

CSS Advanced Tutorial

Contents

복습

position

display:flex

NEXT X 멋쟁이사자처럼 7기 최주원



복습: 선택자

HTML의 각 요소를 ‘선택’하기 위해서

00

HTML페이지 안의 특정 요소들을
어떻게 렌더링(Rendering) 할 것인지 브라우저에게 알려준다!

복습: 선택자

전체 선택자(Universal Selector)

HTML 문서 전체를 가리킴

ID 선택자(ID Selector)

속성 값으로 지정된 요소를 가리킴

태그 선택자(Type Selector)

특정 태그 전체를 가리킴

복합 선택자(Combinator)

두 개 이상의 선택자 요소가 모인 것

클래스 선택자(Class Selector)

속성 값으로 지정된 요소를 가리킴

가상 클래스 선택자(Pseudo-Class Selector)

필요에 의해 임의로 가상의 선택자를
지정하여 사용하는 것

복습: link 태그 연결하는법

CSS 파일을 생성하고, 경로 설정을 해주셔야 합니다!

Advanced CSS

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <meta http-equiv="X-UA-Compatible" content="ie=edge" />
    <title>Document</title>
    <link rel="stylesheet" href="./style.css" />
  </head>
  <body></body>
</html>
```

02

전체 선택자

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <meta http-equiv="X-UA-Compatible" content="ie=edge" />
    <title>Document</title>
    <style>
      * {
        background-color: red;
      }
    </style>
  </head>
  <body></body>
</html>
```

태그 선택자

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <meta http-equiv="X-UA-Compatible" content="ie=edge" />
    <title>Document</title>
    <style>
      img {
        width: 100px;
      }
    </style>
  </head>
  <body>
    
  </body>
</html>
```

클래스 선택자

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <meta http-equiv="X-UA-Compatible" content="ie=edge" />
    <title>Document</title>
    <style>
      .cat_img {
        width: 100px;
      }
    </style>
  </head>
  <body>
    
  </body>
</html>
```

ID 선택자

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <meta http-equiv="X-UA-Compatible" content="ie=edge" />
    <title>Document</title>
    <style>
      #cat_img {
        width: 100px;
      }
    </style>
  </head>
  <body>
    
  </body>
</html>
```

복합 선택자

```
<style>
    .selected ul {
        border-color: ■ red;
    }

    .selected2 > ul {
        border: 5px;
    }
</style>
</head>
<body>
    <div class="selected">
        <ul>
            <li>내용</li>
        </ul>
    </div>
</body>
```

가상 클래스 선택자

```
<style>
  a:link {
    color: #gray;
  }
  a:hover {
    color: #red;
  }
  a:visited {
    color: #violet;
  }
</style>
</head>
<body>
  <a href="https://www.naver.com">네이버 바로가기</a>
</body>
```

복습: cascading

CSS는 Cascading Style Sheet

09

캐스캐이딩이란,
CSS를 적용할 때 적용되는 우선순위 규칙을 말한다!

우선순위를 결정하는 요소들

Advanced CSS

| 중요도

| 명시도

| 코드의 순서

우선순위를 결정하는 요소 - 1) 중요도

CSS가 어디에 선언되었는지에 따라 우선순위가 달라진다!

11

1. head 요소 내의 style 요소
2. head 요소 내의 style 요소 내의 import 문
3. <link>로 연결된 CSS 파일
4. <link>로 연결된 CSS 파일 내의 import 문
5. 브라우저 디폴트 스타일시트

우선순위를 결정하는 요소 - 2) 명시도

대상을 명확하게 특정할수록 우선순위가 높아진다!

12

- 1. !important
- 2. inline style
- 3. id selector (아이디 선택자)
- 4. class selector (클래스 선택자)
- 5. tag selector (태그 선택자)
- 6. universal selector (전체 선택자)

우선순위를 결정하는 요소 - 3) 코드의 순서

나중에 선언 된 것이 우선 적용 된다!

위에서부터 파일을 읽기 때문이라고 생각하면 된다!

FAQ : 주의해야하는 상황

명시도(Specificity)는 상속을 이기고 적용되지 않는다.

14

부모에 적용된 스타일이 자식에게도 적용되는 상속의 케이스에는 명시도의 순서와 상관없이
직접적용 >> 상속적용된 스타일

복습용 실습

div 를 이용해서 가로, 세로 150px 인 빨간색 box 만들어보기

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

div는 아무것도 없는거기 때문에 width 와 height를 지정해줘야 한다!

class 선택자는 .으로 표현!
ID 선택자는 #으로 표현!

```
.box {  
background-color: red;  
width: 150px;  
height: 150px;  
}
```

정말 자주 쓰는 속성들 알아보기

background

배경을 나타냄(background-color 등 다양)

width

가로 길이를 나타냄

border

테두리를 나타냄(border-color 등 다양함)

height

높이를 나타냄

color

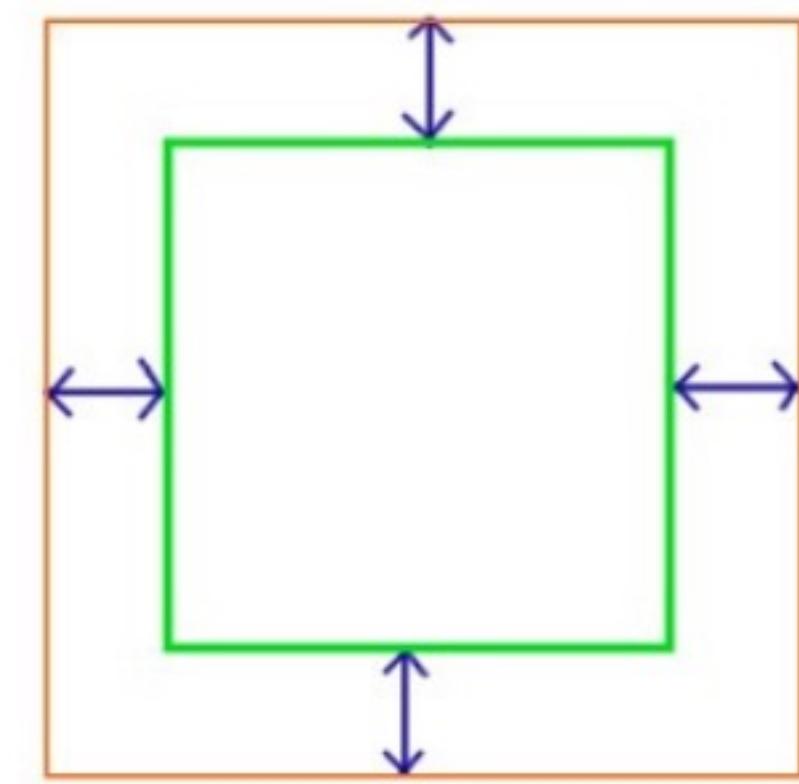
텍스트의 색깔을 나타냄

font-size

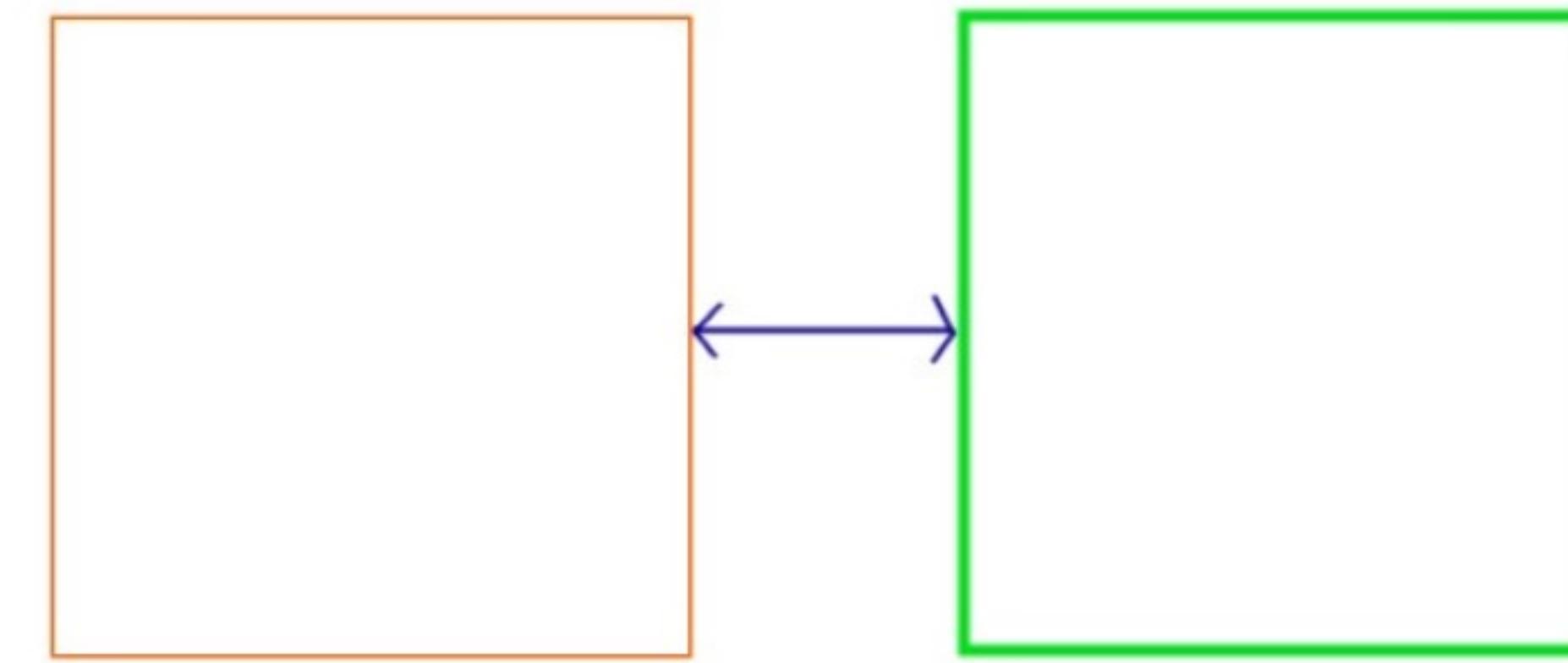
폰트 크기를 조절 할 수 있음

복습: padding과 margin 이란?

실제 개발자도구에서 확인해보자!



Padding



Margin

|| position

CSS를 통해서 위치(레이아웃)를 잡는 방법

position vs flex

flex를 정말 많이 쓰지만...
position 도 알아야 합니다!

position 의 종류

static

relative

absolute

fixed

sticky

position 의 사용법

실제로 그냥 속성일 뿐입니다..!

```
.box {  
    position: relative;  
}
```

```
.box2 {  
    position: absolute;  
}
```

원하는 위치로 이동, position : relative;

static 이었을 때의 위치를 기준으로 이동가능!

24

position : relative; 를 적용하고 나면,
top, right, bottom, left, z-index 등의 속성을 쓸 수 있다.

원하는 위치로 이동, position : relative;

top 을 기준으로! 30px 이동 시킨다는 의미
right, bottom, left 속성도 마찬가지이다!

