11% z Lekce 20

Introduction

PYTHON ACADEMY / PROJECT 5: HR SYSTEM [H] / INTRODUCTION

In this project, we will try to create our first application. It should be an application that will behave as a company HR System.

SOLUTION

HR System contains employee related data. Those are for example:

- **Personal data** (name, surname, birth date, social security data etc.)
- **Compensation data** (salary, so called band, benefits)
- **Contract related data** (contract validity, FTE - number of hours worked/week, additional contracts and agreements)
- **Job data** (job title, level, job description, job code etc.)
- **Organisation hierarchy data** (manager, department etc.)
- **Performance data** (Performance evaluation, performance improvement plan etc.)
- and much more..

The main task of our system will be to:

1. store the data we enter into it,
2. retrieve the data based on input criteria,
3. show the data and
4. remove the data.

SOLUTION

Home Page

PYTHON ACADEMY / PROJECT 5: HR SYSTEM [H] / SOLUTION / HOME PAGE

Home page is the interface which is shown to the user once the application is loaded. It provides an offering of actions that the user can request from the application.

Your task is to create a function that will take care of the following behaviour:

1. In our case, our home page should provide the user the following menu:

```
===============================================================
```

2. And prompt the user to choose one of the actions:

```
=========================================================
| Create New Employee | Find Employee | Remove Employee |
=========================================================
PLEASE SELECT YOUR ACTION (or "q" to quit):
```

3. The user should state what action should be performed and hit enter key:

```
=========================================================
| Create New Employee | Find Employee | Remove Employee |
=========================================================
PLEASE SELECT YOUR ACTION (or "q" to quit): Create New Employee
```

4. The application then should provide an interface to enter data concerning new employee (next task)

5. If the user does not enter none of options provided, the application should inform about this and offer the selection again

```
=========================================================
| Create New Employee | Find Employee | Remove Employee |
=========================================================
PLEASE SELECT YOUR ACTION (or "q" to quit): Crete New Employee
I AM SORRY, BUT THERE IS NO SUCH OPTION
=========================================================
| Create New Employee | Find Employee | Remove Employee |
=========================================================
PLEASE SELECT YOUR ACTION (or "q" to quit):
```

Code Solution

Use dropdown feature below if you want to see, how we wrote the code.

Click to see our problem analysis ▼

11% z Lekce 20

Click to see our solution

Enter a New Employee

PYTHON ACADEMY / PROJECT 5: HR SYSTEM [H] / SOLUTION / ENTER A NEW EMPLOYEE

This functionlity should guide the use over the process of data entry for a specific employee. The computer should be asking user for entry into each individual field. Also, we should enter an acronym for the subsidiary (SK or CZ). This will be used in employee ID generation (see the section Employee ID, below).

SOLUTION

An example of interaction with the user could look like this:

```
=======================================================
| Create New Employee |Find Employee |Remove Employee |
=======================================================
PLEASE SELECT YOUR ACTION (or "q" to quit): Create New Employee
Employee Subsidiary (SK,CZ): SK
Enter record for ID: SK348708
1. Personal/State: Czech Republic
1. Personal/Personal Email: heyho@gmail.com
1. Personal/Citizenship: Czech
...
Entry has been completed for ID: SK348708
{'1. Personal': {'Age': 5,
                 'Citizenship': 'Czech',
                 'City': 'dsa',
                 'Country of Residence': 'Czech Republic',
                 'Date of Birth': '1980-01-01',
                 'First Name': 'John',
                 'Last Name': 'Smith',
                 'National ID': 'XC5424234',
                 'Personal Email': 'heyho@gmail.com'
                 'Telephone Number': '123456789',
                 'ZipCode': '00001',
                 '_Password': 'pass123',
                 '_Username': 'johnny'},
 '2. Employee': {'Band': 5,
                 'Contract Beginning': '2016-08-01',
                 'Contract End': '3333-03-03',
                 'Contract Type': 'Indefinite',
                 'Department': 'Administration',
                 'Employment Status': 'Active',
                 'FTE': 1.0,
                 'Full-Time Salary': 23432,
                 'Is Manager': False,
                 'Job Title': Admin 1st Level,
                 'Manager': 'Bob Francis',
                 'Subsidiary': 'SK'}}
Press ENTER to continue
=======================================================
| Create New Employee |Find Employee |Remove Employee |
=======================================================
```

After all the information has been collected, the program should print out the data again for confirmation to the user that everything has been entered successfully and what employee ID has been assigned to the new employee.

