# 네트워크 프로그래밍

## 웹(Node.js) 기반의 소켓통신프로그래밍

- 2018-11-13
- 김 태 완

# Node.js?

https://ko.wikipedia.org/wiki/Node.js

- 구글 크롬의 V8 Engine JavaScript 엔진
- JavaScript Runtime
- 확장성 있는 네트워크 어플리케이션
- Server-Side 플랫폼
- 뛰어난 확장성(Node)
- 가볍고, 효율적
- 라이선스 (MIT)
- 멀티 플랫폼
  (Windows, Linux, MacOS , etc)

https://nodejs.org/ko/ https://velopert.com/node-js-tutorials

# Node.js?

https://ko.wikipedia.org/wiki/Node.js

- Speed (run on google JS Engine)
- Asynchronous I/O
- Non-Blocking I/O
- Data Streaming
  (HTTP request / response as a single event)
- Real-Time Application
  (client / server side)
- Easy / Fast 코딩
- Open-Source (NPM)

https://nodejs.org/ko/ https://velopert.com/node-js-tutorials

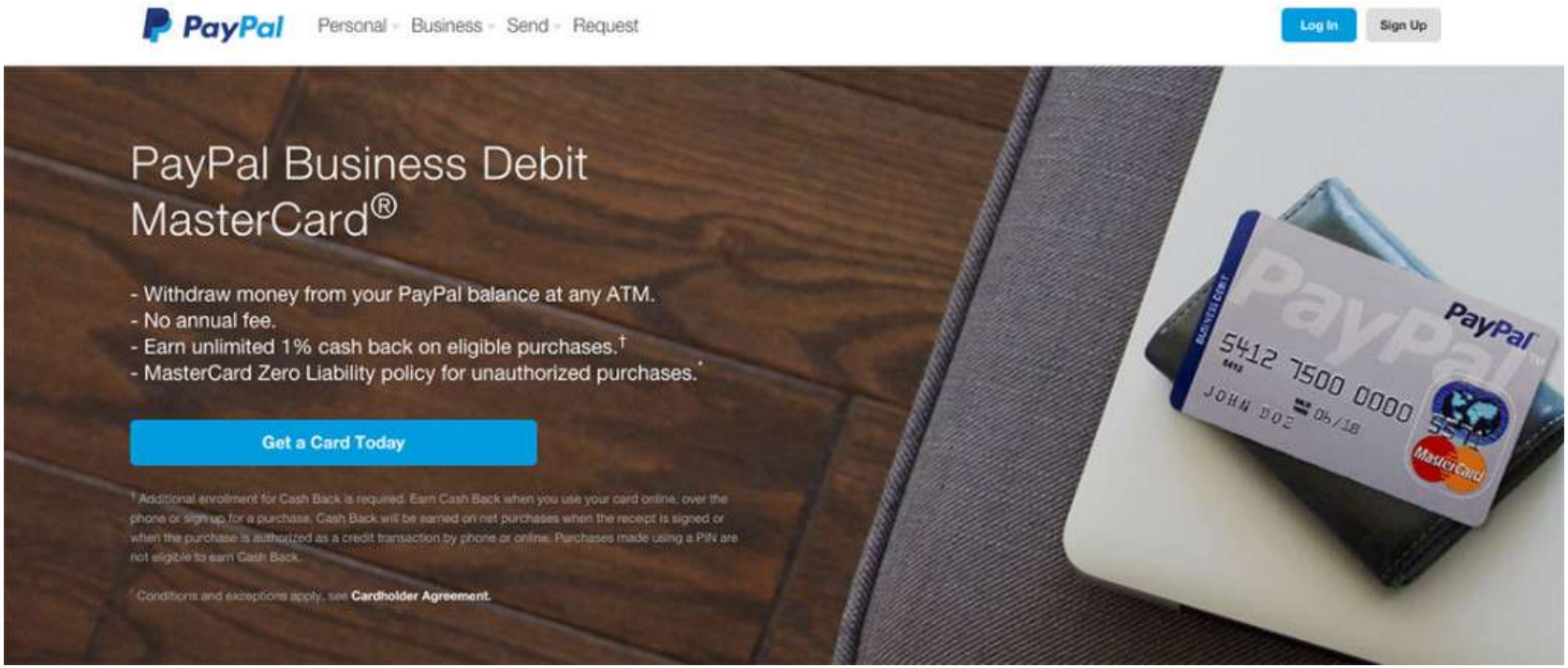
# Node.js?

