



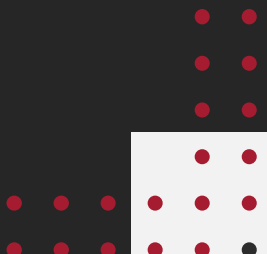
# Fuel Gauge

Problem set



نشانگرهای سوخت یا Fuel Gauge مقدار بنزین باقی مانده در باک را نشان می‌دهند. (اغلب این نشانگرها، بخشی از دایره را به چند قسمت تقسیم کرده و بر اساس اینکه نشانگر بر روی کدام قسمت باشد، مقدار درصد باقی مانده بنزین را به طور تقریبی نشان می‌دهند.)

برای مثال،  $1/4$  نشان‌دهنده این است که 25 درصد از باک پر شده است.  $1/2$  نشان‌دهنده این است که 50 درصد از باک پر شده است و  $3/4$  نشان‌دهنده این است که 75 درصد از باک پر شده است.



در فایلی به نام `fuel.py`، برنامه‌ای بنویسید که کاربر صورت و مخرج یک کسر را به صورت  $x/y$  وارد کند (به فرمی که  $x$  و  $y$  یک عدد `integer` باشند) و در خروجی درصد معادل آن کسر (گرد شده به نزدیک‌ترین عدد یا `integer`) نشان دهد که این عدد مقدار بنزین باقی‌مانده در باک را مشخص می‌کند. اگر مقدار بنزین باقی‌مانده 1% یا کمتر بود، در خروجی حرف `E` را نشان دهد که نمایانگر خالی بودن باک است. اگر بنزین باقی‌مانده 99% یا بیشتر بود، در خروجی حرف `F` را نشان دهد که نمایانگر پُر بودن باک است.

اگر مقادیر  $x$  و  $y$  که از کاربر دریافت می‌شود `integer` نبود، یا  $x$  بزرگ‌تر از  $y$  بود، یا  $y$  برابر صفر بود، دوباره از کاربر درخواست شود تا کسر موردنظر ( $x/y$ ) را وارد کند. (لزومی ندارد که مقدار  $y$  حتماً 4 باشد!)

در نوشتن برنامه خود، مطمئن شوید که هر گونه `exception` مثل `ValueError` یا `ZeroDivisionError` را پوشش دهید.

نکته ▼

- به یاد داشته باشید که string توابع مختلفی دارد به ویژه `split`. توضیحات بیشتر را در اینجا می‌توانید ببینید: [docs.python.org/3/library/stdtypes.html#string-methods](https://docs.python.org/3/library/stdtypes.html#string-methods)
- توجه داشته باشید که می‌توانید دو exception را به طور جداگانه با کدهایی مانند زیر مدیریت کنید:

```
try:
    ...
except ValueError:
    ...
except ZeroDivisionError:
    ...
```

- یا می‌توانید دو exception را به طور همزمان با کد زیر مدیریت کنید:

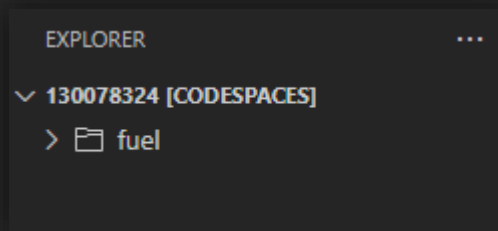
```
try:
    ...
except (ValueError, ZeroDivisionError):
    ...
```

وارد [code.cs50.io](https://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 fuel در codespace ایجاد شود:

```
$ mkdir fuel
```

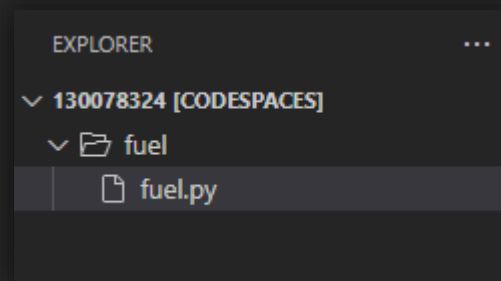


سپس `cd fuel` را اجرا کنید تا به پوشه‌ی `fuel` منتقل شوید. اکنون شما باید آدرس `$ fuel/` را در Terminal مشاهده کنید:

```
$ cd fuel  
fuel/ $
```

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

```
fuel/ $ code fuel.py
```



اکنون می‌توانید در این فایل برنامه‌ی خود را بنویسید.

اکنون می‌خواهیم برنامه‌ی نوشته شده‌ی خود را تست کنیم:

- دستور `python fuel.py` را اجرا کنید. 3/4 را تایپ کنید و کلید Enter را فشار دهید. خروجی شما برابر می‌شود با: 75%
- دستور `python fuel.py` را اجرا کنید. 1/4 را تایپ کنید و کلید Enter را فشار دهید. خروجی شما برابر می‌شود با: 25%
- دستور `python fuel.py` را اجرا کنید. 4/4 را تایپ کنید و کلید Enter را فشار دهید. خروجی شما برابر می‌شود با: F
- دستور `python fuel.py` را اجرا کنید. 0/4 را تایپ کنید و کلید Enter را فشار دهید. خروجی شما برابر می‌شود با: E

اکنون می‌خواهیم برنامه‌ی نوشته شده‌ی خود را تست کنیم:

- دستور `python fuel.py` را اجرا کنید.  $4/0$  را تایپ کنید و کلید Enter را فشار دهید. برنامه‌ی شما باید یک `ZeroDivisionError` را مدیریت کند و یا از کاربر ورودی دیگری بخواهد.
- دستور `python fuel.py` را اجرا کنید. `three/four` را تایپ کنید و کلید Enter را فشار دهید. برنامه‌ی شما باید یک `ValueError` را مدیریت کند و یا از کاربر ورودی دیگری بخواهد.
- دستور `python fuel.py` را اجرا کنید.  $1.5/3$  را تایپ کنید و کلید Enter را فشار دهید. برنامه‌ی شما باید یک `ValueError` را مدیریت کند و یا از کاربر ورودی دیگری بخواهد.
- دستور `python fuel.py` را اجرا کنید.  $5/4$  را تایپ کنید و کلید Enter را فشار دهید. برنامه‌ی شما باید از کاربر ورودی دیگری بخواهد.



❖ 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/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/fuel
```

# CS50x Iran

Harvard's Computer Science 50x Iran

