



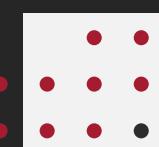
Response Validation

Problem set

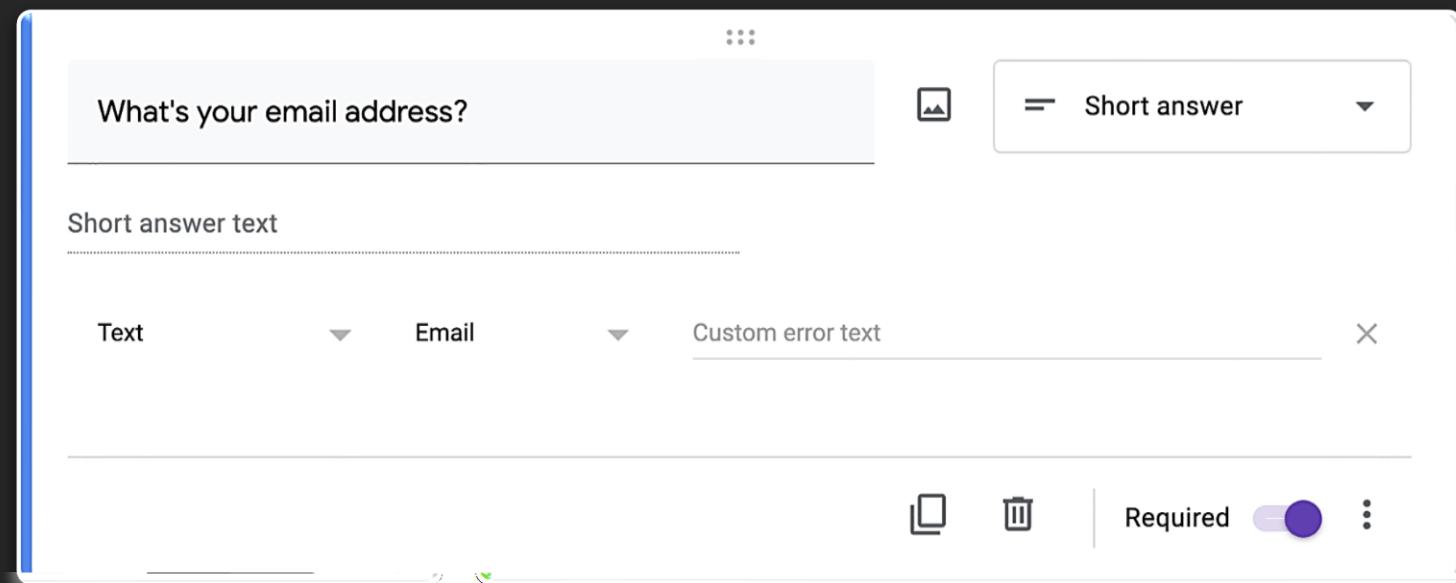


هنگام ایجاد یک Google Form شما می‌توانید از کاربر درخواست یک پاسخ کنید که متناسب با پرسش شما، پاسخ کاربر می‌تواند کوتاه یا بلند (پاراگراف) باشد، می‌توانید پاسخ وارد شده توسط کاربر را اعتبار سنجی (response validation) کنید و یا کاربر را ملزم به وارد کردن پاسخی کنید که با الگوی مورد نظرتان تطابق داشته باشد (regular expression). به عنوان مثال، می‌توانید کاربر را ملزم کنید که پاسخی را که به عنوان آدرس ایمیل خود وارد می‌کند، الگویی (regex) مانند تصویر مثال زیر داشته باشد:

```
^[a-zA-Z0-9.#!$%&'*+\=/=?^_`{|}~-]+@[a-zA-Z0-9](?:[a-zA-Z0-9-]{0,61}[a-zA-Z0-9])?(?:\.[a-zA-Z0-9](?:[a-zA-Z0-9-]{0,61}[a-zA-Z0-9]))?*$
```



یا به راحتی می‌توانید از ابزارهای گوگل‌فرم برای اعتبار سنجی یک آدرس ایمیل استفاده کنید، مانند تصویر زیر. همانند زمانی که از یک کتابخانه در کد خود استفاده کنید:



در یک فایل به نام `response.py`, با استفاده از `validators` یا `validator-collection` های PyPI، یک برنامه پیاده‌سازی کنید که از کاربر یک آدرس ایمیل دریافت کند و سپس در صورتی که ورودی یک آدرس ایمیل معتبر باشد (برای مثال شامل @ و پسوند .com. و ... باشد)، `Invalid` یا `Valid` را به عنوان خروجی پرینت کند.

