



Meal Time

Problem set



فرض کنید در کشوری هستید که در آن مرسوم است صبحانه را بین ساعت 7:00 تا 8:00، ناهار را بین 12:00 تا 13:00 و شام را بین ساعت 18:00 تا 19:00 میل کنید. آیا خوب نیست اگر برنامه ای داشته باشید که به شما بگوید چه زمانی چه بخورید؟

در meal.py، برنامه‌ای بنویسید که از کاربر وارد کردن زمانی را درخواست کند و به عنوان خروجی breakfast time، lunch time یا dinner time را نمایش دهد. اگر وقت صرف غذا نیست، اصلاً چیزی را به عنوان خروجی چاپ نکنید. فرض کنید ورودی کاربر در قالب زمانی 24 ساعت به صورت "##:##" یا "###:###" قالب بندی می‌شود. و فرض کنید که هر وعده غذایی دارای یک محدوده زمانی است. به عنوان مثال، اگر ساعت 7:00، 7:01، 7:59 یا 8:00 باشد، یا هر زمانی در این بین، وقت صبحانه است و خروجی باید breakfast time باشد.

برنامه خود را مطابق زیر ساختار دهید که در آن `convert` تابعی است که می‌توان آن را در `main` فراخوانی کرد. این تابع مقدار `time` (یک `str` در قالب 24 ساعته) را دریافت کرده و معادل آن را به فرمت عدد اعشاری یا `float` تبدیل می‌کند. به عنوان مثال، مقدار 7:30 (یعنی 7 ساعت و 30 دقیقه) را به عنوان `time` به تابع `convert` می‌دهیم و باید 7.5 (یعنی 7.5 ساعت) را برگرداند.

```
meal.py

def main():
    ...

def convert(time):
    ...

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

▼ نکته

- به یاد داشته باشید که string توابع مختلفی دارد. توضیحات بیشتر را در اینجا می‌توانید ببینید: docs.python.org/3/library/stdtypes.html#string-methods که از جمله‌ی آنها `split` است، که یک string را بر اساس مقداری که به آن ورودی داده اید به بخش‌های مختلف تقسیم می‌کند و یک لیستی از مقادیر می‌سازد و می‌توان این لیست را به یکباره به تعدادی از متغیر نسبت داد. به عنوان مثال، اگر `time` یک string مانند "7:30" باشد، آنگاه 7 را به `hours` و 30 را به `minutes` اختصاص خواهد داد.

```
hours, minutes = time.split(":")
```

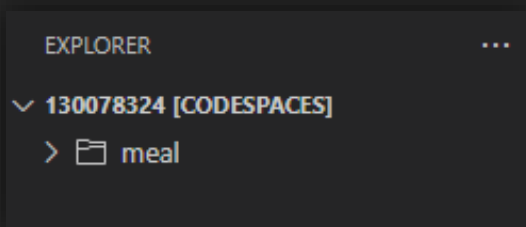
- توجه داشته باشید که 1 ساعت ، 60 دقیقه است.

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 meal در codespace ایجاد شود:

```
$ mkdir meal
```

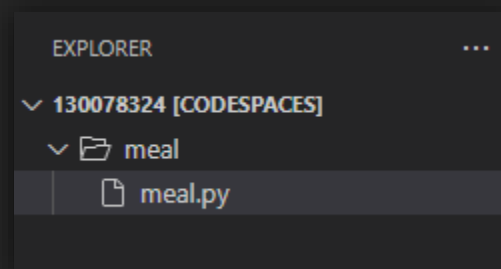


سپس `cd meal` را اجرا کنید تا به پوشه‌ی `meal` منتقل شوید. اکنون شما باید آدرس `meal/ $` را در Terminal مشاهده کنید:

```
$ cd meal  
meal/ $
```

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

```
meal/ $ code meal.py
```



اکنون می‌توانید در این فایل برنامه‌ی خود را بنویسید.

اگر برای یک چالش آماده هستید بیایید به صورت اختیاری، پشتیبانی قالب زمانی 12 ساعته را اضافه کنید و به کاربر اجازه دهید زمان‌ها را در این قالب‌ها نیز وارد کند:

: ## a.m.

9

: ## a.m.

: ## p.m.

9

: ## p.m.

اکنون می‌خواهیم برنامه‌ی نوشته شده‌ی خود را تست کنیم:

- دستور `python meal.py` را اجرا کنید. 7:00 را تایپ کنید و کلید Enter را فشار دهید. خروجی شما برابر می‌شود با `breakfast time`.
- دستور `python meal.py` را اجرا کنید. 7:30 را تایپ کنید و کلید Enter را فشار دهید. خروجی شما برابر می‌شود با `breakfast time`.
- دستور `python meal.py` را اجرا کنید. 12:42 را تایپ کنید و کلید Enter را فشار دهید. خروجی شما برابر می‌شود با `lunch time`.
- دستور `python meal.py` را اجرا کنید. 18:32 را تایپ کنید و کلید Enter را فشار دهید. خروجی شما برابر می‌شود با `dinner time`.
- دستور `python meal.py` را اجرا کنید. 11:11 را تایپ کنید و کلید Enter را فشار دهید. برنامه نباید خروجی بدهد!

❖ 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/meal
```

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/meal
```

CS50x Iran

Harvard's Computer Science 50x Iran