Finally the program should return to the home page.

The program should collect the following information:

1. Personal:

First Name, Middle Name Initial, Last Name, Street Address, City,State, Country of Residence, ZipCode, Personal Email, Telephone Number, Username, Password,Date of Birth, Age, Citizenship, National ID

2. Employee:

11% z Lekce 20

(value between 0.0 and 1.0), Full-Time Salary

The following categories are required (meaning that the user has to input at least something otherwise the program will repeatedly prompt the user for the same input):

- First Name,
- Last Name,
- Contract Beginning,
- Contract End,
- FTE,
- Contract Type,
- Manager,
- Department,
- Job Title,
- Band,
- Full-Time Salary

SOLUTION

You as a application developer should decide

- how the program will know, what categories are to be filled
- in what order the categories will be filled
- how to keep asking for required information
- how you will store the information concerning the newly created employee
- how to keep a database of employees (not only one employee)

Employee ID

Each employee should have his/her **unique ID** generated under which we will be able to search for their records. The ID has to be **generated randomly**.

11% z Lekce 20

Code Solution

Use dropdown feature below if you want to see, how we wrote the code.

Click to see our solution

Current code summary 1

PYTHON ACADEMY / PROJECT 5: HR SYSTEM [H] / SOLUTION / CURRENT CODE SUMMARY 1

SOLUTION

Below you can see:

1. Individual functions
2. Global variables
3. Running the current code
4. Entire code

Individual functions

Homepage

Generating header

Creating employee

11% z Lekce 20

Finding employee

Global variables & running the code

Global variables

Running the current code

SOLUTION

Entire code & running the code

Entire code

Find & Show Employee

PYTHON ACADEMY / PROJECT 5: HR SYSTEM [H] / SOLUTION / FIND & SHOW EMPLOYEE

In case the user selects option:

```
==========================================================
| Create New Employee | Find Employee | Remove Employee |
==========================================================
PLEASE SELECT YOUR ACTION (or "q" to quit): Find Employee
```

The program should prompt user for an ID and:

Finally the program should return to the home page.

Code Solution

Use dropdown feature below if you want to see, how we wrote the code.

Click to see our solution

SOLUTION

Remove Employee

PYTHON ACADEMY / PROJECT 5: HR SYSTEM [H] / SOLUTION / REMOVE EMPLOYEE

If the user chooses the following option, the

```
=========================================================
| Create New Employee | Find Employee | Remove Employee |
=========================================================
PLEASE SELECT YOUR ACTION (or "q" to quit): Create New Employee
```

The program should prompt user for an ID and:

1. print out confirmation that the employee has been removed from the database or
2. inform the user that such an employee has not been found

Finally the program should return to the home page.

If the user runs Find Employee trying to find the previously removed employee record, the program should not be able to find that record.

Code Solution

Click to see our solution ▾

Home Page using *args

PYTHON ACADEMY / PROJECT 5: HR SYSTEM [H] / SOLUTION / HOME PAGE USING *ARGS

If you have look at the section on additional information regarding function inputs, you should be now able to reduce the amount of code needed inside the `home_page()` function.

- The clue is that, not all the functions, that are called inside the `home_page()` need the same number of inputs. Therefore, you could maybe first collect arguments needed to run the function selected by the user and then pass them inside in form of argument unpacking.

SOLUTION

- The program would be even more elegant, if you had a mapping among function objects and function name strings. If a user enters a given name, the mapping could return function object associated to the string that the user has entered.

Code Solution

Use dropdown feature below if you want to see, how we wrote the code.

