

# Electron.js

참고 강의: <https://www.udemy.com/course/electron-from-scratch/>

팁

exe 추출

시작

첫 프로젝트

둘째 프로젝트

## Tip

### ▼ 1. exe 추출 방법

1) electron-packager 설치

```
$ npm i -D electron-packager
```

2) package-json에서 잘 설치되었는지 확인

```
{  
  "devDependencies": {  
    "electron": "^21.2.2",  
    "electron-packager": "^17.1.1",  
    "nodemon": "^2.0.20"  
  }  
},  
You, 3일 전 • Add: completing
```

3) scripts에 해당 코드 추가

```
"scripts": {  
  ...  
  "package-win": "electron-packager . --overwrite --platform=win32 --arch=ia32 -  
-icon=assets/icons/favicon.ico (아이콘 모양, 설정 안 하면 일렉트론 기본 아이콘) --prune=fals  
e --out=release-builds --version-string.CompanyName=CE --version-string.FileDescri
```

```
ption=CE --version-string.ProductName=\"ImageShrink\"\"(앱 이름)
},
```

```
$ yarn package-win
```

4) 명령 실행 후 조금 기다리면 `release-builds` 라는 폴더가 생기고, 그 안에 들어가면 .exe 파일이 생성되어 있다

바탕 화면 > Studying > Electron > image_shrink > release-builds > ImageShrink-win32-ia32				
이름	수정한 날짜	유형	크기	
locales	2022-11-04 오후 3:07	파일 폴더		
resources	2022-11-04 오후 3:07	파일 폴더		
chrome_100_percent.pak	2022-11-04 오후 3:07	PAK 파일	127KB	
chrome_200_percent.pak	2022-11-04 오후 3:07	PAK 파일	176KB	
d3dcompiler_47.dll	2022-11-04 오후 3:07	응용 프로그램 확장	4,013KB	
ffmpeg.dll	2022-11-04 오후 3:07	응용 프로그램 확장	2,463KB	
icudtl.dat	2022-11-04 오후 3:07	DAT 파일	10,206KB	
ImageShrink	2022-11-04 오후 3:07	응용 프로그램	130,747KB	



용량이 커서 git에 안 들어가지니까 .gitignore에 /release\_builds 이하 파일 추가하자

## 프로젝트

### ▼ 1. 시작

- 사용 이유

**Data Security:** Data is usually stored on the users computer, rather than on a server

**Native APIs:** Access and interact with the OS, hardware components, network ports, camera and much more

**Performance:** No network overhead of dealing with a web server

**Deployment:** No server deployment and maintenance

### **All of the benefits we just talked about**

**Single Codebase:** No need for separate languages and technology for multiple platforms

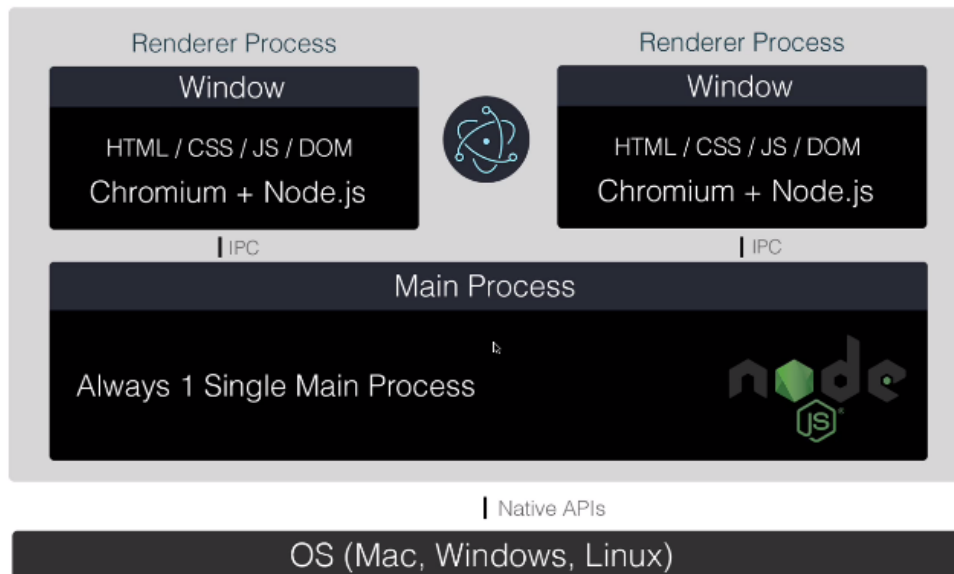
**Cost Effectiveness:** Single codebase is much less expensive