https://ko.wikipedia.org/wiki/Node.js


https://nodejs.org/ko/ https://velopert.com/node-js-tutorials

# Node.js?

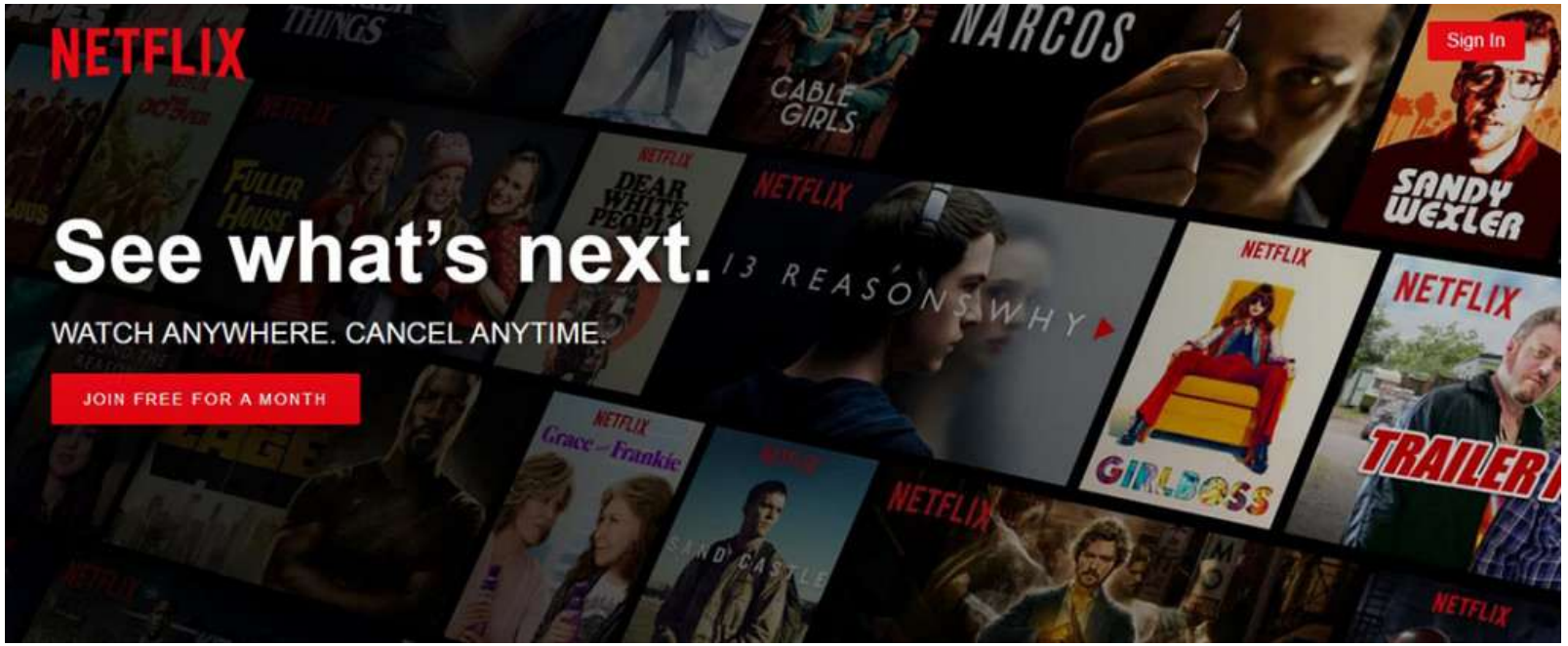
https://ko.wikipedia.org/wiki/Node.js




https://nodejs.org/ko/

# Node.js?

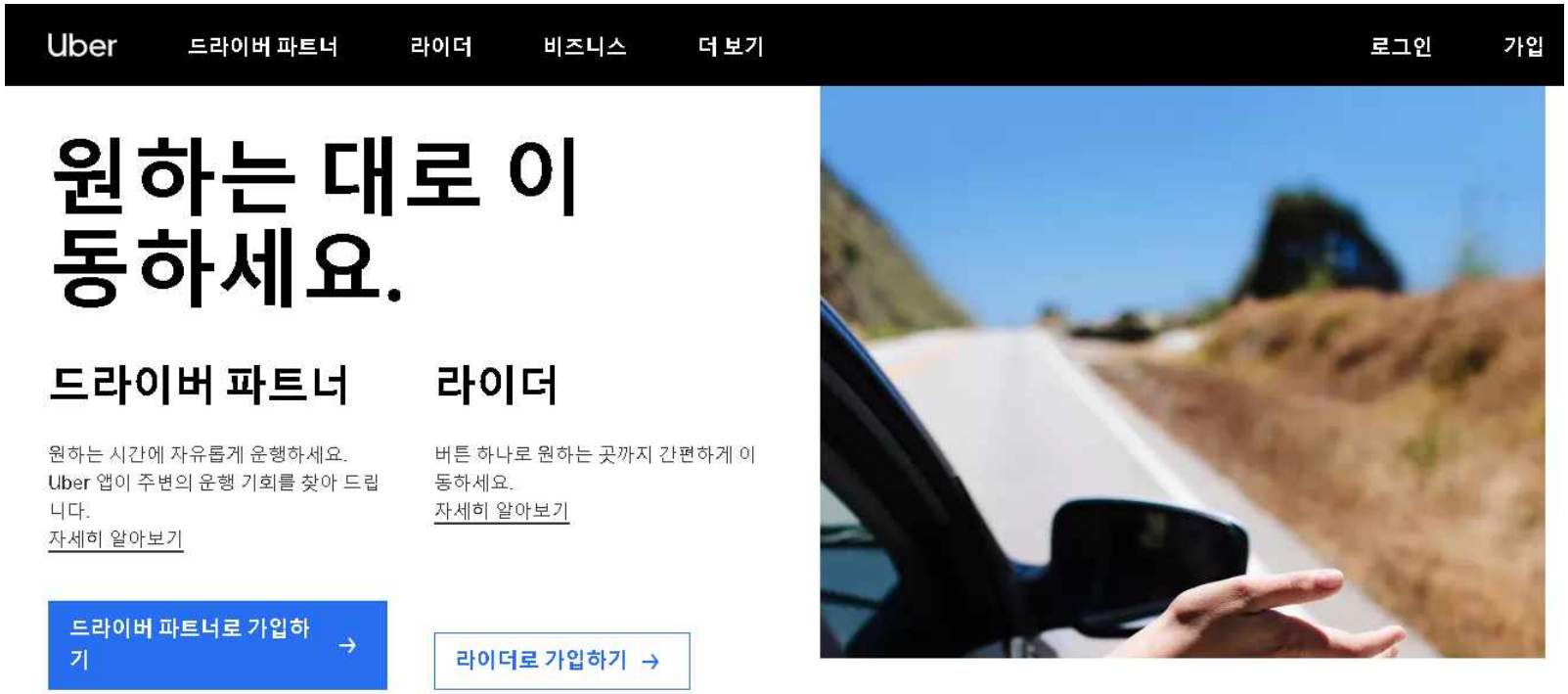
https://ko.wikipedia.org/wiki/Node.js




https://nodejs.org/ko/

# Node.js?

https://ko.wikipedia.org/wiki/Node.js




https://nodejs.org/ko/

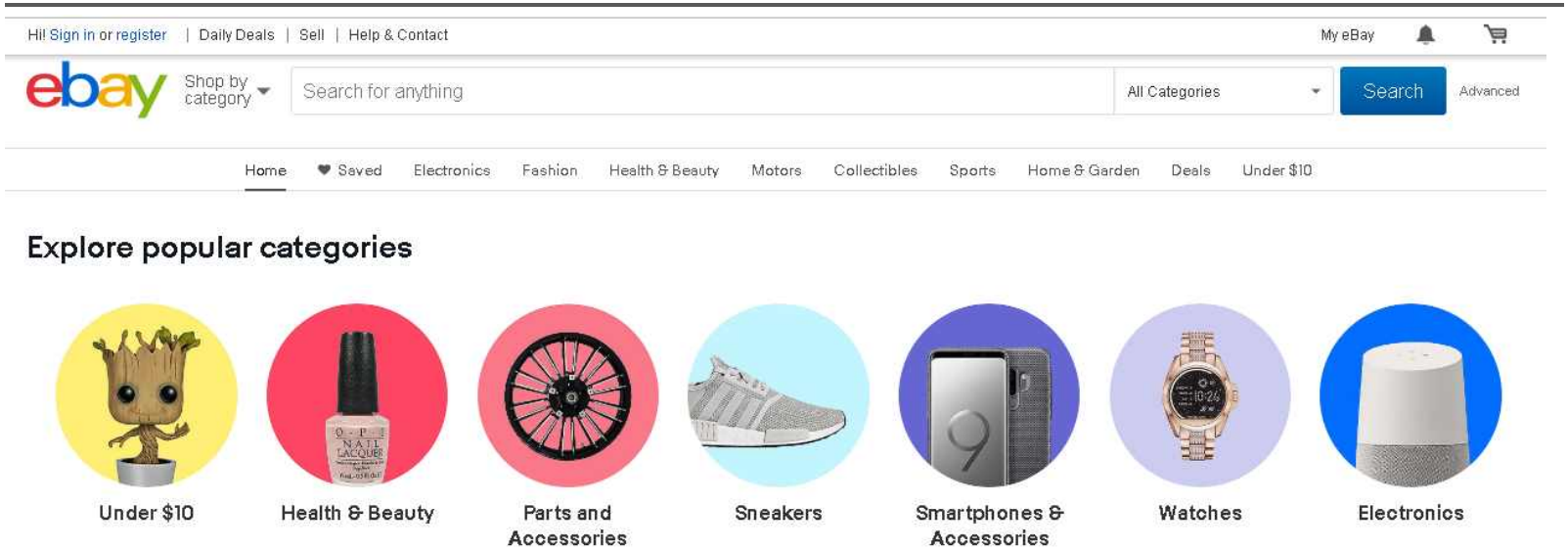
# Node.js?

https://ko.wikipedia.org/wiki/Node.js


https://nodejs.org/ko/

# Node.js?

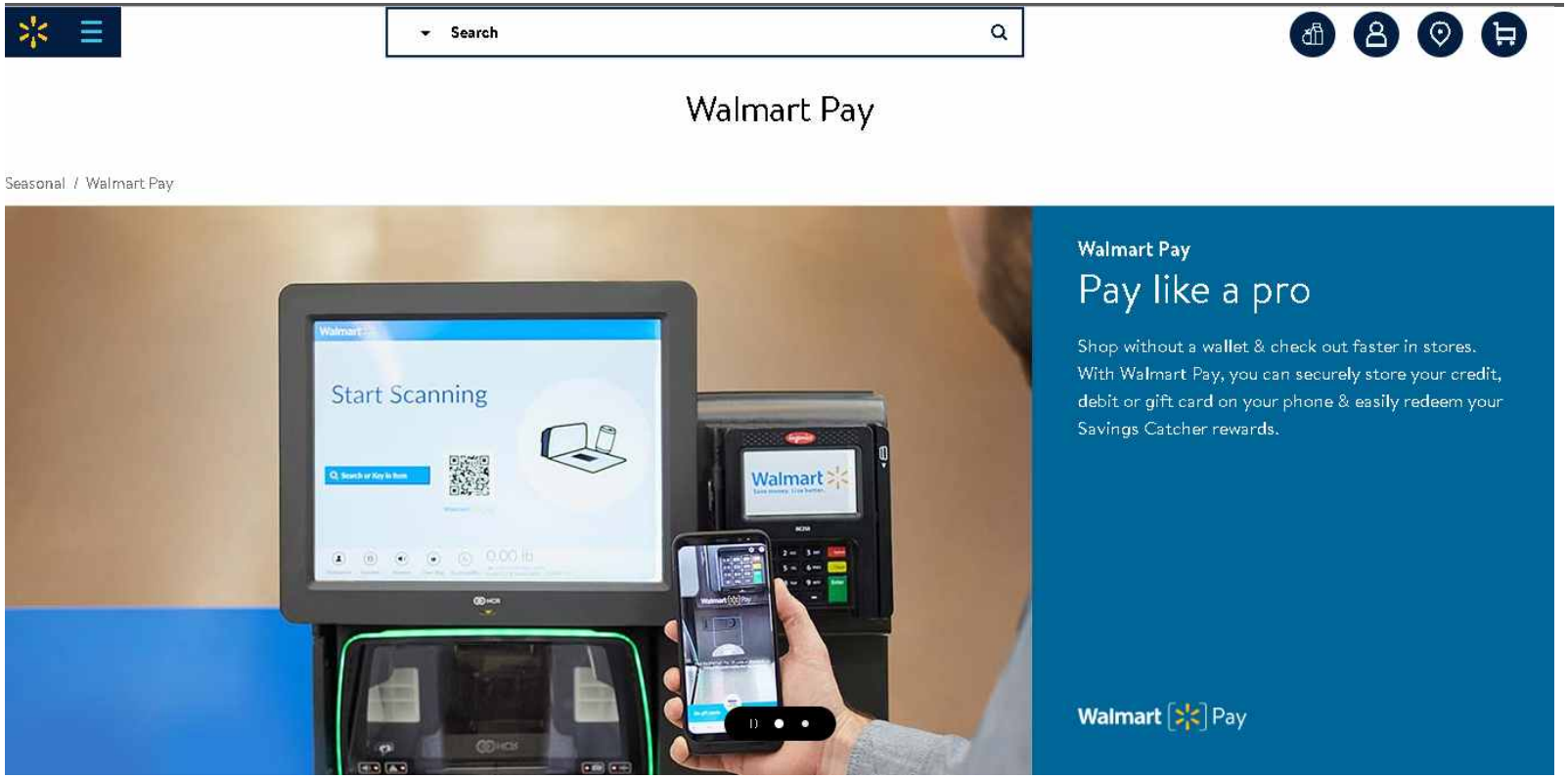
https://ko.wikipedia.org/wiki/Node.js




https://nodejs.org/ko/

# Node.js?

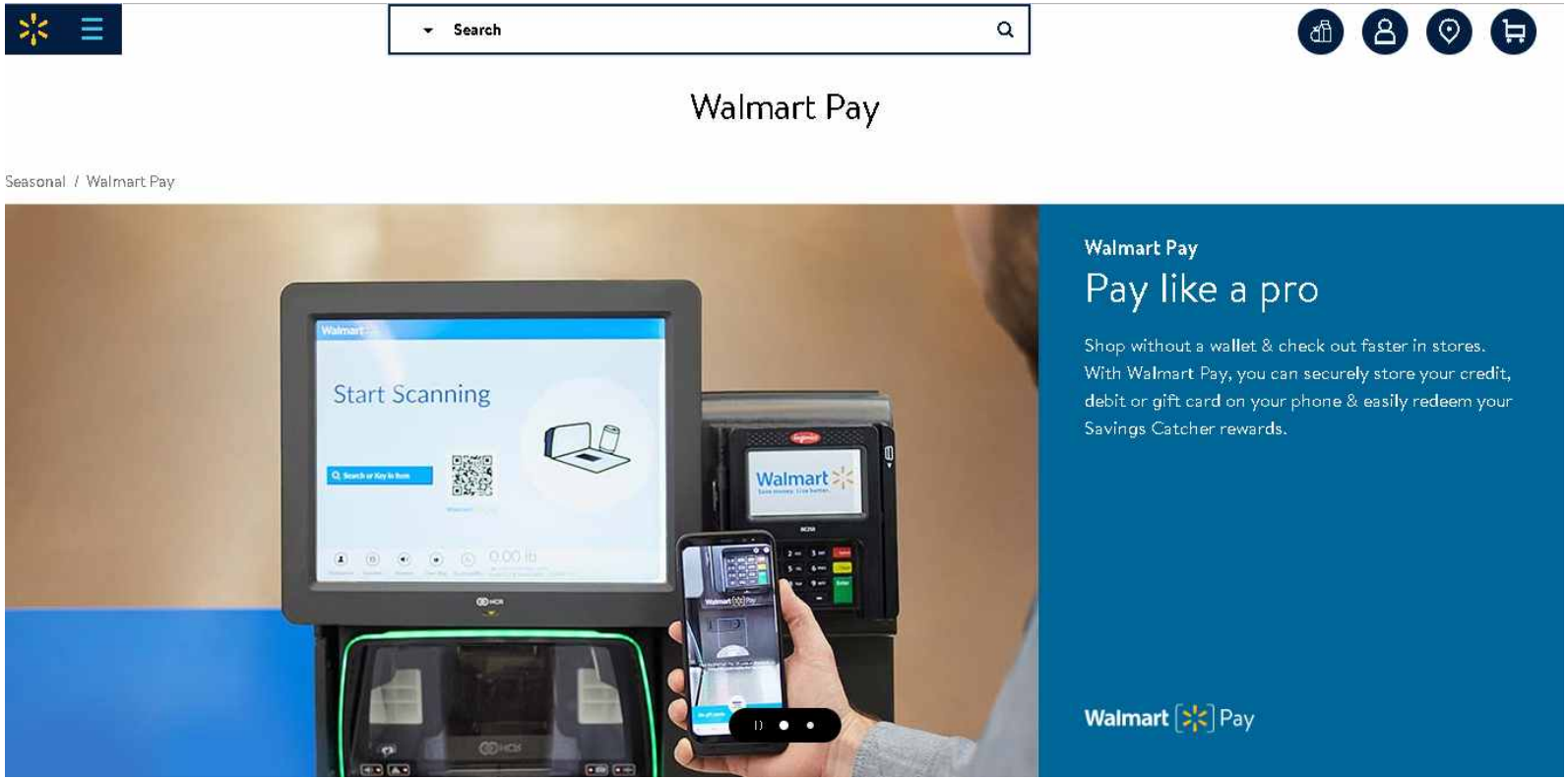
https://ko.wikipedia.org/wiki/Node.js




https://nodejs.org/ko/

# Node.js?

https://ko.wikipedia.org/wiki/Node.js




https://nodejs.org/ko/

# Node.js Open Source Library

https://www.npmjs.com/

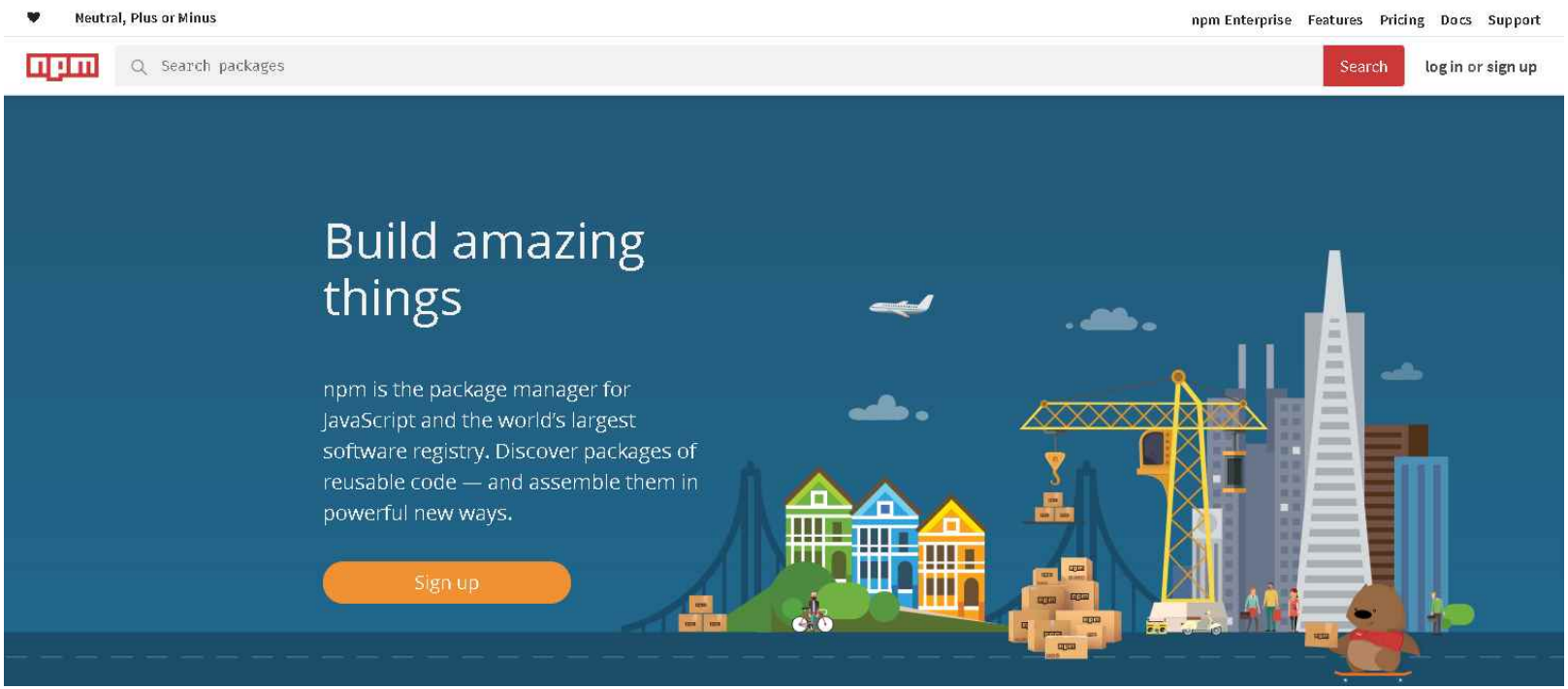

+


Open Source Library?

