

PARK GYEONGHO

NOTE PAD

라인컴퓨터아트학원



INDEX

01

SCHEDULE

p. 3 - 4

02

DESIGN

p. 5 - 7

03

FOLDER STRUCTURE

p. 8 - 9

04

DESCRIPTION

p. 10 - 12

05

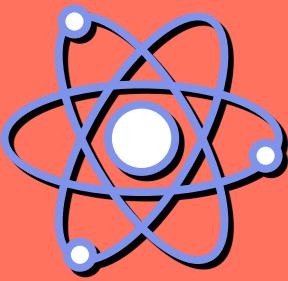
CODE

p. 13 - 16

06

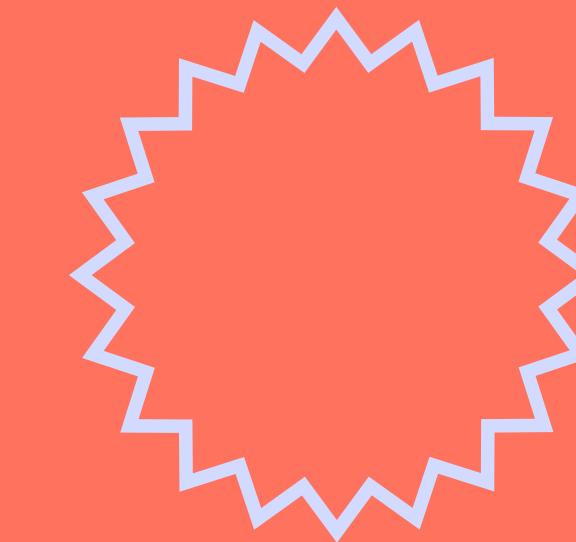
REVIEW

p. 17 - 18

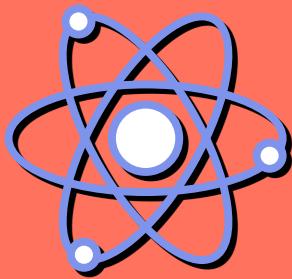


01

SCHEDULE



3



SCHEDULE

DAY 1

디자인 작업
CODE 작업

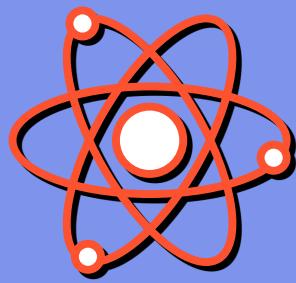
Drawing 작업
Main Code 작업

DAY 2

자료 정리

PPT 자료 정리

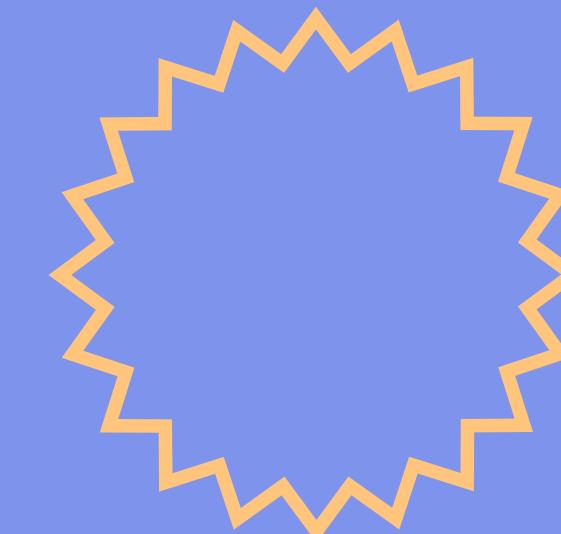


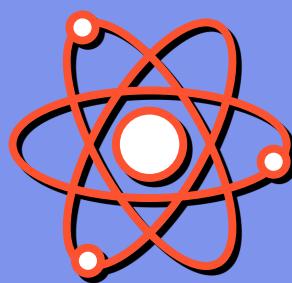


02



DESIGN





DRAWING

FIGMA Program으로 기본적인 색상과 요소 배치 작업을 진행

The screenshot shows a Figma wireframe for a 'NOTE PAD' application. The interface includes a header with the title 'NOTE PAD' and 'Frame 2'. On the left, there's a large input field labeled '메모 제목 작성 -> Input Tag'. Below it is a text area labeled '메모 내용 작성 -> Textarea Tag'. A '저장 버튼' (Save button) is