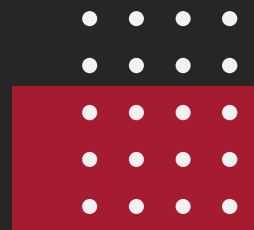
CS50x Iran

Refueling

Problem set



پاییز و زمستان ۱۴۰۴

در یک فایل به نام fuel.py، سوال Fuel Gauge را از پرابلم‌ست هفته‌ی سوم دوباره پیاده‌سازی کنید و کد خود را با توجه به ساختار زیر بازسازی کنید:

- تابع `convert` یک رشته (str) با فرمت X/Y را به عنوان ورودی دریافت می‌کند، که در آن X و Y هر دو عدد صحیح هستند، و این تابع آن کسر را به عنوان درصد گرد شده به نزدیک‌ترین عدد صحیح بین 0 تا 100 (شامل حدهای بالا و پایین) برمی‌گرداند. اگر X و Y یا، X یا Y یک عدد صحیح نباشند، یا اگر X بزرگتر از Y باشد، تابع `convert`، باید یک `ValueError` برگرداند. اگر Y برابر با صفر باشد، تابع `convert` باید یک `ZeroDivisionError` را برگرداند.

- تابع gauge یک عدد صحیح دریافت می‌کند و یک رشته را برمی‌گرداند که:

- اگر این عدد کمتر یا مساوی 1 باشد، رشته "E" بازگردانده می‌شود.
- اگر این عدد بیشتر یا مساوی 99 باشد، رشته "F" بازگردانده می‌شود.
- در غیر این صورت، رشته "Z%" بازگردانده می‌شود، جایی که Z برابر همان عدد ورودی است. مانند:

```
def main():
    ...

def convert(fraction):
    ...

def gauge(percentage):
    ...

if __name__ == "__main__":
    main()
```

سپس، در یک فایل به نام test_fuel.py، دو یا بیشتر از دو تابع پیاده‌سازی کنید که به طور کامل توابع `convert` و `gauge` را تست کنند. نام هر یک از این توابع باید با `test_` شروع شود تا شما بتوانید تست‌های خود را با استفاده از دستور زیر اجرا کنید:

```
pytest test_fuel.py
```

▼ نکته

- استفاده از فایل `twttr` را فراموش نکنید.

```
import fuel
```

یا

```
from fuel import convert, gauge
```

با این کار می‌توانید `convert` و `gauge` را در تست های 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.
- توجه کنید که با استفاده از pytest می‌توانید بررسی کنید که آیا یک تابع خطایی ایجاد کرده است یا خیر.

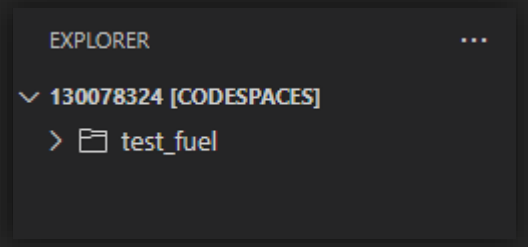
5 قبل از شروع توجه کنید:

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_fuel در codespace ایجاد شود :

```
$ mkdir test_fuel
```

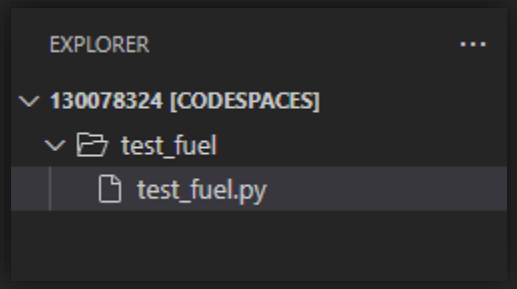


سپس `cd test_fuel` را اجرا کنید تا به پوشه‌ی test_fuel منتقل شوید. اکنون شما باید آدرس `test_fuel/ $` را در Terminal مشاهده کنید:

```
$ cd test_fuel
test_fuel/ $
```

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

```
test_fuel/ $ code test_fuel.py
```



اکنون می‌توانید در این فایل برنامه‌ی خود را بنویسید.

برای استفاده از تست های خود `pytest test_fuel.py` را اجرا کنید و دقت کنید که حتما یک کپی از فایل fuel.py در همان پوشه داشته باشید:

- مطمئن شوید که نسخه درستی از fuel.py را دارید. تست های خود را با `pytest test_fuel.py` اجرا کنید. pytest باید نشان دهد که تمام تست‌های شما درست هستند.
- نسخه درست fuel.py را در نظر بگیرید و مقادیر بازگشتی که برنامه شما تبدیل می‌کند را تغییر دهید. برای مثال ممکن است برنامه شما به اشتباه یک رشته (str) را به جای عدد صحیح (int) برگرداند. اکنون تست‌های خود را با `pytest test_fuel.py` اجرا کنید. pytest باید نشان دهد که حداقل یکی از تست‌های شما مورد قبول نبوده است.
- به طور مشابه، نسخه صحیح fuel.py را تغییر دهید و مقادیر بازگشتی نشانگر را تغییر دهید. برای مثال ممکن است برنامه شما به اشتباه یک % را در رشته (str) حاصل حذف کند. تست‌های خود را با دستور `pytest test_fuel.py` اجرا کنید. pytest باید نشان‌دهد که حداقل یکی از تست‌های شما ناموفق 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/fuel
```

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/fuel
```

CS50x Iran

Harvard's Computer Science 50x Iran