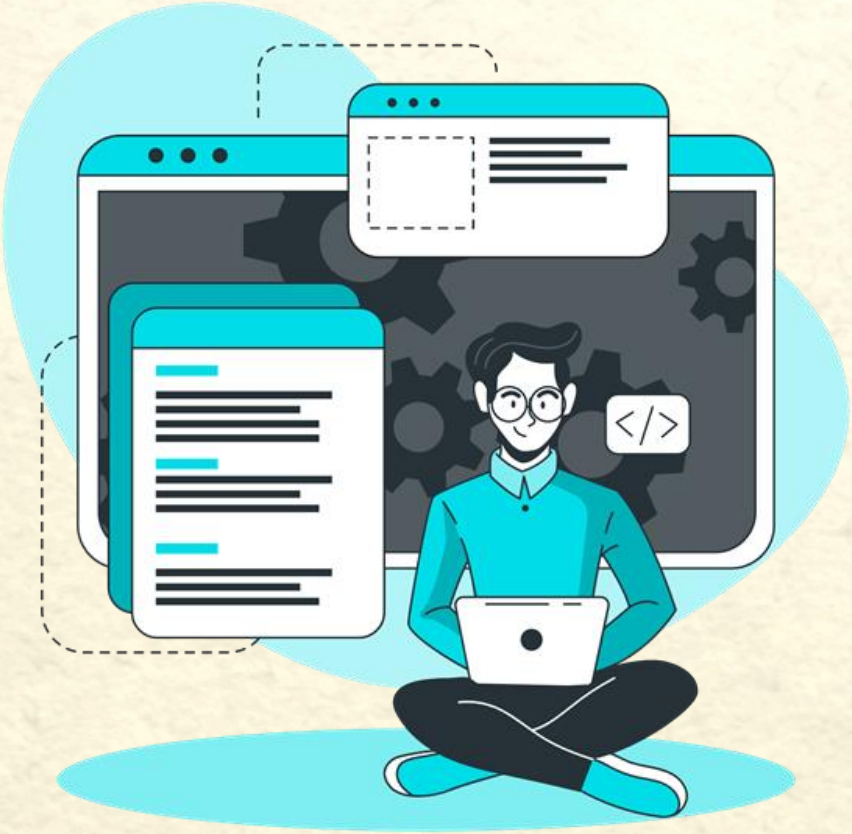
مکتب شریف

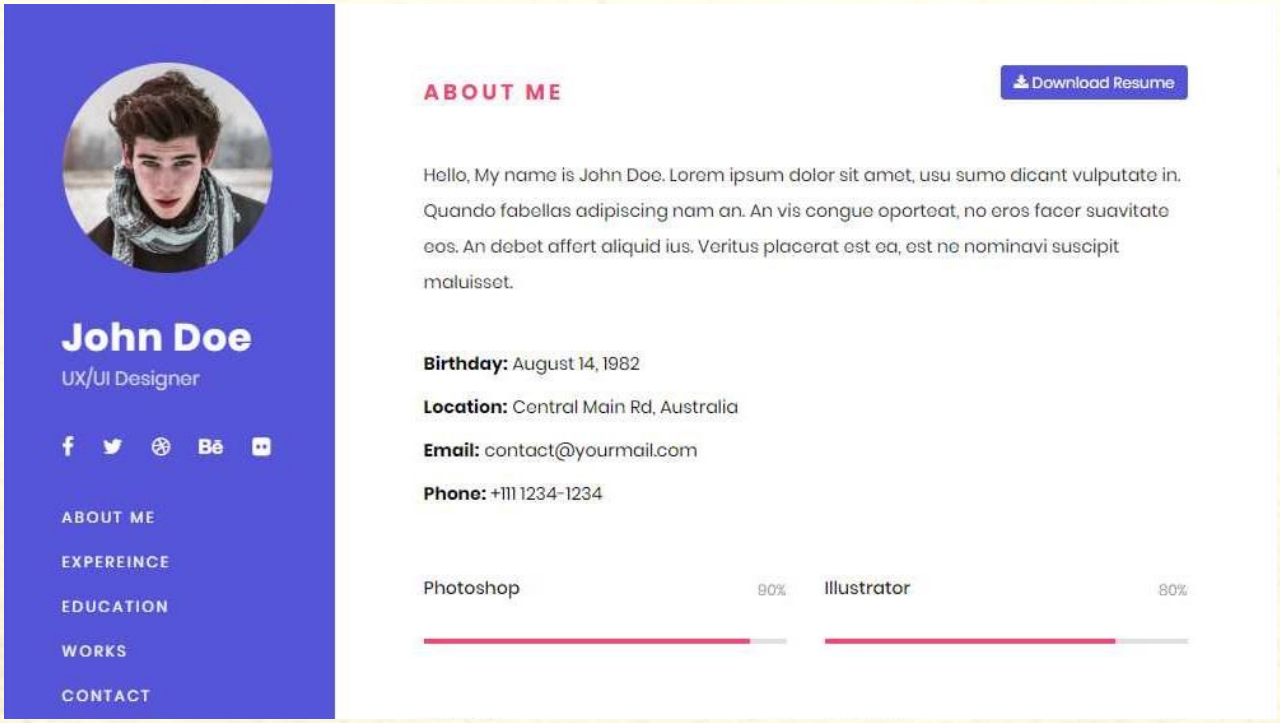
اولین بوتکمپ آموزشی - استخدامی ایران

مکتب ۹۳

هفته چهاردهم



1. **پورتفولیو ضمیمه شده را تنها و تنها با استفاده از HTML و CSS به ازای مشخصات خودتان طراحی نمایید.**



2. با مطالعه مستندات API زیر، طول و عرض جغرافیایی شهر خود را با استفاده از ماژول requests به آن ارسال کرده و سپس با پردازش JSON دریافت شده، زمان طلوع و غروب آفتاب را در شهر خودتان بدست آورید.

https://sunrise-sunset.org/api

در صورت بروز خطا و یا مشکل پروتکل خود را از https به http تغییر داده تا اطلاعات مربوطه دریافت شود.

3. فایلهای person.py و sample.py که در ضمیمه هستند را درنظر بگیرید؛

ابتدا پاسخ دهید که بنظر شما توسعه دهنده مرتکب چه خطاهایی شده؟ فایل اجرایی sample.py میباشد.

دلایل خود را با ذکر تمام مشکلات پیش آمده توضیح داده و کد را با کامنت گذاری مناسب تصحیح نمایید.