https://nodejs.org/ko/

# Node.js Open Source Library

https://www.npmjs.com/




https://nodejs.org/ko/

Nifty Pun Master

npm Enterprise Features Pricing Docs Support

npm socket Search log in or sign up

**4655 packages found**

1 2 3 … 233 »

Sort Packages

Optimal

Popularity

Quality

Maintenance

**Who's Hiring?**

**Triplebyte, MuleSoft, Callisto** and lots of other companies are hiring javascript developers.

See all 19 companies

### socket

Socket is a connect clone for simple socket based applications

mountainmansoftware published 0.0.1 • 6 years ago

### socket.io

Node.js realtime framework server

realtime framework websocket tcp events socket io

darrachequesne published 2.1.1 • 6 months ago

### ws

Simple to use, blazing fast and thoroughly tested websocket client and server for Node.js

HyBi Push RFC-6455 WebSocket WebSockets real-time

lpinca published 6.1.0 • a month ago

### socket.io-client

[![Build Status](https://secure.travis-ci.org/socketio/socket.io-client.svg?branch=master)](http://travis-ci.org/socketio/socket.io-client) [![Dependency Status](https://david-dm.org/socketio/socket.io-client.svg)](https://david-dm.org/socketio/socket.io-

realtime framework websocket tcp events client

darrachequesne published 2.1.1 • 6 months ago

### sockjs-client

SockJS-client is a browser JavaScript library that provides a WebSocket-like object.

npm Search packages Search log in or sign up

Share your code. npm Orgs help your team discover, share, and reuse code. Create a free org »

**WS**

6.1.0 • Public • Published a month ago

Readme | 1 Dependencies | 4,450 Dependents | 99 Versions

# ws: a Node.js WebSocket library

npm v6.1.0 | build passing | build passing | coverage 100%

ws is a simple to use, blazing fast, and thoroughly tested WebSocket client and server implementation.

Passes the quite extensive Autobahn test suite: server, client.

**Note**: This module does not work in the browser. The client in the docs is a reference to a back end with the role of a client in the WebSocket communication. Browser clients must use the native `WebSocket` object. To make the same code work seamlessly on Node.js and the browser, you can use one of the many wrappers available on npm, like isomorphic-ws.

## Table of Contents



install

```
> npm i ws
```

weekly downloads

444,351

| version | license |
|---|---|
| 6.1.0 | MIT |

| open issues | pull requests |
|---|---|
| 13 | 1 |

| homepage | repository |
|---|---|
| github.com | github |

last publish

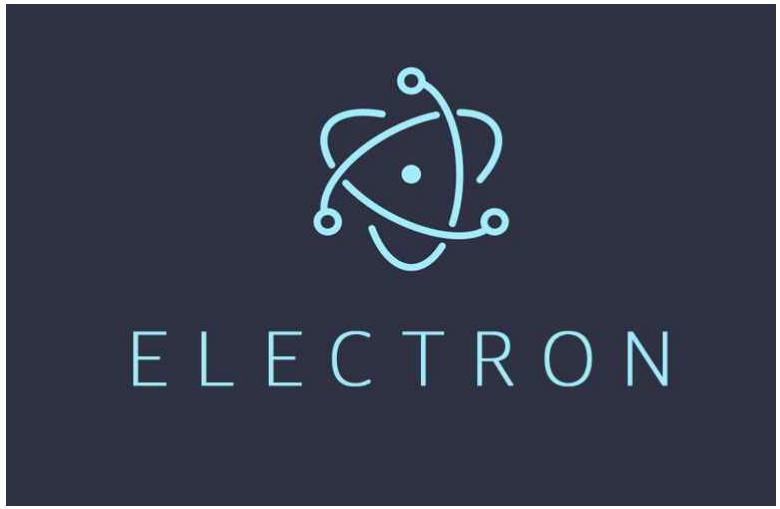
a month ago

collaborators

# Node.js?

ELECTRON

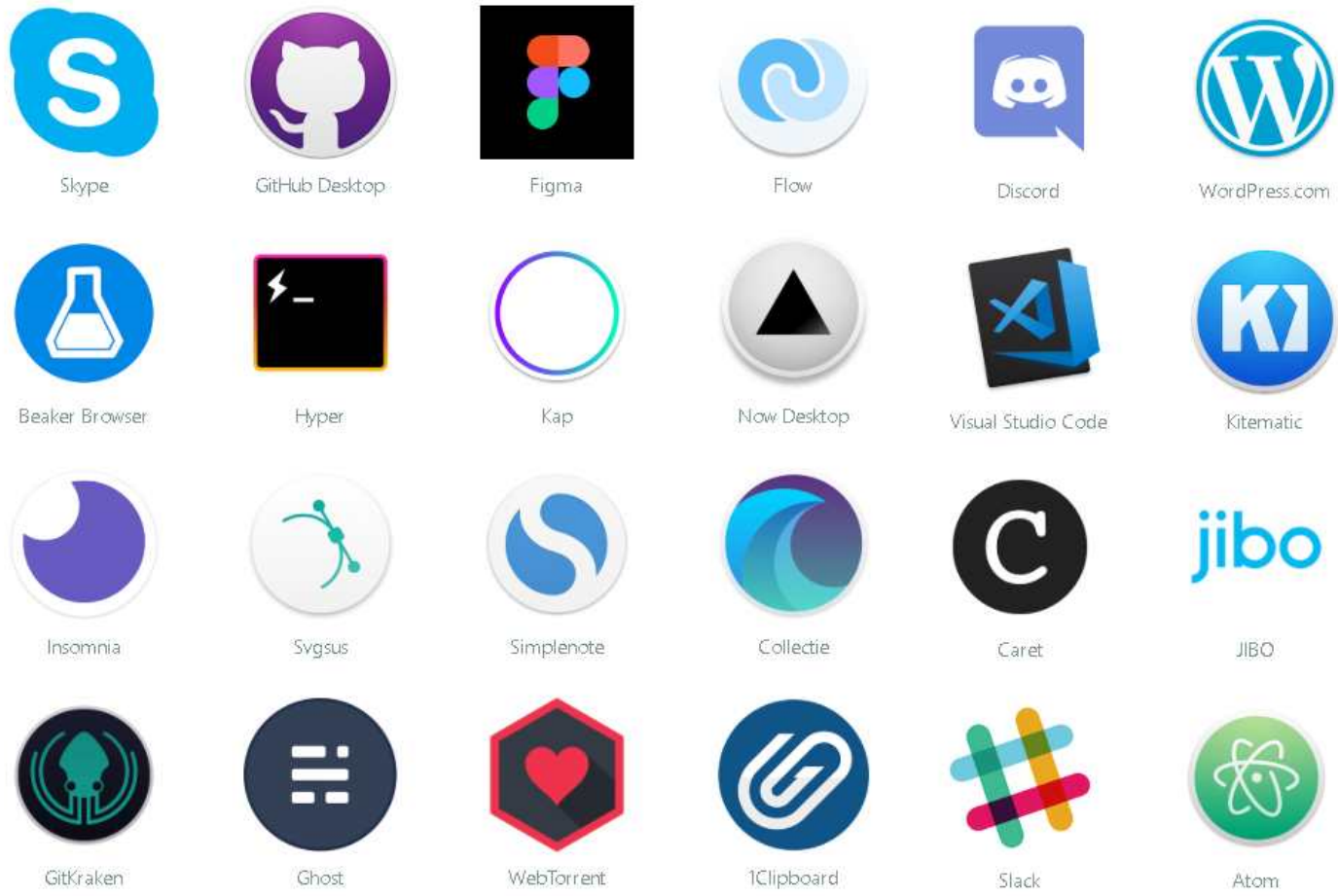

+



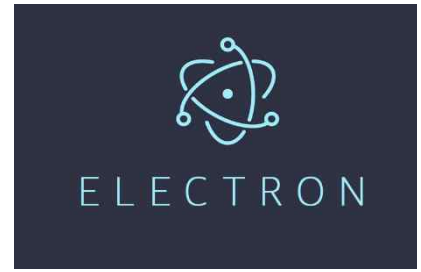

Desktop Application?

