



Scourgify

Problem set



"تانکس در حالی که درب صندوق عقب را بست، گفت: «اه خب، حداقل همه چیز داخل است. میتوانیم کمی نظافت هم انجام بدهیم.» چوب جادویش را به سمت قفس هدویگ گرفت. "اسکورجیفای" چند پر و مدفوع ناپدید شدند."

- متن از کتاب هری پاتر و محفل ققنوس

داده‌ها نیز اغلب باید «پاک‌سازی» شوند، مثلاً با فرمت کردن مجدد (reformatting)، مقادیر در یک شکل منسجم‌تر و مناسب‌تر قرار می‌گیرند. به عنوان مثال، این فایل CSV دانش آموزان هاگوارتز، [before.csv](#) را در نظر بگیرید. (لیست زیر بخشی از این فایل می باشد)

```
name, house
"Abbott, Hannah",Hufflepuff
"Bell, Katie",Gryffindor
"Bones, Susan",Hufflepuff
"Boot, Terry",Ravenclaw
"Brown, Lavender",Gryffindor
"Bulstrode, Millicent",Slytherin
"Chang, Cho",Ravenclaw
"Clearwater,
Penelope",Ravenclaw
"Crabbe, Vincent",Slytherin
"Creevey, Colin",Gryffindor
"Creevey, Dennis",Gryffindor
"Diggory, Cedric",Hufflepuff
"Edgecombe,
Marietta",Ravenclaw
```

.....

منبع:

[https://en.wikipedia.org/wiki/
List_of_Harry_Potter_characters](https://en.wikipedia.org/wiki/List_of_Harry_Potter_characters)

با اینکه هر «ردیف» در فایل دارای سه ستون است (نام خانوادگی، نام و گروه)، دو مقدار اول در یک «ستون» (نام) ترکیب می‌شوند و در دبل کوتیشن (") قرار دارند. نام خانوادگی و نام با کاما و space از یکدیگر جدا شده اند. اگر هاگوارتز بخواهد برای هر دانش آموز از طریق mail merge یک فرم نامه بفرستد، این روش خوب نیست، زیرا شروع نامه با میبینید که عبارت پایین عجیب است :

Dear Potter, Harry,

و باید به شکل زیر باشد! :

Dear Harry,

در یک فایل به نام `scourgify.py` برنامه زیر را پیاده‌سازی کنید:

▪ از کاربر انتظار داریم دو آرگومان `command-line` وارد کند:

1. نام یک فایل `CSV` موجود که می‌خواهیم آن را به عنوان ورودی بخوانیم که ستون های آن به ترتیب، `name` و `house` هستند.

2. نام یک `CSV` جدید برای نوشتن خروجی در آن، که ستون های آن باید به ترتیب، `last, first` و `house` باشند.

▪ ورودی را به خروجی تبدیل کنید و هر ستون نام را به نام و نام خانوادگی تقسیم کنید. فرض کنید هر دانش آموز حتماً نام و هم نام خانوادگی دارد.

اگر کاربر دقیقاً دو آرگومان `command-line` ارائه نکرد، یا اگر اولین آرگومان قابل خواندن نبود، برنامه باید از طریق `sys.exit` با یک پیغام خطا خارج شود.

▼ نکته

- توجه داشته باشید که ماژول `csv` چندین متد دارد ([اطلاعات بیشتر](#)) که از میان آنها می توان به `DictReader` و `DictWriter`، اشاره کرد.
- توجه داشته باشید که با استفاده از `DictWriter` می توانید `fieldnames` خود را با استفاده از `writeheader`، بدون هیچ آرگومانی، در فایل بنویسد.

```
$ python scourgify.py
Too few command-line arguments
$ python scourgify.py 1.csv
Too few command-line arguments
$ python scourgify.py 1.csv 2.csv 3.csv
Too many command-line arguments
$ python scourgify.py 1.csv 2.csv
Could not read 1.csv
$ python scourgify.py before.csv
after.csv
$
```

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 **scourgify** در **codespace** ایجاد شود:

```
$ mkdir scourgify
```

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

```
$ cd scourgify  
scourgify/ $
```


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

```
scourgify/ $ code scourgify.py
```

برای دانلود فایل `before.csv` دستور زیر را اجرا کنید:

```
wget https://cs50.harvard.edu/python/2022/psets/6/scourgify/before.csv
```

در این قسمت توضیح داده شده که چطور خودتون به عنوان یک کاربر برنامه نوشته شده را تست کنید

- برنامه خود را با `python scourgify.py` اجرا کنید. برنامه شما باید با استفاده از `sys.exit` خارج شود و یک پیغام خطا نشان دهد:

```
Too few command-line arguments
```

- فایل های خالی `1.csv`، `2.csv` و `3.csv` را ایجاد کنید. برنامه خود را با
- `python scourgify.py 1.csv 2.csv 3.csv` اجرا کنید. برنامه شما باید خروجی زیر را نشان دهد:

```
Too many command-line arguments
```

- برنامه خود را با `python scourgify.py invalid_file.csv output.csv` اجرا کنید. با فرض اینکه `invalid_file.csv` وجود ندارد، برنامه شما باید با استفاده از `sys.exit` خارج شود و یک پیام خطا نشان دهد:

```
Could not read invalid_file.csv
```

- برنامه خود را با `python scourgify.py before.csv after.csv` اجرا کنید. با فرض وجود فایل `before.csv`، برنامه شما باید یک فایل جدید به نام `after.csv` ایجاد کند که ستون‌های آن باید به ترتیب، `Last,first` و `house` باشند.

❖ 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/scourgify
```

لبخند های سبز :) به این معنی هستند که برنامه‌ی شما در تست **قبول شده** و اخم های قرمز :) یعنی برنامه‌ی شما **دارای ایراد** هست، همچنین پیام های زرد ! بدین معنی است که برای اینکه **check50** بتواند این بخش برنامه شما را بررسی کند باید ابتدا ارورهایی که با اخم های قرمز:) جلوتر آمده اند را برطرف کنید.

. برای دیدن جزییات بیشتر درباره این تست که توسط **check50** صورت گرفت، وارد لینکی که به شما خروجی داده است شوید.

با اجرا کردن صورت زیر در Terminal پاسخ خود را ارسال کنید.

```
$ submit50 cs50/problems/2022/python/scourgify
```

توجه کنید هرچند بار که لازم باشد میتوانید کد خود را اصلاح و ارسال کنید ، نمره آخرین پاسخ شما در نظر گرفته میشود .

CS50x Iran

Harvard's Computer Science 50x Iran

