
코도바 플러그인

13주차_01

한 동 대 학 교
김군오 교수

학습목표: 코도바 플러그인 설치 및 사용하기

학습내용:

- Cordova 플러그인 설치
- Cordova 배터리 상태 플러그인
- Cordova 카메라 플러그인

Cordova 플러그인

- NodeJS 의 NPM 과 같이 Cordova 의 기능을 확장
- 웹 앱에서 OS 기능을 사용하려면 플러그인이 필요
- OS 종속 코드와 자바스크립트 인터페이스로 이루어
짐

Cordova 플러그인 설치 및 삭제

- **설치:**

`cordova add --save {플러그인명}`

- **목록 확인:**

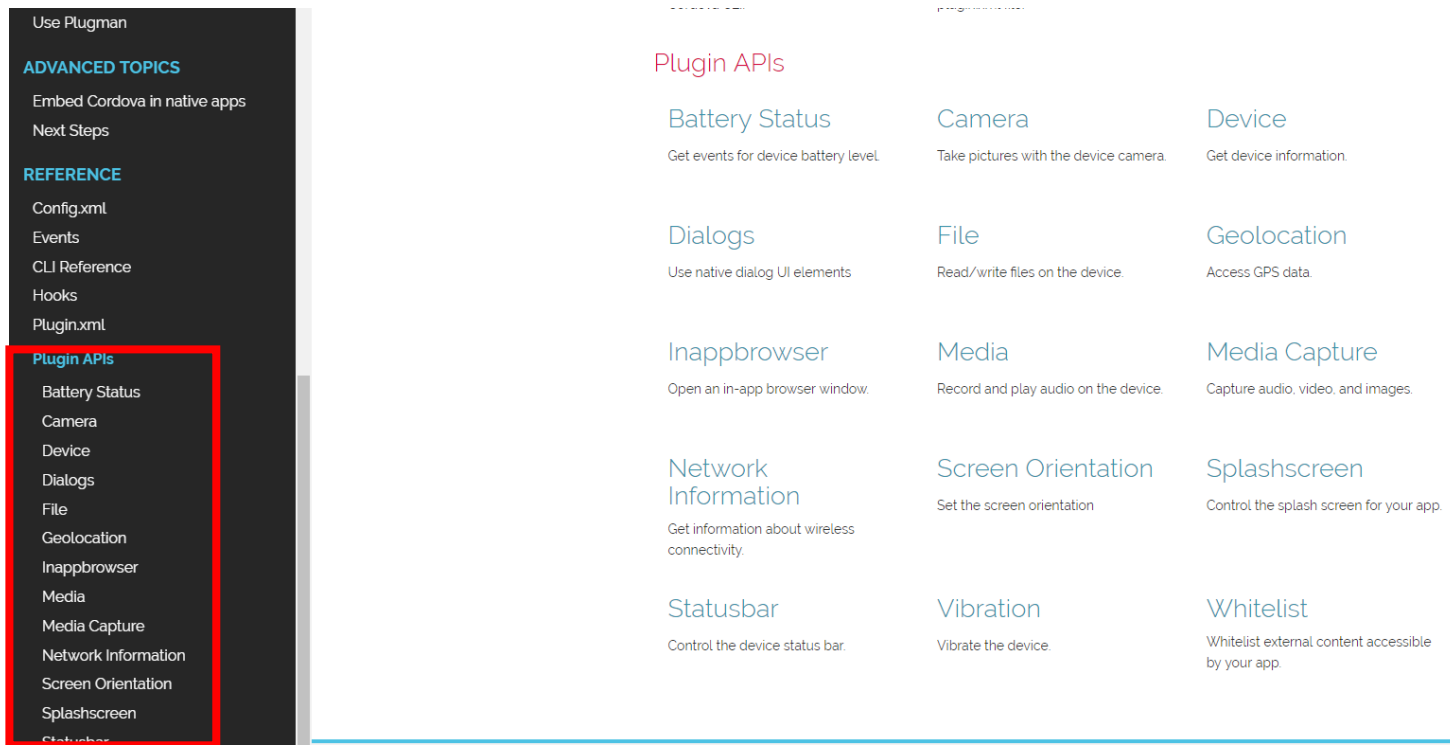
`cordova plugin list`

- **삭제:**

`cordova remove {플러그인명}`

플러그인 검색

- <https://cordova.apache.org/docs/en/latest/>



Battery status 플러그인

- 세가지 기능

1. 디바이스의 배터리 잔량 확인

- batterystatus 이벤트 -> status.level

2. 디바이스의 충전상태 확인

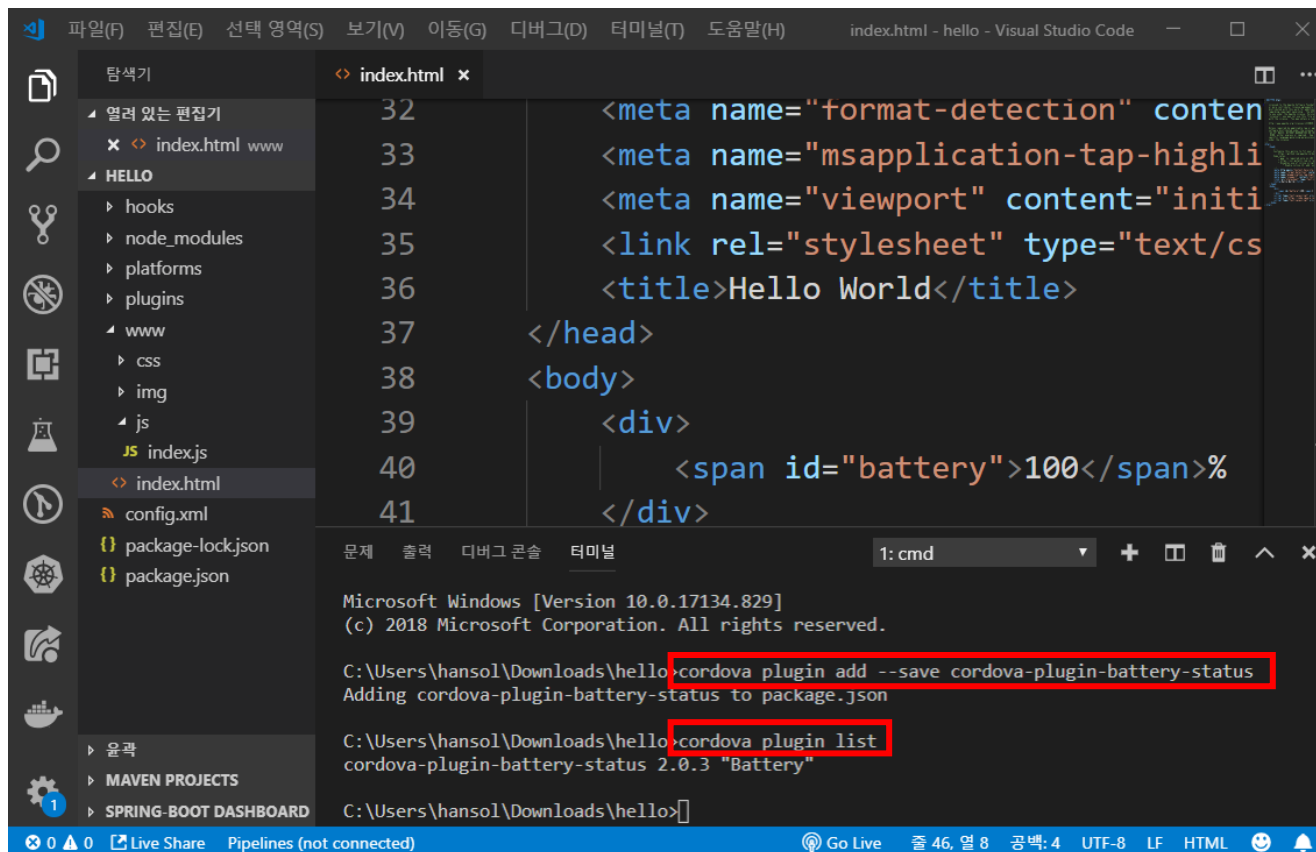
- batterystatus 이벤트 -> status.isPlugged

3. 디바이스의 배터리 경고 확인

- batterylow 이벤트
- batterycritical 이벤트

Battery status 플러그인 설치

cordova-plugin-battery-status



```
index.html x
32 <meta name="format-detection" conten
33 <meta name="msapplication-tap-highli
34 <meta name="viewport" content="initi
35 <link rel="stylesheet" type="text/cs
36 <title>Hello World</title>
37 </head>
38 <body>
39 <div>
40 <span id="battery">100</span>%
41 </div>
```

```
Microsoft Windows [Version 10.0.17134.829]
(c) 2018 Microsoft Corporation. All rights reserved.

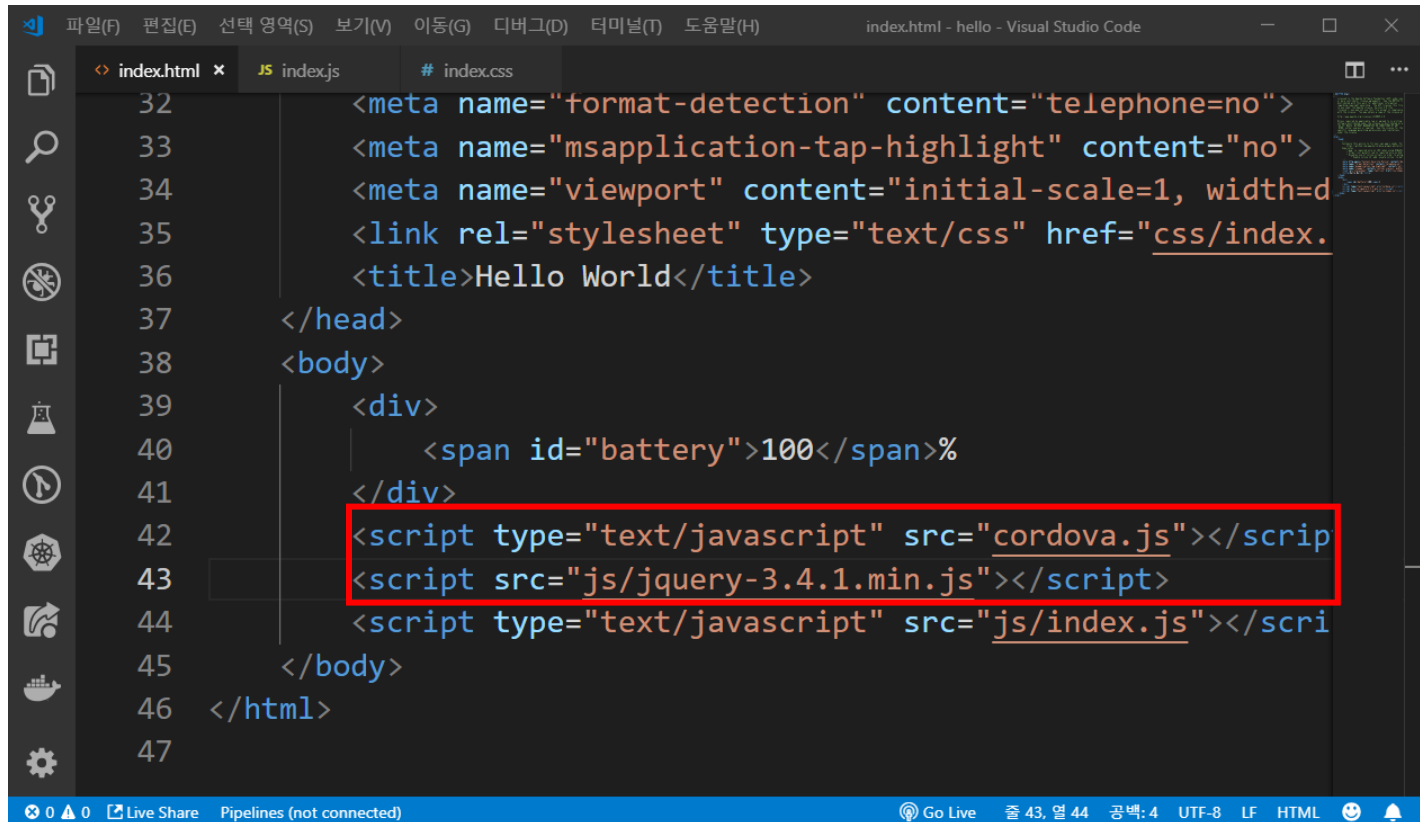
C:\Users\hansol\Downloads\hello>cordova plugin add --save cordova-plugin-battery-status
Adding cordova-plugin-battery-status to package.json

C:\Users\hansol\Downloads\hello>cordova plugin list
cordova-plugin-battery-status 2.0.3 "Battery"

C:\Users\hansol\Downloads\hello>
```

cordova.js

- 플러그인을 사용하려면 cordova.js 를 html 에 삽입해야함



```
32 <meta name="format-detection" content="telephone=no">
33 <meta name="msapplication-tap-highlight" content="no">
34 <meta name="viewport" content="initial-scale=1, width=device-width">
35 <link rel="stylesheet" type="text/css" href="css/index.css">
36 <title>Hello World</title>
37 </head>
38 <body>
39 <div>
40 <span id="battery">100</span>%
41 </div>
42 <script type="text/javascript" src="cordova.js"></script>
43 <script src="js/jquery-3.4.1.min.js"></script>
44 <script type="text/javascript" src="js/index.js"></script>
45 </body>
46 </html>
47
```


Jquery 다운로드

- <https://jquery.com/download/>

jQuery

For help when upgrading jQuery, please see the [upgrade guide](#) most relevant to your version. We [plugin](#).

[Download the compressed, production jQuery 3.4.1](#)

[Download the uncompressed, development jQuery](#)

[Download the map file for jQuery 3.4.1](#)

You can also use the slim build, which excludes the

[Download the compressed, production jQuery 3.4.](#)

[Download the uncompressed, development jQuery](#)

[Download the map file for the jQuery 3.4.1 slim build](#)

새 탭에서 링크 열기(T)

새 창에서 링크 열기(W)

시크릿 창에서 링크 열기(G)

다른 이름으로 링크 저장(K)...