https://electronjs.org/
https://nodejs.org/ko/

# Node.js + Electron

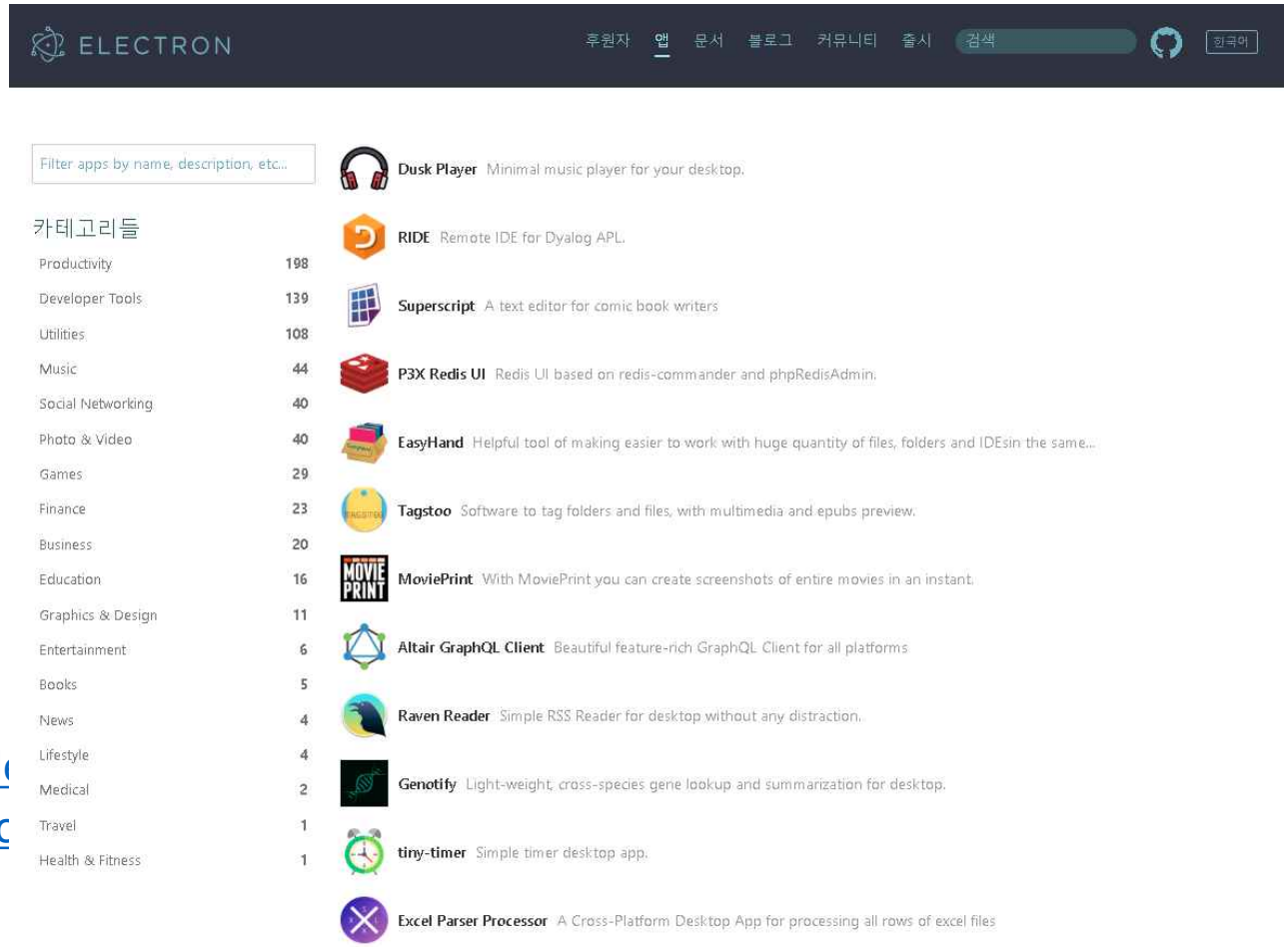



https
https




# Node.js + Electron

https://electronjs.org/apps




https://el

https://nc

# Node.js 준비물

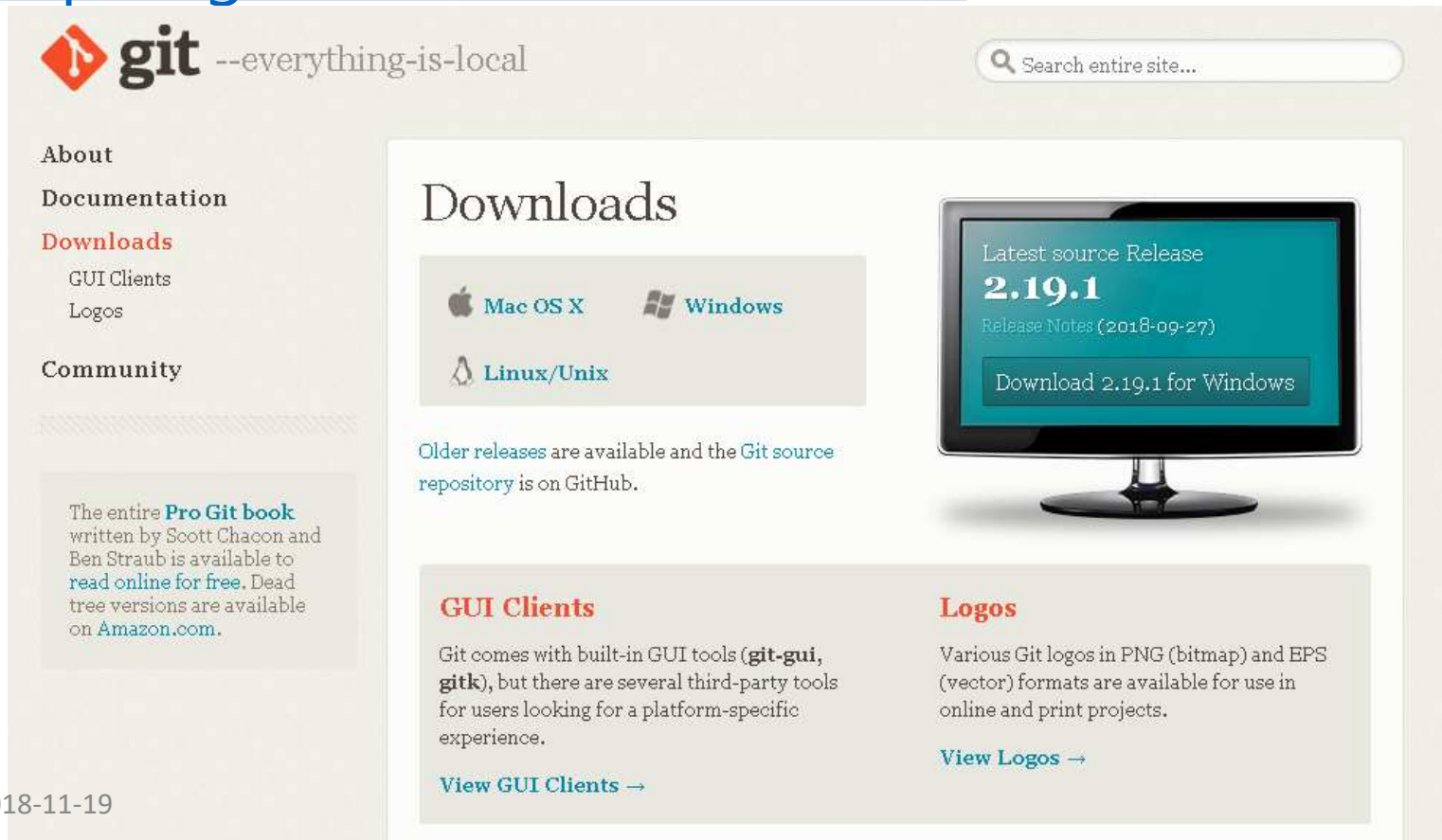
## 1. Node.js 설치하기

https://nodejs.org/ko/

# Node.js 준비물

## 2. git 설치하기

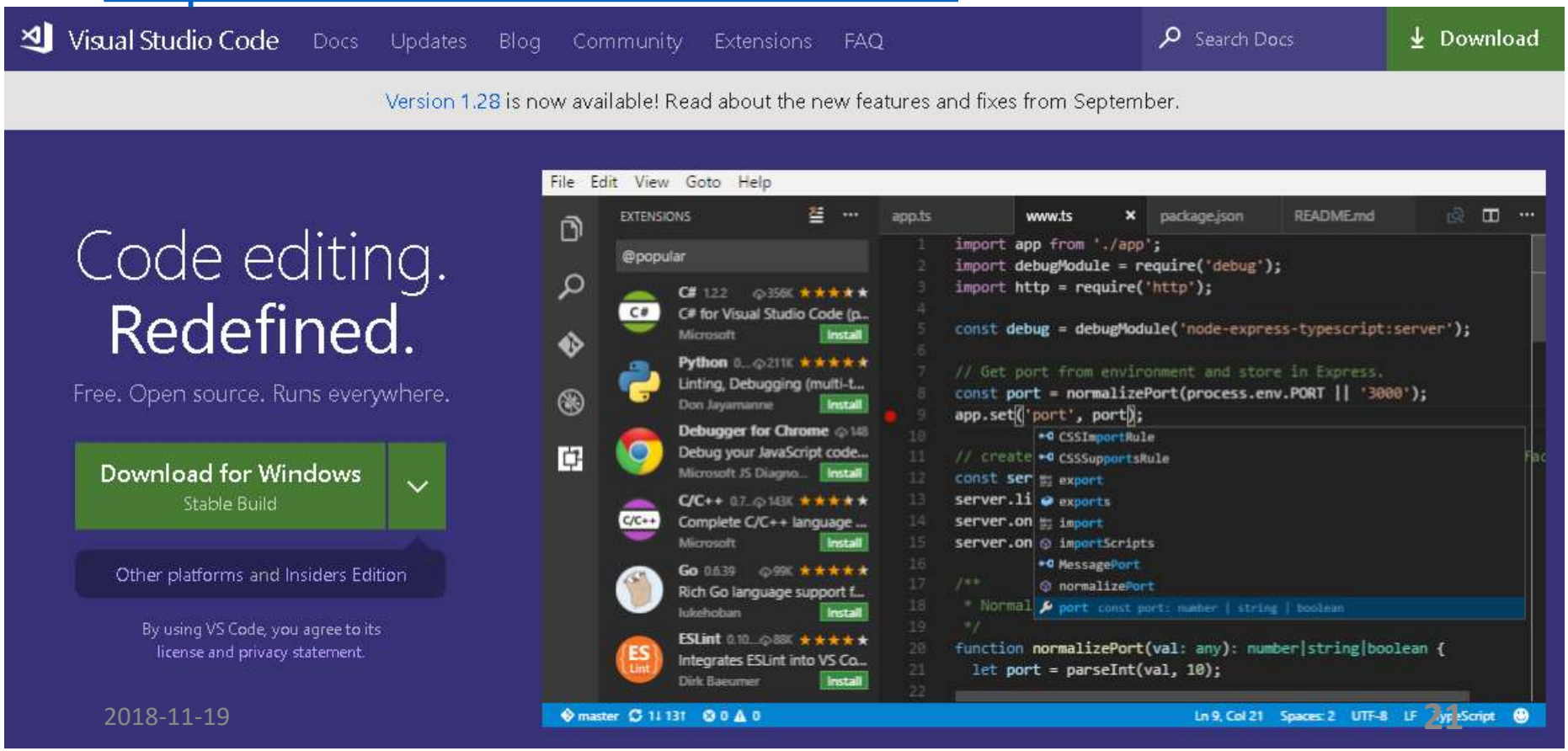
https://git-scm.com/downloads



# Node.js 준비물

## 3. Visual Studio Code 설치하기

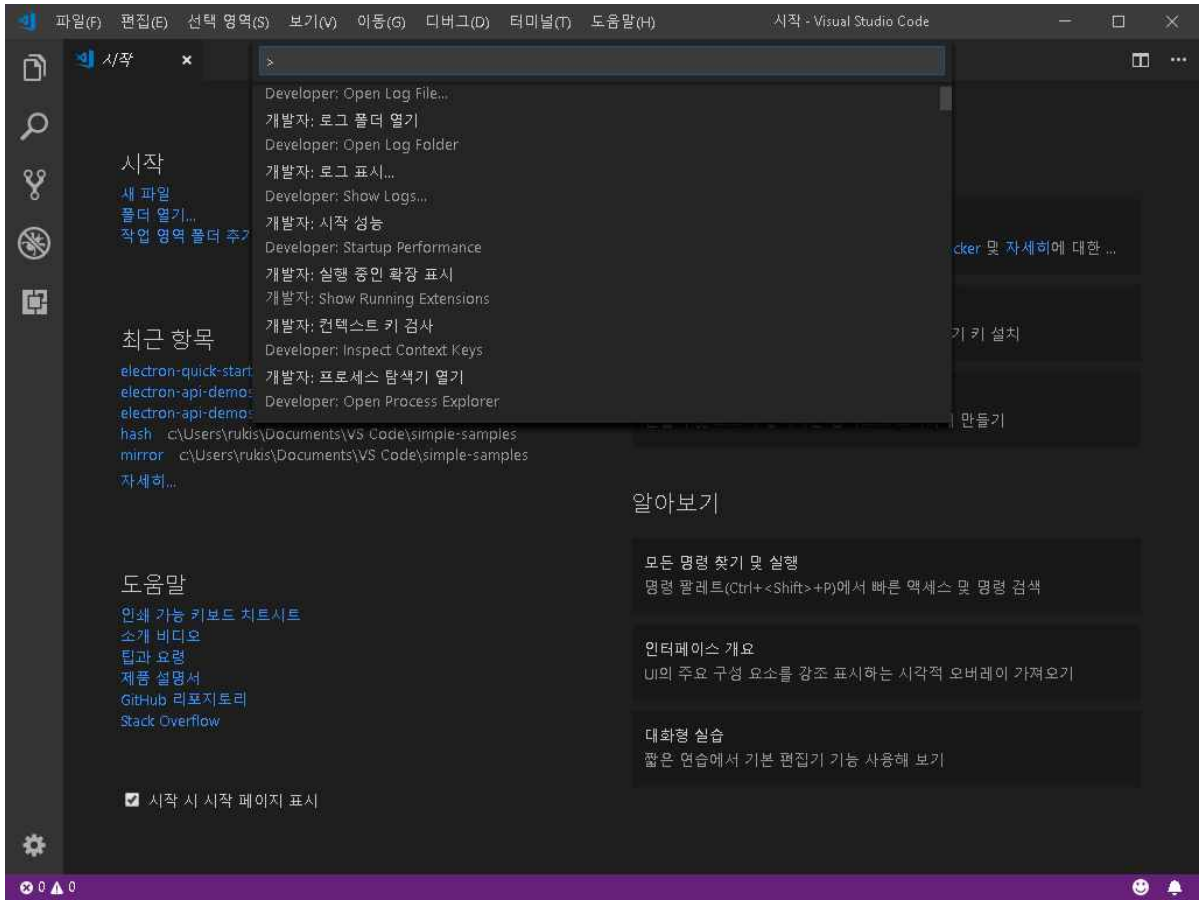
https://code.visualstudio.com/



# Node.js 시작하기

## Visual Studio Code

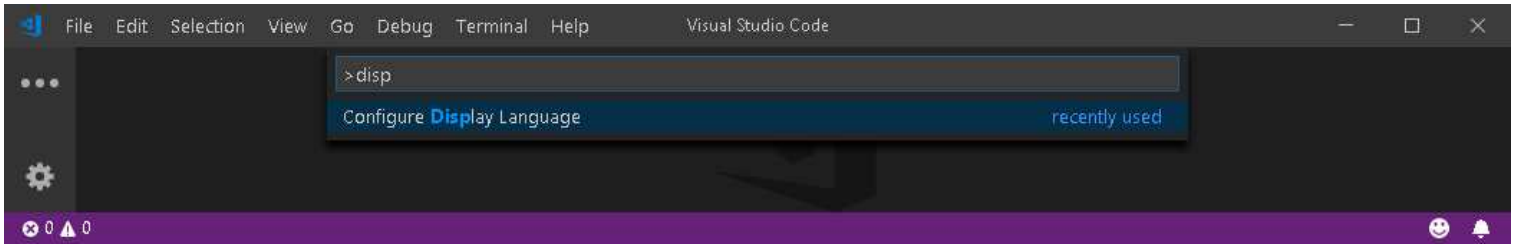
➢Ctrl + Shift + p / F1 (명령팔레트)