```
.box {  
    width: 100px;  
    height: 100px;  
    background-color: lightpink;  
    /* relative */  
    position: relative;  
    top: 30px;  
}
```

부모를 기준으로 움직이는, position : absolute;

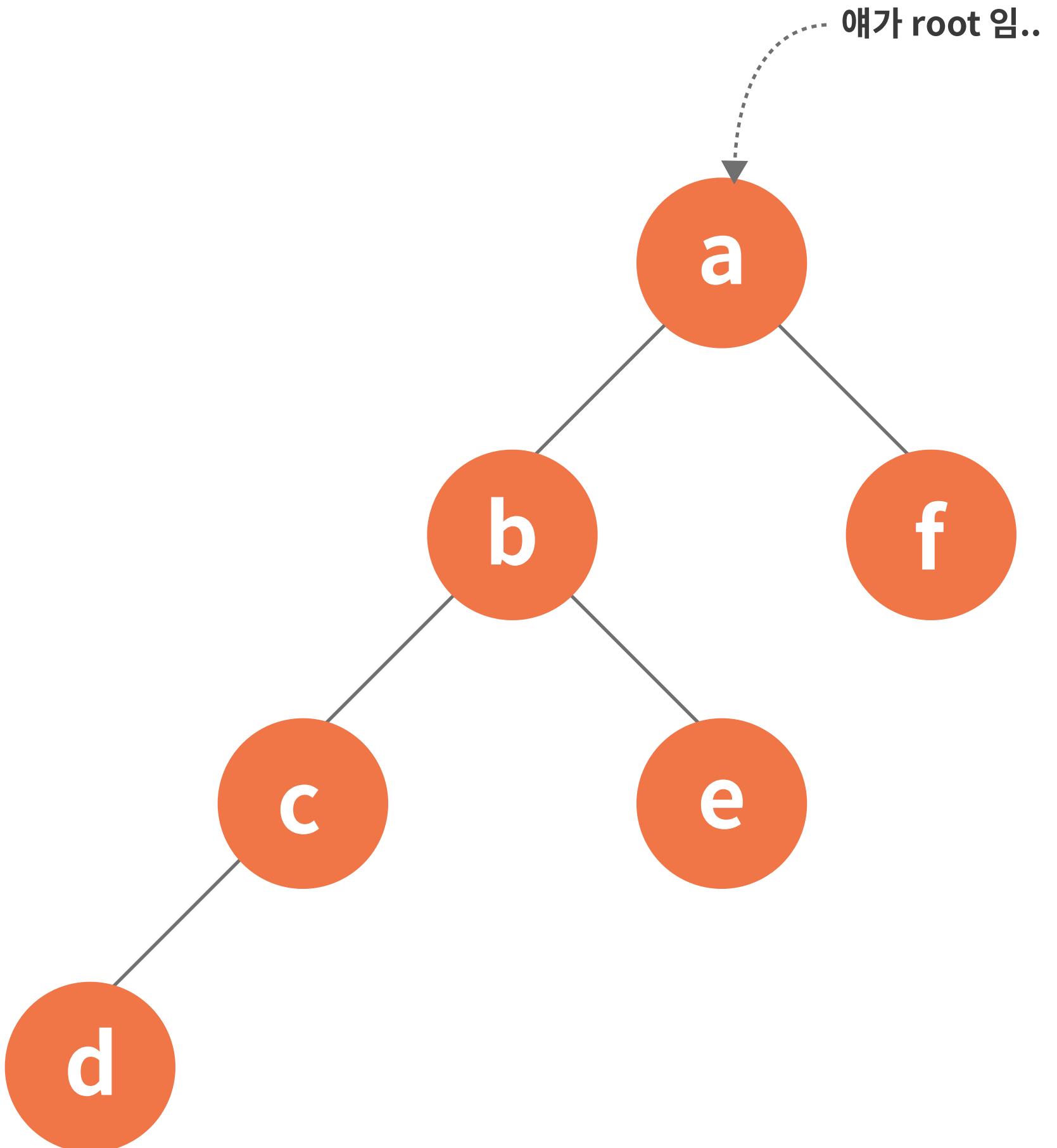
Advanced CSS

position : static을 가지고 있지 않은 부모 기준으로 움직임!

26

근데 부모가 뭘까..?

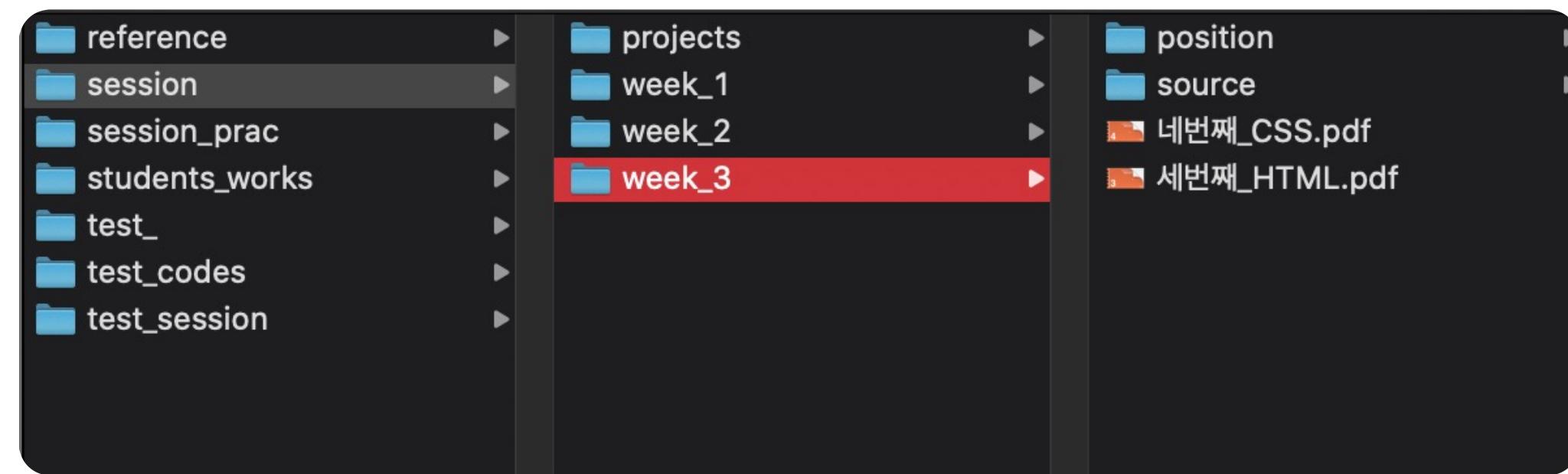
컴퓨터 전반에 있는 tree 구조..



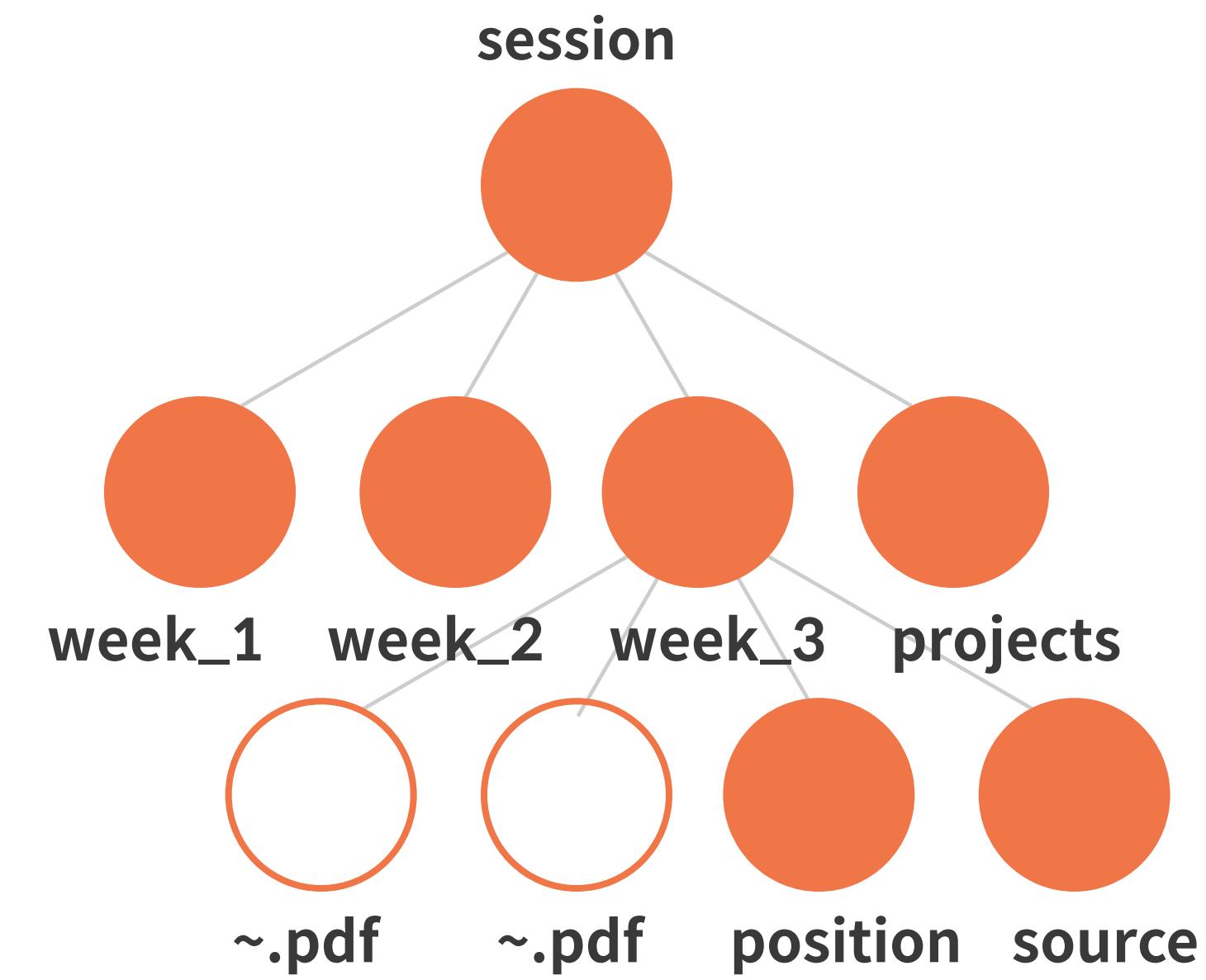
a를 b와 f의 부모 노드,
b와 f를 a의 자식 노드라고 한다!

컴퓨터 전반에 있는 tree 구조..

Advanced CSS



폴더에서도..

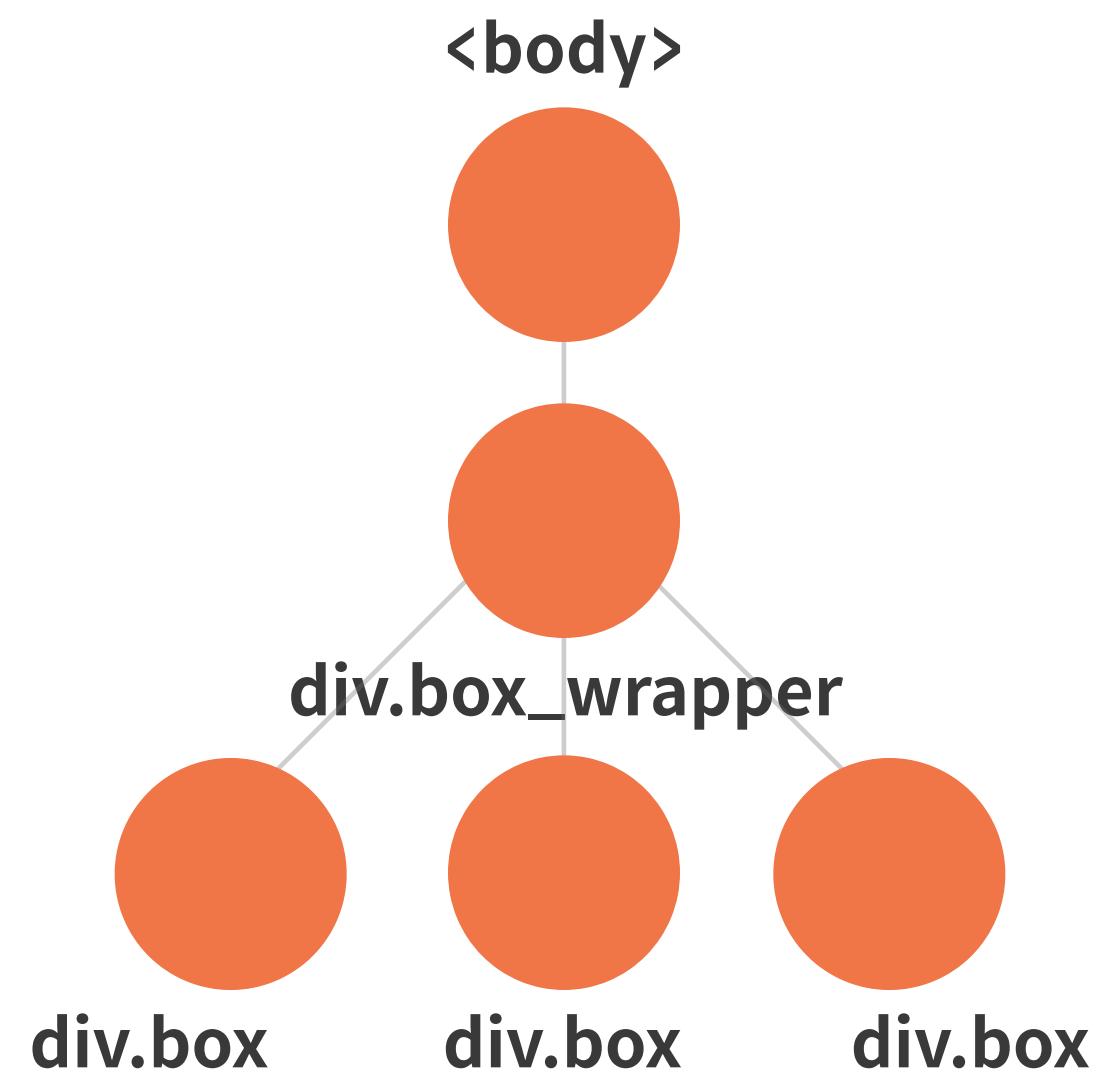


컴퓨터 전반에 있는 tree 구조..

Advanced CSS

HTML에서도..