positioned between these two fields. The main content area contains a list of notes. Each note has a '제목' (Title) section with '제목' and '내용' (Content) sub-sections, and a '삭제' (Delete) button. The notes themselves are labeled '메모지'. A note at the bottom is described as having 'p Tag 2개로 구분 삭제 버튼은 휴지통 모양의 icon 활용' (Delete button uses trash bin icon). The right side of the screen features a sidebar with '제목' (Title), '내용' (Content), and a '닫기' (Close) button. The bottom right corner of the slide contains the text 'C16ST10 박경호'.

NOTE PAD
Frame 2

메모 제목 작성 -> Input Tag

메모 내용 작성 -> Textarea Tag

저장 버튼

메모지 클릭 시 큰 팝업창으로 확대해서 보이기
휴지통 -> X 모양으로 아이콘 변경

제목
내용
삭제

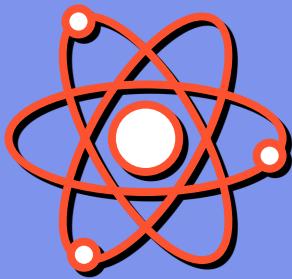
메모지

메모지 생기는 영역

p Tag 2개로 구분 삭제 버튼은 휴지통 모양의 icon 활용
Flex으로 자동 정렬 및 줄바꿈 되도록 설정

제목
내용
닫기

C16ST10 박경호



LAYOUT

- 노트에 집중 되도록 간결하게 구성
- 좌, 우를 작성과 확인 용도로 구분
- 내용 확인 용도의 팝업 창 생성

7

NOTEPAD

제목을 작성해주세요. (20자 이내)

메모 내용을 작성 해주세요.

