



TASK

Working with External Data Sources — Input

Visit our website

Introduction

WELCOME TO THE 'INPUT' TASK!

Your programs will often have to interact with the user, either by soliciting an input or by displaying an output. Until now, the Python code you've been writing has received input in only one manner — through the `input()` function that makes the user type in some input using their keyboard — and has only displayed output in one way — through the `print()` function that displays the output on 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 can handle this for you.



Get in touch
Connect for support

Remember that with our courses, you're not alone! You can contact your mentor to get support on any aspect of your course.

The best way to get help is to login to www.hyperiondev.com/portal to start a chat with your mentor. You can also schedule a call or get support via email.

Your mentor is happy to offer you support that is tailored to your individual career or education needs. Do not hesitate to ask a question or for additional support!



WORKING WITH FILES IN PYTHON

Before moving further, it is necessary to understand how to handle the most simple type of files — text files — using Python.

Python has several functions for creating, reading, updating, and deleting files. It includes a built-in file type which is a bit like a String data structure, but much more complex.

Before you can read a file or write to a file, you will have to open it using Python's built-in `open()` function.

The code below creates a file object named `f` that is linked to the `example.txt` file in this folder. You'll learn about objects in a later task.

```
f = open('example.txt', 'r+')
```

The code in the example above means that `f` is open for reading. The first argument (**example.txt**) is the filename and the second argument is the mode, which can be `'r'`, `'w'`, or `'r+'`, among some others. The mode `'r'` is for reading only, `'w'` is for writing only and `'r+'` allows you to both read and write to the file without having to close the file in between.

Notice the way that this is written is similar to calling a function `open` with two input parameters. As described above, behind the scenes, this is exactly what is happening. We are using a built-in function called `'open'` and passing it the arguments `'example.txt'` and `'r+'`.

Here we intend to read and write from/to a text file named `'example.txt'`, which is already in the same folder as this file. Python will look in this directory for `'example.txt'`, and try to read its content.

The most common way to read from a file is simply to loop over the lines of the file.

We can directly loop over the variable `f`, which is stored in Python as a list of lines — each line is 1 line of the file.

```
for line in f:
    print("The first character of this line is: " + line[0] + "\n")
    print("The entire line is: " + line)
```

Always close files when done with them, by using `f.close()` in order to free up the resources it was using. Notice this is a function that takes in zero input.

We could build up all the lines of the text file into one large String called `contents` as follows:

```
contents = ""
f = open('example.txt', 'r+') # Open the file again!

for line in f:
    contents = contents + line

f.close() # Always close files when done with them.
```

We now have the contents of an external resource (a text file) stored inside our program in a variable called 'contents'. That's pretty powerful! But for now, let's just print the contents to a screen:

```
print (contents)
```



A note from our coding mentor **Ridhaa**

Sorry to interrupt, but I've been reading about how Apple introduced the Macintosh to the world and would like to share that with you.

Apple introduced the Macintosh with the biggest and most expensive advertising platform available: the 1984 Super Bowl. The advert played on the theme of totalitarianism in George Orwell's book 1984. Apple made a sneaky reference to overcoming IBM by conveying the power of personal computing found in a Macintosh by a depiction of the destruction of "Big Brother".

The Macintosh was the first successful mouse-driven computer with a graphical user interface and was based on the Motorola 68000 microprocessor. Its price was \$2,500. The Macintosh came with various applications as part of the package. These included MacPaint, which made use of the mouse and MacWrite, which demonstrated WYSIWYG (What You See Is What You Get) word processing.



Image source:

<https://www.forbes.com/sites/connieguglielmo/2014/01/24/macintosh-at-30-interesting-profound-and-curious-things-said-about-apples-insanely-great-computer/#67b3d4c565e8>

Instructions

First, read **example.py**, open it using IDLE.

- **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. Feel free to write and run your own example code before doing the Task to become more comfortable with Python.

Compulsory Task 1

Write a program that reads the data from the text file called **DOB.txt** and prints it out in two different sections in the format displayed below:

Name

1. Orville Wright
2. Rogelio Holloway
3. Marjorie Figueroa
4. ... (and so on)

Birth date

1. 21 July 1988
2. 13 September 1988
3. 9 October 1988
4. ... (and so on)

Compulsory Task 2

Follow these steps:

Create a new Python file in this folder called **Optional_task.py**.

- Create a new text file in this folder called **input.txt**. In the **input.txt** file enter some text, making sure it is at least a few lines long.
- Write a program that will count the number of characters, words and lines in the file. Your program should also count the number of vowels (A, E, I, O, U being the five vowels) in the file.
- Print out your results.

Completed the task(s)?

Ask your mentor to review your work!

[Review work](#)



Rate us

Share your thoughts

HyperionDev strives to provide internationally-excellent course content that helps you achieve your learning outcomes.

Think that the content of this task, or this course as a whole, can be improved or think we've done a good job?

[Click here](#) to share your thoughts anonymously.

