Socket programming project documentation

Mohammad Andalibi & Mohammadmahdi Mohammadi – summer 2020

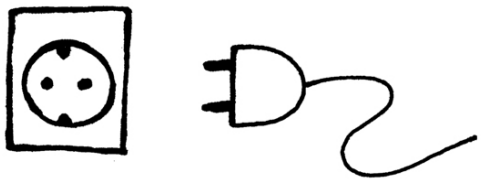
شرح کلی قوانین و فرایند بازی:

Tic-tac-toe یک بازی دو نفره‌است که به‌وسیله یک قلم و کاغذ انجام می‌شود. نام این بازی به دلیل علامت‌های X و O است که در طول بازی استفاده می‌شود. برای آغاز این بازی در یک صفحه جدولی با ۳ ردیف و ۳ ستون رسم می‌شود و هر یک از طرفین یکی از علامت‌های X یا O را انتخاب می‌کنند و تا انتهای بازی برای پر کردن خانه‌های جدول از آن استفاده می‌کنند.برای شروع بازی یکی از طرفین علامت X یا O را که قبلاً انتخاب کرده در یکی از خانه‌های جدول ۹ خانه‌ای قرار می‌دهد. سپس نفر دوم علامت مربوط به خود را در خانه‌های دیگر که هنوز پر نشده‌اند قرار می‌دهد و پس از آن مجدداً نوبت نفر اول خواهد بود.

نقطه پایان بازی در هر مرحله جایی است که یکی از حریفان بتواند علامتی را که در ابتدای بازی انتخاب کرده در یکی از ردیف‌های افقی، عمودی یا قطری قرار دهد و در طول بازی هر یک از طرفین با قرار دادن علامت خود در مقابل علامت‌های حری ف نباید اجازه دهند که حریف یک خط عمودی، افقی یا قطری را با علامت خود پر کند.نمونه یک بازی که در نهایت X بازی را برده‌است :

کم و کیف پیاده سازی بازی در پایتون:

زمانی که شما برنامه نویسی را انجام میدهید، گاهی اوقات نیاز دارید تا با یک کامپیوتر دیگر و یا سرور ارتباط برقرار کنید که این ارتباط از طریق پروتکل انجام می گیرد. برنامه نویسی سوکت در زبان های مختلف مطرح شده است نظیر پایتوان ، جاوا، سی شارپ، اندروید و php و...



سوکت نویسی به برنامه نویس این امکان را میدهد تا با استفاده زبانی که برنامه نویسی میکند بتواند با سیستم های دیگر ارتباط برقرار کرده و اقدام به تبادل اطلاعات نماید. از کاربرد های برنامه نویسی سوکت میتوان در نرم افزار و یا اسکریپت چت نام برد.

سوکت در واقع کانال ارتباطی ما در برنامه است یا دروازه و ابزار ارسال و دریافت اطلاعات بین ما و طرف مقابل در سطح برنامه نویسی.

به طور کلی، هر کامپیوتر (به ازای هر کارت شبکه‌اش) یک IP دارد؛ که از طریق اینIP ، می‌توان به ماشین فوق دسترسی پیدا نمود. این آدرس (IP) به ٦٥٥٣٥ پورت (PORT) تقسیم می‌شود.

PORT نیز یک مفهوم منطقی است که به کمک آن می‌توان بطور همزمان با چندین ماشین دیگر، ارتباط برقرار نمود PORTها به دو گروه رزرو شده (پورت‌های بین ۱ تا ۱۰۲٤) و غیر رزرو شده (سایر پورت‌ها) تقسیم می‌شوند. پورت‌های رزرو شده، برای کاربردهای استاندارد مورد استفاده قرار می‌گیرند. مثلا: در برنامه‌هایServer/Client ، از پورت‌های غیر رزرو شده که آزاد باشند (مورد استفاده‌ی سایر برنامه‌ها نباشند) می‌توان جهت برقراری ارتباطات مورد نیاز، استفاده نمود؛ یعنی می‌تواند به ازای هر پورت، با یک برنامه ارتباط برقرار کند.

نهایتا می توان گفت که سوکت به ترکیب یک آدرس ماشین (IP) و یک شماره درگاه (Port) گفته می‌شود. در برقراری ارتباط بین کامپیوترها در یک شبکه، دو چیز بسیار مهم است:

۱. آدرس ماشینی که می‌خواهیم اطلاعاتی از آن بگیریم یا به آن ارسال کنیم.

