

| 자랑 게시판 | >

[스압] 정부에서 차단한 사이트 파싱하기 ~ 신사의 고집 ~



윤 챗봇 입문자 1:1 채팅

2020.06.08. 18:33 조회 334

댓글 32 URL 복사



안녕하세요. 이번주 인기멤버가 되지 못 하여 슬픈 윤입니다.
슬슬 소재고갈이라 어떤 글을 써야 할지 모르겠습니다. 벌써 고갈이라니, 멍청이!

원래는 **Node.JS** 가 어떻게 싱글스레드로 돌아가는지, **libUV** 와 이벤트 루프에 대한 개념을 작성하려 했는데
솔직히 지루하기도 하고 내용도 방대해서 패려웠습니다.

그보다 이번 제목에 이끌려 오신 분들 많잖아요? (͡° ͜° ͜° ͜° ͜°)
어디 흥미없다고 반박해 보시지!

2019 년 2월 11일, 대대적으로 불법음란물, 불법도박 등 불법 정보를 유통하는 해외 인터넷 사이트 차단되었습니다.
유명한 인터넷 3사 (KT, LG, SKT) 를 사용한다면, SNI 차단이 이루어지고 있습니다.

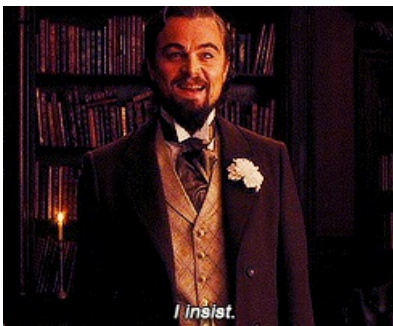
이에 성에 대한 아름다움을 감상, 연구, 분석하는 신사로서 이런 차단은 달갑지 않았습니다.
물론, 시중에 나온 우회 앱을 사용한다면 매우 편하겠지만, 우리 개발자답게 원초적인 방법으로 우회를 시도해 봅시다.

어떠한 패키지 설치도 없는, 순수 **Node.JS** 를 사용하여 프로그래밍을 할 예정입니다.
봇을 만드는 것도 아닌, 우회 이론이니깐요.

- □ 차단되는 기준
- □ DNS Lookup
- □ HTTP Header
- □ TCP 와 HTTP 통신

목차가 4개밖에 없어서 짧죠?
하지만 그 내용은 매우 중요합니다.

□ 차단되는 기준



I insist는 "내가 정말 바라서 그래." 라는 뜻으로 해석 가능하다.
그래. 내가 진짜 바라는 그것은, 낙원이다.

SNI (Server Name Indication) 필드 차단이라고 들어보셨나요?
TCP 통신할 때 Handshake 과정에서 가장 처음에 연결 요청을 할 때는,
정보가 평문으로 노출되기 때문에 이 값을 가지고 차단을 시도하는 것입니다.

2020-06-08-naver-client-s

No.	Time	Source	Destination	Protocol	Length	Info
31	0.245850635			TLSv1.3	583	Client Hello
513	1.178185603			TLSv1.2	583	Client Hello
543	1.214659941			TLSv1.3	697	Client Hello
587	1.317971639			TLSv1.2	583	Client Hello

```

▶ Extension: Reserved (GREASE) (len=0)
▼ Extension: server_name (len=20)
  Type: server_name (0)
  Length: 20
  ▼ Server Name Indication extension
    Server Name list length: 18
    Server Name list: [redacted]

```

```
Server Name type: nost_name (0)
```

```
Server Name length: 15
```

```
Server Name: 1.www.naver.com
```

- ▶ Extension: extended_master_secret (len=0)
- ▶ Extension: renegotiation_info (len=1)
- ▶ Extension: supported_groups (len=10)
- ▶ Extension: ec_point_formats (len=2)
- ▶ Extension: SessionTicket TLS (len=0)

www.naver.com 으로 접근했을 때 잡은 패킷입니다.

직접 패킷을 캡처했을 때, 보이는 저 서버 주소를 보고 차단하는 것이죠.

대놓고 Server Name Indication extension 이라고 나와있네요.

□ DNS Lookup



DNS(Domain Name System) 은, 사람이 읽을 수 있는 도메인 이름 (예. www.naver.com) 을 기계가 읽을 수 있는 IP 주소로 변환합니다.