- حداقل سطح لاگ در کلاس Person برابر DEBUG و در تابع sub برابر INFO باشد.
- لاگ کلاس Person در فایل person.log و لاگ تابع sub در فایل sample.log باشد.
- قالب لاگهای برای کلاس Person مطابق تصویر 1 و برای تابع sub مطابق تصویر 2 باشد؛

همچنین لاگهای تابع sub علاوه بر فایل در کنسول با سطح ERROR و قالب زیر به نمایش درآید.

asctime – levelname - message

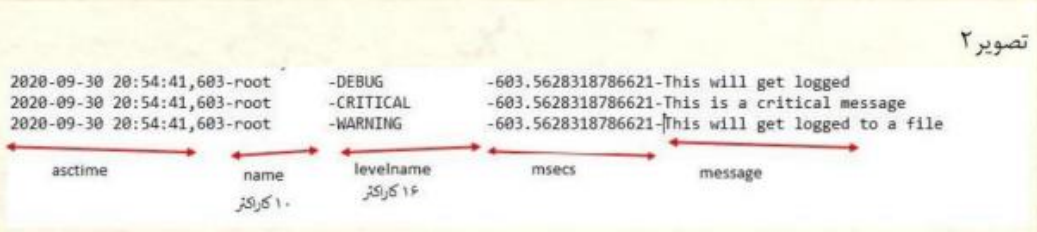
در انتها تابعی بنویسید که ورودی آن نام فایل و رشته ای به نام level باشد؛

خروجی تابع میبایست تعداد لاگهای رخ داده با حداقل سطح level در فایل ورودی باشند.

سپس این تابع را به ازای سطوح مختلف بر روی فایلهای لاگی که تولید کردید تست کرده و خروجی بگیرید.

تصویر ۱

تصویر ۲



توجه داشته باشید که در قالبهای بالا name و levelname به ترتیب همواره 10 و 16 کاراکتر باید باشند.

