

egjs 오픈 서커스에 오신 것을 환영합니다

최연규 NAVER
김현준 NAVER

CONTENTS

1. 오픈 소스를 사용하는 목적
2. egjs의 소개
3. 라이브 코딩: iPhone 홈스크린 제작
4. 라이브 코딩: iPhone 사진첩 제작

사전 준비

1. node.js 12.14
2. npm 6.13
3. git
4. VSCode
5. <https://github.com/NAVER-FEPlatform/devview2020>
6. 발표자료

1.오픈 소스 사용 목적



2.1 오픈 소스는?

오픈 소스는 저작권 소유자가 소스코드를 누구에게나 라이선스에 따라 사용, 연구, 변경 및 배포할 수 있는 권한을 부여하는 것을 의미합니다.



webpack



2.2 오픈 소스를 사용하는 목적

잘 만들어진 외부 오픈 소스를 사용함으로써 개발 비용을 절감할 수 있습니다.



[Background photo created by asierromero - www.freepik.com](http://www.freepik.com)

2.2 오픈 소스를 사용하는 목적

오픈소스는 여러분들이 생각하는 것보다 훨씬 많습니다.



2.3 네이버 검색에서는 어떻게?

DEVIEW
2020

N 공연

통합 어학사전 뉴스 이미지 실시 ...

공연 정보 인기순 날짜순

전체 1,120 연극 클래식 가족/아동 뮤지컬

날짜 전국 장르 테마

< 2020. 10. 오늘 >

일	월	화	수	목	금	토
27	28	29	30	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

베르테르 공연중
2020.08.28. ~ 2020.11.01.
광림아트센터 BBCH홀
서울 | 뮤지컬 | 155분

지도 가격비교 예매

Weather

옥탑방 고양이 - 서울 공연중

N 지하철

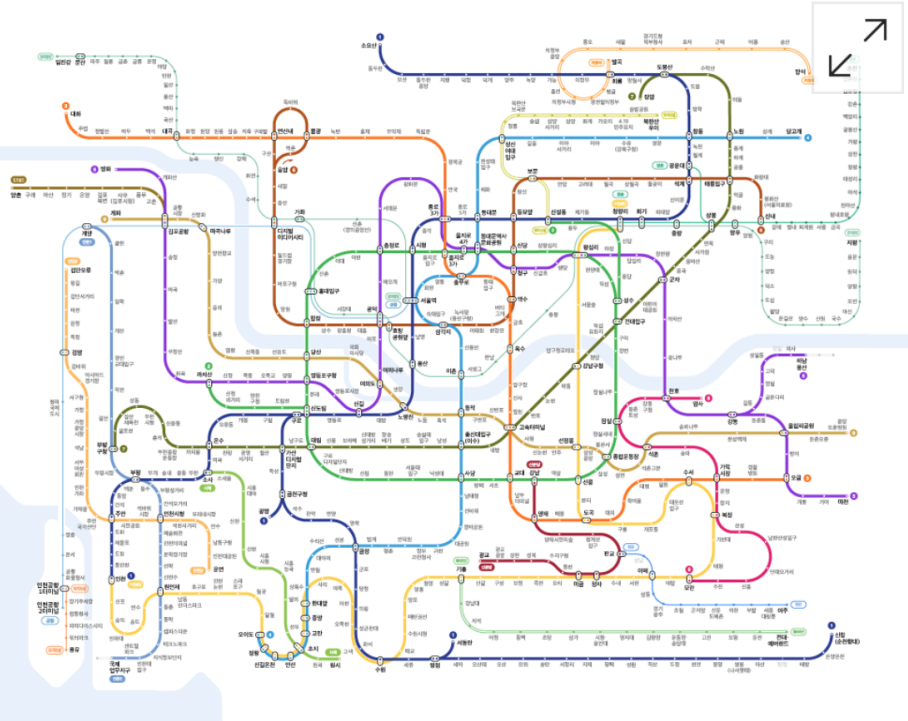
통합 이미지 어학사전 뉴스 실시 ...

연관 지하철노선도, 지하철시간표, 교통공사, 서울지하철, 지하철 시간, 메트로, 길찾기, 지하철 막차시간, ...

지하철 노선도

수도권 부산 대구 광주 대전

전체노선



출발지 검색 에서
도착지 검색 가는 길

보 정보는 우해상화에 따라 식재와 다를 수 있습니다

N 레드벨벳

통합 이미지 실시간 뉴스 지식백 ...

레드벨벳 3,064,743
가수 | SM엔터테인먼트

ON TACT 2020 GANGNAM FESTIVAL 영동대로 K-POP CONCERT
2020.10.18 SUN | 7PM
YouTube SBS KPOP 생중계

웬디 아이린 슬기 조이

네이버TV

ON TACT 2020 GANGNAM FESTIVAL 영동대로 K-POP CONCERT
2020.10.18 SUN | 7PM
YouTube SBS KPOP 생중계

온택트 2020 강남페스티벌 영동대로 K-POP 콘서트 : 티저
▶ 4,275 | 모비딕 Mobicid | 8시간전

한가위 특...
▶ 5,072 | ...

N 노래방 인기차트

통합 이미지 지식IN VIEW 동영상 ...

노래방 노래검색 내 찜곡

신곡 인기곡 10대 인기곡 발라드 인기곡

인기차트

제목/가수	노래번호	제공	찜
인기없던노래 / 돈스파이크 (With 김연우)	35275	TJ	♡
인기 없던 노래 (With 김연우) / 돈스파이크	58597	금영	♡
인기 가수 / 허윤아	59243	금영	♡
인기 있는 남자애 / 가을방학	59276	금영	♡
인기있는남자애 / 가을방학	38144	TJ	♡

< 1 / 3 >

노래방 반주

 [TJ노래방] 인기없던노래 - 돈스파이크(Wit...
TJ 노래방 | 네이버TV

N k5

통합 이미지 지식IN VIEW 동영상 ...

기아 K5
중형세단 | 2020년형



가격 2,351~3,141만원
연료 LPG, 가솔린
연비 복합 9.8~13.8km/ℓ (도심 8~14, 고속 11~16)
출력 146~180hp 엔진
배기 1,598~1,999cc

등급트림 관련영상

1.6 가솔린 터보

제원 180hp 가솔린 13.2~13.8km/ℓ 1,598cc

트렌디	2,430만원
프레스티지	2,709만원
노블레스	2,901만원
시그니처	3,141만원

2.3 네이버 검색에서는 어떻게?



etc...

2.3 네이버 검색에서는 어떻게?

네이버 공연 정보

전체연극클래식가족/아동뮤지컬콘서트뮤직페스티벌

오늘<2020.10.08.>일간공연

지역장르테마2020.10인기순날짜순

베르테르

20.08.28.~20.11.01.
광림아트센터 BBCH홀

가격비교

옥탑방 고양이 - 서울

10.04.06.~오픈런
틴틴홀

예매

쉬어매드니스

15.11.12.~오픈런
콘텐츠박스

예매

오마이갓

17.07.01.~오픈런
봄날아트홀 2관

예매

일월화수목금토

123

45678910

11121314151617

18192021222324

25262728293031

오늘날짜2020.10.08. (목)

N공연

통합어학사전뉴스이미지실시...

공연 정보인기순날짜순

전체1,120연극클래식가족/아동뮤지컬

날짜전국장르테마

<2020. 10.오늘>

일월화수목금토

27282930123

45678910

11121314151617

18192021222324

25262728293031

베르테르 공연중
2020.08.28. ~ 2020.11.01.
광림아트센터 BBCH홀
서울 | 뮤지컬 | 155분



2.3 네이버 검색에서는 어떻게?

N

한 번 다녀왔습니다 같은 드라

×

Q

통합

이미지

실시간

동영상

지식

...

한 번 다녀왔습니다

+

구독 72,446

...

드라마

15세이상

완결

청률

편성정보

관련앨범

함께 볼만한 드라마



오! 삼광빌라!



도시락



구해줘 2



파랑새의 집



내 딸 서영이



왕십리







N

영화 기생충 포토

×

Q

통합

이미지

동영상

VIEW

지식

...

기생충

상영중

...

PARASITE

2019

평점

무비클립

포토

리뷰

명대사

추천

스틸컷 103



↓

포스터 10



N

k5

×

Q

기아 K5

중형세단 | 2020년형

KIA



가격

2,351~3,141만원

연료

LPG, 가솔린

연비

복합 9.8~13.8km/ℓ (도심 8~14, 고속 11~16)

