

API documentation

인스턴스 생성 및 Canvas Element 반환

```
<body>
  <div class="editor-wrap"></div>
</body>

import Editor from "../editor.js";

const editorWrap = document.querySelector(".editor-wrap");

const editor: Editor = new Editor();

const canvasElement: HTMLCanvasElement = editor.getCanvas();

editorWrap.appendChild(canvasElement);
```

Methods

인스턴스 전역 메소드

getCanvas

생성된 ELEMENT를 반환합니다.

```
const canvasElement: HTMLCanvasElement = editor.getCanvas();
```

setImgUrl

문서 이미지를 설정합니다.

```
const url: string = `https://example.com/img.png`;

editor.setImgUrl(url);
```

draw

설정된 데이터를 기준으로 화면을 그립니다.

```
editor.draw();
```

setCalculatedScale

상위 Element를 기준으로 좌우폭 또는 상하 맞춤을 실행합니다.

ex) 이미지가 모두 로드된 후 실행해야 합니다.

```
type FitType = "width" | "height";
const fit = "width";

editor.setCalculatedScale(fit);
```

setFields

전체 필드 데이터를 입력합니다.

```
type Field = {
  id: string;
  text: string;
  dx: number;
  dy: number;
  dWidth: number;
  dHeight: number;
  type: "confirm" | "miss" | "error" | "new";
};

const Fields: Field[] = [
  {
    id: "id",
    text: "text",
    dx: 0,
    dy: 0,
    dWidth: 1,
    dHeight: 1,
    type: "confirm",
  },
  ...n,
];

editor.setFields(Fields);
editor.draw();
```

setField

하나의 필드 데이터를 입력합니다.

```
type Field = {
  id: string;
  text: string;
  dx: number;
  dy: number;
```

```

    dWidth: number;
    dHeight: number;
    type: "confirm" | "miss" | "error" | "new";
};

const Field: Field = {
  id: "id",
  text: "text",
  dx: 0,
  dy: 0,
  dWidth: 1,
  dHeight: 1,
  type: "confirm",
};
editor.setField(field);
editor.draw();

```

modifyField

특정 필드 데이터를 수정합니다.

```

type Field = {
  id: string;
  text: string;
  dx: number;
  dy: number;
  dWidth: number;
  dHeight: number;
  type: "confirm" | "miss" | "error" | "new";
};

const Field: Field = {
  id: "id",
  text: "text",
  dx: 0,
  dy: 0,
  dWidth: 1,
  dHeight: 1,
  type: "confirm",
};
editor.modifyField(field);
editor.draw();

```

setDraw

새로운 필드를 드로우합니다.

```
editor.setDraw();
```

getFields

모든 필드정보를 얻습니다.

```
type Field = {  
  id: string;  
  text: string;  
  dx: number;  
  dy: number;  
  dWidth: number;  
  dHeight: number;  
  type: "confirm" | "miss" | "error" | "new";  
};  
  
const fields: Field[] = editor.getFields();
```

setEditField

수정할 필드정보를 설정 합니다.

```
const id: string = "id";  
editor.setEditField(id);  
editor.draw();
```

getEditField

현재 수정 필드정보를 얻습니다.

```
type Field = {  
  id: string;  
  text: string;  
  dx: number;  
  dy: number;  
  dWidth: number;  
  dHeight: number;  
  type: "confirm" | "miss" | "error" | "new";  
};  
  
const field: Field = editor.getEditField();
```

clearEditField

수정할 필드정보를 설정을 삭제합니다.

```
editor.clearEditField();
```

removeField

특정 필드 데이터를 삭제합니다.

```
const id: string = "id";
editor.removeField(id);
```

removeFields

전체 필드 데이터를 삭제합니다.

```
editor.removeFields();
```

setZoomInOut

Scale을 조절합니다.

```
const depth: number = 110; // 백분을 default 100
editor.setZoomInOut(depth);
```

setRotate

이미지를 회전 조절합니다.

```
const deg: number = 90;
editor.setRotate(deg);
```

getMargin

이미지 마진값을 얻습니다.

```
const margin = editor.getMargin();
```

setIsReadOnly

읽기 전용으로 변경합니다.

```
const isReadOnly: boolean = true;
editor.setIsReadOnly(isIdx);
editor.draw();
```

setIsText

Text 값을 표시합니다.

```
const isText: boolean = true;
editor.setIsText(isText);
editor.draw();
```

setIsIdx

인덱스 값을 표시합니다.

```
const isIdx: boolean = true;
editor.setIsIdx(isIdx);
editor.draw();
```

setMaxScale

최대 Scale 값을 설정합니다.

ex) 백분율

```
const s: number = 200;
editor.setMaxScale(s); // default 150
```

setMinScale

최소 Scale 값을 설정합니다.

ex) 백분율

```
const s: number = -100;
editor.setMinScale(s); // default -50
```

setScale

현재 Scale 값을 얻습니다..

```
const scale = editor.setScale() * 100;
```

loadImage

이미지를 다운로드합니다.

```
editor.downloadImage();
```

addEventListener

각 이벤트 콜백을 설정합니다.

```
type Event = "imgLoaded" | "draw" | "resize" | "selected";  
type DrawCallback = (field?: Field) => void;  
  
const event: Event = "imgLoaded";  
const callback: DrawCallback = () => {};  
  
editor.addEventListener(event, callback);
```