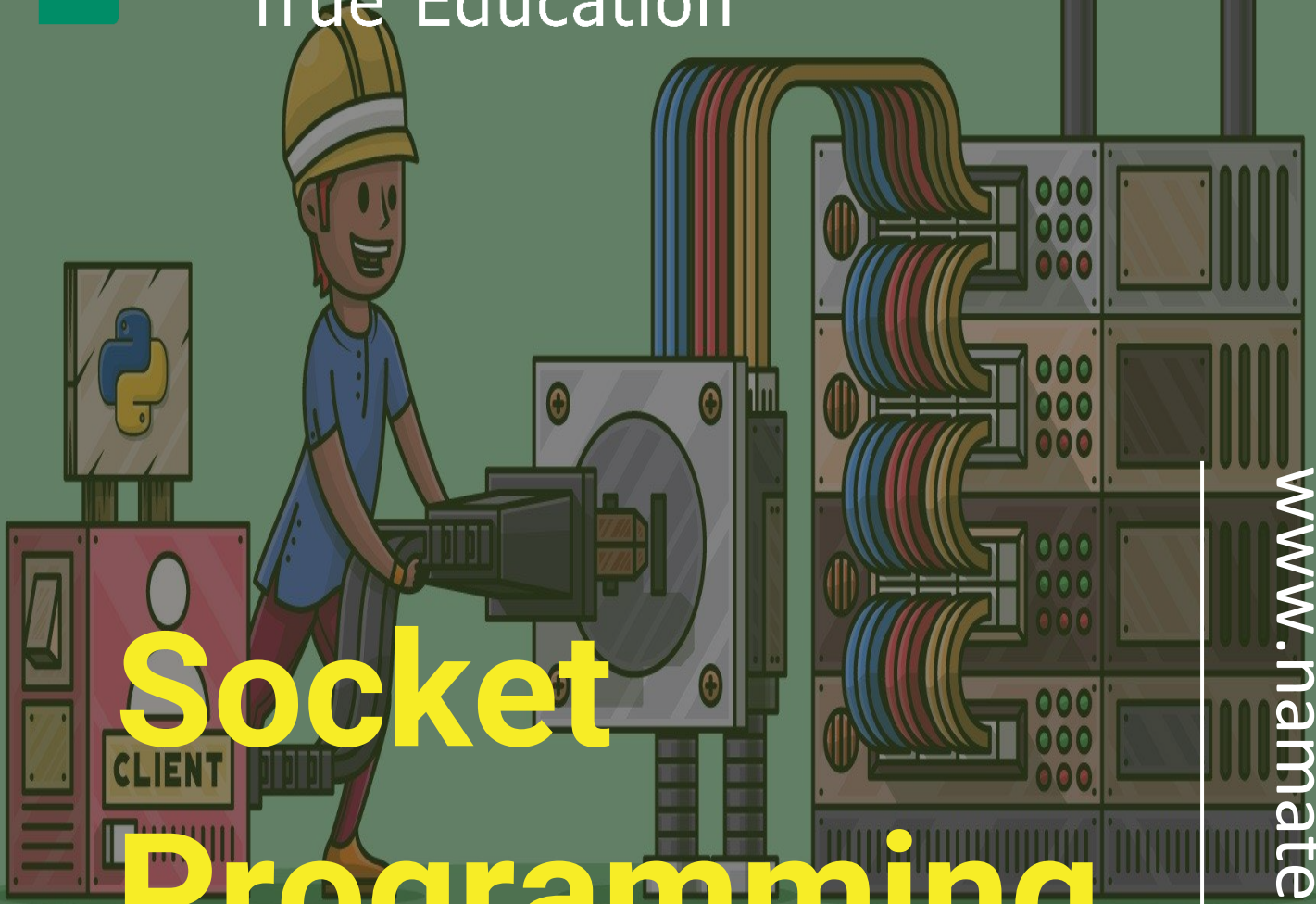




Namatek
True Education



Socket Programming in Python

www.namatek.com

Real Python

سوکت نویسی با
پایتون

فهرست مطالب

۱. تعریف سوکت نویسی با پایتون
۲. انواع سوکت در سوکت نویسی با پایتون
۳. اصطلاحات رایج در شبکه
۴. سرور چیست؟ (server)
۵. کلاینت (client)

سوکت نویسی با پایتون از جمله مباحث جذاب حوزه برنامه نویسی تحت شبکه است که عده زیادی از مردم تمایل دارند اطلاعات گسترده ای در رابطه با آن به دست آورند. استفاده ساده از زبان برنامه پایتون در سوکت نویسی باعث محبوبیت بسیار آن در کاربران این حوزه شده است. از همین رو، در این مقاله سعی کرده ایم تا شما را با نحوه سوکت نویسی (Socket Programming) در پایتون بیشتر آشنا کنیم.