- نباید از `re` استفاده کنید.
- همینطور نباید صحت وجود دامنه آدرس ایمیل را مورد بررسی قرار دهید.

نکته ▼

- توجه داشته باشید که می‌توانید کتابخانه `validator-collection` را با استفاده از دستور زیر نصب کنید.

```
pip install validator-collection
```

- برای اطلاعات بیشتر درباره این کتابخانه می‌توانید بر روی این [لینک کلیک](#) کنید و مطالب بیشتری را بیاموزید.
- همچنین می‌توانید پکیج `validators` را با استفاده از دستور زیر نصب کنید:

```
pip install validators
```

- برای اطلاعات بیشتر درباره این کتابخانه می‌توانید بر روی این [لینک کلیک](#) کنید و مطالب بیشتری را بیاموزید.

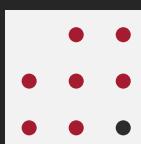
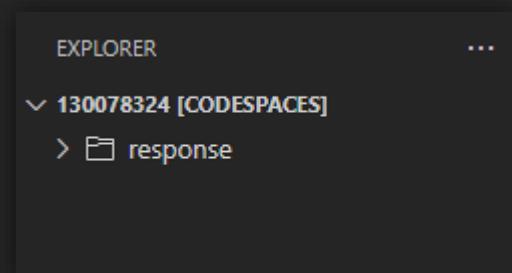


وارد code.cs50.io شوید، سپس بر روی پنجره Terminal کلیک کنید. توجه داشته باشید که دستور پنجره Terminal شما باید به صورت زیر باشد:

```
$
```

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

```
$ mkdir response
```

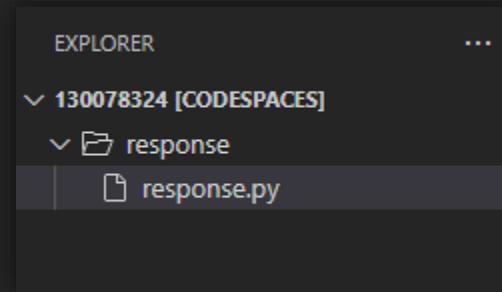


سپس `response` را اجرا کنید تا به پوشه‌ی `response` منتقل شوید. اکنون شما باید آدرس `$/response` را در Terminal مشاهده کنید:

```
$ cd response  
response/ $
```

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

```
response/ $ code response.py
```



اکنون می‌توانید در این فایل برنامه‌ی خود را بنویسید.

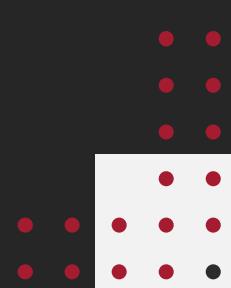
- دستور `python response.py` را اجرا کنید و دقت کنید که برنامه‌تان از شما بخواهد الزاماً یک ایمیل را وارد کنید، سپس `malan@harvard.edu` را تایپ کنید و سپس کلید `Enter` را فشار دهید. برنامه شما باید `Valid` را به عنوان خروجی نشان دهد.
- دستور `python response.py` را اجرا کنید و ایمیل خود را وارد کنید. سپس کلید `Enter` را فشار دهید. برنامه شما باید `Valid` را به عنوان خروجی نشان دهد.
- دستور `python response.py` را تایپ کنید و سپس کلید `Enter` را فشار دهید. برنامه شما باید `Invalid` را به عنوان خروجی نشان دهد.
- دستور `python response.py` را اجرا کنید . ایمیل خود را به صورت غلط وارد کنید. برای مثال، قبل از `.com`. نقطه‌ی اضافی بگذارید سپس کلید `Enter` را فشار دهید. برنامه شما باید `Invalid` را به عنوان خروجی نشان دهد.



- ❖ 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/response
```

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 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/response
```

CS50x Iran

Harvard's Computer Science 50x Iran

