



Vanity Plates

Problem set





در ماساچوست، محل دانشگاه هاروارد، این امکان وجود دارد که به جای حروف و اعداد تصادفی، خودتان نام پلاک خودروی خود را درخواست کنید. اما الزاماتی باید رعایت شوند، این الزامات، عبارتند از:

- همه پلاک ها باید حداقل با دو حرف شروع شوند.
- پلاک ها حداقل 6 کاراکتر (حروف یا اعداد) و حداقل 2 کاراکتر داشته باشند.
- اعداد را نمی‌توان در وسط یک پلاک استفاده کرد. آنها باید در پایان بیایند. به عنوان مثال، AAA222 یک پلاک قابل قبول است اما AAA22A قابل قبول نخواهد بود. همچنین اولین عدد استفاده شده نمی‌تواند 0 باشد.
- هیچ نقطه، فاصله یا علامت نقطه گذاری مجاز نیست.

در فایلی به اسم plates.py، برنامه‌ای بنویسید که از کاربر یک پلاک را به عنوان ورودی بگیرد و سپس در صورتی که همه الزامات را برآورده کند، Valid را به عنوان خروجی برگرداند و در غیر این صورت Invalid خروجی دهد. فرض کنید در ورودی کاربر، تمام حروف بزرگ باشند. ساختار برنامه خود را بر اساس موارد زیر طوری تنظیم کنید که اگر مقدار ورودی ۵ همه‌ی الزامات را برآورده کرد، تابع is_valid، True و اگر اینطور نبود False را برگرداند. همچنین می‌توانید توابع بیشتری را هم پیاده‌سازی کنید (مثلًاً برای هر نیاز یک تابع در نظر بگیرید).

plates.py

```
def main():
    plate = input("Plate: ")
    if is_valid(plate):
        print("Valid")
    else:
        print("Invalid")

def is_valid(s):
    ...
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

- 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:

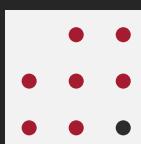
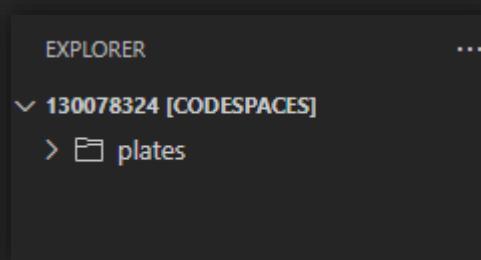
```
for c in s:  
    print(c, end="")
```

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

\$

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

```
$ mkdir plates
```

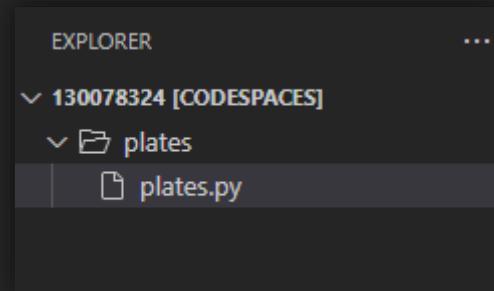


سپس `plates/ $` را اجرا کنید تا به پوشهی `plates` منتقل شوید. اکنون شما باید آدرس `plates/ $` را در Terminal مشاهده کنید:

```
$ cd plates  
plates/ $
```

حالا می توانید دستور `code plates.py` را اجرا کنید تا فایلی با نام `plates.py` ایجاد شود:

```
plates/ $ code plates.py
```



اکنون می توانید در این فایل برنامه‌ی خود را بنویسید.

اکنون می‌خواهیم برنامه‌ی نوشته شده‌ی خود را تست کنیم:

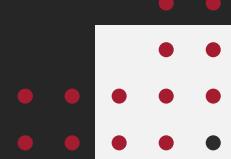
- دستور `python plates.py` را اجرا کنید. CS50 را تایپ کنید و کلید Enter را فشار دهید. خروجی شما برابر می‌شود با: Valid
- دستور `python plates.py` را اجرا کنید. CS05 را تایپ کنید و کلید Enter را فشار دهید. خروجی شما برابر می‌شود با: Invalid
- دستور `python plates.py` را اجرا کنید. CS50P را تایپ کنید و کلید Enter را فشار دهید. خروجی شما برابر می‌شود با: Invalid
- دستور `python plates.py` را اجرا کنید. PI3.14 را تایپ کنید و کلید Enter را فشار دهید. خروجی شما برابر می‌شود با: Invalid
- دستور `python plates.py` را اجرا کنید. H را تایپ کنید و کلید Enter را فشار دهید. خروجی شما برابر می‌شود با: Invalid
- دستور `python plates.py` را اجرا کنید. OUTATIME را تایپ کنید و کلید Enter را فشار دهید. خروجی شما برابر می‌شود با: Invalid



- ❖ 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/plates
```

لبخند های سبز به این معنی هستند که برنامه‌ی شما در تست قبول شده و اخمهای قرمز یعنی برنامه‌ی شما دارای ایراد هست. با مراجعه به URL check50 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/plates
```

CS50x Iran

Harvard's Computer Science 50x Iran