출력

146~180hp 엔진

배기

1,598~1,999cc

등급트림

관련영상

1.6 가솔린 터보

↑

제원

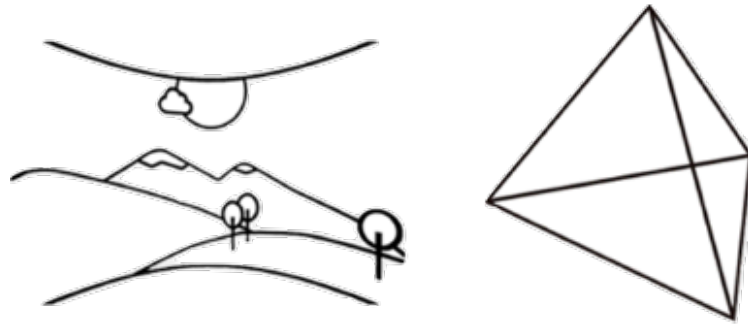
180hp 가솔린 13.2~13.8km/ℓ 1,598cc

트렌디	2,430만원
프레스티지	2,709만원
노블레스	2,901만원
시그니처	3,141만원

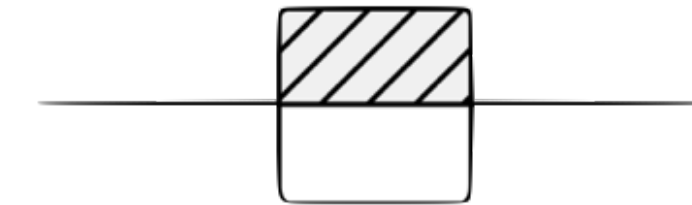
2.egjs

1. egjs의 소개

< Flicking >



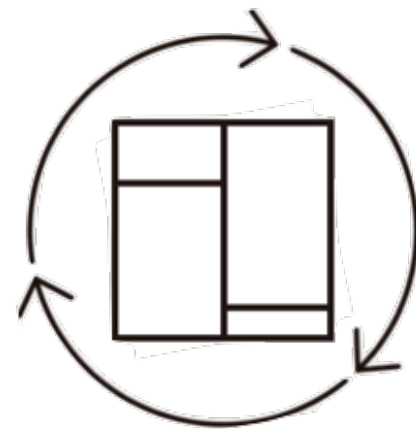
view360



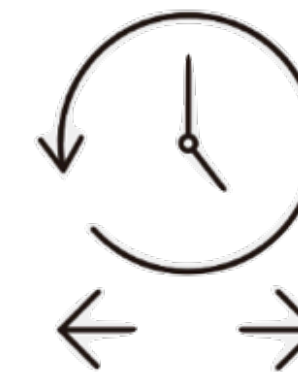
Visible



egjs



InfiniteGrid



Persist

3.오픈 소스를 어떻게 사용할까?

3.1 GitHub

Git 의 분산 버전 제어 및 소스 코드 관리 기능과 자체 기능을 제공합니다. 모든 프로젝트에 대한 액세스 제어 및 버그 추적, 기능 요청, 작업 관리, 지속적인 통합 및 위키와 같은 여러 협업 기능을 제공합니다

The screenshot shows the GitHub interface for the repository 'naver/egjs-infinitegrid'. The top navigation bar includes 'Pull requests', 'Issues', 'Codespaces', 'Marketplace', and 'Explore'. The repository name is 'naver/egjs-infinitegrid' with 18 watchers, 689 stars, and 47 forks. The main content area shows a list of files and folders with their commit history. The 'About' section describes the module as a tool for arranging card elements. The 'Releases' section shows the latest release '3.6.3 Release (2020-05-06)'.

File/Folder	Commit Message	Commit Date
.github	Update stale.yml	21 days ago
config	fix(Parallax): support Parallax type (#268)	2 years ago
demo	chore(release): Release 3.6.3	5 months ago
packages	fix: fix loading element for first mount #340 (#342)	4 months ago
rollup	fix(InfiniteGrid): convert from JavaScript to TypeScript Project (#257)	2 years ago
src	fix: fix SquareLayout's outline caculation (#339)	5 months ago
storybook	demo: migrate to storybook 6 (#355)	18 days ago
test	test: remove useless code	5 months ago
.babelrc	fix(InfiniteGrid): make Rollup config (#244)	2 years ago
.editorconfig	style: add missing editorconfig	11 months ago
.eslintrc	fix(InfiniteGrid): make Rollup config (#244)	2 years ago

```
$ git init
Initialized empty Git repository in /tmp/tmp.IMBYSY7R8Y/.git/
$ cat > README << 'EOF'
> Git is a distributed revision control system.
> EOF
$ git add README
$ git commit
[master (root-commit) e4dcc69] You can edit locally and push
to any remote.
 1 file changed, 1 insertion(+)
 create mode 100644 README
$ git remote add origin git@github.com:cdown/thats.git
$ git push -u origin master
```

출처: <https://en.wikipedia.org/wiki/GitHub>

3.2 CDN

CDN (콘텐츠 전송 네트워크) 서버 그룹이 여러 위치에 흩어져있다. 이러한 서버는 데이터의 중복 사본을 저장하므로 서버는 각 최종 사용자와 가장 가까운 서버를 기반으로 데이터를 가져올 수 있고 트래픽의 영향을 덜 받는다.

1. cdnjs
2. jsdelivr
3. unpkg
4. jspm

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/egjs-infinitegrid/3.5.0/infinitegrid.min.js"></script>
<script>
  new eg.InfiniteGrid(...);
</script>
```

3.3 npm(node package manager)

JavaScript 프로그래밍 언어 용 패키지 관리자 입니다 .

 ✕ Search Sign Up Sign In

10 packages found

Sort Packages

Optimal

Popularity

Quality

Maintenance

@egjs/infinitegrid exact match

A module used to arrange card elements including content infinitely according to layout type. With this module, you can implement various layouts composed of different card elements whose sizes vary. It guarantees performance by maintaining the number of

infinitegrid infinite scroll layout DOM html gallery grid pinterest masonry GridLayout JustifiedLayout PackingLayout FrameLayout

 **younkue** published 3.6.3 • 5 months ago

@egjs/react-infinitegrid

A react component that can easily use egjs-infinitegrid

 **younkue** published 3.0.5 • 4 months ago

@egjs/vue-infinitegrid

A module used to arrange card elements including content infinitely according to layout type. With this module, you can implement various layouts composed of different card elements whose sizes vary. It guarantees performance by maintaining the number of

 **younkue** published 3.0.3 • 5 months ago

@egjs/svelte-infinitegrid

A Svelte component that can easily use egjs-infinitegrid

svelte virtual virtualized virtual-scroll virtual-list infinitegrid infinite scroll layout DOM html gallery grid pinterest View more

 **younkue** published 3.0.4 • 5 months ago

출처:[https://en.wikipedia.org/wiki/Npm_\(software\)](https://en.wikipedia.org/wiki/Npm_(software))

3.3 npm(node package manager)

프로젝트 내 package.json에 의존성 패키지를 추가하고 설정할 수 있습니다.

```
$ npm install
```

```
$ npm install @egjs/flicking
```

```
{ } package.json > ...  
6   "devDependencies": {  
7     "awesome-typescript-loader": "^5.2.0",  
8     "copy-webpack-plugin": "^6.2.0",  
9     "cross-env": "^5.0.1",  
10    "css-loader": "^1.0.0",  
11    "ejs-loader": "^0.3.0",  
12    "eslint": "^5.4.0",  
13    "html-loader": "^0.5.5",  
14    "html-webpack-plugin": "^3.2.0",  
15    "istanbul-instrumenter-loader": "^3.0.1",  
16    "rimraf": "^2.6.1",  
17    "source-map-loader": "^0.2.1",  
18    "style-loader": "^0.22.1",  
19    "tslint": "^5.5.0",  
20    "tslint-loader": "^3.5.3",  
21    "typescript": "^3.0.1",  
22    "webpack": "^4.17.0",  
23    "webpack-cli": "^3.1.0",  
24    "webpack-dev-server": "^3.1.5"  
25  },
```

iPhone 홈스크린 제작 Live Coding

사전 준비

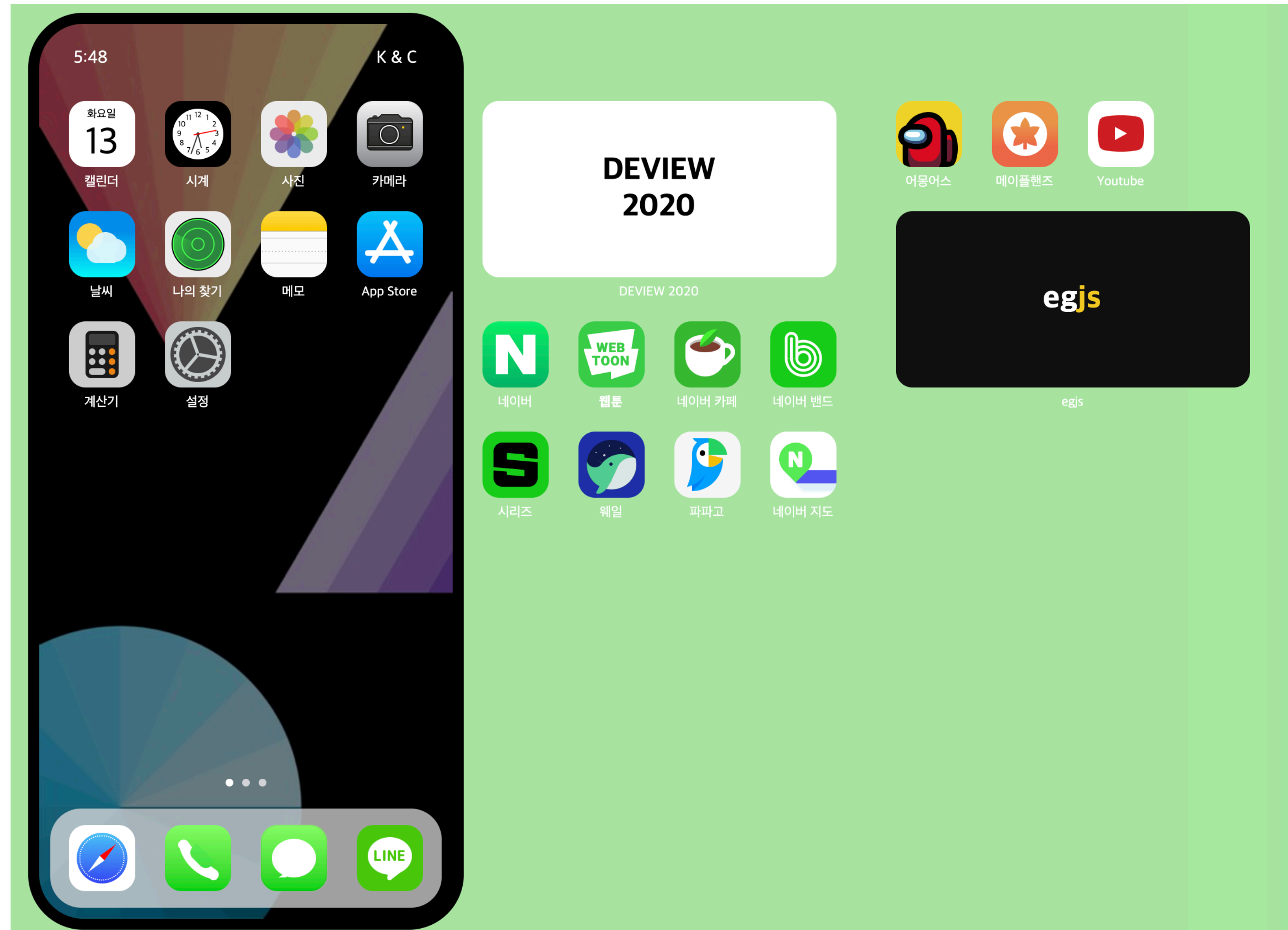
1. node.js 12.14
2. npm 6.13
3. git
4. VSCode
5. 발표 자료

\$ git clone <https://github.com/NAVER-FEPlatform/devview2020>

VS Code를 실행하고 devview2020 폴더를 불러온다.

시작

\$ npm i
\$ npm start



앞으로 만들 기능

1. 화면 밖으로 나온 페이지 숨기고 좌우로 드래그 하여 보이는 페이지 이동
2. 페이지 이동하면 도트 페이지네이션 변경

1. 화면 밖으로 나온 페이지 숨기고 좌우로 드래그 하여 보이는 페이지 이동

\$ npm i @egjs/flicking

