

Modules, Files

Modules

Python contains a standard library that can be used to create powerful programs.

A library is a collection of code that is ready for you to use in your program.

A standard library is considered part of the language and must be included with any Python system.

Python's standard library is organized into modules; each module contains related functions.

In addition to using the standard libraries, you can also create your own modules, however it is rarely necessary to write your own implementations for mathematical or string functions.

[Python Library](#)

Using Modules - import

Before using the functions contained in a module, you must first import the module to your program.

Importing is done using an import statement.

Using the import statement, we can import specific functions within a module or all functions contained in the module at once.

Syntax:

```
from math import sqrt -> makes available only sqrt in math
```

```
from math import sqrt, sin, cos -> makes available specific functions listed
```

```
from math import * -> makes available all function in math module
```

```
import math -> makes available all functions in math module, when
```

calling functions, you must add the module name

```
ex: math.sqrt(25)
```

Creating our own Modules

Modules allow for problem decomposition.

Instead of placing our entire program in a single file, we divide the program into modules (different source code files).

Python modules allow us to easily construct a program from code in multiple files.

A module is a .py file, containing Python definitions and statements.

Example:

- `05_circle.py / 05_circleTest.py / 05_circleTest2.py`

Exercise

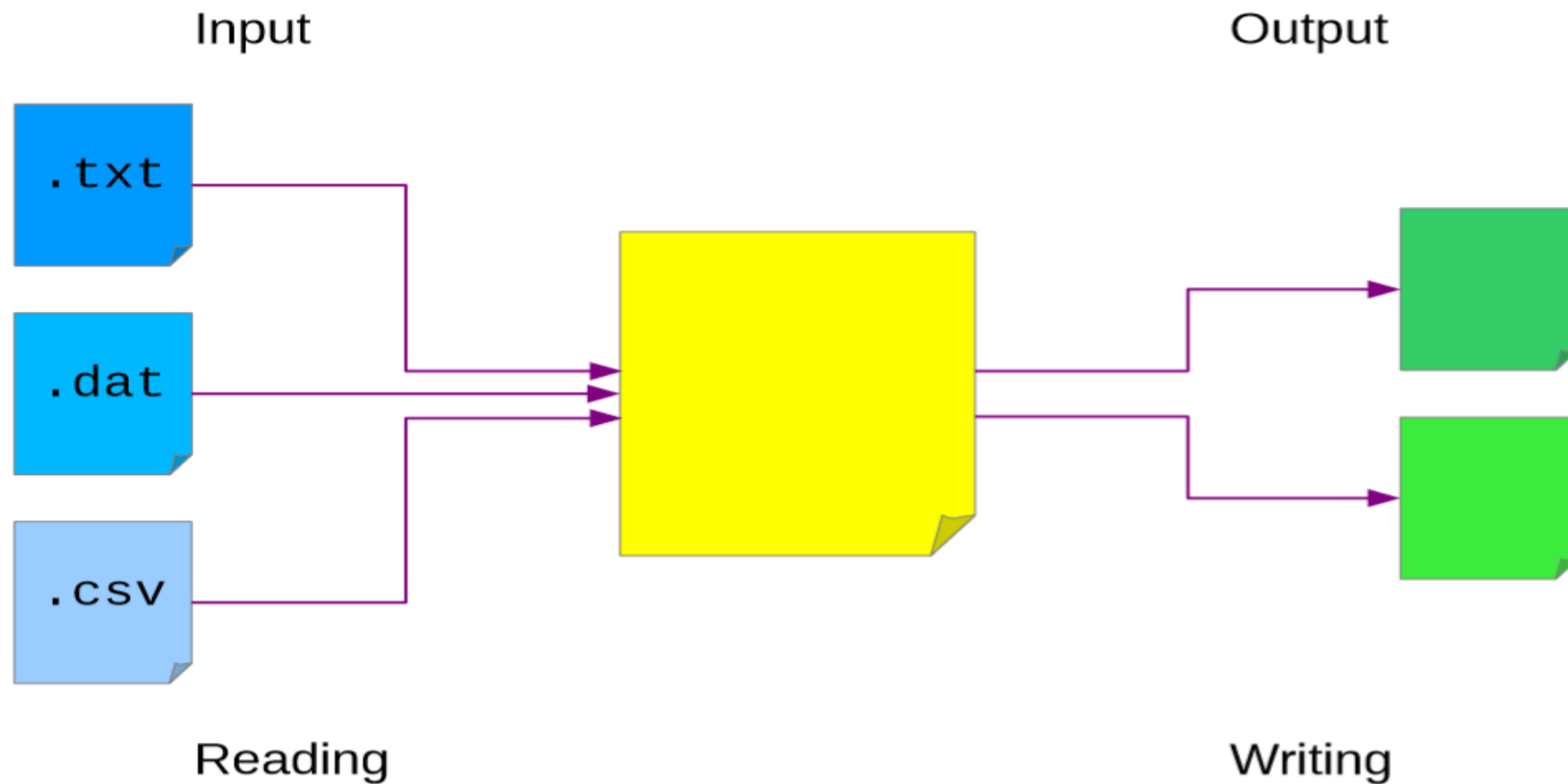
Create a module (`fitness.py`) that stores fitness calculations. Your module should define the following data and functions:

- `MAX_RATE_CONSTANT` (220)
- `MAX_BMI` (24)
- `MIN_BMI` (21)
- `calculate_bmi` (`weight / height2`)
- `display_bmi_category` (30+ obese, 25-30 overweight, 20-25 normal, <25 underweight)
- `calculate_target_heart_rate` (80% of (220 minus age))
- `display_weight_range` (bmi between min and max bmi)

Create a script that inputs the age, height and weight from the user and displays their fitness detail.

