

FECONF

그리드 기반 에디터 / 빌더 구현기

이선협 @kciter

발표자 소개

- 이선협(@kciter)
- 12년차 개발자
- 마플코퍼레이션, 플랫폼 엔지니어 리드
- <https://github.com/kciter>
- <https://kciter.so>



이번 발표는...

- 전반적으로 그리드 기반 에디터를 만들며 겪었던 **시행착오**를 다뤄요
 - 그리드 기반 에디터가 무엇이고 왜 선택했는지?
 - 어떻게 시작해야 할지?
 - 어떻게 구현하면 좋을지?
 - 어떤 문제가 있을 수 있는지?



⚠️ 생각보다 30분이 많이 짧아요!

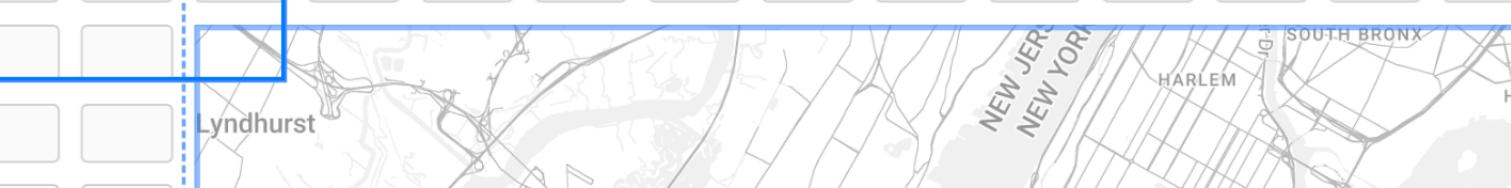
만들면서 느꼈던 꼭 알아야 하는 부분만 빠르게 설명하겠습니다

그리드 기반 에디터



Providing you with a safe space to relax

All your appointments will take place in our soothing office environment with fully private rooms, complimentary tea and water, and individualized care.



왜 굳이 그리드 기반 에디터를 사용했을까요?





어떻게 구현해야하지?

보통 우리에게 익숙한 에디터 종류는 세 가지

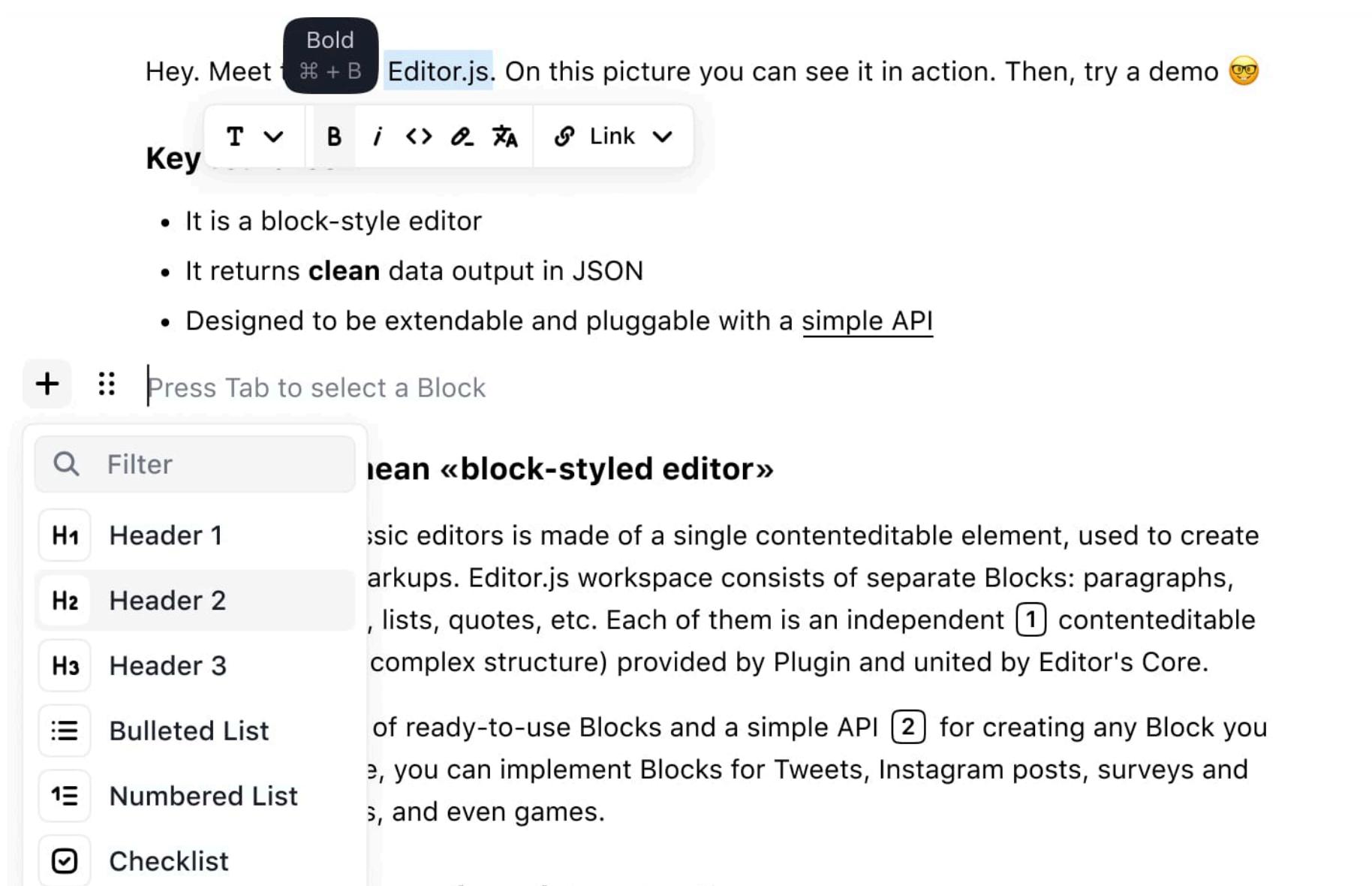
Rich text
based

Block
based

Position
based

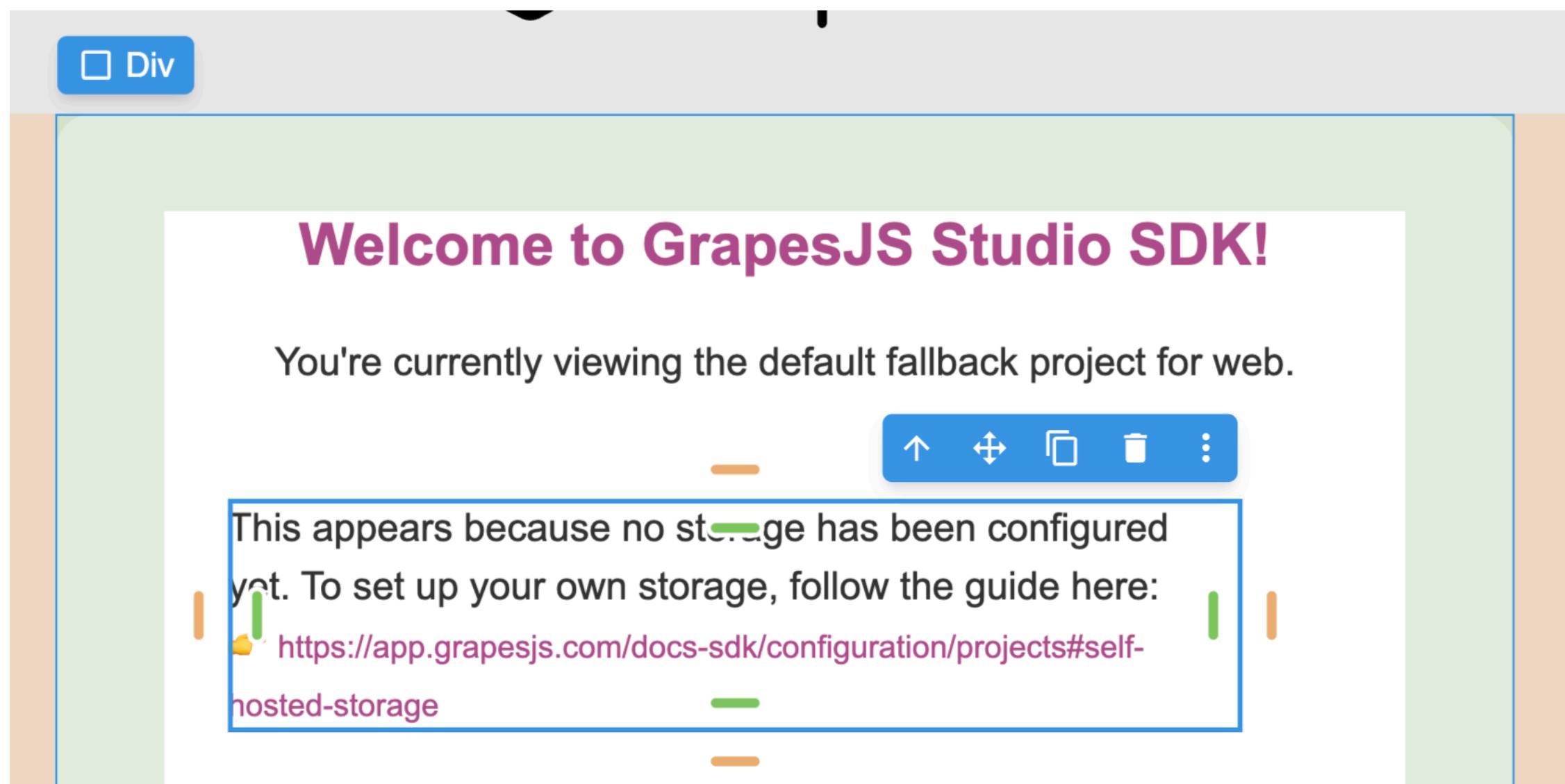
리치 텍스트 기반 에디터

- 리치 텍스트 기반 에디터는 **텍스트 입력을 처리하기에 좋아요**
- 행 단위를 '블럭' 단위로 처리하는 경우도 있어요
 - Notion이 대표적이에요
- 이를 이용하여 웹 페이지를 만드는 경우도 있어요



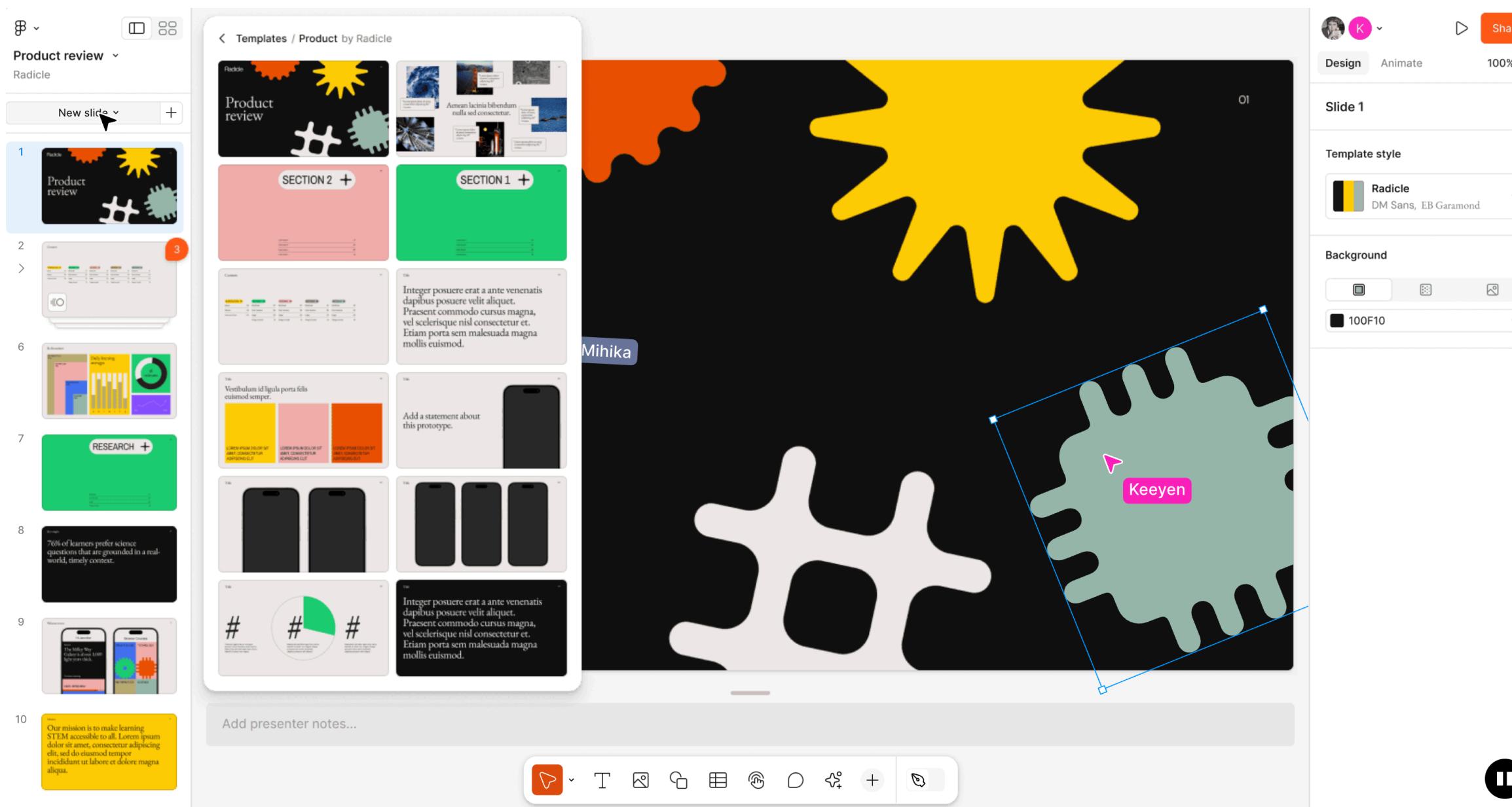
블럭 기반 에디터

- 블럭 기반 에디터는 웹의 동작대로 처리하기에 좋아요
- 웹 빌더 서비스가 이 방식을 사용하는 경우가 있어요
- 사용이 어렵지만 HTML과 CSS를 이해하고 있다면 디자인 결과물을 거의 비슷하게 구현할 수 있어요



