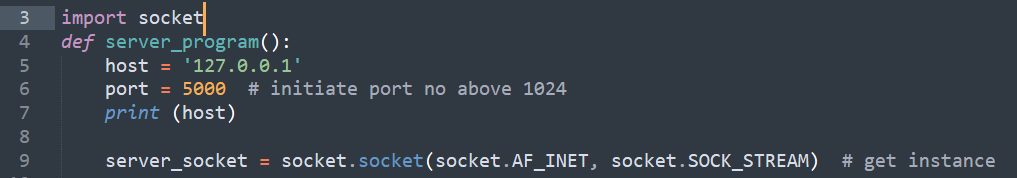
**برنامه نویسی سوکت با پایتون**

**-------------------------------------------------------------------**

در پایتون، برنامه نویسی سوکت (Socket Programming) با استفاده از کتابخانه‌ی استاندارد سوکت (socket) امکان‌پذیر است. در برنامه نویسی سوکت به طور معمول نیاز به برنامه نویسی در دو سمت سرور (Server) و کلاینت (Client) داریم، هر چند امکان برقراری ارتباطات غیر کلاینت-سروری نیز وجود دارد. اما ما در اینجا ارتباطات سرور-کلاینتی را پوشش می‌دهیم، لذا ابتدا کدهای سمت سرور و سپس سمت کلاینت را برنامه نویسی می‌کنیم.

**سمت سرور**

قبل از هر چیز کتابخانه‌ی سوکت را در ابتدای کد خود وارد می‌کنیم و در ادامه یک شی از سوکت می‌سازیم که ما نام آن را server\_socket تعیین کرده‌ایم.



همان‌طور که می‌بینید در هنگام تعریف شی، باید دو آرگومان نیز وارد کنیم که آرگومان اول نوع IP را مشخص می‌کند. در اینجا چون ما قصد استفاده از IPV4 را داریم، گزینه‌ی AF\_INET را وارد کردیم. اما اگر قصد استفاده از IPV6 داشتیم، باید AF\_INET6 را انتخاب می‌کردیم. آرگومان دوم نیز نوع پروتکل را معین می‌کند که در اینجا چون ما از پروتکل TCP استفاده می‌کنیم، گزینه‌ی SOCK\_STREAM را انتخاب کرده‌ایم. حال که سوکت خود را ایجاد کردیم، باید آن را قابل دسترسی کنیم که برای این کار از متد bind() استفاده می‌کنیم. در متد bind() باید دو آرگومان که اولی IP و دومی port است را وارد کنیم. به این صورت:

C:\Users\KPS\Desktop\Capture.PNG

در این مثال، IP استاندارد local host که همان ‘127.0.0.1’ است را وارد کردیم، اما می‌توانیم یک رشته‌ی خالی (" ") نیز استفاده کنیم که در آن صورت، سرور تمامی آدرس‌های IP را می‌پذیرد. ضمن این‌که می‌توانیم به جای IP و رشته‌ی خالی، از نام هاست (hostname) نیز استفاده کنیم، اما ممکن است برنامه رفتار عادی از خود نشان ندهد، لذا بهتر است به جای نام هاست از IP استفاده کنیم. برای شماره پورت نیز می‌توانیم از پورت‌های 1 تا 65000 استفاده کنیم، اما بهتر است از پورت 1 تا 1024 که پورت‌های رزرو شده هستند، استفاده نکنیم. پس از در دسترس قرار دادن سوکت، باید آن را در حالت شنود یا گوش دادن قرار دهیم تا بتواند درخواست اتصال کلاینت‌ها را متوجه شود. برای این کار از متد listen() استفاده می‌کنیم.

C:\Users\KPS\Desktop\Capture.PNG

همان‌طور که مشاهده می‌کنید، ما مقدار 2 را به متد listen() داده‌ایم که این عدد تعداد درخواست همزمان اتصال کلاینت‌ها را تعیین می‌کند و از 0 تا 5 قابل مقداردهی است.