```
<body>
  <div class="box_wrapper">
    <div class="box">1</div>
    <div class="box">2</div>
    <div class="box">3</div>
  </div>
</body>
```



다시 돌아와서! position : absolute;

| position : static; 이 아닌 부모가 있을 때

부모를 기준으로 움직임

```
<div class="box_wrapper">
  <div class="box">1</div>
</div>
```

30

| position : static; 이 아닌 부모가 없을 때

<body>를 기준으로 움직임

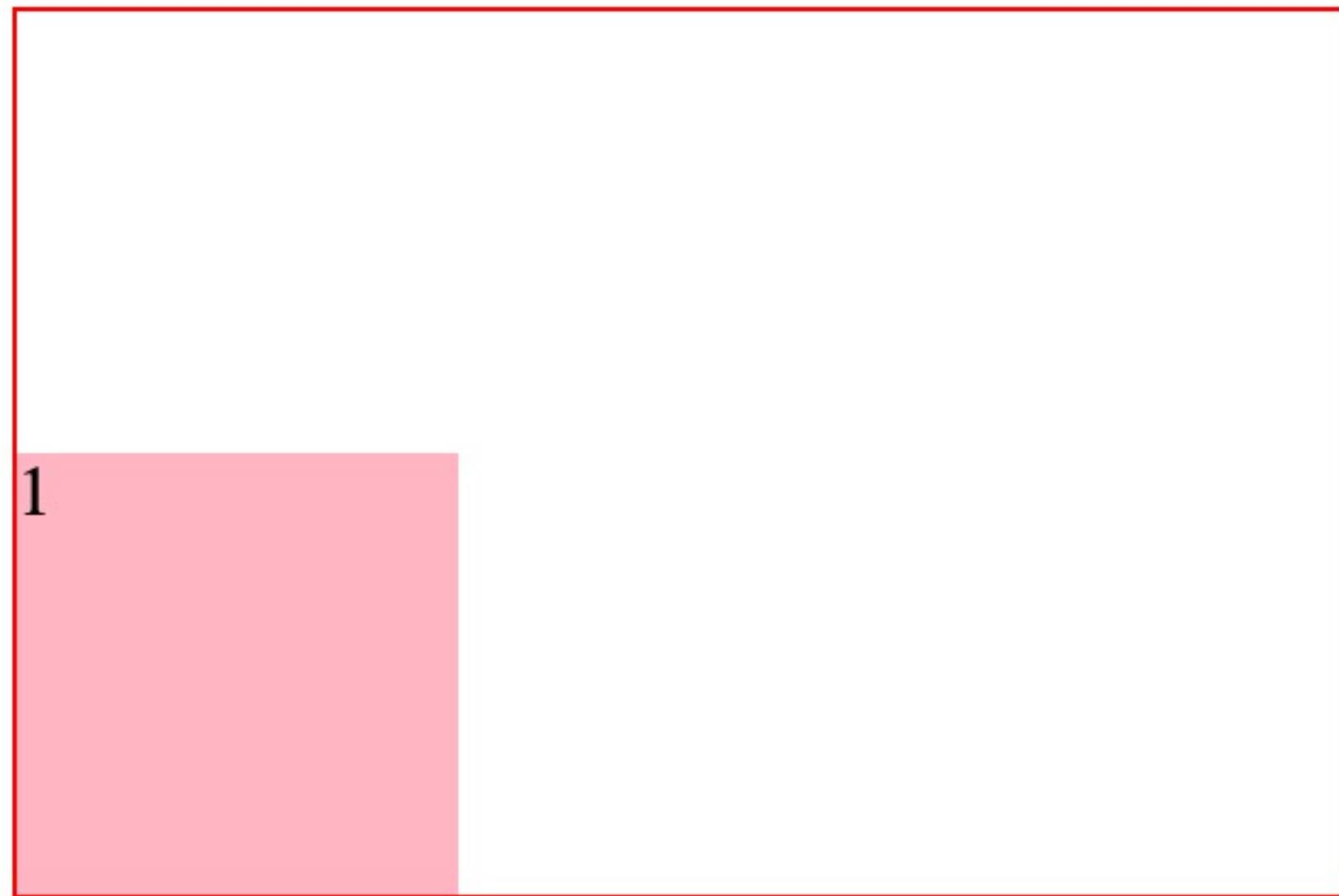
확인해보자 !

position : static이 아닌 부모가 있는 상황

Advanced CSS

```
.box_wrapper {  
    border: 1px solid red;  
    width: 300px;  
    height: 200px;  
    position: relative;  
    top: 100px;  
}  
  
.box {  
    width: 100px;  
    height: 100px;  
    background-color: lightpink;  
    /* relative */  
    position: absolute;  
    top: 100px;  
}
```

부모 안에서 움직인다고 생각하면 됨!



position : static이 아닌 부모가 없는 상황

Advanced CSS

```
.box_wrapper {  
    border: 1px solid red;  
    width: 300px;  
    height: 200px;  
    /* position: relative;  
    top: 100px; */  
}  
  