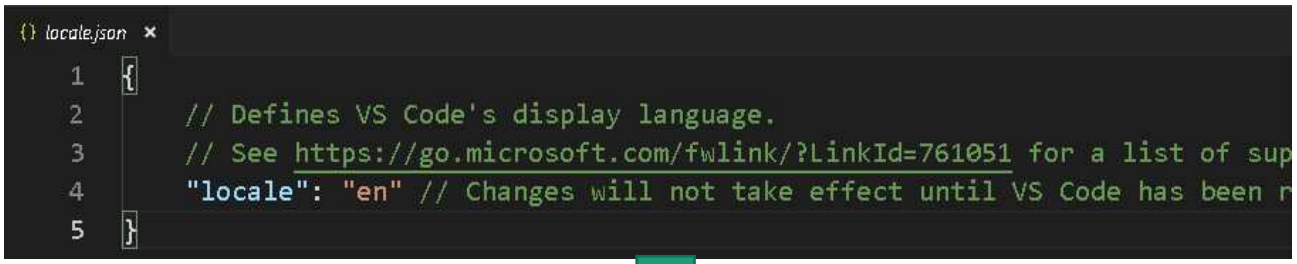


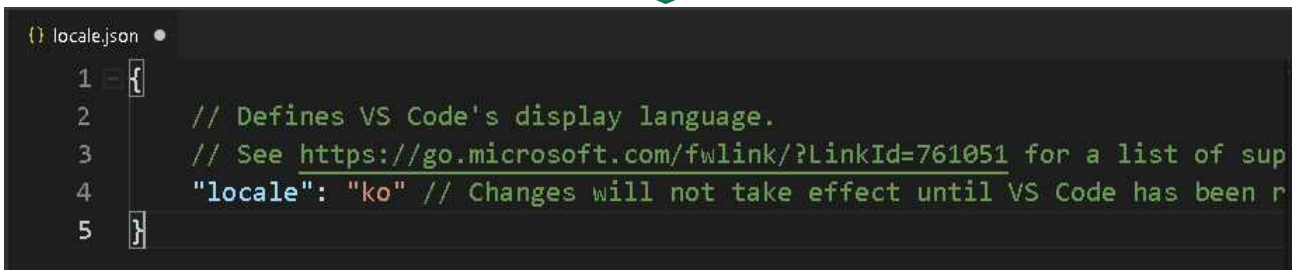
# Node.js 시작하기

## Visual Studio Code

➢ 한글화 -> F1 -> Configure Display Language



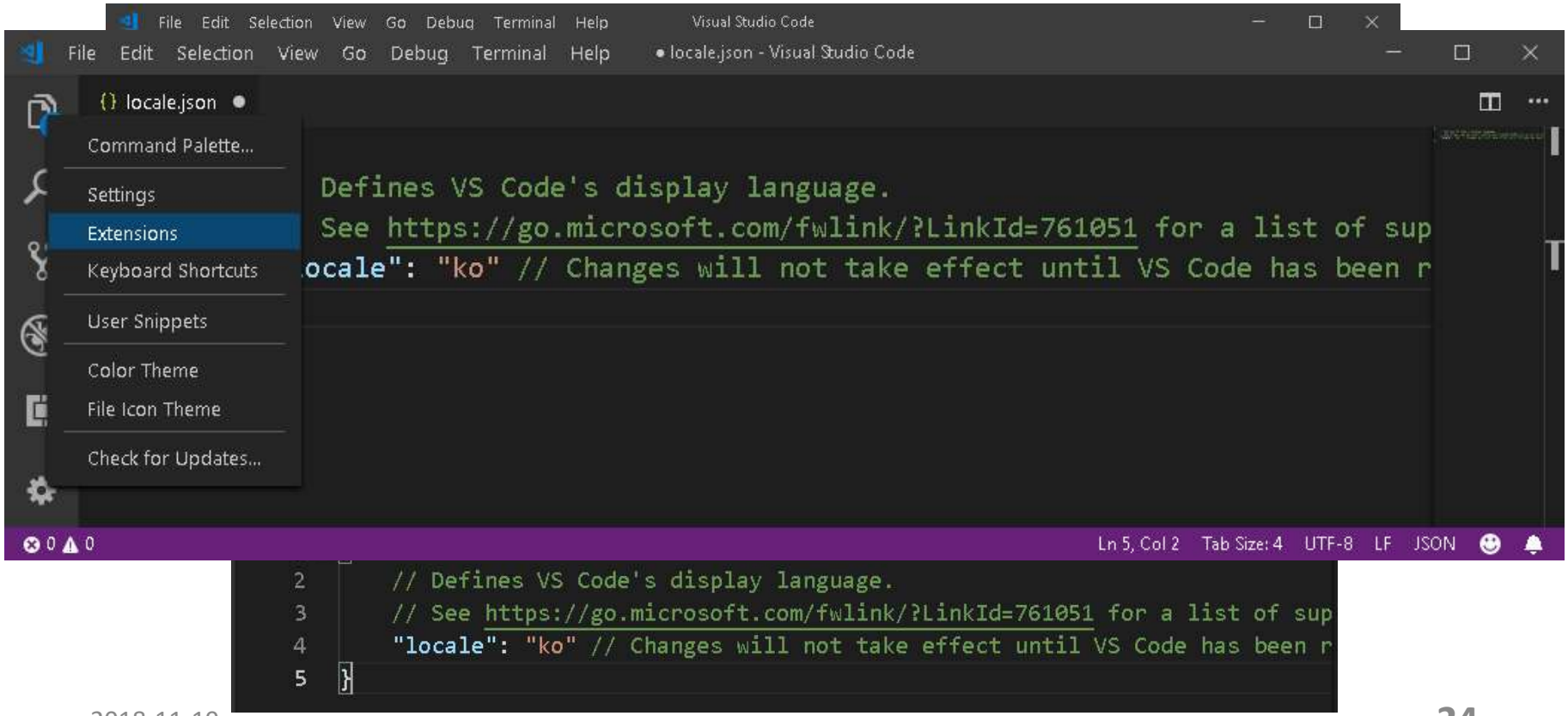



# Node.js 시작하기

## Visual Studio Code

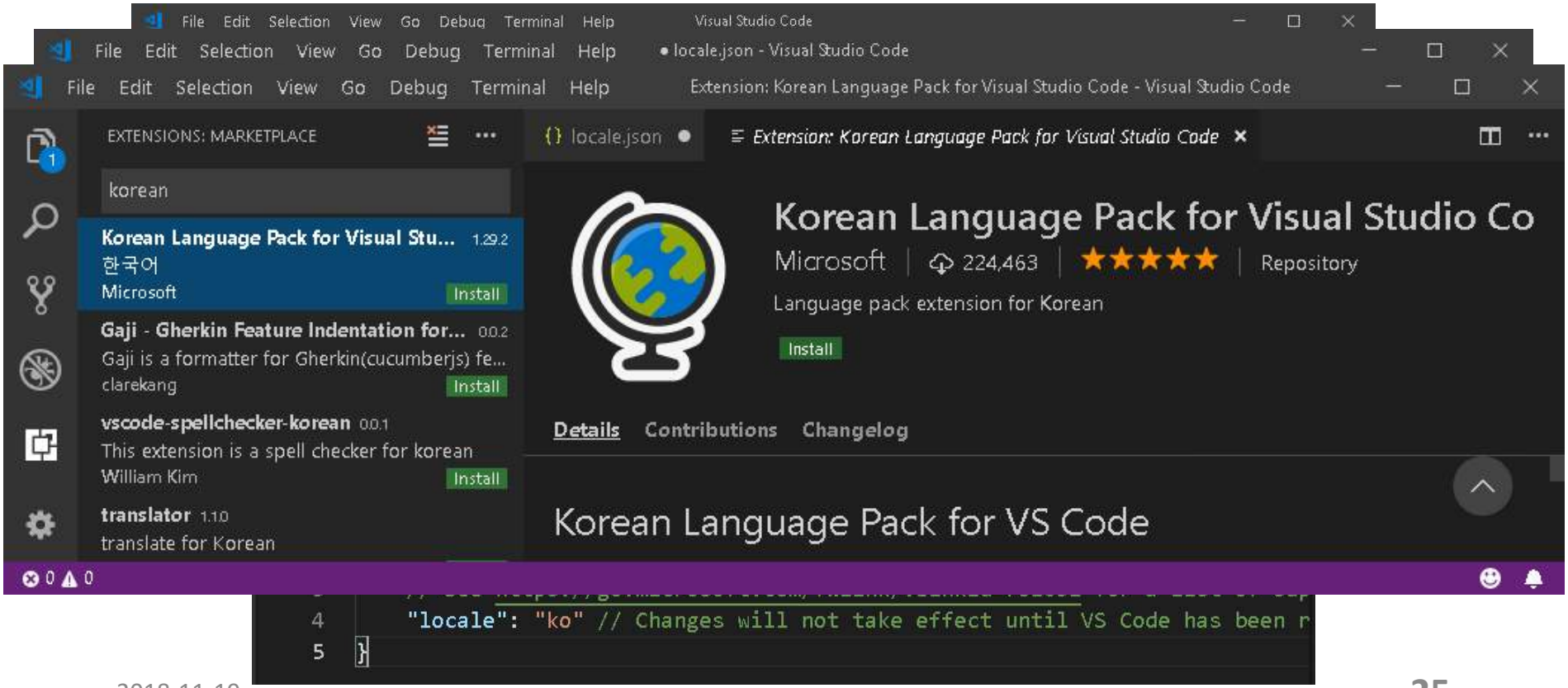
➢ 한글화 -> F1 -> Configure Display Language