IP 는 자주 들어보셨을 테니 설명하지 않겠습니다.

DNS 라는 시스템이 있는 이유는, 사용자 친화성 때문입니다. 매번 xxx.xxx.xxx.xxx 라는 아이피를 외워야 특정 사이트로 가게 된다면, 사람의 머리는 복잡해질 것입니다. 그러니 읽을 수 있는 영어로 IP 주소 대신 입력하는 것이죠.

*** 근데 갑자기 DNS는 왜?**

여러분이 브라우저를 켜서 주소창에 www.naver.com 을 입력하고 이동을 하면 브라우저 내에선 www.naver.com 이 누군지 알기 위해 dns query를 합니다.

"www.naver.com 이 누군지 알려줘." 라는 뜻이죠.

우리도 이 DNS 를 사용해야 하니 설명드렸습니다.

DNS 는 UDP 통신을 사용하지만, NodeJS 는 편리하게 [dns](#) 모듈을 지원합니다.

```

1 const dns = require('dns');
2 const lookupOption = {
3   family: 4,
4   hints: dns.ADDRCONFIG | dns.V4MAPPED,
5 };
6 dns.lookup('naver.com', lookupOption, (err, address, family) => {
7   console.log("address:", address);
8 });
9

```

Colored by Color Scripter

간단하게 위 코드를 실행하면 naver.com 의 ip 주소를 얻을 수 있습니다.
125.209.222.142 라는 ip 를 얻었습니다.

□ HTTP Header

HTTP Header 에 대해 하나하나 설명하고 있으면 손가락 아프고 저도 힘듭니다.
HTTP 헤더는 서버와 클라이언트간 통신 중, 추가적인 정보들을 첨부할 수 있도록 합니다.

POST 요청시 데이터를 보내는 것도 HTTP Header 이후의 Body 영역에 작성하는 것입니다.
사실 이것은 전부 특정 규격을 가진 평문입니다.

더 자세한 해더 정보는 [MDN](#) 을 참고하시기 바랍니다.

우리는 GET 요청을 받아오는 데 필요한 딱 4줄만 작성할 것입니다.

```

1 GET / HTTP/1.1
2 User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.39
3 Host: www.naver.com
4

```

GET: HTTP 메소드 타입을 칭합니다.

/: 접근할 경로입니다. \${Host}\${Path} 입니다.

- 위의 경우, [www.naver.com/](#)

HTTP/1.1: HTTP 규격 버전입니다.

User-Agent: 요청자를 식별할 수 있는 문자열입니다. 이게 등록된 곳이 아니면 접근 차단하는 사이트도 여럿 있습니다.

Host: 도메인 명입니다. ([MDN](#))

3번 째 줄은, Header 가 끝났다는 표시입니다.

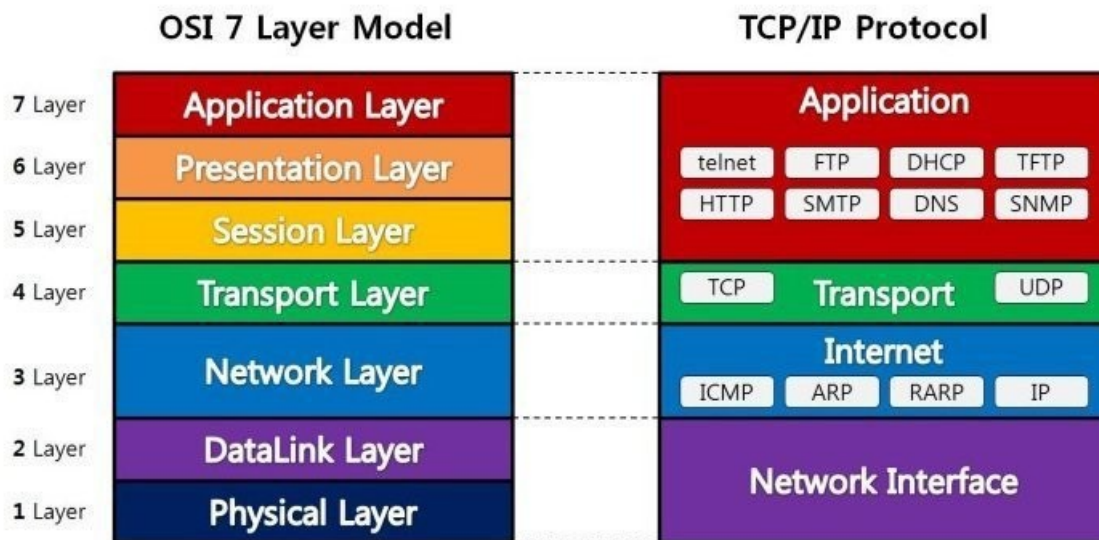
가장 밑에는 개행을 두 개 넣어서 공백을 만들지 않으면 제대로 된 통신이 이뤄지지 않습니다.