SAVE

라인컴퓨터아트학원

PANEL

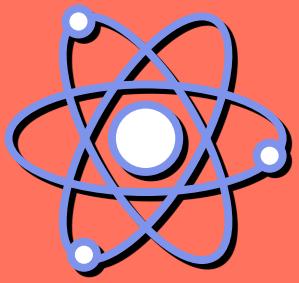
NOTEPAD

TEST 제목
TEST 내용

SAVE

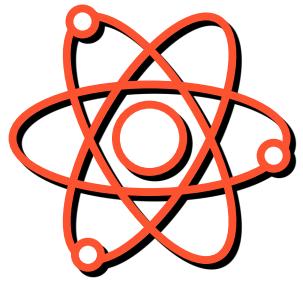
X

C16ST10

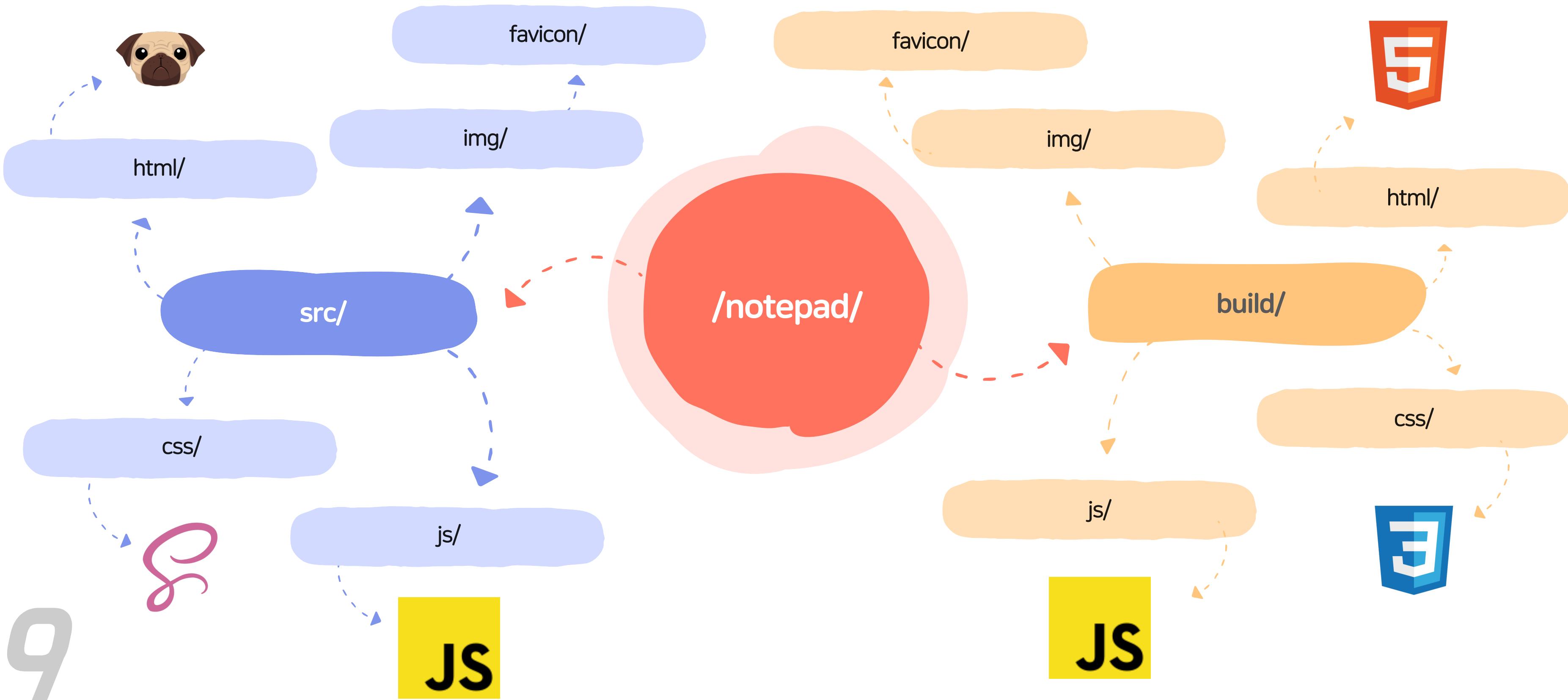


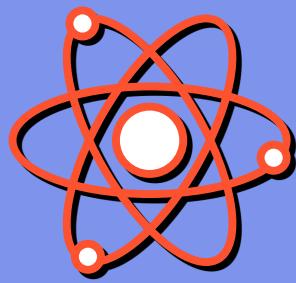
