Introduction to Programming

Python Lab 3:
Arithmetic

PythonLab3 lecture slides.ppt

Ping Brennan (p.brennan@bbk.ac.uk)

15 October 2019 or
31 January 2020

Getting Started

- Create a new folder in your disk space with the name **PythonLab3**

- Launch the Python Integrated Development Environment (IDLE) - begin with the **Start** icon in the lower left corner of the screen.

- If you are in a DCSIS laboratory, search using the keyword **Python** and click on **IDLE (Python 3.6 64-bit)**

  A window with the title **Python 3.6.2** should appear. This window is the *Shell*.

Getting Started (2)

- If you are in the ITS laboratory MAL 109, then right mouse click on the **Start** icon in the lower left corner of the screen.

  A list of menu options should appear and click on ***Search***. Type ***Python*** in the search text box at the bottom of the pop-up window. A list of Apps should appear and select

  Python 3.4 IDLE(PythonGUI)

  A window with the title **Python 3.4.3 Shell** should appear. This window is the *Shell*.

- In the *Shell* click on **File**. A drop down menu will appear.

  Click on **New File.** A window with the `title` **Untitled** should appear. This window is the *Editor*.

Getting Started (3)

- In the *Editor*, click on **File**, and then in the drop down menu click on **Save As...** .

  A window showing a list of folders should appear.
  - To search any folder on the list, double click on the folder.
  - Find the folder PythonLab3 and double click on it.
  - In the box **File name** at the bottom of the window
    1. Type **`PseudoCode.py`**
    2. Then click on the button **Save** in the lower right corner of the window.

    The title of the *Editor* should change to show the location of the file `PseudoCode.py`.

Program `PseudoCode.py` calculates cost of a car

- **Learning objectives**
  - Use Python arithmetic operators: + - * /
  - Understand arithmetic expressions
  - Use variables to store values of type **int** or type **float**

- **Question 2: Problem statement**

  Calculate the cost of owning a car for 10 years. Assume a purchase price of £10,000, a price of £4 per gallon of petrol, a usage of 15,000 miles per year and a fuel efficiency of 20 miles per gallon.

Program `PseudoCode.py` calculates cost of a car (2)

- **Problem solving**

1. Create and initialise **five** variables in a Python program to store the number of years, purchase price, price per gallon, annual miles driven, and fuel efficiency. Choose descriptive variable names, for example,

```
annualMilesDriven = 15000   # stores the annual miles driven

fuelEfficiency = 20         # stores the fuel efficiency
```

2. Convert the pseudo code below into a sequence of Python assignment statements in order to calculate the total cost of a car.

   Create **four** variables (e.g. `annualFuelConsumed`, `annualFuelCost`, etc.) to hold the resulting values of the arithmetic expressions in the Python statements.

```
annual fuel consumed = annual miles driven / fuel efficiency

annual fuel cost = price per gallon * annual fuel consumed

operating cost = number of years * annual fuel cost

total cost = purchase price + operating cost
```

Program PseudoCode.py calculates cost of a car (3)

- **Problem solving** (continued)

  3. Print out the string "Total cost: " followed by the numerical value of the total cost.

- Provide a comment at the beginning of the program to explain the purpose of the program, together with your name and the date.

- Save the program to the file **PseudoCode.py**

- Run your program.

Program `IntegerCalculations.py` shows integer calculations

- Create a new *Editor* for a new file called **`IntegerCalculations.py`**
- **Question 3: Problem statement**

  Write a program that assigns integer values to **two** variables **`x`** and **`y`**, and then prints out the following values.

  i. The sum

  ii. The difference

  iii. The product (that is, the multiplication of the values in the two variables).

  iv. The average

  v. The absolute value of the difference

  vi. The maximum

  vii. The minimum

Program `IntegerCalculations.py` shows integer calculations (2)

- **Problem solving** – convert the steps below into a sequence of Python statements.

1. Create **two** variables `x` and `y`, and assign integer values to both. For example,

```
x = 3
```

2. Calculate and print the **sum**, the **difference**, the **product** and the **average** of the values in the two variables.

   Break down the task for step 2 into **four** separate print statements.

   For example,

```
print("The sum of the two integers is", x+y)
```

Program `IntegerCalculations.py` shows integer calculations (3)

- **Problem solving** (continued)

  3. Use the built-in functions in the table below in your program to calculate and print the **absolute** value of the difference, the **maximum** and the **minimum** of the values in the two variables. For example,

```
print("The absolute value of the difference is", abs(x-y))
```

- The Python built-in functions that you can use in your program is listed in the table below.

