDays=["Mon","Tue","Wed","Thu","Fri","Sat","Sun"]
year = 52
for week in range(year):
    print("Week {0}".format(week + 1))
    for day in range(7):
        print(weekDays[day])
```

NETWORK OF EXCELLENCE COMPUTER SCIENCE TEACHING
COMPUTING AT SCHOOL
EDUCATE · ENGAGE · ENCOURAGE
In collaboration with BCS, The Chartered Institute for IT



Brackets in Python?

What uses () brackets?

What uses [ ] brackets?

What uses { } brackets?

NETWORK OF EXCELLENCE COMPUTER SCIENCE TEACHING
COMPUTING AT SCHOOL
EDUCATE · ENGAGE · ENCOURAGE
In collaboration with BCS, The Chartered Institute for IT

Brackets in Python?

What uses () brackets? Functions and tuples

```
>>> print("hello world")
>>> myTuple=("Fred","Bloggs",21,"Exeter")
```

What uses [ ] brackets? Lists eg mylist[3:4]

```
>>> mylist = ["red","green","blue"]
```

What uses { } brackets? String formatting (and also sets)

```
>>> print("{1} and {0}".format(foodOne,foodTwo))
>>> myset={1,2,5,6,7,7,1,2,5,5,5,5}
```


Debugging

A **debugger** is a diagnostic program that helps in the detection, location and corrections of faults (or bugs) in a program.

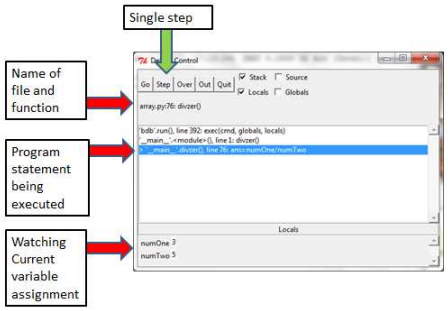
Single-step is when the program is executed one statement at a time under user control.

This allows the user to see the effects of each statement by performing a **step-through** of the program.

NETWORK OF EXCELLENCE COMPUTER SCIENCE TEACHING
COMPUTING AT SCHOOL EDUCATE · ENGAGE · ENCOURAGE In collaboration with BCS, The Chartered Institute for IT

Exploring the IDLE debugger

- Open Python IDLE with an existing program file.
- Start the IDLE debugger using Debug/Debugger. The IDLE shell will display [DEBUG ON].
- To turn off this mode select Debug/Debugger.
- Start running your program. The Debug Control window will be displayed:



NETWORK OF EXCELLENCE COMPUTER SCIENCE TEACHING

COMPUTING AT SCHOOL
EDUCATE · ENGAGE · ENCOURAGE
In collaboration with BCS, The Chartered Institute for IT

The program is now running under the debugger allowing you to single step through the program.

Enter this program and single step through the program – watch the variables and commands

```
total = 0
for value in range(1,10):
        total = total + value
        print(total, value)
print(total)
```

Remember to turn OFF the debugger when you have finished.

NETWORK OF EXCELLENCE COMPUTER SCIENCE TEACHING

COMPUTING AT SCHOOL
EDUCATE · ENGAGE · ENCOURAGE
In collaboration with BCS, The Chartered Institute for IT

Strings in Python

- A string is a data structure (like lists and tuples) that stores a set of characters.
- Strings are immutable which means they cannot be changed.
- They can be treated like lists with each letter being a separate item.
- The position of each character in a string is called the index, which starts from 0

>>> myStr = “HELLO”

H	E	L	L	0
0	1	2	3	4

Strings in Python

H	E	L	L	O
0	1	2	3	4

- Items in a string can be referenced

[start index: end index]

- The end index is the index **after** the one you want the range to finish on…. **up to but not including…**

[1:3] refers to items at index 1 and 2 which is “EL”

[-1] refers to the item at the end of the string which is “O”

Using string indexes

