

Hello,

KDT 웹 개발자 양성 프로젝트

5기!

with





배치1



position과 같이 사용하는 CSS 속성들!
모두 음수를 사용할 수 있어요!

top
bottom
left
right
z-index

요소의 위치 지정 기준

position

static 기준 없음

relative 요소 자신을 기준

absolute 위치 상 부모 요소를 기준

fixed 뷰포트(브라우저)를 기준

sticky 스크롤 영역 기준

위치 상 부모 요소를
꼭 확인해야 해요!



요소의 각 방향별 거리 지정

top, bottom, left, right

auto

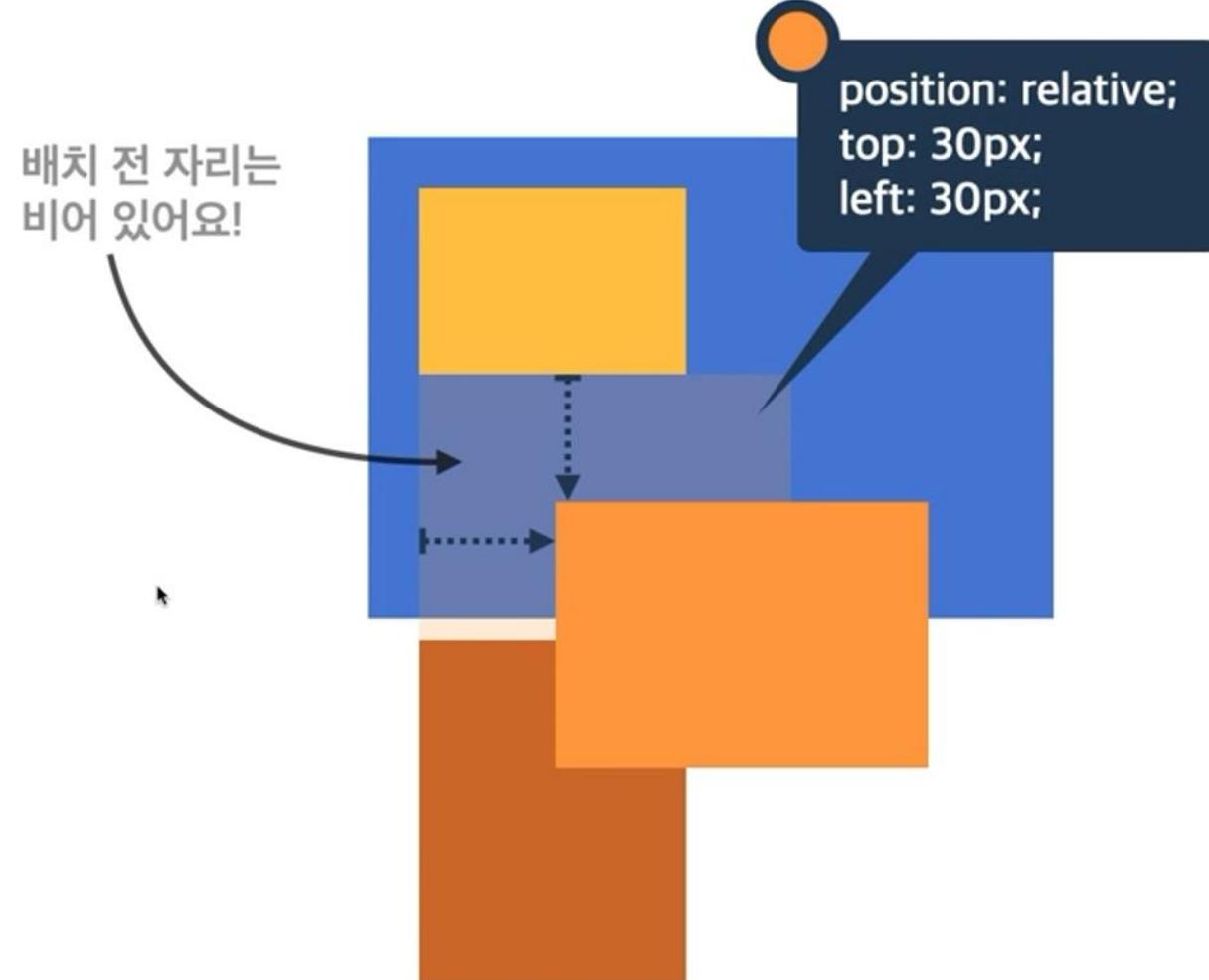
브라우저가 계산

단위

px, em, rem 등 단위로 지정



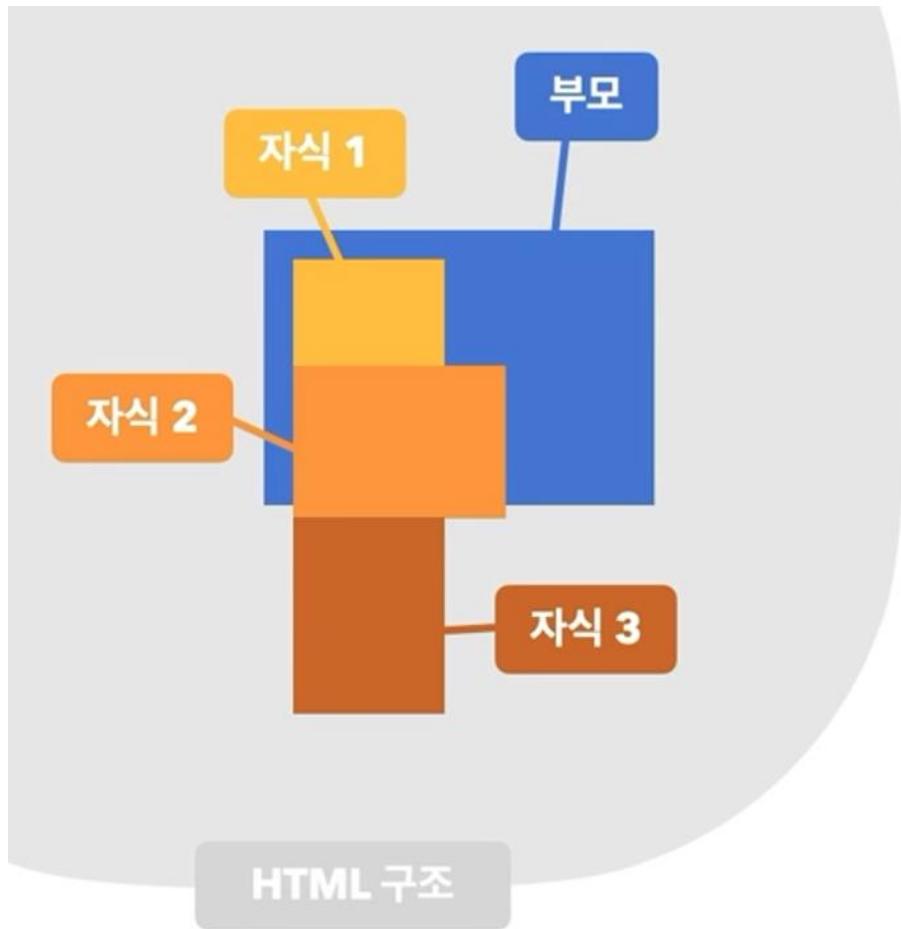
relative



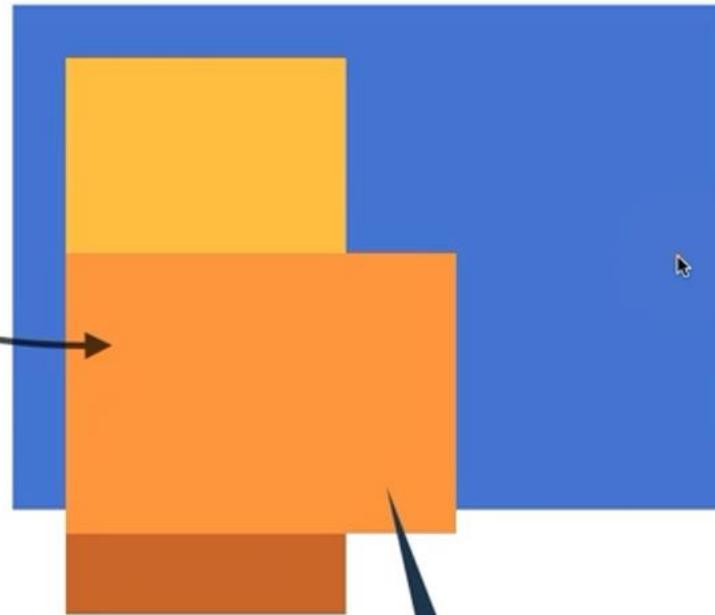
relative 요소 자신을 기준으로 배치!