```
AL01333526:devview2020-live-coding user$ npm i @egjs/flicking  
npm WARN @egjs/devview-live-coding@0.0.1 No repository field.
```

```
+ @egjs/flicking@3.5.1  
updated 1 package in 10.089s
```

```
35 packages are looking for funding  
run `npm fund` for details
```

```
AL01333526:devview2020-live-coding user$
```

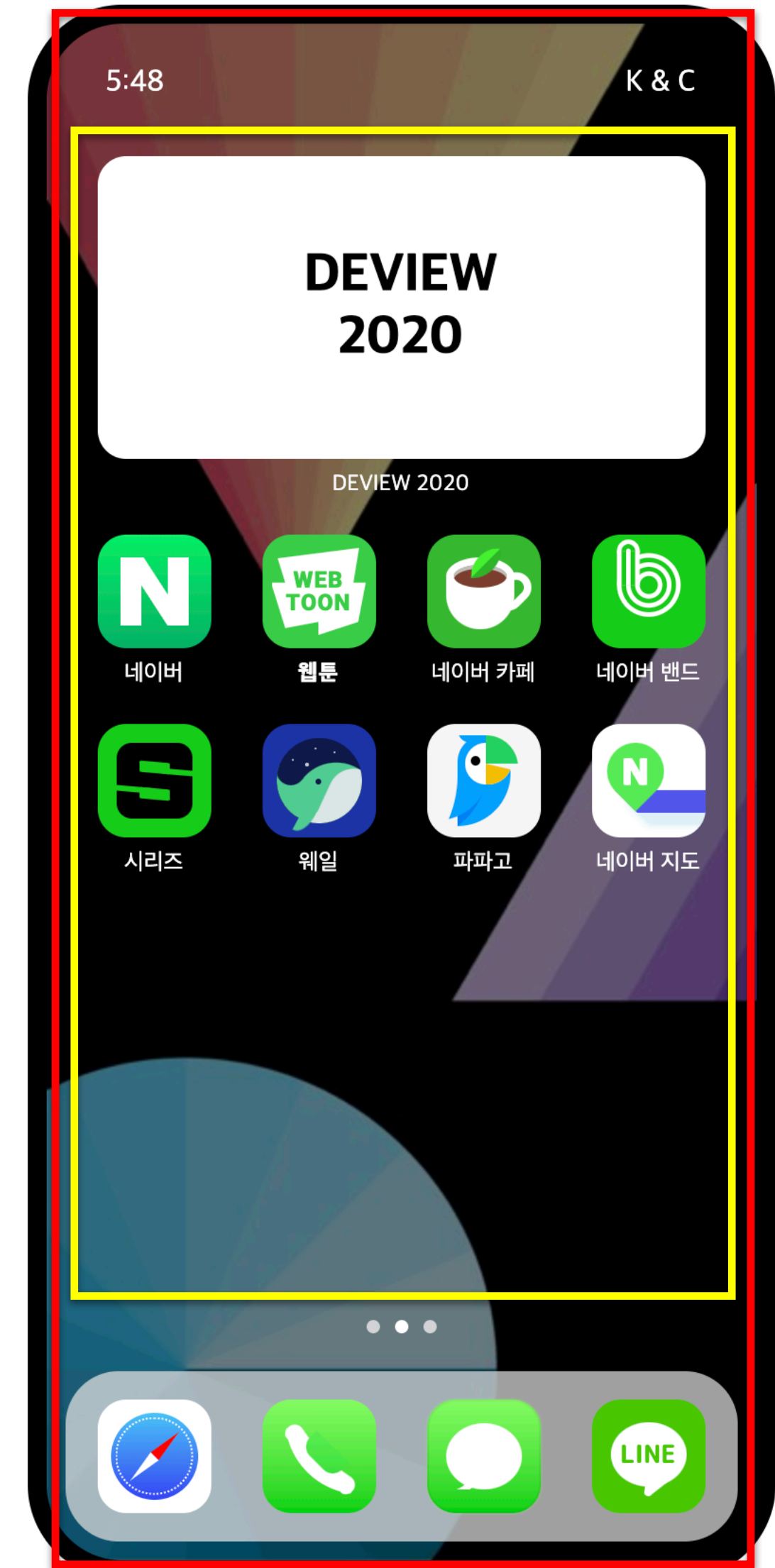

1. 화면 밖으로 나온 페이지 숨기고 좌우로 드래그 하여 보이는 페이지 이동

Flicking 사용법

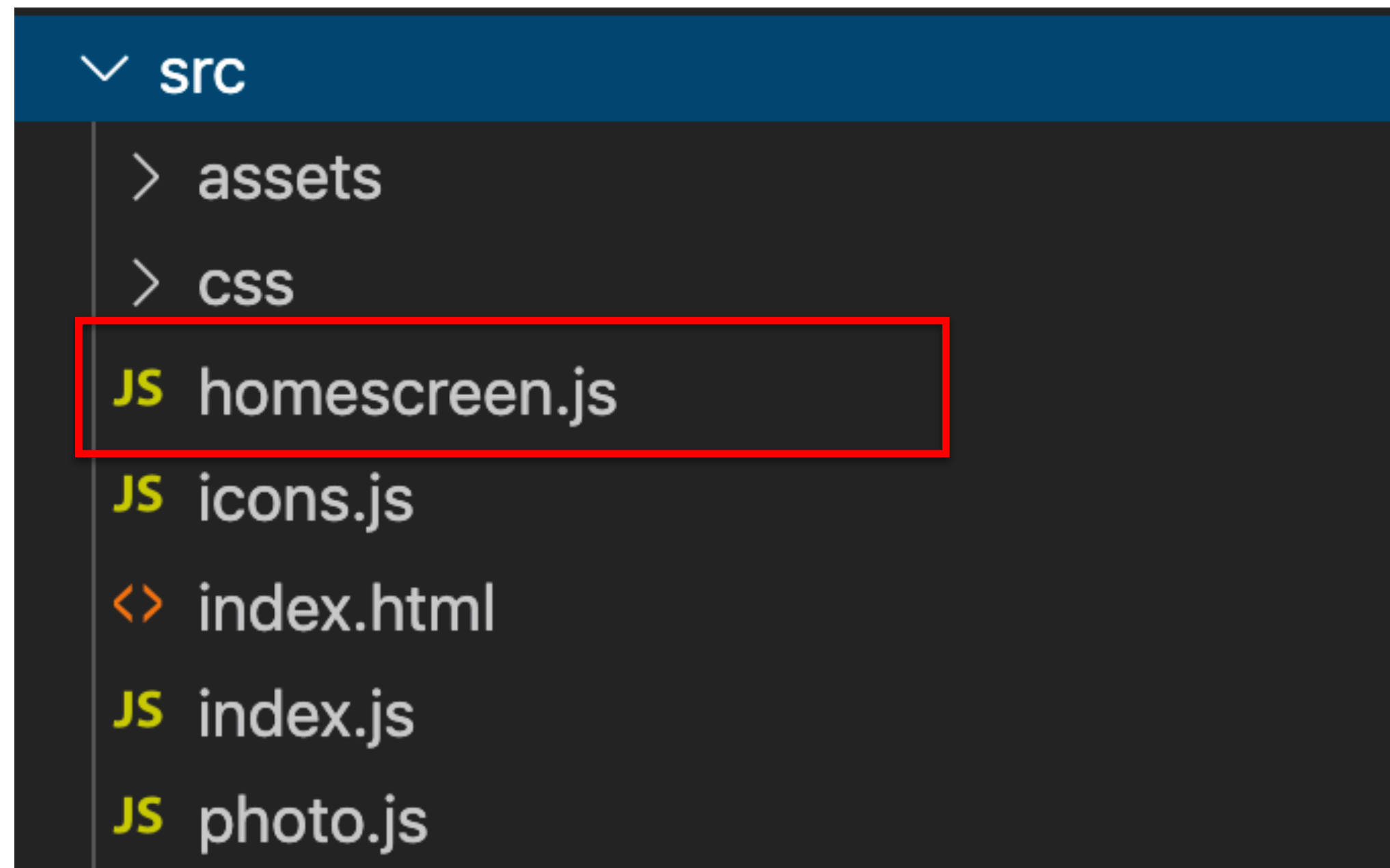
- README: <https://github.com/naver/egjs-flicking>
- API: <https://naver.github.io/egjs-flicking/release/latest/doc/index.html>

Flicking 영역에 대해

