



# NUMB3RS

Problem set



در قسمت بیست و سوم از فصل 5 سریال NUMB3RS (برای دیدن ویدیو روی لینک کلیک کنید)، یک آدرس IP فرضی به صورت 275.3.6.28 در صفحه نمایش داده می‌شود که در واقع یک آدرس معتبر IPv4 (یا IPv6) نیست. آدرس IPv4 یک شناسه عددی است که یک دستگاه برای برقراری ارتباط با اینترنت استفاده می‌کند. مانند یک آدرس پستی در دنیای واقعی، که نمایش آن معمولاً با فرمت dot-decimal notation بصورت #.#.#.# است. اما هر # باید یک عدد بین 0 تا 255، شامل باشد یعنی برا مثال، 275 در این محدوده نیست. این آدرس IP فقط در صورتی که NUMB3RS آن را تایید کند، یک آدرس معتبر است.

در فایلی به نام numb3rs.py، تابعی به نام `validate` را پیاده سازی کنید که یک آدرس IPv4 را که `str` است را ورودی بگیرد و سپس اگر آن ورودی یک آدرس IPv4 معتبر باشد `True` را به عنوان خروجی و در غیر این صورت `False` را برگرداند.

ساختار numb3rs.py باید به صورت زیر باشد، می‌توانید توابع بیشتری را هم پیاده‌سازی کنید اما نمی‌توانید کتابخانه‌های دیگری را `import` کنید.

```
import re
import sys

def main():
    print(validate(input("IPv4 Address: ")))

def validate(ip):

if __name__ == "__main__":
    main()
```

قبل یا بعد از اجرای `validate` در numb3rs.py، دو یا چند تابع را در فایلی به نام test\_numb3rs.py پیاده‌سازی کنید تا درست بودن کد شما را بررسی کنند. نام هر کدام از این توابع باید با test\_ شروع شود تا بتوانید آن را به صورت زیر اجرا کنید:

```
pytest test_numb3rs.py
```

## ▼ نکته

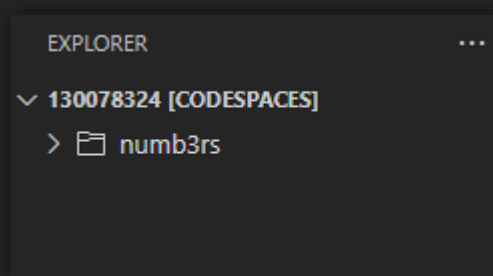
- یادتان باشد که ماژول `re` توابعی مانند `search` دارد که می‌توانید در این لینک اطلاعات بیشتری در مورد آن کسب کنید.
- می‌توانید در Regular Expressions (یا regex) از کاراکترهای ویژه‌ای استفاده کنید. این لینک را مطالعه کنید.
- چون کاراکتر `\` در regex می‌تواند با escape sequences (برای مثال `\` که اگر در متن استفاده شود، حرف بعد از آن به عنوان یک فرمان در نظر گرفته می‌شود. برای مثال `\n` که حرف `n` بعد از `\` به معنای new line بوده و هر جای متن استفاده شود، به معنای رفتن به خط بعد و شکستن متن است) اشتباه گرفته شوند، بهتر است از نماد رشته خام پایتون برای الگوهای regex استفاده کنید. مانند Format strings که با پیشوند `f` مشخص می‌شوند، رشته‌های خام نیز یک پیشوند `r` دارند. مثلاً به جای `"harvard\.edu"` از `r"harvard\.edu"` استفاده کنید.
- توجه داشته باشید که اگر `re.search` الگویی شامل `"grouping capturing"` (گروه‌های قابل استخراج) که با علائمی مثل پرانتز مشخص می‌شوند به عنوان ورودی دریافت کند، طبق (match object) عنصر مطابق با الگو را برمی‌گرداند، که در آن تطابق یافته‌ها (matches)، ایندکس 1 هستند. که می‌توانید به آنها به صورت جداگانه با `group` و یا به صورت جمعی با `groups` دسترسی داشته باشید.