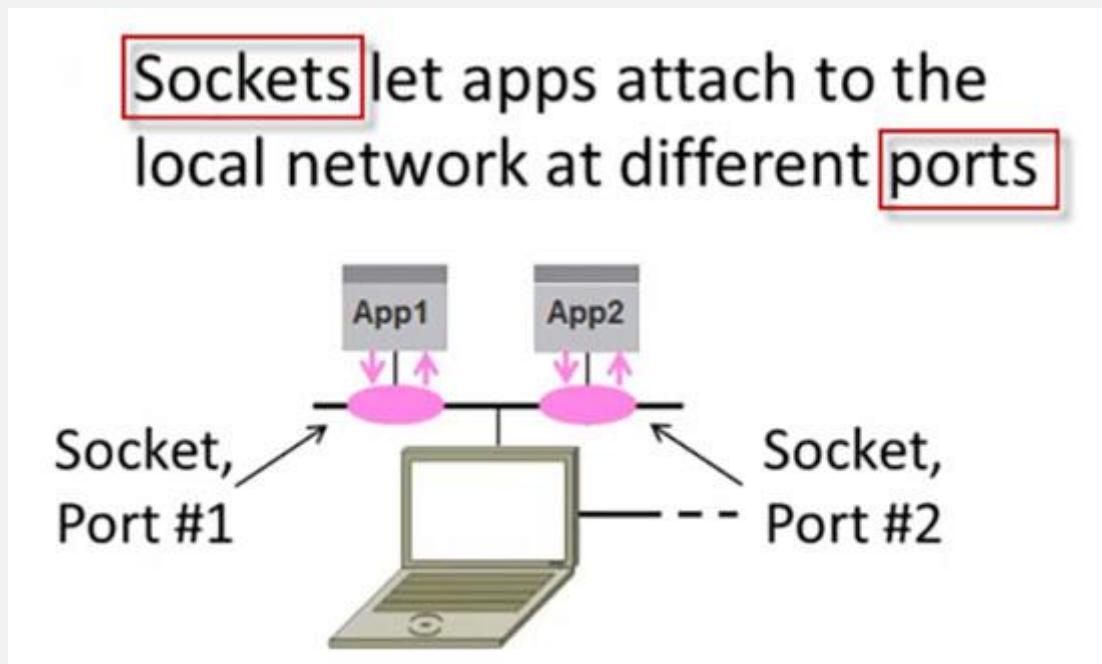
#1 تعریف سوکت نویسی با پایتون

سوکت ها (sockets) کانال های ارتباطی هستند که ارتباط بین دستگاه ها را برقرار می کنند تا ارسال و دریافت داده ها صورت بگیرد. هر دستگاه یا برنامه ارتباطی باید یک socket داشته باشد. در نتیجه هر شبکه واحد دارای دو سوکت خواهد بود. این سوکت ها ترکیبی از آدرس آی پی (IP) و شماره درگاه یا همان پورت (Port) هستند. بر اساس شماره پورت مورد استفاده، یک دستگاه واحد می تواند تعداد n سوکت داشته باشد. پورت های مختلف، برای انواع پروتکل ها وجود دارند. برای برنامه نویسی تحت شبکه در پایتون، باید از فریم ورک (framework) یا کتابخانه استاندارد سوکت استفاده کرد. همچنین لازم است تا کدگذاری در دو سمت سرور (server) و کلاینت (client) صورت بگیرد.



#2 انواع سوکت در سوکت نویسی با پایتون

سوکت ها انواعی دارند که در این جا ما به دو نوع پرکاربرد آن اشاره می کنیم.