Click to see our solution ▾

Current code summary 2

PYTHON ACADEMY / PROJECT 5: HR SYSTEM [H] / SOLUTION / CURRENT CODE SUMMARY 2

11% z Lekce 20

Individual functions

Homepage

Generating header

SOLUTION

Creating employee

Removing employee

Finding employee

Global variables & running the code

Global variables

Running the current code

11% z Lekce 20

Entire code

Entire code

Implement the display functions

PYTHON ACADEMY / PROJECT 5: HR SYSTEM [H] / SOLUTION / IMPLEMENT THE DISPLAY FUNCTIONS

SOLUTION

It this time to implement the functionality, that will nicely format the employee data printed to the terminal. In this lesson we have learned about string formatting and this is exactly the case, where we can use it.

An example of interaction with our program could now look like this:

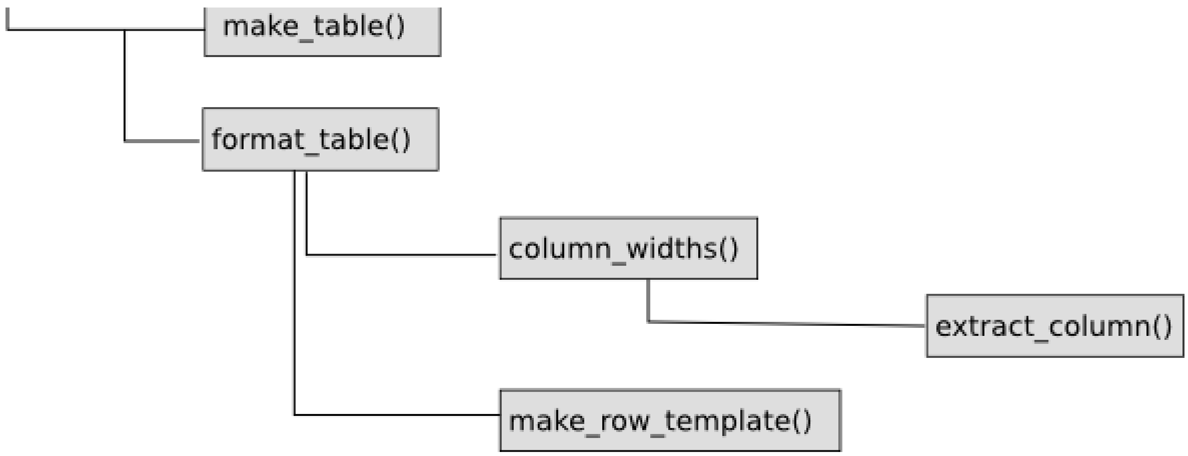
```
=========================================================
| Remove Employee | Find Employee | Create New Employee |
=========================================================
PLEASE SELECT YOUR ACTION (or "q" to quit): Find Employee
Please enter the Employee ID: SK995188
|Country of Residence: United States    ||National ID: 416-74-9086
|
|Street Address: 300 Broad Street       ||Date of Birth: 5/6/1944
|
|_Username:                             ||Personal Email:
MarieAJones@gustr.com    |

|_Password:                             ||Age: 72
|
|ZipCode: 35203                         ||State: Alabama
|
|Last Name: Jones                       ||City: Birmingham
|
|Citizenship:                           ||First Name: Marie
|
|Contract Type: Permanent               ||Manager of Department:
|
|Job Title: Senior Security Engineer    ||FTE: 1
|
|Is Manager: 0                          ||Manager:
|
|Subsidiary: CZ                         ||Band: 6
|
|Employment Status:                     ||Contract Beginning: 6.8.2012
|
|Full-Time Salary: 67523                ||Department: Security - CIA2
|
|Contract End: 3.3.3333
```

We have decided we want to have data ordered in two columns.

Stucture of the code

Below, we visualize the structure of the functions, that are used in order the function `find_employee` can format the data in a neat table. Lines connecting table names tell us, which function is called by which function. So, `find_employee` calls two functions - `make_table()` and then `format_table()` :

11% z Lekce 20



Click to see our solution

SOLUTION

Store Employee Information

PYTHON ACADEMY / PROJECT 5: HR SYSTEM [H] / SOLUTION / STORE EMPLOYEE INFORMATION

In order not to keep the employee data only inside our script, we can store them in a text file for now. The data will still look like a dictionary inside the text file.

However, once the information is read from the file, it will be a string representing a dictionary. In that case, we will need to pass that string into `eval()` built-in function in order the string is converted back into a dictionary.