03

FOLDER STRUCTURE



FOLDER STRUCTURE

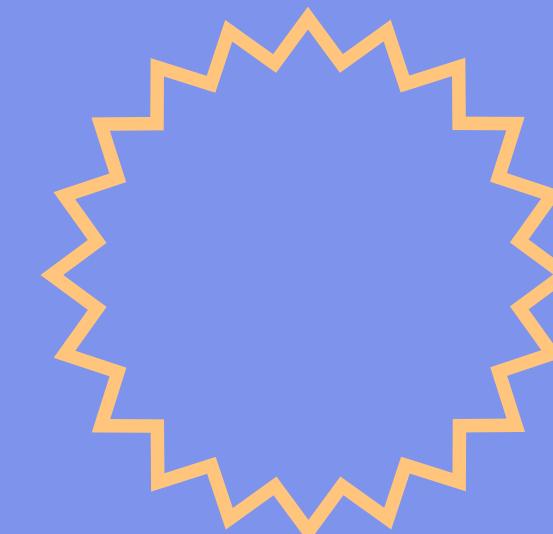


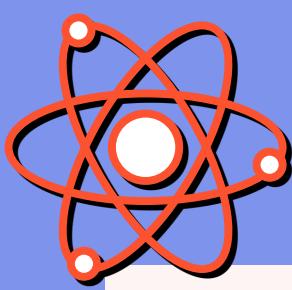


04

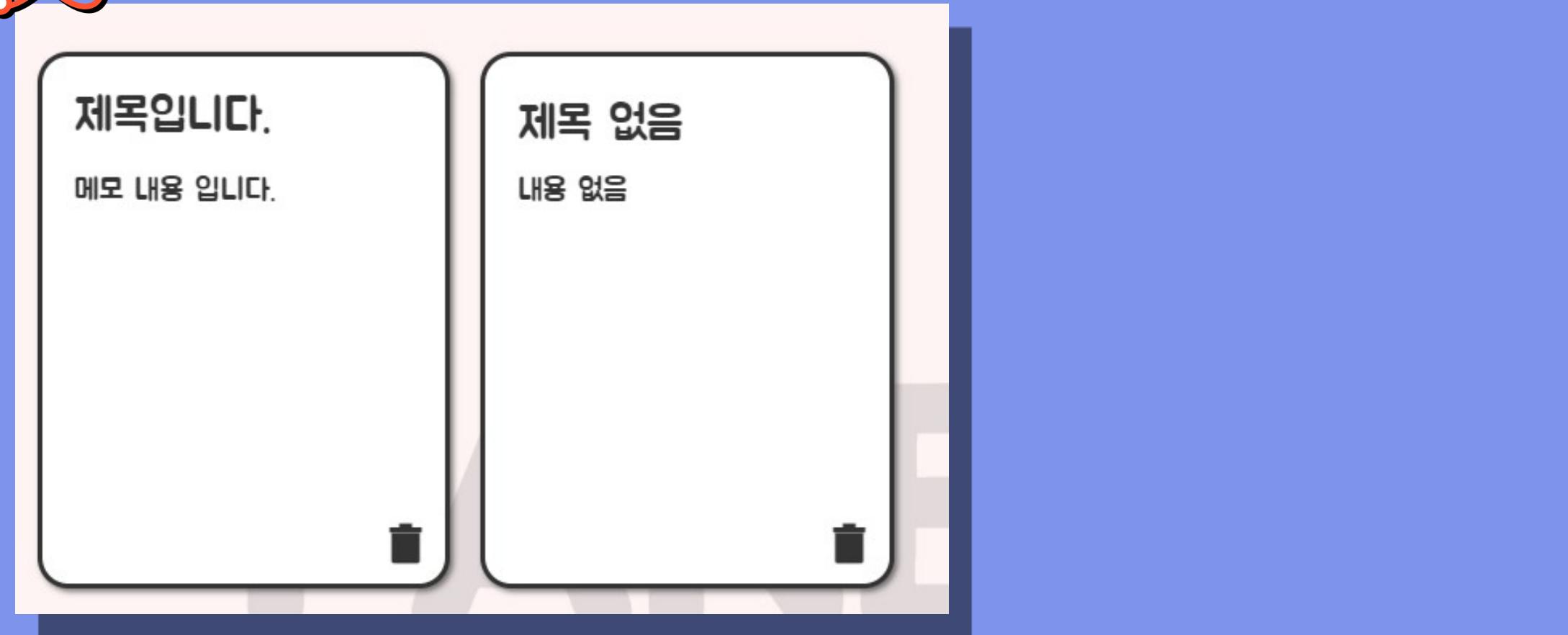


DESCRIPTION





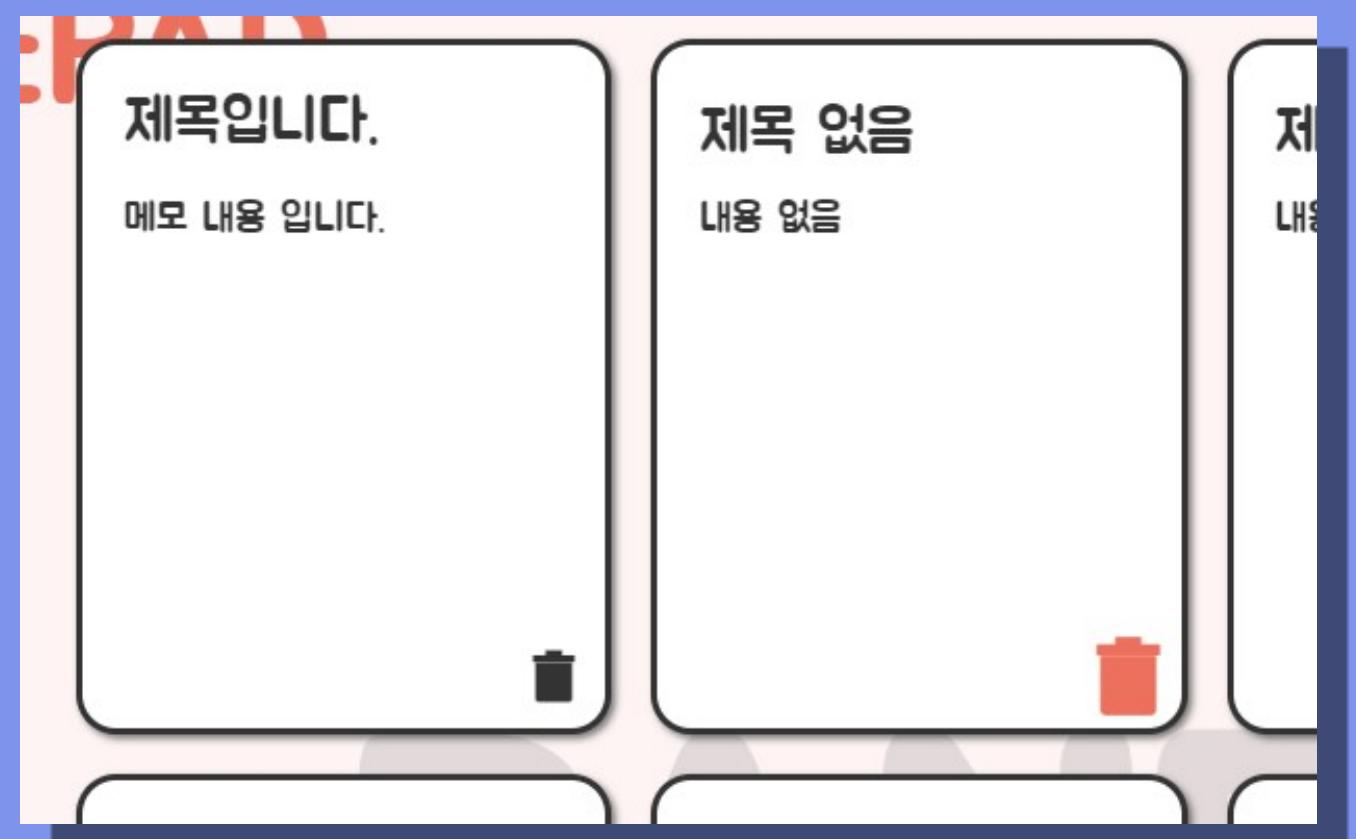
DESCRIPTION

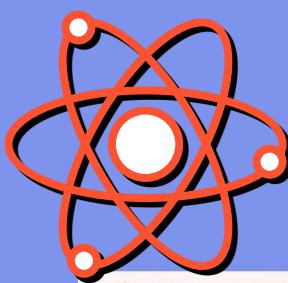


제목, 내용을 공백으로 둘 경우 "제목 없음",
"내용 없음"으로 자동 작성되어 저장



각각의 메모지에서 휴지통 아이콘 클릭 시
해당 메모지 삭제





NOTE

SAVE

제목 없음
내용 없음

제목 없음
내용 없음

제목입니다.

제목입니다.
메모 내용입니다.