포지션 기반 에디터

- 원하는 위치에 요소를 배치할 수 있는 에디터에요
- 디자인 작업, 도식을 그릴 때 많이 사용해요
- 제약 없이 자유롭게 배치할 수 있어요

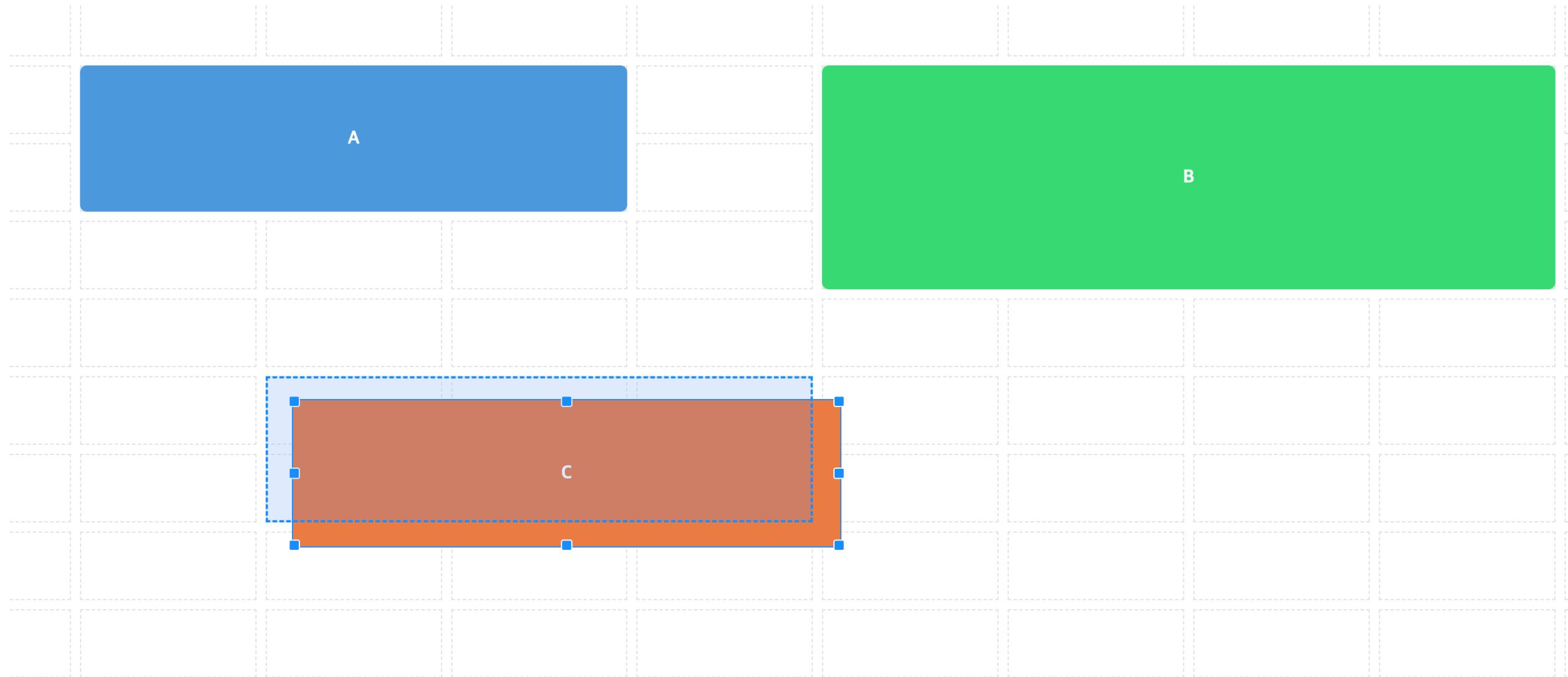


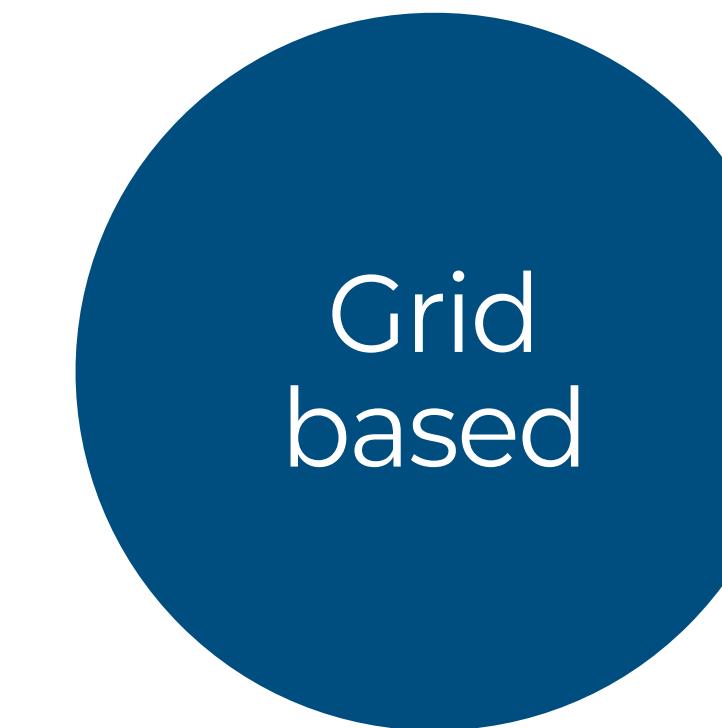
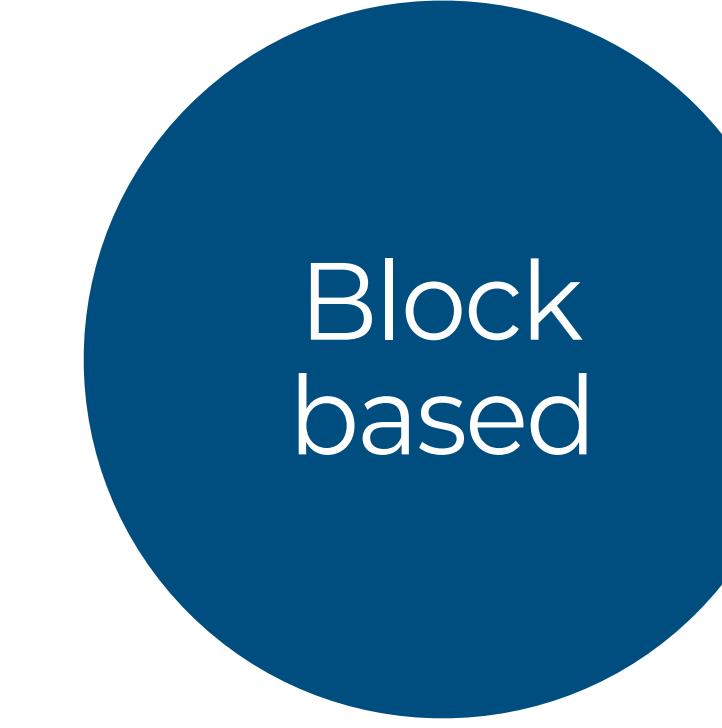
조금 더 범용적으로 쓸 수는 없을까?

- 각 에디터의 쓰임새는 아주 명확해서 벗어나면 불편해요
 - 리치 텍스트 기반 에디터는 문서라는 틀을 벗어나면 만들기 복잡해져요
 - 블럭 기반 에디터는 HTML과 CSS를 잘 모르면 사용하기 어려워요
 - 포지션 기반 에디터는 편하지만 웹 사이트를 구성하기에 적합하지 않아요

그리드 기반 에디터

- 그리드 기반 에디터는 행과 열로 나눈 그리드에 요소를 배치하는 방식을 사용해요
- 포지션 기반 에디터와 비슷해 보이지만 위치 정렬과 리사이징에 대응하기 더 쉬워요
- Squarespace, Retool 등 다양한 업체가 이 방식으로 웹 페이지 빌더를 제공하고 있어요





높은 자유도

직관적인 규칙

어려운 구현

문서 작성 가능

예측 가능한 결과

겹침 불가능

웹 규칙 기반

확장성 높음

어려운 규칙

직관적인 규칙

쉬운 구현

어설픈 자유도

그리드 기반 시작하기

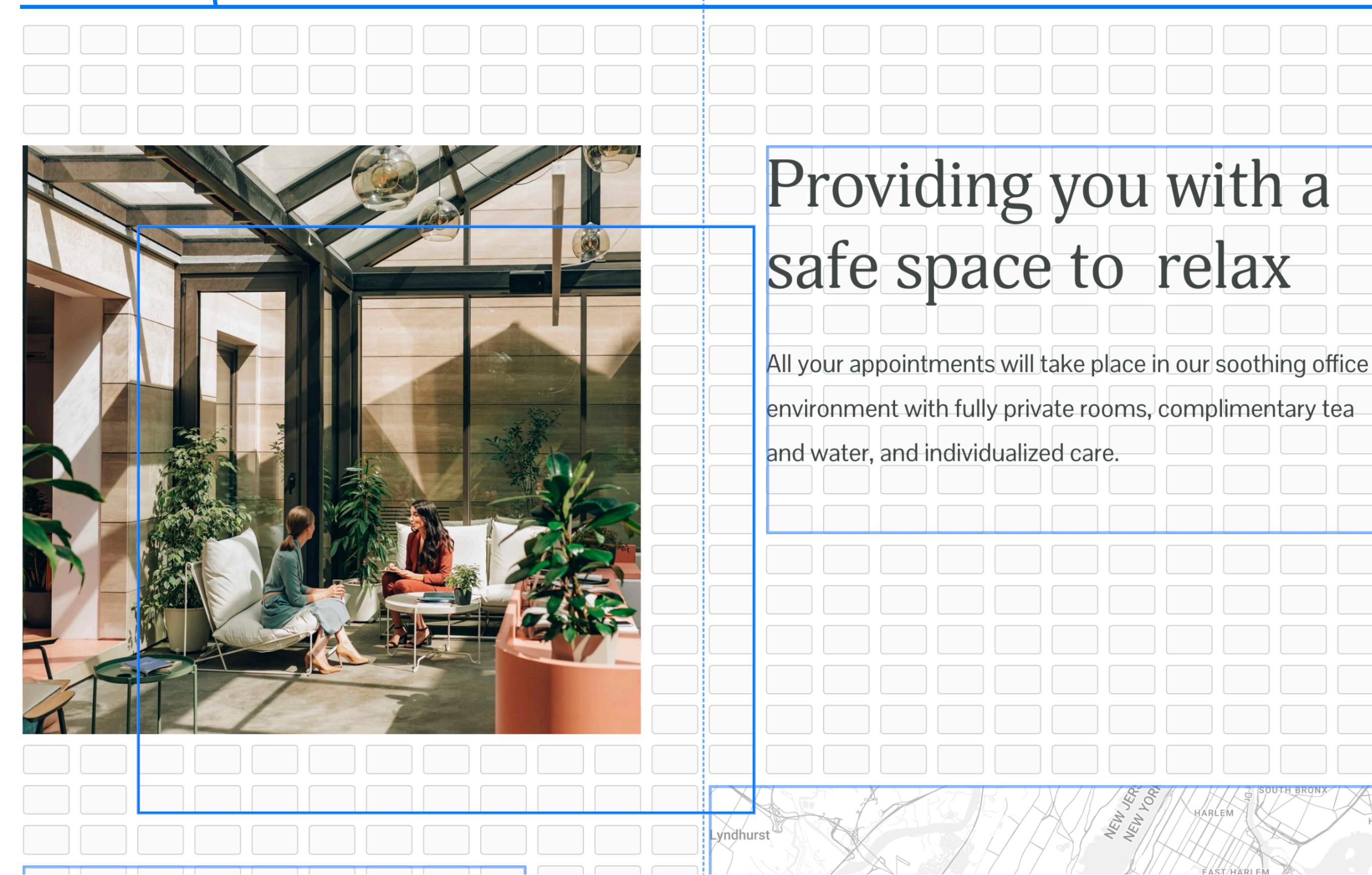


이런거 다 좌표값 계산해야 하는거 아닐까…?

display: grid;

grid-template-columns

grid-template-rows



column-gap, row-gap

The image illustrates a grid-based layout editor interface. At the top and bottom are 5x5 grids of white boxes. In the center-left, there is a photograph of two women in a modern office lounge, framed by a thick blue border. To the right of the photo is a blue-bordered text box containing the text "Providing you with a safe space to relax" and a description of the office environment. Below the main content area is a map of New York City, specifically showing the Bronx, Manhattan, and parts of New Jersey.

Providing you with a safe space to relax

All your appointments will take place in our soothing office environment with fully private rooms, complimentary tea and water, and individualized care.

Lyndhurst NEW JERSEY NEW YORK HARLEM EAST/HARLEM SOUTH BRONX HU

The image shows a grid-based editor interface. At the top and bottom of the interface are two rows of 12 empty grid cells each. In the center, there is a photograph of two women in a modern office setting, enclosed within a blue rectangular frame. To the right of the photograph is a text box with a blue border, containing the text "Providing you with a safe space to relax". Below the text box is another paragraph of text: "All your appointments will take place in our soothing office environment with fully private rooms, complimentary tea and water, and individualized care." At the very bottom of the interface is a map of New York City, specifically the Bronx and Manhattan areas, with labels for Lyndhurst, NEW JERSEY, NEW YORK, HARLEM, and SOUTH BRONX.

Providing you with a
safe space to relax

All your appointments will take place in our soothing office
environment with fully private rooms, complimentary tea
and water, and individualized care.

Lyndhurst NEW JERSEY NEW YORK HARLEM SOUTH BRONX

grid-area

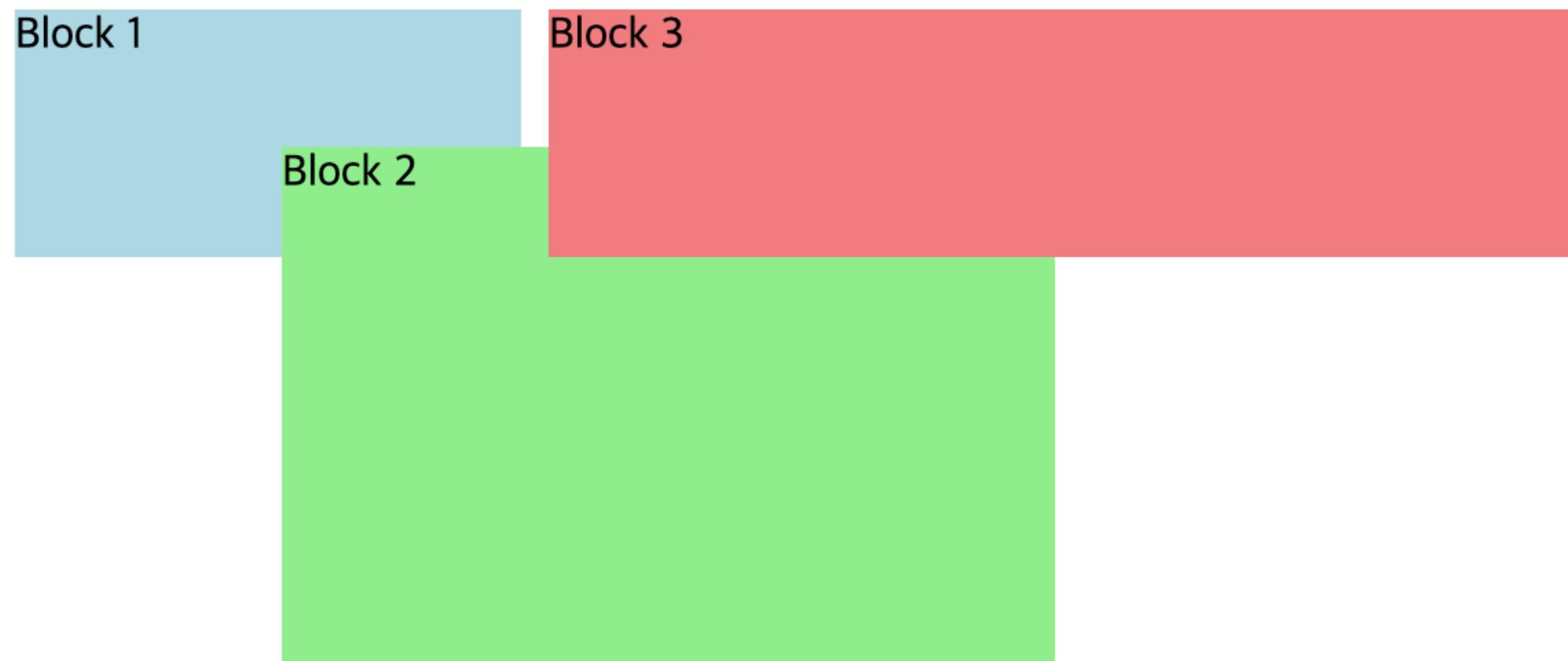
일단 위치 잡는건 사실상 끝!!

모두 CSS 속성만으로 가능해요!

복잡한 계산은 필요없어요

간단한 PoC

```
.container {  
    display: grid;  
    grid-template-columns: repeat(16, 1fr);  
    grid-template-rows: repeat(8, 40px);  
    gap: 10px;  
}  
.block1 {  
    grid-area: 1 / 1 / span 2 / span 2;  
    background-color: lightblue;  
}  
.block2 {  
    grid-area: 2 / 2 / span 4 / span 3;  
    background-color: lightgreen;  
}  
.block3 {  
    grid-area: 1 / 3 / span 2 / span 4;  
    background-color: lightcoral;  
}
```



아직까지는 단순한 HTML과 CSS

데이터를 그리기 위한 렌더러 로직이 필요해요

AST란?