□ TCP 와 HTTP 통신



HTTP 프로토콜은 TCP 통신을 이용합니다.

네트워크에 좀 익숙하신 분들이라면, OSI 7계층을 들어보셨을 겁니다.



위 사진에서 볼 수 있듯, TCP 프로토콜은 Transport, HTTP 는 Application 에서 이루어집니다.

TCP/IP 가 HTTP 를 포함하고 있는 것이죠.

그러면 결과적으로, TCP 통신 모듈만 이용하여 HTTP 통신이 가능하다는 뜻입니다.

TCP 통신은 Node.JS 에서 지원하는 기본 모듈이 있습니다.

HTTPS 는 암호화 된 HTTP여서, TCP가 암호화 된 TLS 를 사용합니다.

근데 이 또한 Node.JS 에서 지원하는 모듈이 있습니다.

TLS 패키지에 대해 자세한 내용은 다음 [문서](#)에서 확인하세요.

```
1 const dns = require('dns');
2 const dnsPromises = dns.promises;
3 const tls = require('tls');
4 const fs = require('fs');
5
6
7 (async () => {
8   try {
9     const host = 'www.naver.com';
10    const res = await dnsPromises.lookup(host, { family: 4, hints: dns.ADDRCONFIG | dns.V4MIF
```

```

11
12 console.log(res);
13 socket = tls.connect({
14   host: res.address,
15   port: 443,
16   cert: fs.readFileSync('public-cert.pem'),
17   rejectUnauthorized: false,
18 }, () => {
19   console.log("connect");
20 });
21
22 socket.setEncoding('utf8');
23 socket.on('data', (data) => {
24   console.log("data:", data);
25 });
26 socket.on('end', () => {
27   socket.end();
28 });
29
30 let req = "";
31 req += "GET / HTTP/1.1\n";
32 req += "User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
33 req += `Host: ${host}\n`;
34 req += "\n";
35 socket.write(req);
36 } catch (err) {
37   console.error(err);
38 }
39 })()
40

```

www.naver.com/ 에 GET 요청을 보내는 코드입니다.

public-cert.pem 은 SSL 암호화에 사용되는 공개 키인데, 설명하기 귀찮으니 하지 않겠습니다.

openssl 명령어를 사용하여 만들었습니다.

port 443 은 https 통신의 기본 포트입니다.

참고로 http 의 기본 통신 포트는 80 입니다.

rejectUnauthorized 는 신뢰할 수 없는 페이지에 대한 차단을 무시하는 옵션입니다.



연결이 비공개로 설정되어 있지 않습니다.

공격자가 127.0.0.1에서 정보(예: 비밀번호, 메시지, 신용카드 등)를 도용하려고 시도 중일 수 있습니다. [자세히 알아보기](#)

[← 안전한 페이지로 돌아가기](#)

이 서버가 127.0.0.1임을 입증할 수 없으며 컴퓨터의 운영체제에서 신뢰하는 보안 인증서가 아닙니다. 서버를 잘못 설정했거나 불법 사용자가 연결을 가로채고 있기 때문일 수 있습니다.

❗ 127.0.0.1(안전하지 않음)(으)로 이동

net::ERR_CERT_AUTHORITY_INVALID

이런 겁니다.

자, 이제 대망의 차단 사이트로의 요청을 넣어봐야죠.