```
<div class="homescreen">  
  <div class="container">  
    <div class="page"></div>  
    <div class="page"></div>  
    <div class="page"></div>  
  </div>  
  <div class="pagination"></div>  
</div>
```



1. 화면 밖으로 나온 페이지 숨기고 좌우로 드래그 하여 보이는 페이지 이동



Flicking을 해당 파일에서 작성

1. 화면 밖으로 나온 페이지 숨기고 좌우로 드래그 하여 보이는 페이지 이동

```
import Flicking from "@egjs/flicking";  
  
const flicking = new Flicking(".homescreen .container");
```

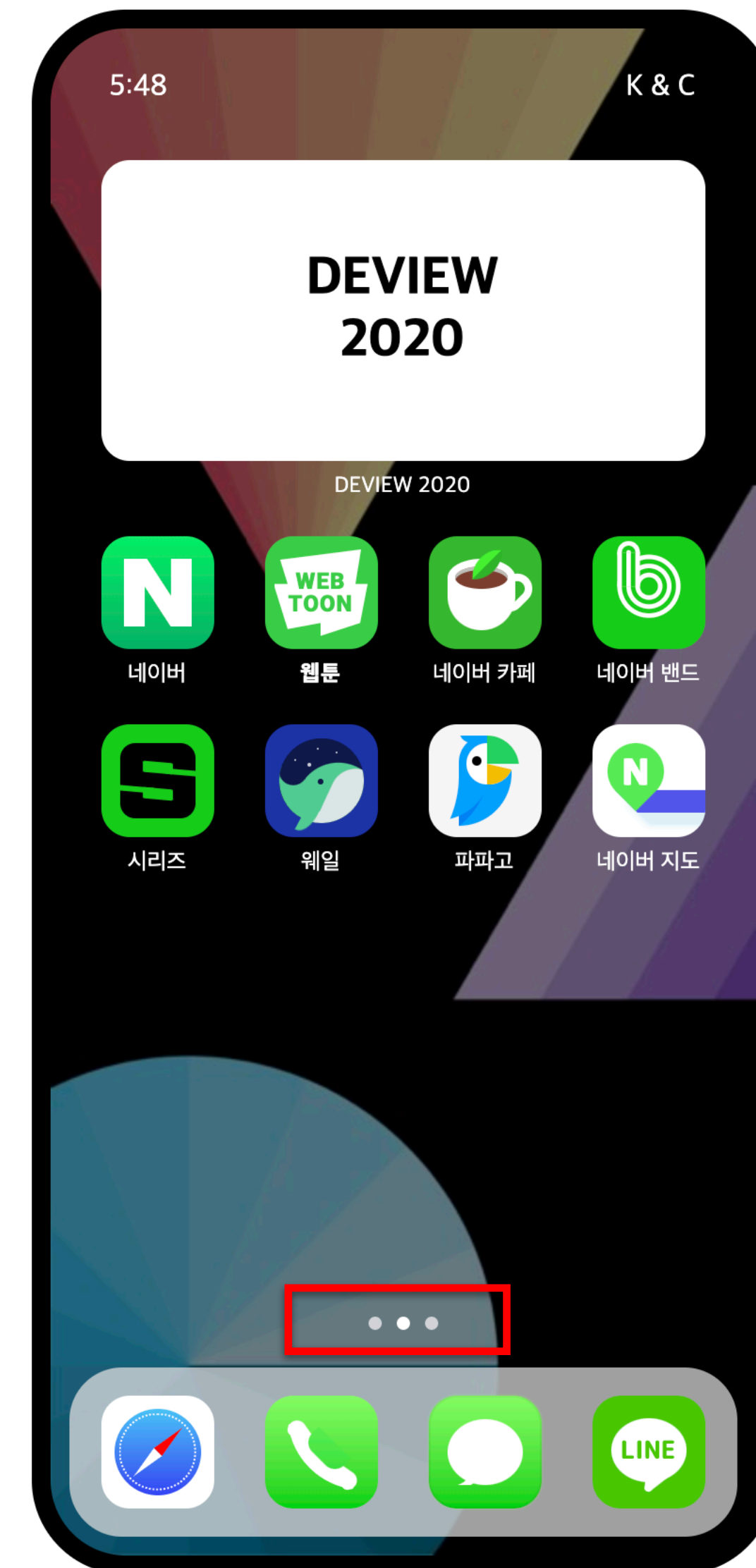
Flicking을 사용하기 위해 import 이후
아래처럼 Flicking 인스턴스 생성

시연

\$ npm start

홈스크린 개발

2. 페이지 이동하면 도트 페이지네이션 변경



```
Methods
addPlugins
append
destroy
disableInput
enableInput
getAllPanels
getCloneCount
getCurrentPanel
getElement
getIndex
getLastIndex
getPanel
getPanelCount
getStatus
getVisiblePanels
hasOn ⓘ
isPlaying
moveTo
next
off ⓘ
on ⓘ
once ⓘ
prepend
prev
remove
removePlugins
replace
resize
setLastIndex
setStatus
trigger ⓘ
```

```
Events
change
holdEnd
holdStart
move
moveEnd
moveStart
needPanel
restore
select
visibleChange
```

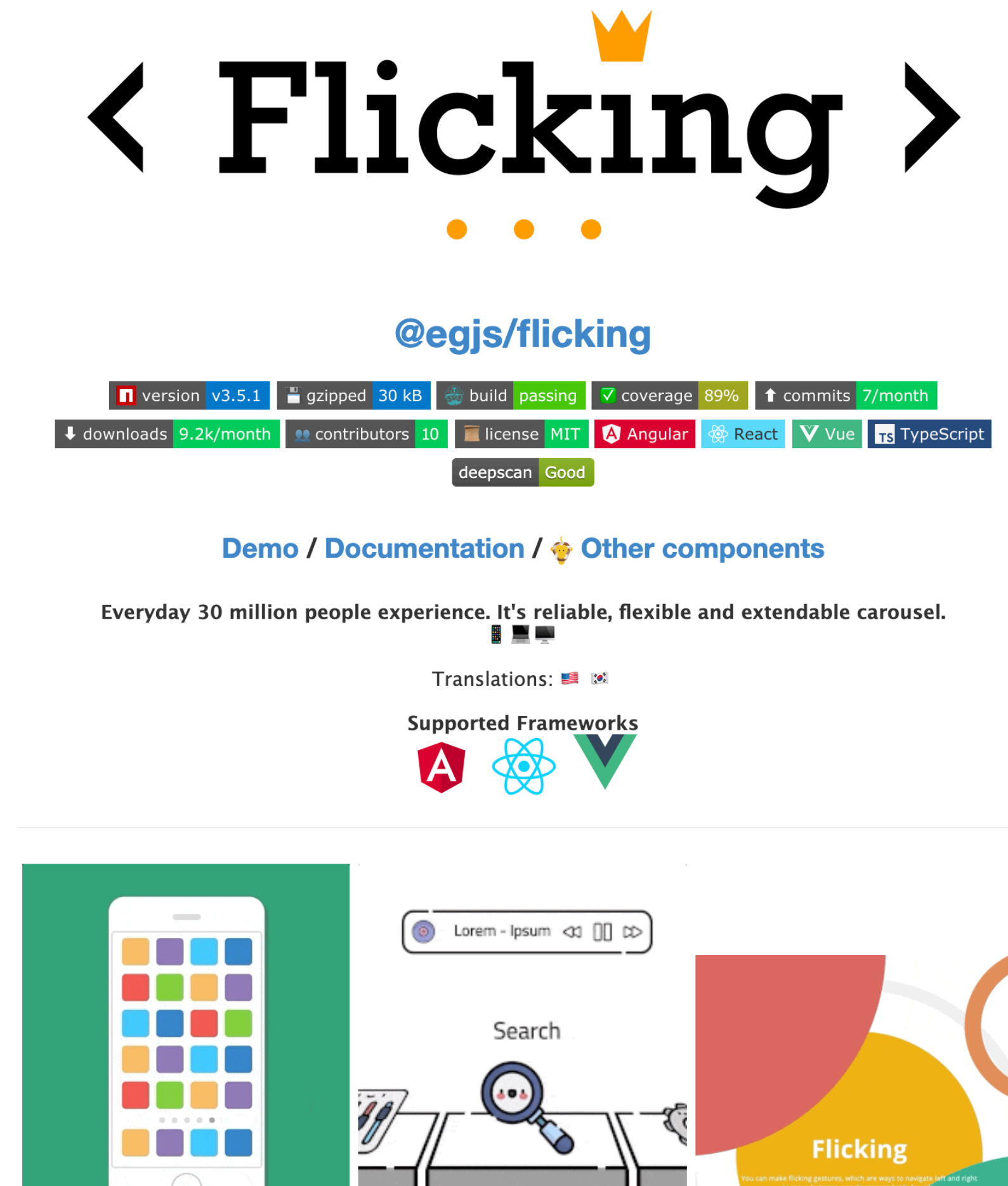
```
egjs Flicking

Search Documentations

MoveTypeFreeScrollOption
MoveTypeOption
MoveTypeSnapOption
NeedPanelEvent
Plugin
SelectEvent
VisibleChangeEvent

