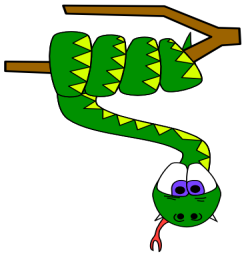


Python Lesson 3
Lists, for loops and while loops

Suffolk One, Ipswich, 4:30 to 6:00
Tuesday Jan 28
Nicky Hughes

NETWORK OF EXCELLENCE COMPUTER SCIENCE TEACHING

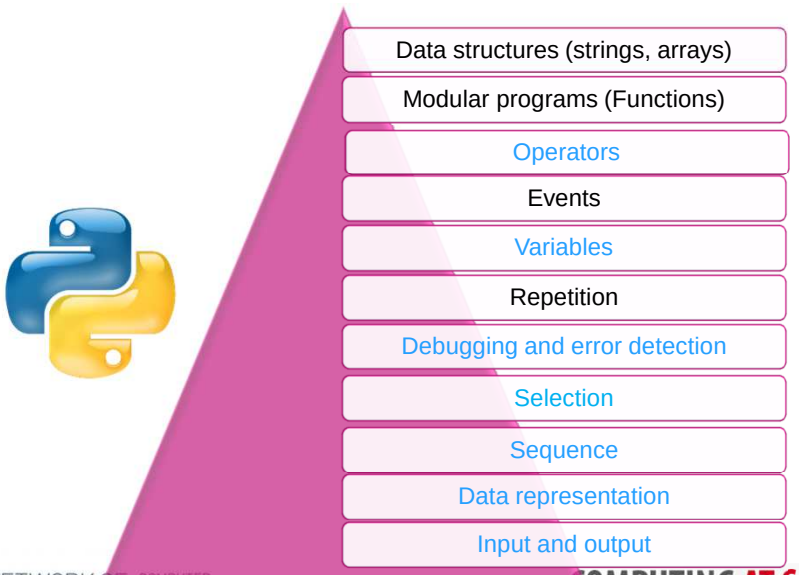
COMPUTING AT SCHOOL
EDUCATE · ENGAGE · ENCOURAGE
In collaboration with BCS, The Chartered Institute for IT

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

Comparison operators and if command
Learning outcomes

- Review last week lessons
- Be able to use **lists** at the interactive prompt
- Be able to use **lists** in scripts
- Be able to use **for** loops
- Be able to use the **range** function
- Introduction to writing your own **functions**

NETWORK OF EXCELLENCE COMPUTER SCIENCE TEACHING

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

Debug this program…

```
x = input("how are you today?"
if x == "sad" or == "tired"
print("I hope you feel better soon")
```


Debug this program…

```
x = input("how are you today?"
if x == "sad" or == "tired"
print("I hope you feel better soon")
```

```
x = input("how are you today?")
if x == "sad" or x == "tired":
    print("I hope you feel better soon")
```

String formatting

What will this display?

```
>>> a="vanilla"
>>> b="chocolate chip"
>>> c="strawberry"
>>> print("{0} and {2} are my favourite".format(a,b,c))
```

What is wrong with this command?

```
>>> print("{1} and {3} are my favourite".format(a,b,c))
```


String formatting

What will this display?

```
>>> a="vanilla"
>>> b="chocolate chip"
>>> c="strawberry"
>>> print("{0} and {2} are my favourite".format(a,b,c))
```

What is wrong with this command?

```
>>> print("{1} and {3} are my favourite".format(a,b,c))
```

```
>>> print("{0} and {2} are my favourite".format(a,b,c))
vanilla and strawberry are my favourite
>>> print("{1} and {3} are my favourite".format(a,b,c))
Traceback (most recent call last):
  File "<pyshell#37>", line 1, in <module>
    print("{1} and {3} are my favourite".format(a,b,c))
IndexError: tuple index out of range
```

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

What does this program do?
Draw the flow diagram for this program

```
number = int(input("Please enter a number"))
if number == 0:
     print("zero")
elif number < 0:
     print("negative")
else :
     print("positive")
```


Explain the difference between = and ==:

```
>>> f = 8
>>> k = 4
>>> f == k
```

What would be the result the command above and why?

What would be the result of the following command?

```
>>> f=k
>>> print(f)
```

Explain the difference between = and ==:

```
>>> f = 8
>>> k = 4
>>> f == k
```

What would be the result the command above and why?

What would be the result of the following command?

```
>>> f=k
>>> print(f)
```

```
>>> f=8
>>> k=4
>>> f==k
False
>>> f=k
>>> print(f)
4
>>> print(k)
4
```

NETWORK OF EXCELLENCE COMPUTER SCIENCE TEACHING

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

Mutability….
mutable or immutable?

If something is mutable it can changed (mutated).

If something is immutable it cannot be changed.

