



# Seasons of Love

Problem set



Five hundred twenty-five thousand, six hundred minutes  
Five hundred twenty-five thousand moments so dear  
Five hundred twenty-five thousand, six hundred minutes  
How do you measure, measure a year?

– “Seasons of Love,” Rent

ترجمه شعر:

پنج هزار و دویست و پنجاه و شش دقیقه  
پنج هزار و دویست و پنجاه لحظه‌ی بسیار عزیز  
پنج هزار و دویست و پنجاه و شش دقیقه  
چگونه یک سال رو محاسبه می‌کنی؟

Rent اثری از Season of Love

با فرض این که یک سال 356 روز است، یک سال برابر با  $365 \times 24 \times 60 = 525600$  دقیقه می‌شود. (چرا که یک روز 24 ساعت و یک ساعت 60 دقیقه است). اما اگر بخواهیم چندین سال را بر حسب دقیقه حساب کنیم باید آن را چگونه حساب کنیم؟ خب، این به تعداد سال کبیسه‌ای که یک روز بیشتر از سال های عادی (366 روز) است بستگی دارد. بنابراین این سال های کبیسه  $1 \times 24 \times 60 = 1440$  دقیقه بیشتر از سال های دیگر هستند. حتی تعداد دقیقه‌هایی که از لحظه متولد شدن شما می‌گذرد نیز به تعداد این سال های کبیسه بستگی دارد. یک الگوریتم برای این سال های کبیسه وجود دارد، اما بیاید وقتمان را برای یادگیری این الگوریتم هدر ندهیم و از یک کتابخانه استفاده کنیم. خوشبختانه، پایتون برای این کار یک ماژول به اسم `datetime` دارد که در آن کلاس `date` وجود دارد و می‌تواند به ما کمک کند. با مراجعه به این لینک می‌توانید اطلاعات بیشتری کسب کنید.

در فایلی به نام `seasons.py`، برنامه‌ای بنویسید که از کاربر روز تولدش را به فرم `YYYY-MM-DD` دریافت کند و سنش بر حسب دقیقه را به حروف (انگلیسی) به جای عدد چاپ کند (در صورتی که سنش اعشاری شد، آن را به سمت بالا گرد کند). سن باید دقیقاً مثل شعری که در ابتدا گفته شد باشد و هیچ `and`ی بین اعداد وجود نداشته باشد. شاید کاربر نداند که در چه ساعتی به دنیا آمده است. شما برای سادگی ساعت تولد اون را نیمه شب آن روز یعنی `(00:00:00)` در نظر بگیرید. همچنین زمان فعلی را نیز نیمه شب در نظر بگیرید. در واقع حتی اگر کاربر برنامه را ظهر اجرا کرد، همچنان شما آن را نیمه شب آن روز در نظر بگیرید. برای به دست آوردن تاریخ امروز از `datetime.date.today` استفاده کنید.

برنامه خود را به صورت زیر علاوه بر تابع `main`، با یک یا چند تابع دیگر پی‌کربندی کنید.

```
from datetime import date

def main():

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

شما مجاز هستید که از کتابخانه‌های دیگر نیز استفاده کنید. اگر کاربر تاریخ را به صورت YYYY-MM-DD وارد نکرد، می‌توانید با دستور `sys.exit()` از برنامه خارج شوید. مطمئن شوید که برنامه شما هیچ `exception`ی را اعلام نمی‌کند.

قبل یا بعد از پیاده سازی برنامه در فایل `seasons.py`، در فایل دیگری به نام `test_seasons.py`، یک یا چند تابع قرار دهید تا پیاده سازی برنامه تان را در حضور توابعی دیگر در کنار `main` در فایل `seasons.py` تست کنید. اسم هر کدام از آن توابع بایستی با `test_` شروع شود تا بتوانید تست های خود را با دستور زیر اجرا کنید:

```
pytest test_seasons.py
```

▼ نکته

- به یاد داشته باشید که date کلاس شامل یک سری متد کاربردی است. خصوصاً طبق این لینک این کلاس یک متد `__sub__` دارد که از – استفاده می‌کند و این استفاده بدین گونه است که یک `date object` را از دیگری کم می‌کند و یک timedelta برمی‌گرداند که خودش با چندین ویژگی به صورت (read-only) همراه است.
- به یاد داشته باشید که ماژول inflect با یک سری متد همراه است. می‌توانید آن را با دستور زیر نصب کنید:

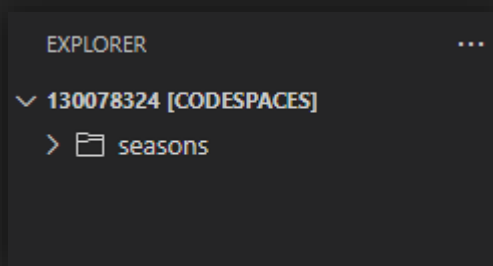
```
pip install inflect
```

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 seasons در codespace ایجاد شود:

