



camelCase

Problem set



در برخی زبان های برنامه نویسی استفاده از camelCase (به کیس ترکیبی یا mixed case هم شناخته می شود) برای نام گذاری متغیرهایی که از چند کلمه تشکیل شده اند رایج است. این نام گذاری به این صورت است که حرف اول از کلمه اول به صورت حروف کوچک انگلیسی (a, b, ...) و حرف اول از کلمات بعدی به صورت حروف بزرگ (A, B, ...) نوشته می شود. فراموش نکنید که این کلمات به هم پیوسته و بدون فاصله تایپ می شوند. به عنوان مثال، با در نظر گرفتن این که یک متغیر برای نام کاربر ممکن است "name" نامیده شود، متغیری برای نام کوچک کاربر می تواند به صورت firstName، و متغیری برای نام کوچک ترجیحی کاربر (مثلاً نام مستعار) به صورت preferredFirstName نوشته شود.



در مقابل، پایتون استفاده snake case یا حالت ماری را برای تعریف نام متغیر توصیه می‌کند. در این روش تمام کلمات با حروف کوچک نوشته و با (-) یا آندرلاین، از هم جدا می‌شوند. برای مثال، همان متغیر ها که در اسلاید قبل گفتیم به ترتیب به صورت: `name`، `first_name` و `preferred_first_name` در پایتون نوشته می‌شوند.

در فایلی به نام `camel.py`، برنامه‌ای بنویسید که از کاربر نام یک متغیر را به صورت camel case درخواست کند و متناظر آن را به صورت snake case خروجی دهد. فرض کنید که ورودی کاربر واقعاً و همیشه camel case خواهد بود.

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

- رشته‌ها (str) تقریباً همانند List، یک دنباله محسوب می‌شوند. یعنی شما می‌توانید روی هر کاراکتر آن در Loop (حلقه‌ی تکرار) پیمایش کنید. برای مثال، اگر s یک رشته (string) باشد، شما می‌توانید با کدی مانند زیر، هر یک از کاراکترهای آن را یکی یکی چاپ کنید:

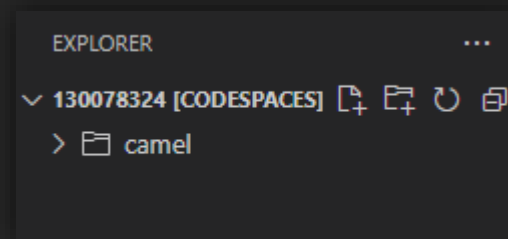
```
for c in s:  
    print(c, end="")
```

وارد code.cs50.io شوید، سپس بر روی پنجره‌ی Terminal کلیک کنید. توجه داشته باشید که دستورات پنجره‌ی Terminal شما باید به صورت زیر باشد:

```
$
```

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

```
$ mkdir camel
```

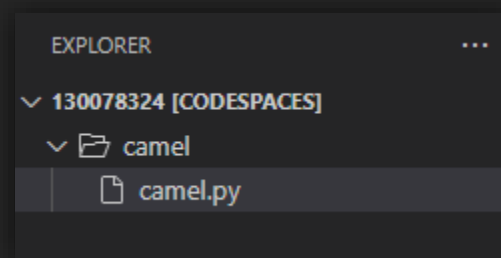


سپس `cd camel` را اجرا کنید تا به پوشه‌ی camel منتقل شوید. اکنون شما باید آدرس `$ camel/` را در Terminal مشاهده کنید:

```
$ cd camel  
camel/ $
```

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

```
camel/ $ code camel.py
```



اکنون می‌توانید در این فایل برنامه‌ی خود را بنویسید.

اکنون می‌خواهیم برنامه‌ی نوشته شده‌ی خود را تست کنیم:

- دستور `python camel.py` را اجرا کنید. `name` را تایپ کنید و کلید `Enter` را فشار دهید. خروجی شما برابر می‌شود با: `name`
- دستور `python camel.py` را اجرا کنید. `firstName` را تایپ کنید و کلید `Enter` را فشار دهید. خروجی شما برابر می‌شود با: `first_name`
- دستور `python camel.py` را اجرا کنید. `preferredFirstName` را تایپ کنید و کلید `Enter` را فشار دهید. خروجی شما برابر می‌شود با: `preferred_first_name`

❖ 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/camel
```

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/camel
```

CS50x Iran

Harvard's Computer Science 50x Iran