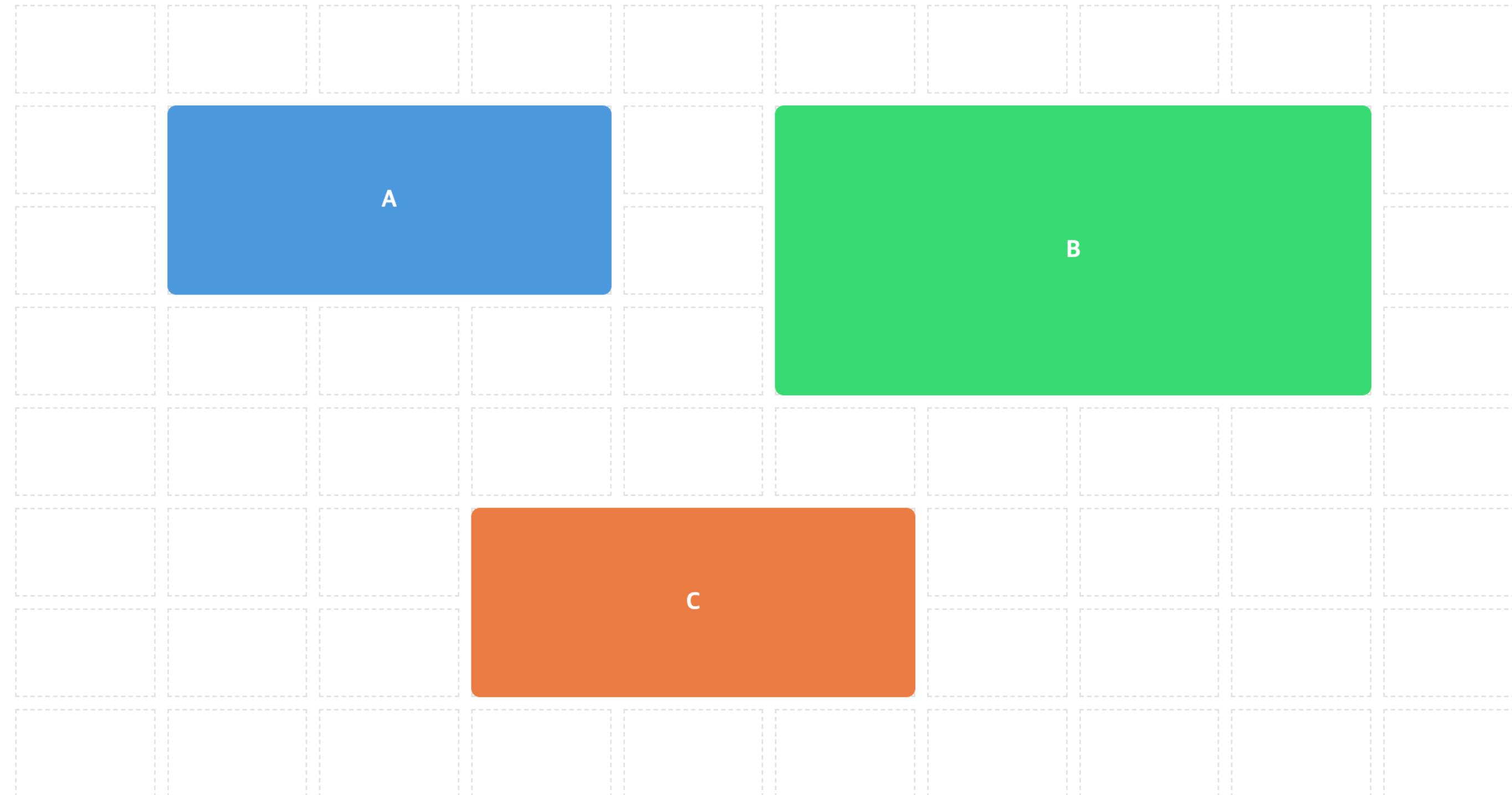
- AST는 추상 구문 트리의 준말로 어떠한 문법 구조를 표현하는 자료구조를 말해요
- 에디터를 표현하기 위한 모든 메타 데이터, 상태를 모두 AST로 표현할 수 있어야 해요
- 보통 편의를 위해 JSON을 이용해요

```
{  
  time: 1754086366542,  
  blocks: [  
    {  
      id: "mhTl6ghSkV",  
      type: "paragraph",  
      data: {  
        text: "Hey. Meet the new Editor. On this picture  
      },  
    },  
    {  
      id: "l98dyx3yjb",  
      type: "header",  
      data: {  
        text: "Key features",  
        level: 3,  
      },  
    },  
    {  
      id: "os_YI4eub4",  
      type: "list",  
      data: {  
        items: [  
          {  
            id: "l98dyx3yjb",  
            type: "header",  
            data: {  
              text: "Key features",  
              level: 3,  
            },  
          },  
          {  
            id: "os_YI4eub4",  
            type: "list",  
            data: {  
              items: [  
                {  
                  id: "l98dyx3yjb",  
                  type: "header",  
                  data: {  
                    text: "Key features",  
                    level: 3,  
                  },  
                },  
                {  
                  id: "os_YI4eub4",  
                  type: "list",  
                  data: {  
                    items: [  
                      {  
                        id: "l98dyx3yjb",  
                        type: "header",  
                        data: {  
                          text: "Key features",  
                          level: 3,  
                        },  
                      },  
                      {  
                        id: "os_YI4eub4",  
                        type: "list",  
                        data: {  
                          items: [  
                            {  
                              id: "l98dyx3yjb",  
                              type: "header",  
                              data: {  
                                text: "Key features",  
                                level: 3,  
                              },  
                            },  
                            {  
                              id: "os_YI4eub4",  
                              type: "list",  
                              data: {  
                                items: [  
                                  {  
                                    id: "l98dyx3yjb",  
                                    type: "header",  
                                    data: {  
                                      text: "Key features",  
                                      level: 3,  
                                    },  
                                  },  
                                  {  
                                    id: "os_YI4eub4",  
                                    type: "list",  
                                    data: {  
                                      items: [  
                                        {  
                                          id: "l98dyx3yjb",  
                                          type: "header",  
                                          data: {  
                                            text: "Key features",  
                                            level: 3,  
                                          },  
                                        },  
                                        {  
                                          id: "os_YI4eub4",  
                                          type: "list",  
                                          data: {  
                                            items: [  
                                              {  
                                                id: "l98dyx3yjb",  
                                                type: "header",  
                                                data: {  
                                                  text: "Key features",  
                                                  level: 3,  
                                                },  
                                              },  
                                              {  
                                                id: "os_YI4eub4",  
                                                type: "list",  
                                                data: {  
                                                  items: [  
                                                    {  
                                                      id: "l98dyx3yjb",  
                                                      type: "header",  
                                                      data: {  
                                                        text: "Key features",  
                                                        level: 3,  
                                                      },  
                                                    },  
                                                    {  
                                                      id: "os_YI4eub4",  
                                                      type: "list",  
                                                      data: {  
                                                        items: [  
                                                          {  
                                                            id: "l98dyx3yjb",  
                                                            type: "header",  
                                                            data: {  
                                                              text: "Key features",  
                                                              level: 3,  
                                                            },  
                                                          },  
                                                          {  
                                                            id: "os_YI4eub4",  
                                                            type: "list",  
                                                            data: {  
                                                              items: [  
                                                                {  
                                                                  id: "l98dyx3yjb",  
                                                                  type: "header",  
                                                                  data: {  
                                                                    text: "Key features",  
                                                                    level: 3,  
                                                                  },  
                                                                },  
                                                                {  
                                                                  id: "os_YI4eub4",  
                                                                  type: "list",  
                                                                  data: {  
                                                                    items: [  
                                                                      {  
                                                                        id: "l98dyx3yjb",  
                                                                        type: "header",  
                                                                        data: {  
                                                                          text: "Key features",  
                                                                          level: 3,  
                                                                        },  
                                                                      },  
                                                                      {  
                                                                        id: "os_YI4eub4",  
                                                                        type: "list",  
                                                                        data: {  
                                                                          items: [  
                                                                            {  
                                                                              id: "l98dyx3yjb",  
                                                                              type: "header",  
                                                                              data: {  
                                                                                text: "Key features",  
                                                                                level: 3,  
                                                                              },  
                                                                            },  
                                                                            {  
                                                                              id: "os_YI4eub4",  
                                                                              type: "list",  
                                                                              data: {  
                                                                                items: [  
                                                                                  {  
                                                                                    id: "l98dyx3yjb",  
                                                                                    type: "header",  
                                                                                    data: {  
                                                                                      text: "Key features",  
                                                                                      level: 3,  
                                                                                    },  
                                                                                  },  
                                                                                  {  
                                                                                    id: "os_YI4eub4",  
                                                                                    type: "list",  
                                                                                    data: {  
                                                                                      items: [  
                                                                                      {  
                        id: "l98dyx3yjb",  
                        type: "header",  
                        data: {  
                          text: "Key features",  
                          level: 3,  
                        },  
                      },  
                      {  
                        id: "os_YI4eub4",  
                        type: "list",  
                        data: {  
                          items: [  
                            {  
                              id: "l98dyx3yjb",  
                              type: "header",  
                              data: {  
                                text: "Key features",  
                                level: 3,  
                              },  
                            },  

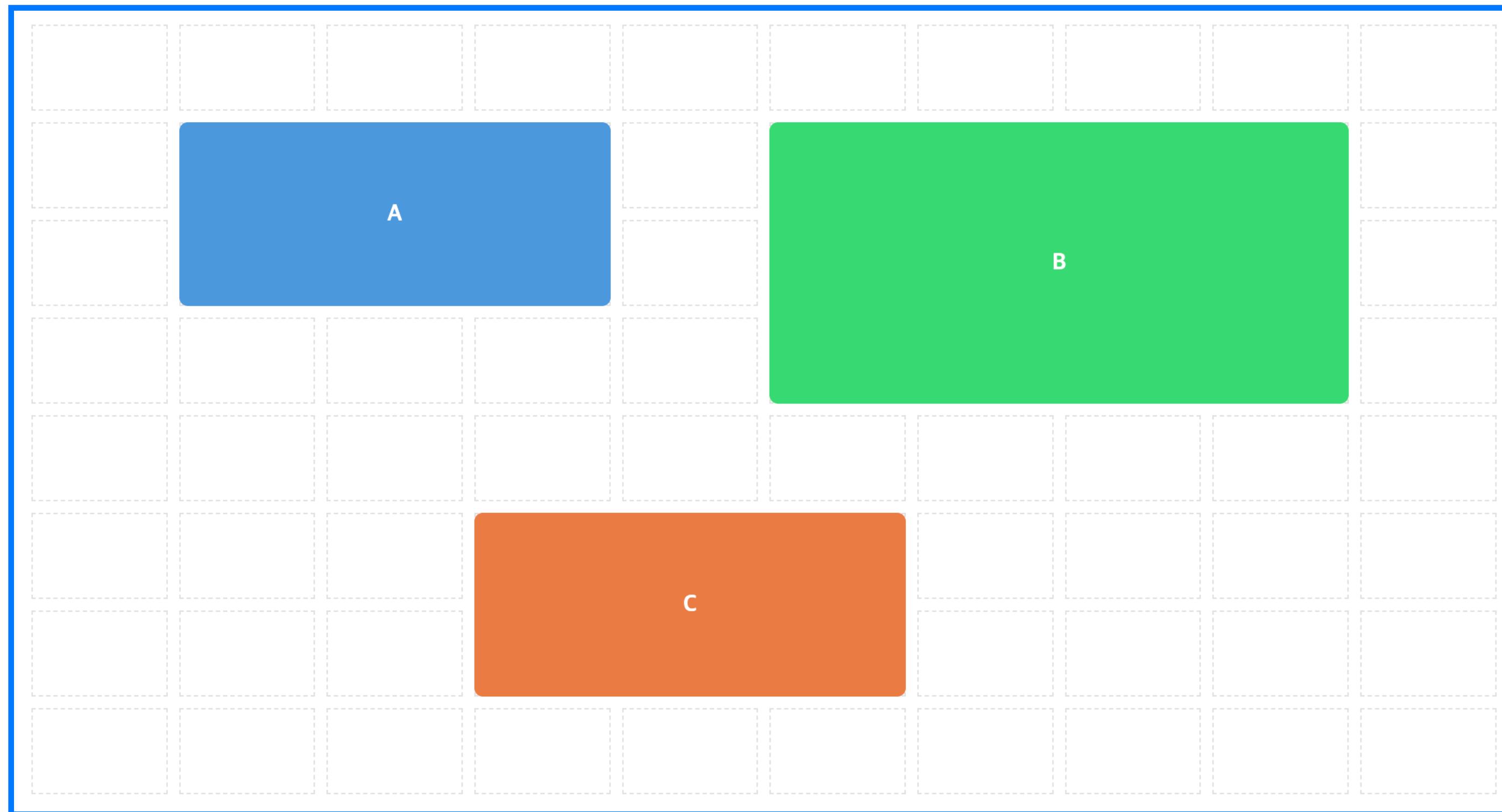
```

중요한건 화면에 보이는 것을 데이터로 추상화하는 것!

무엇을 데이터화 할 수 있을까요?

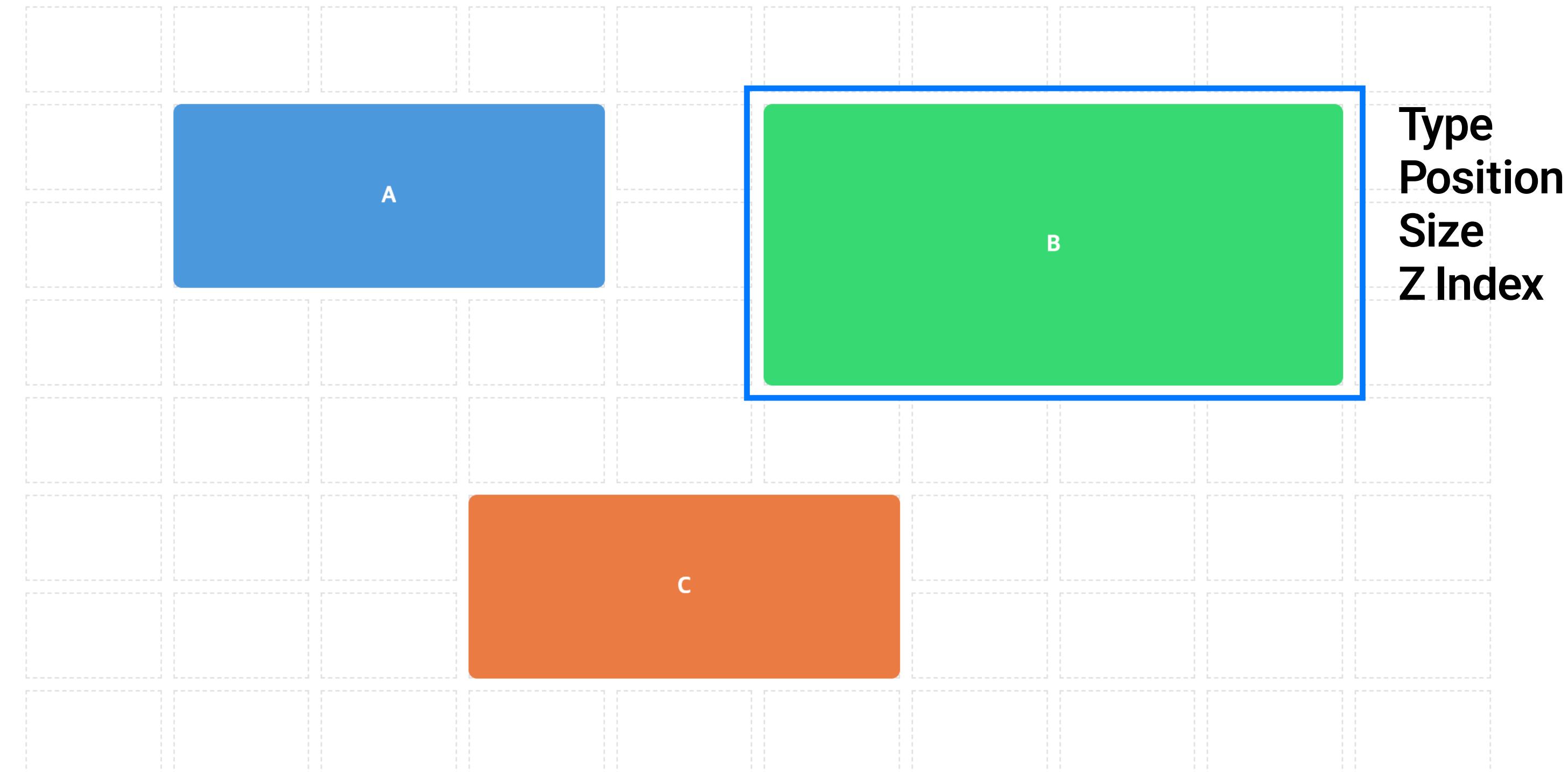


무엇을 데이터화 할 수 있을까요?

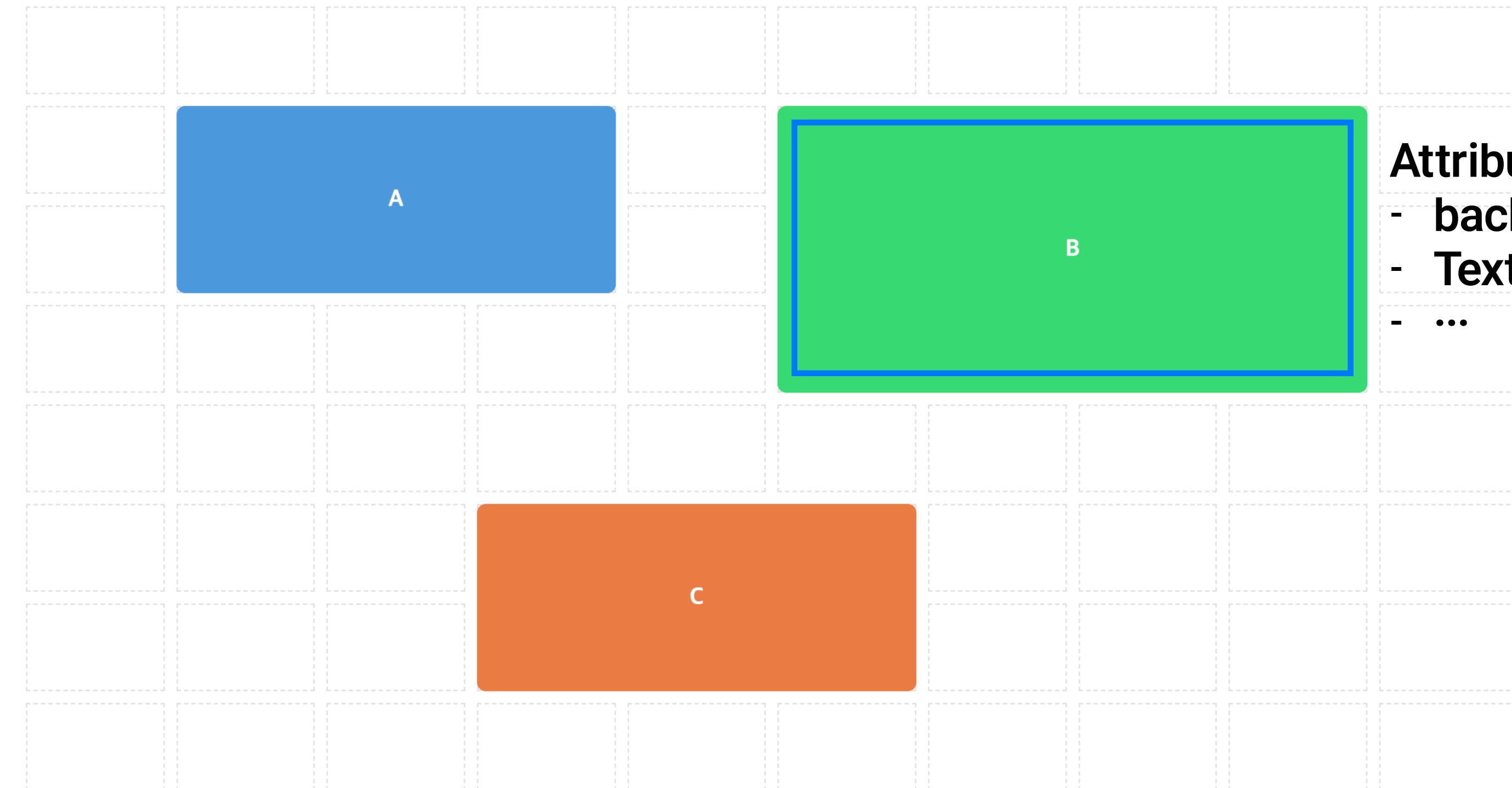


Columns
Rows
Row Height
Gap

무엇을 데이터화 할 수 있을까요?