absolute



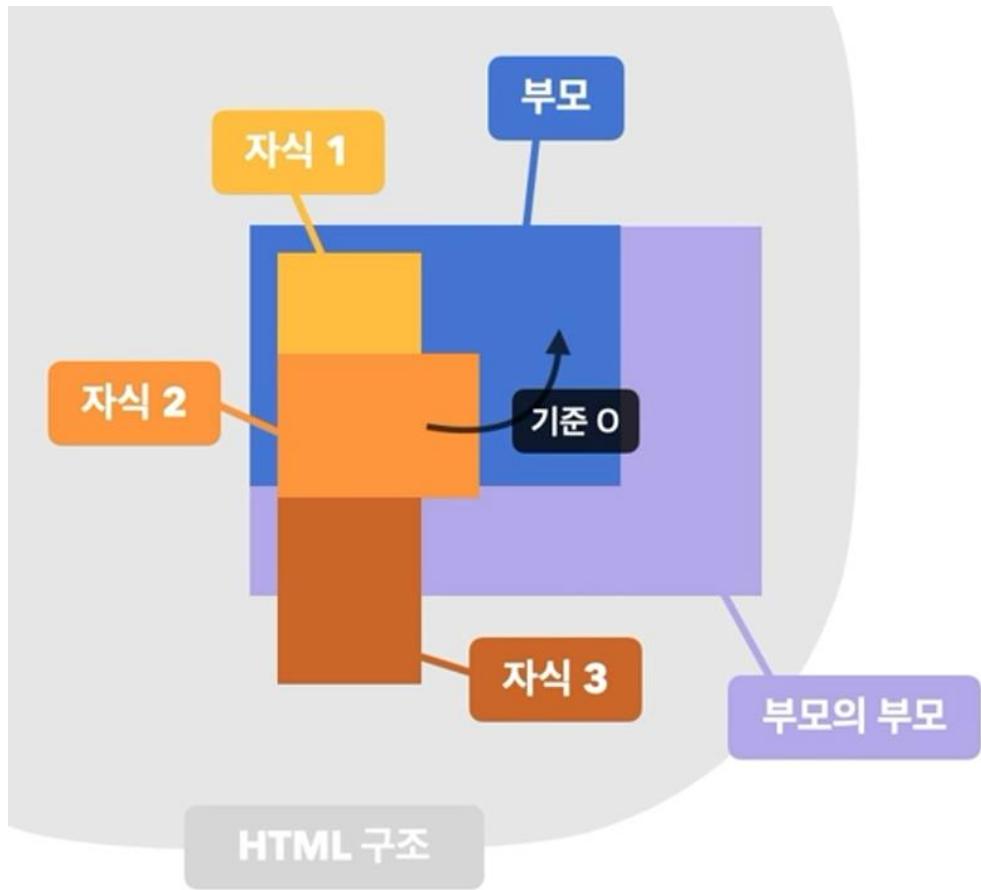
붕~ 뜨면서
요소가 겹쳐요



position: absolute;

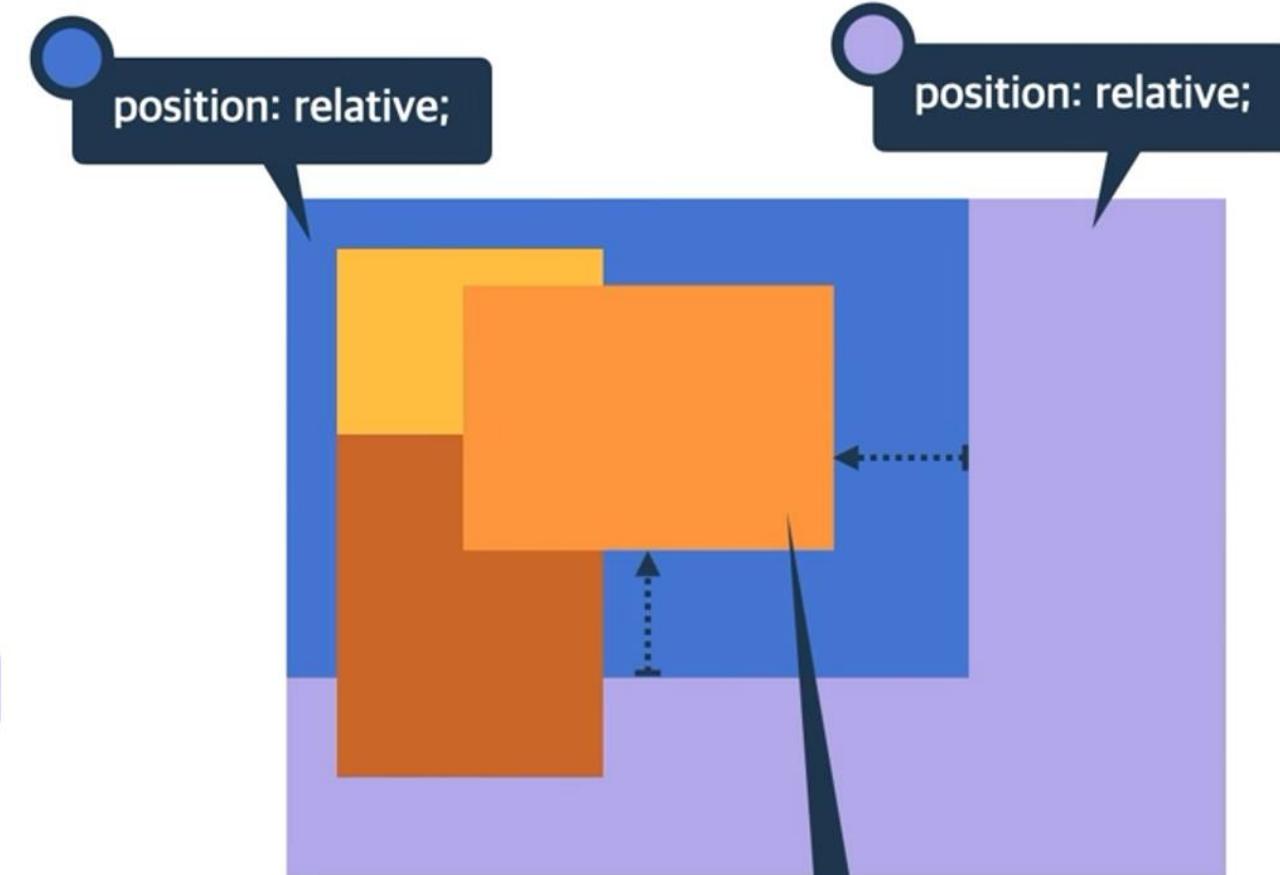
absolute

위치 상 부모 요소를 기준으로 배치!



absolute

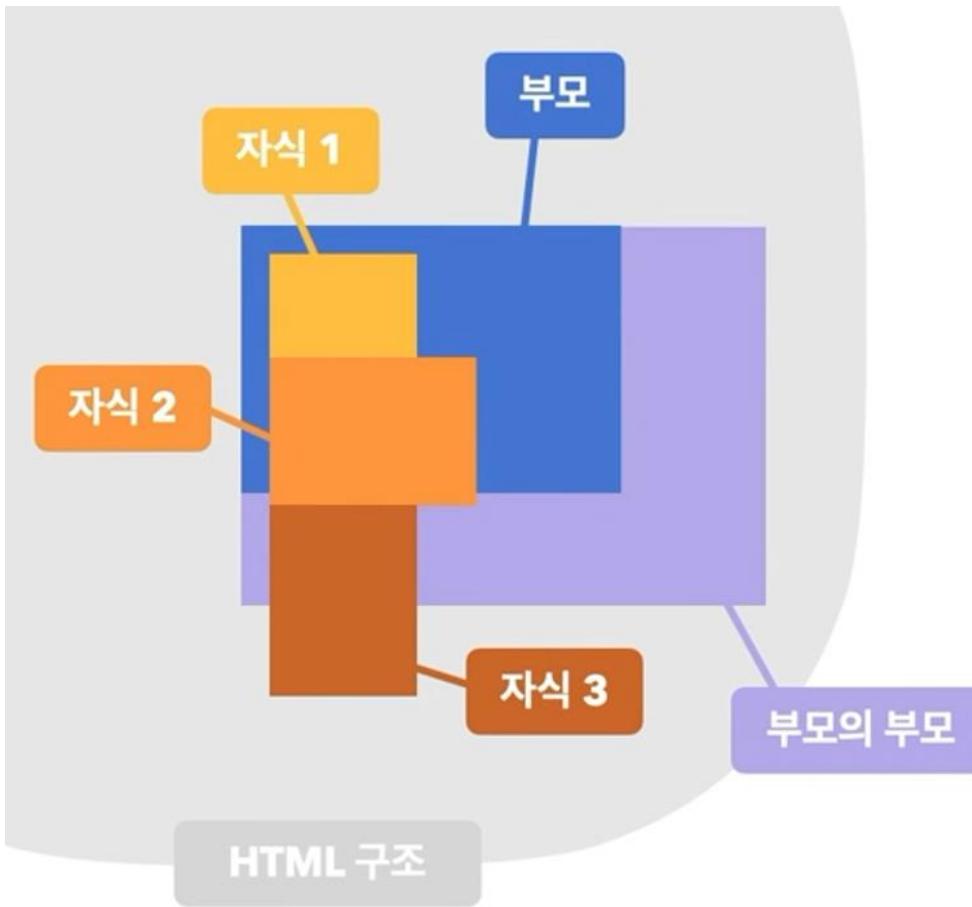
위치 상 부모 요소를 기준으로 배치!