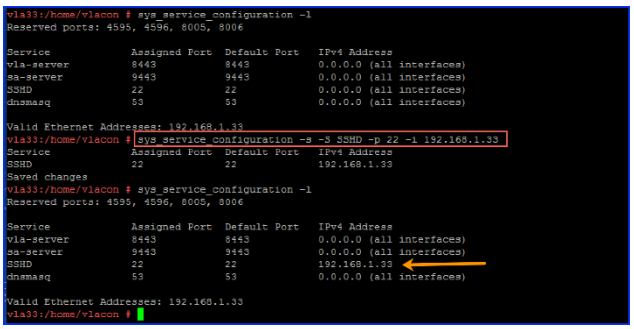
۲. برنامه‌ای از آن ماشین که درخواست اطلاعات کرده یا اینکه می‌خواهیم اطلاعاتی از آن برنامه کسب کنیم.

این دو، یعنی آدرس ماشین و شماره برنامه، به وسیله سوکت در شبکه مشخص می‌شوند. سوکت یک ارتباط قابل اطمینان جهت انتقال داده‌ها بین دو Hostمی‌باشد. سوکت، برنامه‌نویسان را از پیچیدگی‌های فرآیند برقراری ارتباط بین دو ماشین مانند جزئیات کد کردن بسته‌ها، فرآیند ارسال داده‌ها در شبکه، ارسال مجدد بسته‌های خراب و ... دور می‌سازد و برنامه‌نویسان به راحتی قادر به توسعه‌ی برنامه‌های تحت شبکه می‌باشند.

یک سوکت در حقیقت ترکیبی از IP یا HostName و یک شماره پورت (Port Number) از آن IP می‌باشد. و بطور کلی، یک سوکت چهار عمل اصلی زیر را انجام می‌دهد:

اتصال به هاست / ارسال داده ها/ دریافت داده ها/ بستن یا خاتمه اتصال

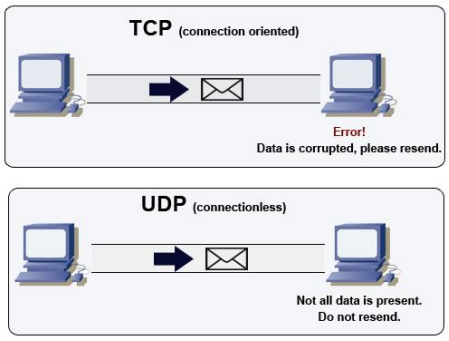
در ارتباطات شبکه، قراردادهایی استفاده می‌شود تحت عنوان Protocol و پروتکول‌های مختلفی ایجاد شده؛ اما در ارتباطات اینترنتی و شبکه‌هایی که ارتباط دو طرفه است، بیشتر از نوع TCP/IP استفاده می‌شود .زمانی که شما اقدام به چت کردن میکنید با یک سیستم دیگر در حال ارتباط هستید که این ارتباط از طریق یک سری پورتکل های از قبل تعریف شده نظیر TCP و UDP انجام می گیرد.



پروتکل TCP :

پروتکلی برای انتقال داده‌هابین دو ماشین با ضریب اعتماد بالا می‌باشد. چنین ارتباطی (بین دو نقطه) را Unicast می‌نامند. TCP پروتکل اتصال‌گرا (Connection-Oriented) است؛ یعنی ارتباط برقرار شده بین کلاینت و سرور تا پایان روند انتقال اطلاعات باقی می‌ماند. این پروتکل در مواردی که اطمینان از انتقال صحیح داده‌ها بین مبدا و مقصد بسیار مهم است (مانند زمان دانلود کردن داده‌ها)، مورد استفاده قرار می‌گیرد. در چنین حالتی، کامپیوتر مقصد با دریافت صحیح هر بسته از مبدا، یک Acknowledgment به منظور اطلاع از دریافت صحیح و بی‌عیب، به ماشین مبدا ارسال می‌کند؛ سپس اگر پیغام اعلام وصول فوق به مبدا نرسد، مبدا دوباره بسته را ارسال می‌کند.

پروتکل UDP :



یک پروتکل بدون اتصال (Connection-Less) می‌باشد. این پروتکل، امکان توزیع داده‌ها را با سرعت بالا تضمین می‌کند؛ ولی هیچ تضمینی در جهت صحت ارسال داده‌ها و دریافت آن‌ها توسط ماشین مقصد ارائه نمی‌دهد. براساس شرایط این پروتکل، اگر در روند انتقال اطلاعات مشکلی پیش آید و بسته‌ی ارسال شده به صورت کامل به مقصد نرسد، بسته‌ی فوق، مجددا برای کامپیوتر مقصد ارسال نخواهد شد.