Lists are mutable

NETWORK OF EXCELLENCE COMPUTER SCIENCE TEACHING

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

A data structure is way of organising data items.

Data structures in Python

- Lists [ ] unordered list that is mutable
- Tuples () unordered list that is immutable
- Sets {} unordered list of unique items
- Dictionaries {} unordered list of key:value pairs where the key is unique
- Strings " " unordered list of characters

LISTS in Python

Python allows you to store items in a collection of data called a **list** (also called an **array**).

- Items in a list are indexed.
- The first item is at index 0, the second at index 1 and so on.

red	green	blue	white	orange
0	1	2	3	4

- All lists are given a name
- Lists can store all data types: strings, integers, floats, Boolean and even other lists

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

LISTS in Python

red	green	blue	white	orange
0	1	2	3	4

- Items in a list 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 “green” and “blue”

Using Lists

Use the interactive prompt to experiment with lists>>>

Create a list	`>>> mylist=["red","green","blue","white"]`
Display a list	`>>> print(mylist)` `['red', 'green', 'blue', 'white']`
Reference an item in a list	`>>> print(mylist[2:4])` `['blue', 'white']`

Delete an item in a list	`>>> del mylist[2]`
	`>>> print(mylist)`
	`['red', 'green', 'white']`

Delete a list	`>>> del mylist`

Append to the end of a list (round brackets)	`>>> mylist.append("purple")`
	`>>> print(mylist)`
	`['red', 'green', 'white', 'purple']`

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

Other useful list methods

list.append(x)	Appends the item x at the end of the list
list.sort()	Sorts the list (need to be a sortable data type)
list.reverse()	Reverses the elements in the list
list.count(x)	Counts the number of times x appears in a list
list.index(x)	Returns the index at which x appears
list.insert(index,x)	Inserts an item x at position before index

Experiment with the list methods

Python **For loops**

- The for statement is a looping command which lets you repeat commands.
- The for statement **iterates** over the items in a list, a string or range.

```
for variable in list :
    command
    command
```

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

Python **For loops**

- for loop end in a colon
- uses indentation to specify which lines are affected by the statement

```
colours = ["red","green","purple","orange"]
for item in colours:
    print(item)
```

What happens when this program runs?
What does the for loop do?
How is the variable "item" used?

NETWORK OF EXCELLENCE COMPUTER SCIENCE TEACHING

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

Try the for loop challenges

Using FOR loops

Challenge 1 Write a program that creates a list called animals containing your 4 favourite animals and then use a for loop to display them one at time on the screen.

Challenge 2 You are stranded on a desert island. Write a program that creates a list of things you would take to the desert island. The program should then display each item a line at a time displaying "On my desert island I would like:" at the beginning of each line.

NETWORK OF EXCELLENCE COMPUTER SCIENCE TEACHING

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

For loops and variables

Challenge 3 Write a program that creates the following list and then adds up all the numbers in the list and displays this answer.

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

Challenge 4 Write a program that create the list containing the marks for 6 students in an exam. Write a program that finds the average mark.

myMarks=[55,76,98,34,21,23]

.

The range function

Range function gives a range of iterable integer numbers, Numbering starts at 0 and ending at the item before “stop” (up to and not including)

(start,stop,step)

range(40)

range(5,15)

range(10,100,5)

```
for item in range(40):
    print(item)
```

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

For loops interate over strings

Try this program

```
for item in "Preston North End":
    print(item)
```

NETWORK OF EXCELLENCE COMPUTER SCIENCE TEACHING

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

For loops and strings

Write a program that asks you to enter a name and then displays each letter of the name on a separate line.

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?

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}
```

AND, OR and NOT (boolean operators) allow you to combine the results to two condition statements

condition_one **and** condition_two

This condition is true only if condition_one and condition_two are both true.

condition_one **or** condition_two

This condition is true if condition_one or condition_twois true.

not condition

This creates a new condition, which is the reverse of the original, setting true to false and false to true.

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

Hint: Try asking yourself the questions:
*Is condition_1 true **OR** condition_2 true? if YES then the answer is true.*

Condition	True or False
(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)	

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

Condition	Result
(78 == 10) **or** (6 == 7)	False – as neither of the conditions are true
(78 == 10) **or** (6 == 6)	True – one of the conditions is true (6 == 6)
(78 == 10) **and** (6 == 6)	False – only one of the conditions is true and both need to be true for an AND
(1 < 10) **and** (2 < 10)	True – as both conditions are true
(1 < 10) **or** (2 < 10)	True – as at least one of the conditions is true
not (5 == 5)	False – as the answer to the condition is true as 5 is equal to 5
not (6 < 4)	True – as the answer to the condition is false 6 is less than 4

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)

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

```
myNumbers=[19,6,7,9,2,25,16]
max = 0
for next in myNumbers:
    if next > max :
        max = next

print("the largest number is", max)
```