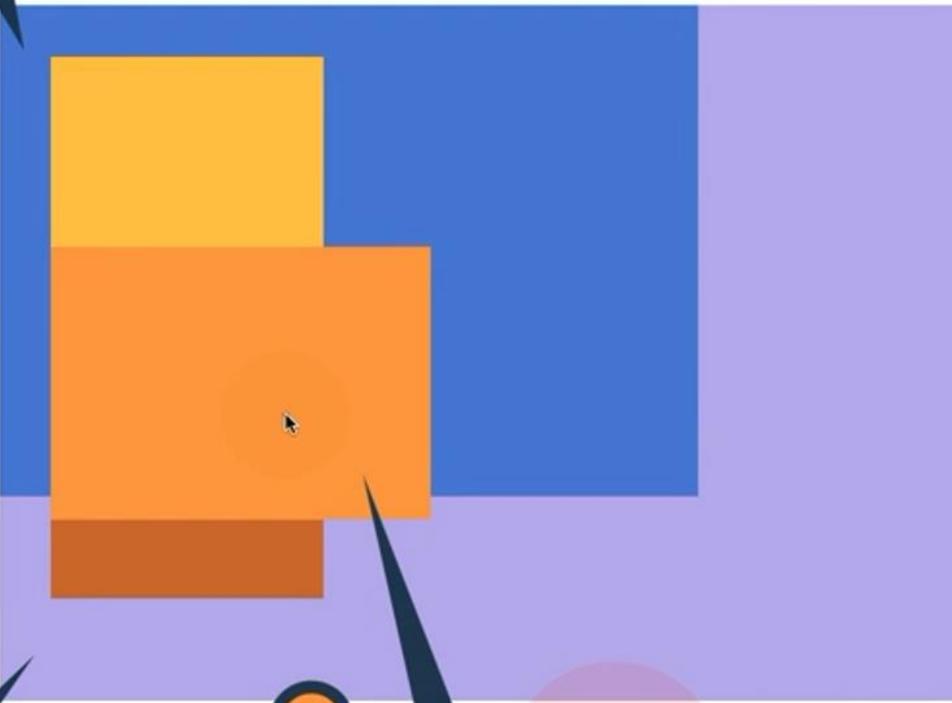
position: absolute;
bottom: 30px;
right: 30px;



fixed



position: relative;

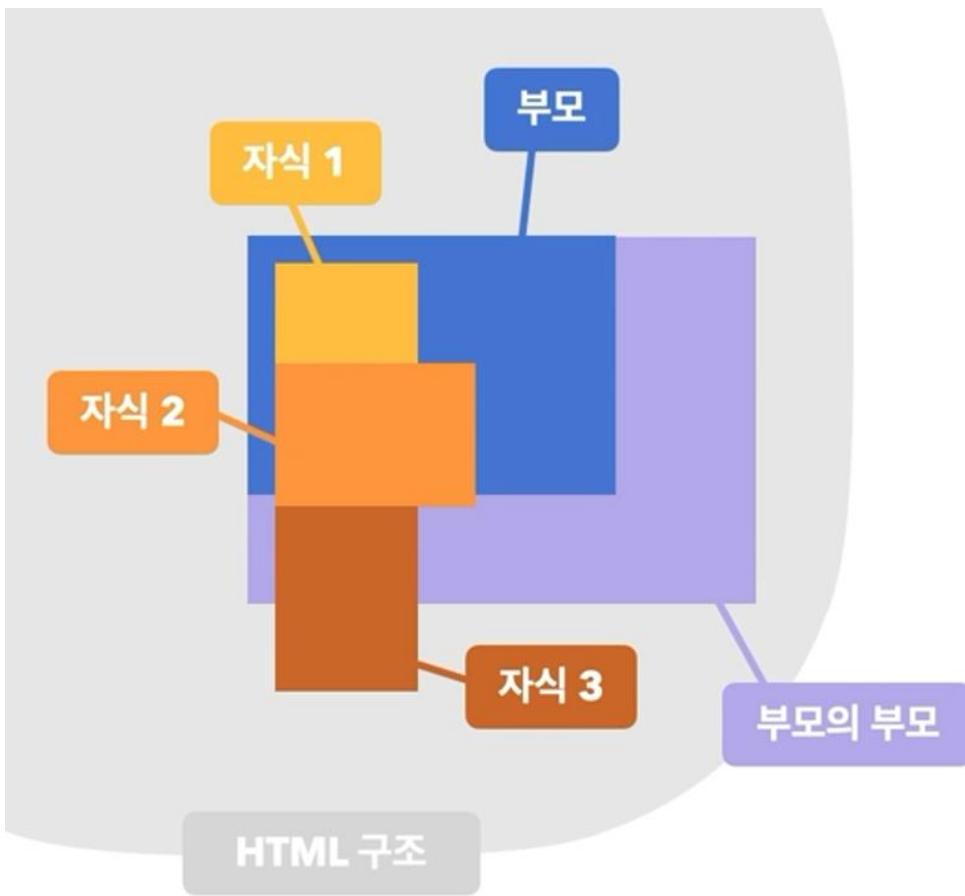


position: relative;

position: fixed;

fixed

뷰포트(브라우저)를 기준으로 배치!



position: relative;

position: relative;

position: fixed;
top: 30px;
right: 30px;

fixed

뷰포트(브라우저)를 기준으로 배치!



요소 쌓임 순서(Stack order)

어떤 요소가 사용자와 더 가깝게 있는지(위에 쌓이는지) 결정

1. 요소에 position 속성의 값이 있는 경우 위에 쌓임.(기본값 static 제외)
2. 1번 조건이 같은 경우, z-index 속성의 숫자 값이 높을 수록 위에 쌓임.
3. 1번과 2번 조건까지 같은 경우, HTML의 다음 구조일 수록 위에 쌓임.



요소의 쌓임 정도를 지정

z-index

auto

부모 요소와 동일한 쌓임 정도

숫자

숫자가 높을 수록 위에 쌓임



요소의 display가 변경됨

position 속성의 값으로 absolute, fixed가 지정된 요소는,
display 속성이 block으로 변경됨.



```
display: block;  
position: absolute;  
top: 30px;  
left: 30px;  
z-index: 1;
```

=

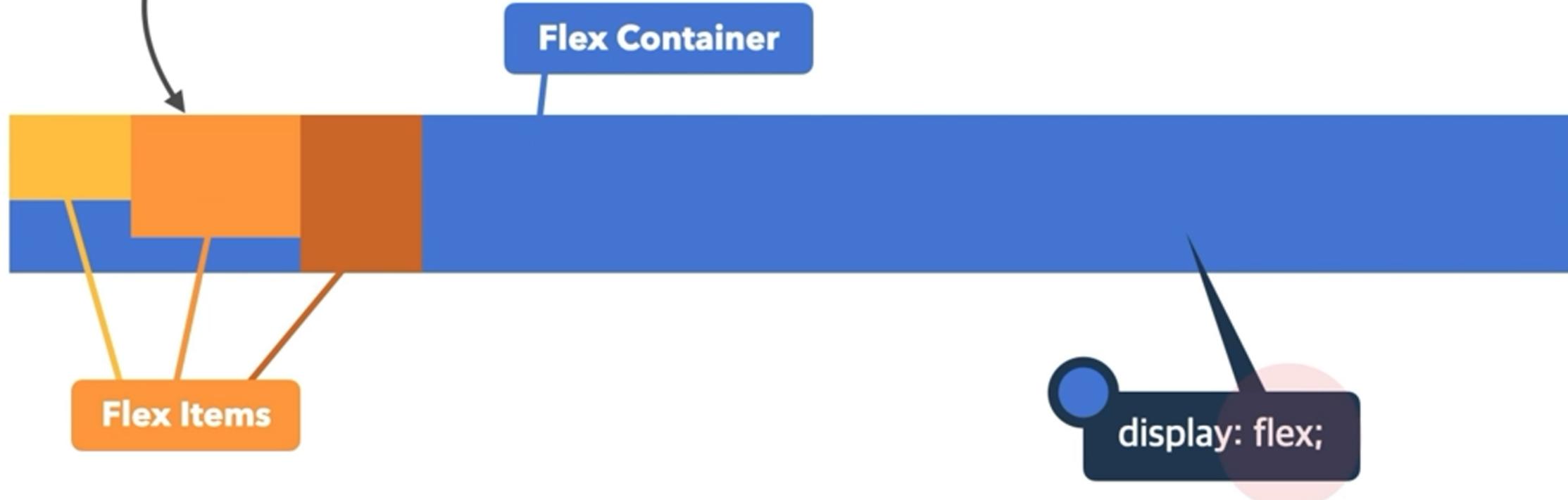
```
position: absolute;  
top: 30px;  
left: 30px;  
z-index: 1;
```



Flex



수평 정렬





Flex Container 속성



Flex Container의 화면 출력(보여짐) 특성

display

flex

블록 요소와 같이 Flex Container 정의

inline-flex

인라인 요소와 같이 Flex Container 정의



Flex Container

`display: inline-flex;`



주 축을 설정

flex-direction

row 행 축 (좌 => 우)

row-reverse 행 축 (우 => 좌)

column 열 축 (위 => 아래)