Your task is to implement function that will load the the content of the file and convert it into a dictionary.

As input data you can use the following employee database: employees.txt

Code Solution

Use dropdown feature below if you want to see, how we wrote the code.

11% z Lekce 20

Current code summary 3

PYTHON ACADEMY / PROJECT 5: HR SYSTEM [H] / SOLUTION / CURRENT CODE SUMMARY 3

Below you can see:

1. Individual functions
2. Global variables
3. Running the current code
4. Entire code

SOLUTION

Individual functions

Homepage

Generating header

Creating employee

Finding employee

Removing employee

11% z Lekce 20

Storing the data

Global variables & running the code

Global variables

Running the current code

SOLUTION

Entire code

Entire code

Implement Error Checking

PYTHON ACADEMY / PROJECT 5: HR SYSTEM [H] / SOLUTION / IMPLEMENT ERROR CHECKING

Implement error checking in the `home_page()` function. Here user can enter invalid action name. So far we had conditional statement to check, whether a action name entered by the user is present among the valid action names in the variable `actions` . Try to change the code by using the `try-except` statement.

Code Solution

Use dropdown feature below if you want to see, how we wrote the code.

11% z Lekce 20

Shortcut Return to Main Menu

PYTHON ACADEMY / PROJECT 5: HR SYSTEM [H] / SOLUTION / SHORTCUT RETURN TO MAIN MENU

We could use `KeyboarInterrupt` exception to tell our program, we want to return to the main menu - `home_page` . `KeyboarInterrupt` error is raised when we press key combination Ctrl + c on our keyboard. So far, if we do this, our program crashes.

Try to implement the key combination Ctrl + c as a shortcut in the HR System, that will take you back to the home page, wherever you are in the program.

SOLUTION

Example of program behaviour:

We can see where the Ctrl + c has been pressed by `^C` symbols.

```
HRS $ python hrs.py
=========================================================
| Create New Employee | Find Employee | Remove Employee |
=========================================================
PLEASE SELECT YOUR ACTION (or "q" to quit): Create New Employee
Employee Subsidiary (SK,CZ): SK
Enter record for ID: SK622629

2. Employee/Employment Status: dad
2. Employee/Subsidiary: das
2. Employee/Manager: ^C
=========================================================
| Create New Employee | Find Employee | Remove Employee |
=========================================================
PLEASE SELECT YOUR ACTION (or "q" to quit):
```

Code Solution

Click to see our solution ▾

Distributing the functionality

PYTHON ACADEMY / PROJECT 5: HR SYSTEM [H] / SOLUTION / DISTRIBUTING THE FUNCTIONALITY

It is time for clean-up. We know that we can distribute specific functionality into isolated python files - modules. Your task is to decide, what modules can be created from the code we have written so far for the HR System project.

SOLUTION

From what we see, there will be some display functionality, some actions required by the user, some global variables and main function, that runs the program.

Solution

Use dropdown feature below if you want to see, how we wrote the code.

Click to see our project structure ▾

Click to see our code solution ▾

Clear the Terminal Screen

PYTHON ACADEMY / PROJECT 5: HR SYSTEM [H] / SOLUTION / CLEAR THE TERMINAL SCREEN

Our goal is to clear the screen and present one-purpose content every time a user would feel more comfortable with that.

An example of such a situation would be:

1. Presenting home page menu:

```
=========================================================
| Find Employee | Remove Employee | Create New Employee |
=========================================================
PLEASE SELECT YOUR ACTION (or "q" to quit):
```

2. Creating new employee - Data Entry

```
CREATED RECORD FOR ID: SK827231
|Age: asd                  ||City: asd                     |
|ZipCode: das              ||Country of Residence: asd     |
|First Name: d             ||Street Address: d             |
|National ID: adsas        ||Date of Birth: dsad           |
|Personal Email: d         ||State: asd                    |
|Citizenship: sa           ||_Password: asas               |
|Telephone Number: as      ||Last Name: as                 |
|_Username: d              ||Subsidiary: d                 |
|Contract Type: d          ||Job Title: d                  |
|Contract End: asd         ||Department: das               |
|Band: asd                 ||Contract Beginning: as        |
|Is Manager: as            ||Manager of Department: d      |
|Employment Status: das    ||FTE: asd                      |
|Full-Time Salary: as      ||Manager: as                   |
Thank you, entry has been completed for ID: SK827231
Press ENTER to continue
```