Function	**Explanation**	**Example**
abs(x)	Returns the **absolute** value of x.	`abs(-5)` #returns the value `5`
max(x, y, z, …)	Returns the **largest** value of the arguments.	`max(1, 2)` #returns the value `2`
min(x, y, z, …)	Returns the **smallest** value of the arguments.	`min(1, 2)` #returns the value `1`

Program `IntegerCalculations.py` shows integer calculations (4)

- The printed output should be in the following style (assuming variable `x` holds an integer value `3` and variable `y` holds an integer value `5`)

```
The sum of the two integers is 8.
```

- Provide a comment at the beginning of the program to explain the purpose of the program, together with your name and the date.
- Save your program to the file `IntegerCalculations.py`
- Run your program.

Program `Rectangle.py` prints properties of Rectangles

- Create a new *Editor* for a new file called `Rectangle.py`
- **Question 4: Problem statement**

  Write a program that assigns integer values to **two** variables `a` and `b` and then prints out the **area** of the rectangle with sides `a`, `b`, the **perimeter** of the rectangle and the **length of a diagonal** of the rectangle.

Program `Rectangle.py` prints properties of Rectangles (2)

- **Problem solving** – convert the steps below into a sequence of Python statements.

  1. Create **two** variables `a` and `b`, and assign integer values to both. For example,

     ```
     a = 4
     ```

  2. Calculate and print the **area** which is the product of the lengths of the sides. For example,

     ```
     print("The area of the rectangle is", a*b)
     ```

  3. Calculate and print the **perimeter** of the rectangle which is the sum of the lengths of the sides.

Program `Rectangle.py` prints properties of Rectangles (3)

- **Problem solving** (continued)

4. Calculate and print the **length of a diagonal of the rectangle**.

   First find the square of the diagonal which is equal to the sum of the squares of the sides, and then work out the square root of the resulting value.

   To use the square root function `sqrt(x)`, place the following `import` statement at the **top of your program code**.

```
from math import sqrt
```

   The function `sqrt` is called with the argument `x` which can be an arithmetic expression, such as `a*a + b*b`. For example,

```
print(sqrt(a*a + b*b))
```

Program `Rectangle.py` prints properties of Rectangles (4)

- Include in the print out a short description of each number that is printed. An example is shown in step 2 of the 'Problem solving' section on page 13.

- Provide a comment at the beginning of the program to explain the purpose of the program, together with your name and the date.

- Save your program to the file `Rectangle.py`

- Run your program.

Program `SeparateDigits.py` prints separate digits

- Create a new *Editor* for a new file called `SeparateDigits.py`
- **Question 5: Problem statement**

  Write a program that assigns a **five digit positive integer** to a variable `x` and prints out the individual digits, separated by spaces. For example, if `x` has the value 16348 then the print out is

```
1 6 3 4 8
```

- **Learning objective**
  - Understand the use of the remainder operator `%` and the floor division operator `//`

  Note that if `x` and `n` are integers then:

Expression and explanation	Example
`x//n` is the quotient on dividing `x` by `n`	`6//4 #returns the value 1`
`x%n` is the remainder on dividing `x` by `n`	`5%4 # returns the value 1`

Program `SeparateDigits.py` prints separate digits (2)

- **Problem solving** – write a sequence of Python statements as follows.

```
# assign a five digit integer to a variable x
x = 16348
# find the unit digit from the value stored in x
d1 = x % 10                # assigns the integer 8 to d1
# find the tenth digit
d2 = (x // 10) % 10        # assigns the integer 4 to d2
# find the hundredth digit
d3 = (x // 100) % 10       # assigns the integer 3 to d3
# add a statement below to find the thousandth digit
# by using a similar arithmetic expression as above.
… Code to be added here by you …
# add a statement below to find the ten thousandth digit
# by using a similar arithmetic expression as above.
… Code to be added here by you …
# add a print statement below to output the individual
# digits separated by spaces.
… Code to be added here by you…
```

Program `SeparateDigits.py` prints separate digits (3)

- Provide a comment at the beginning of the program to explain the purpose of the program, together with your name and the date.

- Save your program to the file `SeparateDigits.py`

- Run your program.

Supplementary Questions for Private Study

- The laboratory worksheet contains supplementary questions 6.1, 6.2 and 6.3 for private study.

- You are encouraged to complete the supplementary questions at home, or in the laboratory if you have time after completing questions 2 to 5.