



دانشگاه صنعتی شریف

دانشکده مهندسی برق

آزمایشگاه برنامه نویسی پایتون

پاییز و زمستان ۱۴۰۰

## Iptables

۱. با استفاده از دستور زیر درخواست های پینگ را در guest os مسدود کنید:

```
iptables -A OUTPUT -s 0/0 -d 216.58.208.78 -j DROP
```

۲. اکنون سعی کنید google.com را پینگ کنید. چه نتیجه ای مشاهده کردید؟

۳. دوباره از iptables -L استفاده کنید ، خروجی چیست؟

۴. از iptables -F برای پاک کردن استفاده کنید (حذف همه قوانین)

۵. سعی کنید با استفاده از دستور زیر درخواست های پینگ را مسدود کنیم:

```
iptables -A INPUT -s 216.58.208.78 -j DROP
```

۶. اکنون سعی کنید دوباره google.com را پینگ کنید. چه مشاهده کردید؟

۷. دوباره از iptables -L استفاده کنید ، خروجی چیست؟

۸. دستور مسدود کردن subnet زیر چیست؟

```
192.168.2.x
```

۹. عملکرد rule های زیر را توضیح دهید

```
iptables -A INPUT -i lo -p all -j ACCEPT
```

```
iptables -A INPUT -p all -s localhost -i eth0 -j DROP
```

```
iptables -A INPUT -s 0/0 -i eth0 -d 192.168.1.1 -p TCP -j ACCEPT
```

```
iptables -A FORWARD -s 0/0 -i eth0 -d 192.168.1.58 -o eth1 -p TCP\  
--sport 1024: 65535 --dport 80 -j ACCEPT
```

۱۰. قانونی ایجاد کنید که SSH را محدود کند و فقط به آدرس های IP خاصی اجازه اتصال دهد.

۱۱. قوانینی ایجاد کنید که هرگونه مشاهده صفحات HTTP و HTTPS را در ماشین مجازی مسدود کند

۱۲. قوانینی ایجاد کنید تا میزان ترافیک ICMP(Ping) را به ۵ بسته در ثانیه محدود کنید

۱۳. فقط اتصالات SSH ورودی را مجاز بگذارید(خروجی محدود شود)

۱۴. کلید ترافیک ورودی TCP را مسدود کنید، حال کلید ترافیک ورودی UDP را مسدود کنید.

۱۵. فرض کنید در حال اجرای یک وب سرور هستید که می خواهید با iptables از آن محافظت کنید. در

مورد نحوه انجام این کار با جزئیات توضیح دهید.

## Python OOP

۱-۱ کلاسی با نام Set برای تعریف مجموعه ها پیاده سازی کنید. هر نمونه از این کلاس، باید شامل تعدادی عضو باشد (برای سادگی فرض کنید اعضا می توانند از نوع عدد یا رشته باشند) و توابع و عملگرهای زیر روی آن تعریف شوند. دقت کنید برای نگهداری اعضای مجموعه باید از لیست استفاده کنید. عملکرد کلاسی که تعریف کرده اید را بررسی نمایید.

عملکرد	تابع یا عملگر
یک مجموعه ی تهی ایجاد می کند.	"a = Set()"
در صورتی که x عضو a نباشد، آن را به a اضافه می کند.	"a.add(x)"
در صورتی که x عضو a باشد، آن را از a حذف می کند.	"a.delete(x)"
یک مجموعه که به ترتیب اجتماع و اشتراک a و b است ایجاد می کند.	"a & b" و "a   b"
اگر دو مجموعه یکسان باشند True و اگر یکسان نباشند False بر می گرداند.	"a == b"

## Python Statistic

### Pandas ۱

Pandas کتابخانه ای متن باز برای پایتون است که ساختارهای داده ای با کارایی بالا برای تجزیه و تحلیل داده ها را ممکن می سازد. این ساختارهای داده ای قابلیت تمیز کردن داده های خام و ابزارهایی برای پر کردن داده های از دست رفته در اختیار کاربر قرار می دهد. سه نوع داده ساختار در Pandas قابل تعریف می باشد. Series برای تعریف داده های یک بعدی ، نوع دوم که پر استفاده ترین است Dataframe می باشد که برای تعریف داده های دوبعدی می باشد. Panel نوع سوم می باشد و برای تعریف داده های سه بعدی و بالاتر استفاده می شود. در این آزمایش تمرکز روی یادگیری Dataframe و استفاده آن برای خواندن و ایجاد تغییر در یک دیتاست در فایل CSV می باشد.

