



# Back to the Bank

Problem set



در فایلی به اسم `bank.py`، پرابلم ست Home Federal Savings Bank از پرابلم ست هفته ی اول را مجدداً به نحوی پیاده سازی کنید که شرایط زیر را داشته باشد.

برنامه شما باید یک `string` ورودی بگیرد به طوری که:

- اگر با `hello` شروع شود، 0 برگرداند.
- اگر با `h` شروع شود (در صورتی که `hello` نباشد)، 20 برگرداند.
- و در غیر این صورت 100 برگرداند.

همچنین تابع `print` فقط میتواند از طریق `main` فراخوانی شود.

```
def main():  
    ...  
def value(greeting):  
    ...  
if __name__ == "__main__":  
    main()
```

سپس در فایلی به نام `test_bank.py`، سه و یا بیشتر از سه تابع پیاده سازی کنید که با استفاده از آنها مقدار `value` را تست کنید. اسم هر کدام از آنها باید با `test_` شروع شود تا بتوانید کد زیر را اجرا کنید.

```
pytest test_bank.py
```

نکته ▼

- استفاده از فایل bank را فراموش نکنید.

```
import bank
```

یا

```
from bank import value
```

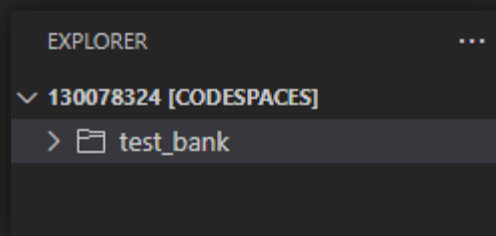
- با این کار می‌توانید `value` را در تست‌های خود فراخوانی کنید.
- همچنین دقت کنید که از `print` فقط در تابع `main` می‌توانید در این سناریو استفاده کنید و در دیگر موارد باید از `return` استفاده کنید.

وارد [code.cs50.io](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 test\_bank در codespace ایجاد شود:

```
$ mkdir test_bank
```

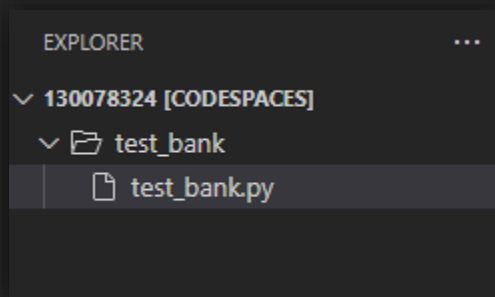


سپس `cd test_bank` را اجرا کنید تا به پوشه‌ی `test_bank` منتقل شوید. اکنون شما باید آدرس `test_bank/ $` در Terminal مشاهده کنید:

```
$ cd test_bank  
test_bank/ $
```

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

```
test_bank/ $ code test_bank.py
```



اکنون می‌توانید در این فایل برنامه‌ی خود را بنویسید.

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

- مطمئن شوید که نسخه درستی از `bank.py` را دارید. تست های خود را با `pytest test_bank.py` اجرا کنید. `pytest` باید نشان دهد که تمام تست های شما درست هستند.

- نسخه درست `bank.py` را در نظر بگیرید و مقادیر متناظر برای هر خوشامدگویی را تغییر دهید. برای مثال ممکن است برنامه شما به اشتباه 100 دلار به مشتری که با Hello خوشامد می گوید، بدهد و 0 دلار به مشتری که با What's up! استقبال می کند، بدهد.

اکنون تست های خود را با `pytest test_bank.py` اجرا کنید. `pytest` باید نشان دهد که حداقل یکی از تست های 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/bank
```

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/bank
```

# CS50x Iran

Harvard's Computer Science 50x Iran