اکنون لازم است سرور خود را منتظر اتصال یک کلاینت کنیم برای همین از متد accept() استفاده می‌کنیم. به این صورت که ابتدا دو متغیر با نام دلخواه، که ما conn و addr را برگزیدیم، تعریف کرده و سپس سوکت ایجاد شده را accept() می‌کنیم.

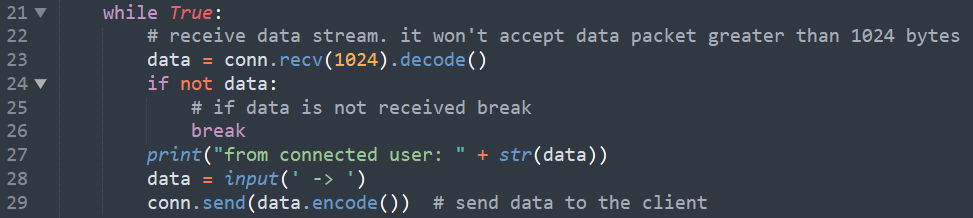
C:\Users\KPS\Desktop\Capture.PNG

در قطعه کد بالا، مشخصات کلاینت متصل شده در conn ذخیره شده و در addr یک تاپل (Tuple) که شامل IP و port کلاینت متصل شده است، ذخیره می‌شود.

برای نمایش برقراری صحیح اتصال به کاربر، با استفاده از دستور print یک پیغام مناسب به همراه مقدار متغیر addr را چاپ می‌کنیم.

C:\Users\KPS\Desktop\Capture.PNG

حال یک حلقه ی true while مطابق زیر در نظر می گیریم تا سرور و کلاینت به یکدیگر پیام ارسال کنند.

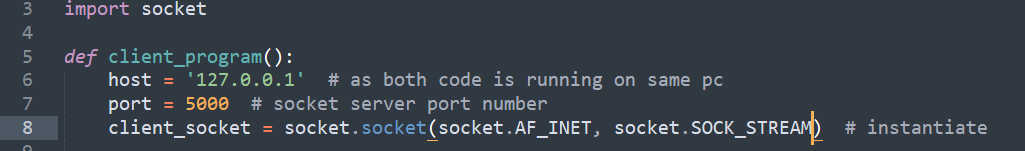


در پایان نیز کلاینت متصل را با استفاده از متد close() می‌بندیم.

C:\Users\KPS\Desktop\Capture.PNG

**سمت کلاینت**

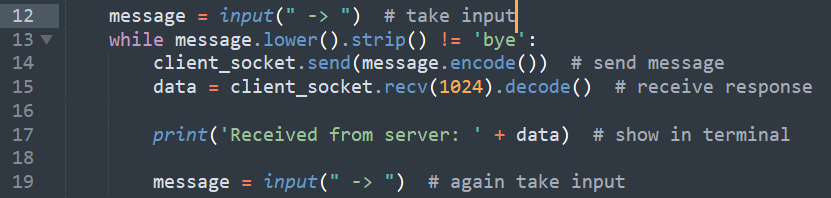
برای برنامه نویسی سمت کلاینت نیز همانند سمت سرور، ابتدا کتابخانه سوکت را داخل کد خود وارد می‌کنیم. سپس یک شی با نام دلخواه از سوکت می‌سازیم توضیحات لازم در مورد آرگومان‌های داخل متد socket() را پیش‌تر در بخش کدهای سمت سرور آورده‌ایم. به این صورت:



اکنون باید سوکت ایجاد شده‌ی خود را به سرور متصل کنیم که برای این کار از متد connect() استفاده می‌کنیم. برای اتصال به سرور باید IP و port داده شده در سرور را در متد connect() وارد کنیم. به این صورت:

**C:\Users\KPS\Desktop\Capture.PNG**

حال یک حلقه ی while مطابق زیر در نظر می گیریم تا سرور و کلاینت به یکدیگر پیام ارسال کنند. پایان ارسال و دریافت پیام ها ارسال کلمه ی “bye” توسط کلاینت است.

****

سپس سوکت خود را با متد close() می‌بندیم.

C:\Users\KPS\Desktop\Capture.PNG