세계 최대 사이트이자, 음지계 구글이라 불리는 PxxxHub 에 요청하도록 하겠습니다.



```
width="150"
class="rotating" data-video-id="301229232" data-thumbs="16" data-path="https://dl.phncdn.com/videos/202004/07/301229232/original/(m=ewdFGaaaa)(mh=EjmcOYxTAMRbtq6)(index).jpg"
Big tit blondes, Katie Morgan, Brandi Love share lil Ric" />
data:
  <div class="marker-overlays js-nofade">
    <var class="duration">12:00</var>
    <span class="hd-thumbnail">HD</span>
  </div>
  <div class="add-to-playlist-icon display-none">
    <button type="button" data-title="Add to a Playlist" class="tooltipTrig open-playlist-link playlist-trigger" onclick="return false;" data-rel="ph5eqca1j39431f"></button>
  </div>
  <div class="thumbnail-info-wrapper clearfix">
    <a href="/view_video.php?viewkey=ph5eqca1j39431f" title="Lil Humpers - Two big tit blondes, Katie Morgan, Brandi Love share lil Ric" class="">
      Lil Humpers - Two big tit blondes, Katie Morgan, Brandi Love share lil Ric
    </a>
    <div class="videoUploaderBlock clearfix">
      <span class="channel-icon main-sprite"></span>
    </div>
    <div class="usernameWrap">
      </div>
    <a href="/channels/lil-humpers" class="bolded">Lil Humpers</a>
  </div>
  <div class="videoDetailsBlock">
    <span class="views">(var)1.6M</var> views</span>
    <div class="rating-container neutral">
      <div class="main-sprite icon"></div>
      <div class="value">73</div>
    </div>
    <var class="added">3 hours ago</var>
  </div>
</li>
<li class="pcVideoListItem js-pop videoblock videoBox" id="v275010561" _vkey="ph5e6321dcce" data-id="275010561" data-segment="straight" data-entrycode="VidPg-preVid">
  <div class="wrap">
    <div class="phimage">
      <div class="preloadLine"></div>
      <div class="fadeUp videoPreviewBg linkVideoThumb js-linkVideoThumb img">
        
      </div>
    </div>
  </li>
</ul>
```

이런 저런 소스가 보이는 것을 보니, 우회하여 요청을 하는데 성공하였습니다.

* 왜 이게 가능한 걸까?

위에서 보았던 SNI 에 대한 패킷 이름은, Server Name Indication extension 입니다.

extension 은 연장, 확장 등으로 해석 가능한데 즉, 원래 있는 것에서 확장을 한다는 뜻입니다.

없어도 그만이란 것이죠.

그래서 저희는 Server Name을 추가하지 않도록 하며, 해당 아이피로 직접 통신을 요청한 것입니다.

패킷에 해당 데이터가 존재하지 않으니, 검열할 것도 없는 것이죠.

마지막으로, cheerio 모듈을 사용하여 파싱해보겠습니다.

```

1 const dns = require('dns');
2 const dnsPromises = dns.promises;
3 const tls = require('tls');
4 const fs = require('fs');
5 const cheerio = require('cheerio');
6
7
8 (async () => {
9     try {
10         const host = 'www.pxxxhub.com';
11         const res = await dnsPromises.lookup(host, { family: 4, hints: dns.ADDRCONFIG | dns.V4MAPPED });
12
13         console.log(res);
14         socket = tls.connect({
15             host: res.address,
16             port: 443,
17             cert: fs.readFileSync('public-cert.pem'),
18             rejectUnauthorized: false,
19         }, () => {
20             console.log("connect");
21         });
22
23         let response = "";
24         let first = true;
25         socket.setEncoding('utf8');
26         socket.on('data', (data) => {
27             if (first) {
28                 first = false;
29                 return;
30             }
31             response += data;
32         });
33         socket.on('end', () => {
34             socket.end();
35
36             let html = response;
37             let $ = cheerio.load(html.trim());
38             let $pcList = $('#mostRecentVideosSection').find('li.pcVideoListItem');
39
40             $pcList.each(function(idx, vid) {
41                 const te = $(vid).find('span.title a');
42                 const user = $(vid).find('div.usernameWrap a');
43
44                 const duration = $(vid).find('var.duration').text().trim();
45                 const url = "https://" + host + te.attr('href').trim();
46                 const title = te.text().trim();
47
48                 const userName = user.text().trim();
49                 const userHref = "https://" + host + user.attr('href')?.trim();
50
51                 console.log("");
52                 console.log("Title      :", title);
53                 console.log("Duration   :", duration);
54                 console.log("Video Url  :", url);
55                 console.log("User Name  :", userName);
56                 console.log("User Url   :", userHref);
57                 console.log("");
58             });
59         });
60
61         let req = "";
62         req += "GET / HTTP/1.1\n";