**Web Technologies:** Much easier to learn than other languages used to created desktop apps

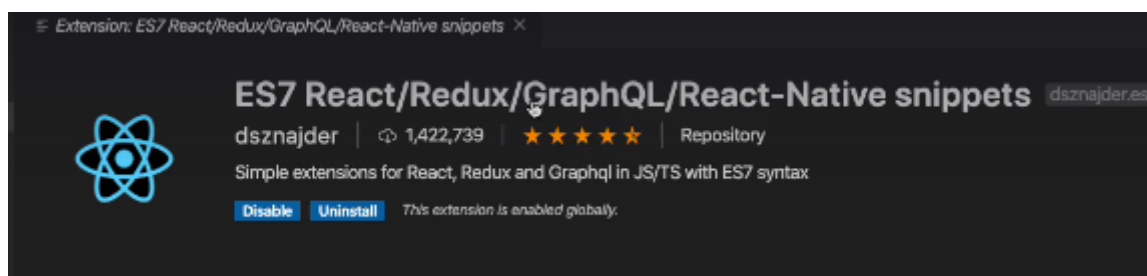
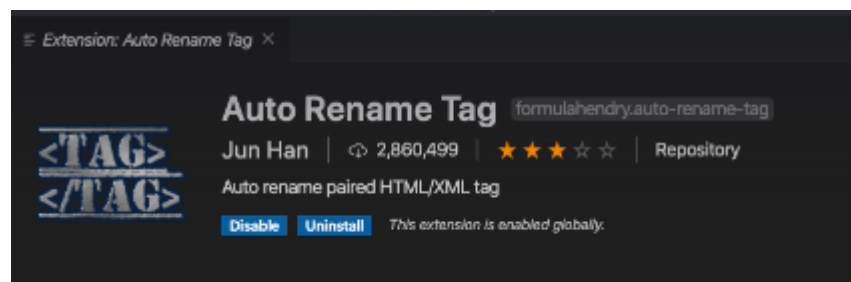
**Desktop / Web Hybrid:** Easily create apps with a desktop and web version