# Node.js 시작하기

## Visual Studio Code

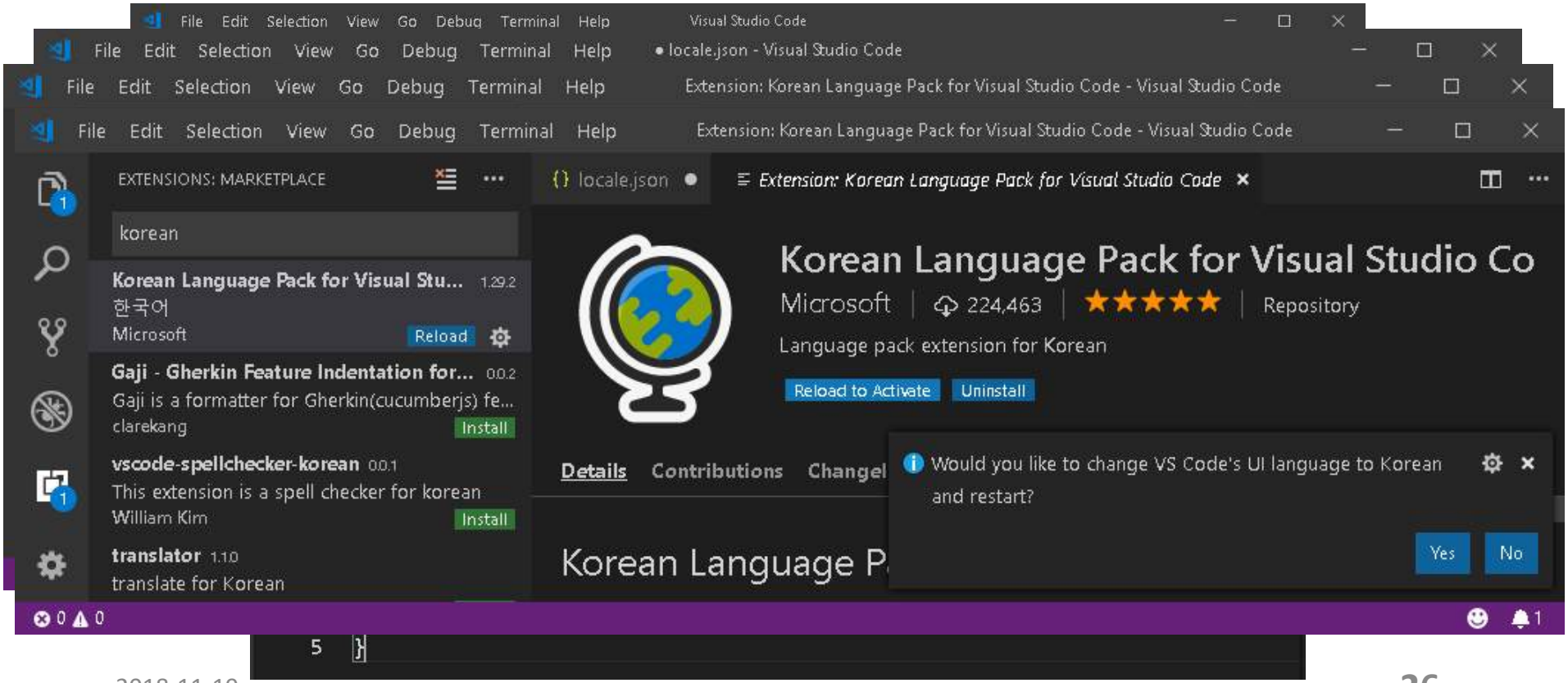
➢ 한글화 -> F1 -> Configure Display Language



# Node.js 시작하기

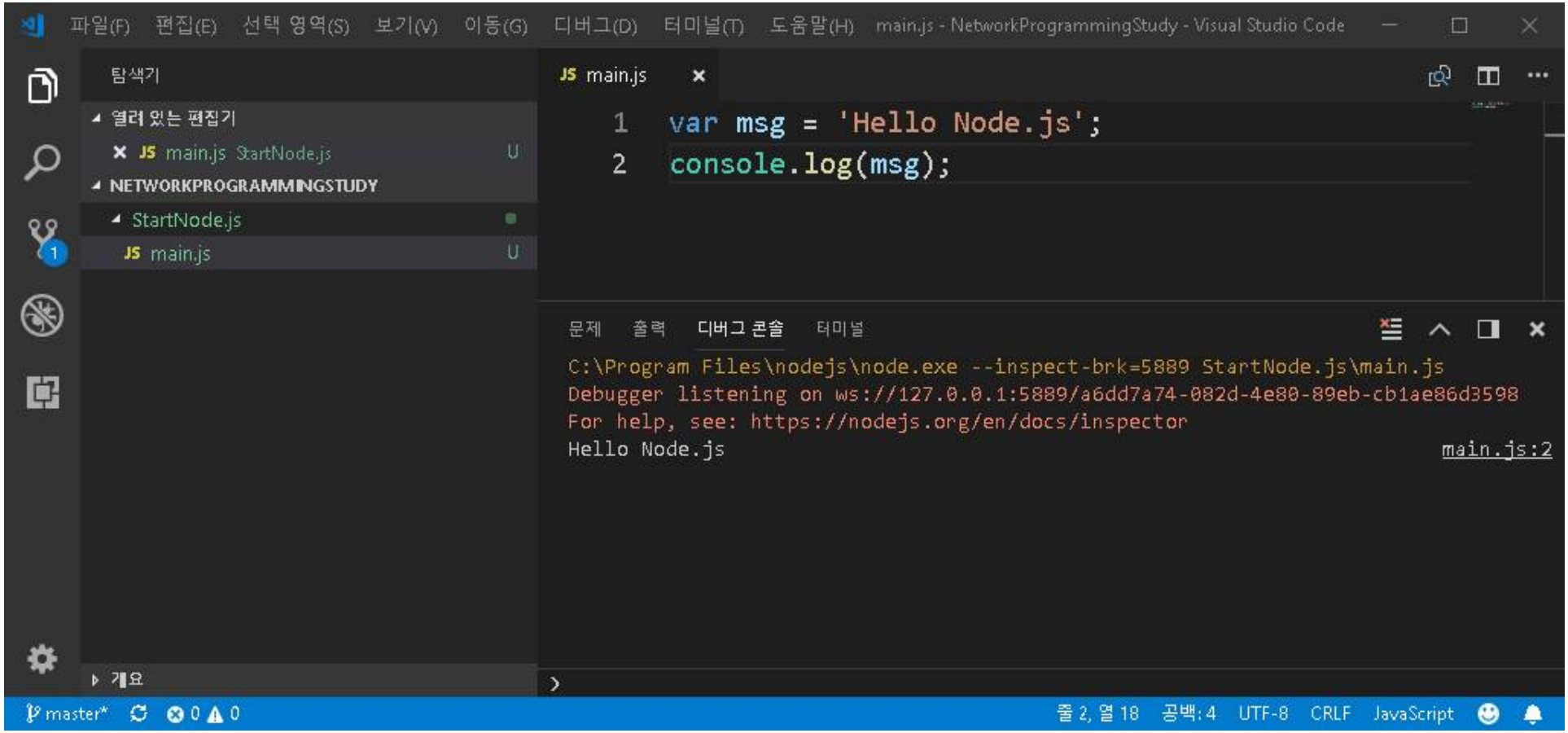
## Visual Studio Code

➢ 한글화 -> F1 -> Configure Display Language



# Node.js 시작하기

➢HELLO NODE JS



# Node.js 프로젝트 생성

- Express 사용하기
  - 파일 -> 폴더 열기 -> 선택
  - npm install -g express-generator

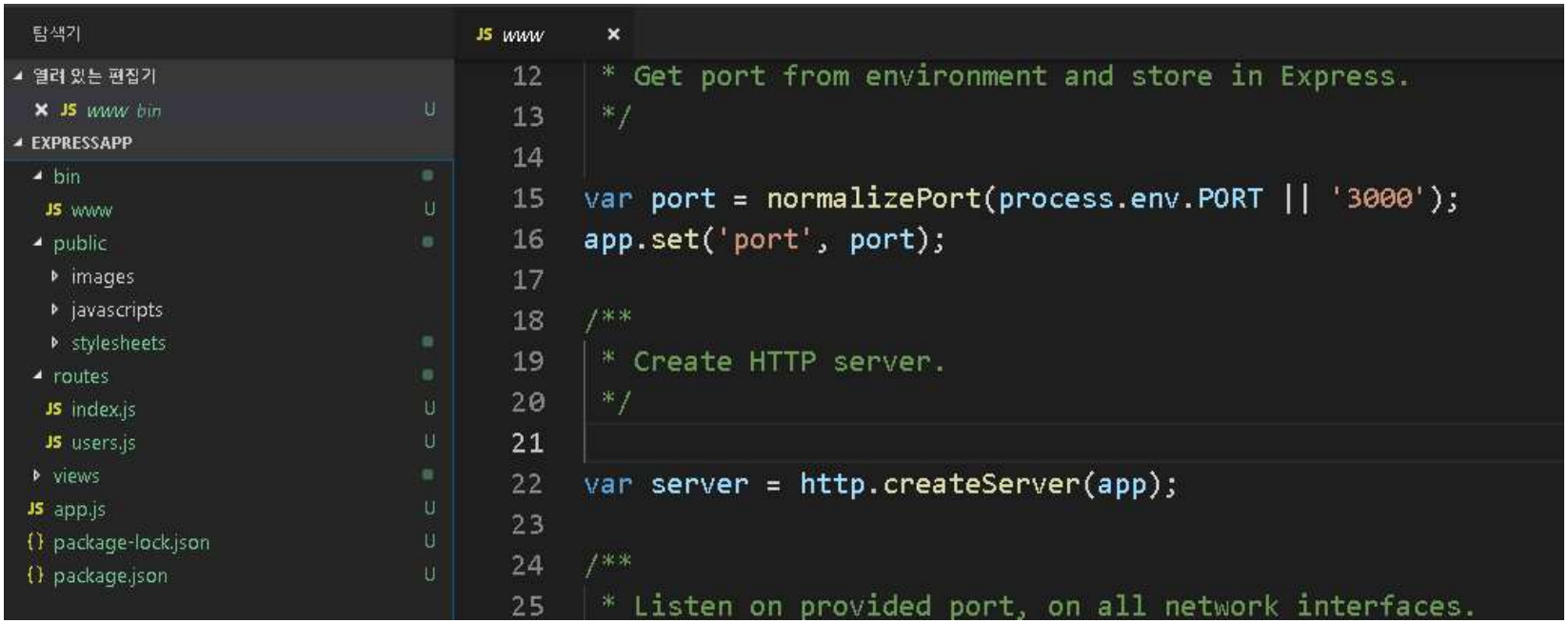
```
문제  출력  디버그 콘솔  터미널
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\rukis\Documents\NetworkProgrammingStudy> npm install -g express-generator
```

  - express -e ./

```
문제  출력  디버그 콘솔  터미널                                    1: powershell
PS C:\Users\rukis\Documents\NetworkProgrammingStudy> express -e ./프로젝트명 혹은 폴더명
```

# Node.js 프로젝트 생성

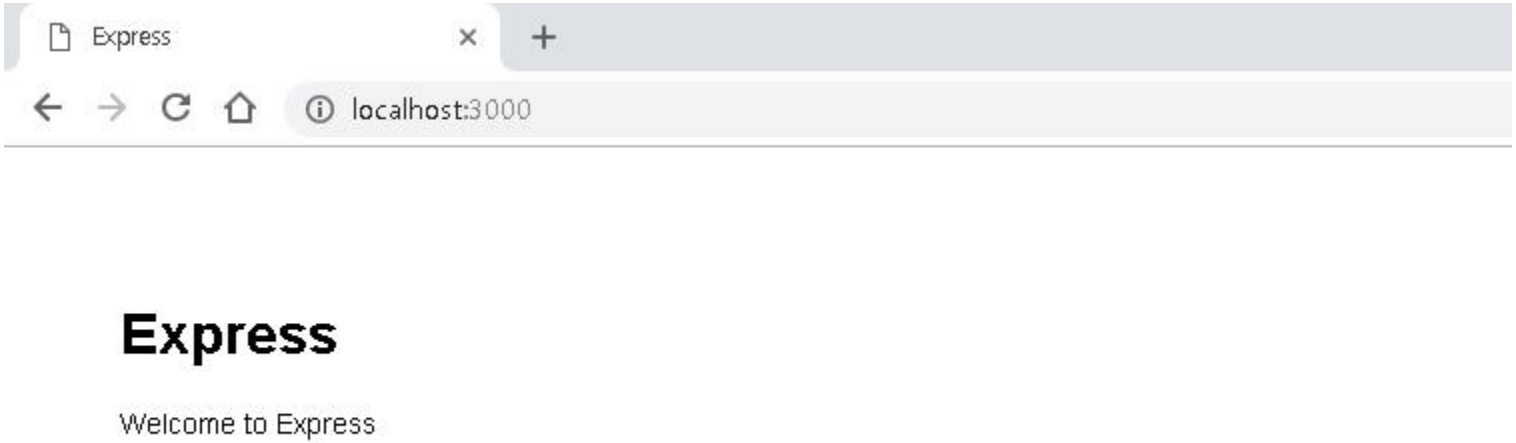
➢Express 프로젝트 생성




➢ F1 -> npm install or 터미널 -> npm install

# Node.js 시작하기

- Express 프로젝트 생성
- 빌드 -> F5, 웹 접속 (localhost:3000)