```



```

63     req += "User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)";
64     req += `Host: ${host}\n`;
65     req += "\n"
66     socket.write(req);
67 } catch (err) {
68     console.error(err);
69 }
70 })()

```

```

Title      : 网红熊猫TV<超人气
Duration   : 1:43
Video Url  : https://www.pornhu
User Name  : meetjsc
User Url   : https://www.pornhu

Title      : StepmomWithBoys -
Duration   : 12:39
Video Url  : https://www.pornhu
User Name  : Stepmom With Boys
User Url   : https://www.pornhu

Title      : BABYFACE (1975) mo
Duration   : 5:38
Video Url  : https://www.pornhu
User Name  :
User Url   : https://www.pornhu

Title      : Sweet Japanese hot
Duration   : 8:08
Video Url  : https://www.pornhu
User Name  : Nippon Hairy
User Url   : https://www.pornhu

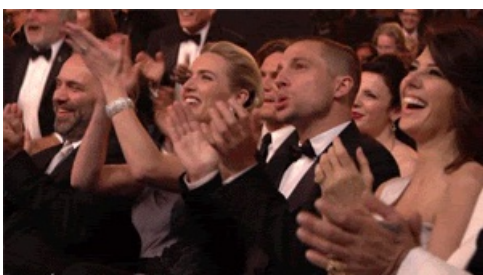
Title      : The teacher depriv
Duration   : 13:28
Video Url  : https://www.pornhu
User Name  : Belleniko
User Url   : https://www.pornhu


Title      : taboo threesome wi
Duration   : 11:25
Video Url  : https://www.pornhu
User Name  : thapa9711
User Url   : https://www.pornhu


Title      : 18videoz - Lindsey
Duration   : 10:20
Video Url  : https://www.pornhu
User Name  : 18 Videoz
User Url   : https://www.pornhu


```

성공입니다.



 [윤님의 게시물 더보기 >](#)

 좋아요 8


 댓글 32


 공유 |  신고


댓글

등록순

최신순




댓글알림 

 윤 **작성자**


네트워크 관련 개발 회사에선, 와이어샤크(패킷 캡처 툴) 가지고 야x 사이트를 분석해도 일하는 것 처럼 보이는 것입니다.

2020.06.08. 18:57 [답글 쓰기](#)

 성빈


우와 1등이다

2020.06.08. 18:36 [답글 쓰기](#)

 윤 **작성자**


축하드립니다. 상품은 글을 읽게 해드립니다.

2020.06.08. 18:37 [답글 쓰기](#)

 a8M9uQ2WRV


감사합니다. 제 밤시간이 늘어나겠네요

2020.06.08. 18:41 [답글 쓰기](#)

 윤 **작성자**


평소 쓰던거나 쓰십송

2020.06.08. 18:42 [답글 쓰기](#)

 사과님


편다. 진짜 유용한 정보네요. 윤님

2020.06.08. 18:44 [답글 쓰기](#)

 윤 **작성자**


유용하긴요. 그냥 각 이론에 대한 설명을 흥미로운 주제로 했을 뿐이지, 실용성 0입니다

2020.06.08. 18:46 [답글 쓰기](#)

 도미 doami2005

이 글을 한국 정부가 싫어합니다.


2020.06.08. 18:45 [답글 쓰기](#)

 윤 **작성자**

저도 한국 정부 ㅅ... ㅅ.... 사랑해요!


2020.06.08. 18:46 [답글 쓰기](#)

삭제된 댓글입니다.

 윤 **작성자**


감사합니다

2020.06.08. 19:22 [답글 쓰기](#)

 williamcom

모든지 위험합니다. 우회 되도록 안하시는게...

2020.06.08. 19:27 [답글 쓰기](#)

 윤 **작성자**

무엇 때문에 위험한 건가요?