۱. سوکت های دیتاگرام (Datagram sockets):

سوکت دیتاگرام یا بدون اتصال (connectionless) بر اساس پروتکل UDP کار می کند. از این رو هیچ تضمینی برای رسیدن داده ها، صحت اطلاعات و حفظ ترتیب آن ها وجود ندارد. با این حال همچنان در انتقال صدا و تصویر از این نوع سوکت ها استفاده می شود. یکی از مهم ترین دلایل استفاده از این سوکت، سرعت بالا در انتقال داده ها است.

۲. سوکت های نوع استریم (stream):

در سوکت نویسی با پایتون اساس کار سوکت استریم یا اتصال گرا (connection oriented) بر پایه پروتکل های TCP است. این نوع سوکت ها داده ها را با نظم و دقت خاصی ارسال و دریافت می کنند؛ زیرا قبل از آن که انتقال اطلاعات صورت گیرد، به یک منبع امن و قدرتمند متصل می شوند.

#۳ اصطلاحات رایج در شبکه

زمانی که قصد آشنایی با سوکت نویسی با پایتون را دارید، بهتر است شناختی راجع به اصطلاحاتی که در شبکه استفاده می شوند، داشته باشید.



۱. آدرس IP

IP از مخفف کلمه Internet Protocol گرفته شده است.

انواع IP عبارتند از:

- IPv4: آی پی نسخه ۴، از چهار بخش عددی تشکیل شده است که با استفاده از نقطه از هم جدا می شوند. در مبنای ۱۰ هر بخش از ۰ تا ۲۵۵ هستند.
- IPv6: افزایش نیاز مردم به اینترنت، تیم مهندسی اینترنت را به این فکر انداخت که آدرس های IP بسیاری را طراحی و تعریف کنند. بر همین اساس، آی پی نسخه ۶ را رونمایی کردند. این IP از فضای آدرس دهی ۱۲۸ بیتی استفاده می کند که می تواند ۲ به توان ۱۲۸ آدرس بدون تکرار را برای ما تعریف کند.

۲. پورت (port)

هر IP دارای ۶۵۵۳۵ پورت است که هر port باعث اتصال رایانه به برنامه و کامپیوتر دیگر می شود.

۳. پروتکل (Protocol)

در ادامه بررسی سوکت نویسی با پایتون باید به پروتکل ها که قوانینی هستند که تعیین می کنند ماشین ها چگونه با یکدیگر ارتباط برقرار نمایند اشاره کنیم. به عبارت دیگر، این قراردادها به عنوان یک زبان واحد بین سرورها عمل می کنند.

پروتکل ها انواعی دارند که در بخش زیر به دو مورد اصلی آن ها می پردازیم:

پروتکل TCP: این قرارداد که مخفف Transmission Control Protocol و به معنای پروتکل کنترل انتقالی است، زمانی کاربرد دارد که بخواهیم داده هایی با امنیت بالا را به ترتیب ارسال کنیم.

پروتکل UDP: از عبارت User Datagram Protocol گرفته شده است. از این قرارداد زمانی استفاده می شود که نیازی به اتصال نباشد.

#۴ سرور چیست؟ (server)

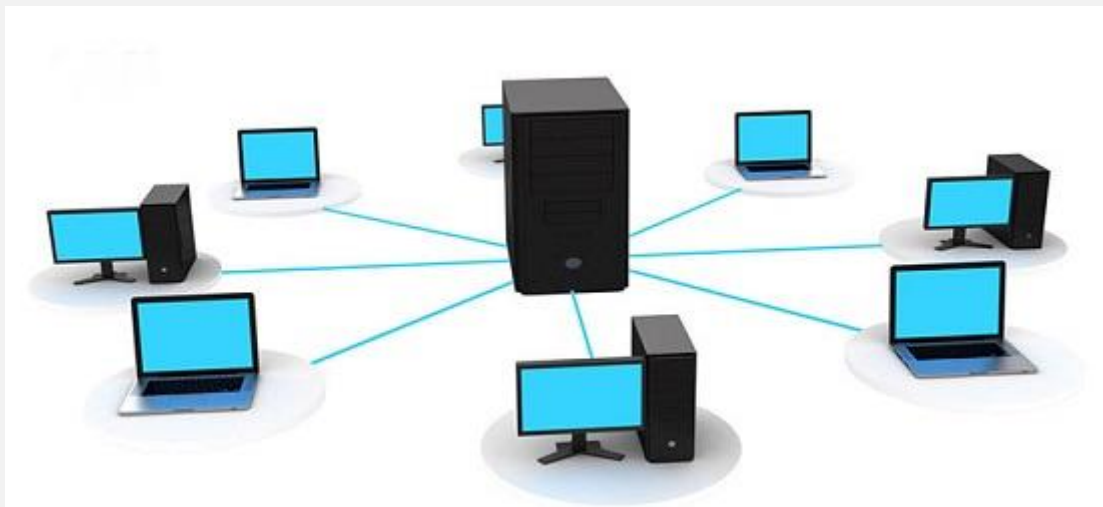
سرورها در سوکت نویسی با پایتون یک رکن اساسی محسوب می شود. سرور در واقع یک کامپیوتر یا دستگاهی است که به مدیریت منابع شبکه اختصاص داده می شود.

