www.hyperiondev.com

Python Web Developer

Task: Input and Output

Task: Input and Output

Introduction

Overview

Until now, the **Python** code you've been writing comes from one source of input and only goes to one place of output. For example, you type it in at the keyboard and the results are displayed in the console.

But what if you want to read information from a file on your computer, and/or write that information to another file?

This process is called **File I/O** (the "I/O" stands for "input/output"), and **Python** has a number of built-in functions that handle this for you. In this task, we will look at different ways of achieving this in **Python**.

How does this relate to prior tasks?

As we have learnt in the prior tasks, we can store data in variables and those variables can be combined to form data structures.

One important fact we must remember is that variables and data structures exist only while the program is running or exists in another process or while your computer still has an energy supply, in other words data contained in computer memory is transient (exists while it is used/needed).

More often that not, in programming, you need to access data from multiple sources such as user interaction (typing text on a keyboard) or through **File I/O** (a program reading a picture of a kitten).

Instead of typing input thousands of times, you can simply store the information in a file and read it whenever it is needed, similarly when you need to store information.

Instructions

First read **example.py**, open it using **Notepad++** (Right click the file and select 'Edit with Notepad++').

- **Example.py** should help you understand some simple **Python**. Every task will have example code to help you get started. Make sure you read all of example.py and try your best to understand.
- You may run **example.py** to see the output. The instructions on how to do this are inside the file. Feel free to write and run your own example code before doing the required task to become more comfortable with **Python**.
- You are not required to read the entirety of Additional Reading.pdf, it is purely for extra reference.

Compulsory exercise to complete the required task

Follow these steps:

Now, create a **Python** file called **forgetful.py**.

Imagine your friend was very forgetful and always entered his email password incorrectly.

You want to write a **Python** program that takes all his incorrect password entries, stores them in a list and then records all his incorrect password entries in a text **file** called **wrongpasswords.txt.**

Example: your friend's password is *rusty*. But he enters *rusty123*, *Rusty*, *rustless* before finally remembering that his password is *rusty* and enters it correctly.

In this situation **wrongpasswords.txt** should read exactly like this:

```
Incorrect password 1: rusty123
Incorrect password 2: Rusty
Incorrect password 3: rustless
Correct password entered on 4th entry.
```

The program should ask the user for input by saying 'Please enter your password'. You can use code from the program you wrote in the prior task. The correct password will always be *rusty* but the user can of course enter any String.

Enrichment task:

Edit your completed program so that the number of characters your friend gets wrong is also stored for each incorrect password.

In the same situation given above, **wrongpasswords.txt** should read this exactly:

```
Incorrect password 1: rusty123 , wrong by 3 characters.
Incorrect password 2: Rusty , wrong by 1 characters.
Incorrect password 3: rustless , wrong by 4 characters.
Correct password entered on 4th entry.
```

You should define a separate function in your code, called **`countDifference`**, that takes in a String.

Need assistance?

Don't hesitate to contact your trainer, we will be available to assist you and help you progress through this course.

Task Statistics

Last update to task: 13/04/2015.
Author: Richard Niescior
Main trainer: Umar Randeree
Task Feedback link: **Hyperion Development Feedback**