Methods
addPlugins
append
destroy
disableInput
enableInput
getAllPanels
getCloneCount
getCurrentPanel
getElement
getIndex
getLastIndex
getPanel
getPanelCount
getStatus
getVisiblePanels
hasOn ⓘ
isPlaying
moveTo
next
off ⓘ
on ⓘ
once ⓘ
prepend
prev
remove
removePlugins
replace
resize
setLastIndex
setStatus
trigger ⓘ

Events
change
holdEnd
holdStart
move
moveEnd
moveStart
needPanel
restore
select
visibleChange
```



Events

change

types.ts, line 262

`restore` 되지 않고, 인덱스가 변경될 것임을 나타내는 이벤트. 실제 인덱스는 `moveEnd` 이벤트에서 변경된다.

사용자가 입력을 마쳤을 때, 혹은 메소드를 통해 이동을 시작했을 때 발생한다.

동일 패널로 이동시에는 발생되지 않지만, `circular` 모드에서 한 바퀴 이상 순환하여 동일 패널로 도착했을 때에도 발생된다.

이벤트의 `stop()` 을 호출시 패널로의 이동을 막는다.

`moveTo()` , `prev()` , `next()` 와 같은 메소드에 의해 호출되었을 경우 `event` 내의 `axesEvent` 프로퍼티 값은 `undefined`이다.

Type:

- [eg.Flicking.ChangeEvent](#)

Type Definitions

eg.Flicking.ChangeEvent TSInterface

`restore` 되지 않고, 인덱스가 변경될 것임을 나타내는 이벤트. 실제 인덱스는 `moveEnd` 이벤트에서 변경된다.

Properties:

Name	Type	Argument	Description
<code>type</code>	string		이벤트명
<code>index</code>	number		애니메이션 종료 시점에 도착할 것으로 예측되는 패널의 인덱스.
<code>panel</code>	FlickingPanel null		애니메이션 종료 시점에 도착할 것으로 예측되는 패널.
<code>progress</code>	number		첫 번째 패널로부터 얼마만큼 진행했는지를 나타내는 진행도. 첫번째 패널에서 0, 마지막 패널에서 1의 값을 갖는다.
<code>isTrusted</code>	boolean		사용자 액션("mouse" 또는 "touch")에 의해 이벤트가 생성된 경우 <code>true</code> . 그 외는 <code>false</code> .
<code>holding</code>	boolean		사용자가 입력 장치를 통해 입력중인지 여부. (마우스 장치라면 'mousedown' 여부, 터치 장치라면 'touchmove' 여부)
<code>stop</code>	() => void		이벤트의 기본동작을 취소하고, 해당 이벤트 뒤에 발생할 이벤트들을 전부 발생하지 않도록 한다. <code>-End</code> 가 접미사로 붙은 이벤트에서는 유효한 동작을 하지 않는다.
<code>direction</code>	ValueOf<Direction> null		예측되는 패널 이동 방향.
<code>axesEvent</code>	any	<optional>	내부의 Axes 인스턴스로부터 발생된 원본 이벤트. <code>moveTo()</code> , <code>prev()</code> , <code>next()</code> 와 같은 메소드에 의해 발생되었을 경우 <code>undefined</code> .
<code>currentTarget</code>	Flicking		이벤트를 발생시킨 Flicking의 인스턴스

2. 페이지 이동하면 도트 페이지네이션 변경

```
▼<div class="pagination"> == $0  
  <div class="dot active">1</div>  
  <div class="dot">2</div>  
  <div class="dot">3</div>  
</div>
```

2. 페이지 이동하면 도트 페이지네이션 변경

```
const dots = document.querySelectorAll(".dot");

flicking.on("change", e => {
  dots.forEach(dot => {
    dot.classList.remove("active");
  });
  dots[e.index].classList.add("active");
});
```

시연

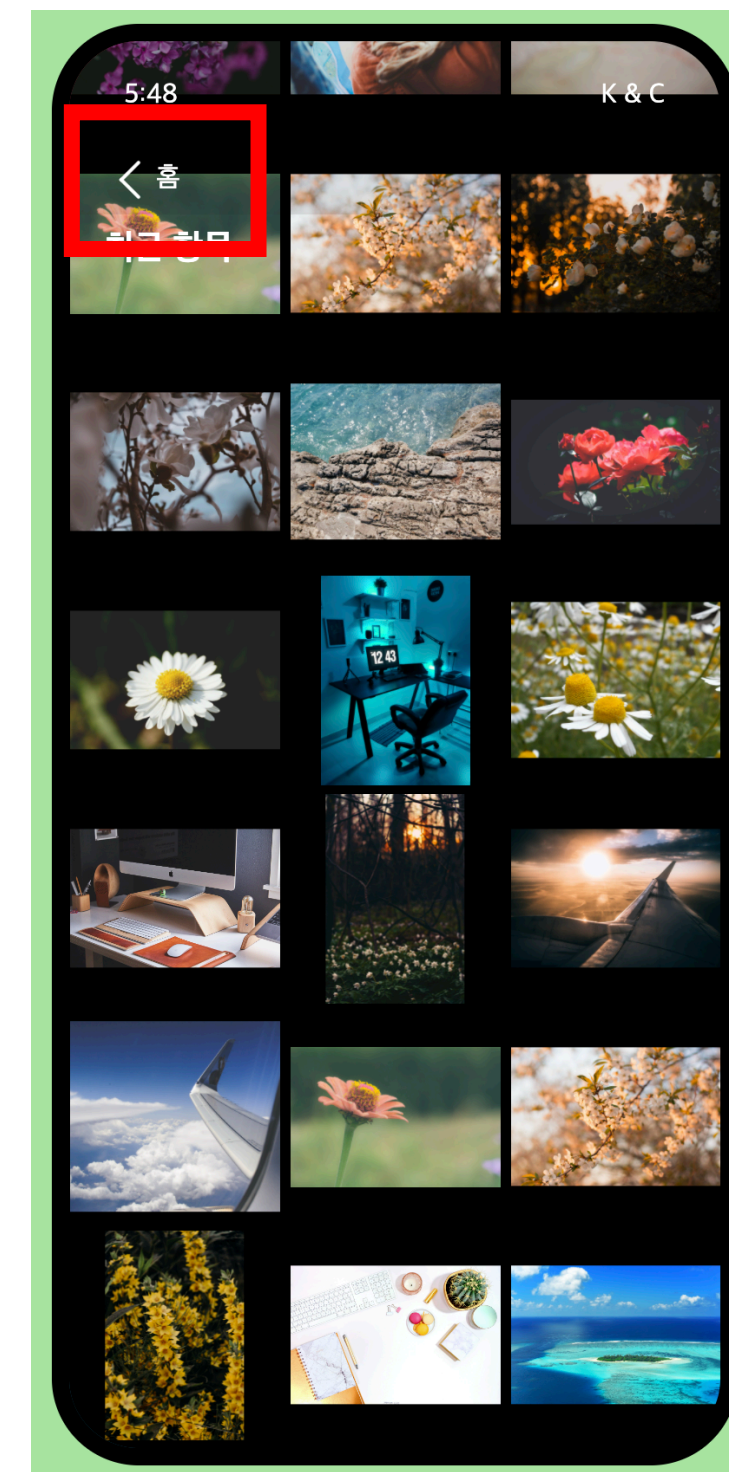
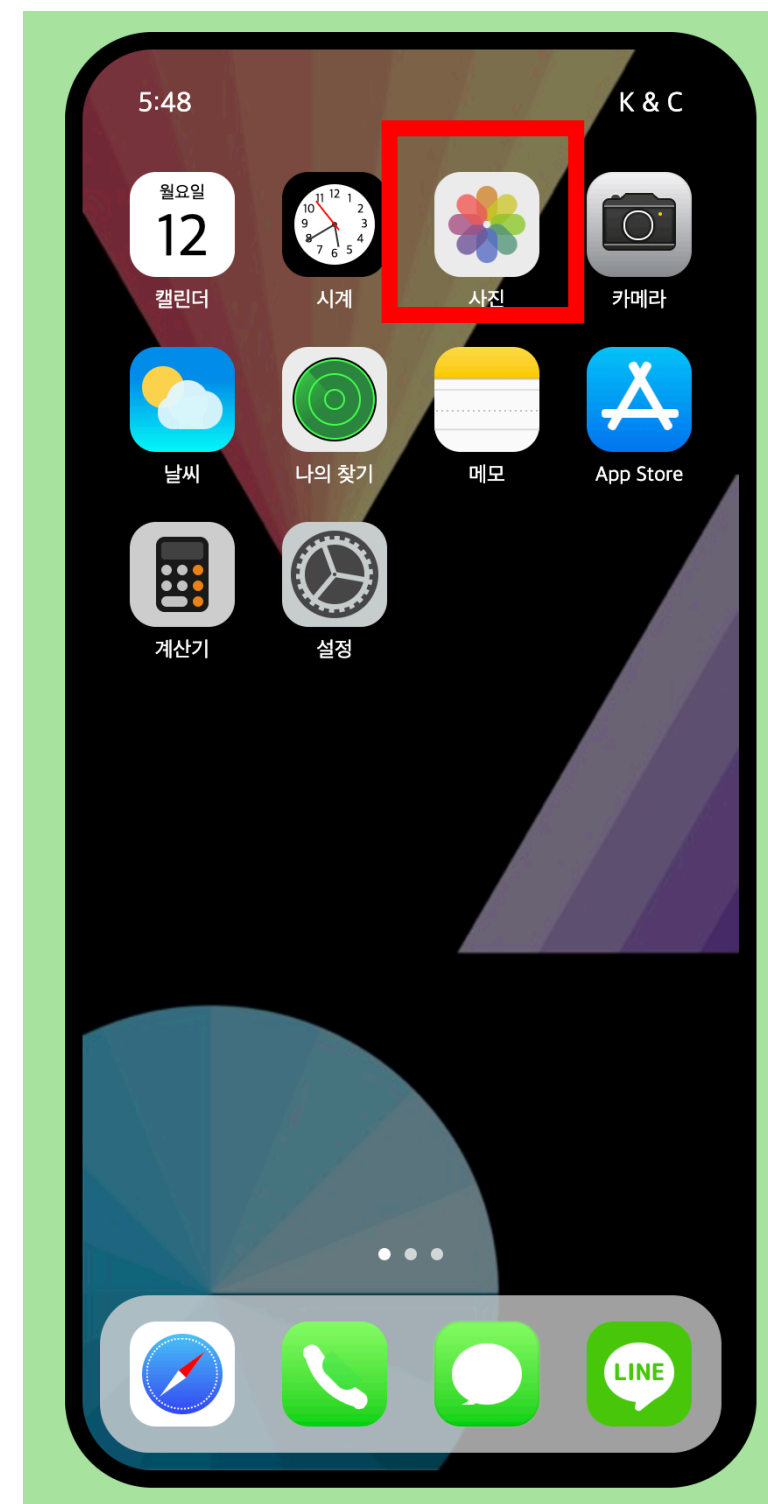
N DEVIEW
2020

iPhone 앱 개발 Live Coding

iPhone 사진첩 개발 Live Coding

앞으로 만들 기능

1. 사진첩 아이콘을 누르면 사진첩 앱이 열리는 기능 추가
2. 뒤로 가기 버튼을 누르면 사진첩 앱이 닫히는 기능 추가



1. 사진첩 아이콘을 누르면 사진첩 앱이 열리는 기능 추가

- 사진첩 아이콘의 클래스 이름: icon photo
- 사진첩 앱의 클래스 이름: photo-app

