



Introduction to programming

Input, output and variables - Lab

Exercise 1 – favourite_film.py

You are starting to build a database of your friend's favourite films so you decide to write a program that asks each person in turn. Write a program that displays a message to the user that asks them to enter their favourite film. Store this in a variable and repeat back to them what they have entered. Save this in your Week 2 folder as **favourite_film.py** and test that your program works as expected.

Exercise 2 – login.py

Your manager has asked you to start writing the login section of a much larger program. You will need to write a program that asks users for their password and stores it in a variable. As a test whilst the rest of the program is being developed, include in your program a boolean expression that results in **True** if the user enters the password as **password123**. Print the result of the expression to the user. Save this in your Week 2 folder as **login.py** and test that your program works as expected.

Exercise 3 – interactive_calculator.py

You have seen how Python can be used as a calculator using IDLE by simply typing mathematical expressions. Create a program that provides a nicer interface to users by asking them for two numbers, multiply them together and display the result. You will need to use the **int()** function when trying to multiply the users responses taken using **raw_input()** (as these will be strings). Save the program in your Week 2 folder as **interactive_calculator.py** and test that your program works correctly.

Extension task – operating_system.py

Your manager has asked you to create a program that displays information about the computers in the office. You have decided the best way to do this is to write a Python program and run it on each of the computers. You will need to do this by accessing the environment variables on your computer.

To access an environment variable first import the **os library** with this statement at the top of your program:

```
import os
```



You can then print environment variables with:

```
print os.environ["VARIABLE_NAME_HERE"]
```

Replacing **VARIABLE_NAME_HERE** with a reference to one of the environment variables. Some examples of useful environment variables are:

- COMPUTERNAME
- OS
- PROCESSOR_ARCHITECTURE
- NUMBER_OF_PROCESSORS

You can find a full list of environment variable names by running the following statement in IDLE:

```
print os.environ.keys()
```

You can format the output in any way you wish. Save your completed file in your Week 2 folder as **operating_system.py**.