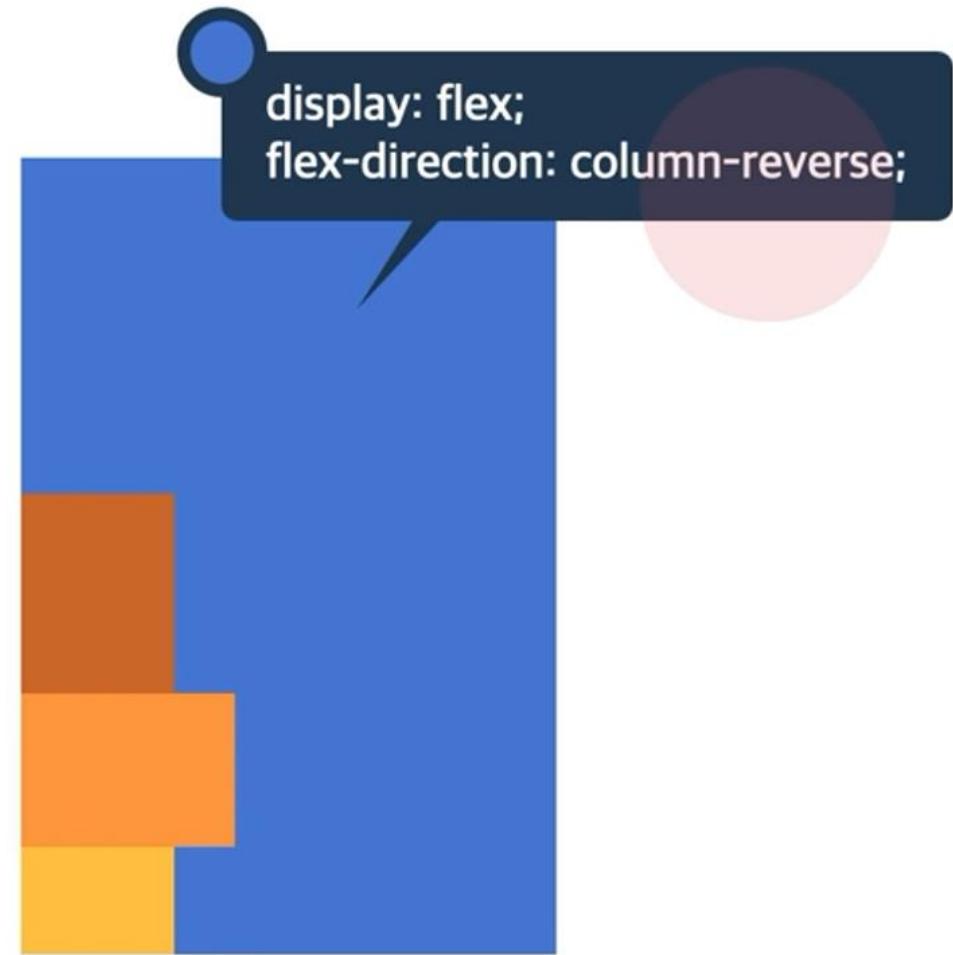
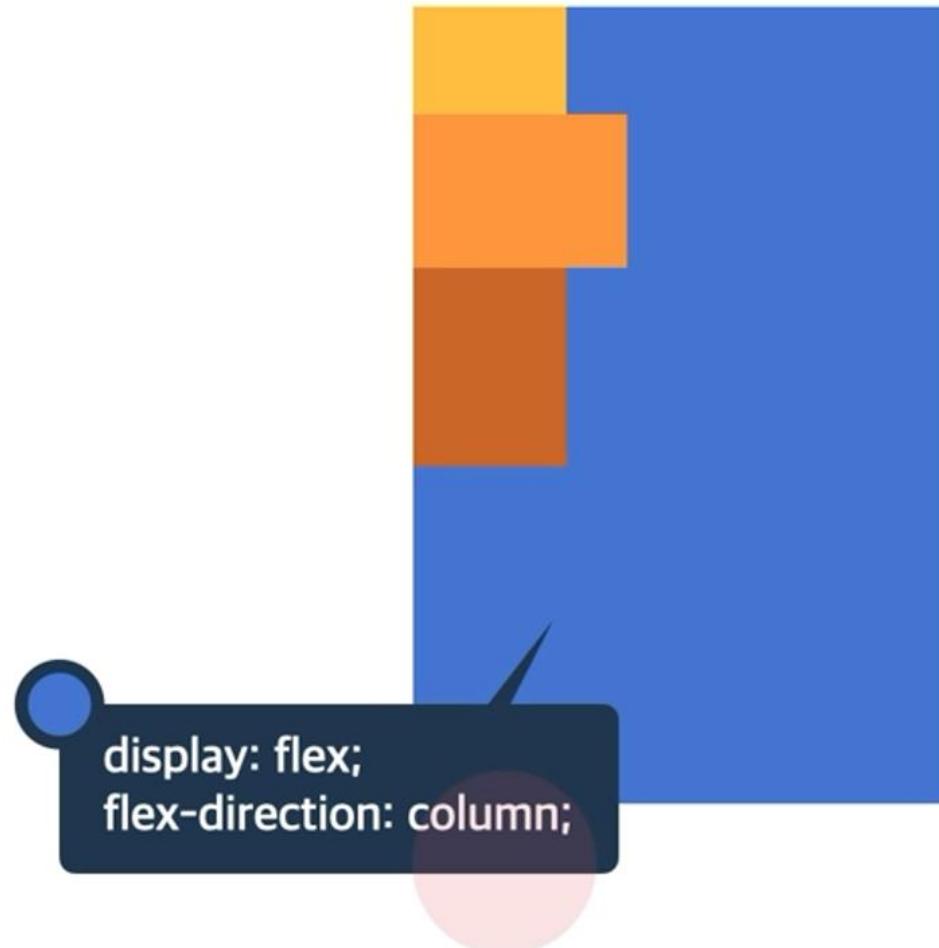
column-reverse 열 축 (아래 => 위)



display: flex;
flex-direction: row;



display: flex;
flex-direction: row-reverse;





Flex Items 묶음(줄 바꿈) 여부

flex-wrap

nowrap

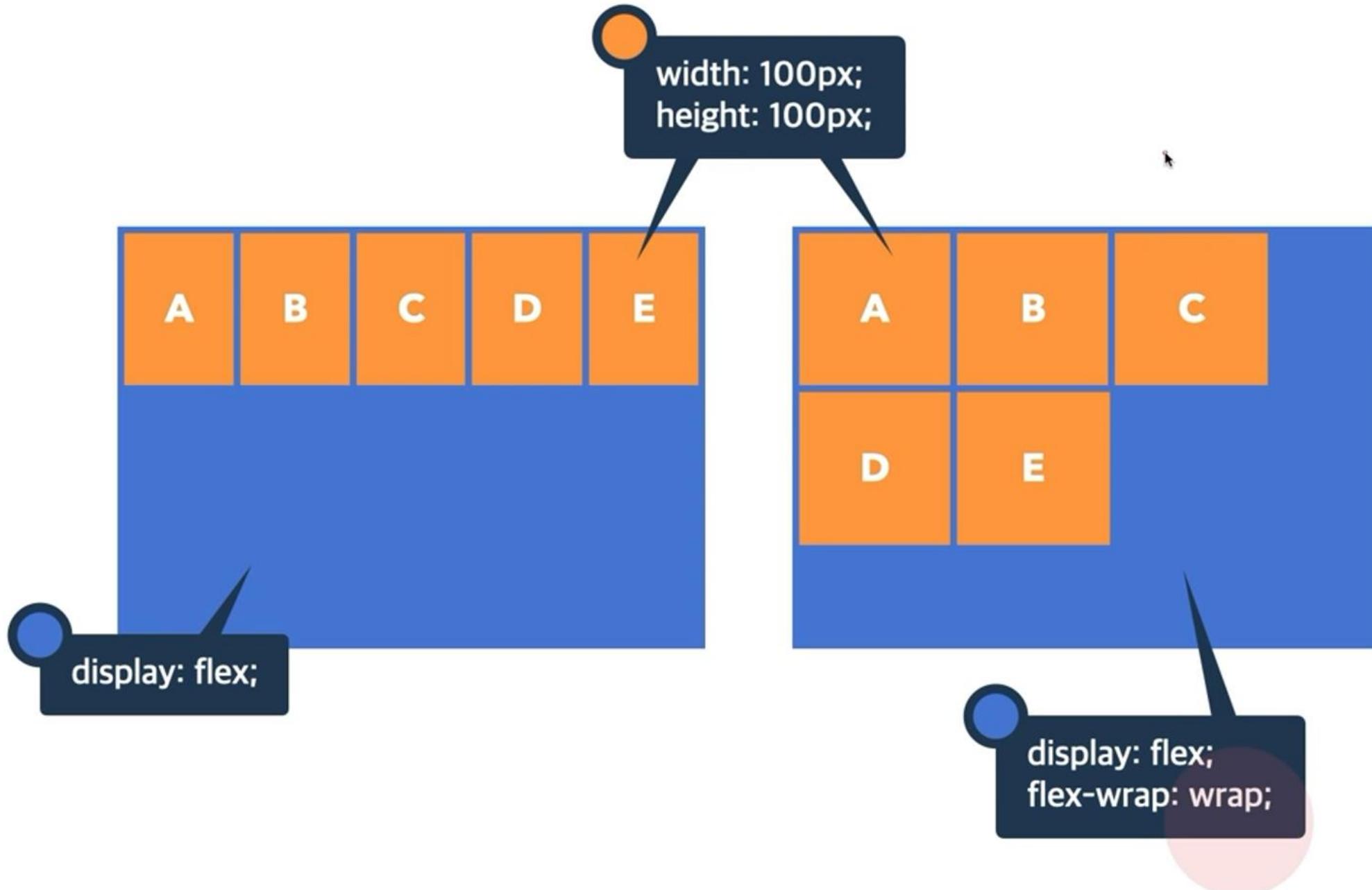
묶음(줄 바꿈) 없음

wrap

여러 줄로 묶음

wrap-reverse

wrap의 반대 방향으로 묶음





주 축의 정렬 방법

justify-content

flex-start

Flex Items를 시작점으로 정렬

flex-end

Flex Items를 끝점으로 정렬

center

Flex Items를 가운데 정렬

space-between

각 Flex Item 사이를 균등하게 정렬

space-around

각 Flex Item의 외부 여백을 균등하게 정렬



display: flex;

display: flex;
justify-content: flex-end;



display: flex;
justify-content: center;





