

Lunch Time?



Programming Construct Three: Repetition



Repetition Statements

- Python supports two types of loop (repetition) statement:
 1. For Loops, and
 2. While Loops.

While Loops



The While Loop Statement

```
While <TEST> :
    <STATEMENTS>
```

- General purpose loop for repeating some sequence of statements.

Problem Example 8: ASCII Character Set



ASCII Character Set Requirements

Produce a Python program that outputs the ASCII (**American Standard Code for Information Interchange**) character set from 0 to 127 (inclusive).

Note that some ASCII character codes are unprintable. For example:

7	=	Beep
9	=	Horizontal tab
10	=	Line feed
27	=	Escape
32	=	Space
127	=	Delete

ASCII Character Set Source

- Load `PythonExampleProblems\Repetition\ASCIIcharacterSet\asciiCharacterSet.py` into the IDLE editor.

```
counter = 0;

while counter < 128 :
    print('{0} = {1}'.format(counter, chr(counter)))
    counter += 1
```

Run The ASCII Character Set Source Code

- Try printing out the first 96 (codes 0 to 95 inclusive) characters.

Problem Example 9: Menu Input



Menu Input Requirements

Design and implement some Python code that allows the user to select from five different menu options on a continuous loop, including an option to quit the program. Include an error handling mechanism for the situation where an unrecognised menu option is input by the user.

Menu Input Source

- Load `PythonExampleProblems\Repetition\MenuInput\menuInput.py` into the IDLE editor.
- Note that it features an infinite loop (the test statement comprises the Boolean value `True` which evaluates to itself) and that the termination statement is embedded in the loop using an “if” statement.

```
while True :
    <STATEMENTS>
    if <TEST> :
        quit()
```

or

```
while True :
    <STATEMENTS>
    if <TEST> :
        break
```

Run The Menu Input Source Code

For Loops



The For Loop Statement

```
for <TARGET> in <OBJECT>:  
    <STATEMENTS>
```

- Typically used for processing lists and dictionaries.

Problem Example 10: Landscape Gardening Material Costs Update



Landscape Gardening Material Cost Update Requirements

- AQA GCSE Specimen Controlled Assessment example, Task 3:

“Develop a way to allow the user of the system to store the costs of raw materials in an external file so that these costs can be changed when the price of the materials rise and fall.”

Landscape Gardening Material Cost Update Source Code

- Load PythonExampleProblems \FileHandling\MaterialCostUpdate \materialCostDictionary.py into the IDLE text editor.

Landscape Gardening Material Cost Update Comments (1)

- Note how the for loop to assign values to the material costs dictionary is defined:

```
index = 0
for key in materialCost.keys():
    materialCost[key] = \
        float(newMaterialCosts[index])
    index=index+1
```

Landscape Gardening Material Cost Update Comments (2)

- Note how the for loop to output the material costs dictionary is defined:

```
for key, value in materialCost.items():
    print('{0} = {1}'.format(key,value))
```

- To obtain just the dictionary values:

```
for value in <DICTIONARY>.values():
    print(value)
```

Run Landscape Gardening Material Cost Update Source Code

- There is a material costs file that can be used:

```
newMaterialCosts.txt
```

- Try editing this file and then loading it again.

More Loop Examples



Problem Example 11: Monthly Quote Summarisation Report



Monthly Quote Summarisation Report Requirements

- AQA GCSE Specimen Controlled Assessment example, Task 4:

“The Managing Director of the company has asked if there is a way that she can use this system to prepare a monthly report that compares the total cost of all materials purchased per month across all jobs undertaken.

The user of the system would type in the name of the month or the month number (e.g. 6 for June). The program will then read all of the materials purchased for that month and calculate the totals ...”

Monthly Quote Summarisation Report Source Code

- Load PythonExampleProblems \FileHandling\SummarisationReport \summarisationReport.py into the IDLE text editor.

Monthly Quote Summarisation Report Comments (1)

- Assumes the existence of a quotesToDateFile.txt of the form:

```
April quoteNumber1.txt
April quoteNumber2.txt
May quoteNumber3.txt
May quoteNumber4.txt
```

- And a set of associated quote files of the form:

```
May
3 7 325.5 420.0
0 0 0.0 0.0
1 150.0 60.0
```

Monthly Quote Summarisation Report Comments (2)

- Note the way that the quote file input loop operates:

```
index = 0
while True :
    if (quotes[index]==month) :
        inputQuoteFromFile2(quotes[index+1])
        <STATEMENTS>
        index = index+2
    if (index>=len(quotes)) :
        break
```

Run The Quote Summarisation Report Source Code

- By default six quote files are used, two for “April”, two for “May” and two for “June”:

```
quoteNumber1.txt
quoteNumber2.txt
quoteNumber3.txt
quoteNumber4.txt
```

Problem Example 12: Landscape Gardening III, The Final Cut!



Landscape Gardening III Source Code

- Load PythonExampleProblems
 \FileHandling
 \LandscapeGardeningIII
 \landsGardQuoteIII.py into the IDLE
 text editor.

Run The Landscape Gardening III Source Code

Home Time?