۱. ابتدا با دستور `import pandas as pd` پکیج Pandas را وارد کنید. لیست `indexes` را از اعداد یک تا ۶ بسازید. متغیر `columns` را نیز به همین ترتیب از حروف A تا D بسازید. سپس متغیر `df` را که یک Dataframe با ۴ ستون و ۶ سطر است از با استفاده از اعداد تصادفی بسازید. در حین ساختن این دیتافریم `columns` را برابر متغیر `columns` و `index` را برابر متغیر `indexes` قرار دهید. حال با دستور `print` این متغیر را مشاهده نمایید.

۲. با ویژگی‌های `head` و `tail` دوسطر اول و آخر دیتافریم را نمایش دهید.
۳. دستورهای `df.values` و `df.columns` و `df.index` و `df.describe` را چاپ کنید و نتایج و کاربرد هر دستور را تحلیل کنید.
۴. با دستور مناسب این دیتا فریم را بر اساس ستون `B` و به ترتیب نزولی و در صورتی که مقادیر در این ستون مساوی بودند بر اساس صعودی و نسبت به ستون `C` مرتب کنید. به صورتی مرتب سازی را انجام دهید که همزمان در متغیر `df` ذخیره سازی هم صورت بگیرد.
۵. متغیر `S` را با استفاده از `pd.Series` و از اعداد ۱ تا ۶ بسازید. ستون `F` را با متغیر `S` به `df` اضافه کنید.
۶. با استفاده از `df.at` یا `df.iat` سطر سوم و پنجم ستون `F` را برابر `np.nan` قرار دهید. حال با استفاده از `df.loc` یا `df.iloc` سه سطر اول ستون های `A` و `F` را نمایش دهید.
۷. متغیر `df1` را برابر متغیر `df` بدون خانه های `NaN` قرار دهید. (برای این کار میتوانید از `df.dropna` استفاده کنید).
۸. متغیر `df2` را برابر متغیر `df` با پر کردن خانه های `NaN` با میانگین ستون آخر قرار دهید. (برای این کار میتوانید از `df.fillna` استفاده کنید).

## ۲ matplotlib

- پکیج `matplotlib` ابزاری بسیار قوی برای ترسیم انواع نمودارها و گراف ها در پایتون است. در این قسمت با استفاده از این پکیج توزیع داده‌های یک `Dataframe` را ترسیم می‌کنیم.
۱. فایل `"train.csv"` را خوانده و در یک `Dataframe` ذخیره کنید. هیستوگرام داده‌ها را بر اساس `ApplicantIncome` و در ۵۰ بین (`bin`) رسم کنید. نمودار داده‌ها را این بار با نمودار `boxplot` در حالتی که ستون های آن بر اساس `ApplicantIncome` و محور افقی آن بر اساس `Education` می‌باشد رسم کنید.
  ۲. با استفاده از `plt.figure` یک `figure` ایجاد کنید. حال یک `subplot` اضافه کرده و در آن تعداد `Applicant` ها را بر اساس `Credit_History` نمایش دهید. (محور `X` را `Credit_History` و محور `Y` را تعداد `Applicant` ها قرار دهید). حال یک `subplot` دیگر اضافه کنید و در آن احتمال دریافت وام (`Loan_Status`) را بر اساس `Credit_History` رسم کنید. (ابتدا باید `Loan_Status` را از حالت `numerical` به `categorical` تبدیل کنید).

## ۳ Numpy & Scipy

پکیج های `numpy`<sup>۱</sup> و `scipy`<sup>۲</sup> برای محاسبات عددی و کاربردهای ریاضیاتی استفاده می گردند و قابلیت هایی همچون محاسبات آماری، حل مسائل بهینه سازی، جبرخطی، انتگرال گیری، پردازش سیگنال و ... را فراهم می آورند. همچنین پکیج `matplotlib`<sup>۳</sup> ابزاری بسیار قوی برای ترسیم انواع نمودارها و گراف ها در پایتون است.

۱. آرایه های `numpy` نسبت به لیست های پایتون از نظر حافظه و سرعت عملکرد بهتری دارند و قابلیت های کاربردی بیشتری نیز در اختیار می گذارند. یک آرایه ۶ عضوی با کمک تابع `arange` ایجاد نمایید و با کمک تابع `reshape` آن را به صورت  $3 \times 2$  در آورید. سپس یک آرایه ی دیگر با اندازه ی  $3 \times 3$  ایجاد نمایید و با کمک `stack` دو آرایه را زیر هم قرار دهید تا به یک آرایه ی  $3 \times 5$  تبدیل شوند. با استفاده از `ravel` این آرایه را به یک آرایه ی مسطح تبدیل کنید و با استفاده از `argmax` اندیس بزرگترین عضو آن را بیابید.