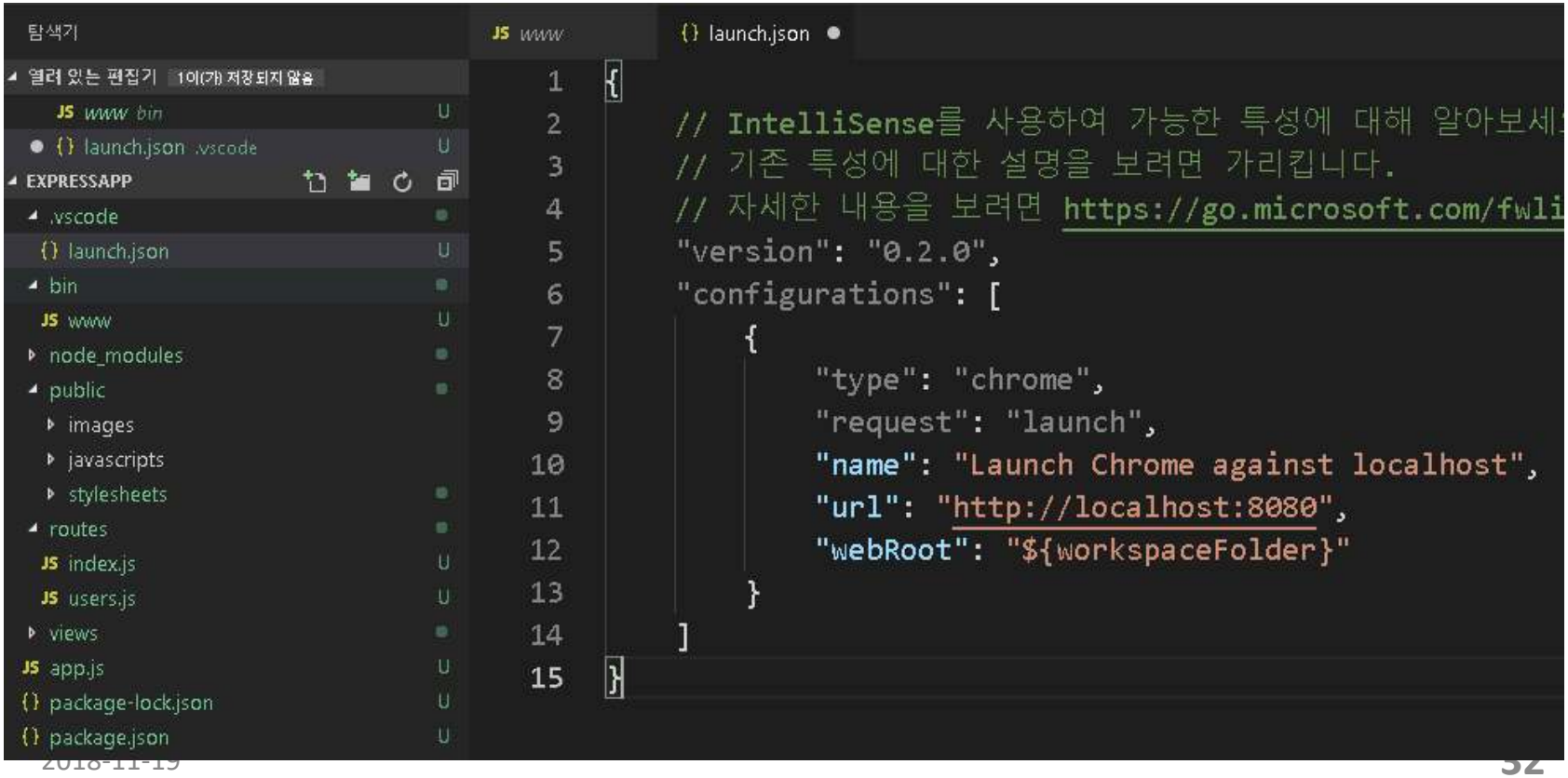
# Node.js 시작하기

- Express 프로젝트 (Debugging)
- Debugger for Chrome 확장 설치


- 빌드 > F5 > Chrome

# Node.js 시작하기

- Express 프로젝트 (Debugging)
- .vscode/launch.json 파일 자동생성



# Node.js 시작하기

➢Express 프로젝트 (Debugging)

launch.json - ExpressApp - Visual Studio Code

```
{
    // IntelliSense를 사용하여 가능한 특성에 대해 알아보세요.
    // 기존 특성에 대한 설명을 보려면 가리킵니다.
    // 자세한 내용을 보려면 https://go.microsoft.com/fwlink/?linkid=830387을(를) 방문하세요.
    "version": "0.2.0",
    "compounds": [
        {
            "name": "프로그램 시작",
            "configurations": ["Launch Program", "Launch Chrome"]
        }
    ],
    "configurations": [
        {
            "type": "chrome",
            "request": "launch",
            "name": "Launch Chrome",
            "url": "http://localhost:3000",
            "webRoot": "${workspaceFolder}"
        },
        {
            "type": "node",
            "request": "launch",
            "name": "Launch Program",
            "program": "${workspaceFolder}\\bin\\www"
        }
    ]
}
```

```
C:\Program Files\nodejs\node.exe bin\www
GET / 304 389.901 ms - -
GET /stylesheets/style.css 304 1.871 ms - -
GET /favicon.ico 404 16.503 ms - 1412
```

# Node.js + Express + Html

http://jade-lang.com/
https://ejs.co/

- Jade vs Ejs
  - Jade 의 경우 기능이 더 뛰어나다.
  - Ejs 의 경우 html 코드를 그대로 사용하여 이질감이 없다.
- html 파일 사용하기
  - views/index.ejs => views/index.html 파일명 수정
- app.js 코드 수정

```
// view engine setup
app.set('views', path.join(__dirname, 'views'));
/// ********** html 사용을 위한 ejs module 추가
App.engine('html', require('ejs').renderFile);
App.set('view engine', 'html');
```

# Node.js + Express + Html

- routes/index.js

https://www.w3schools.com/html/

https://ejs.co/

```
res.render('index', { title: 'test' });
```

- views/index.html

```
<html> <head>
<title><%=title%></title> </head>
<body>
<h1>This is a Heading</h1>
<p>This is a paragraph.</p>
</body> </html>
```

# Node.js 데이터 입력

https://www.w3schools.com/html/

➢routes/index.js

```
// POST
router.post('/', function (req, res) {
var returnVal = { result: '' };
if (req.body.msg) {
console.log("client message : " + req.body.msg);
returnVal.result= 'success';
}
else {
returnVal.result= 'failure';
}
res.json(returnVal);
});
```

# Node.js 데이터 입력

- javacripts/test.js

https://www.w3schools.com/html/
https://api.jquery.com/

```
function test()
{
    var data = { msg:
        $("#name").val()
    };
    console.log(data);
    $("#result").text(data.msg
    $.ajax({
    type: 'POST',
    url: '/',
    data: data,
    conntentType:
    'application/json',
    async: true,
    success: function (data) {
    var res = data;
    if (res.result == 'success') {
    console.log("success!!");
    } else if (res.result ==
    'failure') {
    console.log("failure!!");
    } else {
    alert(JSON.stringify(res));
    }
    $("#result").text(res.result)
    }
});
}
```

# Node.js 데이터 입력

https://www.w3schools.com/html/

➢ views/index.html

```
<head>
<title><%=title%></title>
<script type="text/javascript" src="https://code.jquery.com/jquery-3.3.1.min.js">
<script type="text/javascript" src="/javascripts/test.js"/>
</head>
```

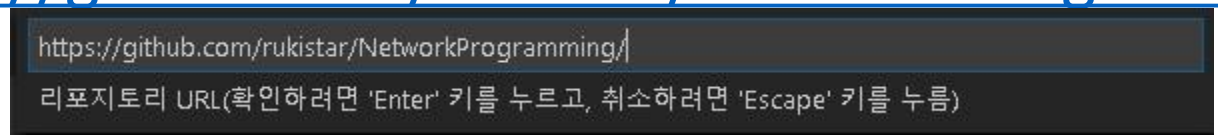
```
<body>
    User Name: <input type="text" id="name" name="name" value="test">
    <br/>
    <button onclick="test()">input</button>
    <br/>
    result: <label id="result"></label>
</body>
```