무엇을 데이터화 할 수 있을까요?



Attributes

- backgroundColor
- Text
- ...

```
{  
  "grid": {  
    "columns": 12,  
    "rows": 10,  
    "rowHeight": 60,  
    "gap": 8  
  },  
  "blocks": [  
    {  
      "id": "4rh2a6v7j",  
      "type": "box",  
      "position": {  
        "x": 1,  
        "y": 2,  
        "zIndex": 1  
      },  
      "size": {  
        "width": 3,  
        "height": 2  
      },  
      "attributes": {  
        "color": "#ff7875",  
        "text": "Box"  
      }  
    }  
  ]  
}
```

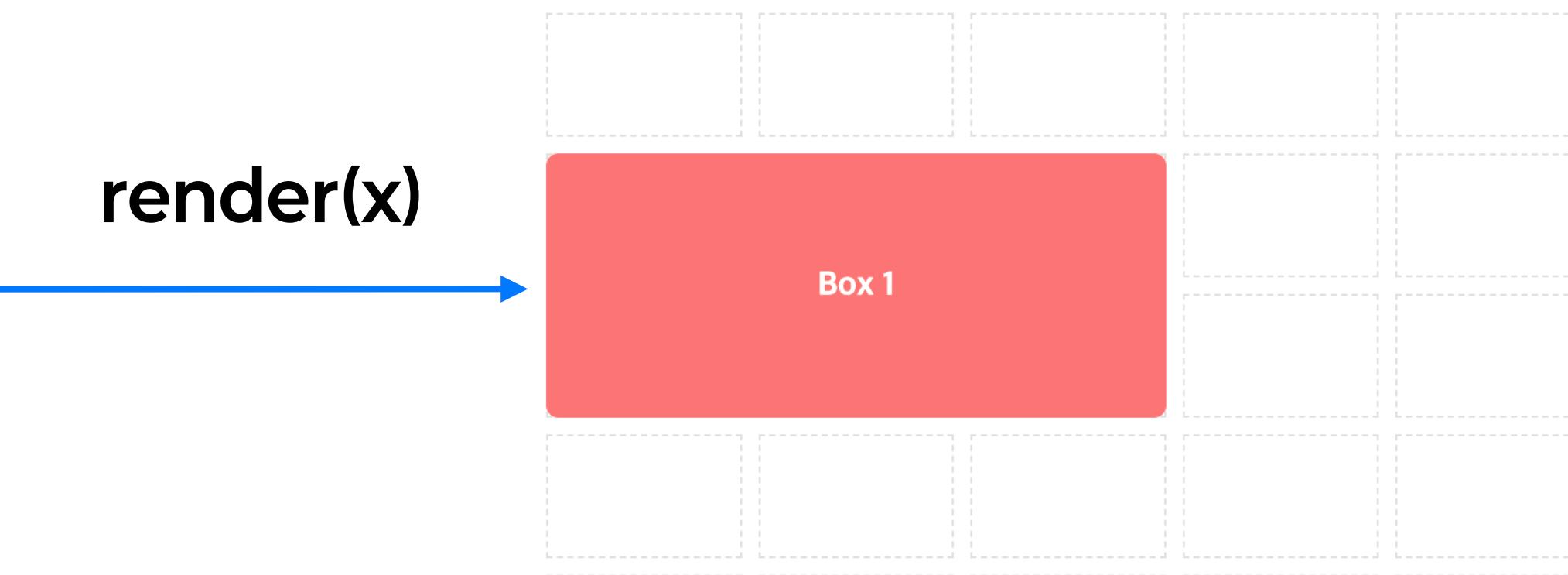
→ Grid 레이아웃 관련 데이터

→ Block 관련 데이터

→ Box Block 속성 데이터

```
{  
  "grid": {  
    "columns": 12,  
    "rows": 10,  
    "rowHeight": 60,  
    "gap": 8  
  },  
  "blocks": [  
    {  
      "id": "4rh2a6v7j",  
      "type": "box",  
      "position": {  
        "x": 1,  
        "y": 2,  
        "zIndex": 1  
      },  
      "size": {  
        "width": 3,  
        "height": 2  
      },  
      "attributes": {  
        "color": "#ff7875",  
        "text": "Box"  
      }  
    }  
  ]  
}
```

render(x)

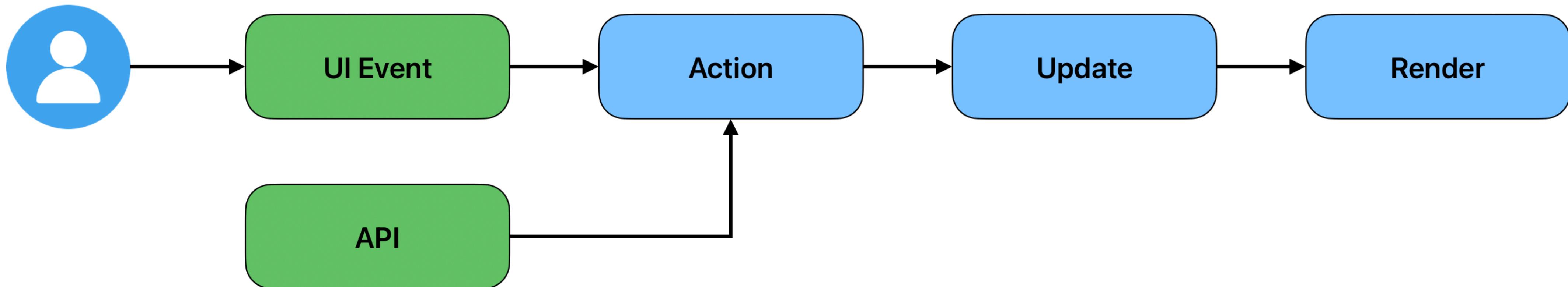


상호 작용

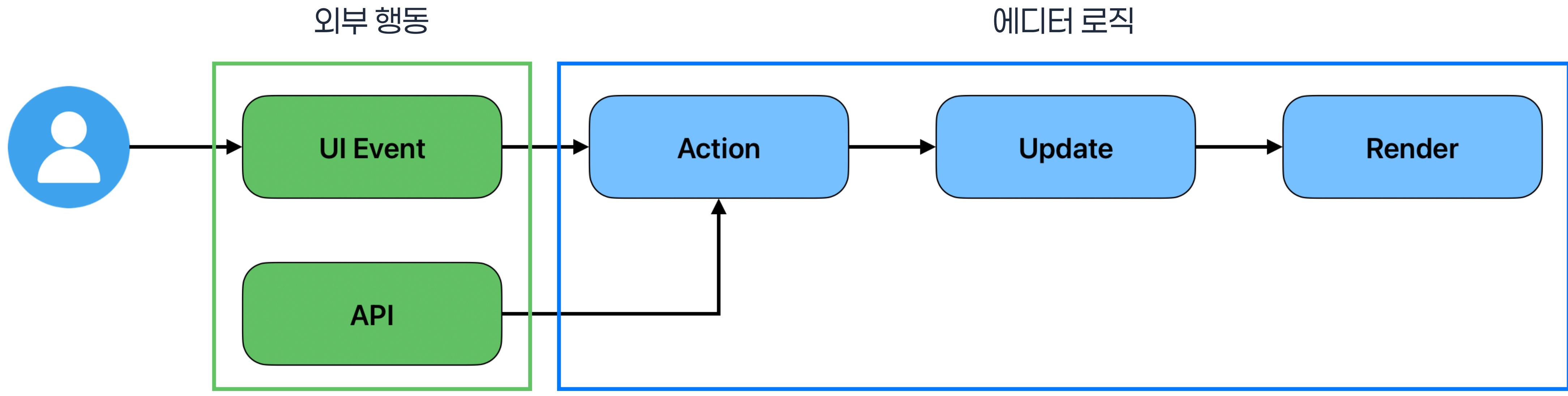
에디터가 되기 위해 필요한 것

- 에디터는 **사람과 시스템의 상호작용**이 매우 중요한 소프트웨어에요
- 편집한 것이 그대로 실제 화면이 되는 WYSIWYG이 되면 좋아요
 - UI 이벤트를 잘 처리하는 것이 중요해요
 - 자연스러운 UI/UX를 위한 상태가 필요해요
- 에디터 시스템을 어떻게 설계하면 좋을까요?

에디터 동작의 흐름

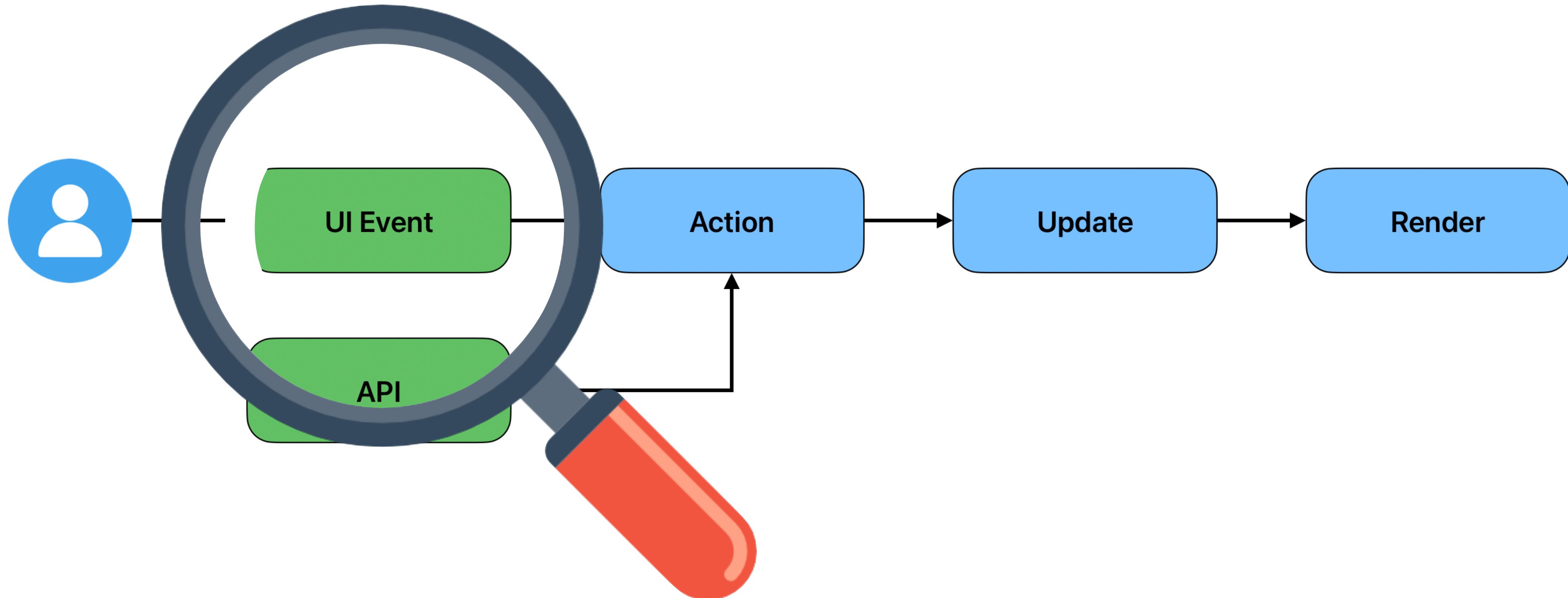


에디터 동작의 흐름



철저하게 구분해야 훗날 편함!

에디터 동작의 흐름





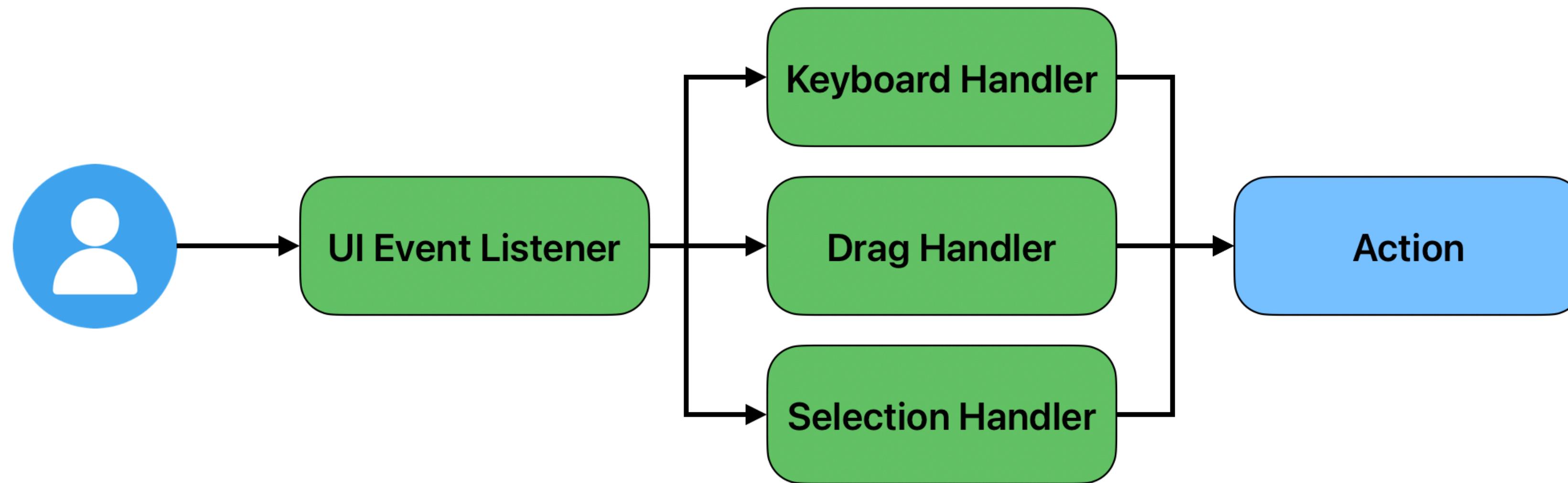
에디터는 매우 엄격한 룰 기반 소프트웨어

사용자가 할 수 있는 것과 아닌 것을 정리할 필요가 있어요

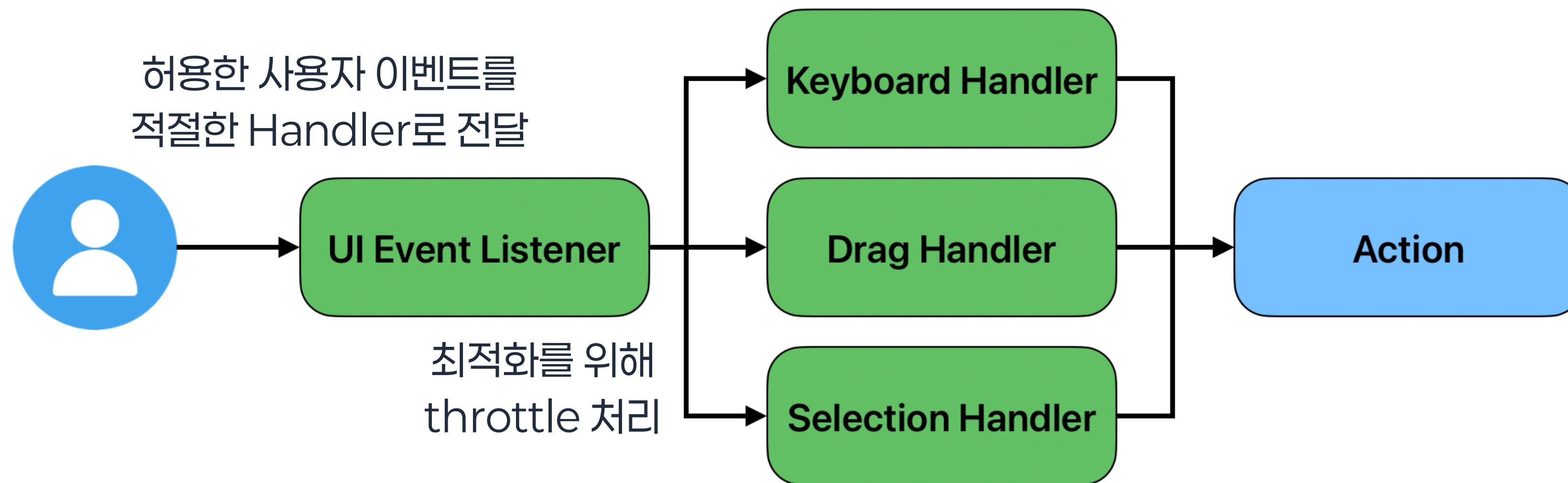
어떤 룰이 필요한가?