انواع سرورها عبارتند از:

- سرورهای پایگاه داده
- سرورهای شبکه
- سرورهای چاپ

سرورها معمولا از روش های زیر برای برقراری ارتباط استفاده می کنند:

- Socket.Socket ()
- Socket.bind ()
- Socket.Listen ()



#۴-۱ سمت سرور در سوکت نویسی با پایتون

برای این که با برنامه نویسی کدهای سمت سرور در سوکت نویسی با پایتون آشنا شوید، توجه شما را به مثال زیر جلب می نمایم.

```
import socket

s=socket.socket(socket.AF_INET, socket.SOCK_STREAM)

s.bind((socket.gethostname(),1234))

#port number can be anything between 0-65535(we
usually specify non-privileged ports which are > 1023)

s.listen(5)
```

برای ایجاد سرور ابتدا باید دو آرگومان را وارد کنیم. آرگومان اول نوع IP را مشخص می کند و آرگومان دوم نوع پروتکل را تعیین می کند. اگر قصد استفاده از IPv4 را داشته باشیم، باید گزینه AF_INET را وارد نمایم و در صورت به کار گیری IPv6، لازم است AF_INET6 را انتخاب کنیم. در این مثال، چون ما از پروتکل TCP استفاده کرده ایم، پس گزینه SOCK_STREAM را بر می گزینیم.

۲-۴ # استفاده از socket.bind()

بعد از اینکه socket خود را ایجاد کردیم، باید امکان دسترسی به آن را فراهم نماییم. در این مرحله از روش socket.bind() استفاده می‌کنیم. در حقیقت، دو آرگومان را وارد خواهیم کرد که اولی IP و دومی port است. برای شماره پورت می‌توان از اعداد بین ۰ تا ۶۵۵۳۵ استفاده کرد؛ اما بهتر است پورت‌های ۱ تا ۱۰۲۴ را به کار نگیریم.

۳-۴ # استفاده از socket.listen()

برای سوکت نویسی با پایتون، بعد از اینکه امکان دسترسی به سوکت فراهم شد باید آن را در حالت listen قرار دهیم تا بتواند درخواست کلاینت‌ها را دریافت کند. برای این کار از روش socket.listen() استفاده می‌کنیم. همان‌طور که در مثال بالا می‌بینید، ما به متد listen() عدد ۵ را داده‌ایم که این نشان‌دهنده تعداد client‌ها است و می‌تواند از ۰ تا ۵ متغیر باشد.

۴-۴ # استفاده از socket.accept()

```
:while True
```

```
    clt,adr=s.accept()
```

```
    print (f"Connection to {adr}established")
```

#f string is literal string prefixed with f which

#contains python expressions inside braces

```
clt.send(bytes("Socket Programming in Python","utf-8 "))  
#to send info to clientsocket
```

بعد از مراحل بالا برای اینکه سرور ما تا بی نهایت پاسخگو باشد، ما یک حالت `while true` را نوشته و ادامه کد را در آن می نویسیم. اکنون باید سرور خود را در انتظار یک کلاینت قرار دهیم. برای این کار از روش `socket.accept()` استفاده کرده و دو متغیر با نام دلخواه انتخاب می نماییم. در مثال بالا مشخصات `client` متصل شده، به صورت `IP` و `port` در قسمت `adr` ذخیره می شود.

