

# ckeditor #

## ckeditor? #

- 자바스크립트 오픈소스 기반 웹 에디터
- 기본적이며 가장 많이 사용하는 웹 에디터
- 설정이 간편하며 이미 보편화 되어 많이 사용하여 안전성이 입증
- 친숙한 UI 제공
- 편의성에 맞는 4가지 모드 제공
- Standard
- Full featured
- Inline editing
- Widgets
- 추가 내용은 (<https://ckeditor.com/>)에서 확인

## browser 호환성 #

### 1) CKEditor4

desktop 환경 :

Internet Explorer :

8.0 및 9.0 - 거의 완전한 지원,

10 및 11 - 완전한 지원,

Microsoft Edge의 IE 모드 - 전체 지원,

9.0 퀵 모드 및 9.0 호환 모드 - 제한된 지원.

Firefox, Chrome, Safari, Microsoft Edge, Opera :

최신 안정 릴리스 - 완전한 지원.

mobile 환경 :

Safari (iOS 6+) - 거의 완전한 지원,

Chrome (Android) - 거의 완벽하게 지원.

### 2) CKEditor5

Chrome (최신 안정 버전).

Firefox (최신 안정 릴리스).

Safari (최신 안정 릴리스).

Opera (최신 안정 릴리스).

Electron (최신 안정 방출).

Edge (최신 안정 릴리스). 알려진 문제 : 선택과 관련된 사소한 문제 : 1.

아직 지원되지 않음 :

Internet Explorer 11. IE11과의 호환성 티켓을 참조.

## 설정 #

- CKEditor 에서 다운로드
  - 기본설정
1. 사이트(<https://ckeditor.com/>)에 접속해서 원하는 버전을 선택한다.
  2. 원하는 패키지를 선택한 후, 패키지를 다운로드 받을지 CDN으로 사용할지 선택한다.
  3. 패키지로 다운로드 받았을 시 아래와 같이 ckeditor.js 파일을 포함한다.

```
<script type="text/javascript" src="프로젝트 경로/ckeditor/ckeditor.js"></script>
```

4. CDN으로 사용할 시 아래와 같이 추가한다.

```
<script src="//cdn.ckeditor.com/4.13.0/standard/ckeditor.js"></script>
```

### 1) CKEditor4

다운받은 ckeditor 스크립트를 추가한 후

1. Class 명으로 CKEditor 사용

<textarea name="ckArea" class="ckeditor" rows="2" cols="2">

2. 스크립트로 ckeditor 실행. (height 지정이 따로 가능하다.)

window.onload=function(){

CKEDITOR.replace('EDITOR가 삽입될 ID', {height: 500});

}

### 2) CKEditor5

다운받은 ckeditor 스크립트를 추가한 후

```
1. id명으로 CKEditor 사용
    <div id="editor"></div>
2. 스크립트로 ckeditor 실행
    ClassicEditor.create( document.querySelector( '#editor' ), {
        plugins: [ Essentials, Paragraph, Bold, Italic, Image, InsertImage, ImageCaption ],
        toolbar: [ 'bold', 'italic', 'insertImage' ]
    }).then(function(newEditor) {
        $scope.editor = newEditor;
    }).catch(function(error) {
        console.error( error );
    });
```

## 기능 추가 #

### 한글언어 변환 #

- 기본적으로 ckEditor의 설정은 config.js에서 할 수있다.
- 다운받은 파일의 ckeditor디렉토리에 있다(root) CKEDITOR.editorConfig = function( config ){
- 한글설정.
- 언어설정 파일경로 : ckeditor/lang config.language = 'ko';

### 이미지 업로드 기능 추가 #

- ckeditor/config.js 에 기능을 추가한다.
- image 버튼을 눌렀을 때 upload부분이 활성화된다.
- 서버로 전송을 누르면 설정해놓은 upload.ashx를 호출하여 파일이 업로드된다. 업로드가 끝나면 자바스크립트로 파라미터와 함께 callback함수를 호출해주면 이미지 정보로 callback함수에 넘긴 파라미터로 정보가 입력된다.

## 설정 옵션 #

CKEditor의 설정은 config.js 파일을 통해 조절되며 다양한 설정 옵션이 제공된다. 아래는 몇 가지 주요 설정 옵션들이다.

- language** : 편집기에서 사용할 언어를 설정한다.

```
config.language = 'en';
```

- toolbar** : 툴바에 표시될 아이콘들을 정의한다.

```
config.toolbar = [
    { name: 'basicstyles', items: ['Bold', 'Italic', 'Underline', 'Strike'] },
    { name: 'styles', items: ['Format'] },
    { name: 'paragraph', items: ['NumberedList', 'BulletedList', 'Blockquote'] },
    { name: 'links', items: ['Link', 'Unlink'] },
];
```

- removeButtons** : 기본 툴바에서 제거할 버튼을 설정한다.

```
config.removeButtons = 'Subscript,Superscript';
```

- filebrowserBrowseUrl, filebrowserUploadUrl** : 이미지 또는 파일을 업로드하기 위한 브라우저 다이얼로그의 URL을 설정한다.

```
config.filebrowserBrowseUrl = '/ckfinder/ckfinder.html';
config.filebrowserUploadUrl = '/ckfinder/core/connector/php/connector.php?command=QuickUpload&type=Files';
```

- height / width** : 에디터의 높이를 300, 너비를 100%으로 설정한다.

```
config.height = 300;
config.width = '100%';
```

- enterMode** : 엔터 키를 눌렀을 때의 동작을 설정한다.(CKEDITOR.ENTER\_P, CKEDITOR.ENTER\_BR, CKEDITOR.ENTER\_DIV 등)

```
config.enterMode = CKEDITOR.ENTER_P;
```

- extraPlugins** : 추가 플러그인을 로드한다.

```
config.extraPlugins = 'image2,wordcount,autogrow';
```

이 외에도 많은 설정 옵션이 있으며, 필요에 따라 CKEditor 공식 문서(<https://ckeditor.com/docs/>)를 참조하여 사용하면 된다.