- 요구사항에 따라 다르지만 ‘그리드 기반 에디터’의 보편적인 룰이 있어요
- 빠르게 훑고 지나갈게요
 - 에디터가 편집 가능한 상태인가? (Editable)
 - 블럭을 이동할 수 있다
 - 블럭의 크기를 조절할 수 있다
 - 블럭은 그리드 영역 밖으로 나갈 수 없다
 - 블럭과 블럭이 겹칠 수 있는가?
 - 겹칠 수 없다면 이동, 리사이즈할 때 겹치는지 확인
 - 블럭을 선택할 수 있다
 - 블럭의 크기 제한이 있을 수 있다
 - 여러 개의 블럭을 선택할 수 있다

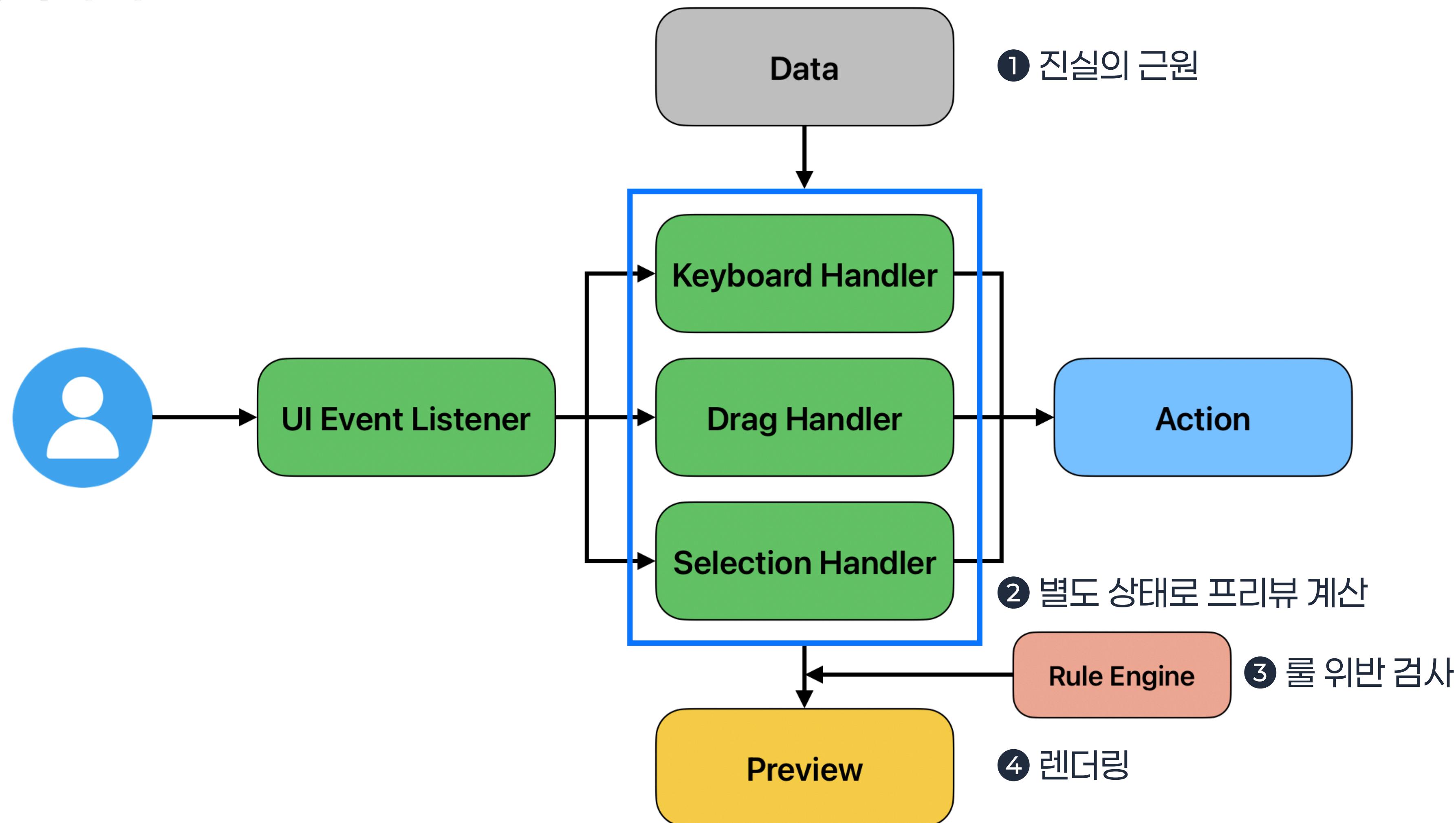
UI 이벤트 처리하기



UI 이벤트 처리하기

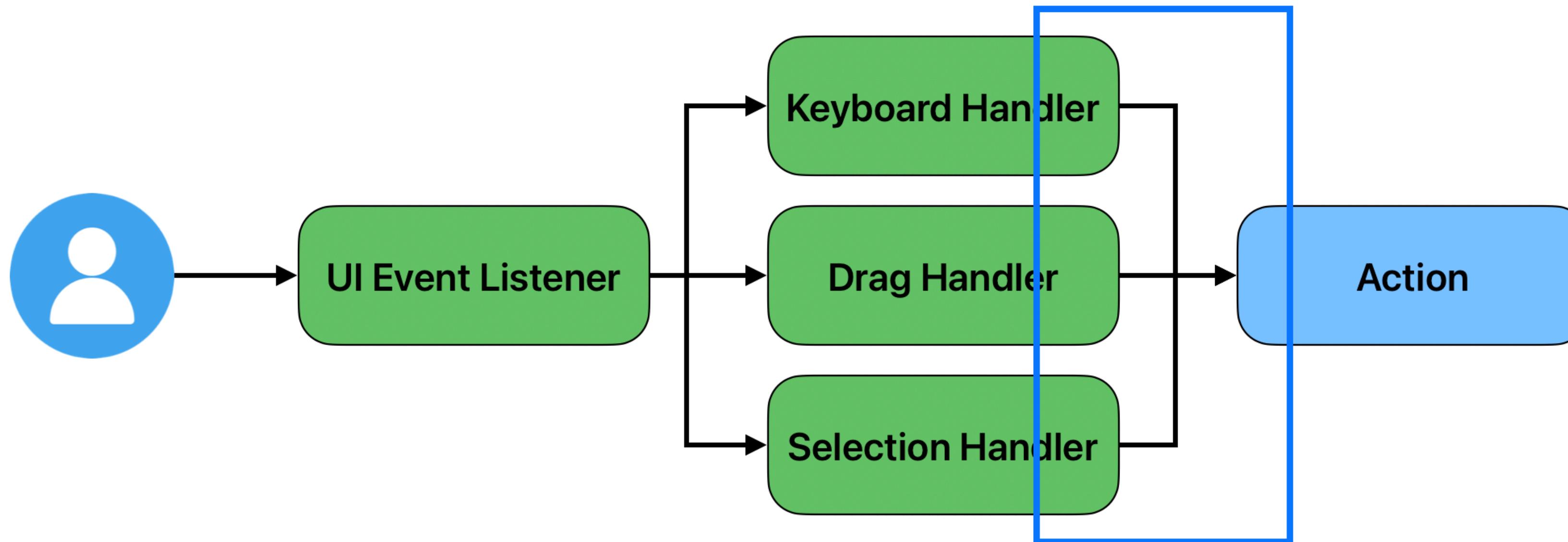


UI 이벤트 처리하기



UI 이벤트 처리하기

사용자 행동(이벤트) 종료
상황에 따라 FLIP 애니메이션 처리



FLIP 방식 애니메이션

- **First, Last, Invert, Play**
 - First: 현재 대상의 위치와 크기 기록
 - Last: 레이아웃이 변경된 후 대상의 위치와 크기 기록
 - Invert: 최초 값과 최종 값의 변화량을 계산
 - Play: 애니메이션 재생
- CSS를 이용해서 빨라요
- 구현이 편해요

그리드 기반 에디터 / 빌더 구현기

```
function flipAnimate(element, mutateLayout) {
  // 1) First
  const first = element.getBoundingClientRect();

  // 2) 레이아웃 변경 (최종 상태를 먼저 만든다)
  mutateLayout();

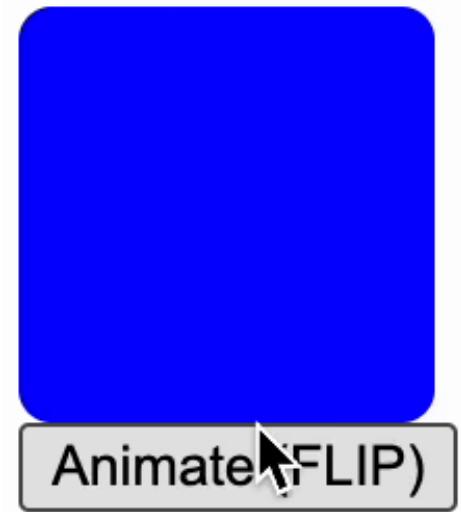
  // 3) Last
  const last = element.getBoundingClientRect();

  // 4) Invert (First 기준으로 보정)
  const dx = first.left - last.left;
  const dy = first.top - last.top;
  const sx = first.width / last.width || 1;
  const sy = first.height / last.height || 1;

  // 현재 프레임에 Invert 상태 적용
  element.style.transition = 'none';
  element.style.transform = `translate(${dx}px, ${dy}px) scale(${sx}, ${sy})`;

  // 6) 다음 프레임에서 Play (원래 위치/크기로)
  requestAnimationFrame(() => {
    element.style.transition = 'transform 300ms ease';
    element.style.transform = 'none';

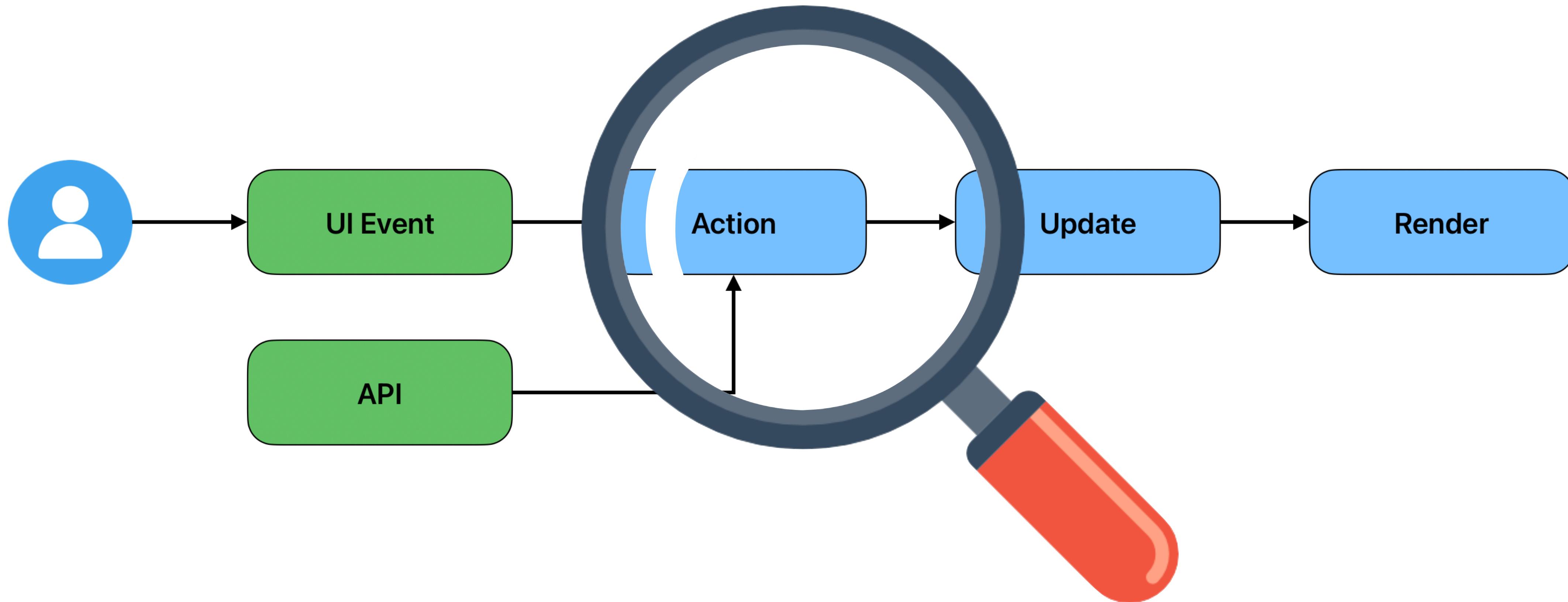
    // 7) 정리: 애니메이션 끝나면 transition 제거
    const onEnd = (e) => {
      if (e.propertyName !== 'transform') return;
      element.style.transition = '';
      element.removeEventListener('transitionend', onEnd);
    };
    element.addEventListener('transitionend', onEnd, { once: true });
  });
}
```



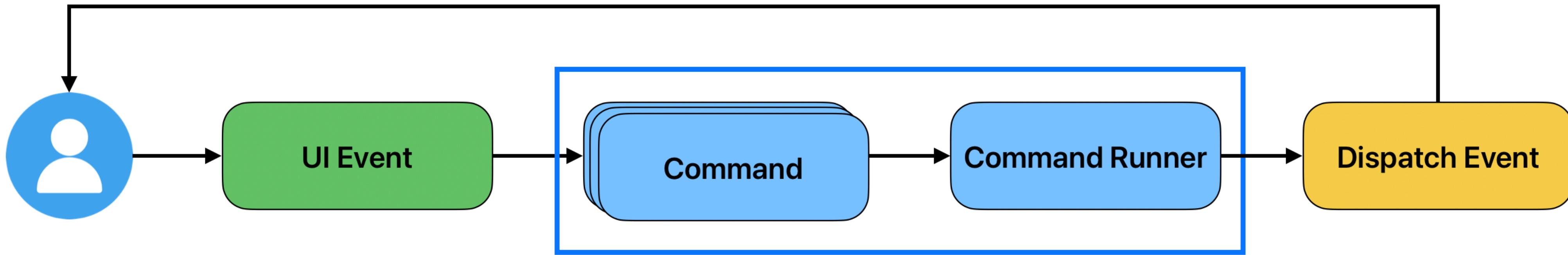
명령 처리하기

에디터에게 명령 내리기

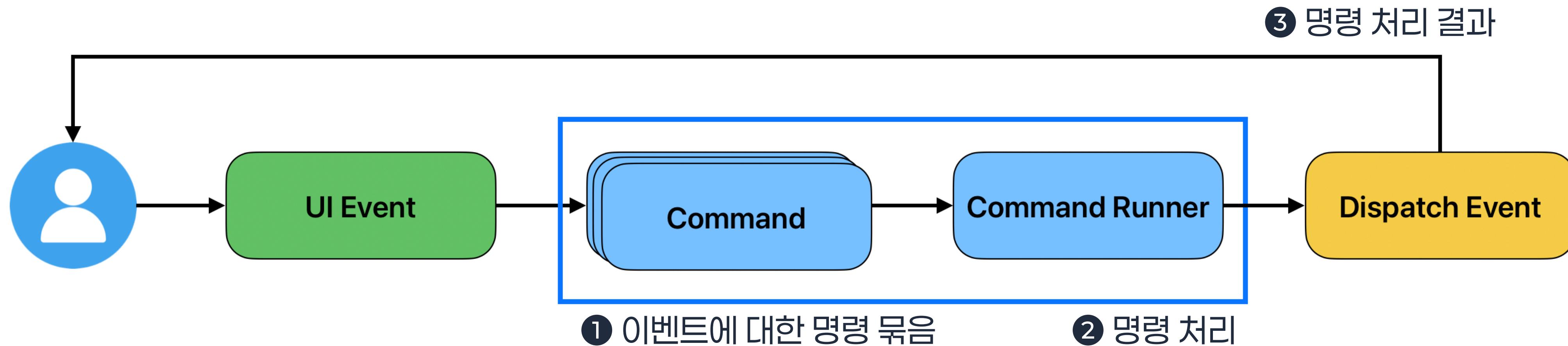
- 상호 작용 후 혹은 API 접근을 통한 명령/실행 로직을 통해 **최종적인 데이터**를 만들어야해요
- UI 쪽보다 구현이 쉽지만 신경쓸 것이 몇 가지 있어요
 - 트랜잭션
 - API 인터페이스
 - 이벤트 전파