2020.06.08. 19:30 [답글 쓰기](#)

 한국인



저도 같은 신사로서 글이 매우 유용했습니다. 감사합니다

2020.06.08. 19:55 답글 쓰기



윤 작성자

이게 유용하다니, 고수시군요 전 쓸모도 없는 것 같아 보입니다

2020.06.08. 20:10 답글 쓰기



반호BanHo

(감탄사)

2020.06.08. 19:59 답글 쓰기



윤 작성자

(꾸벅)

2020.06.08. 20:10 답글 쓰기



재승

ㅏ? 제가 뭘 본거ㄷ

2020.06.08. 21:32 답글 쓰기



윤 작성자

의미없는 빨글이었습니다

2020.06.08. 21:52 답글 쓰기



재승

윤 어어? 보면 안되는 것을 본거 같음음

2020.06.08. 21:52 답글 쓰기



줄려

적당히 응용해서 깃허브 뷰어를 만들어 사용하면 되는건가요

2020.06.08. 21:56 답글 쓰기



윤 작성자

그러려면 영상 스트리밍 기술이 가미되어야 하겠네요

2020.06.08. 21:56 답글 쓰기



terror

cheerio 모듈이 뭔가요?

2020.06.09. 00:35 답글 쓰기



윤 작성자

html을 파싱할 수 있는 npm 모듈입니다. 원랜 html 받아오는 것 까지만 하려고 했지만 파싱하는 걸 뒤늦게 추가하여 설명이 부족했습니다

2020.06.09. 08:17 답글 쓰기



네블링

node로 가져오는법은 알겠는데.. 카톡봇을 어떻게 node로 돌리죠..

2020.06.09. 08:37 답글 쓰기



도미 doami2005

termux 등

2020.06.09. 10:14 답글 쓰기

삭제된 댓글입니다.



윤 작성자

가능!

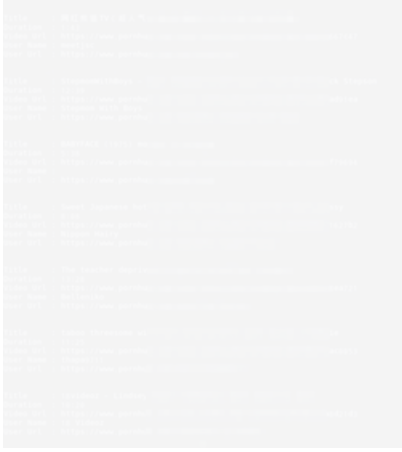
2020.06.09. 11:41 답글 쓰기



야웅

```
try {  
  const host = 'www.pxxxhub.com';  
  const res = await dnsPromises.lookup(host, { family: 4, hints: dns.ADDRCONFIG | dns.V4MAPPED })  
  
  console.log(res);  
}
```

//어라 알겠다(?)



2020.06.10. 16:09 답글 쓰기



윤 작성자

똑똑하시군요

2020.06.10. 16:11 답글 쓰기



아웅

윤 와 칭찬바다따(?)

2020.06.10. 23:36 답글 쓰기



하프

이제보니까 이분 개발자 윤군님이시구나... 어디서 봤던 프로필인가 했더니

2020.06.13. 21:33 답글 쓰기



윤 작성자

저 누군지 아심?

2020.06.13. 22:40 답글 쓰기



하프

윤 아는건 아니고 C언어 모카페에서 지나가다 봤네요.
이상하게 지나가는 사람 프사랑 이름을 기억하는 습관이 있어서;;ㅋㅋ

2020.06.13. 22:42 답글 쓰기

Hibot

댓글을 남겨보세요



등록

글쓰기

답글

목록

▲ TOP

조금 간단한 러시안룰렛 🎲 [3]	terror	2020.06.09.
디코봇 성공! 🤖 [6]	BCode	2020.06.09.
[스압] 정부에서 차단한 사이트 파싱하기 ~ 신사의 고집 ~ 🤖 [32]	윤	2020.06.08.
만들고 있는것 🤖 [13]	SP	2020.06.08.
수능 디데이 🤖 [13]	성빈	2020.06.08.

이 카페 인기글

안녕하십니까.

JSR
♡0 💬8

카봇커 채팅방을 그룹으로 만들면 오실분?

성빈
♡1 💬52

카페 오픈챗 인원제한 늘렸어요

성빈
♡0 💬7

이발

키알봇
♡0 💬5

이사람은 나오세여

폰수리가
♡0 💬19



아키네이터

기본 개념에 대한 강좌 | 조건문과 반복문

OtakoidTony
♡0 💬6

원주율 100자리 가져오기

T봇
♡0 💬6

기기 성능

은댕이
♡0 💬2