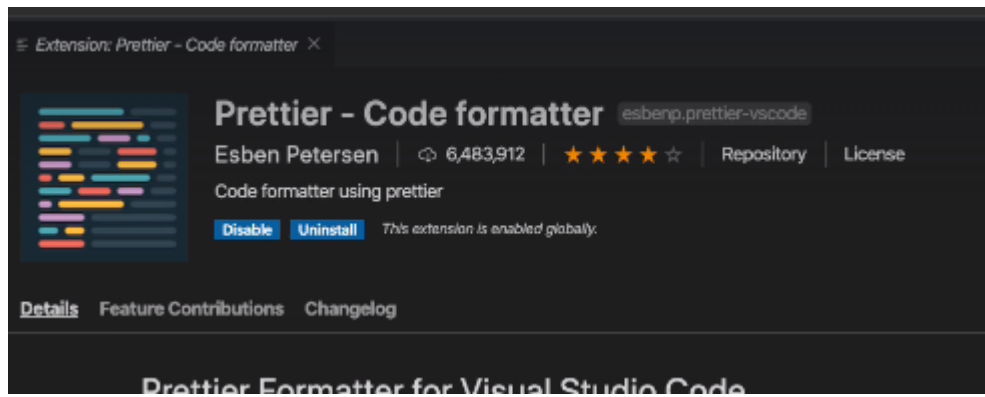
- 구동 원리

# How Electron Works

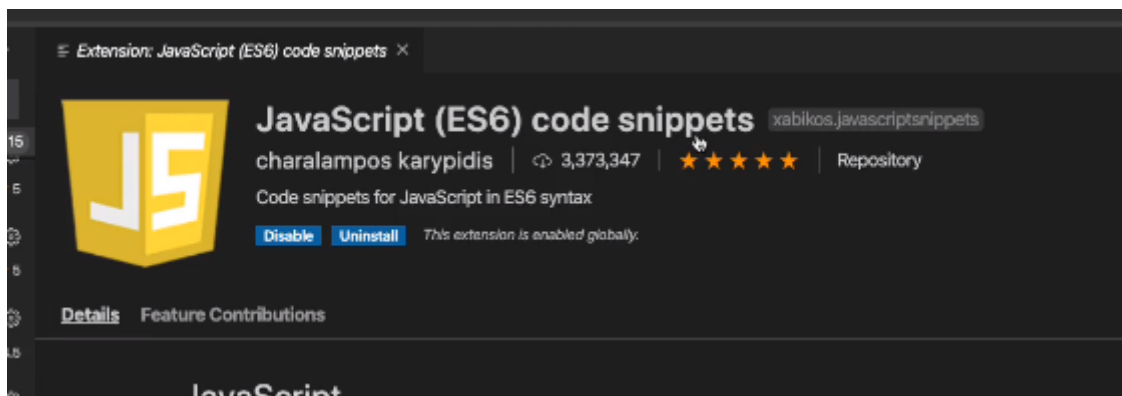


- 확장 프로그램





- format on save에서 저장 시 적용으로 바꾸기



- Vs Code icons

## ▼ 2. 첫 프로젝트

- 쓸 만한 것들: 단축키로 시작 및 종료, 바로가기 생성, 시작 페이지에 추가, 우클릭에 프로그램 추가

### 1. 첫 시작 때 콘솔 창에 더럽게 뜨는 것들 제거할 코드

```
// index.html
<meta
  http-equiv="Content-Security-Policy"
  content="script-src 'self'
  'unsafe-inline'"
/>

//main.js
app.allowRendererProcessReuse = true;
```

### 3. Mac과 Windows 환경 따로 나눠서 세팅 필요

```
// mac 환경인지 여부 확인
const isMac = process.platform === "darwin" ? true : false;
const isWin = process.platform === "win32" ? true : false;
```

#### 4. 개발자 모드와 일반 유저 모드 나눠야 함

```
//저장 시 앱 자동 리로드
$ npm i -D nodemon

//main.js
process.env.NODE_ENV = "development";
const isDev = process.env.NODE_ENV !== "production" ? true : false;

// 개발자 모드가 아니라면?
process.env.NODE_ENV = "production";

//package.json
"scripts": {
  "dev": "nodemon --exec electron ."
},
...
...(isDev
  ? [
    {
      label: "Developer",
      submenu: [
        { role: "reload" },
        { role: "forcereload" },
        { type: "separator" },
        { role: "toggledevtools" }
      ]
    }
  ]
  : []),
```

- dev로 켜올 때만 특정 메뉴나 키워드 보이게 하기
  - 예) 개발자 모드, 콘솔 창

#### 5. 창 여러 개 띄우기 가능

```
let mainWindow;
let aboutWindow;

function createMainWindow() {
  mainWindow = new BrowserWindow({
    title: "ImageShrink",
    width: 500,
    height: 600,
```

```

        icon: "./assets/icons/icon.png",
        resizable: isDev ? true : false,
        backgroundColor: "yellow",
    });

    //mainWindow.loadURL(`file://${__dirname}/app/index.html`);
    mainWindow.loadFile("./app/index.html");
}

function createAboutWindow() {
    aboutWindow = new BrowserWindow({
        title: "About ImageShrink",
        width: 300,
        height: 300,
        icon: "./assets/icons/icon.png",
        resizable: false,
        backgroundColor: "pink",
    });

    //mainWindow.loadURL(`file://${__dirname}/app/index.html`);
    aboutWindow.loadFile("./app/about.html");
}

```

- 튜토리얼 모드 따로 띄우기도 가능할 것 같다



6.