.box {  
    width: 100px;  
    height: 100px;  
    background-color: lightpink;  
    /* relative */  
    position: absolute;  
    top: 100px;  
}
```

<body>태그를 기준으로 움직임!



고정시키자! position : fixed;

Advanced CSS

fixed로 위치를 정하면, 스크롤을 해도 그 위치에 고정 된다!

33

항상 그 위치에 고정 가능!

스크롤 하다가 고정시키자! position : sticky;

sticky로 위치를 정하면, 스크롤을 하다 부딪치는 순간, 그 위치에 고정 된다!

34

안되는 브라우저가 아직은 많음..
fixed 와 차이는?

<https://jsfiddle.net/daehwan/xk0od1av/28/>

z-index 란..?

원래는 나중에 나온 태그가 z축 상 ‘위에’ 배치된다!

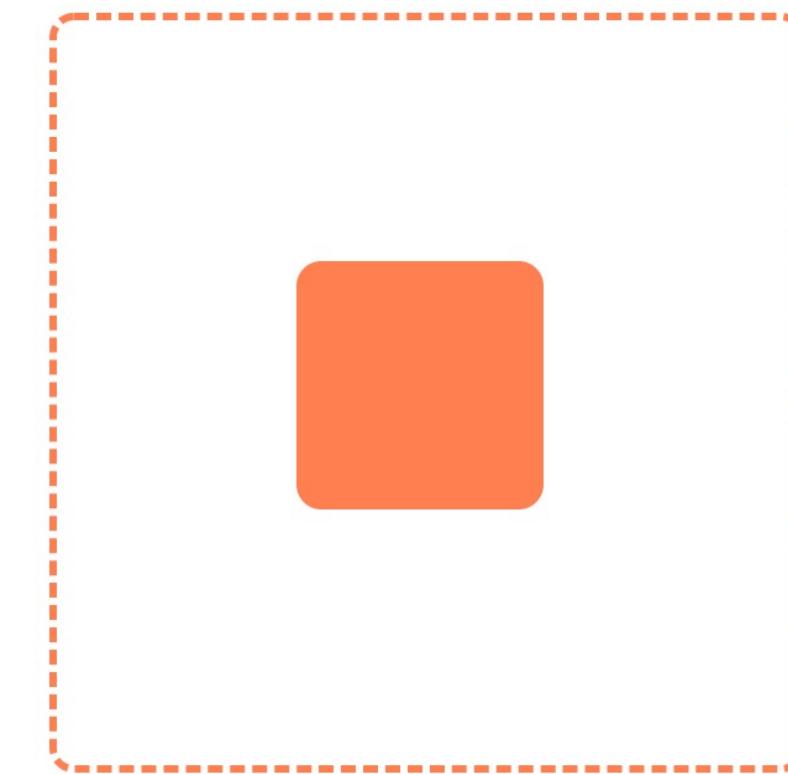
35

position : static이 아니라면,
z-index를 조절 할 수 있다!

실습 1 : prac_1.html

position 기본, static 과 relative 차이 이해하기

- todo_box 를 goal 중앙에 위치하게 옮겨 보기!

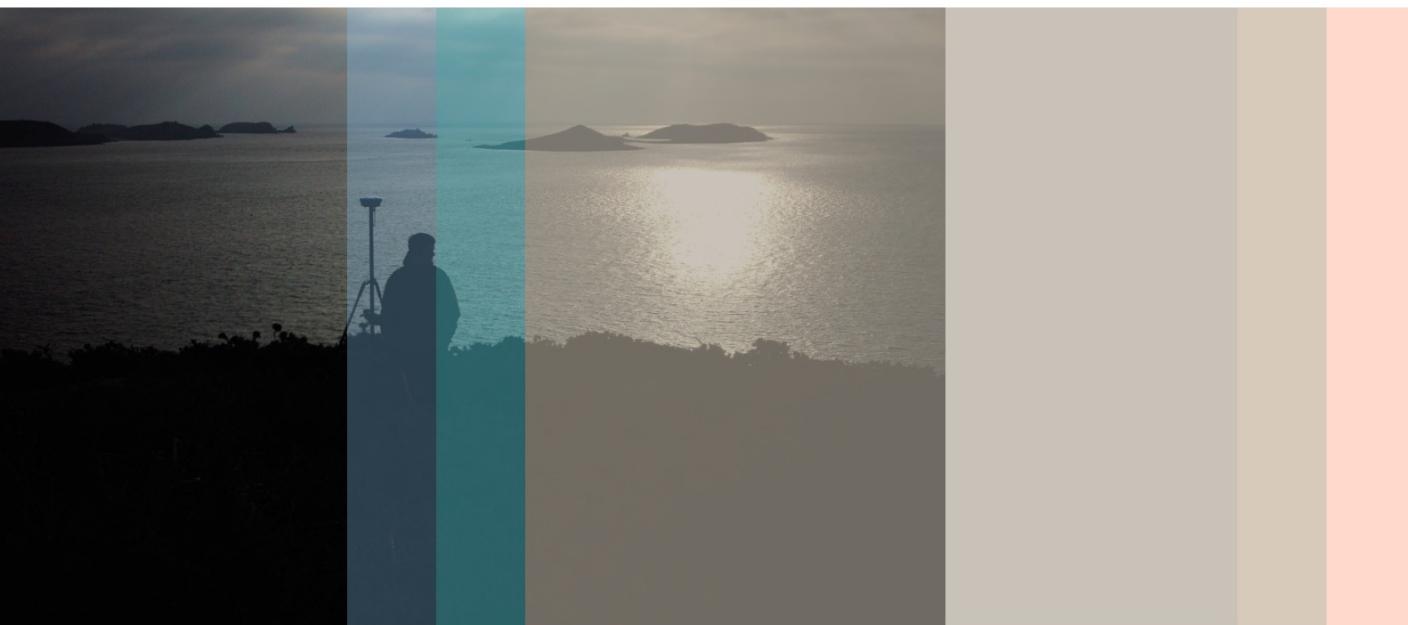


<성공 예시>

실습 2 : prac_2.html

position 과 z-index 이해하기

- 아래의 이미지처럼 만들어보기 !



<성공 예시>

(opacity 라는 속성으로 불투명도를 조절하셔야 합니다. 구글링해보기)

|| display: flex;

display 속성이란 ?

화면에 요소를 어떻게 표시할지를 선택하는 속성!

display 속성의 종류

inline

block

none

inline-block

display : inline ;

inline 은 다음 요소를 줄바꿈하지 않는 것을 말한다!

40

block content 인 요소를 inline으로 바꿀 수 있다!

inline 인 요소들은 width, height 속성을 지정해도 무시된다!(ex)span)
또한, margin과 padding의 상하간격 조절도 안됨.

display : block;

block 은 다음 요소를 줄바꿈 하는 것을 말한다!

41

inline content 인 요소를 block 으로 바꿀 수 있다!

display : none;

none은 요소를 화면에 표시하지 않음

42

~~안보였다가 보이고 싶을때 등등..(feat. 자바스크립트)~~