# Node.js 로그인 처리

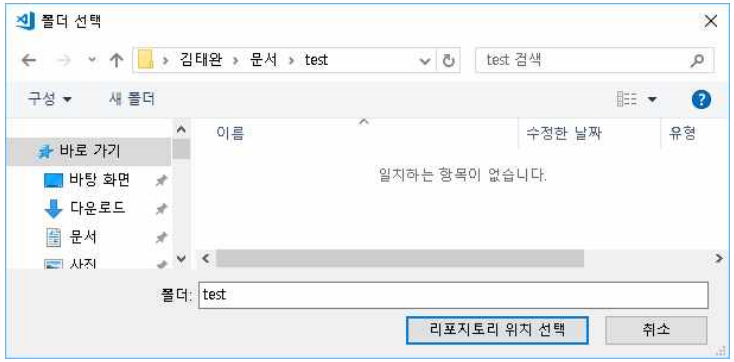
https://www.w3schools.com/html/

- F1 -> Git Clone


- 주소 입력 : https://github.com/rukistar/NetworkProgramming/

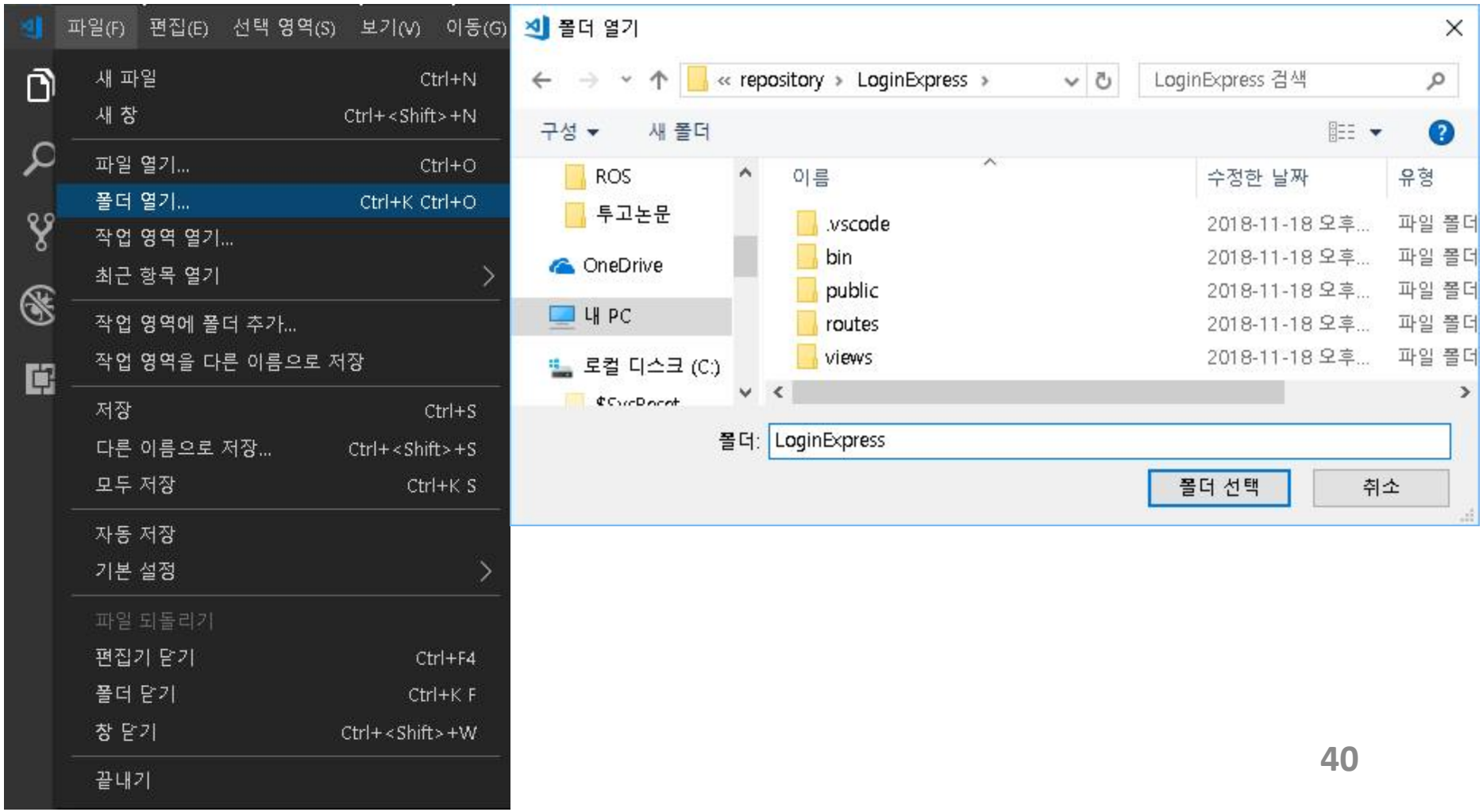



- 폴더지정



# Node.js 로그인 처리

➢ 파일 -> 폴더열기 https://github.com/rukistar/NetworkProgramming/

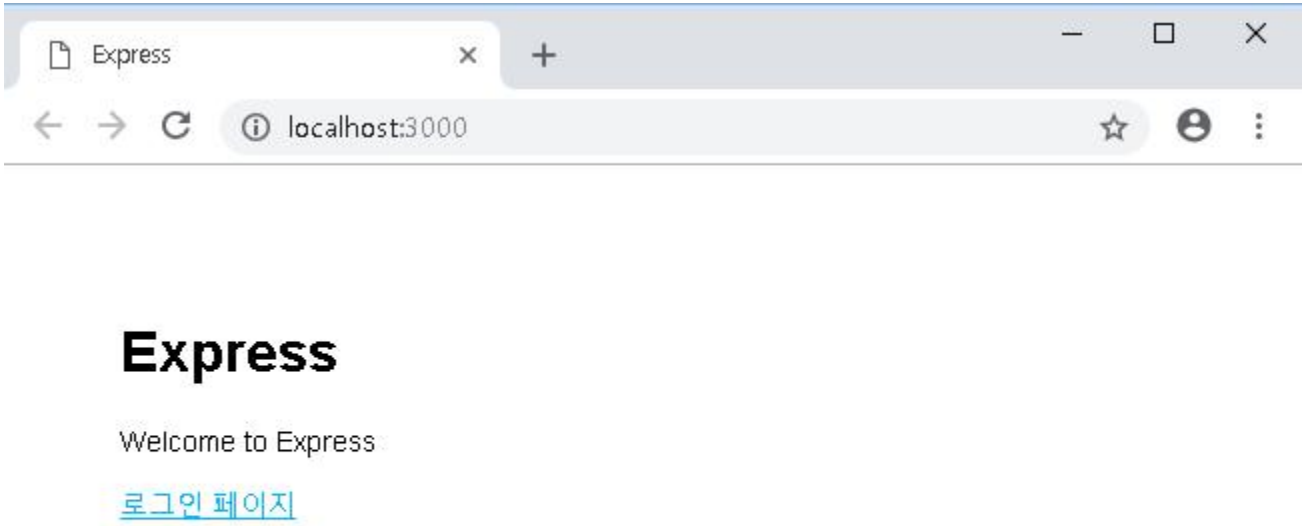


# Node.js 로그인 처리

➢index.html

/LoginExpress

https://www.w3schools.com/w3css

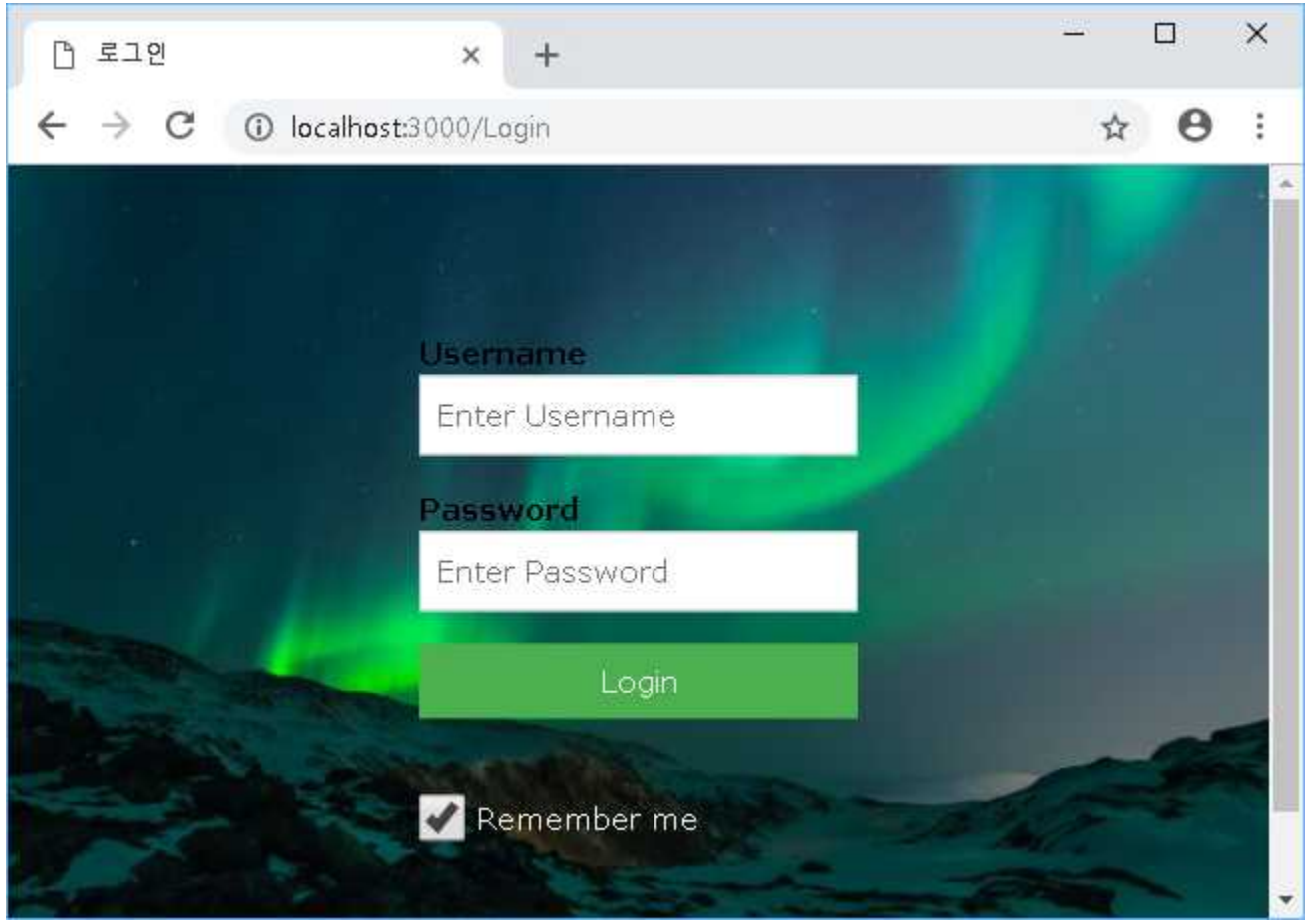


# Node.js 로그인 처리

➢login.html

/LoginExpress

https://www.w3schools.com/w3css



# Node.js 소켓 프로그래밍

https://www.npmjs.com/package/ws

- Websocket vs Socket.io
- Websocket
  - http Web은 http프로토콜을 이용 요청 / 응답 동작
  - Web에서 TCP/IP Socket 처럼 실시간 통신 할 수 없음
  - 그래서 등장 WebSocket(ws) 프로토콜
  - Websocket을 이용하면 웹브라우저
  - 실시간 데이터 주고 받을 수 있음
  - 최근 대부분 브라우저에서 Websocket지원
  - InternetExplorer 의 경우 version 10부터 지원

# Node.js 소켓 프로그래밍

https://www.npmjs.com/package/ws

- Websocket vs Socket.io
- Websocket
  - npm install -g express-generator
  - express -e ./
  - npm install
  - npm install ws

# Node.js 소켓 프로그래밍

- Websocket vs Socket.io

https://socket.io

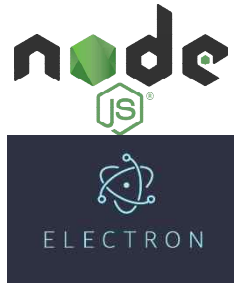
- Socket.io
  - Node.js 기반 OpenSource Library
  - WebSocket처럼 실시간 데이터 처리 가능
  - 멀티 디바이스 (Web, android, iOS, Windows, 등 )
  - WebSocket이 지원되지 않는 브라우저도 가능

# Node.js 소켓 프로그래밍

- Socket.io
  - Server
    - npm install socket.io
  - Client
    - npm install socket.io-client

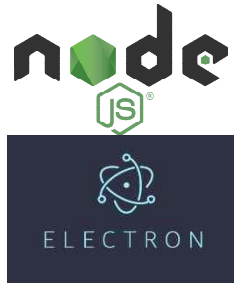
https://socket.io

# Node.js 응용프로그래밍



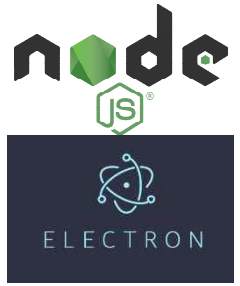

- Electron
  - https://electronjs.org/docs/tutorial/first-app
  - https://github.com/electron/electron-quick-start
  - https://github.com/electron/electron-api-demos

# Node.js 응용프로그래밍




- Electron – Socket.io, Chat
  - Git clone https://github.com/electron/electron-quick-start
  - npm install
  - npm install jquery
  - npm install socket.io
  - npm install socket.io-client
  - /socket/server.js
  - index.html 수정

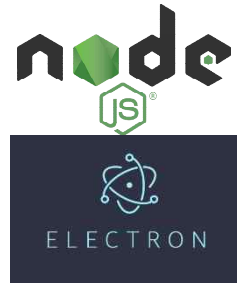
# Node.js 응용프로그래밍




- Electron – Socket.io, Chat
  - /socket/server.js

```
var app = require('express')();
var http = require('http').Server(app);
var io = require('socket.io')(http);
var port = process.env.PORT || 3000;
app.get('/', function(req, res){
  res.sendFile(__dirname + '/index.html');
});
io.on('connection', function(socket){
  socket.on('chat message', function(msg){
    io.emit('chat message', msg);
  });
});
http.listen(port, function(){
  console.log('listening on *:' + port);
});
```

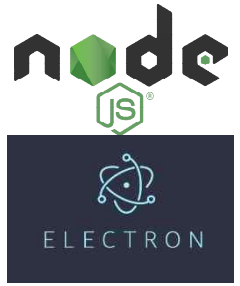
# Node.js 응용프로그래밍




- Electron – Socket.io, Chat
  - index.html 수정

```
<body>
<ul id="messages"></ul>
<form action="">
<input id="m" autocomplete="off" /><button>Send</button>
</form>
<script>
window.$ = window.jQuery = require('jquery');
require('./renderer.js')
</script>
</body>
```

# Node.js 응용프로그래밍




- Electron – Socket.io, Chat
  - ./renderer.js

```
$(function () {
  var socket = require('socket.io-client')
('http://localhost:3000');
// var socket = io();
  $('form').submit(function () {
    socket.emit('chat message', $('#m').val());
    $('#m').val('');
    return false;
  });
  socket.on('chat message', function (msg) {
    $('#messages').append($('<li>').text(msg));
    window.scrollTo(0, document.body.scrollHeight);
  });
});
```

# 팀 프로젝트

- 소켓 통신
- 다중 디바이스 (옵션*)
  (Electron / Android / Arduino / Raspberry PI 등등…)
- 아이디어 / 기획 / 기능 정의
- 시스템 설계
- 화면 구성 (사용자 인터페이스)
- 구현

# 참조

- **Html**
  - https://www.w3schools.com/html/default.asp
- **Javascript**
  - https://www.w3schools.com/js/default.asp **(javascript)**
  - https://www.w3schools.com/jquery/default.asp **(jquery)**
- **CSS**
  - https://www.w3schools.com/css/default.asp **(css)**
  - https://www.w3schools.com/w3css/default.asp **(w3.css library)**
- **Node.js**
  - https://nodejs.org/dist/latest-v10.x/docs/api/ **(node.js document)**
  - https://velopert.com/category/dev-log/tech-log/nodejs **(blog)**
  - http://bcho.tistory.com/tag/node.js **(blog)**

# 참조

- **Socket.io**
  - **https://socket.io/**
- **Electron**
  - **https://electronjs.org/docs** **(electron document)**
  - **https://electronjs.org/apps** **(electron 참조)**
  - **https://electronjs.org/docs/tutorial/first-app** **(electron tutorial)**
- npm
  - **https://www.npmjs.com/** (node.js library)