### ▼ 3. 둘째 프로젝트

1) 앱에 기재할 각종 정보 알아내기

```
// monitor.js
const path = require("path");
const osu = require("node-os-utils");
const cpu = osu.cpu;
```



```
const mem = osu.mem;
const os = osu.os;

// set model
document.getElementById("cpu-model").innerText = cpu.model();


// computer name
// 뒤에 있는 건 문서 참고
document.getElementById("pc-name").innerText = os.hostname();

// OS
document.getElementById("os").innerText = `${os.type()} ${os.arch()}`;

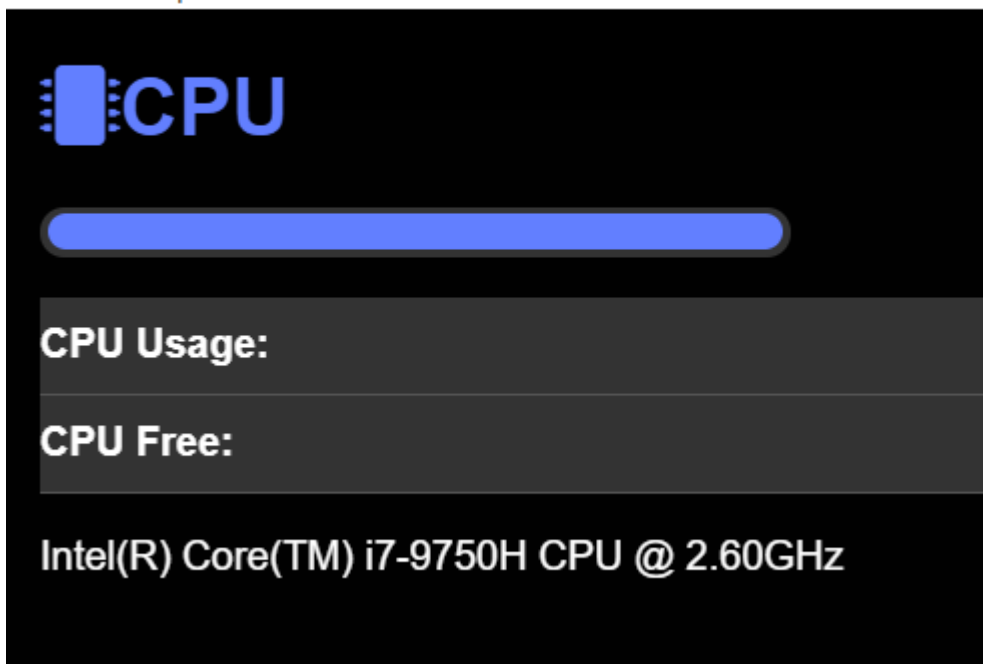
// Total memory
// promise 활용
mem.info().then((info) => {
  document.getElementById("mem-total").innerText = info.totalMemMb;
});
```

- 뒤에 붙는 것들은 해당 문서 참고
- `getElementById` 와 `innerText` 이용하여 cpu-model 알아내기
- 그전에 bash에서 node-os-utils 설치 후 제대로 설치되었는지 package.json 확인

```
$ npm i node-os-utils
```

 CPU Usage | System Monitor

File Developer



## **i** System Info

**Computer Name:** DESKTOP-KVCQHCD

**OS:** Windows\_NT x64

**System Uptime:**

**Memory Total:** 16315.72