۲. ماتریس هایی به صورت زیر تشکیل دهید و با کمک `linalg`<sup>۴</sup> عملیاتی که ذکر شده را روی آنها انجام دهید.

ماتریس	عملیات
<code>A = np.mat([[۳,۱,۴],[۱,۵,۹],[۲,۶,۵]])</code> <code>b = np.mat([۲,۴,-۱])</code>	$A^T, A^{-1},  A $ <code>eig(A), svd(A), qr(A)</code> Solve: $Ax=bT$

۳. ابتدا یک بردار تصادفی  $t$  با  $1000$  عضو ایجاد کنید و آن را به بازه  $-10\pi \leq t \leq 10\pi$  تصویر کنید. سپس بردار ایجاد شده را مرتب نمایید و به ازای نقاط این بردار تابع  $y = \text{sinc}(t)$  را محاسبه نمایید. در مرحله ی بعد تبدیل فوریه  $z = \text{fft}(y)$  را محاسبه کنید. در نهایت با کمک `pyplot` گراف تابع اصلی و اندازه ی تبدیل فوریه آن را در یک شکل و در قالب دو `subplot` زیر هم رسم نمایید. برچسب گذاری نمودار را به نحو صحیح انجام دهید و گراف حاصل را به صورت فایل `'figure.png'` ذخیره نمایید.

<sup>۱</sup> <http://www.numpy.org>

<sup>۲</sup> <https://www.scipy.org>

<sup>۳</sup> <https://matplotlib.org>

<sup>۴</sup> <https://docs.scipy.org/doc/scipy/reference/linalg.html>

## Python Network

### ۱ آشنایی با برنامه نویسی شبکه

socket ها از اجزا پایه ای در شبکه می باشند که برقراری ارتباط بین دو نقطه را ممکن می سازند. در این بخش با چند مثال ساده با مدل ارتباط client-server آشنا می شویم.

۱. با استفاده از کتابخانه `socket`<sup>5</sup> یک برنامه ی `server` و یک برنامه ی `client`، هر دو بر روی `local host`، برای ارتباط `TCP` روی پورت دلخواه بنویسید. `server` منتظر `connection` می ماند، و هنگام اتصال پیغام خوش آمد گویی را ارسال می نماید. `client` به `server` متصل می شود و پیغامی که دریافت می کند را چاپ می نماید. برنامه ها را به ترتیب اجرا کنید و نتیجه را مشاهده کنید.

۲. برنامه ی `server` و `client` ی برای انتقال فایل بر روی `local host` بنویسید. یک فایل `'data.txt'` که در خط اول نام و در خط دوم شماره دانشجوییتان را در بر دارد ایجاد کرده و در محل کد `server` قرار دهید. روال کار به این صورت است که وقتی `client` به `server` متصل شد، `server` فایل را خوانده و برای `client` ارسال می کند و `client` نیز اطلاعات دریافتی را با همان فرمت در فایل با همان نام ذخیره می نماید.

۳. `thread` ها برای اجرای هم روند (و نه همزمان) برنامه ها مورد استفاده قرار می گیرند. با کمک ماژول `thread`، با ایجاد کردن یک `thread` برای سرویس دهی به هر `client`، مثال قبل را به نحوی گسترش دهید تا `server` قابلیت سرویس دهی به هر تعداد `client` به طور همزمان را داشته باشد. برای این منظور، دو فایل `'data1.txt'` و `'data2.txt'` با محتوای متفاوت را در محل `server` قرار دهید. سناریو به این شکل است که هر `client` هنگامی که به `server` متصل شد نام فایل را از کاربر سوال کرده و به `server` اعلام می کند. در صورتی که فایل مورد نظر در محل `server` وجود داشت، برای `client` ارسال می شود و در صورت عدم وجود فایل، این مساله به `client` اطلاع رسانی می شود تا به کاربر اعلام گردد.

۴. (امتیازی) کتابخانه ی `threading`<sup>6</sup> رابط سطح بالا برای تسهیل استفاده از ماژول `thread` می باشد. همین مثال را یک بار دیگر با این کتابخانه پیاده سازی کنید.

<sup>5</sup> <https://docs.python.org/3/howto/sockets.html>

<sup>6</sup> <https://docs.python.org/3/library/threading.html>

## ۲ آشنایی با Web Server

در این بخش با نحوه ی تعامل با یک web server بر مبنای پروتکل http و نیز با فرمت json آشنا می شویم.

۱. نام و شماره دانشجویی خود را (هر کدام با key مناسب) در یک دیکشنری ذخیره کنید و دیکشنری حاصل را با کمک دستور dumps از کتابخانه ی <sup>7</sup>json، به یک json object تبدیل کرده و نتیجه را مشاهده نمایید.<sup>۸</sup> با کمک دستور loads از همین کتابخانه، رشته ی حاصل را به فرمت اولیه برگردانید و سعی کنید با مقدار دهی به key، به value متناظر دسترسی پیدا کنید.