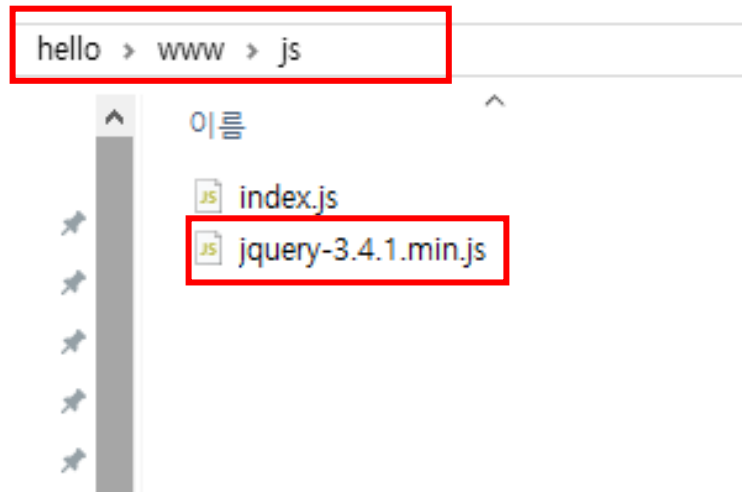
링크 주소 복사(E)

AdBlock은

검사(N)

Ctrl+Shift+I

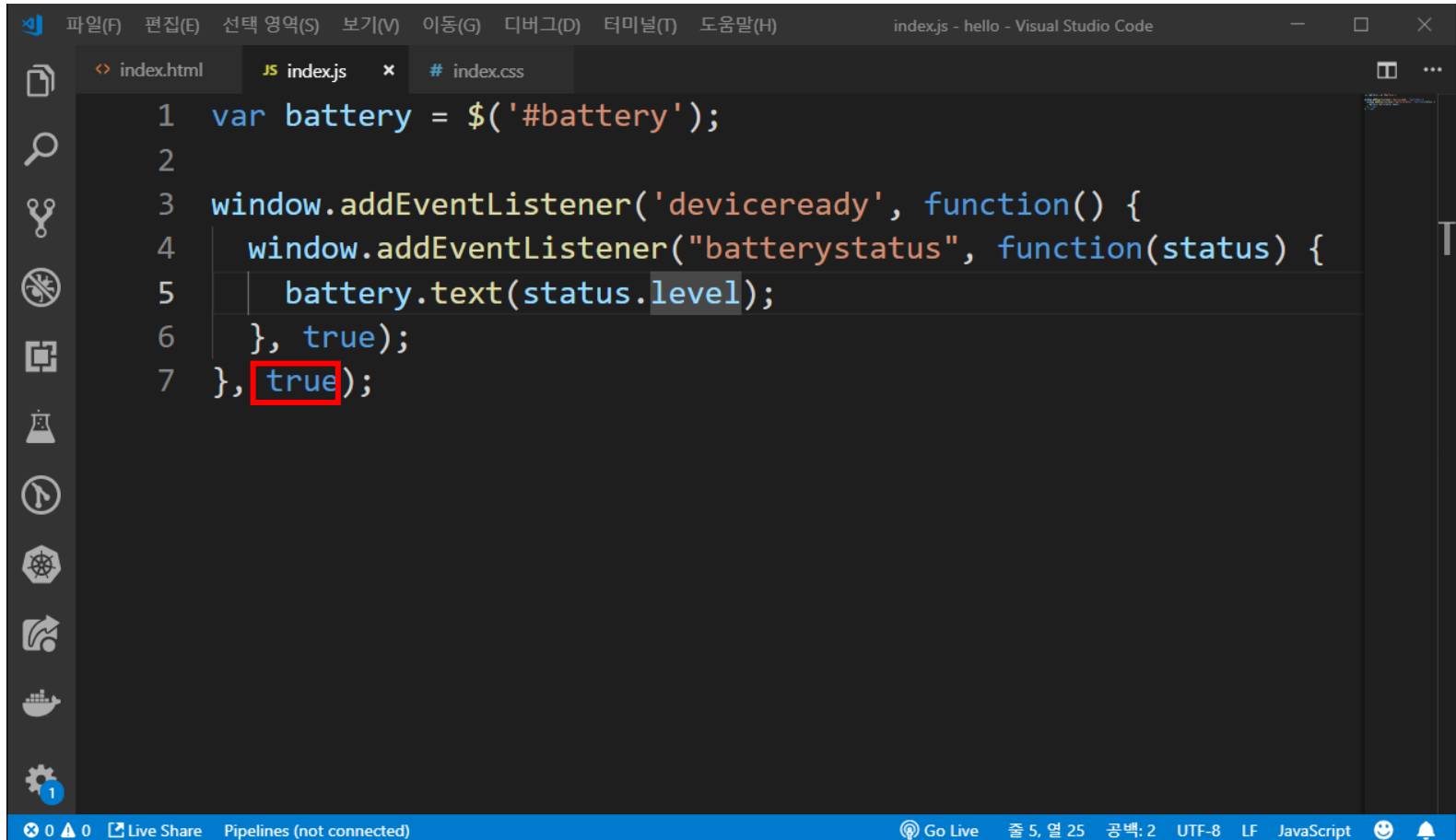
Jquery 다운로드



deviceready 이벤트

- Cordova 플러그인이 모두 로드 된 후 발생
- `window.addEventListener` 함수를 통해 등록
- 플러그인은 deviceready 이벤트 발생 이후 사용하여야 함

deviceready 이벤트처리기 등록

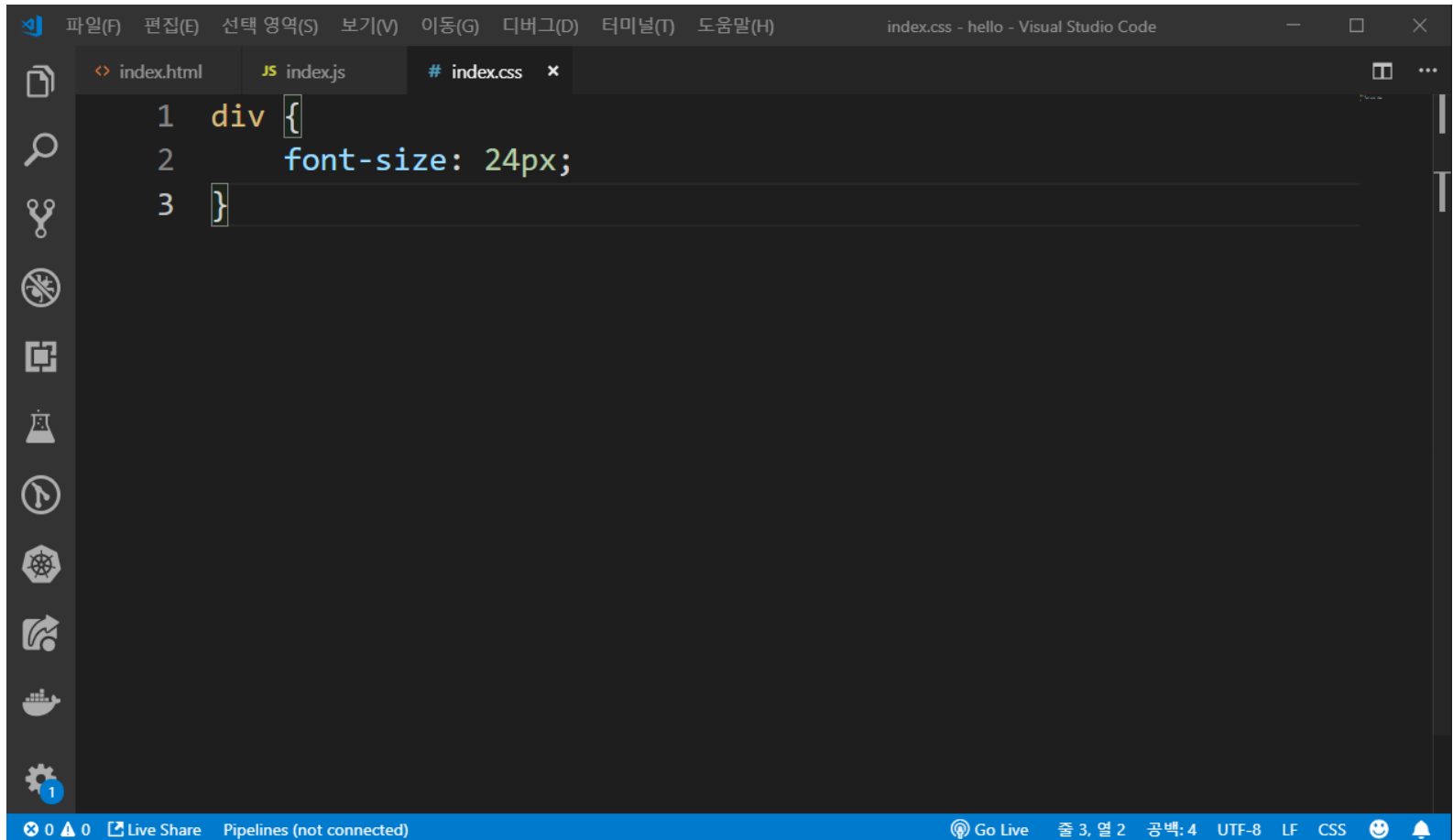


The image shows a screenshot of the Visual Studio Code editor interface. The top menu bar includes options like '파일(F)', '편집(E)', '선택 영역(S)', '보기(V)', '이동(G)', '디버그(D)', '터미널(T)', and '도움말(H)'. The title bar indicates the file is 'index.js - hello - Visual Studio Code'. The editor has three tabs open: 'index.html', 'JS index.js', and '# index.css'. The 'JS index.js' tab is active, displaying the following JavaScript code:

```
1 var battery = $('#battery');
2
3 window.addEventListener('deviceready', function() {
4     window.addEventListener("batterystatus", function(status) {
5         battery.text(status.level);
6     }, true);
7 }, true);
```

The code is written in a dark-themed editor. The 'true' value in the second call to `addEventListener` on line 6 is highlighted with a red rectangle. The status bar at the bottom shows '0' errors, '0' warnings, 'Live Share' status, 'Pipelines (not connected)', 'Go Live' button, and file encoding/line ending information: '줄 5, 열 25', '공백: 2', 'UTF-8', 'LF', 'JavaScript'.

CSS 수정



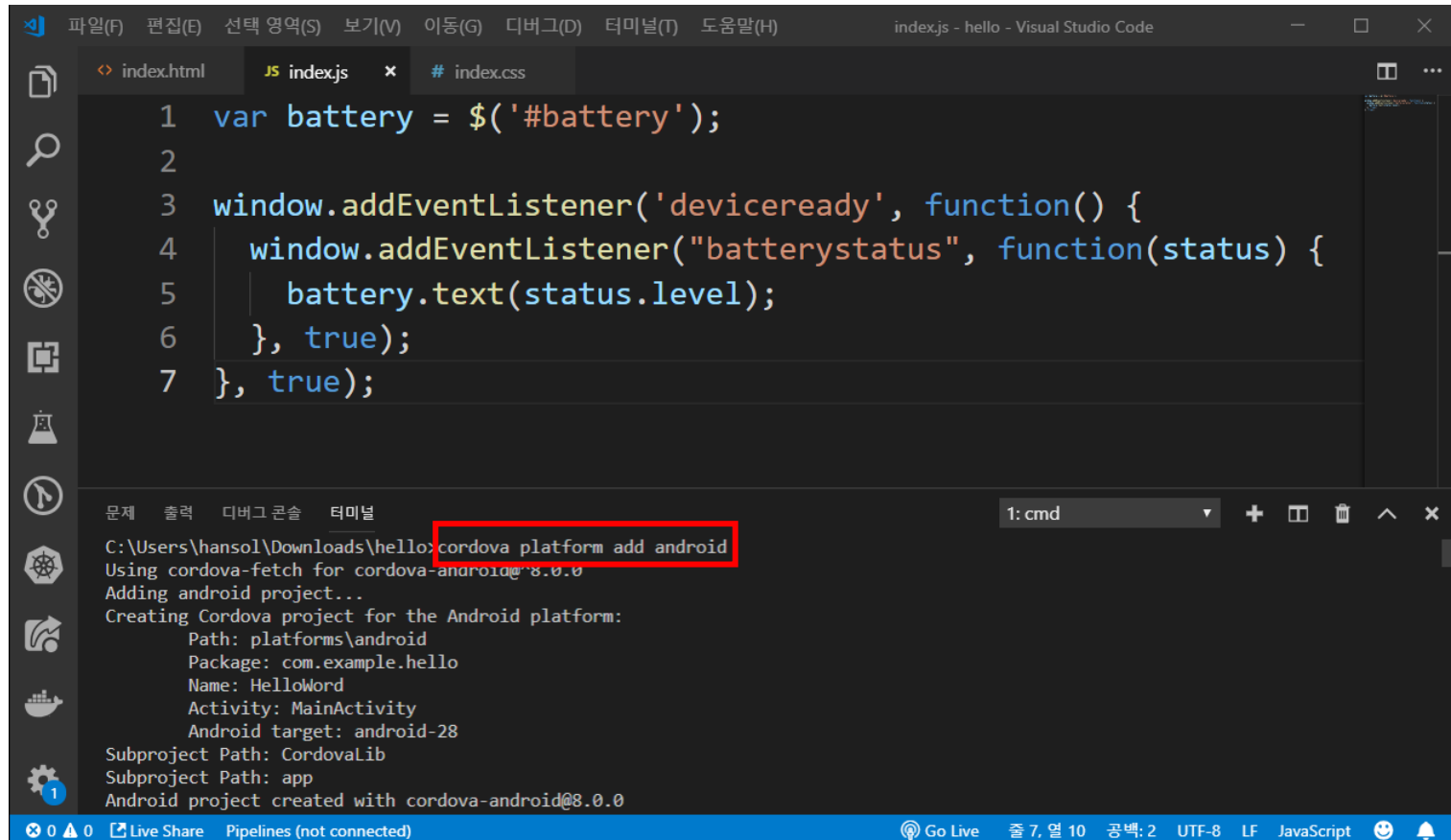
The screenshot shows the Visual Studio Code editor interface. The top menu bar includes options like '파일(F)', '편집(E)', '선택 영역(S)', '보기(V)', '이동(G)', '디버그(D)', '터미널(T)', and '도움말(H)'. The title bar indicates the active file is 'index.css - hello - Visual Studio Code'. The editor has three tabs: 'index.html', 'index.js', and the active 'index.css'. The CSS code in the editor is as follows:

```
1  div {  
2    font-size: 24px;  
3  }
```

The status bar at the bottom shows '0 0 0' on the left, 'Live Share Pipelines (not connected)' in the middle, and 'Go Live', '줄 3, 열 2', '공백: 4', 'UTF-8', 'LF', 'CSS', and a bell icon on the right.