- `05_fitness_module.py`

Files



File Functions

Function	Purpose
<code>open(fileName, 'w')</code>	Creates new a file for writing and returns the file object (file handle).
<code>open(fileName, 'r')</code>	Opens an existing file for reading and returns a file handle.
<code>open(fileName, 'a')</code>	Opens an existing file for appending and returns a file handle.
<code>fh.read()</code>	Returns a string containing the contents of the file associated with the file handle.
<code>fh.readline()</code>	Returns the next line in the file associated with the file handle.
<code>fh.readlines()</code>	Returns list, each element of which is one line of the file associated with the file handle.
<code>fh.write(s)</code>	Writes the string <code>s</code> to the end of the file associated with the file handle.
<code>fh.writelines(S)</code>	<code>S</code> is a sequence of strings, writes each element of <code>S</code> as a separate line to the file associated with the file handle.
<code>fh.close()</code>	Closes the file associated with the file handle.

Writing Data to a File

```
#writing data to a file
fileHandle = open( '05_file1.txt', 'w')
for i in range(2):
    name = input('Enter name: ')
    fileHandle.write( name+'\n')
fileHandle.close()
```

File Contents:

Melisa Aksoy

Sam Smith

Appending Data to a File

```
#append data to end of file  
fileHandle = open( '05_file1.txt', 'a')  
fileHandle.write('Jane Doe\n')  
fileHandle.close()
```

File Contents:

Melisa Aksoy

Sam Smith

Jane Doe

Reading Data from File – read()

```
#reading data from a file - read()
fileHandle = open( '05_file1.txt', 'r')
fileContents = fileHandle.read()
print(fileContents)
fileHandle.close()
```

Reading Lines from a File – readline()

```
#reading specific lines from a file - readline()
fileHandle = open( '05_file1.txt', 'r')
for i in range(2) :
    line = fileHandle.readline()
    print(line[:-1])
fileHandle.close()
```

Note:

using the `[:-1]` syntax in the print method call stops the print method from outputting the newline character from the file data.

We can also use the string `strip()` function, which removes the leading and trailing whitespace characters and returns the new string.

Reading Data from a File – for loop

```
#read all lines from the file
fileHandle = open( '05_file1.txt', 'r')
for line in fileHandle:
    print(line[:-1])
fileHandle.close()
```

Useful String Functions

When processing files, there are several built-in string functions that might be useful.

Function Name	Purpose
<code>s.count(s1)</code>	Counts how many times the string <code>s1</code> occurs in <code>s</code> .
<code>s.find(s1)</code>	Returns the index of the first occurrence of the substring <code>s1</code> in <code>s</code> , and -1 if <code>s1</code> is not in <code>s</code> .
<code>s.find(s1, pos)</code>	Same as <code>find</code> , but starts searching from the given <code>pos</code> .
<code>s.rfind(s1)</code>	Same as <code>find</code> , but starts from the end of <code>s</code> (the 'r' in <code>rfind</code> stands for reverse)
<code>s.replace(old, new)</code>	Replaces all occurrences of the string <code>old</code> in <code>s</code> with the string <code>new</code> .
<code>s.strip()</code>	Removes the leading and trailing whitespace from <code>s</code> .
<code>s.rstrip()</code>	Removes the trailing whitespace from <code>s</code> .

Sample String Question

Write a program that inputs two strings str1 and str2 from the user. The program will call a function, **form**, that will concatenate the part before the first occurrence of str2 in str1 and the part after the last occurrence of str2 in str1 and return the new string. The program should continue to input pairs of strings until the user types 'exit' in any case for the first string.

See: 05_string_exercise.py

Sample Runs:

Enter first string (exit to quit): this is his coat

Enter second string: is

Formed new string = "th coat"

Enter first string (exit to quit): I am in the house

Enter second string: the

Formed new string = "I am in house"

Enter first string (exit to quit): I am there

Enter second string: are

Formed new string = ""

Enter first string: EXIT

Exercises:

1. Write a program that reads hotel data from a file, and writes all hotels in the city input by the user to a new file with the name `<city>hotels.txt`. Note: you may assume that the city name is at the beginning of each line, and that the city is followed by a dash(-) then the hotel name.
 - `05_hotel.py / 05_hotels.txt`
2. Write a program to input two sorted files containing the names of flowers and merges the two files into a new sorted file.
 - `05_flowerExercise.py / 05_flower1.txt / 05_flower2.txt`
3. Write a program that reads two country data files, `05_worldpop.txt` and `05_worldarea.txt`. Both files contain the same countries in the same order. Write a file `world_pop_density.txt` that contains country names and population densities (people per square km).
 - `05_country.py / 05_worldpop.txt / 05_worldarea.txt`

Terms of Use

- This presentation was adapted from lecture materials provided in [MIT Introduction to Computer Science and Programming in Python](#).
- Licenced under terms of [Creative Commons License](#).



Attribution-NonCommercial-ShareAlike 4.0 International