



Re-requesting a Vanity Plate

Problem set



در یک فایل به نام `plates.py`، سوال Vanity Plates را از پرابلمست هفته‌ی دوم، دوباره پیاده سازی کنید، بازنویسی کد خود را به صورت زیر بنویسید بطوریکه تابع `is_valid()` یک رشته (string) را به عنوان ورودی دریافت کند و یک Boolean (True یا False) را به عنوان خروجی بازگرداند بدین صورت که اگر ورودی، شرایط مورد نظر را داشت، خروجی True و اگر شرایط مورد نظر را نداشت، خروجی False برمی‌گرداند. اما تابع `main()` تنها زمانی فرخوانی می‌شود که عبارت `__name__` برابر با `"__main__"` باشد.

```
def main():  
    ...  
  
def is_valid(s):  
    ...  
  
if __name__ == "__main__":  
    main()
```

در فایل `test_plates.py`، چهار یا بیشتر از چهار تابع پیاده سازی کنید تا به طور جمعی پیاده سازی تابع `is_valid()` را بطور کامل تست کنید. اسم هر کدام از توابع بایستی با `test_` شروع شود تا شما بتوانید تست های خود را با دستور زیر اجرا کنید:

```
pytest test_plates.py
```

نکته ▼

- استفاده از فایل `test_plates` را فراموش نکنید.

```
import plates
```

یا

```
from plates import is_valid
```

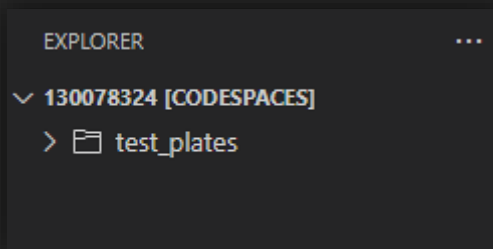
- با این کار می‌توانید `is_valid` را در تست‌های خود فراخوانی کنید.
- همچنین دقت کنید که از `print` فقط در تابع `main` می‌توانید در این سناریو استفاده کنید و در دیگر موارد باید از `return` استفاده کنید.

وارد code.cs50.io شوید، سپس بر روی پنجره‌ی Terminal کلیک کنید. توجه داشته باشید که دستور پنجره‌ی Terminal شما باید به صورت زیر باشد:

```
$
```

سپس کد زیر را اجرا کنید تا یک پوشه به نام test_plates در codespace ایجاد شود:

```
$ mkdir test_plates
```

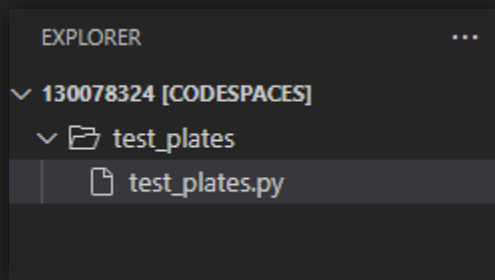


سپس `cd test_plates` را اجرا کنید تا به پوشه‌ی `test_plates` منتقل شوید. اکنون شما باید آدرس `$ test_plates/` را در Terminal مشاهده کنید:

```
$ cd test_plates  
test_plates/ $
```

حالا می‌توانید دستور `code test_plates.py` را اجرا کنید تا فایل‌ی با نام `test_plates.py` ایجاد شود:

```
test_plates/ $ code test_plates.py
```



اکنون می‌توانید در این فایل برنامه‌ی خود را بنویسید.

برای استفاده از تست های خود `pytest test_plates.py` را اجرا کنید و دقت کنید که حتما یک کپی از فایل `plates.py` در همان پوشه داشته باشید. سعی کنید نسخه‌های درست و غلط `plates.py` را استفاده کنید تا بفهمید چقدر تست‌های شما، خطاها را تشخیص می‌دهد:

- مطمئن شوید که نسخه درستی از `plates.py` را دارید. تست های خود را با `pytest test_plates.py` اجرا کنید. `pytest` باید نشان دهد که تمام تست‌های شما قبول هستند.
- نسخه درست `plates.py` را در نظر بگیرید و مقادیر بازگشتی که برنامه شما تبدیل می‌کند را تغییر دهید. احتمالاً برنامه‌ی شما برای مثال عبارت `Valid` به طور اشتباهی برای هر طولی از `license plate` چاپ کند. اکنون تست‌های خود را با `pytest test_plates.py` اجرا کنید. `pytest` باید نشان دهد که حداقل یکی از تست‌های شما مورد قبول نبوده است.

❖ شما می‌توانید از آدرس زیر برای بررسی کردن کد خود استفاده کنید. CS50 از این برنامه برای آزمایش کد شما استفاده می‌کند. از این دستور استفاده کنید تا کدهایتان را امتحان کنید.

```
$ check50 cs50/problems/2022/python/tests/plates
```

لبخند های سبز به این معنی هستند که برنامه‌ی شما در تست قبول شده و اخم های قرمز یعنی برنامه‌ی شما دارای ایراد هست. با مراجعه به check50 URL می‌توانید خروجی مورد انتظار و خروجی برنامه‌ی خود را بررسی کنید.

با اجرا کردن صورت زیر در Terminal پاسخ خود را ارسال کنید.

```
$ submit50 cs50/problems/2022/python/tests/plates
```

CS50x Iran

Harvard's Computer Science 50x Iran