Android 플랫폼 추가

- cordova platform add android



The screenshot shows the Visual Studio Code interface with a JavaScript file open. The terminal at the bottom displays the command `cordova platform add android` being executed. The command is highlighted with a red box. The terminal output shows the process of adding the Android platform to the Cordova project.

```
1 var battery = $('#battery');
2
3 window.addEventListener('deviceready', function() {
4   window.addEventListener("batterystatus", function(status) {
5     battery.text(status.level);
6   }, true);
7 }, true);
```

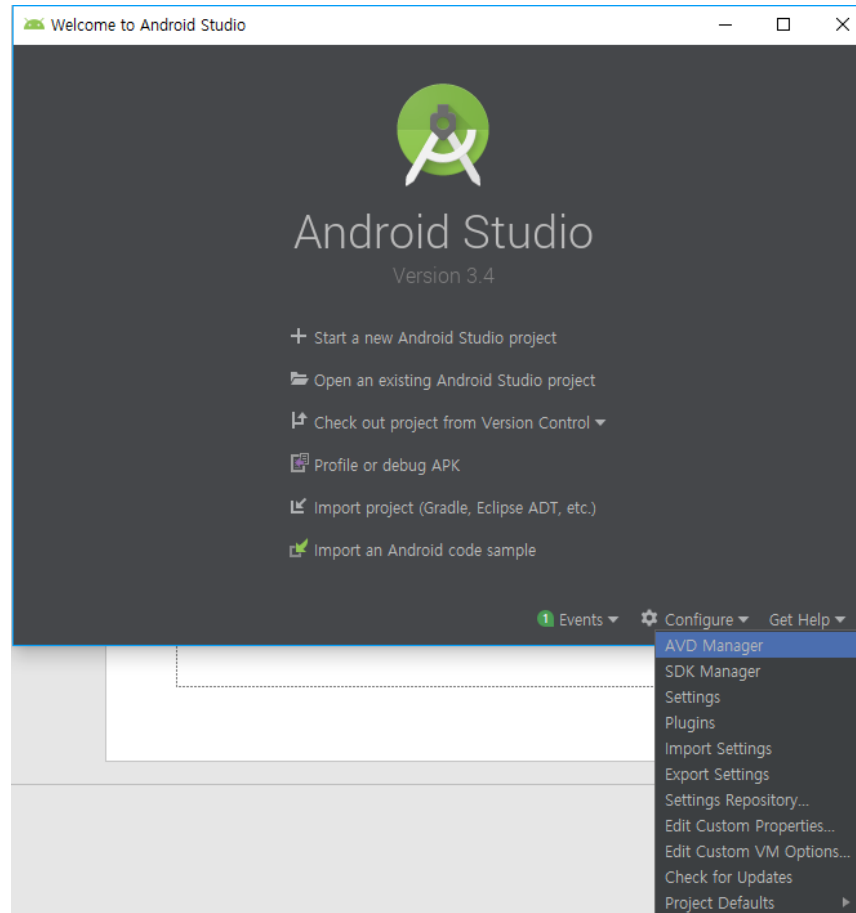
문제 출력 디버그 콘솔 터미널

1: cmd

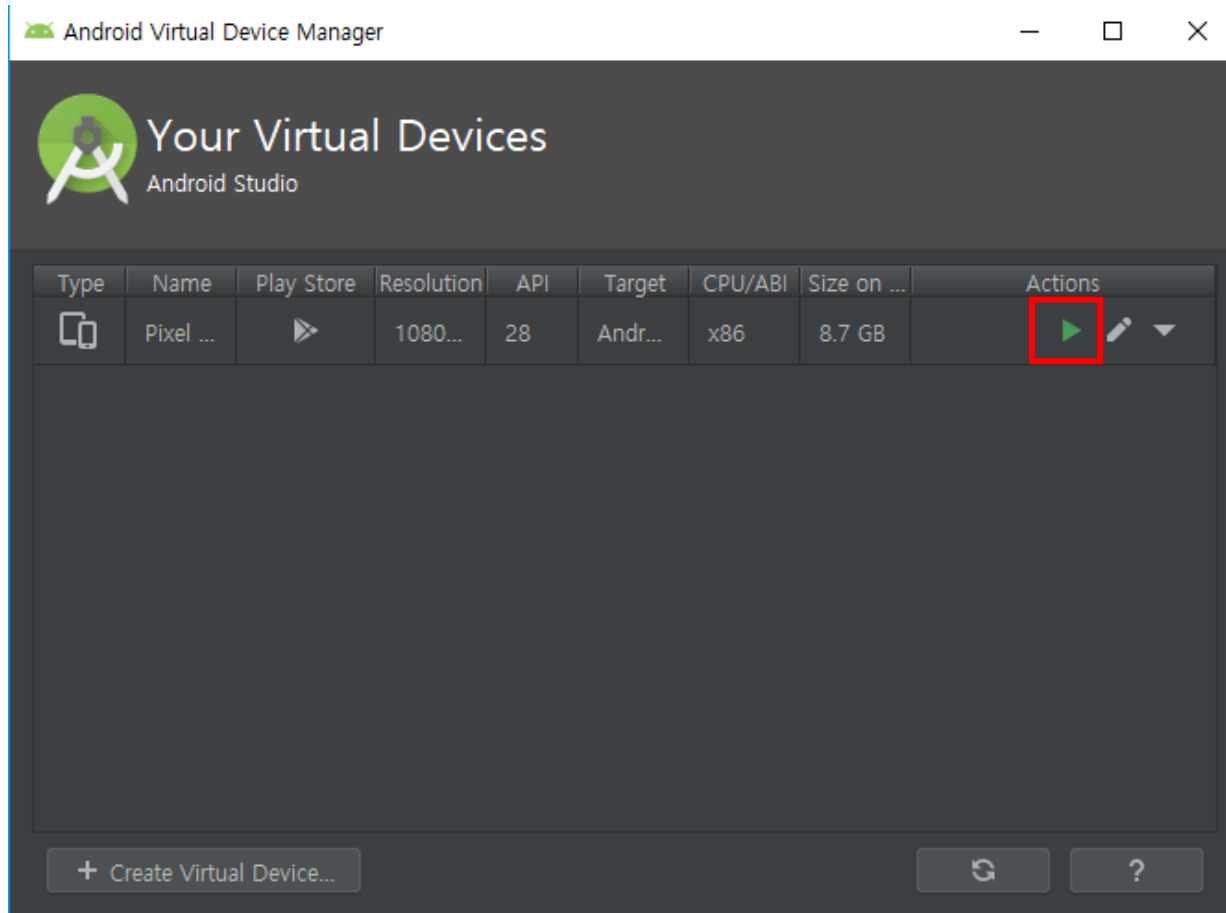
```
C:\Users\hansol\Downloads\hello>cordova platform add android
Using cordova-fetch for cordova-android@8.0.0
Adding android project...
Creating Cordova project for the Android platform:
  Path: platforms\android
  Package: com.example.hello
  Name: HelloWorld
  Activity: MainActivity
  Android target: android-28
Subproject Path: CordovaLib
Subproject Path: app
Android project created with cordova-android@8.0.0
```

0 0 0 Live Share Pipelines (not connected) Go Live 줄 7, 열 10 공백: 2 UTF-8 LF JavaScript

AVD 실행

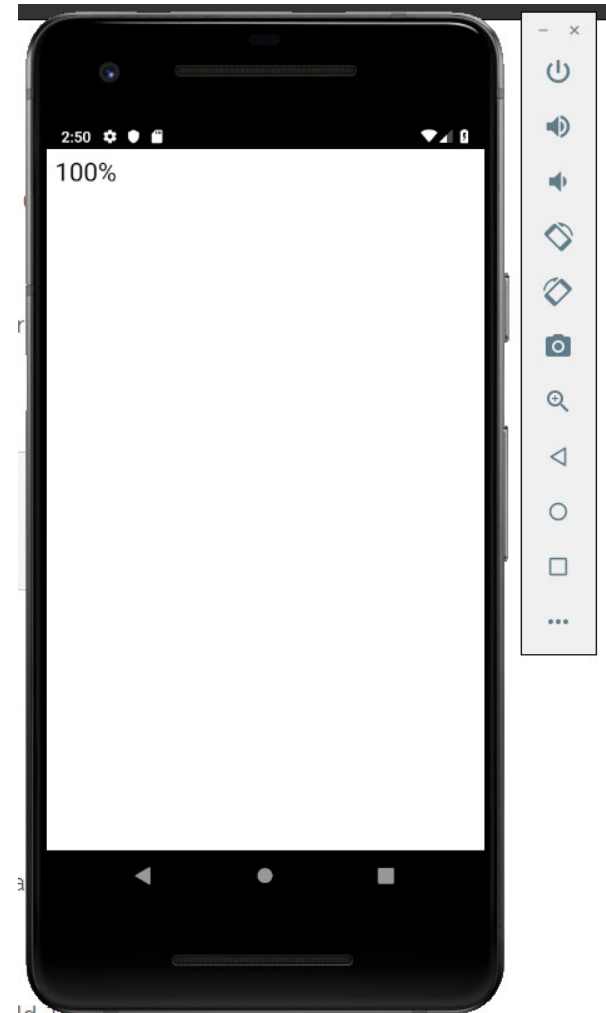


AVD 실행

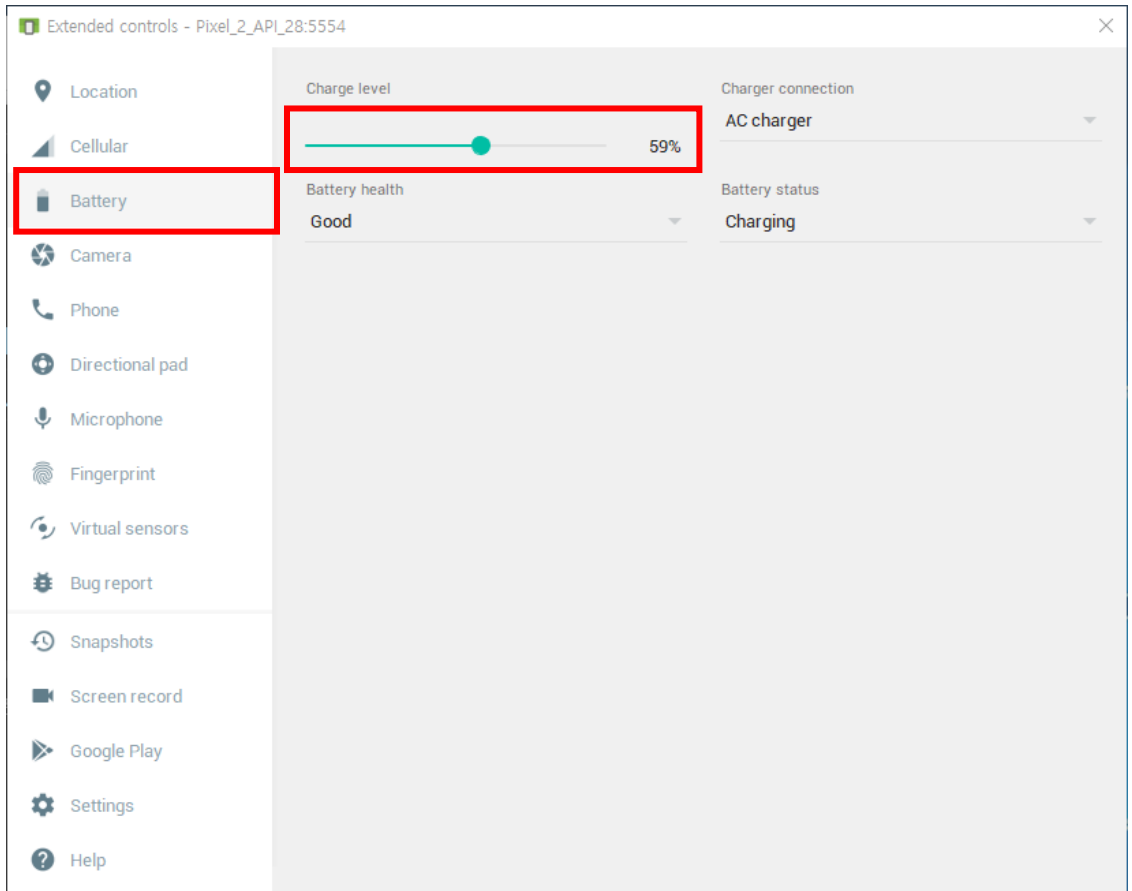
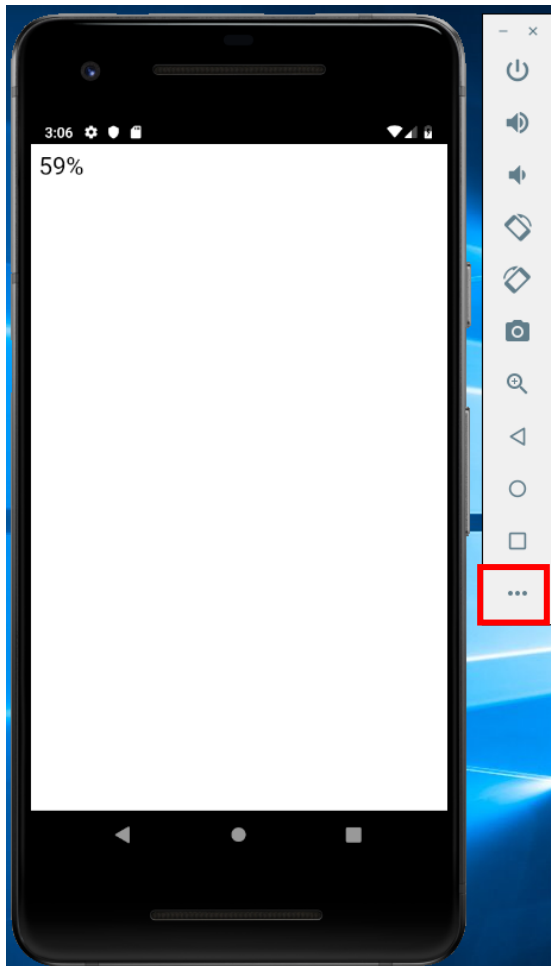


Android 앱 실행

- cordova run android



배터리 잔량 수정



카메라 플러그인

- 디바이스의 카메라로 사진을 찍어 불러오기
- 디바이스의 이미지 라이브러리에서 사진 불러오기
- **getPicture 함수**

`camera.getPicture(success, error, options)`

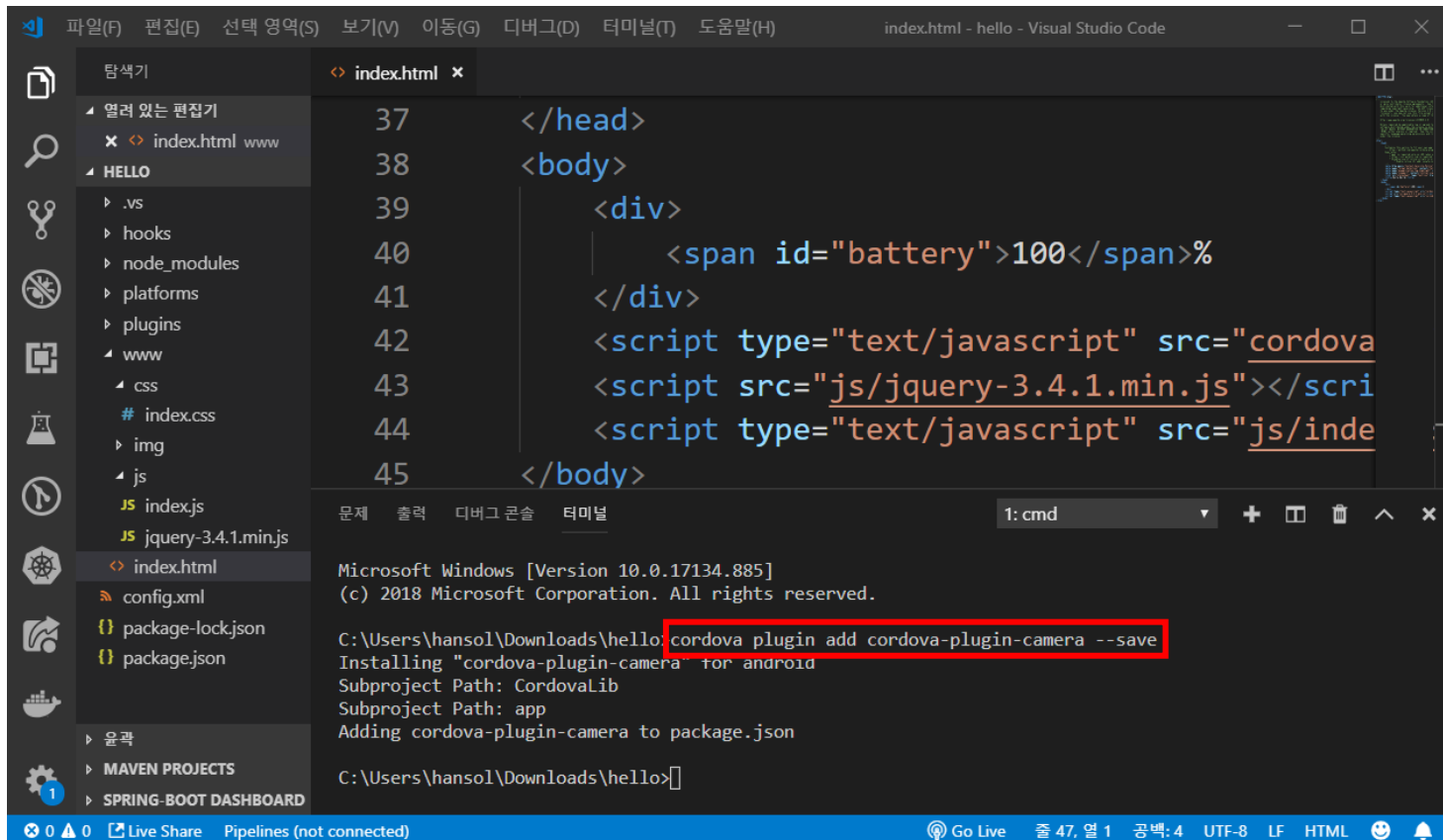
- `success` : 사진을 불러오는데 성공했을 때 실행되는 콜백함수
- `error` : 사진 불러오기에 실패했을 때 실행되는 콜백함수
- `options` : 사진을 가져오기 위한 option 객체