4. در این سوال، یک ماژول پایتونی به شما داده شده که شامل یک کلاس برای مدیریت یک سیستم ساده انبارداری است. وظیفه شما این است که با استفاده از pytest، تست های واحد (Unit Test) مناسب بنویسید تا از صحت کارایی سیستم انبارداری اطمینان حاصل کنید. مطمئن شوید که Edge Case ها و Exception های احتمالی را نظر بگیرید.

یک فایل به نام test_inventory.py بسازید و در آن Unit Test های مناسب برای کد زیر را با استفاده از چارچوب pytest بنویسید.

inventory.py:

```
class Inventory:
    def __init__(self):
        self.items = {}

    def add_item(self, name, quantity):
        if name in self.items:
            self.items[name] += quantity
        else:
            self.items[name] = quantity

    def remove_item(self, name, quantity):
        if name not in self.items:
            raise KeyError("Item not found")
        if quantity > self.items[name]:
            raise ValueError("Insufficient quantity")
        self.items[name] -= quantity
        if self.items[name] == 0:
            del self.items[name]

    def get_item_quantity(self, name):
        return self.items.get(name, 0)

    def get_total_items(self):
        return sum(self.items.values())
```

5. نوشتن کد برای پاس کردن تست‌های داده‌شده با استفاده از TDD

برای این سوال، مجموعه‌ای از Unit Test ها برای یک ابزار که مساحت و محیط مستطیل را محاسبه می‌کند، به شما داده می‌شود. وظیفه شما این است که قابلیت‌ها را در یک کلاس به نام Rectangle پیاده‌سازی کنید به طوری که از تمام تست‌ها عبور کند. علاوه بر این، در مورد TDD و منافع آن بیاموزید و آن را در فرآیند توسعه خود به کار ببرید.

یک فایل به نام rectangle.py بسازید و کلاس Rectangle را پیاده‌سازی کنید به طوری که از تمام تست‌های داده‌شده در test_rectangle.py عبور کند.

(با TDD آشنا شوید و سعی کنید آن را در فرآیند توسعه خود پیاده‌سازی کنید. TDD این مراحل را دنبال میکند:

1. یک مورد تست ناموفق بنویسید.
2. حداقل مقدار کد لازم برای قبول شدن تست را بنویسید.
3. در صورت لزوم کد را Refactor کنید.
4. این فرآیند را برای هر قابلیت تکرار کنید.)

به یاد داشته باشید که هدف اصلی این تمرین، تمرین کردن TDD و یادگیری منافع آن است.

```
import pytest
from rectangle import Rectangle

def test_area():
    rectangle = Rectangle(4, 5)
    assert rectangle.area() == 20

def test_perimeter():
    rectangle = Rectangle(4, 5)
    assert rectangle.perimeter() == 18

def test_invalid_dimensions():
    with pytest.raises(ValueError, match="Dimensions must be positive"):
        Rectangle(-1, 5)
    with pytest.raises(ValueError, match="Dimensions must be positive"):
        Rectangle(4, -5)
    with pytest.raises(ValueError, match="Dimensions must be positive"):
        Rectangle(-1, -5)
```

نکات

- مهلت ارسال تمرین تا پایان روز چهار شنبه 1402/02/20 است.
- زین پس تمامی تحویل تمارین تنها و تنها از طریق گیتهاب(Github)صورت میپذیرد.
- بدین منظور لازم است یک مخزن(repository) خصوصی(private) ساخته شود.
- آدرس ریپازیتوری را در کارتابل شخصی خود اعلام کرده و تیم تدریس را بعنوان collaborator بیفزایید:
  - پوریا منصوری
  - سجاد کاشی
  - آرین همدانی
  - امیرحسین حسنی
  - الهه پاسبان
- مالک و معیار ارزیابی تاریخ آخرین commit شما میباشد(بصورت استاندارد و اصولی کامیت انجام شود).
- قطعا هدف از تمارین صرفا رسیدن به جواب نهایی نیست و تمیز بودن کد و خلاقیتی که در انجام آن به خرج می دهید از اهمیت و امتیاز بالایی برخوردار است. ارائه راه حل کلی و عمومی برای یک مسئله که حالت های مختلف آن را در نظر بگیرد و فراتر از خواسته ی مسئله است (خواسته ی مسئله گسترش داده شود یا حالت های خاص مسئله را پوشش دهد. قطعا مشمول امتیاز بیشتری خواهد شد.)

در صورتیکه سوالی دارید در گروه راکت چت بپرسید.