از این پروتکل، به منظور انتقال داده‌ها به چندین ماشین و با استفاده از BroadCast و یا MultiCast استفاده می‌شود. در شبکه‌هایی که ارتباط دو طرفه نیاز است، از TCP/IP استفاده میشود چون انواع دیگری هم از جمله UDP وجود دارند که تقریبا ارتباط یک طرفه می‌باشد؛ مثلا برای پخش برنامه‌های رادیویی.

تفاوت بین IPv4 و IPv6

IP یک آدرس دودویی است اما برای درک بهتر از سوی کاربران به‌صورت اعداد ده‌دهی در قالب یک رشته نمایش داده می‌شود. برای مثال یک آدرس مبتنی بر IPv4 به‌صورت چهار دسته عدد سه‌تایی نوشته می‌شود که توسط یک نقطه از هم جدا می‌شوند. هر یک از چهار دسته عدد سه‌تایی می‌تواند مقادیر صفر تا ۲۵۵ داشته باشد. برای مثال ۱۰/۱۶۰/۱/۲۴۰ یک آدرس اینترنتی مبتنی بر IPv4 است.

IPv6 یک آدرس اینترنتی ۱۲۸ بیتی است که به‌صورت هگزادسیمال (دستگاه اعداد مبنای ۱۶) نوشته می‌شود و اجزای آن با استفاده از کالِن از هم منفک می‌شوند. در دستگاه هگزادسیمال، علاوه بر اعداد ۰ تا ۹، اعداد ۱۰، ۱۱، ۱۲، ۱۳، ۱۴ و ۱۵ با حروف A تا F نمایش داده می‌شوند.

بررسی کتابخانه ها و توابع مورد استفاده در پیاده سازی:

ماژول سوکت - برنامه نویسی سمت سرور:

برای پیاده سازی باید از پلتفورم سوکت نویسی در پایتون که همان کتابخانه سوکت استفاده میباشد استفاده کرد.

در اولین گام، باید کتابخانه سوکت رو صدا بزنیم.

```
Import socket
s = socket.socket( socket.AF_INET , socket.sock_stream)
s.bind((socket.gethostname() , 1024))
s.listen(5)
```

در گام دوم، باید از روی کتابخانه socket یک شی بسازیم

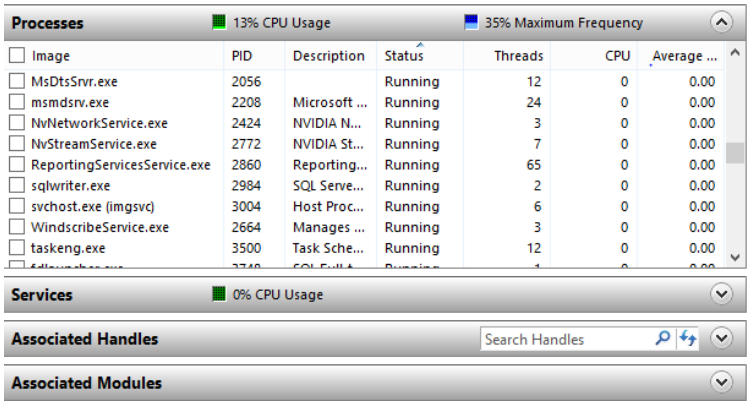
حالا باید پورت و شماره ای پی را اصطلاحا bind کرد.