```
<div class="homescreen">
  <div class="content">
    ...
    <div class="icon photo">...</div>
  </div>
</div>
<div class="photo-app" style="display: none">
  ...
</div>
```


1. 사진첩 아이콘을 누르면 사진첩이 열리는 기능 추가

- 사진첩 아이콘의 클래스 이름: icon photo
- 사진첩 앱의 클래스 이름: photo-app

```
const photoApp = document.querySelector(".photo-app");  
const photoIcon = document.querySelector(".icon.photo");  
  
photoIcon.addEventListener("click", () => {  
  photoApp.style.display = "block";  
});
```

2. 뒤로 가기 버튼을 누르면 사진첩이 닫히는 기능 추가

- 사진첩 앱의 뒤로가기 버튼의 클래스 이름: back

```
<div class="photo-app" style="display: block">  
  <div class="app-header">  
    <div class="back">...</div>  
  </div>  
  ...  
</div>
```


2. 뒤로 가기 버튼을 누르면 사진첩이 닫히는 기능 추가

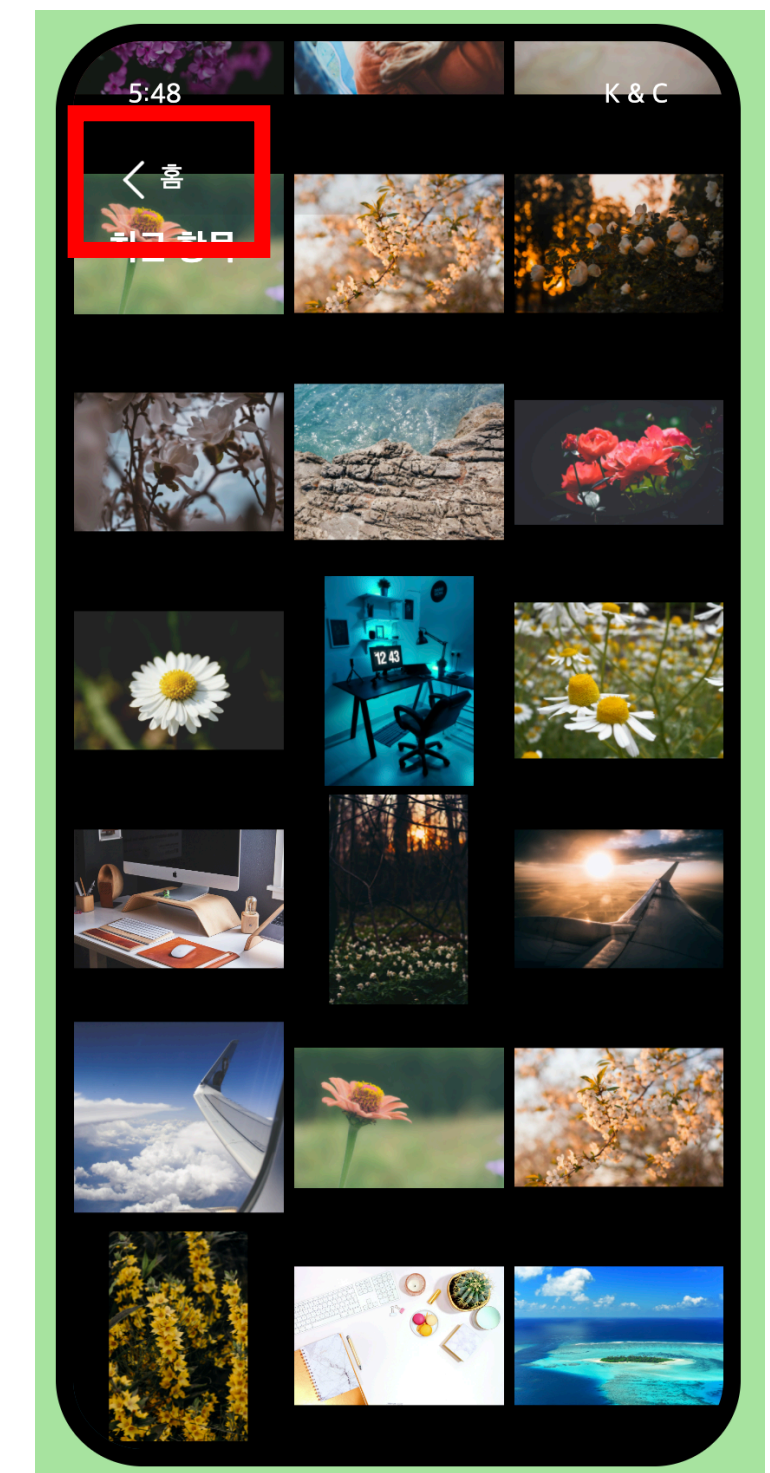
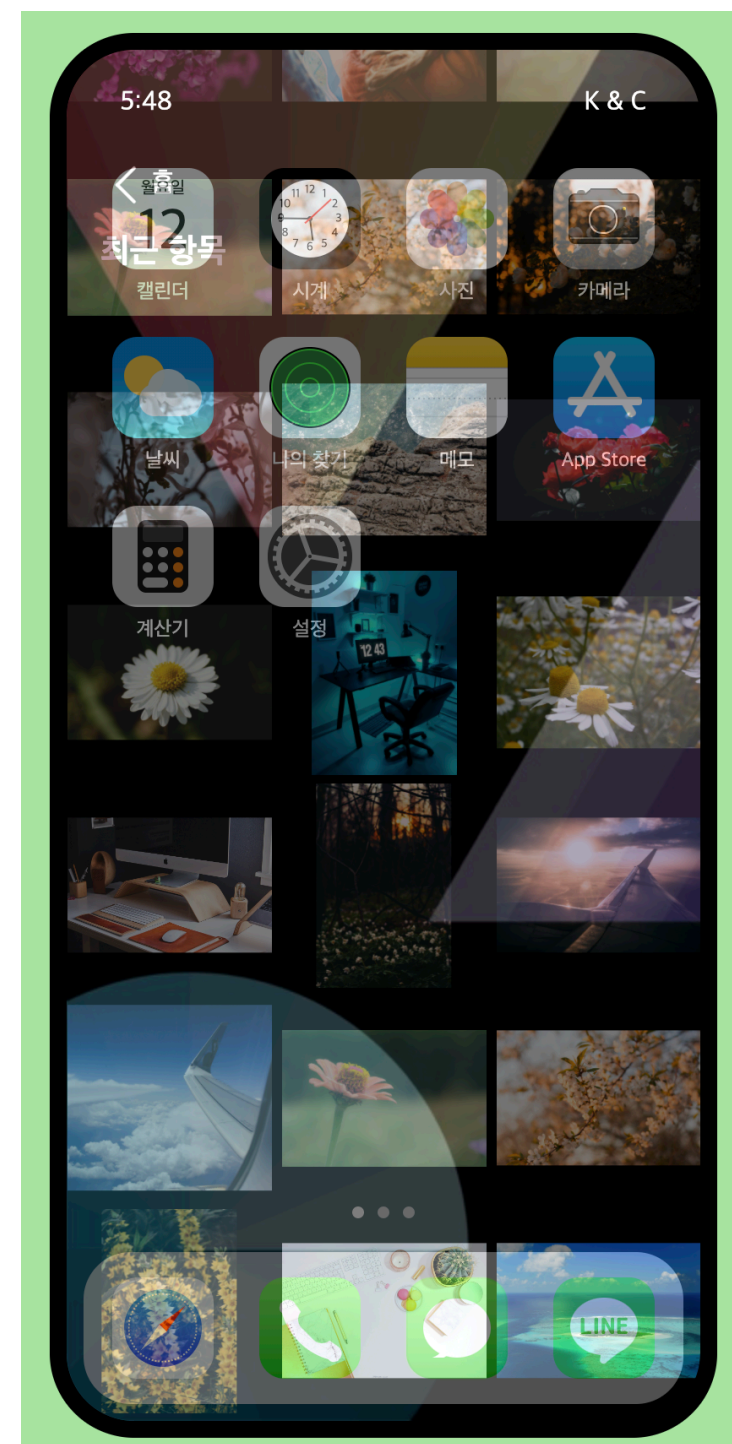
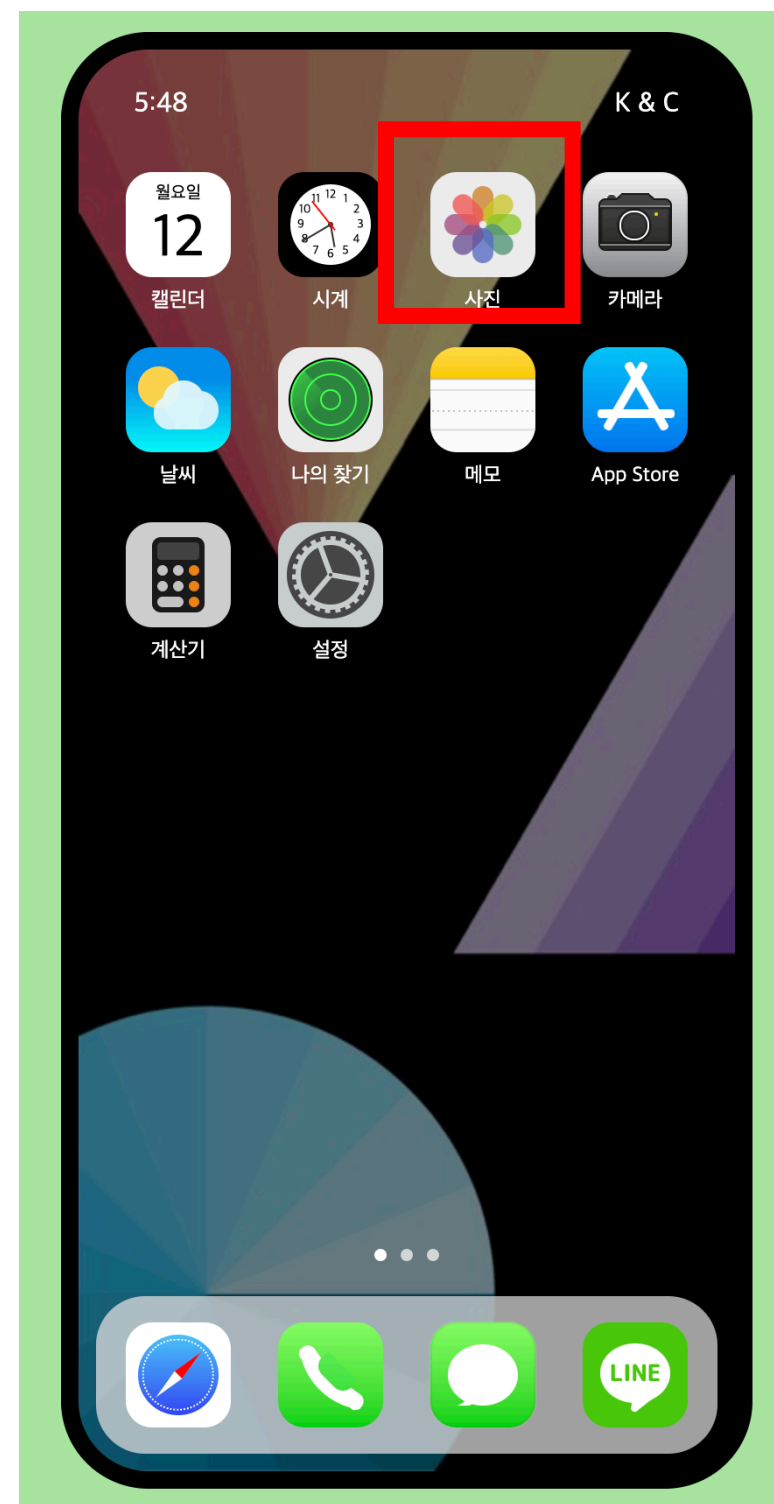
- 사진첩 앱의 뒤로가기 버튼의 클래스 이름: back