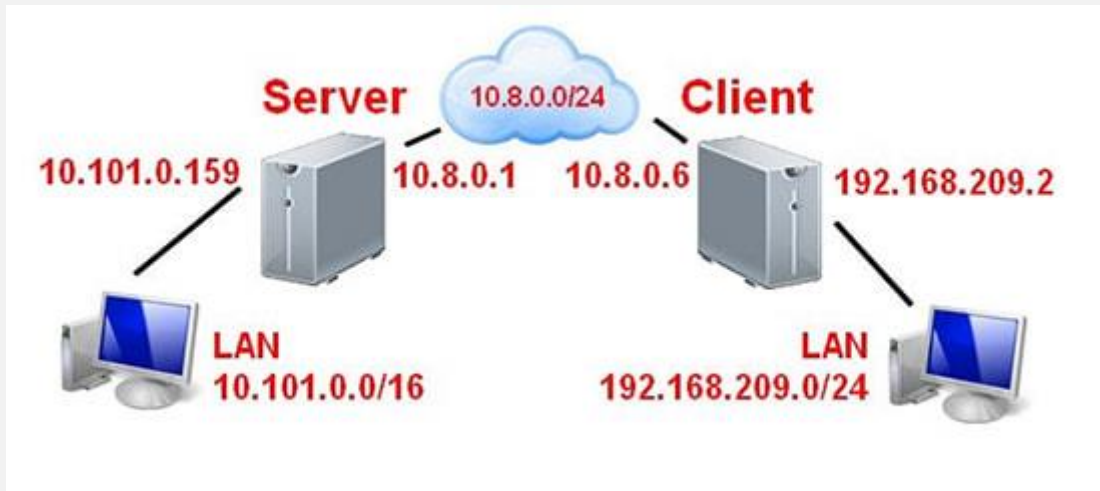
#۴-۵ استفاده از `socket.send()`

برای نمایش برقراری اتصال به کاربر، با استفاده از دستور `print` یک پیغام مناسب به همراه مقدار متغیر `adr` را چاپ می کنیم. اکنون باید پیامی به کلاینتی که به سرور متصل شده بفرستیم. برای این کار از روش `socket.send()` استفاده خواهیم کرد.

#۴-۶ استفاده از `socket.close()`

در پایان تمام مراحل فوق، باید کلاینت متصل شده را با استفاده از متد `socket.close()` ببندیم.

#5 کلاینت (client)



در ادامه سوکت نویسی با پایتون با کلاینت ها سر و کار دارید. Client در شبکه به معنی سرویس گیرنده است که به سرورها متصل شده و از آن ها اطلاعات دریافت می کند. بهترین مثال برای client بازی ها و چت های google chrome است.

#5-1 سمت کلاینت در سوکت نویسی با پایتون

در این بخش با نحوه کدگذاری کلاینت در برنامه نویسی با پایتون بیشتر آشنا می شویم.

```
import socket
```

```
s=socket.socket(socket.AF_INET, socket.SOCK_STREAM)
```

```
s.connect((socket.gethostname(), 2346))
```

```
msg=s.recv(1024)
```

```
print(msg.decode("utf-8"))
```

در اولین قدم، باید ماژول سوکت را مطابق زمانی که سرور را ایجاد کردیم، درست کنیم. بعد از این کار باید از متد connect() استفاده کرده و IP و port داده شده را وارد کنیم. سپس برای دریافت داده ای که از سمت سرور ارسال می شود، یک متغیر به نام msg تعریف کرده و با استفاده از روش socket.recv() پیام را دریافت می کنیم. اکنون متغیر msg را با دستور print چاپ کرده و در آخر سوکت ایجاد شده را با روش socket.close() می بندیم.

حرف آخر

یکی از کاربردی ترین مباحث در حوزه برنامه نویسی، سوکت نویسی با پایتون است که بازار کار خوبی دارد. از این جهت ما به شما توصیه می کنیم به طور جدی به یادگیری این علم بپردازید. امیدواریم توانسته باشیم اطلاعات مفیدی در رابطه با Socket Programming با پایتون خدمت شما ارائه دهیم.