برای اینکار باید آی پی را با پورت سیستم نگاشت کنیم که به این منظور احتیاج داریم پورت های مشغول و آزاد سیستم را بررسی کنیم که پورت انتخابی مشغول نباشد.به این منظور از دستور netstat –a استفاده میکنیم که خروجی آن به شکل زیر میباشد.

```
TCP    0.0.0.0:50248          Mohammadmahdi:0        LISTENING
TCP    127.0.0.1:1001         Mohammadmahdi:0        LISTENING
TCP    127.0.0.1:1434         Mohammadmahdi:0        LISTENING
TCP    127.0.0.1:5354         Mohammadmahdi:0        LISTENING
TCP    127.0.0.1:6942         Mohammadmahdi:0        LISTENING
TCP    127.0.0.1:9990         Mohammadmahdi:0        LISTENING
TCP    127.0.0.1:50507        activate:50508         ESTABLISHED
TCP    127.0.0.1:50508        activate:50507         ESTABLISHED
TCP    127.0.0.1:50509        activate:50510         ESTABLISHED
TCP    127.0.0.1:50510        activate:50509         ESTABLISHED
TCP    127.0.0.1:50511        activate:50512         ESTABLISHED
TCP    127.0.0.1:50512        activate:50511         ESTABLISHED
TCP    127.0.0.1:50844        activate:50845         ESTABLISHED
TCP    127.0.0.1:50845        activate:50844         ESTABLISHED
TCP    127.0.0.1:50846        activate:50847         ESTABLISHED
TCP    127.0.0.1:50847        activate:50846         ESTABLISHED
TCP    127.0.0.1:50848        activate:50849         ESTABLISHED
TCP    127.0.0.1:50849        activate:50848         ESTABLISHED
TCP    127.0.0.1:50850        activate:50851         ESTABLISHED
TCP    127.0.0.1:50851        activate:50850         ESTABLISHED
TCP    127.0.0.1:50852        activate:50853         ESTABLISHED
TCP    127.0.0.1:50853        activate:50852         ESTABLISHED
TCP    127.0.0.1:50854        activate:50855         ESTABLISHED
TCP    127.0.0.1:50855        activate:50854         ESTABLISHED
```

آدرس ۱۲۷ که همان آدرس لوپ بک میباشد و در ادامه پورت های مشغول و آزاد شبکه مشخص گردیده است.اگر پورتی مشغول بود و نیاز به شناسایی برنامه مورد استفاده از آن وجود داشت میتوان از قسمت زیر استفاده کرد.

resource monitor > processes> search by PID



معمولا برای شماره پورت، عدد ۸۰۰۰ رو انتخاب میشود؛ چون هنگام اتصال به نظر میرسد که قوی تر عمل میکند!

در شکل بالا خط دوم کد دو آرگومان ورودی دارد.در این دستور آرگومان اول به نوع IPV6 / IPV4 بودن شبکه و قسمت دوم به UDP / TCP بودن پروتکل شبکه اشاره دارد.

همانگونه که گفته شد UDP به منظور انتقال داده‌ها به چندین ماشین و با استفاده از BroadCast و یا MultiCastاستفاده می‌شود. در شبکه‌هایی که ارتباط دو طرفه نیاز است، از TCP/IP استفاده میشود پس در این شبیه سازی از پروتکل TCP استفاده شده است و به دلیل استفاده از socket.af_inet شیوه آدرس دهی نیز به مدل IPV4 نگاشت شده است.

در زیر شرح هر یک از این پارامترها را مشاهده می کنید:

یا AF_UNIX: این پارامتر، همان طور که در بالا توضیح داده شد، می تواند socket_family باشد. AF_INET باشد.SOCK_DGRAM و یا SOCK_STREAM: این پارامتر می تواند یا socket_type تنظیم می شود.۰: این پارامتر اختیاری بوده و به صورت پیش فرض بر روی protocol

، می توانید با استفاده از توابع لازم، برنامه های سمت سرویس دهنده و سمت socket پس از تعریف آبجکت سرویس گیرنده ی خود را تعریف نمایید. جداول زیر لیست توابع لازم برای این منظور را معرفی می کند.

متد	شرح
s.bind()	این متد آدرس (hostname یا اسم سرویس دهنده، جفت آدرس پورت یا port number pair) را به socket به صورت دو طرفه وصل می کند.
s.listen()	این متد یک گوش فرادهنده (Listener) به TCP تنظیم و راه اندازی می کند. در واقع این متد به
s.accept()	این متد درخواست اتصال به سرویس دهنده را می پذیرد و به عبارتی ارتباط معلق را به سرور معرفی می کند.

متدهای ماژول socket مربوط به سمت سرویس گیرنده

متد	شرح
s.connect()	این متد اتصال به سرویس دهنده ی را بر اساس TCP راه اندازی می کند.

متدهای کلی ماژول socket

متد	شرح
s.recv()	این متد پیغام TCP را دریافت می کند.
s.send()	متد حاضر پیغام TCP را ارسال می کند.
s.recvfrom()	متد جاری پیغام UDP را دریافت می کند.
s.sendto()	این متد پیغام UDP را ارسال می کند.
s.close()	این متد socket را می بندد.
socket.gethostname()	اسم سرویس دهنده (hostname) را در خروجی برمی گرداند.