Camera option 객체

프로퍼티	타입	기본값	설명
quality	number	50	사진 품질
sourceType	0, 1	1	0: 사진 라이브러리 1: 카메라
correctOrientation	boolean		
saveToPhotoAlbum	boolean		
cameraDirection	0, 1	0	0: 후면카메라 1: 전면카메라

카메라 플러그인 설치하기

- `cordova plugin add cordova-plugin-camera --save`



The screenshot shows the Visual Studio Code interface with a project named 'hello'. The file explorer on the left shows the project structure, including 'index.html' and 'index.js'. The main editor displays the content of 'index.html', which includes a head section and a body section with a div containing a span with the text '100'. The terminal at the bottom shows the command `cordova plugin add cordova-plugin-camera --save` being executed, with the output indicating the successful installation of the plugin.

```
index.html - hello - Visual Studio Code

index.html x
37 </head>
38 <body>
39   <div>
40     <span id="battery">100</span>%
41   </div>
42   <script type="text/javascript" src="cordova
43   <script src="js/jquery-3.4.1.min.js"></scri
44   <script type="text/javascript" src="js/inde
45 </body>

문제 출력 디버그 콘솔 터미널
1: cmd
Microsoft Windows [Version 10.0.17134.885]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\hansol\Downloads\hello> cordova plugin add cordova-plugin-camera --save
Installing "cordova-plugin-camera" for android
Subproject Path: CordovaLib
Subproject Path: app
Adding cordova-plugin-camera to package.json
C:\Users\hansol\Downloads\hello>
```

카메라 플러그인 설치하기

- `cordova plugin add cordova-plugin-camera --save`

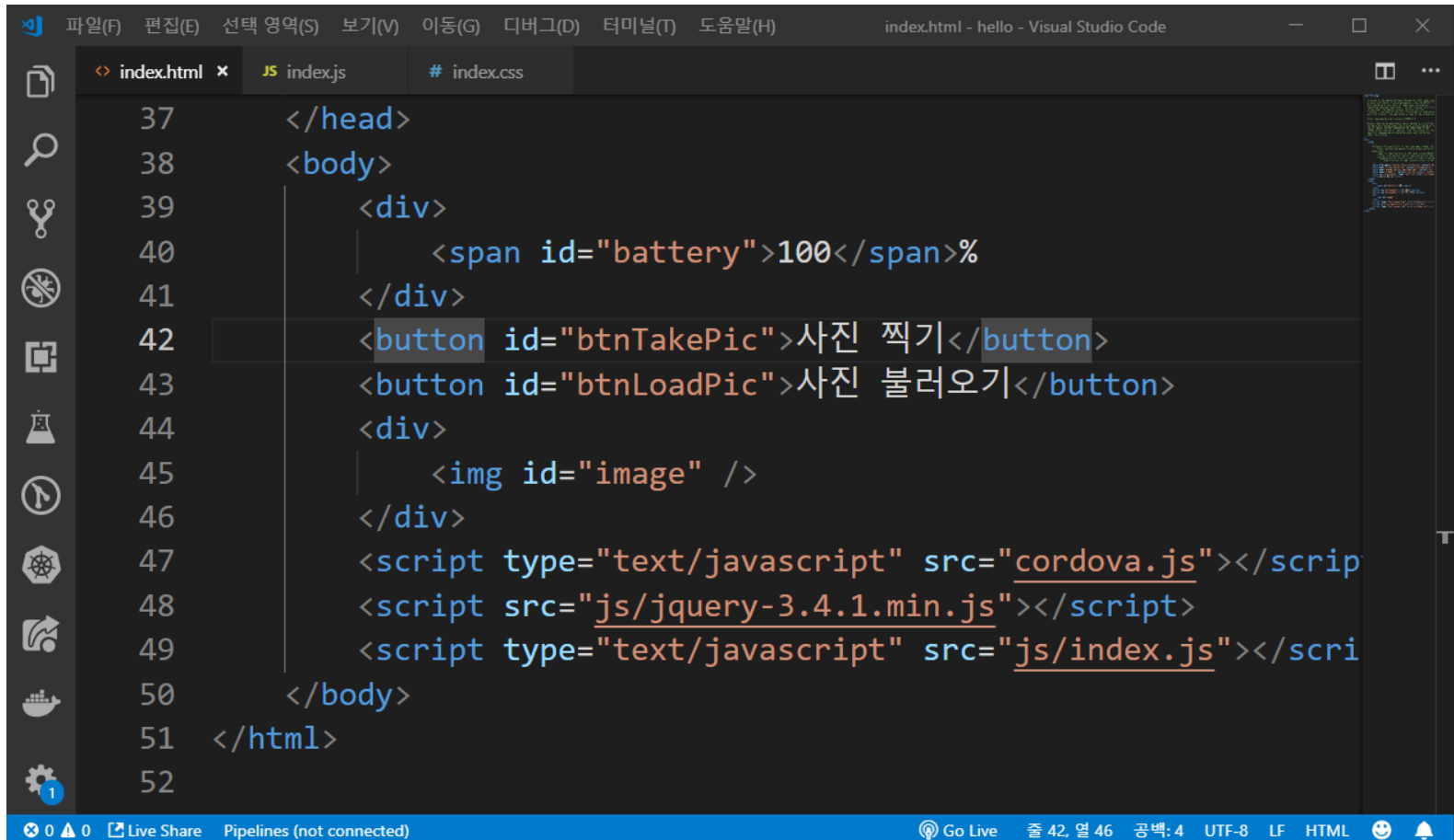
문제 출력 디버그 콘솔 터미널

1: cmd

```
Installing "cordova-plugin-camera" for android
Subproject Path: CordovaLib
Subproject Path: app
Adding cordova-plugin-camera to package.json
```

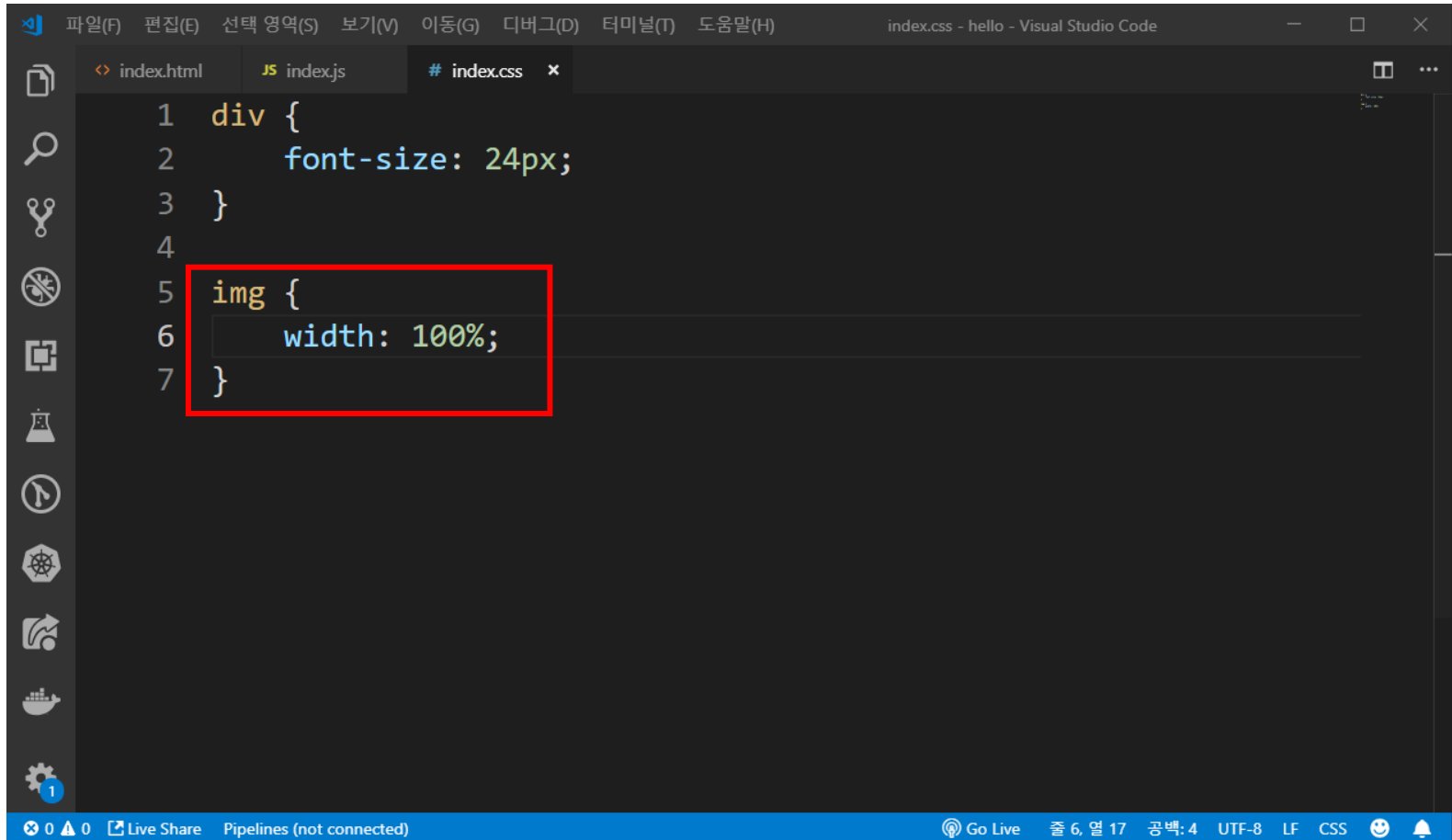
```
C:\Users\hansol\Downloads\hello>cordova plugin list
cordova-plugin-battery-status 2.0.3 "Battery"
cordova-plugin-camera 4.0.3 "Camera"
cordova-plugin-whitelist 1.3.4 "Whitelist"
```

카메라 플러그인 사용하기



```
37     </head>
38     <body>
39         <div>
40             <span id="battery">100</span>%
41         </div>
42         <button id="btnTakePic">사진 찍기</button>
43         <button id="btnLoadPic">사진 불러오기</button>
44         <div>
45             <img id="image" />
46         </div>
47         <script type="text/javascript" src="cordova.js"></scrip
48         <script src="js/jquery-3.4.1.min.js"></script>
49         <script type="text/javascript" src="js/index.js"></scri
50     </body>
51 </html>
52
```

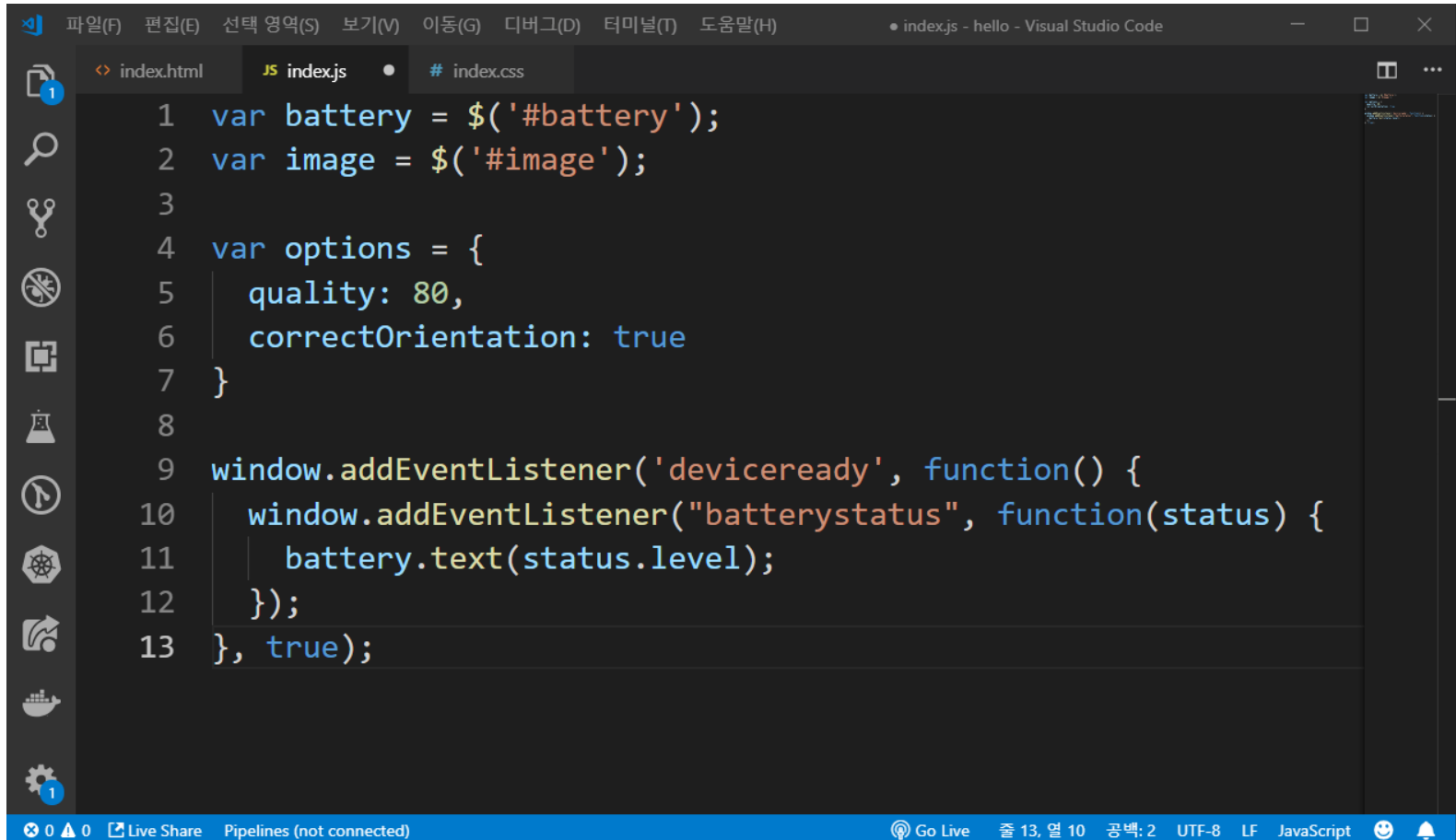
카메라 플러그인 사용하기



```
1  div {  
2      font-size: 24px;  
3  }  
4  
5  img {  
6      width: 100%;  
7  }
```

The screenshot shows the Visual Studio Code interface with the 'index.css' file open. The code defines a 'div' with a font size of 24px and an 'img' with a width of 100%. The 'img' block is highlighted with a red rectangle. The status bar at the bottom shows '0 errors, 0 warnings, 0 Live Share, Pipelines (not connected), Go Live, 줄 6, 열 17, 공백: 4, UTF-8, LF, CSS'.