3. Creating new employe - Summary of entered values

```
1. Personal/First Name: das
1. Personal/Last Name: d
```

```
1. Personal/State: asd
1. Personal/Country of Residence: as
1. Personal/ZipCode: d
1. Personal/Personal Email: asd
1. Personal/Telephone Number: as
1. Personal/_Username: d
1. Personal/_Password: asd
1. Personal/Date of Birth:
1. Personal/Date of Birth: asd
1. Personal/Age: as
1. Personal/Citizenship: d
1. Personal/National ID: sad
2. Employee/Employment Status: a
2. Employee/Subsidiary: d
2. Employee/Manager: asd
2. Employee/Department: asd
2. Employee/Job Title: as
2. Employee/Band: d
2. Employee/Contract Beginning:
2. Employee/Contract Beginning: asd
2. Employee/Contract Type: sa
2. Employee/Contract End:
2. Employee/Contract End: asad
2. Employee/Is Manager: a
2. Employee/Manager of Department: dd
2. Employee/FTE: asa
2. Employee/Full-Time Salary: sd
```

4. Searching for an employee - Presenting the results

```
AT100866
|City: Birmingham                       ||Middle Name Initial: A
|
|Telephone Number: 205-442-2958         ||First Name: Marie
|
|_Password:                             ||_Username:
```

```
Street          |
|Last Name: Jones                       ||State: Alabama
|
|National ID: 416-74-9086               ||Personal Email:
MarieAJones@gustr.com    |
|Date of Birth: 5/6/1944                ||Country of Residence: United
States      |
|Citizenship:                           ||ZipCode: 35203
|
|Department: Security – CIA2            ||FTE: 1
|
|Contract End: 3.3.3333                 ||Is Manager: 0
|
|Employment Status:                     ||Subsidiary: CZ
|
|Job Title: Senior Security Engineer    ||Band: 6
|
|Manager of Department:                 ||Contract Beginning: 6.8.2012
|
|Manager:                               ||Contract Type: Permanent
|
|Full-Time Salary: 67523                |
Press ENTER to continue
```

Code Solution

Use dropdown feature below if you want to see, how we wrote the code.

Click to see our solution ▼

11% z Lekce 20

PYTHON ACADEMY / PROJECT 5: HR SYSTEM [H] / SOLUTION / JSON DATABASE

Adjust your HR System code in order it stores the data in JSON file from now on. You should load and store the `employees` dictionary using `json` library.

Do not forget, you should create a json file - copy of the original `txt` file.

Before you get into it, you should know that simply creating a copy of the `txt` file and changing the suffix to `json` won't be enough.

Code Solution

Use dropdown feature below if you want to see, how we wrote the code.

SOLUTION

Click to see our solution ▾

Database Backup

PYTHON ACADEMY / PROJECT 5: HR SYSTEM [H] / SOLUTION / DATABASE BACKUP

Your task is to add a new functionality to the system. It is possible, that form time to time, the user would like to create a backup of the json database just in case the "production" database would get corrupted.

Therefore if the user launches the program adding a keywordd `backup`, a new json file copy should be created inside the folder called `Backups` located in the current working directory of `hrs.py`:

Example of running the script with backup option:

```
$ python hrs.py backup
New back-up file created: employees_2017-01-01T16:44:32.915202'
```

11% z Lekce 20

today's date following the so called ISO format.

```
from datetime import datetime as dt
```

And the function is used as follows:

```
>>> dt.isoformat(dt.today())
```

We used another function from the datetime library called `today()` , that returns today's date and time information in form of the datetime object.

Code Solution

SOLUTION

Use dropdown feature below if you want to see, how we wrote the code.

Click to see our solution

Entire Solution

PYTHON ACADEMY / PROJECT 5: HR SYSTEM [H] / SOLUTION / ENTIRE SOLUTION

Module actions.py

Module display.py

Module config.py

11% z Lekce 20

Konec kurzu

ZPÁTKY NA PŘEHLED KURZU

SOLUTION