교차 축의 한 줄 정렬 방법

align-items

stretch

Flex Items를 교차 축으로 늘림

flex-start

Flex Items를 각 줄의 시작점으로 정렬

flex-end

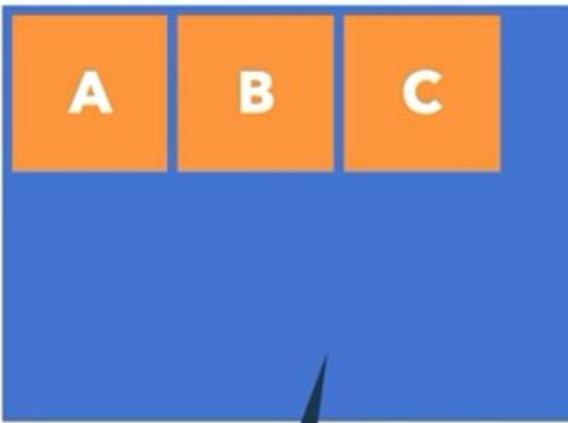
Flex Items를 각 줄의 끝점으로 정렬

center

Flex Items를 각 줄의 가운데 정렬

baseline

Flex Items를 각 줄의 문자 기준선에 정렬



display: flex;
align-items: flex-start;



display: flex;
align-items: center;



display: flex;
align-items: flex-end;



Flex

Item 속성



Flex Item의 순서

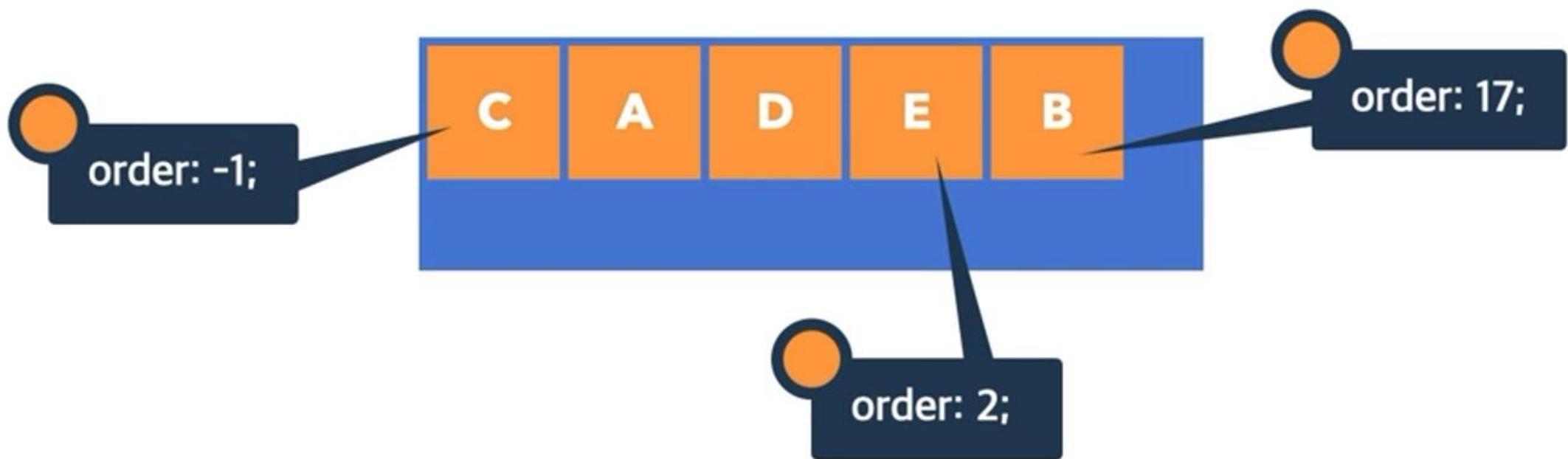
order

0

순서 없음

숫자

숫자가 작을 수록 먼저





Flex Item의 증가 너비 비율

flex-grow

0

증가 비율 없음

숫자

증가 비율





Flex Item의 감소 너비 비율

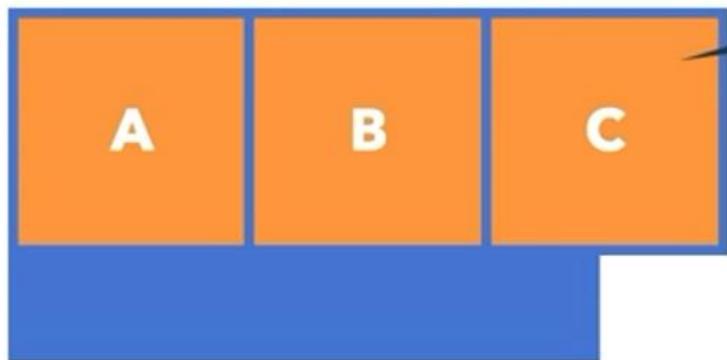
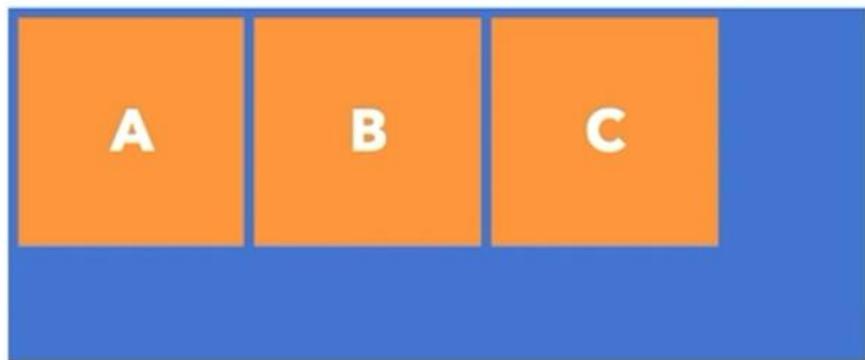
flex-shrink

1

Flex Container 너비에 따라 감소 비율 적용

숫자

감소 비율



`flex-shrink: 0;`



Flex Item의 공간 배분 전 기본 너비

flex-basis

auto

요소의 Content 너비

단위

px, em, rem 등 단위로 지정



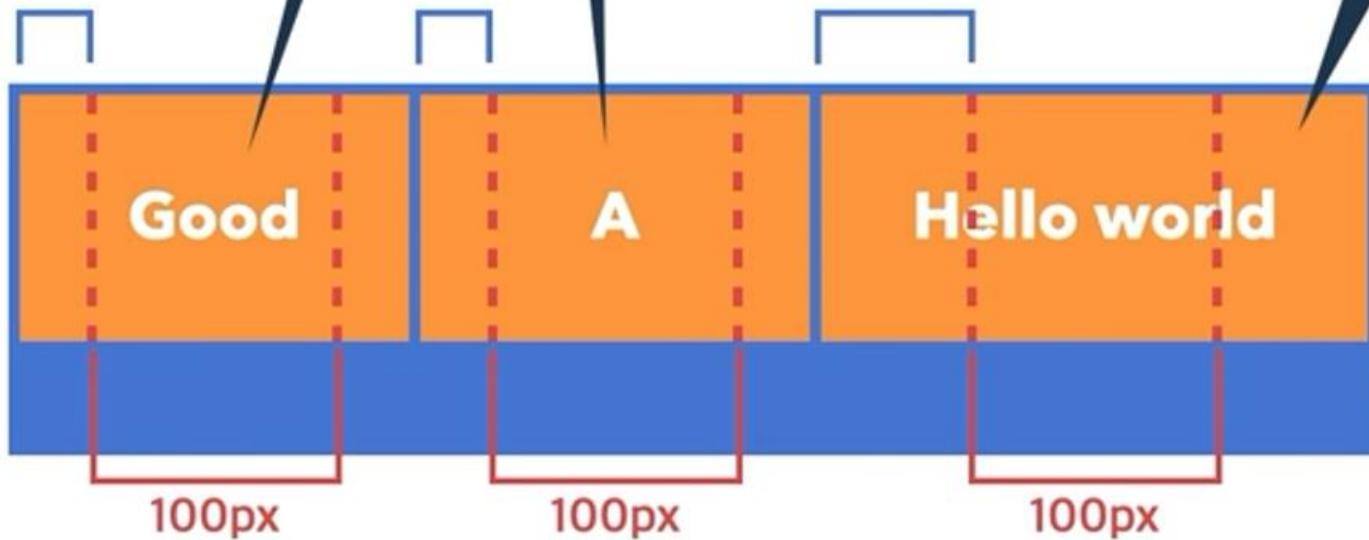
flex-grow: 1;
flex-basis: 100px;

1

1

2

flex-grow: 2;
flex-basis: 100px;





지금 시작합니다



전환

(Transition)



단축형으로 작성할 때,
필수 포함 속성!