وارد [code.cs50.io](https://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 numb3rs در codespace ایجاد شود:

```
$ mkdir numb3rs
```

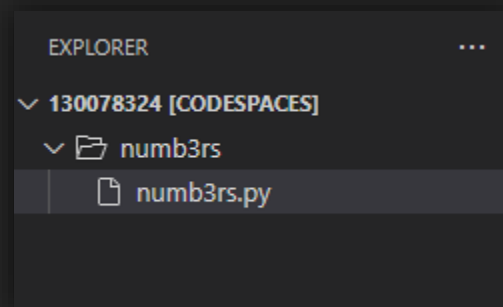


سپس `cd numb3rs` را اجرا کنید تا به پوشه‌ی `numb3rs` منتقل شوید. اکنون شما باید آدرس `$ numb3rs/` را در Terminal مشاهده کنید:

```
$ cd numb3rs
numb3rs/ $
```

حالا می‌توانید دستور `code numb3rs.py` را اجرا کنید تا فایل‌ی با نام `numb3rs.py` ایجاد شود:

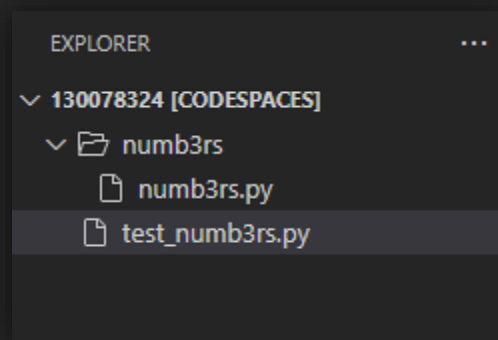
```
numb3rs/ $ code numb3rs.py
```



اکنون می‌توانید در این فایل برنامه‌ی خود را بنویسید.

یادتان باشد که دستور `code test_num3rs.py` را نیز اجرا کنید تا فایل با نام `test_num3rs.py` ایجاد کنید:

```
numb3rs/ $ code test_num3rs.py
```



اکنون می‌توانید در این فایل برنامه‌ی خود را بنویسید.

در اینجا نحوه آزمایش `numb3rs.py` به صورت دستی آمده است:

- برنامه خود را با `python numb3rs.py` اجرا کنید. مطمئن شوید که برنامه شما یک آدرس IPv4 را از شما ورودی می‌گیرد. `127.0.0.1` را تایپ کنید و سپس کلید `Enter` را بزنید. تابع `validate` شما باید `True` را برگرداند.
- برنامه خود را با `python numb3rs.py` اجرا کنید. `255.255.255.255` را تایپ کنید و سپس `Enter` را بزنید. تابع `validate` شما باید `True` را برگرداند.
- برنامه خود را با `python numb3rs.py` اجرا کنید. `512.512.512.512` را تایپ کنید و سپس `Enter` را بزنید. تابع `validate` شما باید `False` را برگرداند.
- برنامه خود را با `python numb3rs.py` اجرا کنید. `1.2.3.1000` را تایپ کنید و سپس `Enter` را بزنید. تابع `validate` شما باید `False` را برگرداند.
- برنامه خود را با `python numb3rs.py` اجرا کنید. `cat` را تایپ کنید و سپس `Enter` را بزنید. تابع `validate` شما باید `False` را برگرداند.



حال به نحوه تست با استفاده از `test_numb3rs.py` می‌پردازیم :

- برای آزمایش تست های خود، برنامه را با `pytest test_numb3rs.py` اجرا کنید. سعی کنید هم از نسخه‌های صحیح و هم نادرست `numb3rs.py` استفاده کنید تا درست بودن توابع `test_` را بهتر بررسی کنید و تمام حالات ممکن را در نظر بگیرید.
- برای شروع از نسخه درست برنامه ای که نوشتید استفاده کنید و `pytest test_numb3rs.py` را اجرا کنید در این صورت باید `pytest` تمام تست کیس‌هایی که دارید را قبول کرده باشد.
- حال با تغییر تابع `validate` به طوری که این تابع فقط بخش اول IP را چک می‌کند که درست هست یا نه، نسخه نادرستی از برنامه را داریم و در بعضی حالات جواب غلطی را خروجی می‌دهد و بعد از اجرای `pytest test_numb3rs.py`، `pytest` باید نشان دهد که حداقل یکی از تست‌های شما ناموفق بوده است چرا که تابع جدید `validate` هر ۴ قسمت IP را چک نمی‌کند.

❖ 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/numb3rs
```

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/numb3rs
```

# CS50x Iran

Harvard's Computer Science 50x Iran