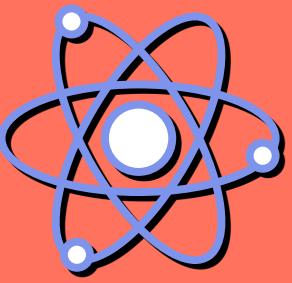
SAVE

제목 없음
내용 없음
제목 없음
내용 없음
C16ST10 발행일

DESCRIPTION

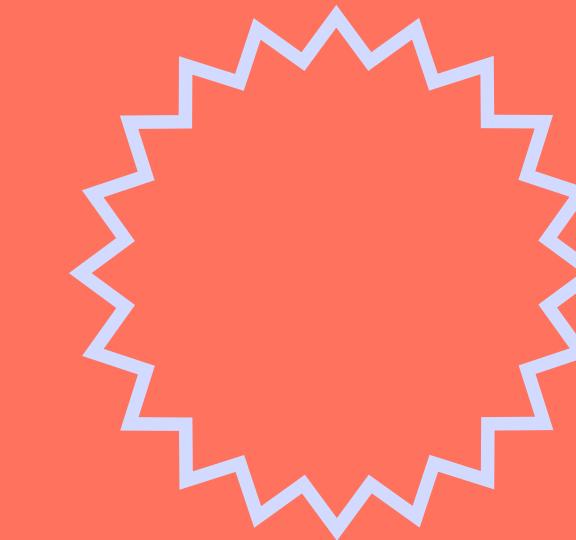
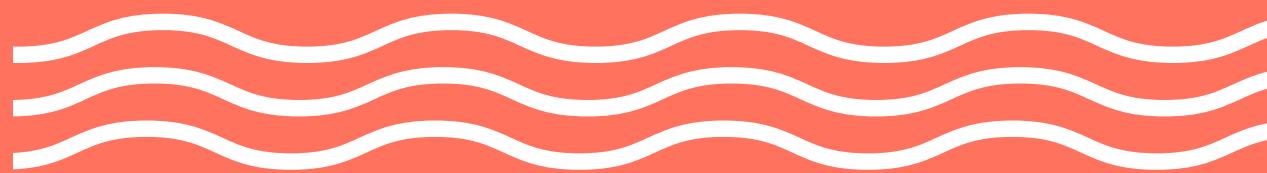
메모지 양이 일부 초과하게 되면 스크롤 하며
볼 수 있다.

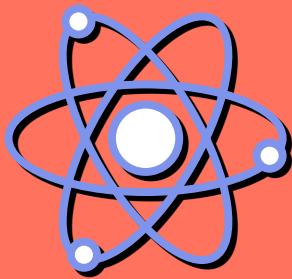
메모지를 Click 하면 해당 메모를 크게 볼 수 있다.



05

CODE

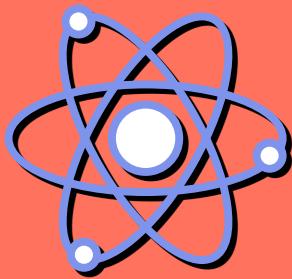




CODE - 1

- 개별의 메모지 생성을 위한 클래스 구현
- 메모지를 생성 및 저장, 관리하는 클래스 구현
- Main 클래스 Instance 및 Method 실행

```
//--- Each Note Class ---//  
class Note_Item {  
    constructor(_id, _title, _main)  
    {  
        this.id = _id;  
        this.title = _title;  
        this.main = _main;  
    }  
}  
  
//--- NotePad Main Class ---//  
class Note_Box {  
    constructor()  
    {  
        this.items = []; //--- Push Note Class ---/  
        this.text_head = document.querySelector(`#text_head`);  
        this.text_body = document.querySelector(`#text_body`);  
        this.btn_save = document.querySelector(`.btn_save`);  
        this.item_area = document.querySelector(`.item_area`);  
    }  
}  
  
//--- Main NotePad Class Instance ---//  
const note_box = new Note_Box();  
//--- NotePad Class Main Method Call ---//  
note_box.event_func();
```

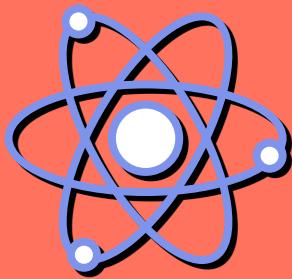


CODE - 2

- Main Method인 event_func() 실행
- Save Button Click 시 제목, 내용 값 가져오기
- 미리 정해둔 Tag값 설정 후 Panel Box에 Append
- 생성된 i, div Tag 재 정의 후 각각의 Method 실행
- 개별 메모지의 삭제 기능 Method 구현

```
event_func(){ //--- Write Text & Click Save & Main Method ---//  
    this.btn_save.addEventListener('click', (e)=>{  
        if(this.text_head.value == "") this.text_head.value = "제목 없음";  
        if(this.text_body.value == "") this.text_body.value = "내용 없음";  
        this.items.push(new Note_Item(this.note_index, this.text_head.value, this.te  
        let set_item = `<article><p>${this.text_head.value}</p><p>${this.text_body.v  
        class="xi-trash"></i><div></div></article>`;  
        $(this.item_area).append(set_item);  
        this.text_head.value = '';  
        this.text_body.value = '';  
        this.delete_items = `<.item_area>article>i`);  
        this.text_items = `<.item_area>article>div`);  
        this.click_delete();  
        this.click_item();  
        this.note_index++;  
    });
```

```
click_delete(){ //--- Click Delete Each Note ---//  
    $(this.delete_items).off('click');  
    $(this.delete_items).on('click', (e)=>{  
        $(e.target.parentNode).remove();  
        this.items.splice(this.note_index, 1);  
        this.note_index--;  
        this.delete_items = `<.item_area>article>i`);  
        this.text_items = `<.item_area>article>div`);  
    })  
}
```