카메라 플러그인 사용하기

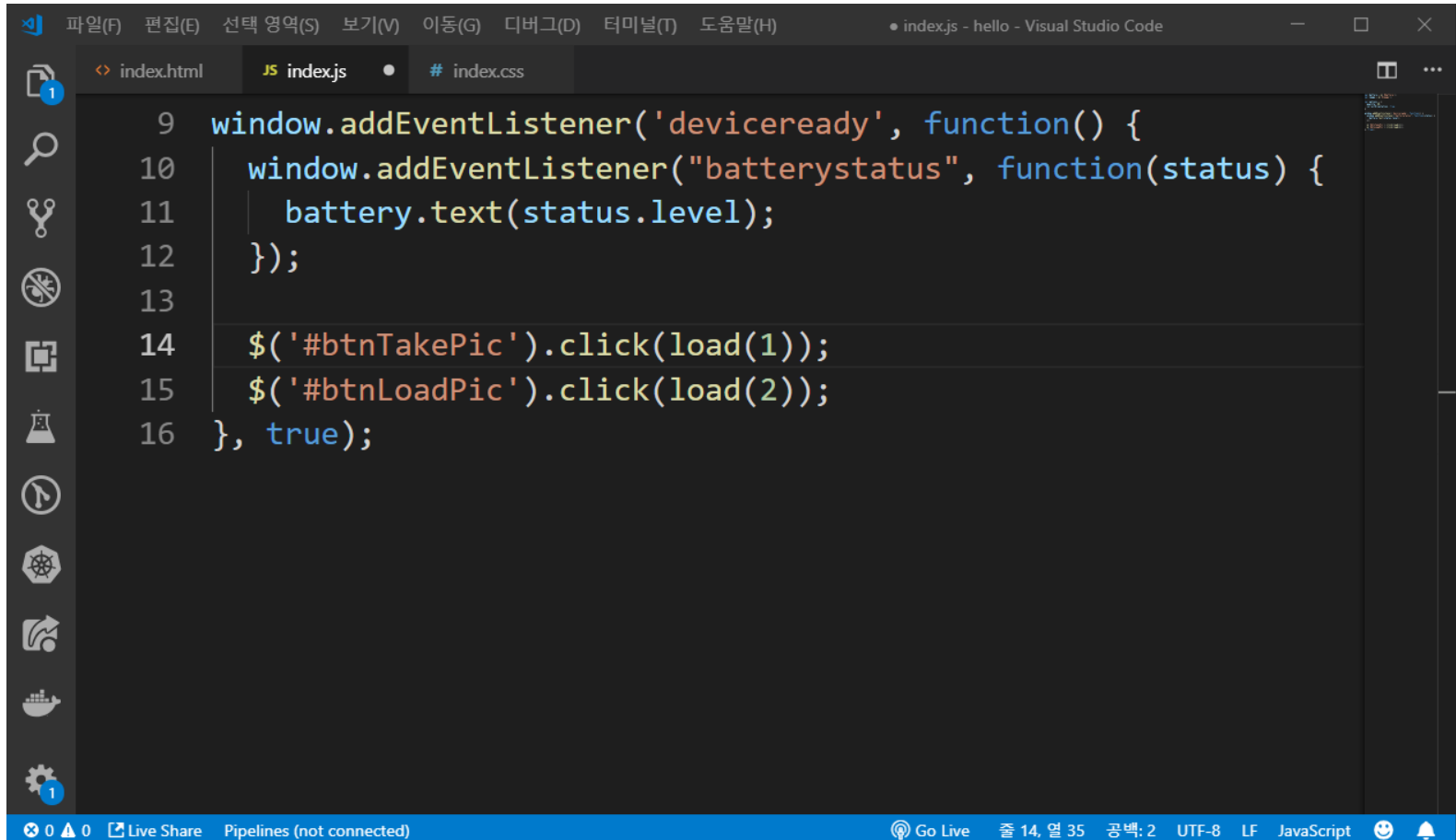


The image shows a screenshot of the Visual Studio Code editor interface. The top menu bar includes options like '파일(F)', '편집(E)', '선택 영역(S)', '보기(V)', '이동(G)', '디버그(D)', '터미널(T)', and '도움말(H)'. The title bar indicates the current file is 'index.js - hello - Visual Studio Code'. The editor has three tabs open: 'index.html', 'JS index.js', and '# index.css'. The 'JS index.js' tab is active, displaying the following JavaScript code:

```
1 var battery = $('#battery');
2 var image = $('#image');
3
4 var options = {
5   quality: 80,
6   correctOrientation: true
7 }
8
9 window.addEventListener('deviceready', function() {
10   window.addEventListener("batterystatus", function(status) {
11     battery.text(status.level);
12   });
13 }, true);
```

The bottom status bar shows '0 0 0', 'Live Share', 'Pipelines (not connected)', 'Go Live', '줄 13, 열 10', '공백: 2', 'UTF-8', 'LF', 'JavaScript', and a smiley face icon.

카메라 플러그인 사용하기

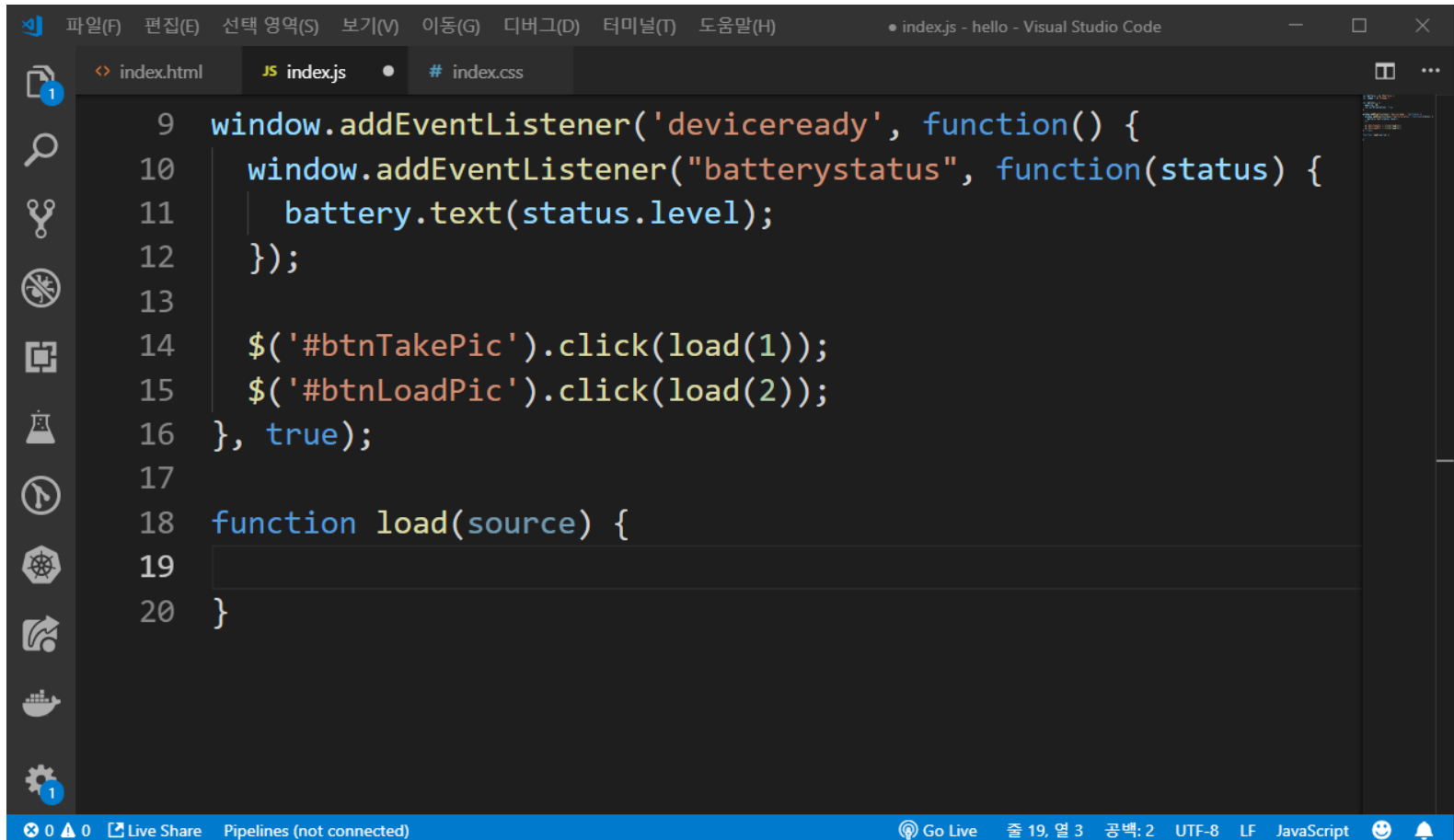


The screenshot shows the Visual Studio Code editor interface. The top menu bar includes options like '파일(F)', '편집(E)', '선택 영역(S)', '보기(V)', '이동(G)', '디버그(D)', '터미널(T)', and '도움말(H)'. The title bar indicates the current file is 'index.js - hello - Visual Studio Code'. The editor has three tabs: 'index.html', 'JS index.js', and '# index.css'. The 'JS index.js' tab is active, displaying the following JavaScript code:

```
9 window.addEventListener('deviceready', function() {
10     window.addEventListener("batterystatus", function(status) {
11         battery.text(status.level);
12     });
13
14     $('#btnTakePic').click(load(1));
15     $('#btnLoadPic').click(load(2));
16 }, true);
```

The bottom status bar shows '0 0', 'Live Share', 'Pipelines (not connected)', 'Go Live', '줄 14, 열 35', '공백: 2', 'UTF-8', 'LF', 'JavaScript', and a smiley face icon.

카메라 플러그인 사용하기

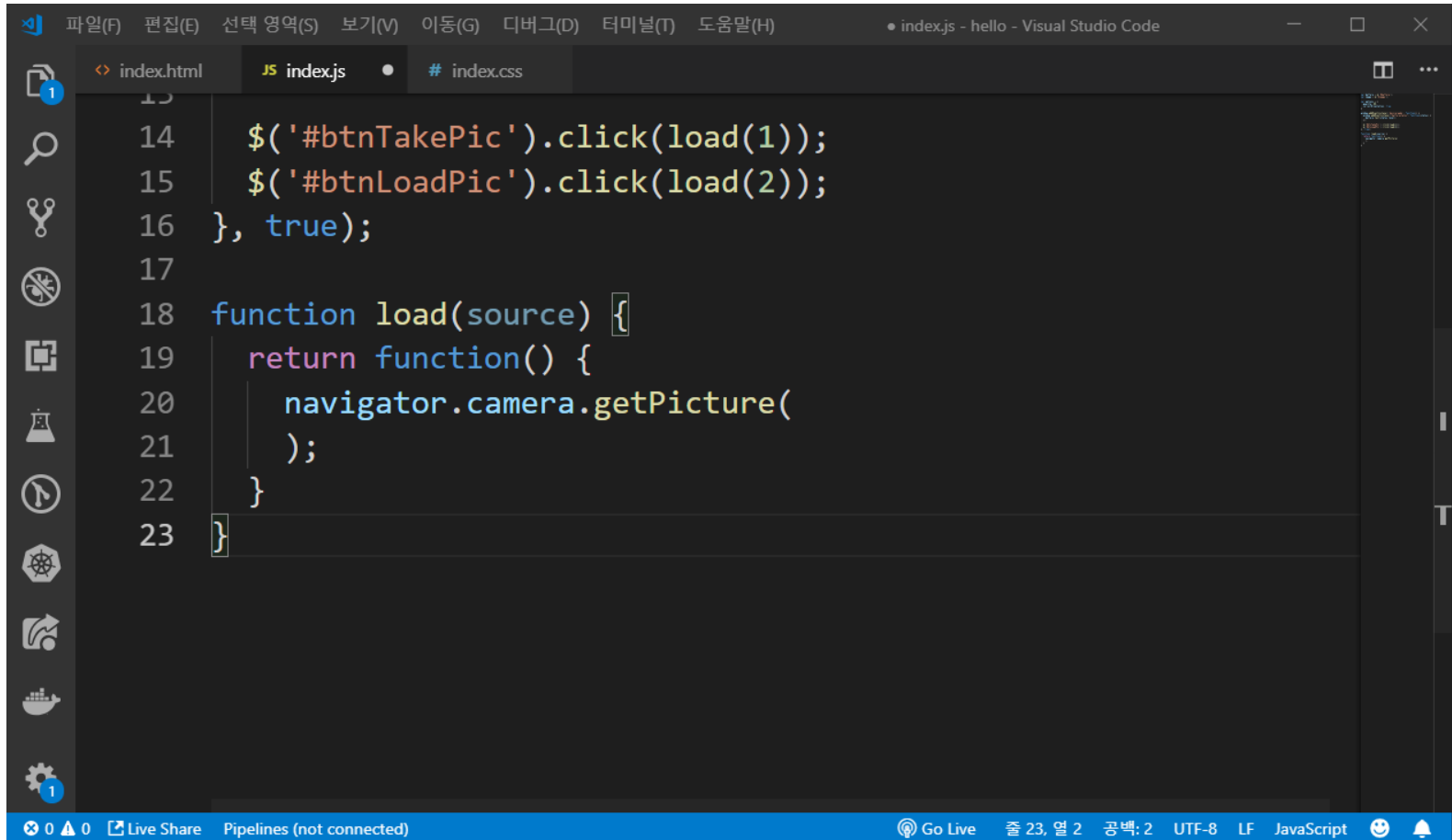


The image shows a screenshot of the Visual Studio Code editor interface. The top menu bar includes options like '파일(F)', '편집(E)', '선택 영역(S)', '보기(V)', '이동(G)', '디버그(D)', '터미널(T)', and '도움말(H)'. The title bar indicates the current file is 'index.js - hello - Visual Studio Code'. The editor has three tabs open: 'index.html', 'JS index.js', and '# index.css'. The 'JS index.js' tab is active, displaying the following JavaScript code:

```
9 window.addEventListener('deviceready', function() {
10   window.addEventListener("batterystatus", function(status) {
11     battery.text(status.level);
12   });
13
14   $('#btnTakePic').click(load(1));
15   $('#btnLoadPic').click(load(2));
16 }, true);
17
18 function load(source) {
19
20 }
```

The bottom status bar shows '0 0', 'Live Share', 'Pipelines (not connected)', 'Go Live', '줄 19, 열 3', '공백: 2', 'UTF-8', 'LF', 'JavaScript', and a smiley face icon.