```
>>> mystr="HELLO"
>>> mystr[2]
'L'
>>> mystr[1:3]
'EL'
>>> mystr[-1]
'O'
>>> mystr[-3:-1]
'LL'
```

H	E	L	L	O
0	1	2	3	4
-5	-4	-3	-2	-1

Things you can do with strings

Command	Explanation
stringa **in** stringb	Returns true if stringa is in stringb
stringa **not in** stringb	Returns true is stringa is not in stringb
len(stringa)	Returns the length of the string
stringa + stringb	Concatenates stringa and stringb
stringa * number	Concatenates stringa number times
str.isalnum()	Returns true if all the characters are alphanumeric otherwise false
str.isdigit()	Returns true is all the characters are digits
str.islower()	Returns true if all lowercase
str.replace(old,new)	Returns a copy of the string with all the occurrences of old replaced by new
str.find(sub)	Returns the lowest index in the string when the substring sub is found

```
>>> strA="Hello World"
>>> strB="W"
>>> strB in strA
True
>>> len(strA)
11
>>> answer = strA + strB
>>> answer
'Hello WorldW'
>>> strA * 19
'Hello WorldHello WorldHello WorldHello WorldHello WorldHello WorldHello WorldHe
llo WorldHello WorldHello WorldHello WorldHello WorldHello WorldHello WorldHello
 WorldHello WorldHello WorldHello WorldHello World'
>>> strA.isalnum()
False
>>> "123abc".isalnum()
True
>>> "567".isdigit()
True
```

```
>>> "r" in "Rabbit"
False
>>> "R" in "Rabbit"
True
>>> "R" not in "Rabbit"
False
```

NETWORK OF EXCELLENCE COMPUTER SCIENCE TEACHING

COMPUTING AT SCHOOL
EDUCATE · ENGAGE · ENCOURAGE
In collaboration with BCS, The Chartered Institute for IT

Challenges using strings

NETWORK OF EXCELLENCE COMPUTER SCIENCE TEACHING

COMPUTING AT SCHOOL
EDUCATE · ENGAGE · ENCOURAGE
In collaboration with BCS, The Chartered Institute for IT

Use the string “MonTueWedThuFriSatSun”
Write a program that lists the 3 letter code for each day, one per line.

```
months="MonTueWedThuFriSatSun"
startValue=0
for item in range(7):
      endValue=startValue+3
      day=months[startValue:endValue]
      print(day)
      startValue=startValue+3
```


The ord() function returns the number for a character (ASCII)
The chr() fuction returns the character represented by a number (ASCII)

We can use this to write a simple Ceasar encryption cipher.

```
# encrypt a ceasar cipher
myString=input("Enter string to be encrypted")
encryptString=""

for character in myString:
      number=ord(character)
      encNumber = number + 1
      encryptString=encryptString + chr(encNumber)
print(encryptString)
```

To decrypt

```
# decrypt a ceasar cipher
myString=input("Enter string to be decrypted")
decryptString=""

for character in myString:
     number=ord(character)
     decNumber = number - 1
     decryptString=decryptString + chr(decNumber)
print(decryptString)
```


Using pre-written functions

- Python has many “built-in functions” that have () after their name and are shown in purple.
- Examples include print(), range(), ord(), chr()
- Python has modules that contain pre-written function libraries that we can import into a program.

Three examples of using modules and pre-written functions:
- Using random numbers (import random) While
- Using mathematical functions (import maths)
- Using browser calls (import webbrowser)

Import random to use random numbers

```
import random
number = random.randint(1,10)
print("my random number is ", number)
```

- Run the program a few times to see how it selects a different random number.
- Change the range from which the random number is selected.

Challenge

Write a program that selects a random number between 0 and 10. It then asks the user to guess their number. If they are correct is says "Well done – you guessed the correct number". Else it says "Wrong answer".

NETWORK OF EXCELLENCE COMPUTER SCIENCE TEACHING
COMPUTING AT SCHOOL
EDUCATE · ENGAGE · ENCOURAGE
In collaboration with BCS, The Chartered Institute for IT

Challenge

Write a program that selects a random number between 0 and 10. It then asks the user to guess their number. If they are correct is says "Well done – you guessed the correct number". Else it says "Wrong answer".