display: inline-block;

inline과 같지만, width, height 속성 사용 가능!

43

width, height 모두 사용 가능하고,
margin과 padding으로 상하 조절 또한 가능하다!

CSS의 많이 쓰는 단위들..

px

em

rem

%

vh 와 vw

px 이란?

px 은 픽셀(화소) 단위로, 디바이스마다 다르다..!

디바이스의 해상도에 따라서 다르기 때문에,
여러 디바이스 모두를 고려하는 상황에서는 부적절하다.

%란 ?

부모를 기준으로 하는 상대적인 크기를 말한다!

46

부모의 크기가 지정되어 있으면, 그 크기의 상대적인 크기를 가짐.

(ex- 부모 : width: 1000px / 자식 : width : 50% 이면, $1000 * 0.5$)

em 이란 ?

특정 요소에 지정된 사이즈의 상대적인 사이즈를 말함!

47

1em은 같은, 2em은 2배..

rem 이란?

최상위 요소의 사이즈를 기준으로 삼는 상대적인 사이즈!

제일 최상위에 사이즈를 정해두고 rem으로 상대적인 사이즈를 설정할 수 있다!
사용자의 브라우저에 따른 대응이 가능하다는 장점이 있다.
(r은 root를 의미함)

vh와 vw 란?

반응형 웹을 위한 Viewport에 상대적인 사이즈 조절 가능!

49

vw : viewport 너비의 1/100

vh : viewport 높이의 1/100

반응형 CSS 시간에 자세히.. 일단은 알고만 계세요..!

display : flex; 란 ?!

Advanced CSS

50



display : flex; 란 ?!

flex 와 함께라면 요소를 자유자재로 위치 시킬 수 있다!

51

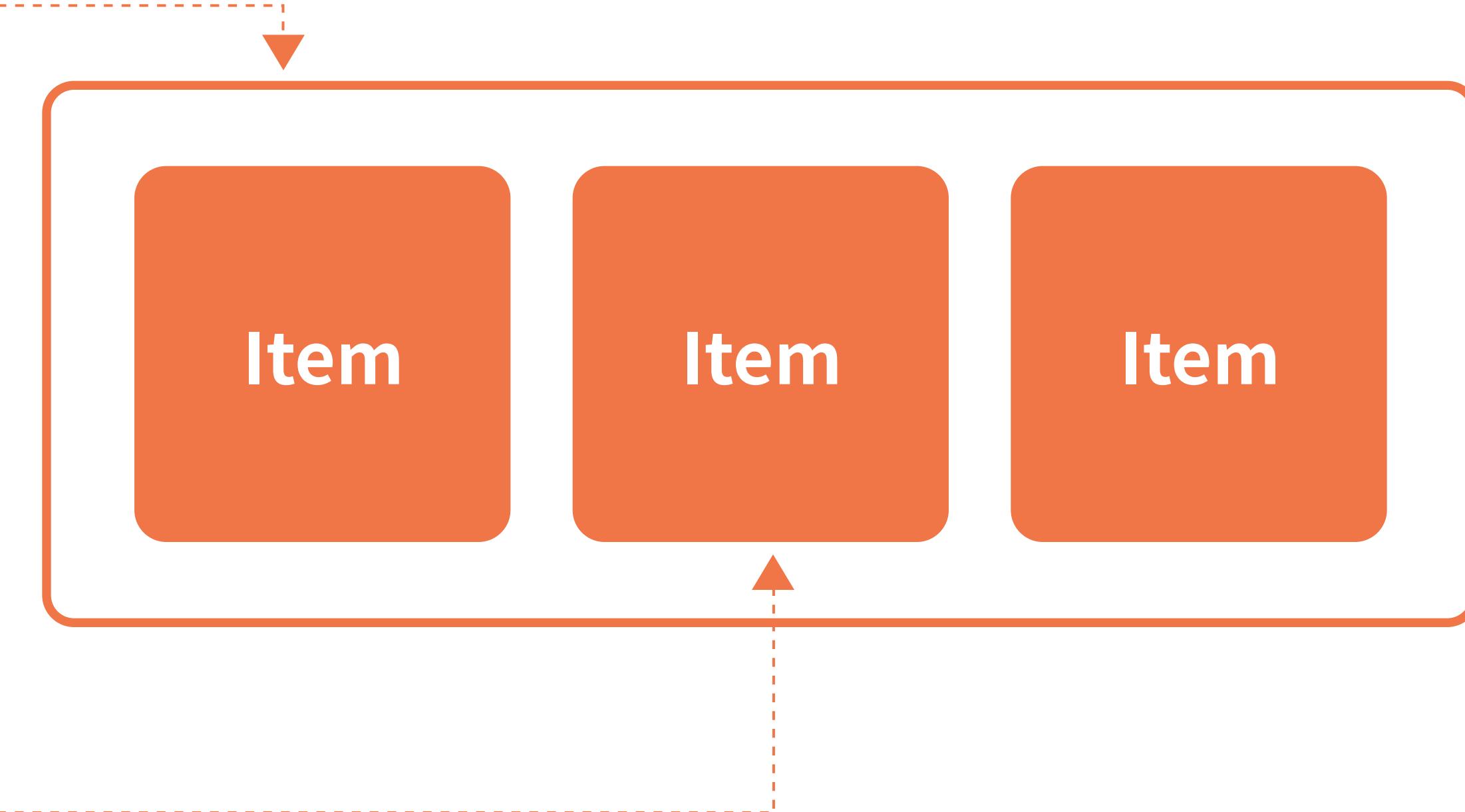
flex로 모든 레이아웃을 다 잡는다고 해도 과언이 아닙니다..!

flexbox는 “Container”와 “Items”로 이루어져 있다!

| Container

Items를 감싸는 “부모” 요소

| Items



Container와 Items에 사용가능한 속성이 다르다

속성을 다 외워야 하는건 절대 아닙니다! 자주 쓰는것(밑줄)만 기억 해주세요

Container

Items를 감싸는 부모 요소 ->

display, flex-direction, flex-wrap, justify-content, align-content, align-items

53

Items

order, flex-grow, align-self

Container 속성 1 : display

display : flex; 로 선언하면 Container가 된다!

54

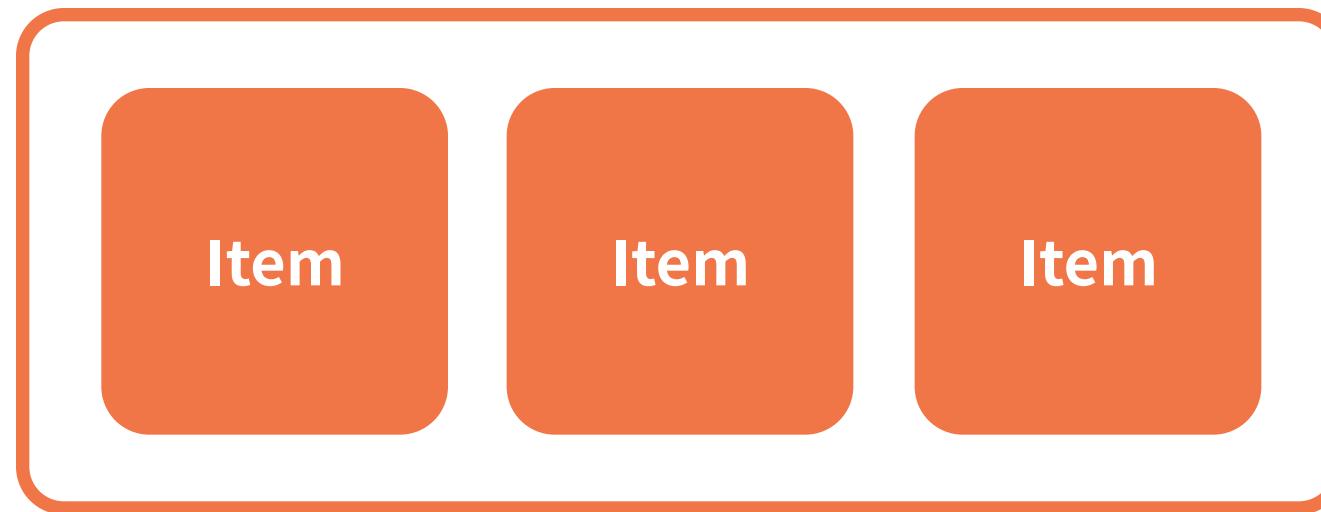