۲. پروتکل http از دو متد اصلی get و post به ترتیب برای درخواست و ارسال اطلاعات در تعامل با web server ها استفاده می نماید. هنگامی که با استفاده از مرورگر خود یک صفحه ی وب را باز می نمایید، در واقع متد get روی آدرس url آن صفحه فراخوانی می گردد. در نتیجه web server مربوطه بر مبنای پروتکل http، اطلاعات صفحه را در قالب html ارسال می نماید و مرورگر به این طریق صفحه ی درخواست شده را نمایش می دهد. web service هایی بر بستر اینترنت وجود دارند که بجای html، در پاسخ به فراخوانی متدهای خاص http، json object بر می گردانند. این وب سایت ها توسط توسعه دهندگانی که قصد تست نرم افزارهای خود را دارند مورد استفاده قرار می گیرند. این آدرس<sup>۹</sup> را در مرورگر خود باز نمایید و با ساختن یک bin id یک دریافت کنید.

۳. کتابخانه ی <sup>10</sup>requests کار با پروتکل http را بسیار تسهیل می نماید. json object ساخته شده در قسمت قبل را با متد post روی این ID ارسال کنید و با بررسی status code و تطبیق دادن آن با لیست استاندارد status code ها<sup>۱۱</sup> وضعیت درخواست خود را بررسی کنید. حال به کمک متد get محتویات موجود روی همین bin id را خوانده و نمایش دهید. همچنین می توانید json object خروجی را در صفحه ی مرورگر مشاهده نمایید. این web server در پاسخ به درخواستی که دریافت می کند، اطلاعاتی از جمله آدرس ip فرستنده و... را بر می گرداند.

<sup>7</sup> <https://docs.python.org/3/library/json.html>

<sup>8</sup> در واقع این json object چیزی جز یک رشته نیست که در شبکه منتقل می شود.

<sup>9</sup> <https://postb.in/>

<sup>10</sup> <http://docs.python-requests.org/en/master/user/quickstart>

<sup>11</sup> [https://en.wikipedia.org/wiki/List\\_of\\_HTTP\\_status\\_codes](https://en.wikipedia.org/wiki/List_of_HTTP_status_codes)

## Final Project

در این پروژه قصد داریم با استفاده از داکر سرویسی را پیاده سازی کنیم که در آن قابلیت انجام پردازش روی فایل های عکس مهیا شود، به همین منظور لازم است که ابتدا با مفهوم داکر و فرایند داکرایز کردن سرویس آشنا شوید. در ادامه سرویسی را به صورت سرور کلاینت تعریف کنید. وظیفه سرور پایتونی شما آن است که به صورت Image قابل اجرا در داکر درآمده و با هر بار ساخت آن، فولدری را که به عنوان Volume برای آن تعریف کرده اید را بگردد و لیستی از تمام تصاویر داخل آن ایجاد کند. این سرور با هر بار اجرا، روی IP لوکال، وب سروری را مهیا کرده تا نسخه کلاینت از آن استفاده کند. این IP به منظور یک API برای نسخه کلاینت عمل میکند، بدین صورت که نسخه کلاینت شما، با استفاده از دستور زیر می تواند به سرور کوئری زده و درخواست پردازش روی عکس مدنظر را دهد:

$$Data = \{PicName: "Name", Func: "FaceDetection" OR "Serial"\}$$

$$Q = request.get(IP_{Local}, data = Data)$$

در صورتی که در Func عملیات Serial درخواست شده باشد:

سرور وظیفه دارد در جواب درخواست، نام عکس مذکور را در بین فایل های موجود در فایل Volume چک کرده و در صورت وجود، سریال شده عکس را برگرداند. اما در صورتی که عکس مذکور در Volume وجود نداشت پیام عدم وجود عکس را در جواب درخواست برگرداند.

در صورتی که در Func عملیات Face Detection درخواست شده باشد:

سرور وظیفه دارد در جواب درخواست، نام عکس مذکور را در بین فایل های موجود در فایل Volume چک کرده و در صورت وجود، جواب پرسش Face Detection تصویر مذکور را برگرداند. اما در صورتی که عکس مذکور در Volume وجود نداشت پیام عدم وجود عکس را در جواب درخواست برگرداند.

خاطر نشان می گردد برای اجرای قسمت های مختلف هیچ گونه محدودیتی در استفاده از کتابخانه های متعدد وجود ندارد.

مهلت ارائه: ۱۷ دی ۱۴۰۰