```
import random
number = random.randint(1,10)
answer = int(input("Can you guess my number between 1 and 10? "))
if answer == number:
     print("Well done - you have guessed the correct number")
else:
     print("Wrong answer")
```

NETWORK OF EXCELLENCE COMPUTER SCIENCE TEACHING

COMPUTING AT SCHOOL
EDUCATE · ENGAGE · ENCOURAGE
In collaboration with BCS, The Chartered Institute for IT

While loop

A **while** command loops while some condition remains true.

while <condition_is_true >:
    indented block of commands

The condition in the while loop often uses a variable which must be changed within the while loop (called the "guard" variable).

The guard variable is given a starting value before the while loop starts

The conditions that can be tested are the same as those used in the "if" command".

NETWORK OF EXCELLENCE COMPUTER SCIENCE TEACHING

COMPUTING AT SCHOOL
EDUCATE · ENGAGE · ENCOURAGE
In collaboration with BCS, The Chartered Institute for IT

Lets re-write the guessing game so that it keeps asking the user to enter a number while the number they enter is not the same as the random number.

Lets re-write the guessing game so that it keeps asking the user to enter a number while the number they enter is not the same as the random number.

```
import random
number = random.randint(1,10)
answer = int(input("Can you guess my number between 1 and 10? "))
while number != answer:
     print("Wrong answer - have another go")
     answer = int(input("Guess my number between 1 and 10"))
print("Well done you have guessed the correct number")
```

What is the condition in the while loop?

What is the “guard variable”?

NETWORK OF EXCELLENCE COMPUTER SCIENCE TEACHING

COMPUTING AT SCHOOL
EDUCATE · ENGAGE · ENCOURAGE
In collaboration with BCS, The Chartered Institute for IT

Import math module includes most math functions…. examples

```
import math
print(math.pi)


print(math.sqrt(16))
```


Import webbrowser module

```
import webbrowser
```

webbrowser.open(URL)	opens the URL(uniform resource locator) in the browser.
Webbrowser. open_new_tab	opens the URL in a new tab in the browser

Open the program **webprogram.py** and look at the code.

Amend to go to the URLs of your choice

```
import webbrowser

website1="http://www.bbc.co.uk/sport/0/"
website2="http://www.bbc.co.uk/cbbc/"
website3="http://www.bbc.co.uk/nature/"

print("Which website would you like to visit?")
print("1 to open sport")
print("2 to open cbbc")
print("3 to open nature")

answer = input("Please enter your choice of site: ")

if answer == "1":
    webbrowser.open_new_tab(website1)
if answer == "2":
    webbrowser.open_new_tab(website2)
if answer == "3":
    webbrowser.open_new_tab(website3)
```

Functions in python

- Functions have a name, brackets to pass parameters into the function () and a colon.
- The commands must be indented.

```
def function_name():
    commands
    commands
```

Why functions?
Functions contain pre-written code that we can re-use.
Functions allow us to decompose the program into manageable parts for writing and testing

FUNCTION Terminology	
Function	A subprogram that returns a value.
Return value	The value returned by a subprogram.
Parameter	The names that appear in a function definition when passing data to a function.
Argument	A piece of information/value that is required by a function to perform a task, e.g. function(argument1, argument2)

NETWORK OF EXCELLENCE COMPUTER SCIENCE TEACHING
COMPUTING AT SCHOOL
EDUCATE · ENGAGE · ENCOURAGE
In collaboration with BCS, The Chartered Institute for IT

Challenge: write a program which writes the verses for the “Wheels on the bus” using functions.

Each time through this song the name and the actions changes.

The wheels on the bus go round and round
the wipers on the bus go swish and swish
The horn on the bus go beep and beep
The lights on the bus go flash and flash
The wheels on the bus go round and round


