



Outdated

Problem set



در ایالات متحده، تاریخ‌ها معمولاً به ترتیب ماه-روز-سال (MM/DD/YYYY) قالب‌بندی می‌شوند، که به‌عنوان ترتیب میانی شناخته می‌شود. مسلماً طراحی بدی است زیرا تاریخ‌ها در این قالب را نمی‌توان به راحتی مرتب کرد. سال تاریخ به جای این که در اول بیاید، در آخر آمده است. به عنوان مثال 2/2/1800، 3/3/1900، و 1/1/2000 را به ترتیب زمانی در هر برنامه ای (به عنوان مثال excel) مرتب کنید. تاریخ‌ها در آن قالب نیز مبهم هستند.

هاروارد در 8 سپتامبر 1636 تاسیس شد اما 9/8/1636 را می‌توان به 9 اوت 1636 نیز تعبیر کرد!

خوشبختانه، کامپیوترها تمایل دارند از قالب ISO 8601 استفاده کنند که یک استاندارد بین‌المللی است و قالب آن باید به ترتیب سال-ماه-روز (YYYY-MM-DD) فرمت شوند. بدون توجه به کشور، قالب بندی سال‌ها با چهار رقم، ماه‌ها با دو رقم، و روزها هم با دو رقم است. برای مثال در صورتی که روز برابر 3 باشد به دلیل اینکه بر طبق این استاندارد روز دو رقم است باید در جای خالی عدد صفر قرار بگیرد یعنی 03.

در فایلی به نام `outdated.py`، برنامه‌ای بنویسید که از کاربر می‌خواهد تاریخ بدون عدد صفر، به ترتیب ماه-روز-سال، با قالب‌بندی‌هایی مانند `9/8/1636` یا `8 سپتامبر 1636` وارد کند. ماه متنی می‌تواند یکی از مقادیر `list` زیر باشد:

```
[  
    "January",  
    "February",  
    "March",  
    "April",  
    "May",  
    "June",  
    "July",  
    "August",  
    "September",  
    "October",  
    "November",  
    "December"  
]
```

سپس همان تاریخ را با فرمت `YYYY-MM-DD` خروجی بگیرید. اگر ورودی کاربر در هر یک از قالب‌ها تاریخ معتبری ندارد، دوباره از کاربر درخواست کنید. فرض کنید هر ماه بیشتر از 31 روز نیست. نیازی به تأیید اعتبار یک ماه 28، 29، 30 یا 31 روزه نیست.

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
- به یاد داشته باشید که list توابع مختلفی دارد به ویژه index. توضیحات بیشتر را در اینجا می‌توانید ببینید:
docs.python.org/3/tutorial/datastructures.html#more-on-lists
- دقت داشته باشید شما می‌توانید در ابتدای int های تک رقمی به روش زیر یک صفر قرار دهید:

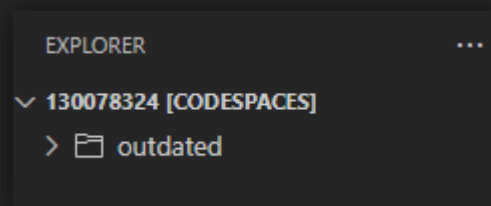
```
print(f"{n:02}")
```

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سپس کد زیر را اجرا کنید تا یک پوشه به نام outdated در codespace ایجاد شود:

```
$ mkdir outdated
```

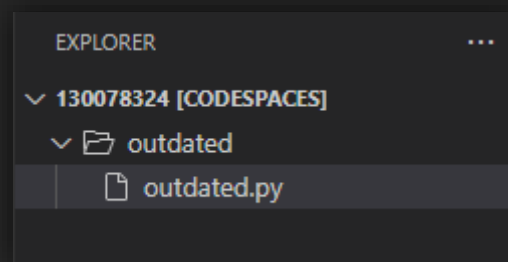


سپس `cd outdated` را اجرا کنید تا به پوشه‌ی outdated منتقل شوید. اکنون شما باید آدرس `$ outdated/` را در Terminal مشاهده کنید:

```
$ cd outdated  
outdated/ $
```

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

```
outdated/ $ code outdated.py
```



اکنون می‌توانید در این فایل برنامه‌ی خود را بنویسید.

اکنون می‌خواهیم برنامه‌ی نوشته شده‌ی خود را تست کنیم:

- دستور `python outdated.py` را اجرا کنید. 9/8/1636 را تایپ کنید و کلید Enter را فشار دهید. خروجی شما برابر می‌شود با: 1636-09-08
- دستور `python outdated.py` را اجرا کنید. September 8, 1636 را تایپ کنید و کلید Enter را فشار دهید. خروجی شما برابر می‌شود با: 1636-09-08
- دستور `python outdated.py` را اجرا کنید. 23/6/1912 را تایپ کنید و کلید Enter را فشار دهید. برنامه شما باید از کاربر دوباره درخواست ورودی کند.
- دستور `python outdated.py` را اجرا کنید. December 80, 1980 را تایپ کنید و کلید Enter را فشار دهید. برنامه شما باید از کاربر دوباره درخواست ورودی کند.

❖ 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/outdated
```

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/outdated
```

CS50x Iran

Harvard's Computer Science 50x Iran