شبه کد های سرور و کلاینت

```
#!/usr/bin/python       # This is server.py file
import socket           # Import socket module
s = socket.socket()     # Create a socket object
host = socket.gethostname() # Get local machine name
port = 12345            # Reserve a port for your service.
s.bind((host, port))    # Bind to the port
s.listen(5)             # Now wait for client connection.
while True:
  c, addr = s.accept()  # Establish connection with client.
  print 'Got connection from', addr
  c.send('Thank you for connecting')
  c.close()             # Close the connection
```

```
#!/usr/bin/python       # This is client.py file
import socket           # Import socket module
s = socket.socket()     # Create a socket object
host = socket.gethostname() # Get local machine name
port = 12345            # Reserve a port for your service.
s.connect((host, port))
print s.recv(1024)
s.close                 # Close the socket when done
```

```
# Following would start a server in background.
$ python server.py &
# Once server is started run client as follows:
$ python client.py
```

بحث زمان و جزییات اطلاعات بازی:

زمان در سیستم‌های کامپیوتری با ثانیه سنجیده می‌شود.همه زمان ها یک مبدأ زمانی دارند. مبدا زمان کامپیوترها یکم Jan سال ۱۹۷۰ میلادی ساعت ۰۰:۰۰ بامداد است. زمان در کامپیوتر به صورت ثانیه‌های گذشته شده از ساعت مبدا محاسبه می‌شود. سپس با تبدیل محاسباتی خاص، می‌توان آن‌را تبدیل به ساعت‌ها و تاریخ کرد.برای استفاده از این ماژول ابتدا باید آن‌را وارد برنامه کنیم. حال می‌توانیم با صدا زدن توابع مختلف روی timeکارهای مربوط به زمان را انجام دهیم.

اولین تابعی که در این ماژول وجود دارد، تابع time()است. این تابع، زمان فعلی سیستم را به ما خروجی می‌دهد. خروجی به صورت floatبوده و همان ثانیه های سپری شده از مبدا می‌باشد. تابع دیگر از ماژول time در پایتون، تابع localtime()است. این تابع یک شئ از نوع time.struct_timeباز می‌گرداند. این شئ در اصل یک tuple(یا چند تایی) حاوی اطلاعاتی از زمان حال است.

دیدیم که به کمک تابع time()می‌توانیم زمان را به صورت ثانیه گرفته و نگه داریم. حال ممکن است یک مقدار عددی (ثانیه) داشته باشیم که بخواهیم آن‌را تبدیل به زمان کنیم. برای این کار از تابع ctime()در کتابخانه زمان پایتون استفاده خواهیم کرد.

```
import time
t = time.ctime(time.time())
now = time.localtime()
file_name = "test2.txt"
file = open(file_name, "w")
file.write(t)
file.write(" \n")
file.close()
```

سایر متود ها :

I. **sys.argv** یک فهرست است که شامل همه آرگومان‌های وارد شده از سوی کاربر در زمان اجرای اسکریپت است. این فهرست شامل نام خود اسکریپت نیز می‌شود.

II. یک برنامه با استفاده از نخ‌ها می‌تواند به صورت همزمان به اجرای چندین عملیات در یک فضای فرایند بپردازد. یک نخ دارای یک نقطه‌ی شروع، دنباله‌ی اجرا و یک نتیجه است و همچنین دارای نشان‌گر دستورالعمل است که وضعیت فعلی نخ را نگه داشته و ترتیب اجرای دستورات بعدی را کنترل می‌کند. بنابراین توانایی یک فرایند برای اجرای چندین نخ به صورت موازی را می‌توان چند نخی **(multithreading)** نامید. چند نخی می‌تواند عملکرد هر برنامه را به صورت قابل توجهی بهبود ببخشد .

نحوه کار با کتابخانه‌یthreading

کتابخانه‌ی threading دارای چندین متد است که در پیاده‌سازی نخ‌ها به ما کمک می‌کند.

- run این متد نقطه ورود هر نخی است.
- start وظیفه این متد راه اندازی نخ‌هاست زمانی که متد run فراخوانی شود.
- join این متد به برنامه این امکان را می‌دهد تا برای خاتمه دادن نخ‌ها منتظر بماند.
- isAlive متدی است که نشان دهنده‌ی فعال بودن یا نبودن نخ است.
- getName با این متد می‌توان نام نخ را بازیابی کرد.
- setName توسط این متد می‌توان نخ را نام گذاری کرد.