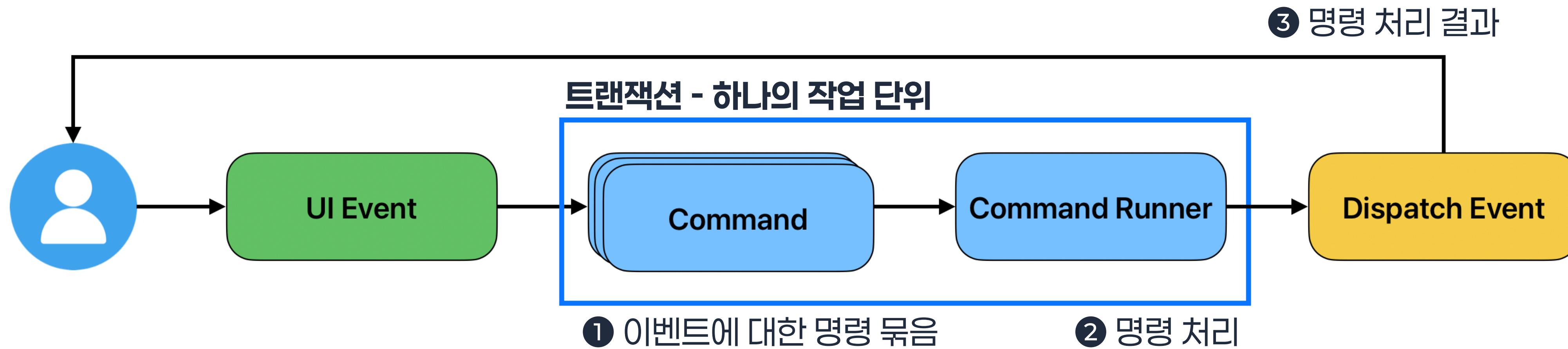
명령 처리하기



명령 처리하기



명령 처리하기



트랜잭션을 두는 이유

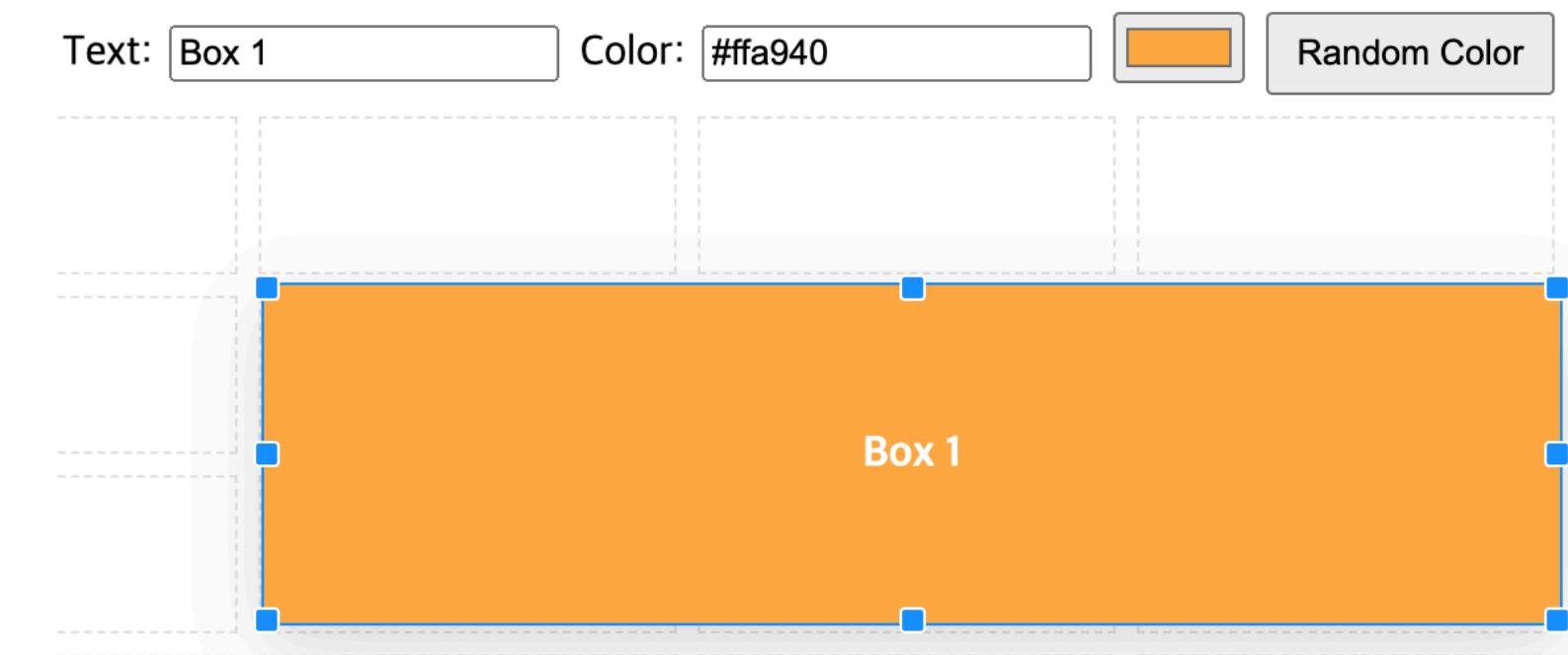
- 트랜잭션은 Commit/Rollback으로 작업의 성공, 실패를 처리할 수 있어요
- 각각의 명령을 **하나의 작업**으로 묶기 좋아요
- 여러 명령을 처리하던 중 **룰 위반이 발견되면 롤백**하기 좋아요
- **트랜잭션 단위로 Undo/Redo**를 처리하면 편해요
- 추후 **커스텀 로직으로 확장**하기 좋아요

```
pegboard.transaction((rollback) => {
  if (!selected) {
    rollback();
    return;
  }
  pegboard.moveBlockToPosition(selected, { x: 0, y: 3, zIndex: 1 });
  pegboard.resizeBlock(selected, { width: 6, height: 4 });
  pegboard.updateBlock(selected, { attributes: { text: 'Moved & Resized' } });
});
```

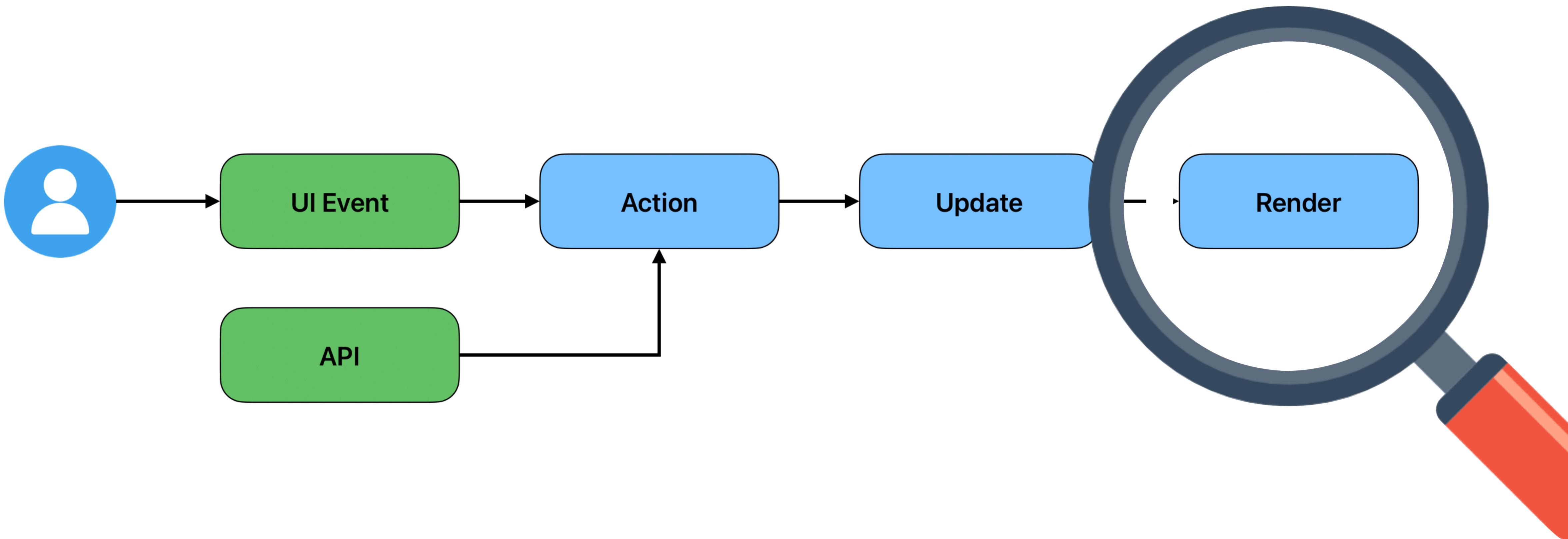
명령에 대한 결과 알려주기

- 확장성을 위해 처리한 결과에 대해서 알려주는 것이 좋아요
 - 에디터 외부에 ToolBar를 두거나 동시 편집에 대한 UI를 구현하는 등 다양한 곳에 쓰여요
- 최대한 다양한 이벤트를 지원할 수록 확장성에 좋아요

```
pegboard.on('block:selected', (data) => {
  if (!data.block) return;
  const attributes = data.block.attributes;
  inspector.setAttribute('text', attributes.text);
  inspector.setAttribute('color', attributes.color);
});
```



UI/UX에 신경쓰기

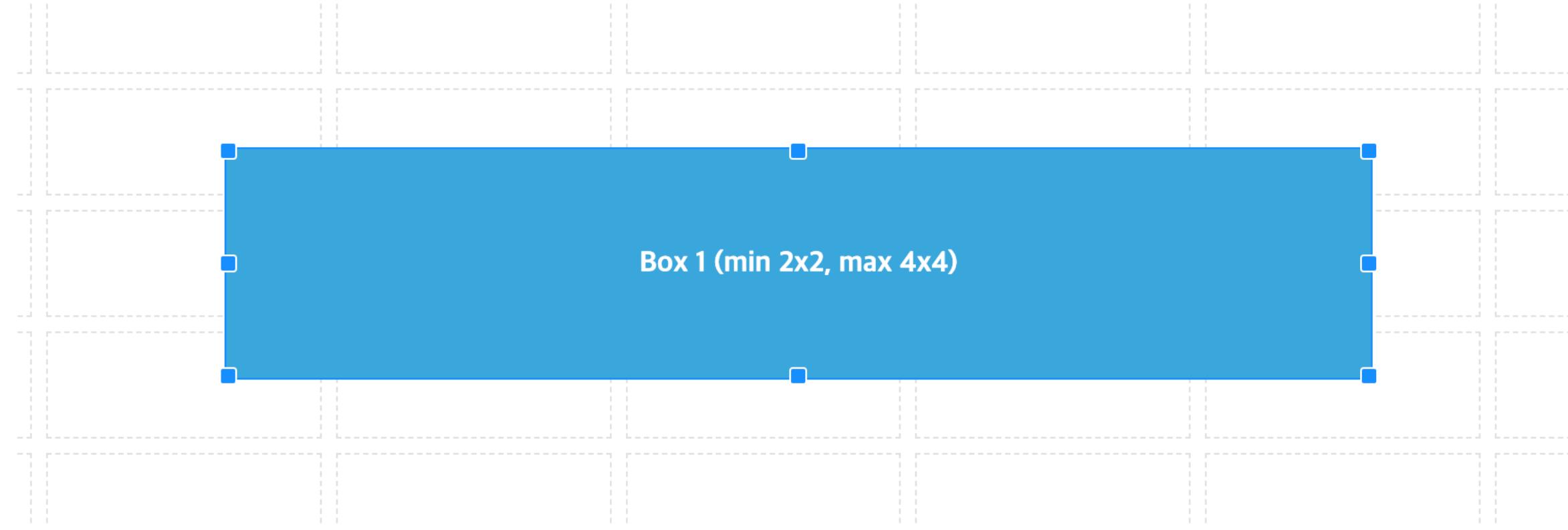


많이 듣는 에디터 사용 피드백

- 어떻게 사용하는건지 잘 모르겠어요
- 이거 왜 안되는거에요?
- 갑자기 사라졌어요

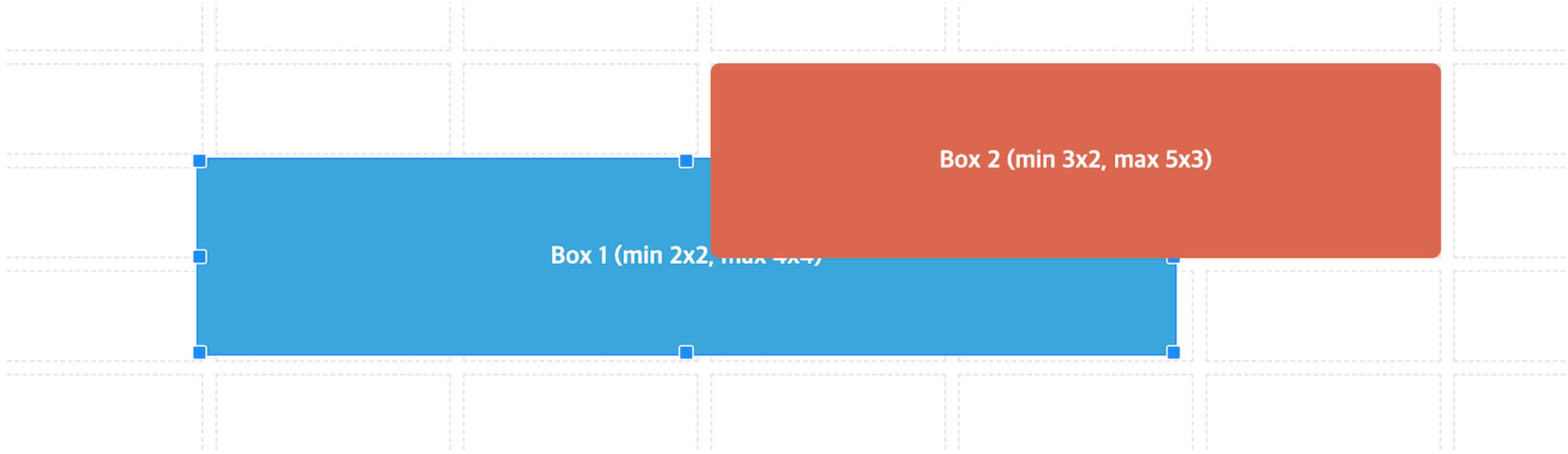
이런 피드백은 대부분 UI 표현이 부족했기 때문!