(inline-flex도 있는데 말그대로 inline content 라는 뜻입니다.
그냥 flex는 block 입니다.)

Container 속성 2 : flex-direction

| **items의 주 축(main-axis)를 바꾼다!**

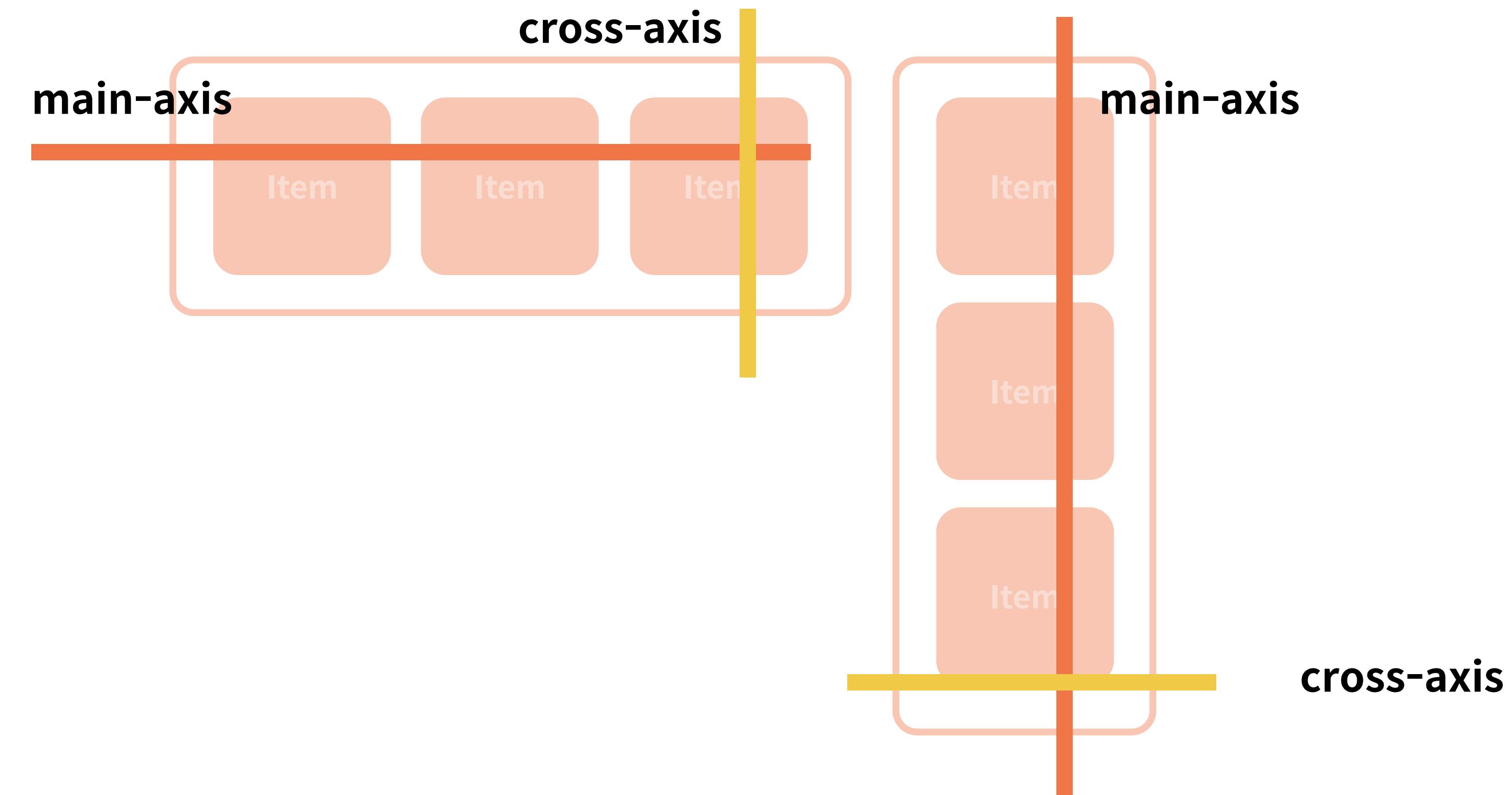
이외에도 column-reverse, row-reverse 도 있습니다.

flex-direction : row ; (default)



flex-direction : column ;

main-axis와 cross-axis 개념 잡기



Container 속성 3 : flex-wrap 이란?

flex-wrap : wrap;으로 여러줄로 묶을 수 있다.

57

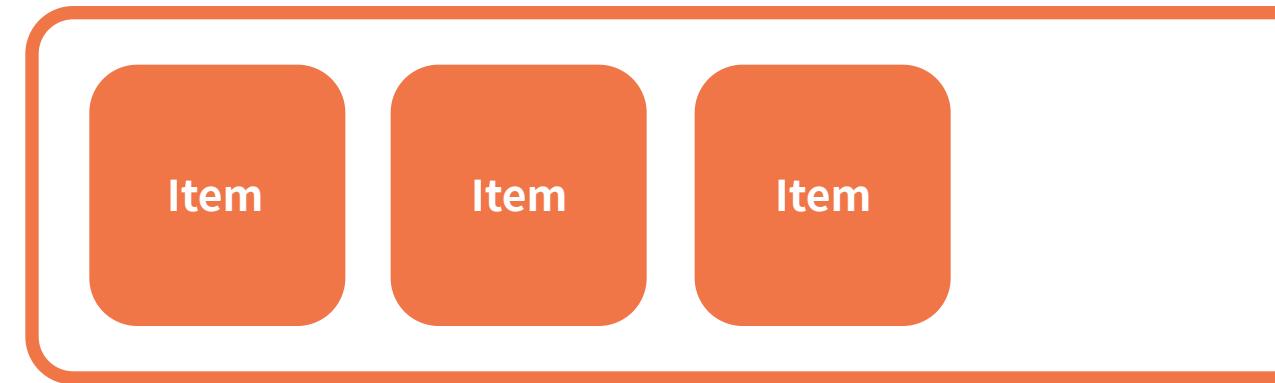
(default 는 nowrap 이다.)

인스타그램 피드를 떠올리면 됨!

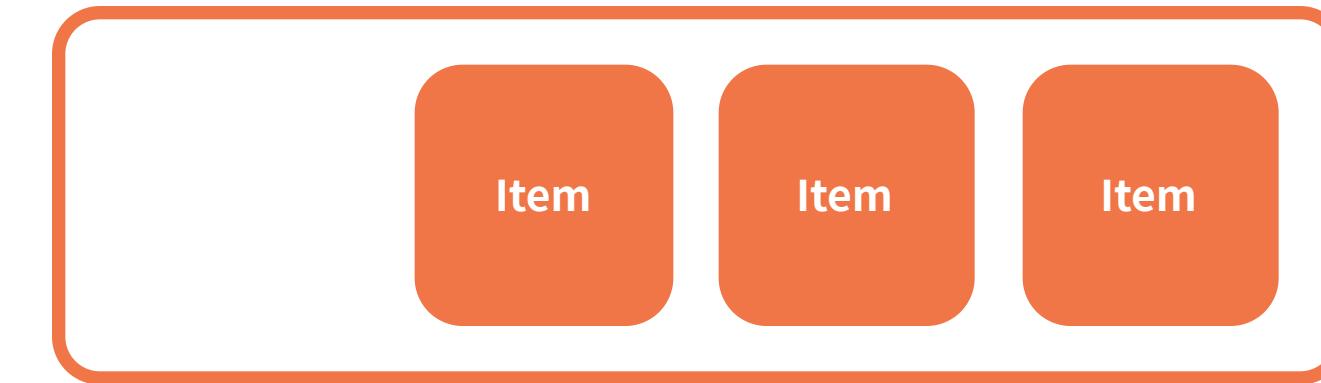
Container 속성 4 : justify-content 란? (중요)

main-axis 의 정렬 방법을 정의 하는 것!

justify-content : flex-start; (default)



justify-content : flex-end;



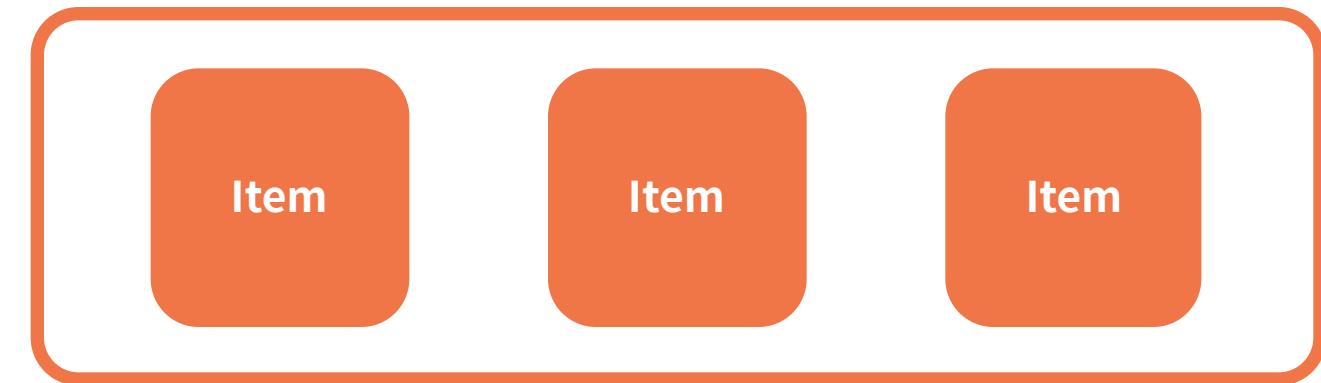
justify-content : center;



justify-content : space-between;



justify-content : space-around;



Container 속성 5 : align-content 란?

flex-wrap : wrap;으로 여러 줄이 묶여있을때만 사용가능

59

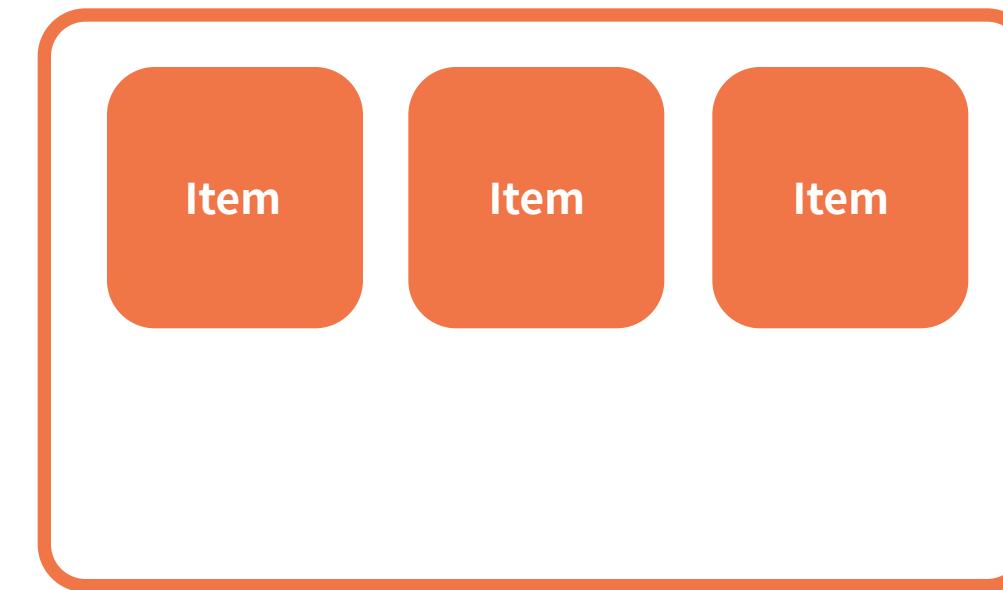
(cross-axis에 대한 정렬 방법이다.)

Container 속성 6 : align-items 란? (중요)

cross-axis 의 정렬 방법을 정의 하는 것!

Advanced CSS

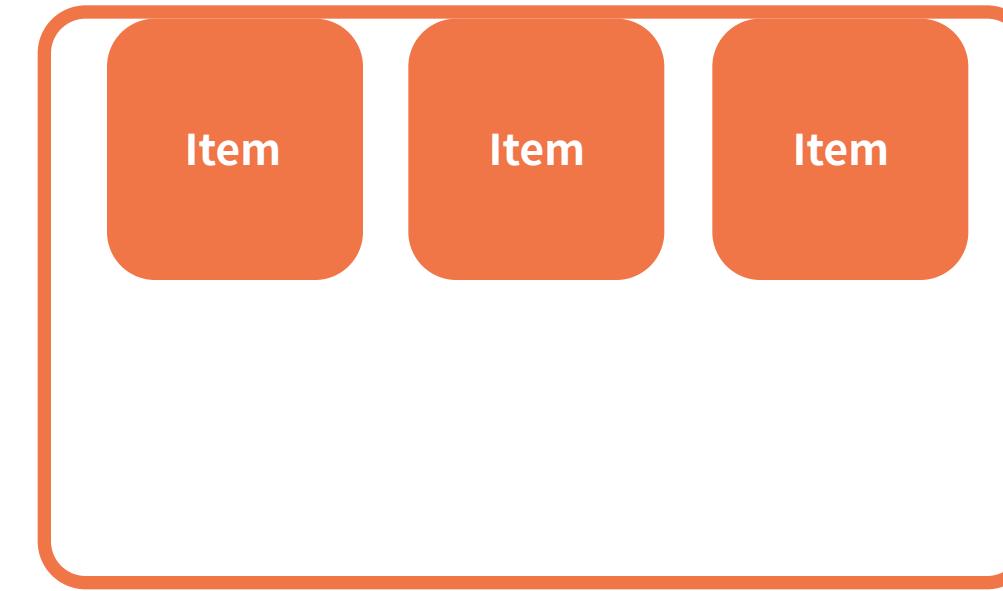
align-items : stretch; (default)



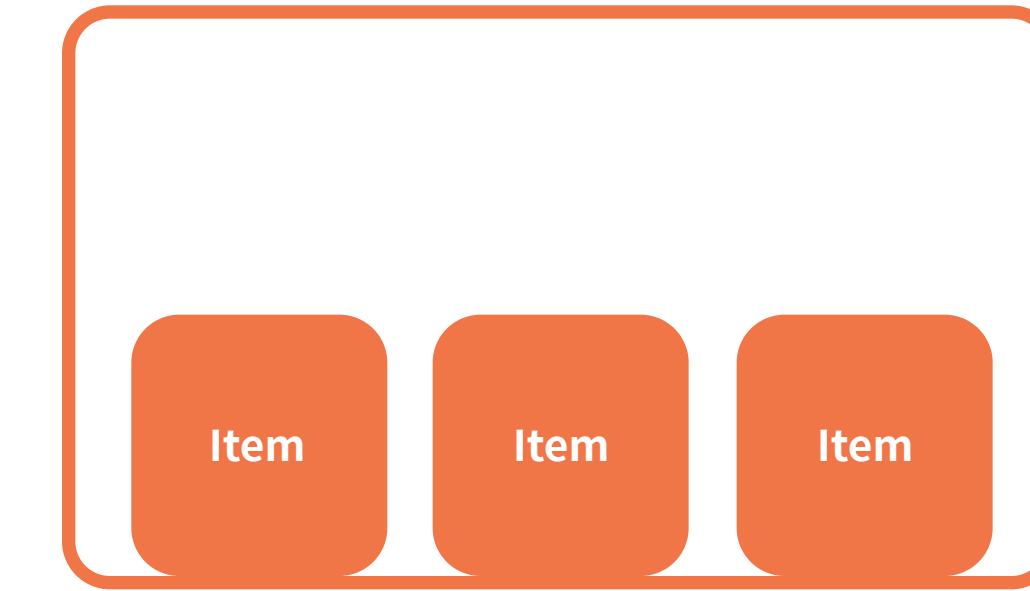
align-items : center;



60



align-items : flex-start;



align-items : flex-end;

그렇다면 flex를 활용한 가운데 정렬은 ?!

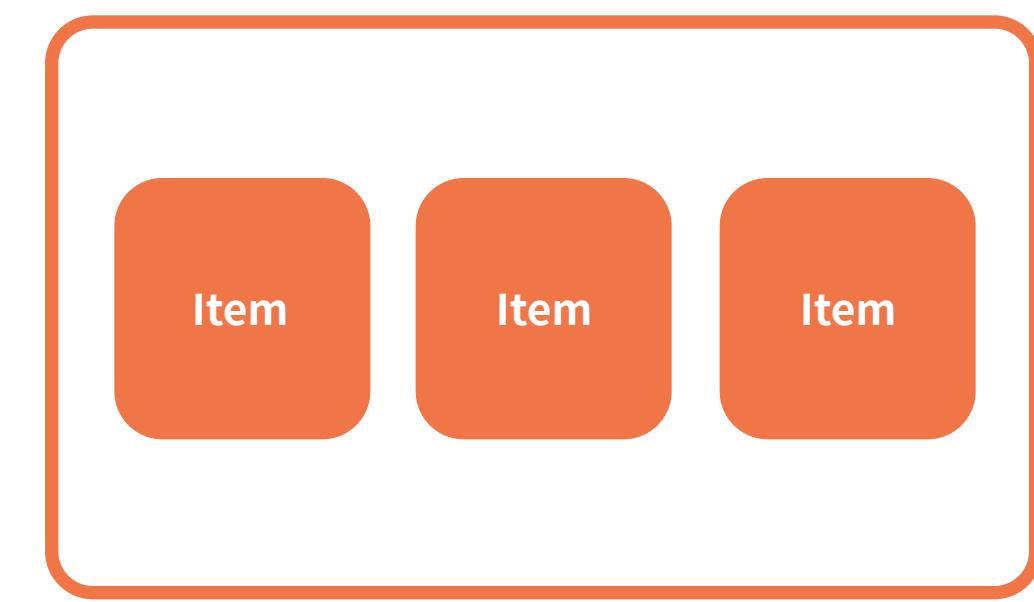
Advanced CSS

