

Exercise: People

Table of Contents

Background	1
Notes	1
Instructions	1
Hints	3

Background

In this exercise you will be writing a script that will take a text file and parse the text in each row in the text file so that we can extract the important information from each row. This text file represents an example of a messy data file that you might find yourself having to analyze at some point in time. In this case, each line in the text file contains data for one person. It contains the first and last name, address and email in that order.

You will be writing three functions (`parse_name`, `parse_address`, `parse_email`) that use regular expressions to extract data from the string that we pass into the functions. You will also write two classes that will act as data containers. These classes are `address` and `employee`.

The `address` class will have 3 attributes, `street`, `city` and `state`. The `employee` class will have four attributes, `first_name`, `last_name`, `address`, and `email`. Your script should also contain a `main` function that will have one parameter, a path to the file that we are extracting data from. Your `main` should read the file and will create `employee` instances out of each row in the file.

Notes

- The name of your file should be `people.py`.
- This exercise does not require the use of command line arguments.

Instructions

- Download the `people.txt` file and place it within the same `cwd` as your python script.
- Your script should import the `"re"` module.
- Your script should define the function `parse_name`.
 - This function will contain one parameter, `text`, a string representing a single line of the file.
 - This function should use regular expressions in order to capture the first name and last name of the person in question.
 - This function will return a tuple containing the first and last name as strings.
- Your script should define the function `parse_address`.

- This function will contain one parameter, text, a string representing a single line of the file.
- This function should use regular expressions in order to capture the street, city and state of the person in question.
- This function will create and return an address object using the street, city and state identified.
- Your script should define the function `parse_email`.
 - This function will contain one parameter, text, a string representing a single line of the file.
 - This function should use regular expressions in order to capture the email of the person in question.
 - This function will return the email identified.
- Write a class `Address`.
 - This class will have 3 attributes (street, city, state) that are created from the arguments that are passed in when an instance of address is created.
 - This class will not have any methods.
- Write a class `Employee`.
 - This class will have 4 attributes (first_name, last_name, address, email).
 - These attributes will all be created by passing in a row of the file when an instance of employee is created. This meaning, the init method will only have one parameter other than self.
 - The first_name and last_name attributes are created by calling the parse_name function and passing in the parameter of the init method of employee as an argument in the parse_name function call.
 - The address attribute is created by calling the parse_address function and passing in the parameter of the init method of employee as an argument in the parse_address function call.
 - The email attribute is created by calling the parse_email function and passing in the parameter of the init method of employee as an argument in the parse_email function call.
 - This class will not have any methods.
- Write a main function. Your main should have one parameter, path, which is the path to the file that is being parsed.
 - Create an empty list called `employee_list`.
 - Open up the file from the path.
 - For each line in the file, create an instance of employee by passing in the string which is the line of the file that you are reading.
 - Append this instance to the `employee_list`.
 - Within the main, return the `employee_list`
- Write an if `name == main` statement

- Call the main using the "people.txt" file, save the returned value.
- Print this returned value.

Hints

- You will notice that the text file is formatted in a particular way. Use this formatting to your advantage when writing the regular expressions.
- You will notice that as you iterate over the lines in the file, that you are in essence breaking up the strings that you are working on to be just the single lines, meaning, you might find it advantageous to use anchors (^ or \$) when writing regular expressions to indicate that you are matching items at the beginning or ends of the strings.