사례



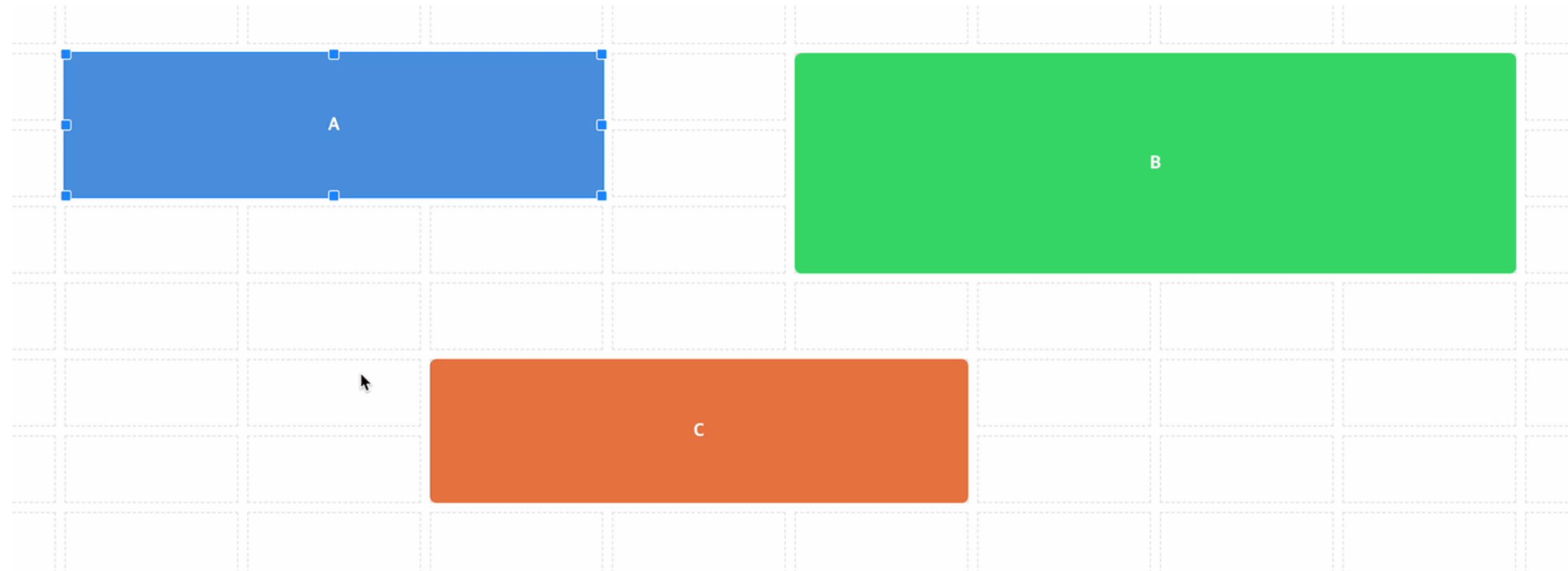
이동 중인지, 드롭하면 어디에 들어갈지 알 수 없음

사례



배치 가능한지 아닌지 알 수 없음

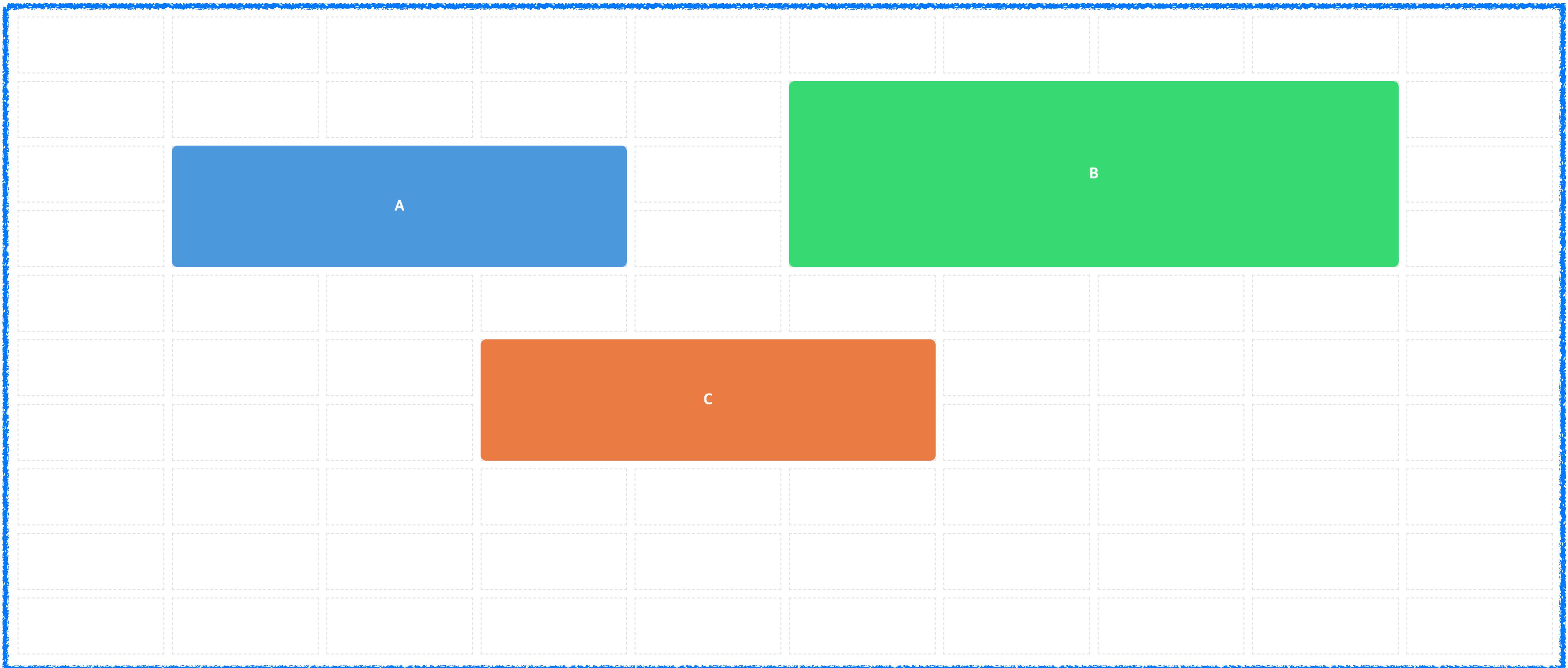
사례



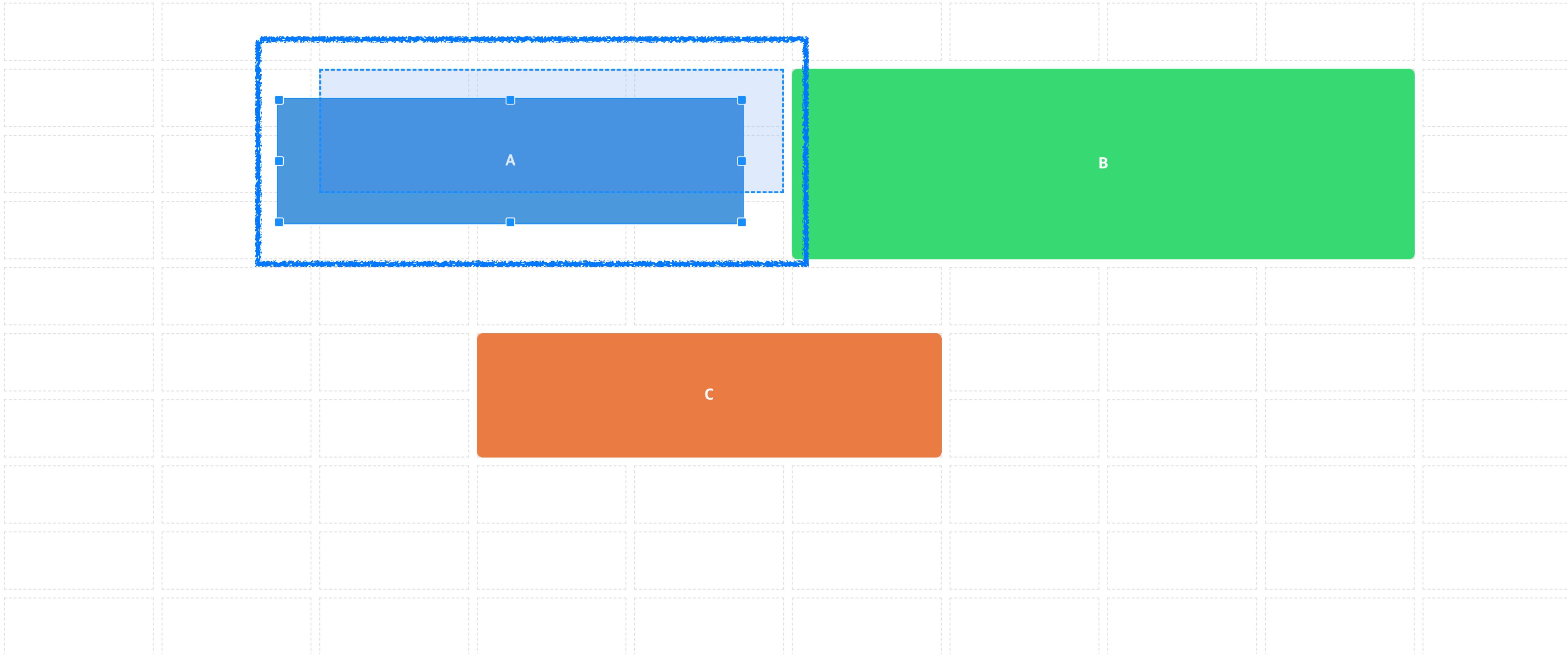
집중하지 않으면 놓칠 수 있음

규칙과 변화를 UI/UX에 녹여야해요

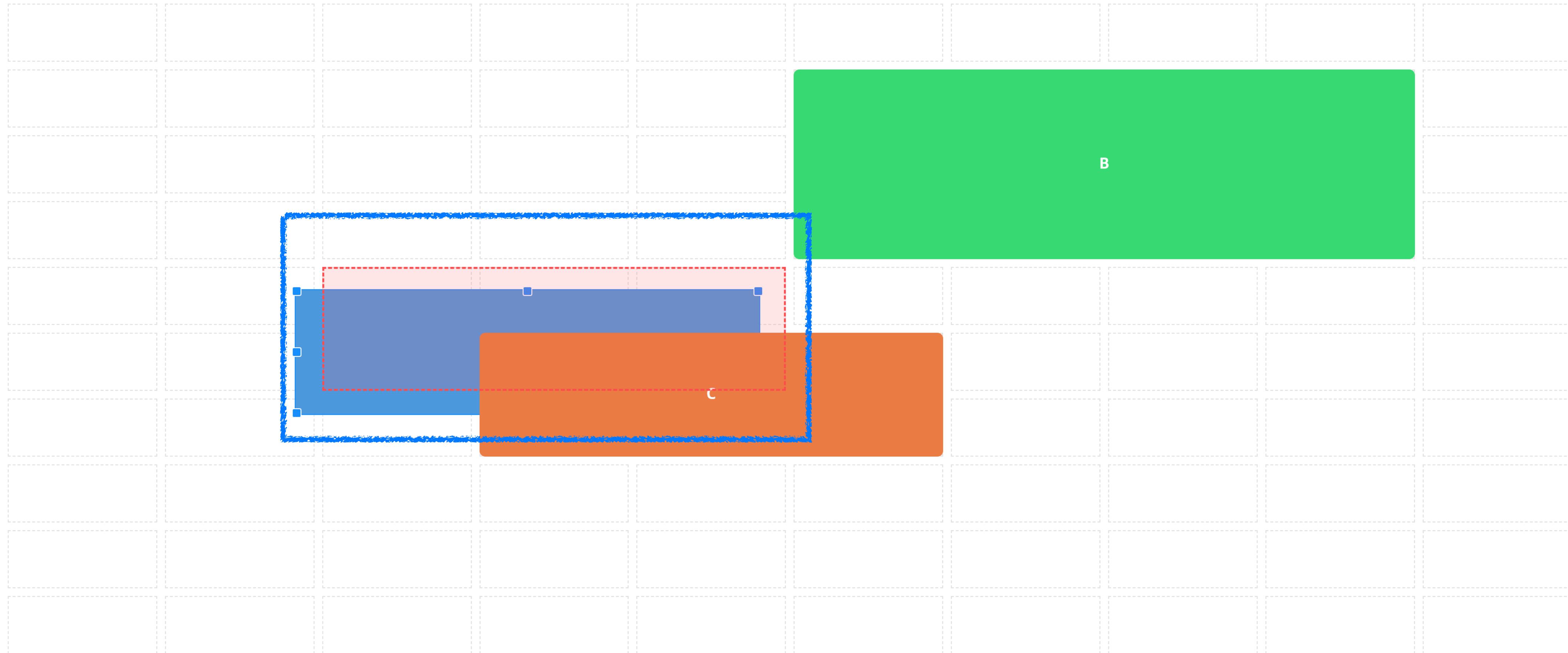
규칙을 암시하기



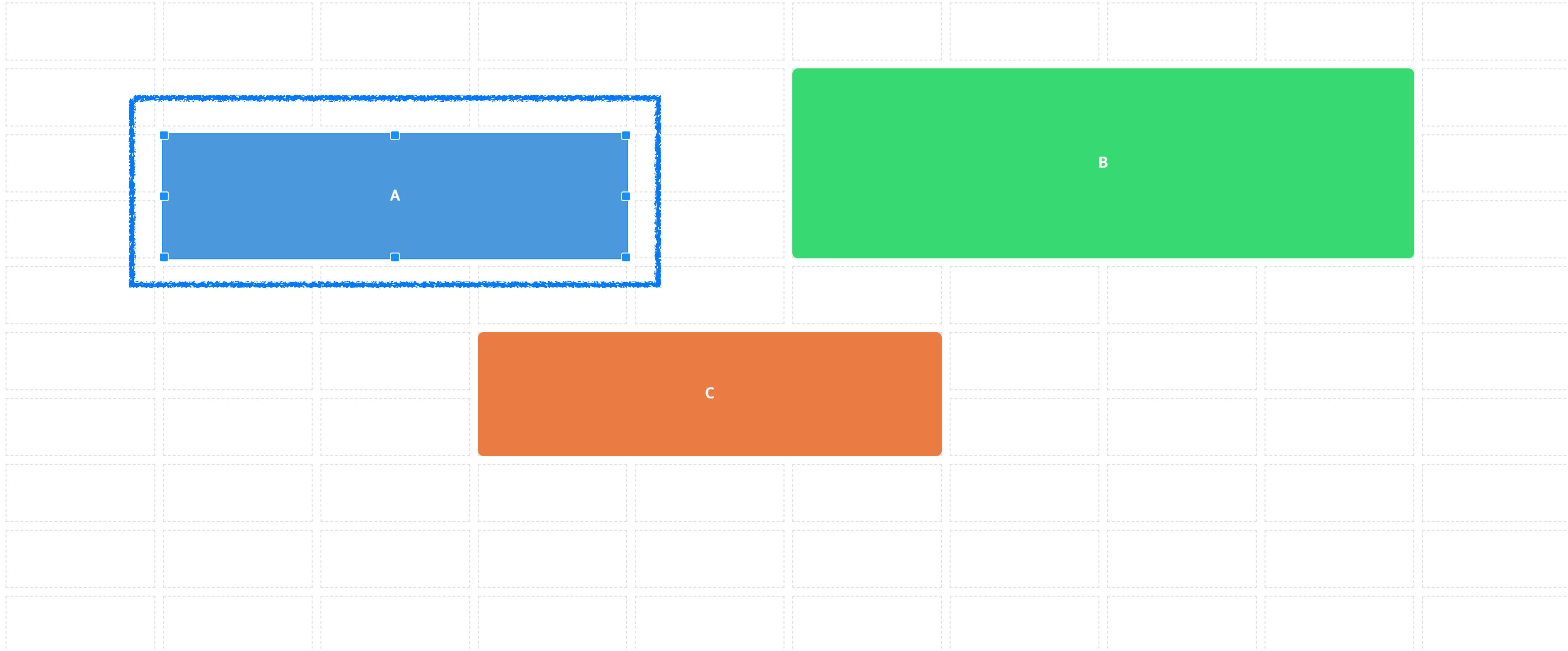
규칙을 암시하기



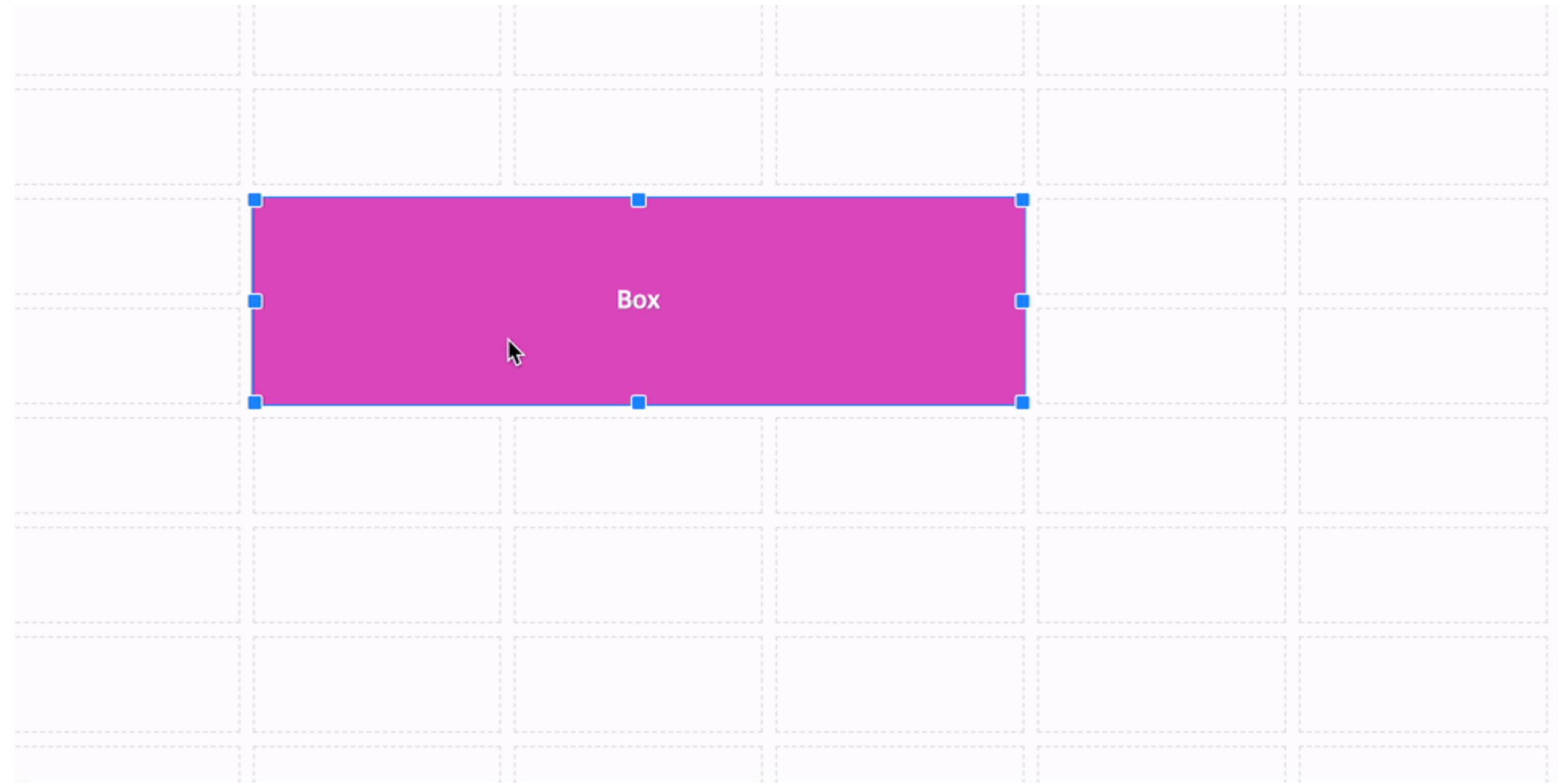
규칙을 암시하기



상태 변화를 알려주기



상태 변화를 알려주기



상태 변화를 알려주기

dragReflow = axis-shift | autoGrowRows=true



정리

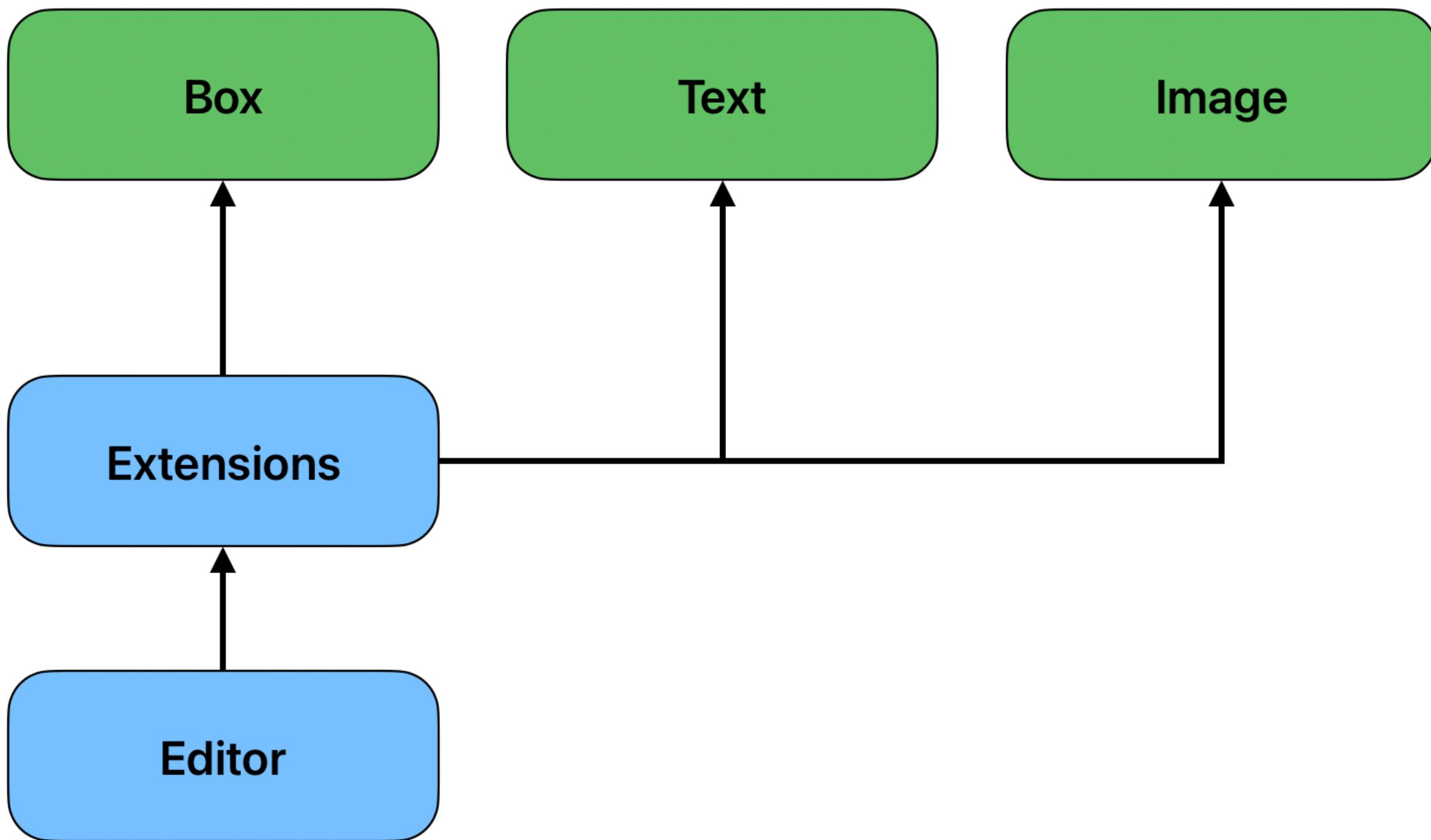
- 편집 중일 때는 **규칙을 암시하는 프리뷰**를 보여주면 좋아요
- **현재 상태**에 대해 정확히 보여주는 것이 좋아요
- 적절한 **트랜지션**은 **상태 변화**를 이해하기에 좋아요

좋은 사용자 경험을 위해선 UI에 신경써야 합니다

블럭 확장하기

에디터를 제품에 활용하기 위해서는 다양한 블럭을 표현할 수 있어야 해요

이런 경우 Extension을 등록하게 하여 확장할 수 있어요



```
import type { BlockData } from './types';

You, 15초 전 | 2 authors (kciter and one other)
export abstract class BlockExtension<Attrs extends Record<string, any>> {
    abstract readonly type: string;
    readonly defaultAttributes?: Attrs;
    readonly allowEditMode?: boolean;

    // render method to be implemented by subclasses
    abstract render(data: BlockData<Attrs>, container: HTMLElement, isEditorMode: boolean): void;

    // lifecycle hooks
    onCreate?(data: BlockData<Attrs>, container: HTMLElement, isEditorMode: boolean): void;
    onDestroy?(data: BlockData<Attrs>): void;
    onBeforeRender?(data: BlockData<Attrs>, container: HTMLElement, isEditorMode: boolean): void;
    onAfterRender?(data: BlockData<Attrs>, container: HTMLElement, isEditorMode: boolean): void;
    onUpdateAttributes?(data: BlockData<Attrs>, container: HTMLElement, isEditorMode: boolean): void;
    onEnterEditMode?(data: BlockData<Attrs>, container: HTMLElement): void;
    onExitEditMode?(data: BlockData<Attrs>, container: HTMLElement): void;
}
```

```
export interface BoxBlockAttributes {
  text: string;
  color: string;
}

export class BoxBlock extends BlockExtension<BoxBlockAttributes> {
  readonly type = 'box';
  readonly defaultAttributes = { color: '#888', text: 'Box' };

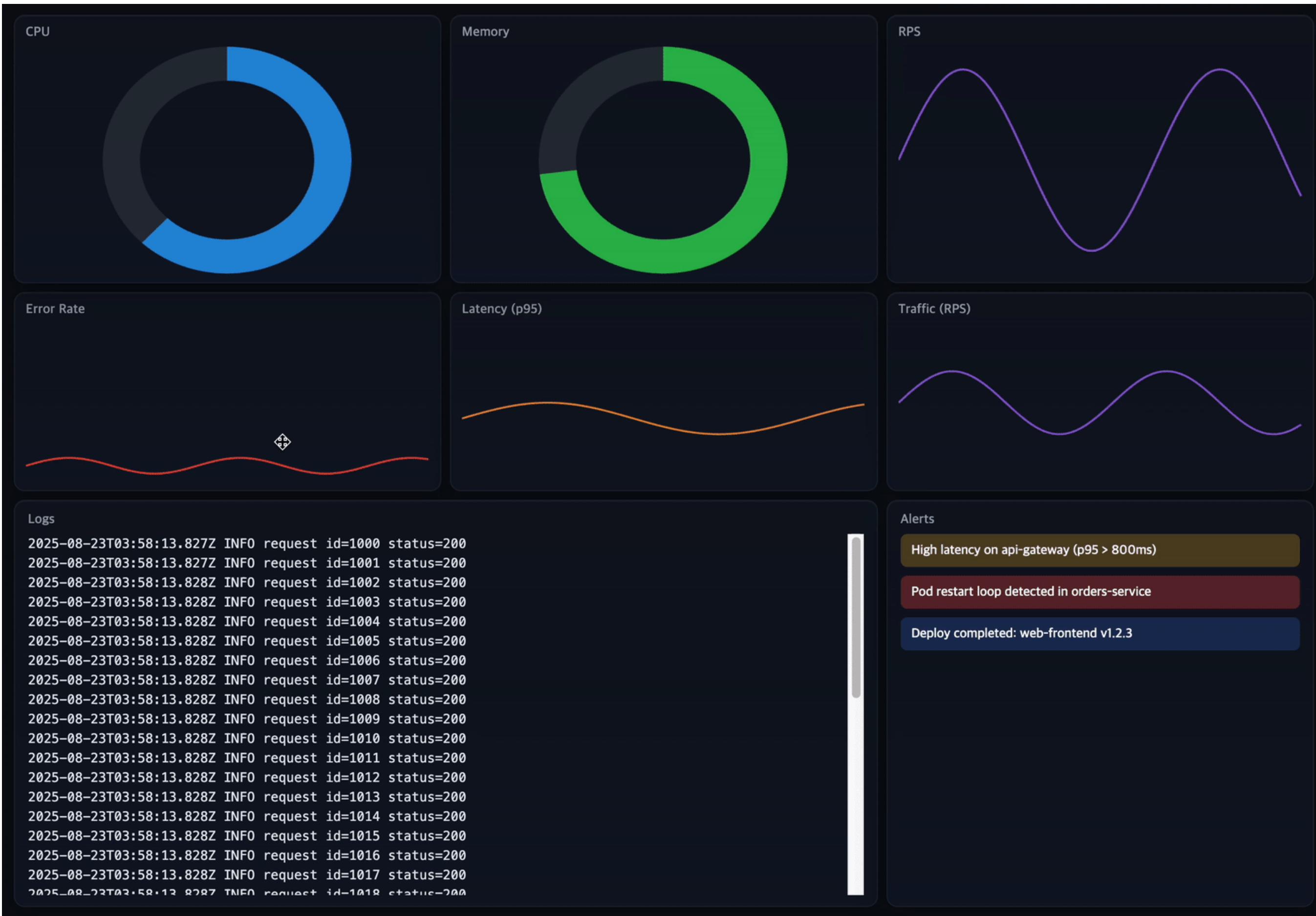
  render(data: BlockData<BoxBlockAttributes>, container: HTMLElement) {
    const el = document.createElement('div');

    el.style.width = '100%';
    el.style.height = '100%';
    el.style.borderRadius = '6px';
    el.style.background = data.attributes.color;
    el.style.display = 'flex';
    el.style.alignItems = 'center';
    el.style.justifyContent = 'center';