```
display: flex;  
justify-content: center;  
align-items: center;
```

justify-content : center;



align-items : center;



61

Items 속성 1 : order

order의 순서로 아이템 나열 순서를 바꿀 수 있다!

62

기본값은 0이고, 숫자를 키울수록 뒤로 밀린다.

Items 속성 1 : flex-grow

item이 차지하는 비율을 설정 할 수 있다!

63

아이템 3 개(a, b, c)의 flex-grow가 각각 (2, 2, 6) 이라면,
(2/10, 2/10, 6/10)을 차지하게 된다!

Items 속성 2 : align-self

cross-axis에서 item 개별 정렬 방법을 바꿀 수 있다!

64

align-items 보다 우선되기 때문에 각각의 item의 정렬을 손댈 때 좋다.

실수 빈번한거..

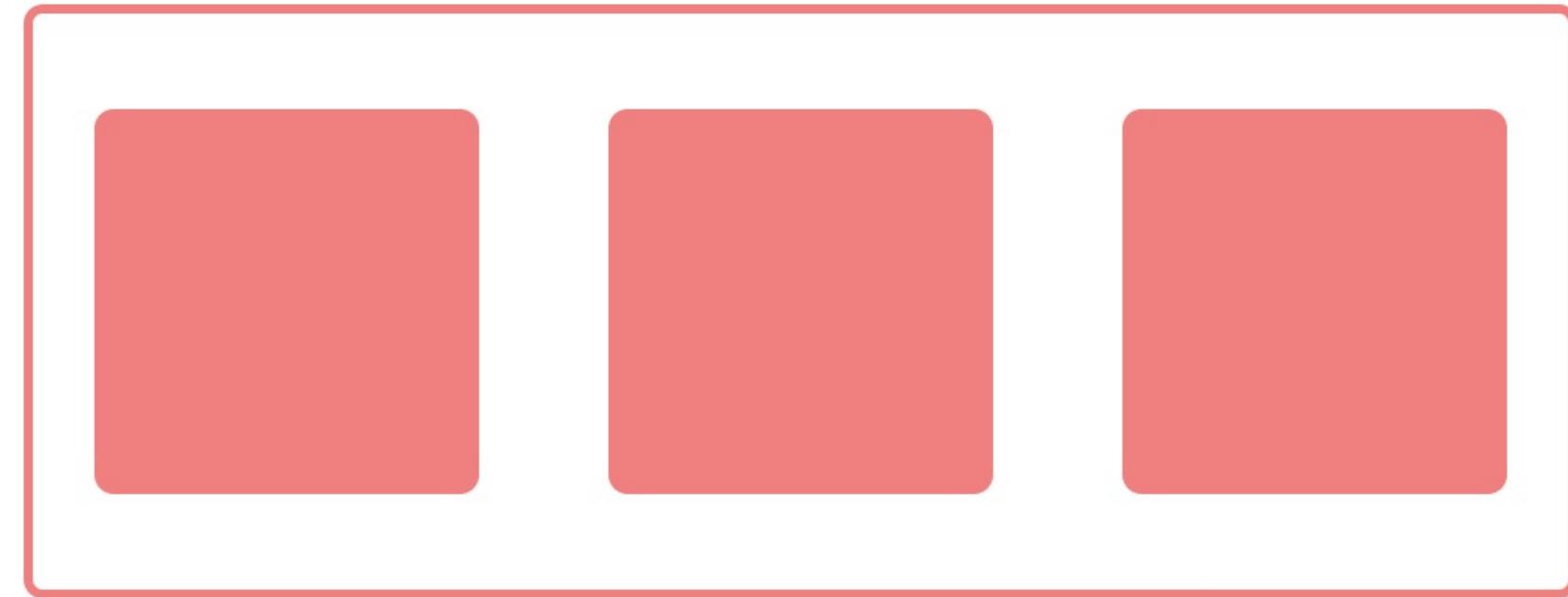
justify-items 는 없었는데.....

나때는

실습 1

■ flex 기본 이해하기

- 아래와 같은 모양 만들어보기

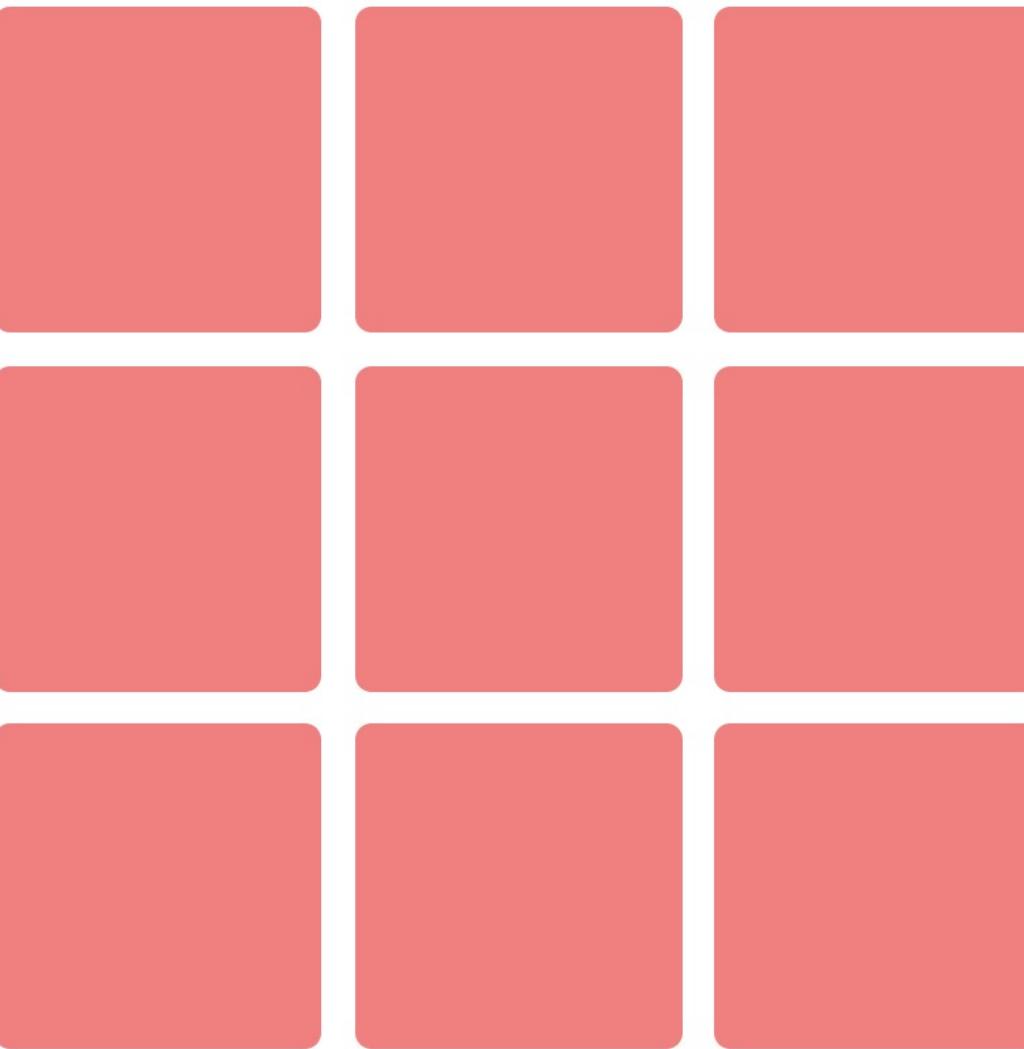


<성공 예시>

실습 2

flex와 기본 html 구조 이해하기

- 오른쪽과 같은 모양을
flex-wrap 없이 만들어보기

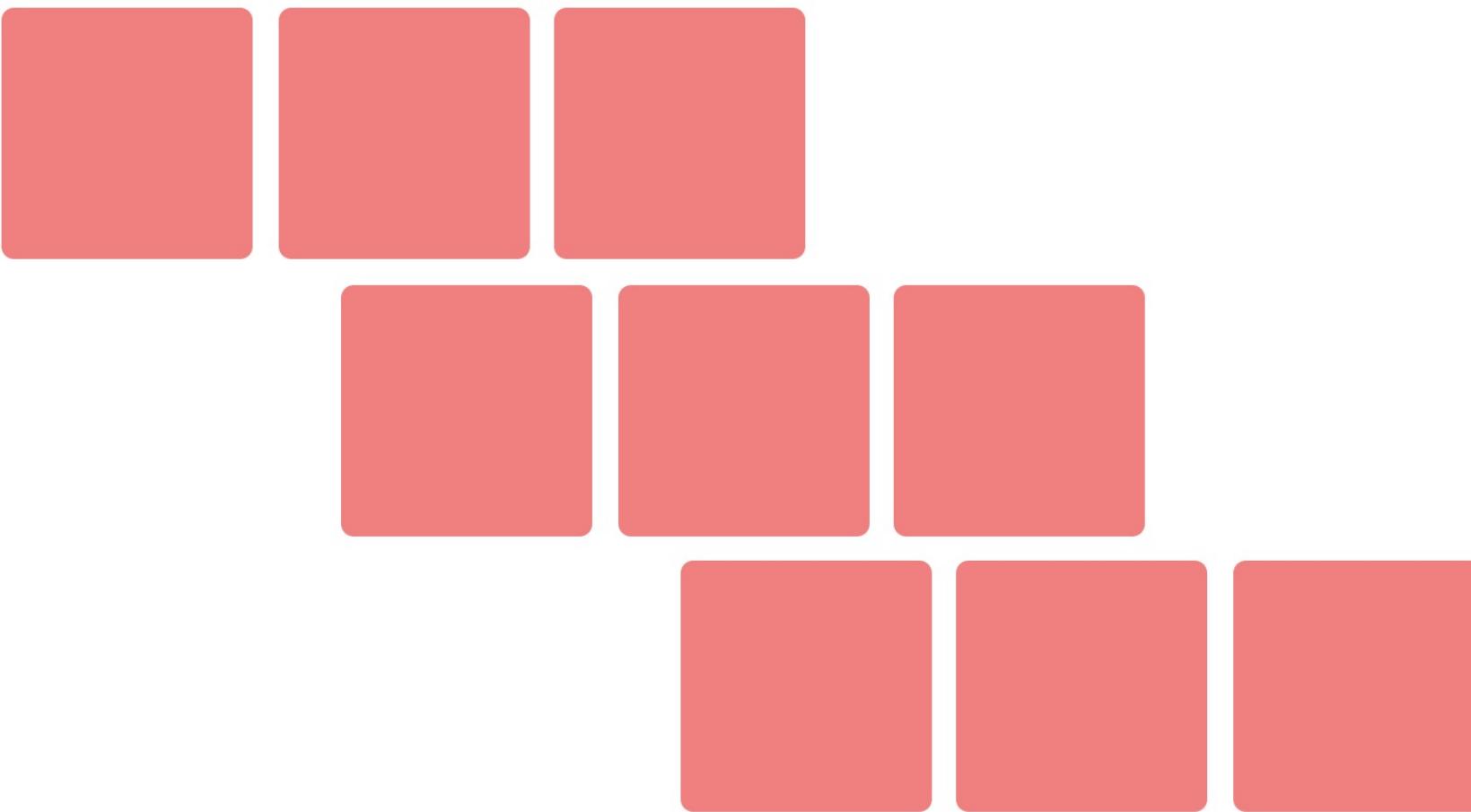


<성공 예시>

실습 3

| **justify-content 이해하기**

- 오른쪽과 같은 모양 만들어보기



<성공 예시>

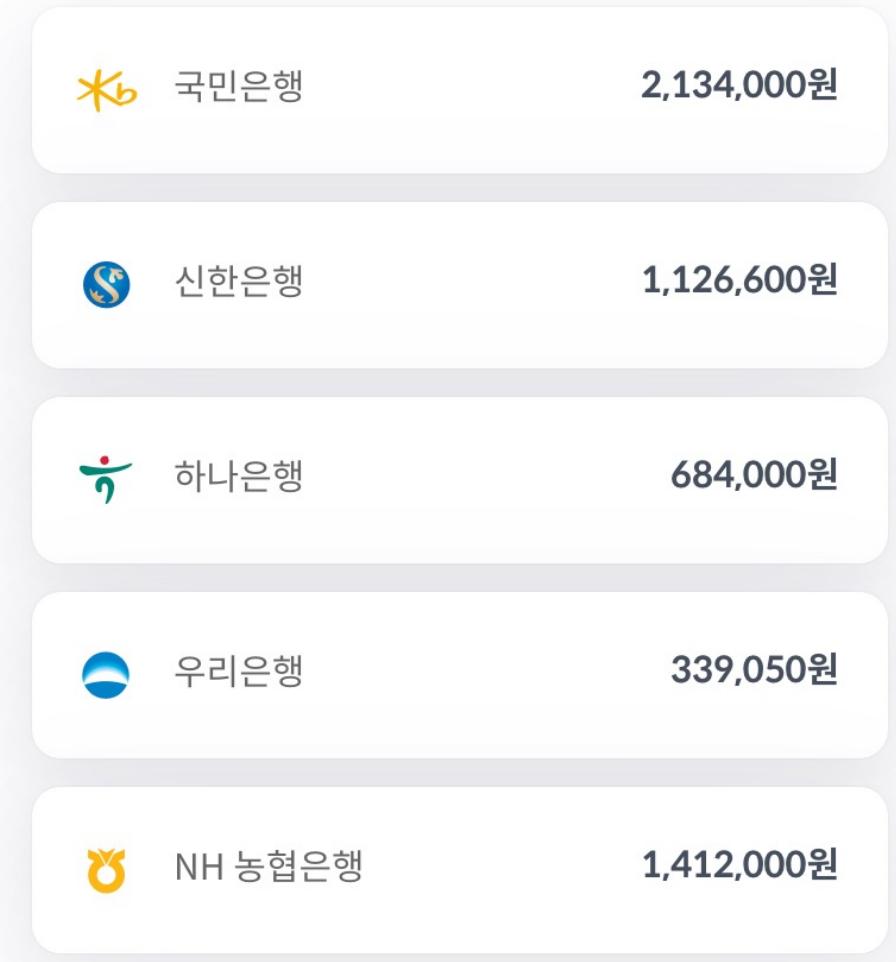
실습 4 : 토스 클론 코딩

Advanced CSS

**계좌 관리는 번거롭지만
토스는 쉽습니다**

간편한 인증 한 번으로
소유한 계좌를 한 번에 관리할 수 있습니다.

[자세히 알아보기 >](#)



Hint

- 카드의 그림자 속성은 -> box-shadow:
0 30px 60px 12px rgba(0, 0, 33, 0.05),
0 4px 24px 0 rgba(0, 0, 33, 0.05),
0 0 1px 0 rgba(0, 0, 33, 0.09) ;
- 로고 이미지는 압축파일에 있음
- 폰트는 SpoqaHanSans (폰트 적용 어떻게..?)

69

부록 : 구글 폰트 사용하기

Advanced CSS

Google Fonts

Noto Sans KR

Designed by Google

Select styles Glyphs About License Pairings

Styles

Type here to preview text

Almost before we knew it, we had left the ground.

Thin 100
Almost before we knew it, we had left the ground.

Light 300
Almost before we knew it, we had left the ground.

Regular 400
Almost before we knew it, we had left the ground.

Medium 500
Almost before we knew it, we had left the ground.