## 플러그인 #

- CKEditor에는 다양한 플러그인 및 기능이 포함되어 있다. 아래와 같은 코드를 사용해서 ckeditor에서 사용 가능한 '플러그인 목록'을 간편하게 확인할 수 있다.

```
ClassicEditor.builtinPlugins.map( plugin => plugin.pluginName );
```

- 또한, 원하는 플러그인을 각각 다운받아 ckeditor의 plugins 폴더 안에 넣고 config.js안에 아래와 같은 코드를 입력하여 사용할 수 있다.

```
config.extraPlugins = 'plugin이름'; // 하나일 경우
config.extraPlugins = 'plugin이름1, plugin이름2, plugin이름3'; // 여러 개일 경우 (공백없이)
```

## 플러그인 종류 #

| 종류          | 내용  | 사용법                                  |
|-------------|---|--------------------------------------|
| Autogrow    | 에디터의 크기를 동적으로 조절하여 콘텐츠에 맞게 자동으로 늘어나게 하는 플러그인      | config.extraPlugins = 'autogrow';    |
| Emoji       | 이모지를 에디터에 쉽게 추가할 수 있는 플러그인                        | config.extraPlugins = 'emoji';       |
| CodeSnippet | 코드 조각을 삽입하고 편집할 수 있는 기능을 추가할 수 있는 플러그인            | config.extraPlugins = 'codesnippet'; |
| AutoCorrect | 일반적인 오타를 자동으로 교정하는 플러그인                           | config.extraPlugins = 'autocorrect'; |
| Table       | 더 나은 표 관리를 위한 플러그인으로 표 삽입, 행 및 열 조작 등을 향상시키는 플러그인 | config.extraPlugins = 'table';       |
| MediaEmbed  | 비디오 및 오디오 등의 미디어 콘텐츠를 에디터에 쉽게 삽입할 수 있는 플러그인       | config.extraPlugins = 'mediaembed';  |

## 플러그인 제거 #

- 미리 정의된 CKEditor5 빌드에는 배포 패키지에 기본으로 포함된 모든 기능이 기본적으로 활성화되어 있다.
- 이러한 기능들은 CKEditor5의 플러그인으로 정의되어 있는데, 어떤 경우에는 동일한 빌드를 기반으로 한 응용 프로그램에서 서로 다른 편집기 설정을 원할 수 있다.
- 이를 위해 편집기에서 사용할 수 있는 플러그인을 제어해야 한다.
- [ Heading 플러그인이 제거 예시 ]

```
ClassicEditor
  .create( document.querySelector( '#editor' ), {
    removePlugins: [ 'Heading' ],
    toolbar: [ 'bold', 'italic', 'bulletedList', 'numberedList', 'blockQuote' , 'link' ]
  } )
  .catch( error => {
    console.log( error );
  } );
```

- [ Link 플러그인 제거 예시 ]

```
ClassicEditor
  .create( document.querySelector( '#editor' ), {
    removePlugins: [ 'Heading', 'Link' ],
    toolbar: [ 'bold', 'italic', 'bulletedList', 'numberedList', 'blockQuote' ]
  } )
  .catch( error => {
    console.log( error );
  } );
```

- [ 주의 ]

```
CKEditorError: plugincollection-required {"plugin":"Link","requiredBy":"CKFinder"}`
Read more: [https://ckeditor.com/docs/ckeditor5/latest/support/error-codes.html#error-plugincollection-required] (http
s://ckeditor.com/docs/ckeditor5/latest/support/error-codes.html#error-plugincollection-required)
```

그러나 CKEditor 5의 공식 클래식 빌드와 함께 코드 스니펫을 사용하면 브라우저 콘솔에 위와 같은 오류가 발생할 것이다.  
이는 일부 플러그인이 서로 의존하는 경우가 있다는 것인데, 이 경우에는 CKFinder 플러그인이 Link 플러그인이 있어야 한다는 것을 의미한다.  
위 코드 스피넷을 원하는 대로 작동시키려면 CKFinder 플러그인도 삭제해 줘야 한다.

```
ClassicEditor
  .create( document.querySelector( '#editor' ), {
    removePlugins: [ 'Heading', 'Link', 'CKFinder' ],
    toolbar: [ 'bold', 'italic', 'bulletedList', 'numberedList', 'blockQuote' ]
  } )
  .catch( error => {
    console.log( error );
  } );
```

- config.removePlugins를 사용하여 CKEditor 빌드에서 플러그인을 제거할 때는 주의가 필요하다. 제거된 플러그인이 툴바 버튼을 제공했던 경우, 빌드에 포함된 기본 툴바 구성은 무효가 되므로, 위의 예시처럼 업데이트된 툴바 구성을 제공하거나 config.toolbar.removeItems를 사용하여 제거할 툴바 항목만 제공해야 한다.