카메라 플러그인 사용하기

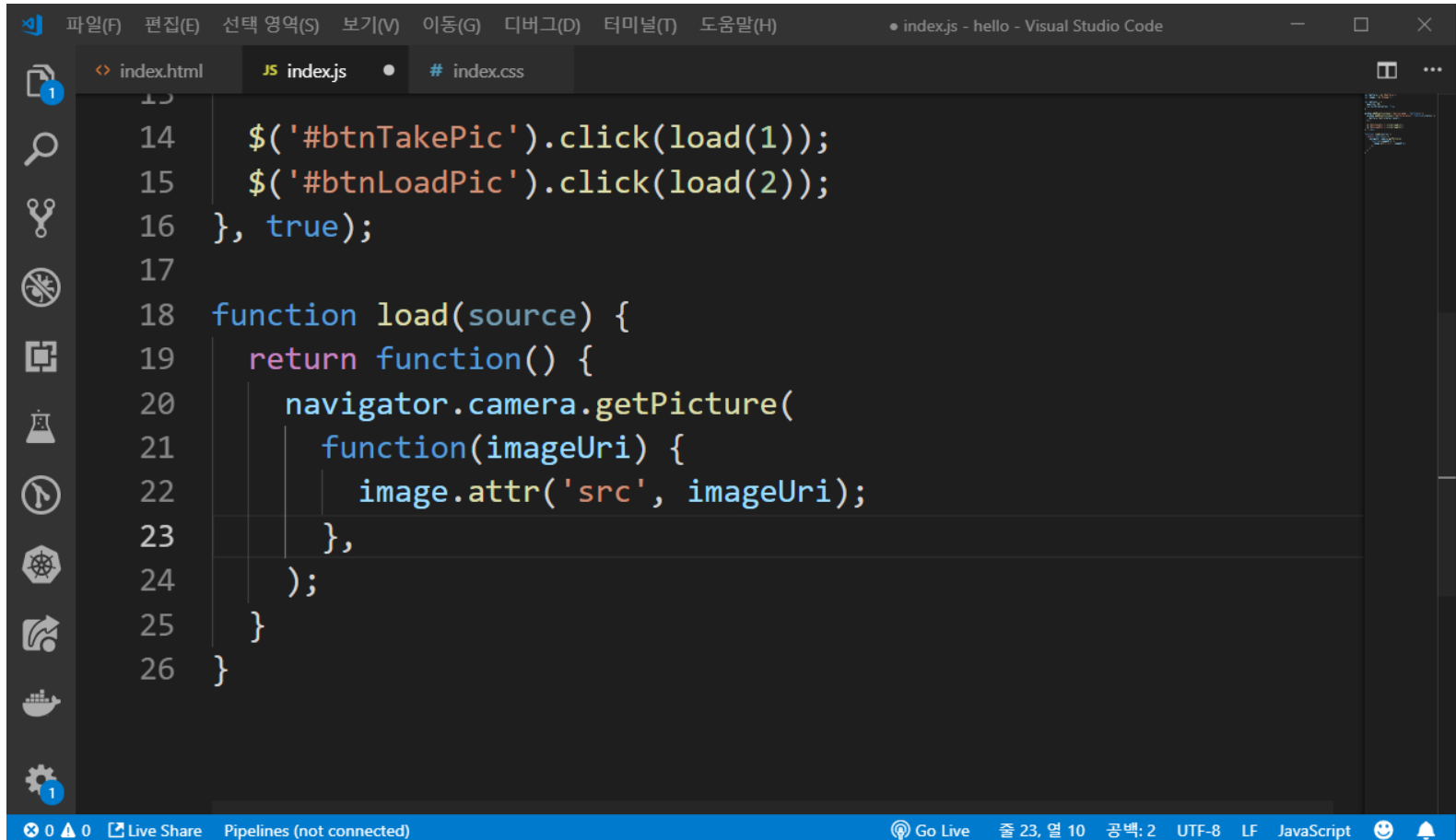


The screenshot shows the Visual Studio Code editor interface with a dark theme. The top menu bar includes options like '파일(F)', '편집(E)', '선택 영역(S)', '보기(V)', '이동(G)', '디버그(D)', '터미널(T)', and '도움말(H)'. The title bar indicates the current file is 'index.js - hello - Visual Studio Code'. The editor has three tabs open: 'index.html', 'JS index.js', and '# index.css'. The 'JS index.js' tab is active, displaying the following JavaScript code:

```
14    $('#btnTakePic').click(load(1));
15    $('#btnLoadPic').click(load(2));
16 }, true);
17
18 function load(source) {
19     return function() {
20         navigator.camera.getPicture(
21             );
22     }
23 }
```

The bottom status bar shows '0 0', 'Live Share', 'Pipelines (not connected)', 'Go Live', '줄 23, 열 2', '공백: 2', 'UTF-8', 'LF', 'JavaScript', and a smiley face icon.

카메라 플러그인 사용하기

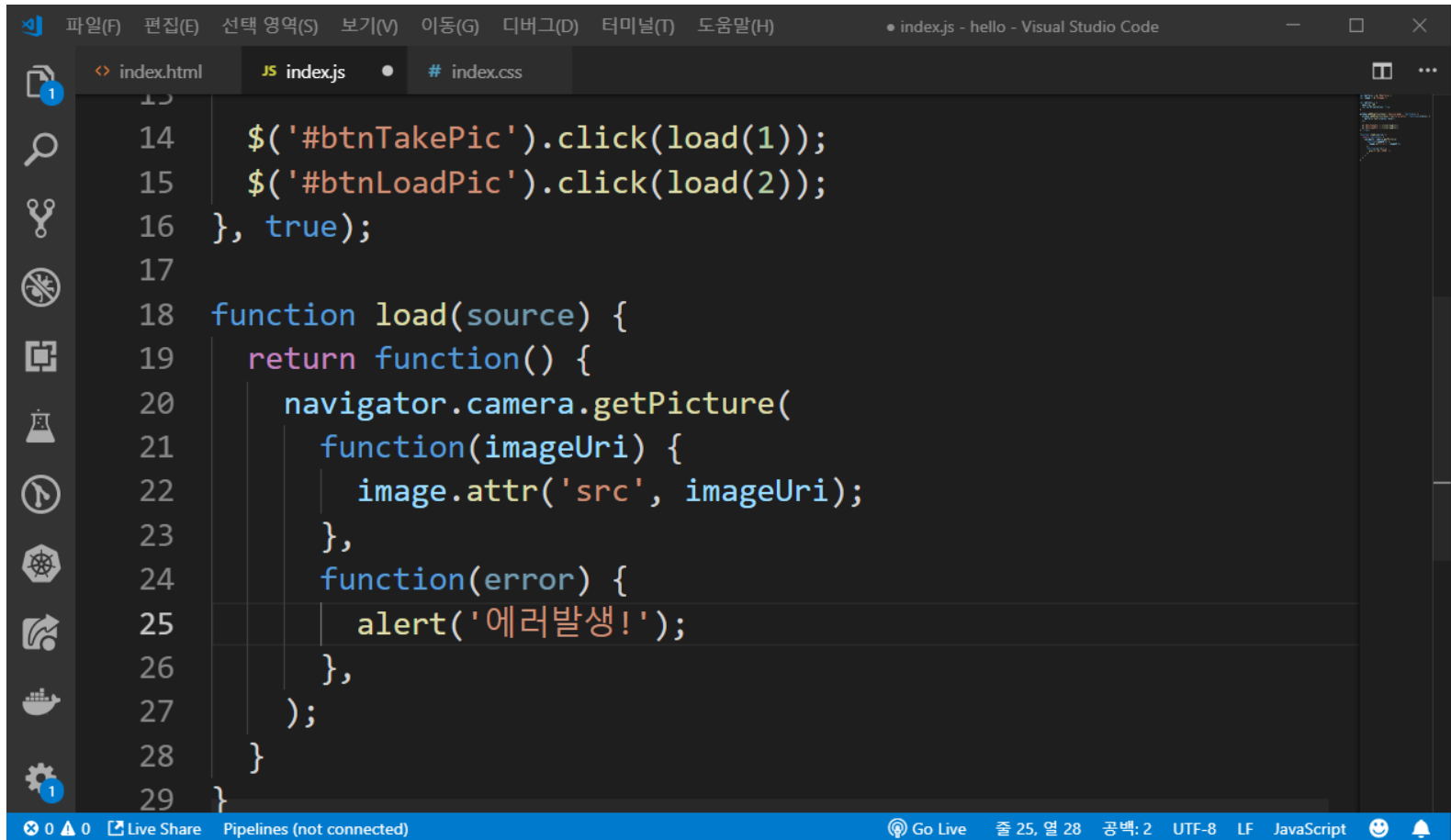


The screenshot shows the Visual Studio Code editor interface with a dark theme. The top menu bar includes options like '파일(F)', '편집(E)', '선택 영역(S)', '보기(V)', '이동(G)', '디버그(D)', '터미널(T)', and '도움말(H)'. The title bar indicates the active file is 'index.js - hello - Visual Studio Code'. The editor has three tabs: 'index.html', 'JS index.js', and '# index.css'. The 'JS index.js' tab is active, displaying the following JavaScript code:

```
14  $('#btnTakePic').click(load(1));
15  $('#btnLoadPic').click(load(2));
16  }, true);
17
18  function load(source) {
19      return function() {
20          navigator.camera.getPicture(
21              function(imageUri) {
22                  image.attr('src', imageUri);
23              },
24          );
25      }
26  }
```

The bottom status bar shows '0 0', 'Live Share', 'Pipelines (not connected)', 'Go Live', '줄 23, 열 10', '공백: 2', 'UTF-8', 'LF', 'JavaScript', and a smiley face icon.

카메라 플러그인 사용하기

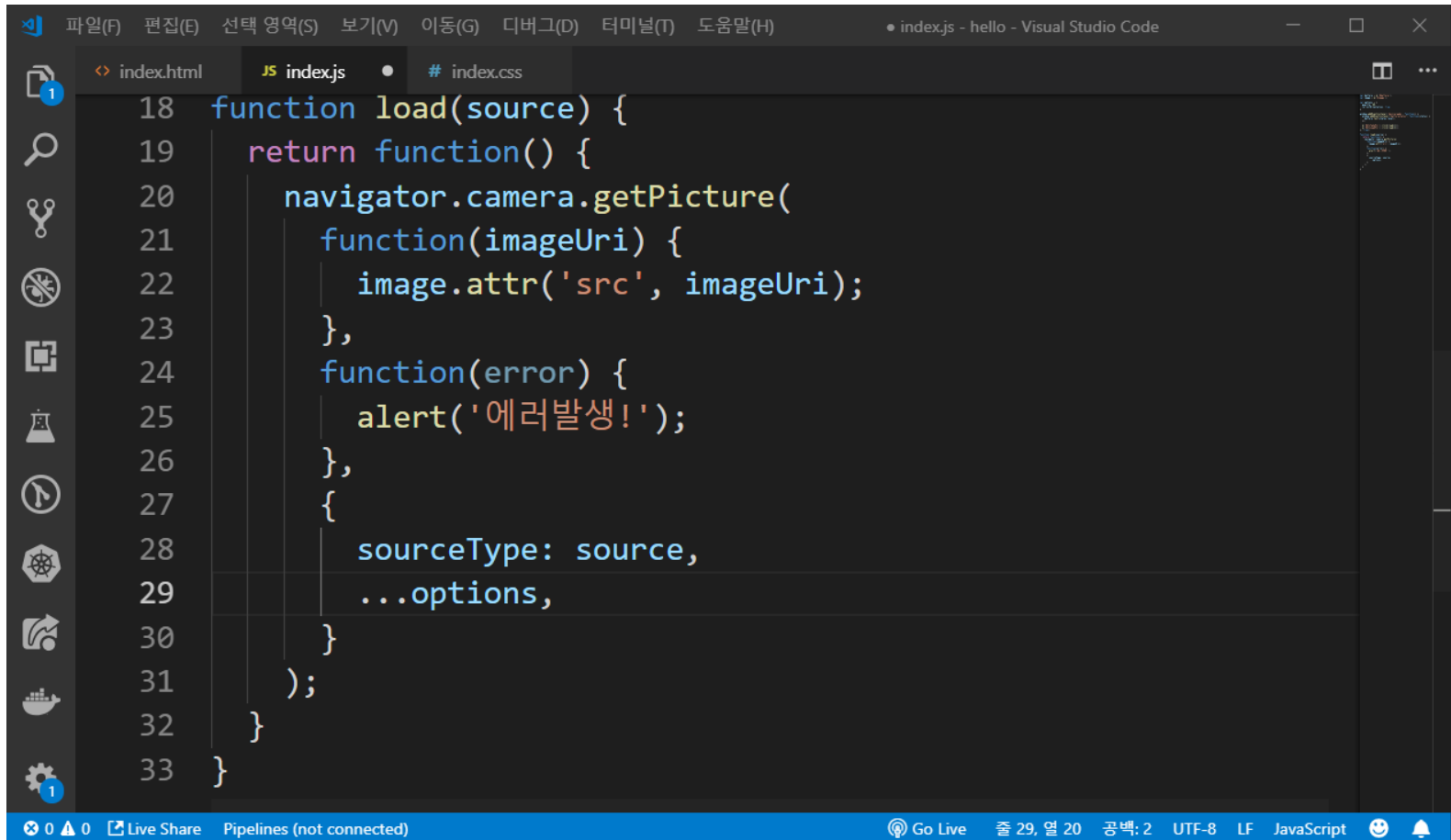


The image shows a screenshot of the Visual Studio Code editor interface. The top menu bar includes options like '파일(F)', '편집(E)', '선택 영역(S)', '보기(V)', '이동(G)', '디버그(D)', '터미널(T)', and '도움말(H)'. The title bar indicates the current file is 'index.js - hello - Visual Studio Code'. The editor has three tabs open: 'index.html', 'index.js', and 'index.css'. The 'index.js' tab is active, displaying the following JavaScript code:

```
14  $('#btnTakePic').click(load(1));
15  $('#btnLoadPic').click(load(2));
16  }, true);
17
18  function load(source) {
19      return function() {
20          navigator.camera.getPicture(
21              function(imageUri) {
22                  image.attr('src', imageUri);
23              },
24              function(error) {
25                  alert('에러발생!');
26              },
27          );
28      }
29  }
```

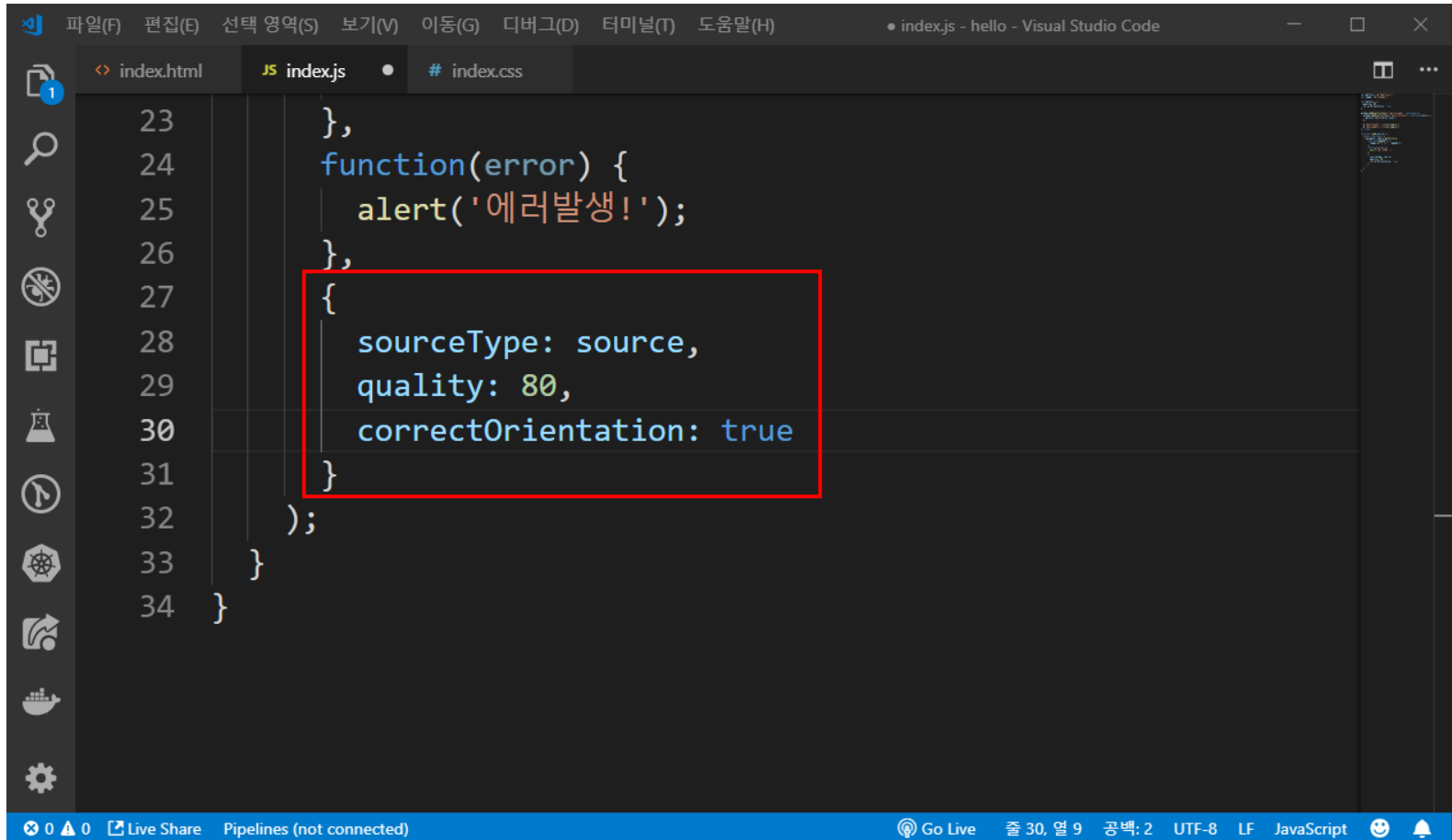
The bottom status bar shows '0 0 0', 'Live Share', 'Pipelines (not connected)', 'Go Live', '줄 25, 열 28', '공백: 2', 'UTF-8', 'LF', 'JavaScript', and a smiley face icon.