요소의 전환(시작과 끝) 효과를 지정하는 단축 속성

transition: 속성명 **지속시간** 타이밍함수 대기시간;

transition-property

transition-duration

transition-timing-function

transition-delay



전환 효과의 지속시간을 지정

transition-duration

0s

전환 효과 없음

시간

지속시간(s)을 지정



```
<style>
  div {
    width: 100px;
    height: 100px;
    background-color: orange;
    transition-duration: 2s;
  }

  div:hover {
    width: 200px;
    height: 200px;
    background-color: skyblue;
  }
</style>
```



전환 효과를 사용할 속성 이름을 지정

transition-property

all

모든 속성에 적용

속성이름

전환 효과를 사용할 속성 이름 명시



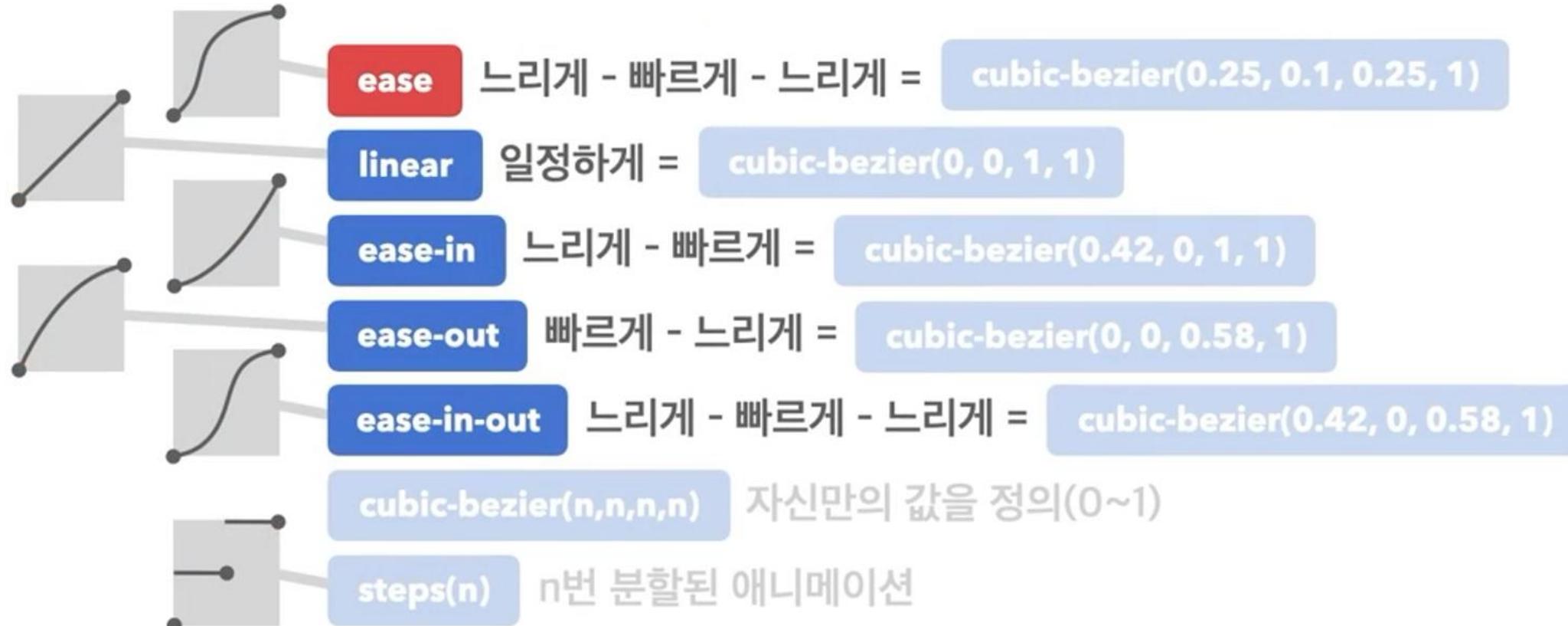
```
<style>
  div {
    width: 100px;
    height: 100px;
    background-color: orange;
    transition-duration: 2s;
    transition-property: width, height;
  }

  div:hover {
    width: 200px;
    height: 200px;
    background-color: skyblue;
  }
</style>
```



전환 효과의 타이밍(Easing) 함수를 지정

transition-timing-function





전환 효과가 몇 초 뒤에 시작할지 대기시간을 지정

transition-delay

0s

대기시간 없음

시간

대기시간(s)을 지정

실습, 트랜지션 속성 사용하기



- 오렌지색 div(100×100) 요소를 클릭하면, 오른쪽(300×100)과
같이 변하도록 만들어 주세요
- 배경색은 1초, 넓이 값은 3초에 걸쳐 변화가 일어나야 합니다!



전환

(Transform)



요소의 변환 효과

transform: 변환함수1 변환함수2 변환함수3 ... ;
transform: 원근법 이동 크기 회전 기울임;



px

2D 변환 함수

deg

translate(x, y)

이동(x축, y축)

translateX(x)

이동(x축)

translateY(y)

이동(y축)

scale(x, y)

크기(x축, y축)

scaleX(x)

크기(x축)

scaleY(y)

크기(y축)

없음(배수)

rotate(degree)

회전(각도)

skew(x, y)

기울임(x축, y축)

skewX(x)

기울임(x축)

skewY(y)

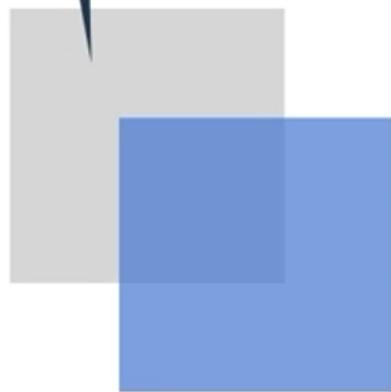
기울임(y축)

matrix(n,n,n,n,n,n)

2차원 변환 효과



`transform: translate(40px, 40px);`



`transform: translateY(40px);`



`transform: translateX(40px);`

또는

`transform: translate(40px, 0);`

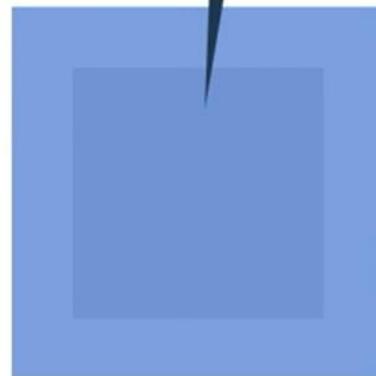




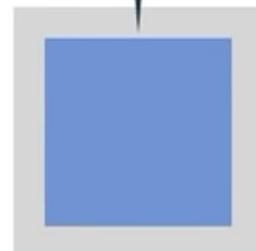
`transform: scale(1.5, 1.5);`

또는

`transform: scale(1.5);`



`transform: scale(0.7);`



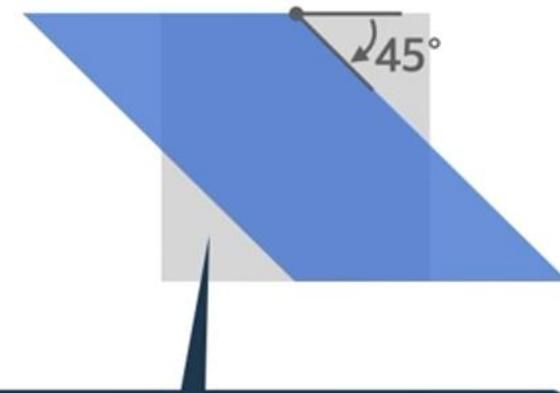
`transform: scaleX(2);`





`transform: skewY(45deg);`

`transform: skewX(-45deg);`



`transform: skewX(45deg);`

또는

`transform: skew(45deg, 0);`





3D 변환 함수

px

`translateZ(z)`

이동(z축)

`translate3d(x, y, z)`

이동(x축, y축, z축)

`scaleZ(z)`

크기(z축)

`scale3d(x, y, z)`

크기(x축, y축, z축)

없음(배수)

`perspective(n)`

원근법(거리)

px

`matrix3d(n, n, n)`

3차원 변환 효과

deg

`rotateX(x)`

회전(x축)

`rotateY(y)`

회전(y축)

`rotateZ(z)`

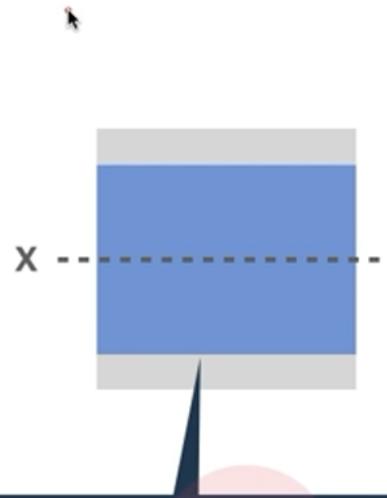
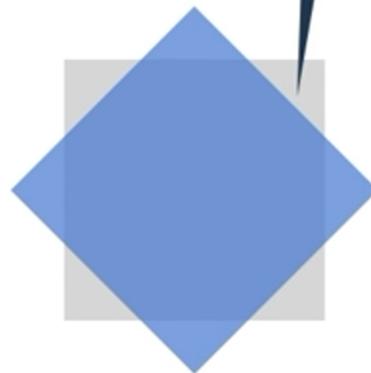
회전(z축)

`rotate3d(x, y, z, a)`

회전(x축, y축, z축, 각도)

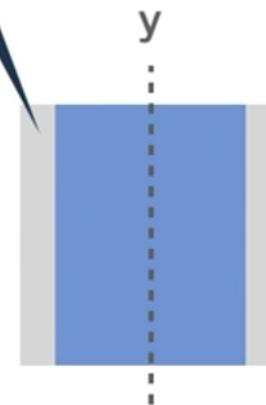


`transform: rotate(45deg);`



`transform: rotateX(45deg);`

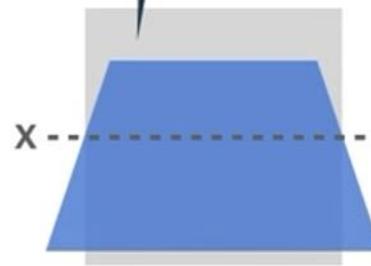
`transform: rotateY(45deg);`



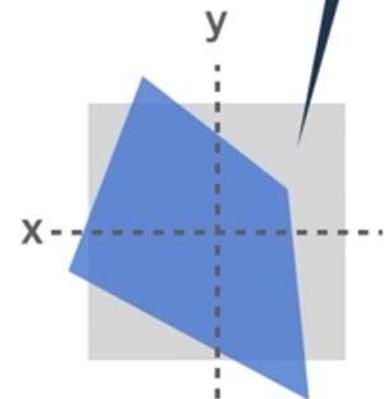
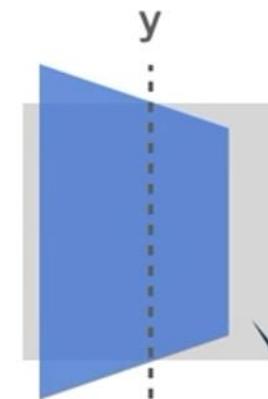


원근법 함수는
제일 앞에 작성해야 합니다!!

