Python: The Easy Way

# Lab 2

# Lab 3 Assignments

## we chose for you

Write a program that choose a random website name from list and open it in your browser :

### Input

No Input

**Hint**: Use Standard Library Modules

### Output

Website Visit on Web Browser

# Story

“Samy is an Employee, He works in ITI and He has a car. He goes everyday except weekends to ITI Smart Village Office by his fiat 128 car”

# Story Members

Samy

Employee

Person

ITI

Office

Fiat128

Car

# Additional Info

- **ITI** is an Office that has many employees and **Samy** is one of them.

- **Samy** is an Employee and He has a **fiat 128** Car.

- The distance from **Samy** Home to **ITI** Smart Village Office is **20 km**.

- **Samy** should arrive to **ITI** at before **9:00** unless that he will be late.

- **Velocity** (v) = Distance(d) / time(t).

- **FuelRate** decrease by **10%** every **10km** distance.

## Setup All Classes

1- Create the following classes:

- Person Class:
  - attributes (name, money, mood, healthRate).
  - methods (sleep, eat, buy).

- Employee Class (is a Person):
  - attributes (id , car, email, salary, distanceToWork)
  - methods (work, drive, refuel, send_mail)

- Office Class:
  - attributes (name, employees)
  - methods (get_all_employees, get_employee, hire, fire, calculate_lateness, deduct, reward)

- Car Class:
  - attributes (name, fuelRate, velocity)
  - methods (run, stop)

# Implement Employee Methods

2- Implement the following methods:

- sleep (hours): - Method in Person Class (7 hours → happy, <7 hours → tired, >7 hours → Lazy)
- eat (meals): - Method in Person Class (3 meals → 100% hth , 2 meals →75% , 1 meal → 50%)
- buy (items): - Method in Person Class (1 item → decrease money 10 L.E)
- work (hours): - Method in Employee Class (8 hours→happy, >8 hours → tired,<8 hours → Lazy)

- send_mail(to, subject, msg, receiver_name): *(optional)*

  - Create Email File like the next page specification (Email Composer)

- salary Property: must be 1000 or more.
- email Property: must be valid.
- healthRate Property: must be between 0 to 100.
- There is moods class variable which is tuple of happy, tired and lazy

# Email Composer

*Write a program that generate a file that contains a structured email message*

## Input

strings

Email Subject

from@mail.com

to@mail.com

Mohamed

## Output

File

```
From: from@mail.com
To: to@mail.com

Hi, Mohamed
This is an email tamplate
thanks
```

*Email subject*

## Implement Car Methods

3 - Implement the following methods:

- **drive (distance):**

  - Method in Employee Class (Give the order to run method and give it distance and velocity).

- **refuel (gasAmount = 100):**

  - Method in Employee Class (add gasAmount to **fuelRate**).

- **run (velocity, distance):**

  - Method in Car Class (When invoked it decreases the **fuelRate** and change the velocity to the input parameter of velocity . And it invoke the stop method and give it the remain distance (It is possible to stop before arrive the destination because **fuelRate** become 0).

- **stop ():**

  - Method in Car Class (Stop make the velocity changed to 0 and print notification with the remain distance or that you arrive the destintation )

- **Velocity** Property: must be between 0 to 200.

- **Fuel Rate** Property: must be between 0 to 100.

# LAB +50 pts.

## Implement Office Methods

3- Implement the following methods:

- **get_all_employees ():** Method in Office Class (Return a list of the current Employees)
- **get_employee (empId):** Method in Office Class (Return the Employees of given id)
- **hire (Employee):** Method in Office Class (Hire the given Employee)
- **fire (empId):** Method in Office Class (Fire Employee with the given id)
- **deduct (empId, deduction):** Method in Office Class (Deduce Money from salary from Employee)
- **reward (empId, reward):** Method in Office Class (add Money to salary from Employee)
- **check_lateness (empId, moveHour):** Method in Office Class (Check if employee is late or not and deduce if he is late -10 and reward if he is  not late +10)
- **calculate_lateness (targetHour , moveHour, distance, velocity): Static** Method in Office Class (Calculate If employee is late or not )
- employeesNum class variable which declared the number of Employees in all offices.
- change_emps_num **(num)** class method which modify the number of  Employees in all offices.

# LAB +100 pts.

## Save Office data as JSON

Save the previous data of the ITI office in a json file.

# Thank You