카메라 플러그인 사용하기



```
18 function load(source) {
19     return function() {
20         navigator.camera.getPicture(
21             function(imageUri) {
22                 image.attr('src', imageUri);
23             },
24             function(error) {
25                 alert('에러발생!');
26             },
27             {
28                 sourceType: source,
29                 ...options,
30             }
31         );
32     }
33 }
```

카메라 플러그인 사용하기



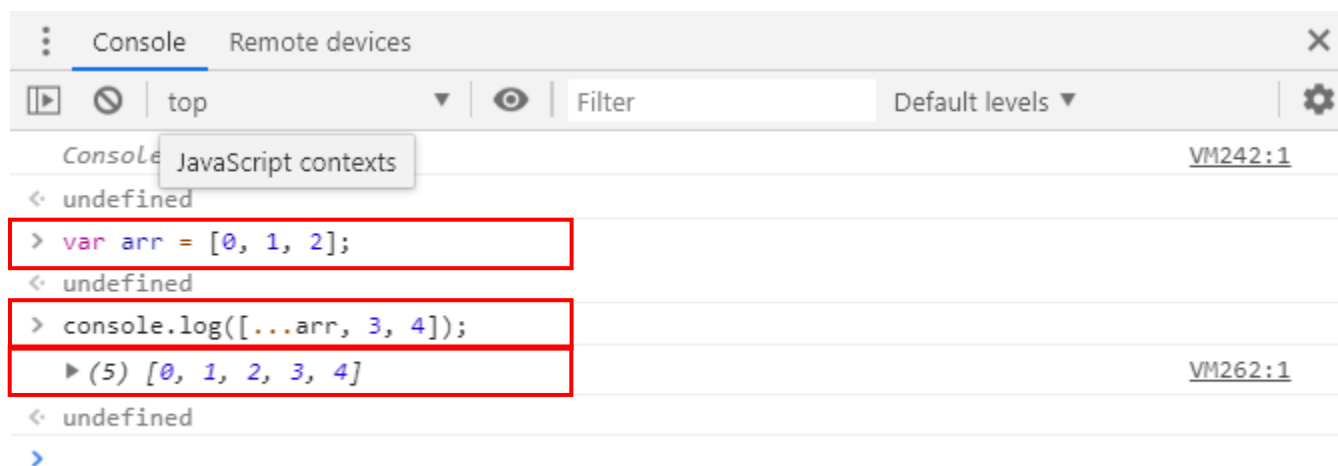
```
23     },
24     function(error) {
25         alert('에러발생!');
26     },
27     {
28         sourceType: source,
29         quality: 80,
30         correctOrientation: true
31     }
32 );
33 }
34 }
```

The screenshot shows the Visual Studio Code interface with a JavaScript file named `index.js` open. The code defines a function that takes an error callback and a camera options object. The options object, which is highlighted by a red box, contains the following properties: `sourceType` (set to `source`), `quality` (set to `80`), and `correctOrientation` (set to `true`). The status bar at the bottom indicates the file is `index.js` in the `hello` project, using `UTF-8` encoding and `LF` line endings.

펼침 연산자

- **펼침 연산자 형식**
... (점 3개)
- **배열이나 객체를 피연산자로 받음**
- **피연산자로 전달된 배열이나 객체를 펼침**
- **다른 배열이나 객체에 값을 복사해 추가할 경우 사용**

펼침 연산자

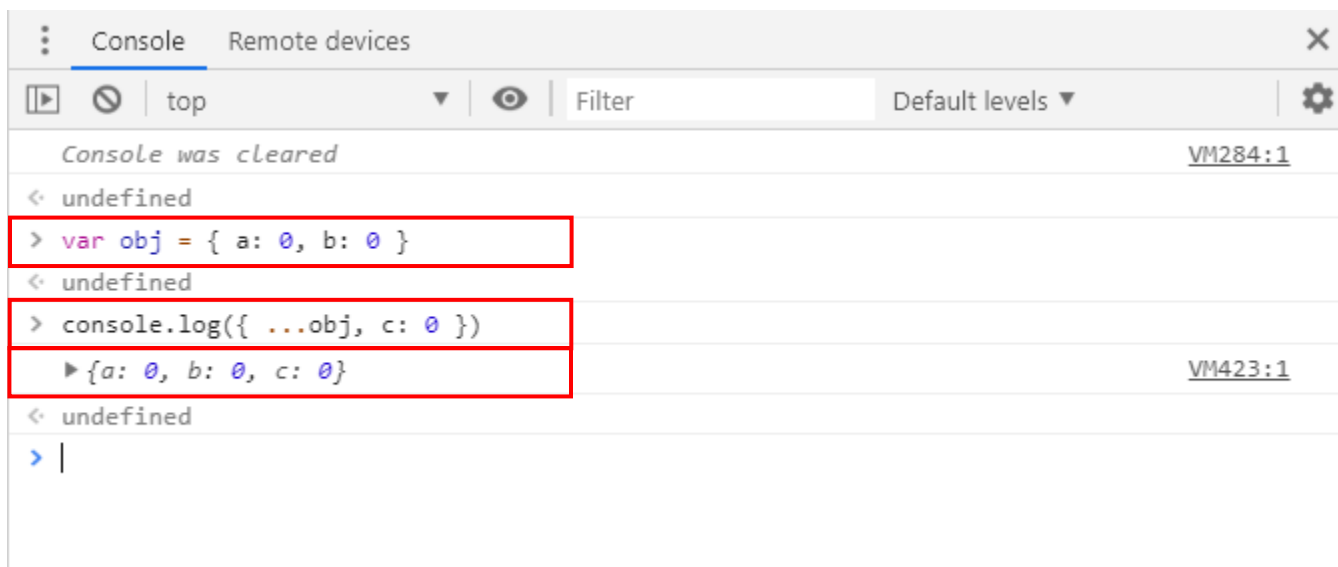


The screenshot shows a web browser's developer console with the 'Console' tab selected. The console displays the following sequence of operations:

- Initial state: `< undefined`
- Code execution: `> var arr = [0, 1, 2];` (highlighted with a red box)
- Result: `< undefined`
- Code execution: `> console.log([...arr, 3, 4]);` (highlighted with a red box)
- Output: `▶ (5) [0, 1, 2, 3, 4]` (highlighted with a red box, with `VM262:1` on the right)
- Final state: `< undefined`

A tooltip labeled 'JavaScript contexts' is visible over the console area.

펼침 연산자



The screenshot shows a web browser's developer console with the 'Console' tab selected. The console displays the following sequence of events:

- A message: *Console was cleared* (source: VM284:1).
- A prompt: `< undefined`.
- A code execution line: `> var obj = { a: 0, b: 0 }` (highlighted with a red box).
- A prompt: `< undefined`.
- A code execution line: `> console.log({ ...obj, c: 0 })` (highlighted with a red box).
- A log output: `▶ {a: 0, b: 0, c: 0}` (highlighted with a red box, source: VM423:1).
- A prompt: `< undefined`.
- A cursor: `> |`.

앱 실행

문제 출력 디버그 콘솔 터미널

LAUNCH SUCCESS

C:\Users\hansol\Downloads\hello>cordova run android

Checking Java JDK and Android SDK versions

ANDROID_SDK_ROOT=undefined (recommended setting)

ANDROID_HOME=C:\Android\sdk (DEPRECATED)

Subproject Path: CordovaLib

Subproject Path: app

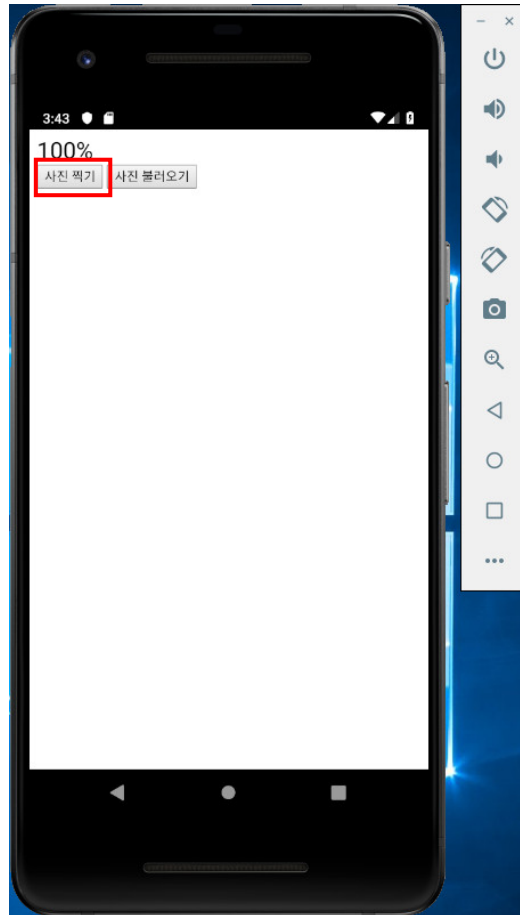
> Task :app:preBuild UP-TO-DATE

> Task :CordovaLib:preBuild UP-TO-DATE

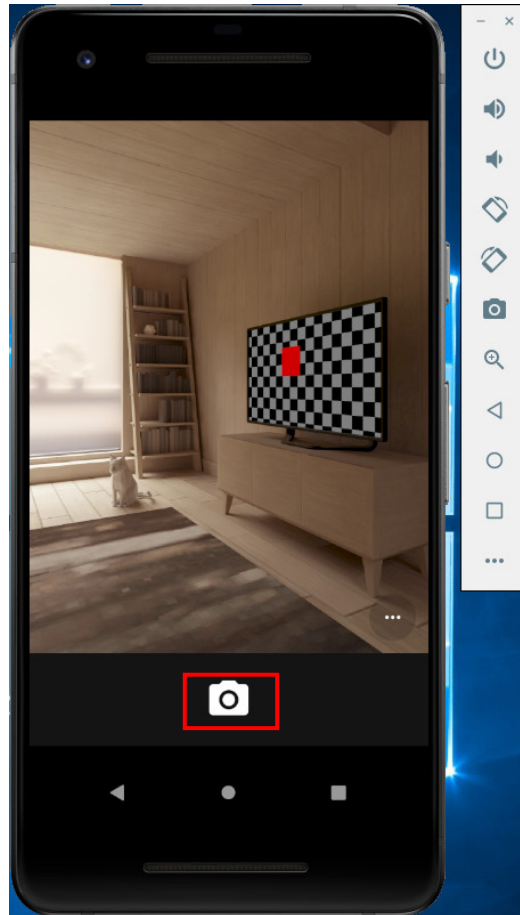
> Task :CordovaLib:preDebugBuild UP-TO-DATE