`transform: perspective(500px) rotateX(45deg);`



`transform: perspective(500px) rotateY(45deg) rotateX(45deg);`



`transform: perspective(500px) rotateY(45deg);`



하위 요소를 관찰하는 원근 거리를 지정

perspective

단위

px 등 단위로 지정

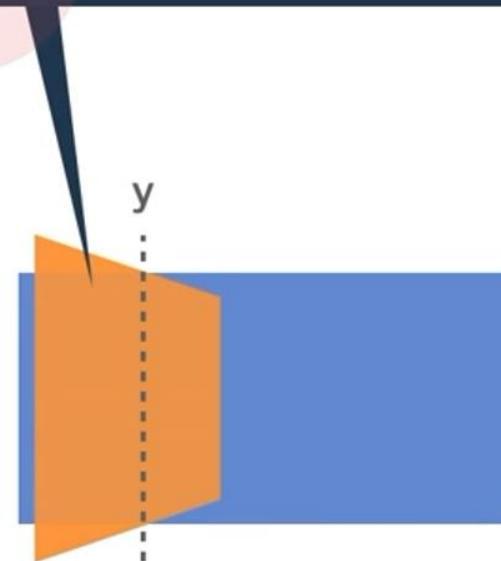
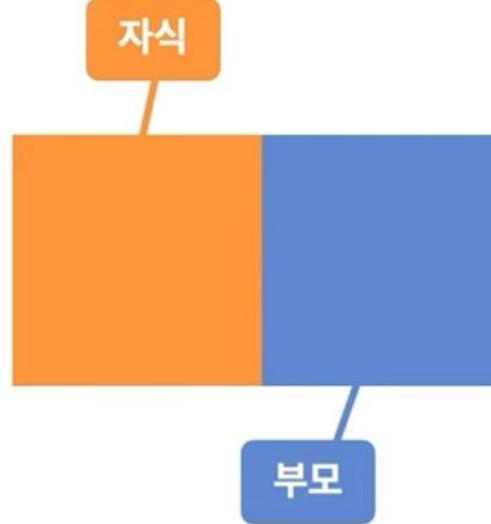


perspective 속성과 함수 차이점

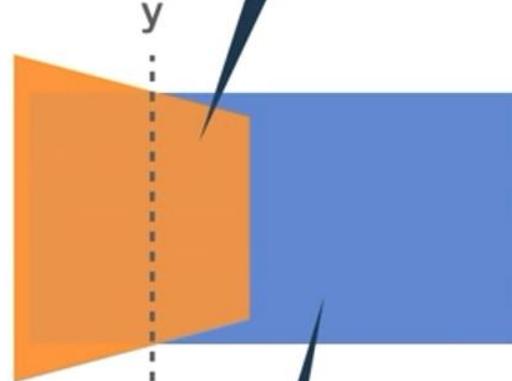
| 속성 / 함수 | 적용 대상 | 기준점 설정 |
|--|-----------|---------------------------------|
| <code>perspective: 600px;</code> | 관찰 대상의 부모 | <code>perspective-origin</code> |
| <code>transform: perspective(600px)</code> | 관찰 대상 | <code>transform-origin</code> |



transform: perspective(500px) rotateY(45deg);



transform: rotateY(45deg);



perspective: 500px;



3D 변환으로 회전된 요소의 뒷면 숨김 여부

backface-visibility

visible

뒷면 보임

hidden

뒷면 숨김



`transform: rotateY(180deg);`

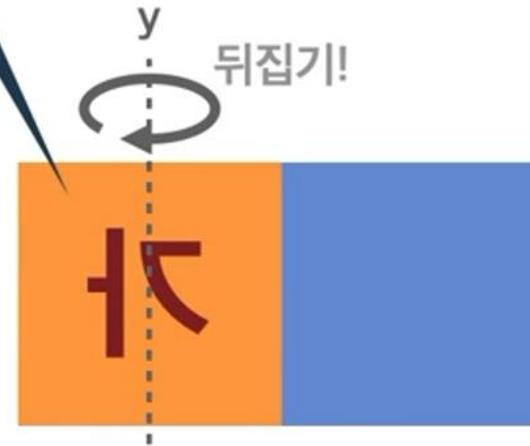


자식

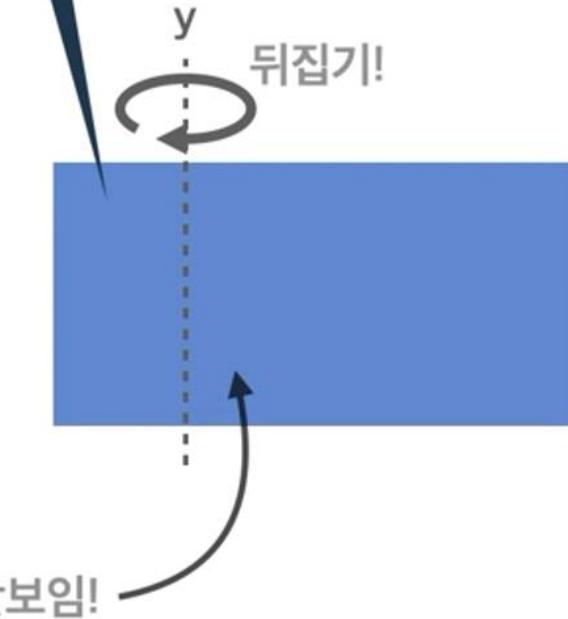
가

부모

뒤집기!

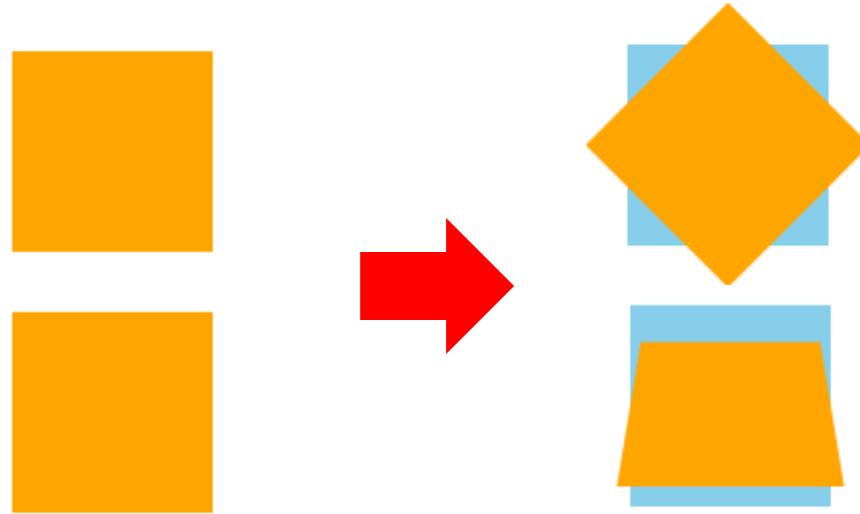


`transform: rotateY(180deg);
backface-visibility: hidden;`



안보임!

실습, 트랜스폼 속성 사용하기



- 각각 오렌지색 div 요소에 마우스가 hover 되면 위와 같이 변화하는 코드를 작성해 주세요!
- 포지션 개념을 이용해서 오렌지 div 요소 아래에 위치하는 하늘색 div 요소도 작성해 주세요~!



Responsive web

반응형 웹 디자인 예시들



- 카카오 모빌리티
 - <https://www.kakaomobility.com/>
- 삼성 뉴스 룸
 - <https://news.samsung.com/kr/>
 - 워드 프레스 사용
- 네이버
 - <https://m.naver.com/>
 - <https://naver.com/>

왜? 반응형이 필요할까요?