    el.style.color = '#fff';
    el.style.fontWeight = 'bold';
    el.textContent = data.attributes.text;
    container.innerHTML = '';
    container.appendChild(el);
  }
}
```

```
pegboard.registerExtension(new BoxBlock());

const randomColor = () => `hsl(${Math.floor(Math.random() * 360)}, 70%, 60%)`;
pegboard.addBlock({
  type: 'box',
  position: { x: 1, y: 1, zIndex: 0 },
  size: { width: 3, height: 2 },
  attributes: {
    text: 'Box',
    color: randomColor(),
  },
});
```

필요한 블럭을 정의해서 특수한 페이지를 만드는 것도 가능합니다



트러블 슈팅

에디터/빌더를 구현할 때 주의할 점

- 보안 문제
- 임시 저장 문제
- 백엔드 전송 크기 / 저장 용량
- 충돌 방지 전략

발생할 수 있는 보안 문제

- 데이터에 많은 정보가 들어가 있는 것이 문제가 될 수 있음
 - 특히 XSS(Cross-Site Scripting)에 주의
 - innerHTML이 로직에 포함된 블럭이 있다면 특별히 주의
- 링크를 걸거나 iframe을 추가할 수 있다면 더 조심해야함
 - 링크에 sanitize 로직 필요
 - iframe은 정해진 도메인만 허용
- contentEditable을 사용하는 블럭
 - 복사/붙여넣기에 대한 대책 필요

임시 저장 문제

- LocalStorage는 보통 5MB가 넘어가면 안됨
 - 만약 여러 문서를 임시 저장하는 경우 터질 수도 있으므로 localStorage 용량 체크 및 데이터 삭제 필수
 - 주기적으로 서버에 임시 저장, 브라우저 혹은 페이지를 이탈하려는 경우 서버 임시 저장 API 호출 필요
 - 용량이 큰 문서를 편집해야 한다면 IndexedDB 사용을 추천
- 경우에 따라에디터 데이터는 용량이 큰 경우가 있을 수 있음
 - 이런 경우 수정이 발생할 때마다 빠르게 저장하는 것은 추천하지 않음
 - debounce를 조금 크게 걸거는 것을 추천

백엔드 전송 크기 / 저장 용량

- 매우 주의가 필요한 문제!
 - 사용자가 가장 화가나는 것은 내가 작업한게 전부 날아갔을 때…
- 신경쓰지 않다가 문제가 발생하는 경우가 많음
- POST 요청 크기 제한이나 데이터베이스 필드 크기 제한
 - 보통 기본 POST 요청 크기는 크지 않음
 - 최대 문서 크기를 정하고 그에 따라 데이터베이스 필드 타입 설정
 - 웹 페이지에서도 최대 문서 크기를 안내해야함

충돌 방지 전략

- 여러 사용자가 한 문서에서 작업할 경우 저장 충돌이 발생할 수 있음
 - 처리를 안하는 경우 특정 사용자의 작업은 무시됨
- 서비스 특성에 따라 전략 선택이 필요
 - 잠금: 특정 사용자가 편집 중일 때 다른 사용자는 편집 불가
 - 3-way 머지: 공통 조상을 기준으로 자동 병합, 충돌이 있으면 사용자가 선택해 최종본을 만드는 전략
 - CRDT: 여러 사용자의 문서 편집을 알고리즘으로 보장하지만 구현이 복잡함
 - 최종 승리 선택: 마지막에 전송된 요청으로 무조건 덮어쓰기

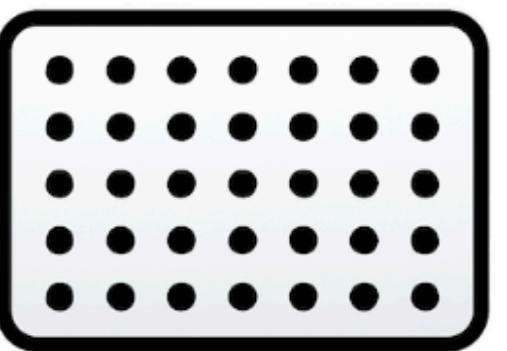
마치며

이번 발표에서 챙기지 못한 것

- 블럭 추가 전략
- Lasso 선택을 통해 여러 블럭을 드래그로 선택하는 기능
- 여러 에디터 영역이 있을 때 블럭을 공유하는 방법
- 블럭 겹치기 불가능일 때 Reflow 전략
- 블럭 정렬 기능
- 더블 클릭을 통한 블럭 내부 요소 접근

그리드 기반 에디터를 체험해보고 싶다면…

- <https://github.com/kciter/pegboard>
- 스토리북에 여러 예제를 만들어뒀어요
- 아직 완성하지는 못했어요
 - 현생 이슈…
 - 추후 다양한 예제와 기능으로 보완해서 라이브러리 배포 예정이에요



Pegboard

CI passing license MIT Storybook sponsor

The grid based editor for your web applications.



어떻게 코드로 나타내야 할지 모르겠다면

- 이번 발표에서는 전체적으로 큰 그림, 주의할 점에 대해서만 이야기했어요
- 그래서 코드를 어떻게 작성해야 할지 감이 안 올 수 있어요
- 의외로 바이브 코딩으로 구현이 잘된다는 점
 - 앞서 공개한 오픈소스도 90%는 바이브 코딩으로 작성했어요
 - 발표에서 설명한 **설계와 주의할 점을 프롬프트로** 안내하고 각 기능을 구현해달라고 요청하면 잘해줘요

궁금한 점이 있다면...

- kciter@naver.com
- LinkedIn: @kciter



감사합니다

이선협 @kciter