Bold 700
Almost before we knew it, we had left the ground.

Black 900
Almost before we knew it, we had left the ground.

Size: 30px

[Download family](#)

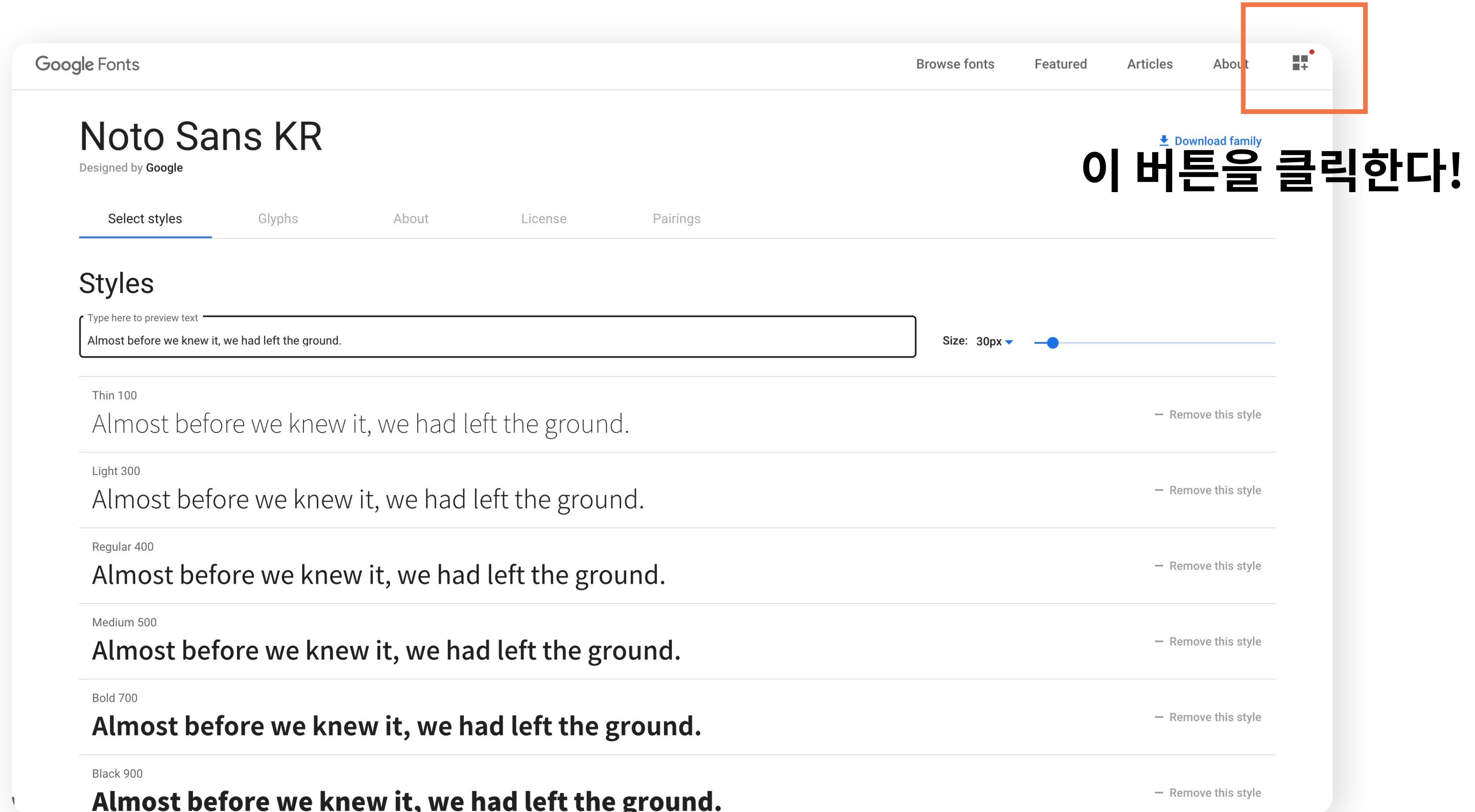
[Select this style](#)

필요한 굵기들을 선택한다!

70

부록 : 구글 폰트 사용하기

Advanced CSS

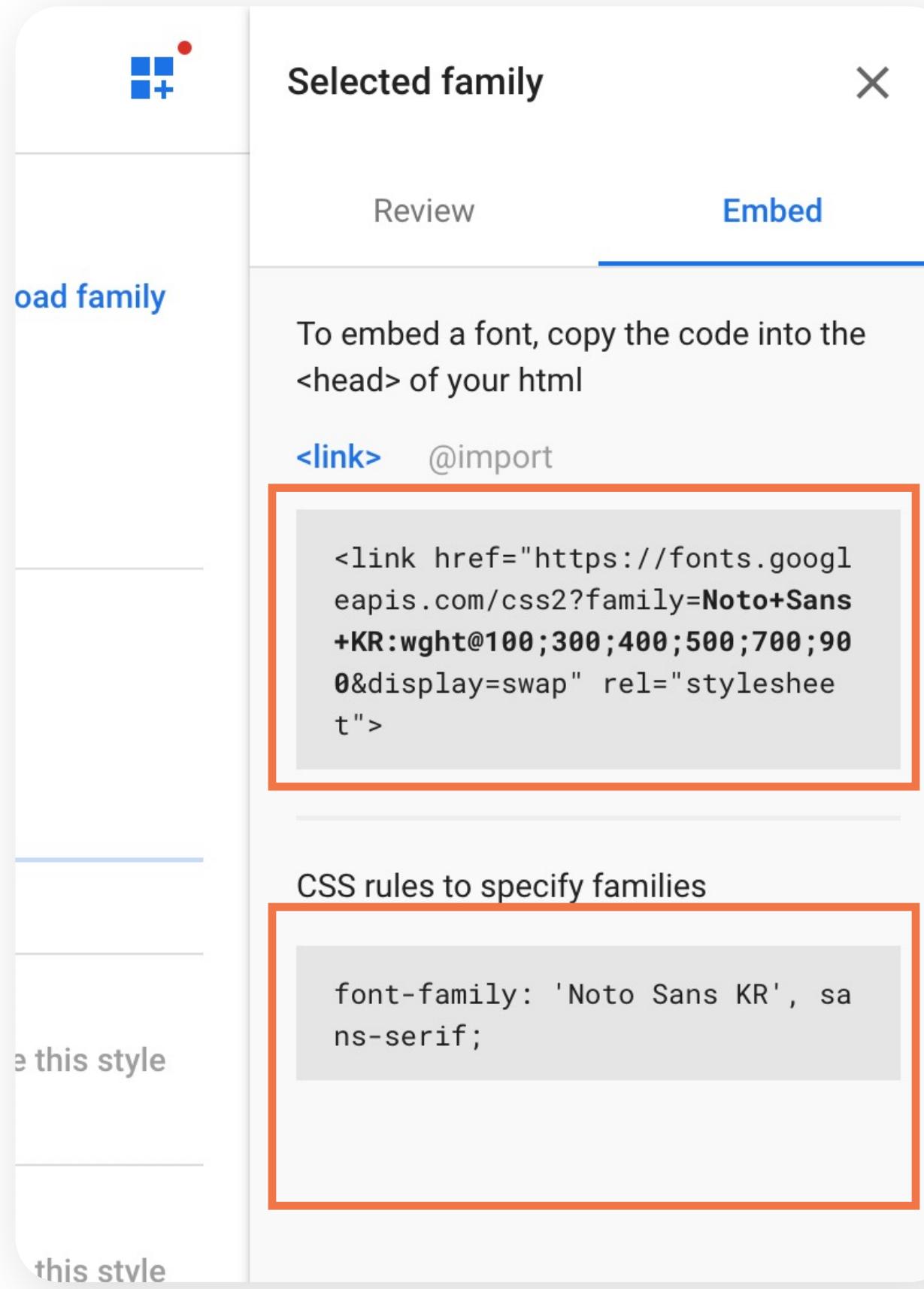


The screenshot shows the Google Fonts interface for the Noto Sans KR font. At the top, there's a navigation bar with 'Google Fonts', 'Browse fonts', 'Featured', 'Articles', 'About', and a user icon. The 'About' button is highlighted with a red box. Below the navigation, the font name 'Noto Sans KR' is displayed, along with the note 'Designed by Google'. There are tabs for 'Select styles', 'Glyphs', 'About', 'License', and 'Pairings', with 'Select styles' being the active tab. A preview area shows the text 'Almost before we knew it, we had left the ground.' in various styles: Thin 100, Light 300, Regular 400, Medium 500, Bold 700, and Black 900. Each preview has a 'Remove this style' link to its right. A size selector at the top right of the preview area is set to '30px'. To the right of the preview area, a large Korean text '이 버튼을 클릭한다!' is overlaid, with the word '버튼' pointing towards the 'About' button.

71

부록 : 구글 폰트 사용하기

Advanced CSS



**CSS를 연결하듯이
link를 head에 넣어주고,**

예시와 같이 font-family 작성 !

근데 SpoqaHanSans 는 구글폰트에 없다..그러면?

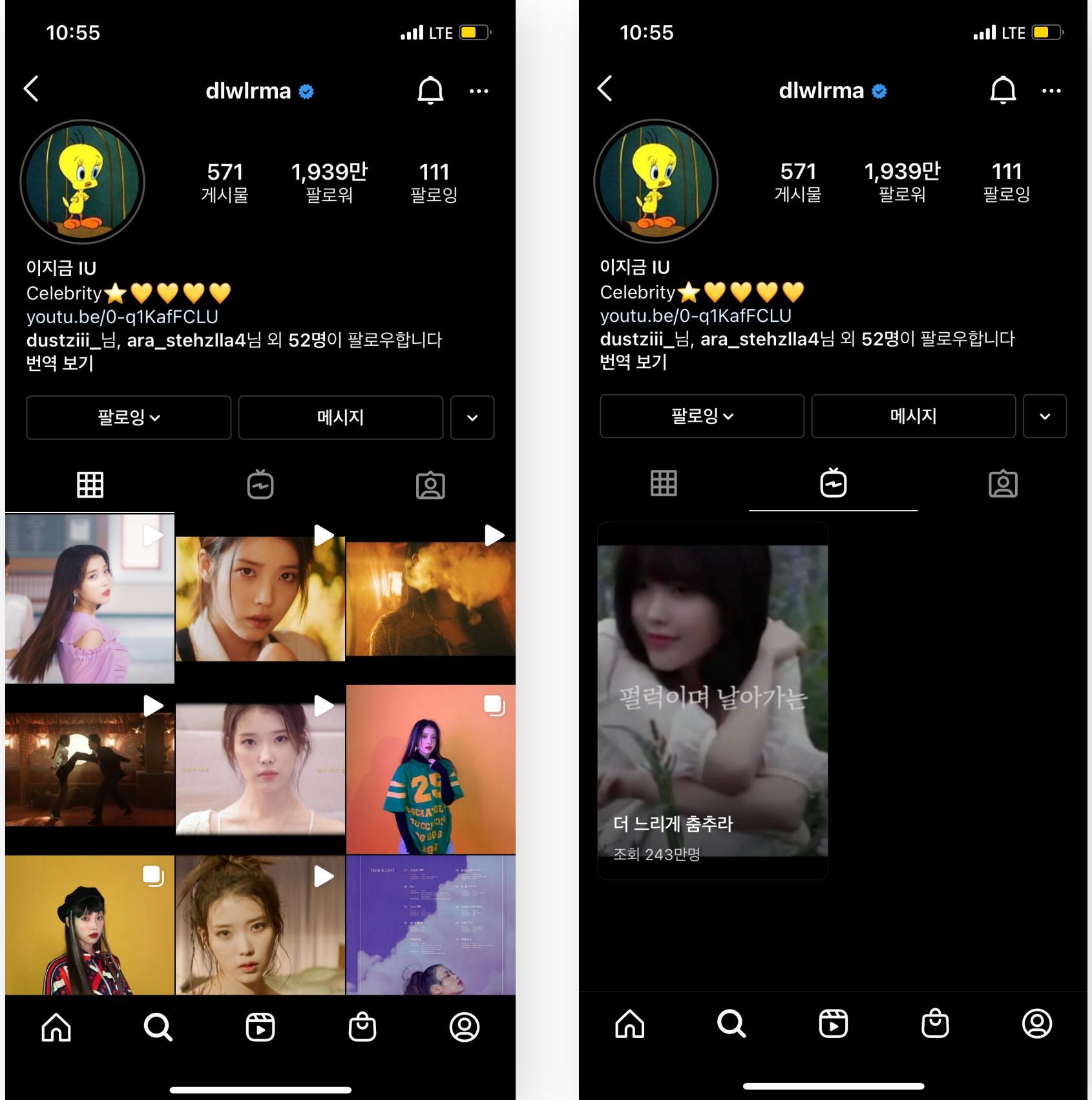
<https://spoqa.github.io/spoqa-han-sans/en-US/#webfont>

73

여기로 들어가서 Web font 로 들어가면 구글폰트 사용법과 똑같이 사용 가능!

과제 : 인스타그램 모바일 클론코딩

Advanced CSS



힌트

1. 아이콘은 Google Material Icons 추천
2. form 을 이용하면 실제로 클릭시 이동하게끔 할 수 있지 않을까?