```
const backButton = document.querySelector(".back");

backButton.addEventListener("click", () => {
  photoApp.style.display = "none";
});
```

앞으로 만들 기능

3. 사진첩이 열리고 / 닫힐 때 Fade In / Out 효과 적용



3. 사진첩이 열리고 / 닫힐 때 Fade In / Out 효과 적용

- 사진첩이 열릴 때 Fade In

```
.photo-app {  
  ✓ position: absolute;  
  ✓ width: 100%;  
  ✓ height: 100%;  
  ✓ top: 0;  
  ✓ left: 0;  
  ✓ display: none;  
  ✓ background: ▶ ■ #000;  
  ✓ transition: ▶ opacity ease 0.5s;  
  ✓ border-radius: ▶ 40px;  
  ✓ overflow: ▶ hidden;  
}
```


3. 사진첩이 열리고 / 닫힐 때 Fade In / Out 효과 적용

- 사진첩이 열릴 때 Fade In

```
photoIcon.addEventListener("click", () => {  
    photoApp.style.display = "block";  
    photoApp.style.opacity = "0";  
  
    requestAnimationFrame(() => {  
        requestAnimationFrame(() => {  
            photoApp.style.opacity = "1";  
        });  
    });  
});
```

3. 사진첩이 열리고 / 닫힐 때 Fade In / Out 효과 적용

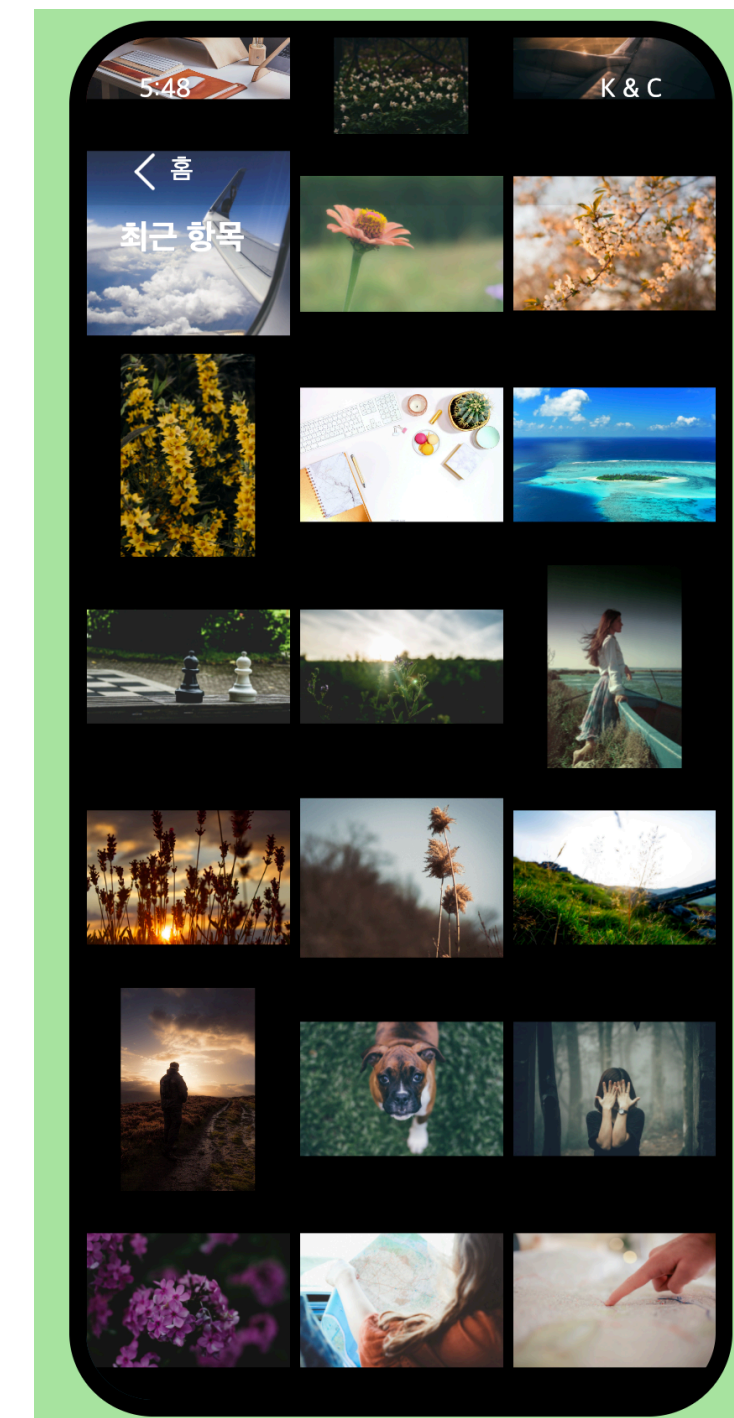
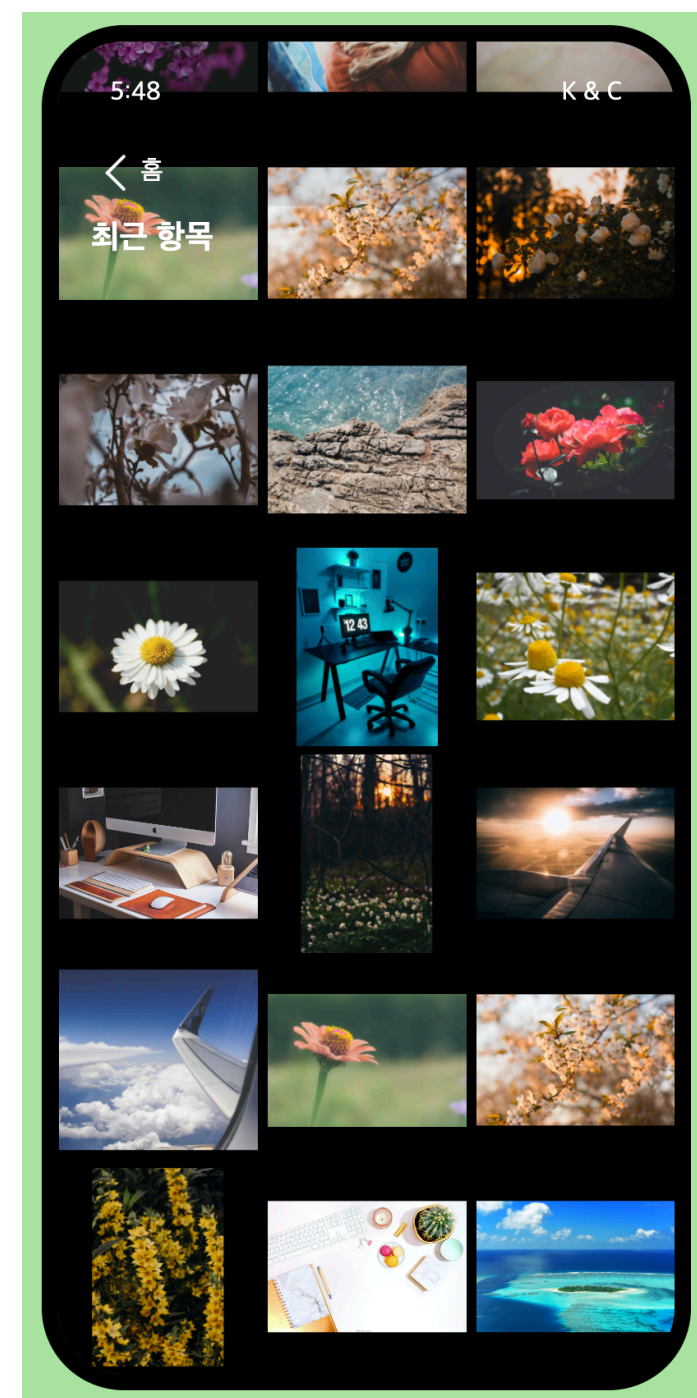
- 사진첩이 닫힐 때 Fade Out

3. 사진첩이 열리고 / 닫힐 때 Fade In / Out 효과 적용

- 사진첩이 닫힐 때 Fade Out