قدم‌های ایجاد نخ

۱. زیر کلاسی از کلاس Thread ایجاد می‌کنیم.
۲. متد __init__ را طبق نیاز خود override می‌کنیم.
۳. در آخر هم متد run را با توجه به منطق برنامه می‌نویسیم.

زمانی که زیر کلاس ساخته شد، باید از آن نمونه‌ای بسازیم و سپس متد start را فراخوانی کنیم. متد start پیکربندی‌های اولیه‌ی ساخت نخ را انجام می‌دهد و سپس متد run را فراخوانی می‌کند.

همگام سازی نخ‌ها

کتابخانه threading عملکردی برای قفل کردن نخ‌ها دارد. عملیات قفل کردن به ما این امکان را می‌دهد تا حافظه مشترک را مدیریت و از خراب شدن داده‌ها جلوگیری کنیم.برای قفل کردن از متد Lock استفاده می‌کنیم. سپس با استفاده از متد acquire قفل را فعال و با متد release آزاد می‌کنیم.

III. **ماژول logging :** فایل های log به صورت اتوماتیک توسط برنامه ها ایجاد و گزارشی از عملکرد، اتفاقات و مشکلاتی که آنها رخ داده است را داخل آنها نوشته تا برنامه نویس بعدا به اون فایل رجوع کرده و وضعیت برنامه رو بررسی کند.

به این پیغام ها اصطلاحا log گفته می شود. برای نوشتن فایل های log به صورت استاندارد، می توان از ماژول logging در پایتون استفاده نمود. به طورکلی، می توان سطوح پیغام های فایل log را به ۵ سطح (level)تقسیم نمود:

سطح ۱، debugدر این سطح، پیغام هایی با بیشترین جزییات در فایل log نوشته می شوند.

سطح ۲، infoدر این سطح، پیغام هایی نوشته می شوند که مشخص کننده اجرای برنامه مطابق انتظار است.

سطح ۳، warningدر این سطح، پیغام های اخطاری نوشته می شوند که مشخص می کند مشکلی در برنامه وجود دارد که باید به آن توجه نمود و ممکن است در آینده باعث ایجاد مشکلی در اجرای برنامه شود. ولی در حال حاضر، برنامه بدون مشکل در حال اجرا است.

سطح ٤، errorدر این سطح، پیغام های خطایی نوشته می شود که مشخص می کند برنامه دچار خطایی شده و برخی از قسمت های آن درست اجرا نمی شوند.

سطح ٥، criticalدر این سطح، پیغام های خطایی نوشته می شود که مشخص کننده خطایی جدی است و اجرای برنامه نمی تواند ادامه یابد.

basicConfig(filename, format, level)

۱. filenameدر این پارامتر که ورودی آن یک رشته (string) است، می بایست آدرس و نام فایل مشخص گردد.

۲. formatدر این پارامتر، می بایست چیدمان مشخصاتی که در هر پیغام فایل log مورد نظر هستند(مانند سطحlog ، زمان و تاریخ، پیغام) تعیین شود. ماژول logging قابلیت های فراوانی را برای نوشتن فایل log در اختیار کاربر می گذارد. از مهمترین پارامترهای قابل ذکر در فایل log می توان به موارد زیر اشاره نمود.

۳. Levelدر این پارامتر تعیین می شود که چه سطح از پیغام هایی در فایل log نوشته شود. ممکن است در کد پیغام ها با سطوح مختلف تعیین شده باشند ولی فقط بخش خاصی از آنها مورد نظر کاربر باشد. در این پارامتر، هر سطحی که تعیین شود، پیغام های آن سطح به بالا نمایش داده می شود. برای مثال اگر سطح، برابر یا WARNING قرار بگیرد، پیغام هایwarning ، errorو critical نمایش داده می شوند.

برای تعیین سطح در این پارامتر، حتما باید تمامی حروف سطح مورد نظر، بزرگ نوشته شوند.پس از تعیین مشخصات فایلlog ، می توان با استفاده از روش هایdebug ، info، warning، errorپیغام critical، های مورد نظر در هر زمینه را نوشت. ورودی تمامی این روش ها، رشته (string) است.

اگر بخواهیم، یک یا چند متغیر را در پیغام log نمایش دهیم، می توان از دو حالت استفاده نمود:

- استفاده از تابع str
- استفاده از کاراکتر % و حروف تعریف