```
def verse(name, action):
    print("The ", name,"on the bus go", action, "and", action)
    print("All day long")
    print("")


def main():
    verse("wheels","round")
    verse("wipers","swish")
```

To run use
>>> main()

Name the two functions?

What are the parameters to the verse function?

What are the arguments to the verse function?

Does main have any parameters?

Finish the song..
horn/beep, lights/flash

Homework on Functions:
Give pupils a program containing several functions and ask them to identify (using different colours)

- Where the functions are called
- The function definitions
- Any parameters
- Any data passed as arguments
- Any return values that are passed back into the program

Using functions challenge
You are going to complete a program which calculates the area of a shape.
The program will ask which shape to calculate the area for and then ask the user to enter the size(s) of the shape.

The program will then calculate and display the area.
Part of this program has been written to find the area of a circle.
Open the Python program **area.py** and copy into your folder.
Look at the code.

Find the **main()** function. This function runs all the other functions. How does it work?

```
def main():
    print("Welcome - this program calculates the area of shapes")
    print("Please select the shape from the list")
    print("1 Circle")
    print("2 Square")
    print("3 Rectangle")
    answer = input("Please enter number 1,2 or 3 to select type of shape")
    if answer == "1":
        circle()
    if answer == "2":
        square()
    if answer == "3":
        rectangle()
```

. Enter **main()** in the shell prompt to start the program. Select the circle **(1) (this is the only function that has been written for you)**

Enter a number for the radius and check the program calculates the area correctly for the circle.

This is the function that calculates the area of the circle.

```
def circle():
    # function to calculate the area of a circle = pi * (radius) squared
    import math
    radius=float(input("Please enter radius of circle "))
    print(math.pi)
    area = math.pi * (radius*radius)
    print ("Area of a circle with radius of",radius,"is ",area)
```

Challenge: Write a function to ask the user to enter the length of one side of the square and then calculates the area of a square.

Repeat for the rectangle (width * height) (need 2 inputs) and the triangle (area = ½ base * height)

Useful resources

http://www.pythonschool.net/

http://www.edexcel.com/quals/gcse/gcse-2013/computer-science/Pages/default.aspx

Downloadable scheme of work to teach Python (Year 10)

Challenge: Write a program that asks the user to enter their four favourite numbers and stores them in a list. Display the list on the screen.

```
myNumbers=[]
for index in range(4):
     answer=int(input("please enter a lucky number"))
     myNumbers.append(answer)
print("Your lucky numbers are", myNumbers)
```

NETWORK OF EXCELLENCE COMPUTER SCIENCE TEACHING
COMPUTING AT SCHOOL
EDUCATE · ENGAGE · ENCOURAGE
In collaboration with BCS, The Chartered Institute for IT

Challenge: Write a program that asks the user to enter their four favourite numbers between 1 and 20 and stores them in a list. Display the list on the screen.

```
myNumbers=[]
for index in range(4):
    answer=input("please enter a lucky number")
    myNumbers.append(answer)
print("Your lucky numbers are", myNumbers)
```

Extension: Make this into a guessing game. Use the rand() function to find a random number between 1 and 20. if this number is the same as one of the numbers entered by the user display “you have won” else display “sorry – you have lost”)

NETWORK OF EXCELLENCE COMPUTER SCIENCE TEACHING

COMPUTING AT SCHOOL
EDUCATE · ENGAGE · ENCOURAGE
In collaboration with BCS, The Chartered Institute for IT

Challenge: Write a program that asks the user to enter their four favourite numbers between 1 and 20 and stores them in a list. Display the list on the screen.

```
myNumbers=[]
for index in range(4):
    answer=input("please enter a lucky number")
    myNumbers.append(answer)
print("Your lucky numbers are", myNumbers)
```

Extension: Make this into a guessing game. Use the rand() function to find a random number between 1 and 20. if this number is the same as one of the numbers entered by the user display “you have won” else display “sorry – you have lost”)

NETWORK OF EXCELLENCE COMPUTER SCIENCE TEACHING

COMPUTING AT SCHOOL
EDUCATE · ENGAGE · ENCOURAGE
In collaboration with BCS, The Chartered Institute for IT