```
backButton.addEventListener("click", () => {  
  photoApp.style.opacity = "0";  
  photoApp.addEventListener("transitionend", () => {  
    photoApp.style.display = "none";  
  }, {  
    once: true,  
  });  
});
```

4. 사진첩은 최근 이미지부터 오래된 이미지 순서대로 아래에서 위로 무한히 추가



4. 사진첩은 최근 이미지부터 오래된 이미지 순서대로 아래에서 위로 무한히 추가

- 사진첩 앱의 컨테이너 클래스 이름: container
- 사진 주소: /assets/photos/(1 ~ 30).jpg
- 사진 아이템 구조

```
<div class="item">  
    
</div>
```


4. 사진첩은 최근 이미지부터 오래된 이미지 순서대로 아래에서 위로 무한히 추가

InfiniteGrid

<https://github.com/naver/egjs-infinitegrid>

```
$ npm i @egjs/infinitegrid
```

4. 사진첩은 최근 이미지부터 오래된 이미지 순서대로 아래에서 위로 무한히 추가

InfiniteGrid 문서에서 아래에서 위로 진행되는 기능 찾기

- REAMDE: <https://github.com/naver/egjs-infinitegrid>
- API: <https://naver.github.io/egjs-infinitegrid/release/latest/doc/>

4. 사진첩은 최근 이미지부터 오래된 이미지 순서대로 아래에서 위로 무한히 추가

prepend 메서드와 prepend 이벤트를 활용

- prepend 메서드: <https://naver.github.io/egjs-infinitegrid/release/latest/doc/eg.InfiniteGrid.html#prepend>
- prepend 이벤트: <https://naver.github.io/egjs-infinitegrid/release/latest/doc/eg.InfiniteGrid.html#event:prepend>

4. 사진첩은 최근 이미지부터 오래된 이미지 순서대로 아래에서 위로 무한히 추가

사진첩 컨테이너에 스크롤이 발생

- isOverflowScroll: <https://naver.github.io/egjs-infinitegrid/release/latest/doc/eg.InfiniteGrid.html>

4. 사진첩은 최근 이미지부터 오래된 이미지 순서대로 아래에서 위로 무한히 추가

- 사진첩이 열릴 때 InfiniteGrid 초기화 및 사진 아이템들 추가

```
import InfiniteGrid, { GridLayout } from "@egjs/infinitegrid";

ig = new InfiniteGrid(".photo-app .container", {
  isOverflowScroll: true,
});
ig.setLayout(GridLayout, {
  margin: 5,
});
ig.prepend(getItems());
```


4. 사진첩은 최근 이미지부터 오래된 이미지 순서대로 아래에서 위로 무한히 추가

- 사진첩이 열릴 때 InfiniteGrid 초기화 및 사진 아이템들 추가

```
function.getItems() {  
  const items = [];  
  for (let i = 1; i <= 30; ++i) {  
    items.push(`<div class="item">  
        
    </div>`);  
  }  
  return items;  
}
```

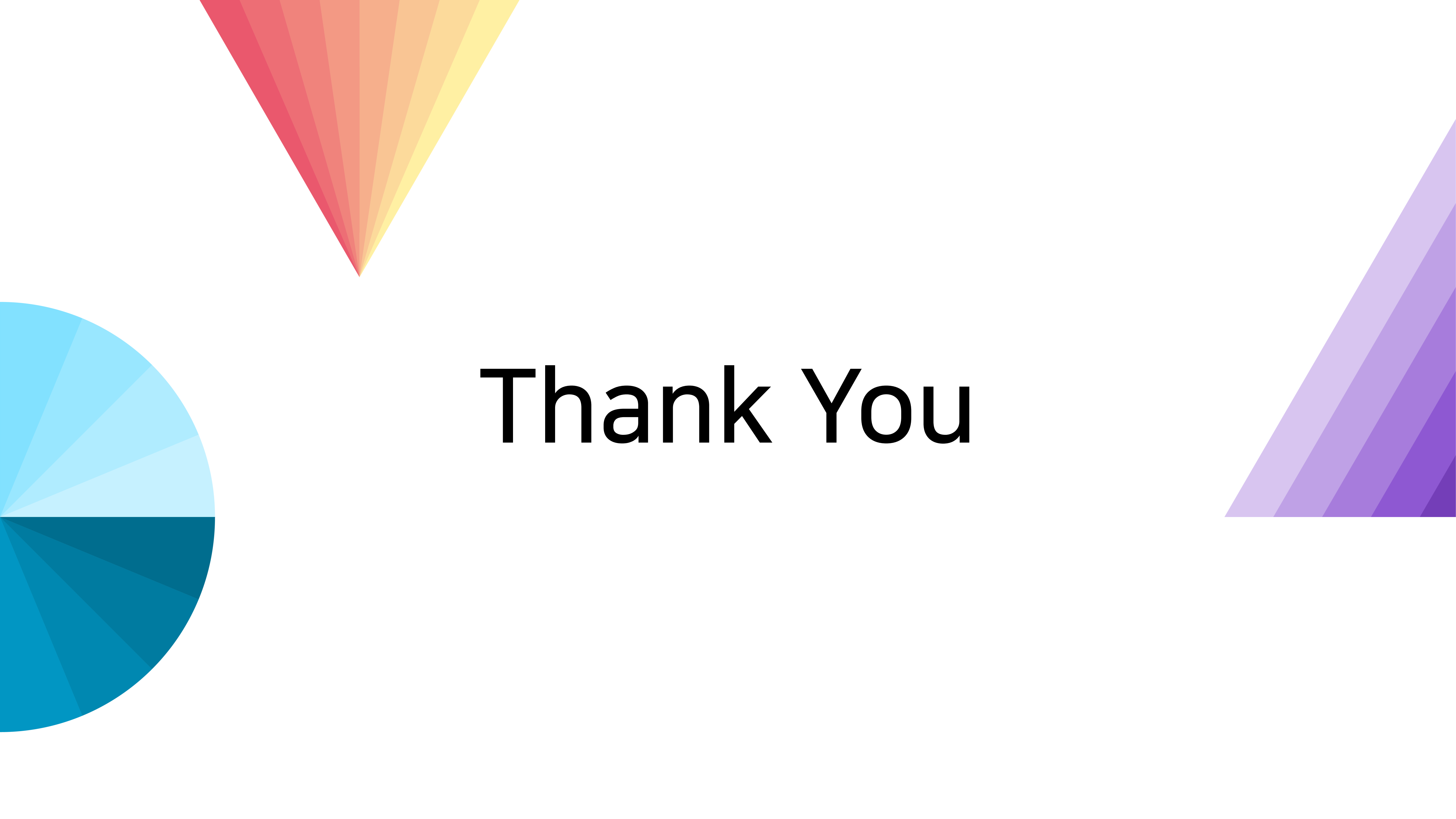
4. 사진첩은 최근 이미지부터 오래된 이미지 순서대로 아래에서 위로 무한히 추가

- 위로 스크롤시 prepend 이벤트가 발생하면서 prepend 메서드를 통해 사진 아이템들 추가

```
ig = new InfiniteGrid(".photo-app .container", {  
  isOverflowScroll: true,  
}).on("prepend", e => {  
  ig.prepend(getItems());  
});
```


마무리

N DEVIEW
2020



Thank You