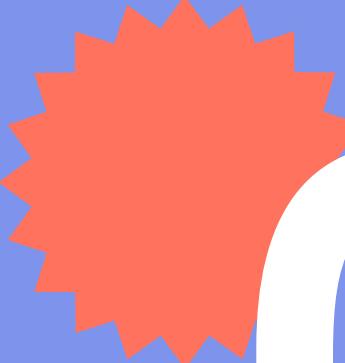


CODE - 3

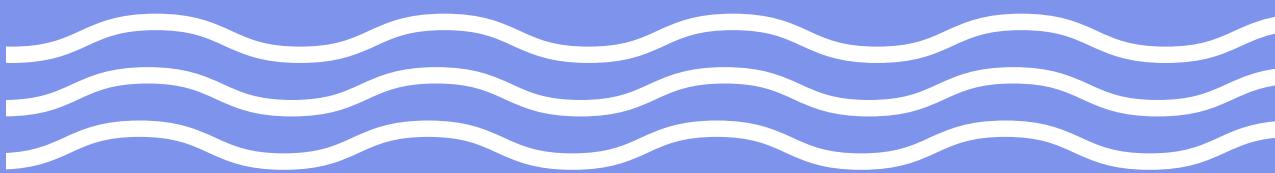
- 개별 메모지 클릭 시 해당 내용 확대 Method 작성
- 확대된 팝업창 닫기 버튼 기능 Method 작성

```
click_item(){ //--- Click Each Note & Size Up Note ---//  
    $(this.text_items).off('click');  
    $(this.text_items).on('click', (e)=>{  
        let copy_box = e.target.parentNode.cloneNode(true);  
        $(this.detail_popup).append(copy_box);  
        this.close_popup = document.querySelector(`.detail_popup>article`);  
        $(this.close_popup).toggleClass('xi-close');  
        $(this.close_popup).toggleClass('xi-trash');  
        gsap.fromTo(this.detail_popup, {scale: 0}, {scale: 1, ease: "Power 3"},  
        this.close_func());  
    })  
}
```

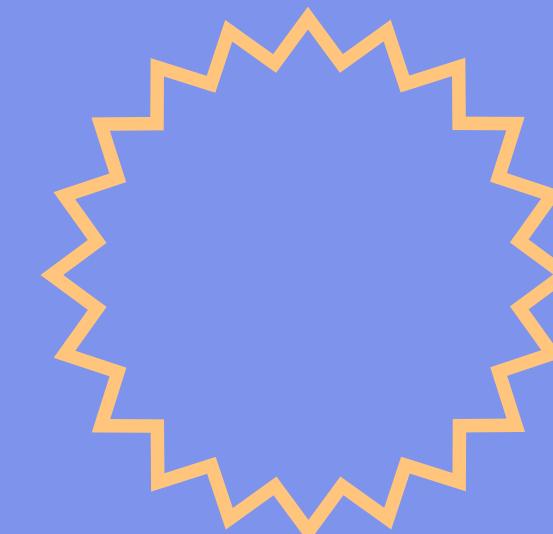
```
close_func(){ //--- Popup Page Close Method ---//  
    $(this.close_popup).off('click');  
    $(this.close_popup).on('click', (e)=>{  
        gsap.fromTo(this.detail_popup, {scale: 1}, {scale: 0, ease: "Power 3"},  
        $(this.detail_popup).empty());  
    });  
}
```



06



REVIEW



REVIEW

01

메모의 기능에 집중 되도록 과한 디자인을 제외하며
간단한 메모장 사이트를 제작했습니다.

02

간단하지만 클래스 내 Method를 통하여 각각의 Event를
Control 하는 방식과 JS 내에서 특정 Element의 부모, 자
식 요소에 대한 접근을 연습할 수 있었습니다

THANK YOU

박경호