> Task :CordovaLib:checkDebugManifest UP-TO-DATE

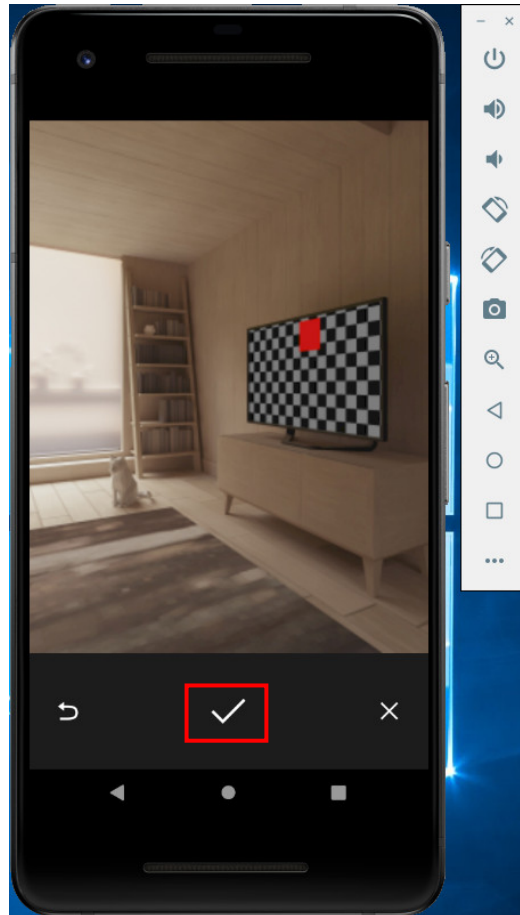
앱 실행



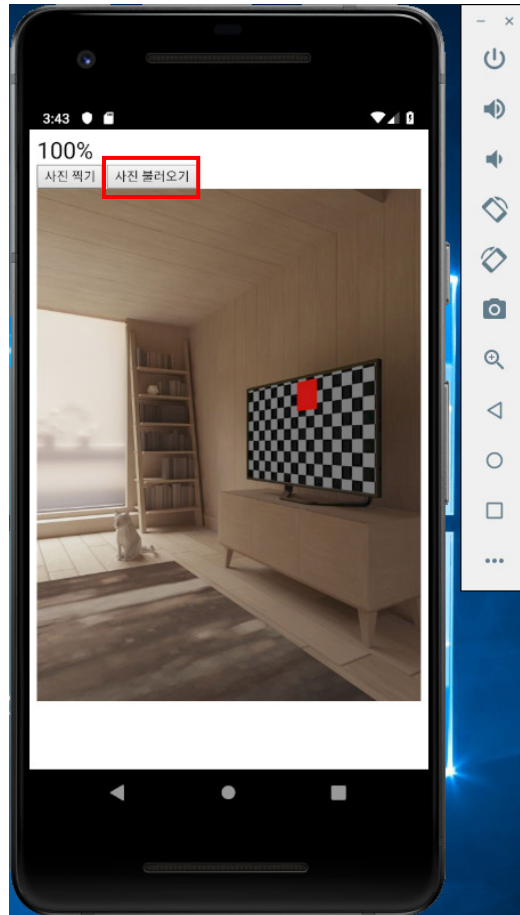
앱 실행



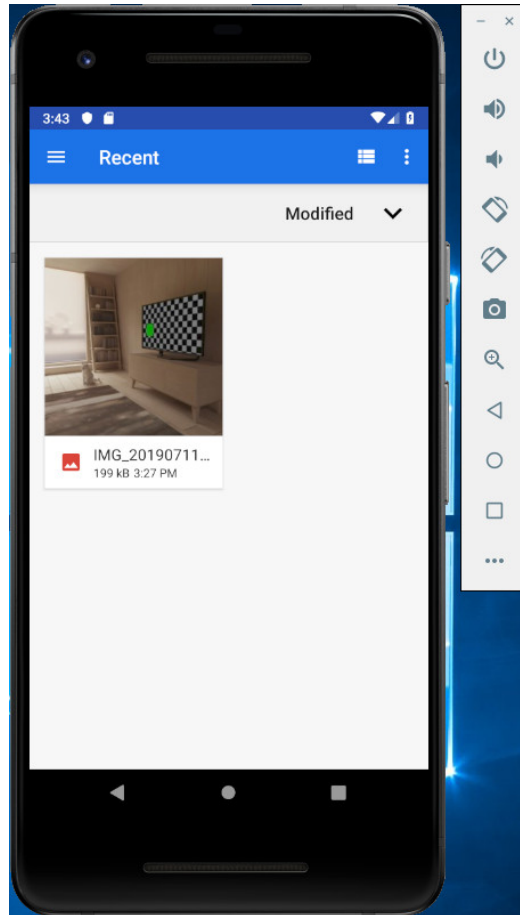
앱 실행



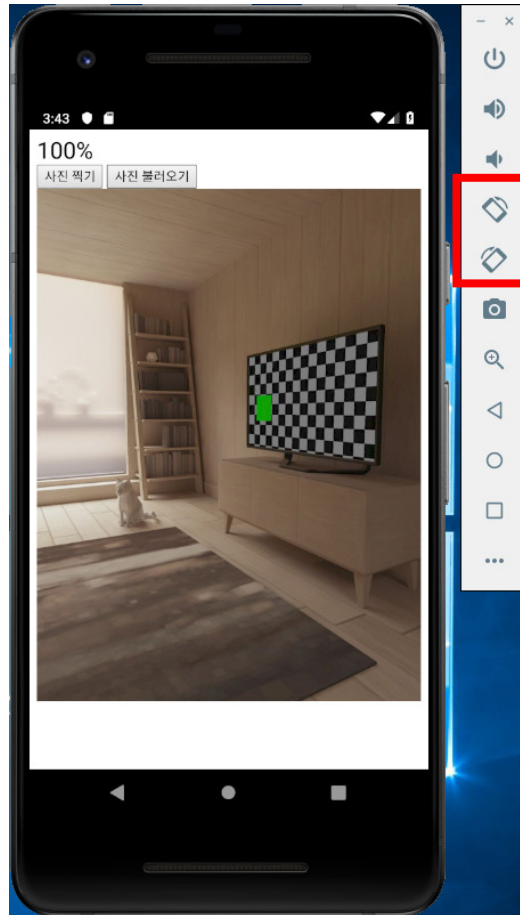
앱 실행



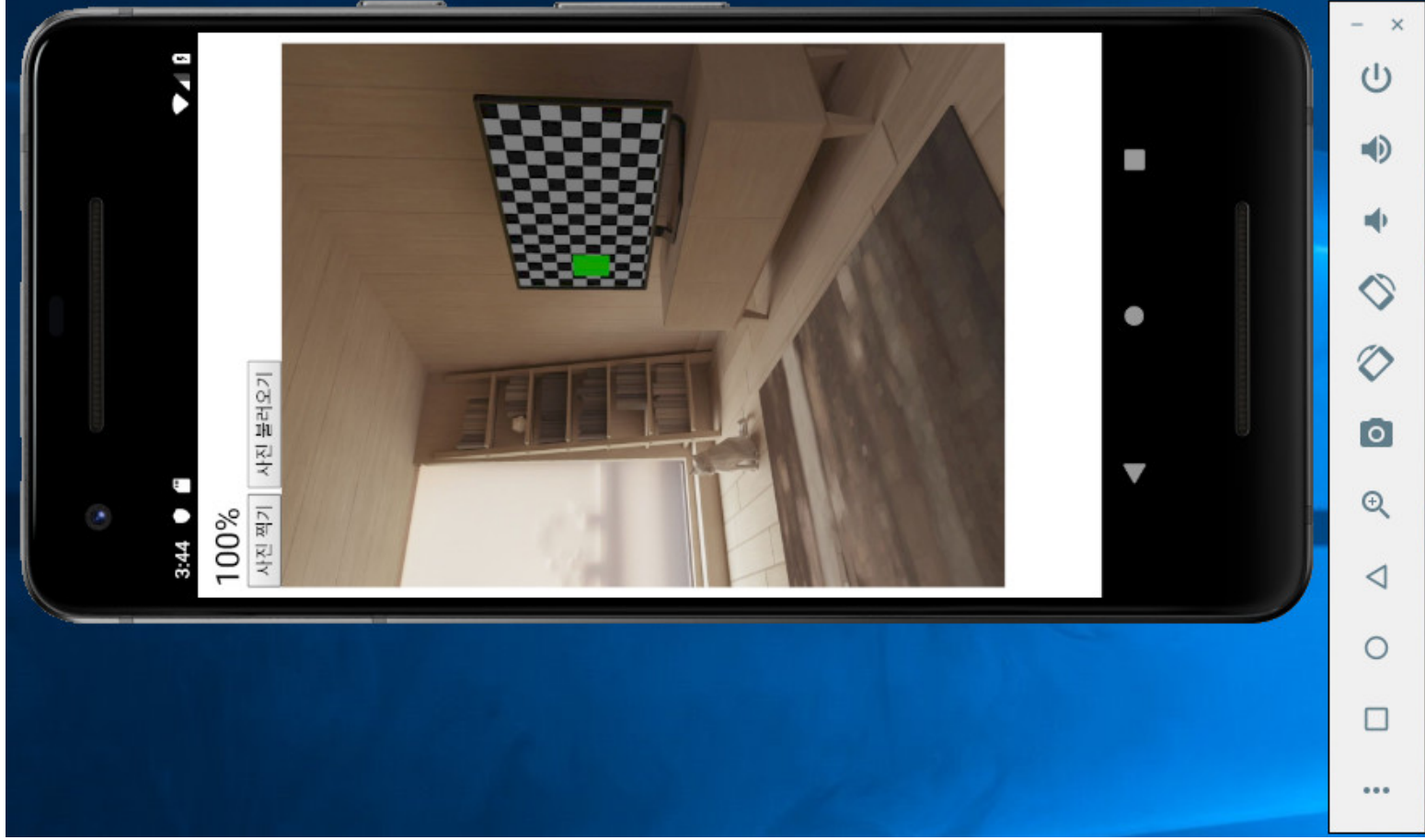
앱 실행



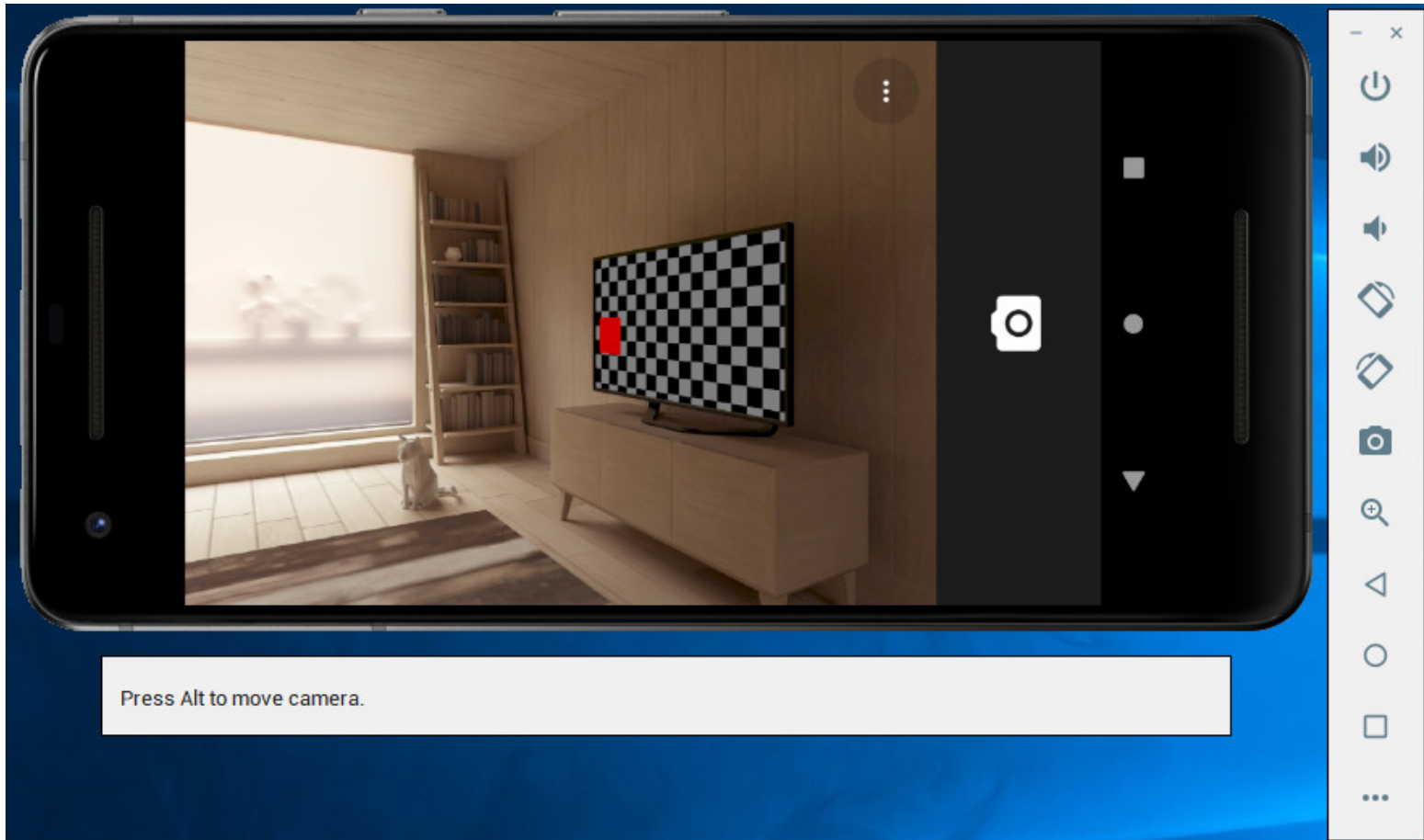
앱 실행



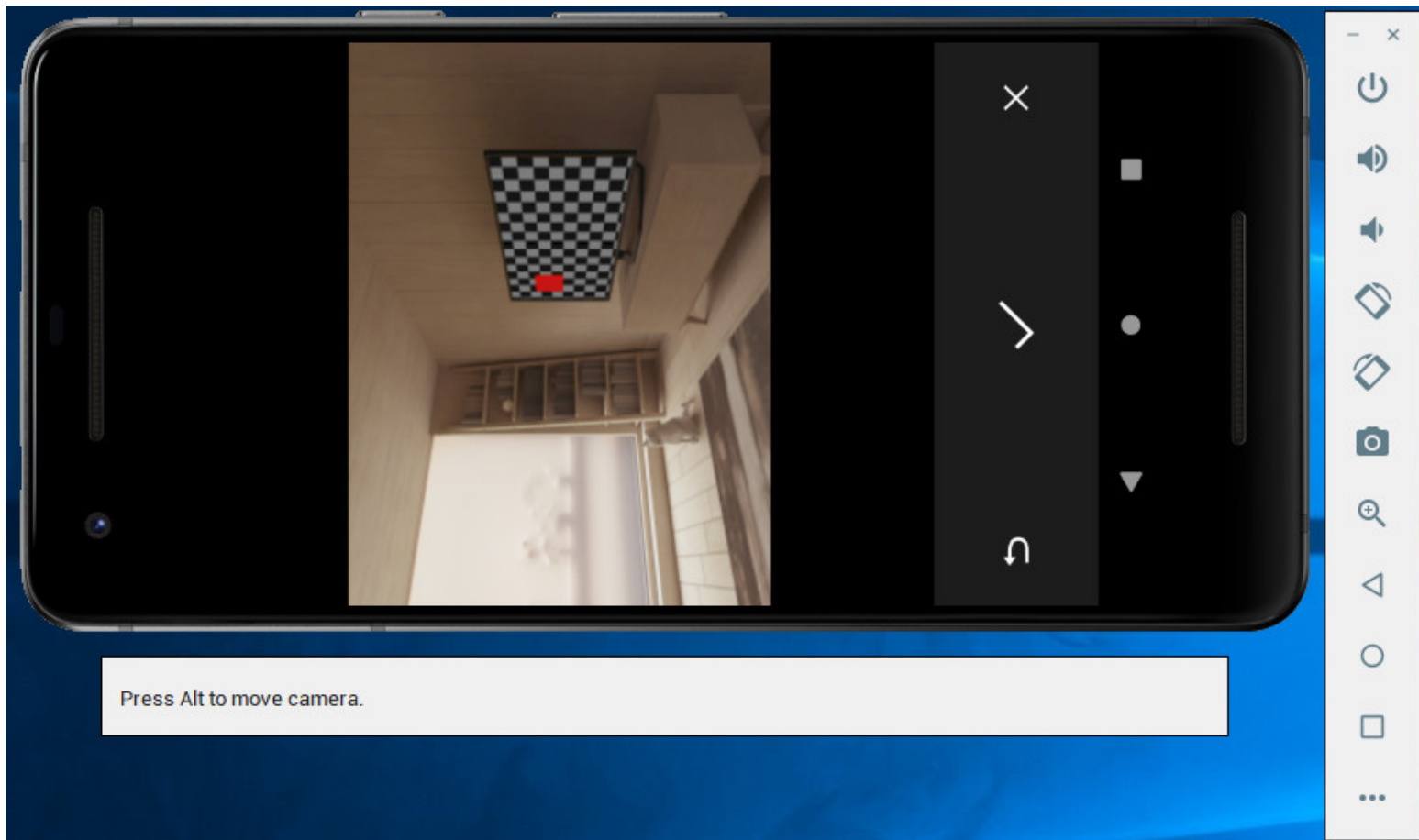
앱 실행



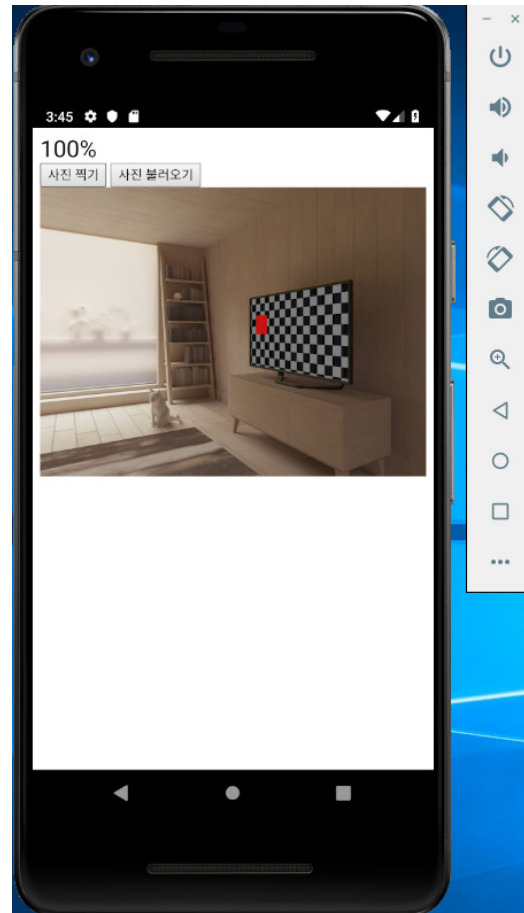
앱 실행



앱 실행



앱 실행



요약

- 코도바 플러그인 설치
- 코도바 플러그인 작동원리
- 배터리 상태 플러그인
- 카메라 플러그인

차시 예고

- **13-2 : 코도바로 todo list 앱 만들기**
 - datepicker 플러그인 추가
 - local notification 플러그인 추가

감사합니다

13주차_01 Cordova 플러그인 설치 및 사용하기