```
$ mkdir seasons
```

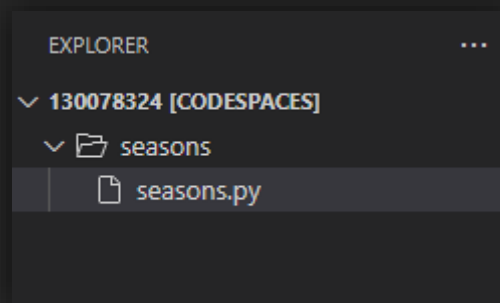


سپس `cd seasons` را اجرا کنید تا به پوشه‌ی `seasons` منتقل شوید. اکنون شما باید آدرس `$ seasons/` را در Terminal مشاهده کنید:

```
$ cd seasons  
seasons/ $
```

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

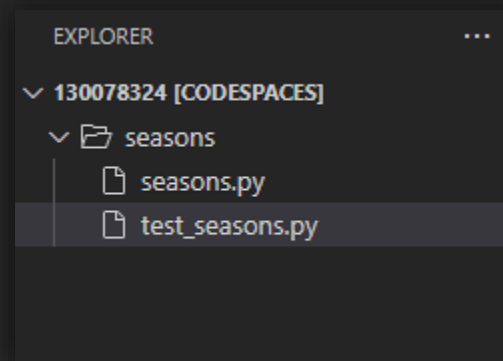
```
seasons/ $ code seasons.py
```





یادتان باشد که دستور `code test_seasons.py` را نیز اجرا کنید تا فایلی با نام `test_seasons.py` ایجاد کنید:

```
seasons/ $ code test_seasons.py
```



اکنون می‌توانید در این فایل برنامه‌ی خود را بنویسید.

چگونه فایل `seasons.py` را تست کنیم:

- دستور `python seasons.py` را اجرا کنید. مطمئن شوید برنامه از شما تاریخ تولد (`birthdate`) را می‌خواهد. دقیقاً تاریخ یک سال پیش را به فرمت گفته شده وارد کنید و سپس دکمه Enter را بزنید. برنامه شما باید Five hundred twenty-five thousand, six hundred minutes را به عنوان خروجی برگرداند.
- دستور `python seasons.py` را اجرا کنید. تاریخ دقیقاً دو سال پیش را به فرمت گفته شده وارد کنید و سپس دکمه Enter را بزنید. برنامه شما باید One million, fifty-one thousand, two hundred minutes را به عنوان خروجی برگرداند.
- دستور `python seasons.py` را اجرا کنید. تاریخ دلخواه خود را وارد کنید، اما این بار فرمت اشتباهی را وارد کنید و سپس دکمه Enter را بزنید. برنامه شما باید با استفاده از `sys.exit()` خارج شود و بدون اینکه Exception‌ای را اعلام کند.

چگونه فایل `test_seasons.py` را تست کنیم:

- برای تست کردن فایل‌های `test` خود دستور `pytest test_seasons.py` را اجرا کنید. سعی کنید از نسخه درست و نادرست `seasons.py` استفاده کنید تا مطمئن شوید تست‌هایتان تا جای ممکن ارورها را تشخیص می‌دهد.
- مطمئن شوید که نسخه درستی از `seasons.py` را دارید. تست‌هایتان را با دستور `pytest test_seasons.py` اجرا کنید. `pytest` بایستی نشان دهد که تمامی تست‌های شما تایید شده است.
- یکی از توابعی موجود در فایل `seasons.py` را که در فایل `test_seasons.py` قرار داده‌اید را تغییر دهید. یکی از توابع ممکن است برای مثال موفق نشود در صورت نیاز `ValueError` اعلام کند. تست‌هایتان را با `pytest test_seasons.py` اجرا کنید. `pytest` باید نشان دهد که حداقل یکی از تست‌های شما ناموفق بوده است.
- به تغییر دادن رفتار `seasons.py` با ساختن نسخه‌های نادرست پیاده‌سازی‌تان ادامه دهید. تست‌هایتان را با دستور `pytest test_seasons.py` اجرا کنید. آیا انتظار دارید که تست‌هایتان ناموفق باشند؟ آیا ناموفق هستند؟

❖ 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/seasons
```

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/seasons
```

# CS50x Iran

Harvard's Computer Science 50x Iran