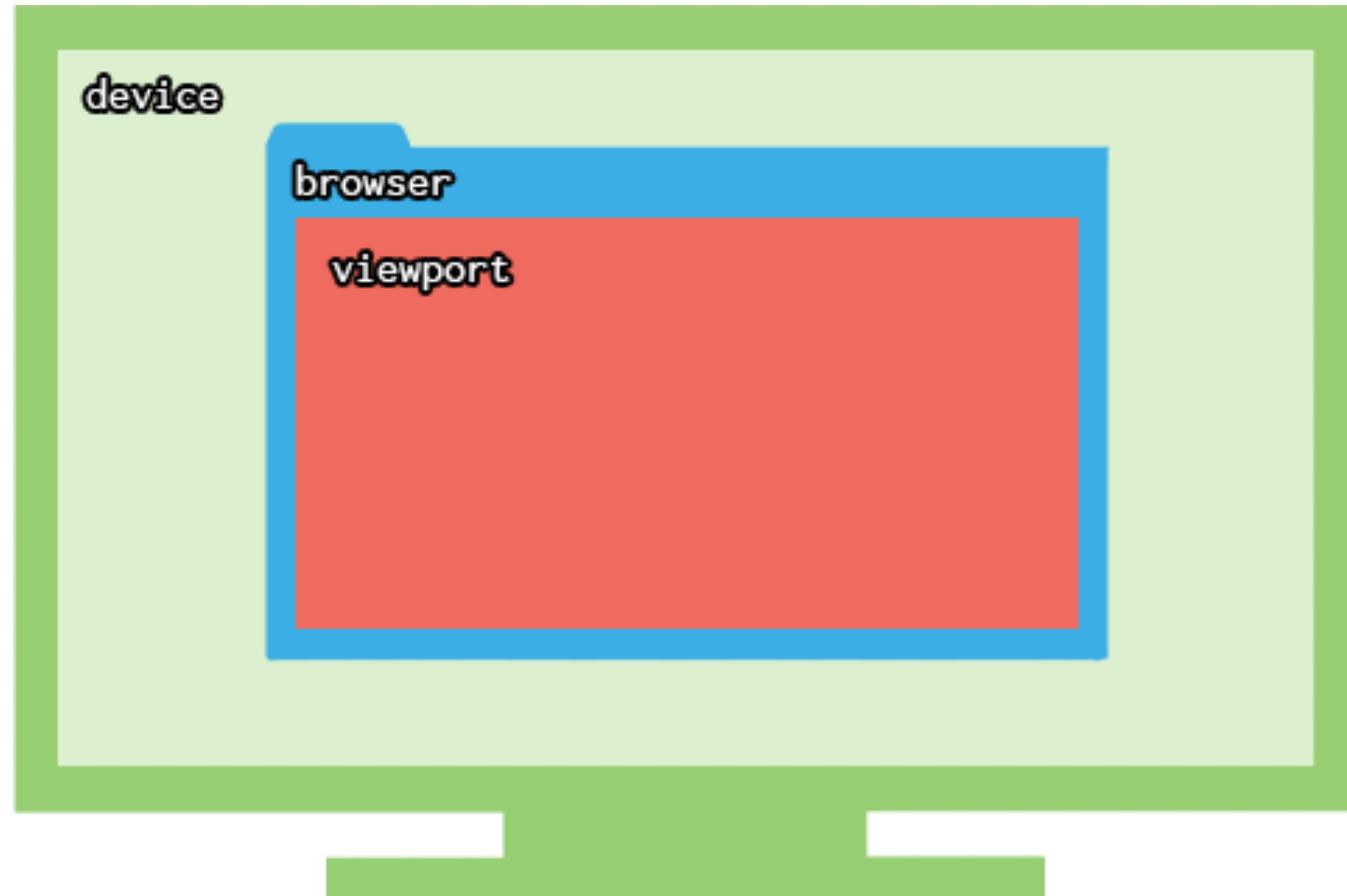
<https://youtu.be/bdWnD6VqLAs>

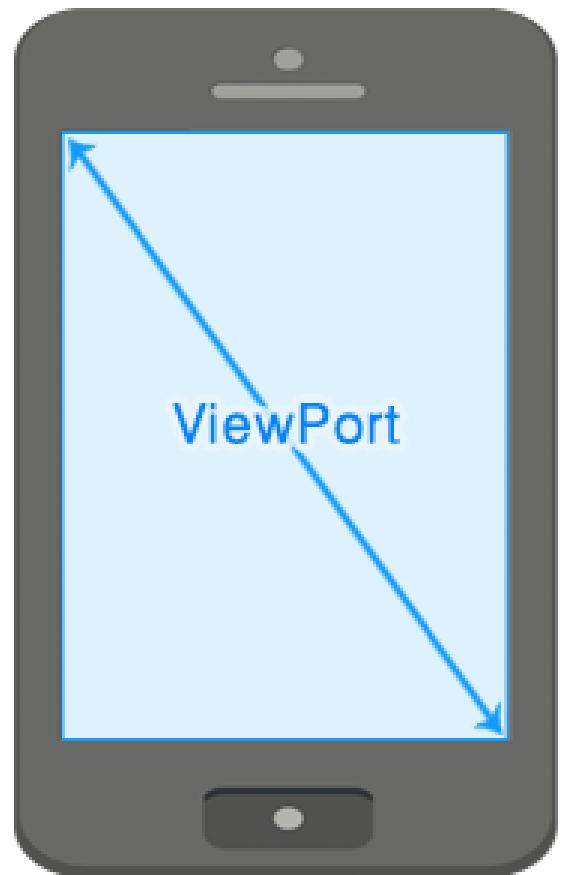


Viewport!

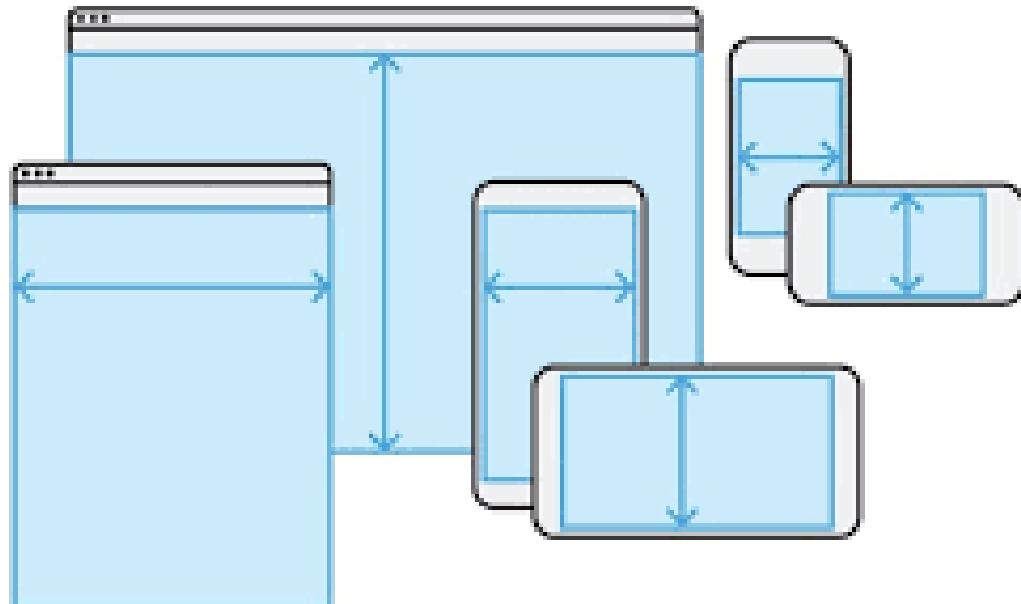


- 그래서 탄생! Viewport
- 기기마다 화면 사이즈가 다르기 때문에 기기에 맞춰 디자인을 하기 위한 크기 요소
- 디바이스 화면 크기를 고려하여 사용자에게 최적화 된 웹페이지 제공 가능
- Html 기본 코드를 생성해 보세요! Meta 태그가 보이나요?





100vmin = 100% of smallest viewport dimension



@media (미디어 쿼리)



- 서로 다른 미디어 타입(디바이스 화면)에 따라 별도의 CSS를 지정
하게 하는 기술

```
<style>
  body {
    background-color: aqua;
  }
  @media screen and (min-width: 480px) {
    body {
      background-color: lightgreen;
    }
  }
</style>
```



화면 크기에 따른 CSS 적용

@media 미디어유형 and (크기 규칙)

미디어 유형 : All(전부) / Print(인쇄) / Screen(전부)

크기 규칙: min(최소 뷰포트 넓이 설정) / max(최대 뷰포트 넓이 설정)



@media screen and (max-width: 480px)

→ 화면 넓이가 480px 이하일 경우 화면에 적용

@media print and (min-width: 481px)

→ 화면 넓이가 480px 이상일 경우 인쇄에 적용

실습, 미디어 쿼리 사용하기



- <p>미디어 쿼리 실습</p> 태그 삽입
- 화면 넓이 500px을 기준으로 500px 이상일 경우 <body> 태그의 배경색이 하늘색, 300px 이하일 경우 배경색이 오렌지
- 화면 높이 300px을 기준으로 300px 이하일 경우 <p> 태그의 폰트 사이즈를 변경 하고, 500px 이상일 경우 배경색이 초록색



<https://gs.statcounter.com/>







오버워치 캐릭터

선택창 만들기





OVERWATCH



Reset CSS?



Safari

Apple

MacOS, iOS



Firefox

Mozilla

MacOS, MS Windows, Linux OS
Android OS



Chrome

Google

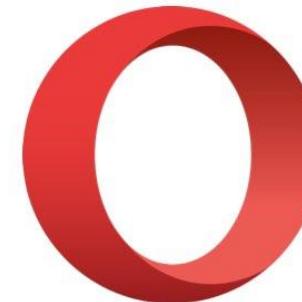
MacOS, MS Windows, Linux OS,
Android OS, Chrome OS



Edge new

Microsoft

MS Windows, MacOS, iOS
Android OS



Opera

Opera Software

MacOS, MS Windows, Linux OS,
Android OS





브라우저의 기본 CSS 속성을 초기화



- <https://www.jsdelivr.com/package/npm/reset-css>
- Copy HTML
- Link 붙여 넣기!



HTML

구성하기

기본 구조는 이렇습니다!

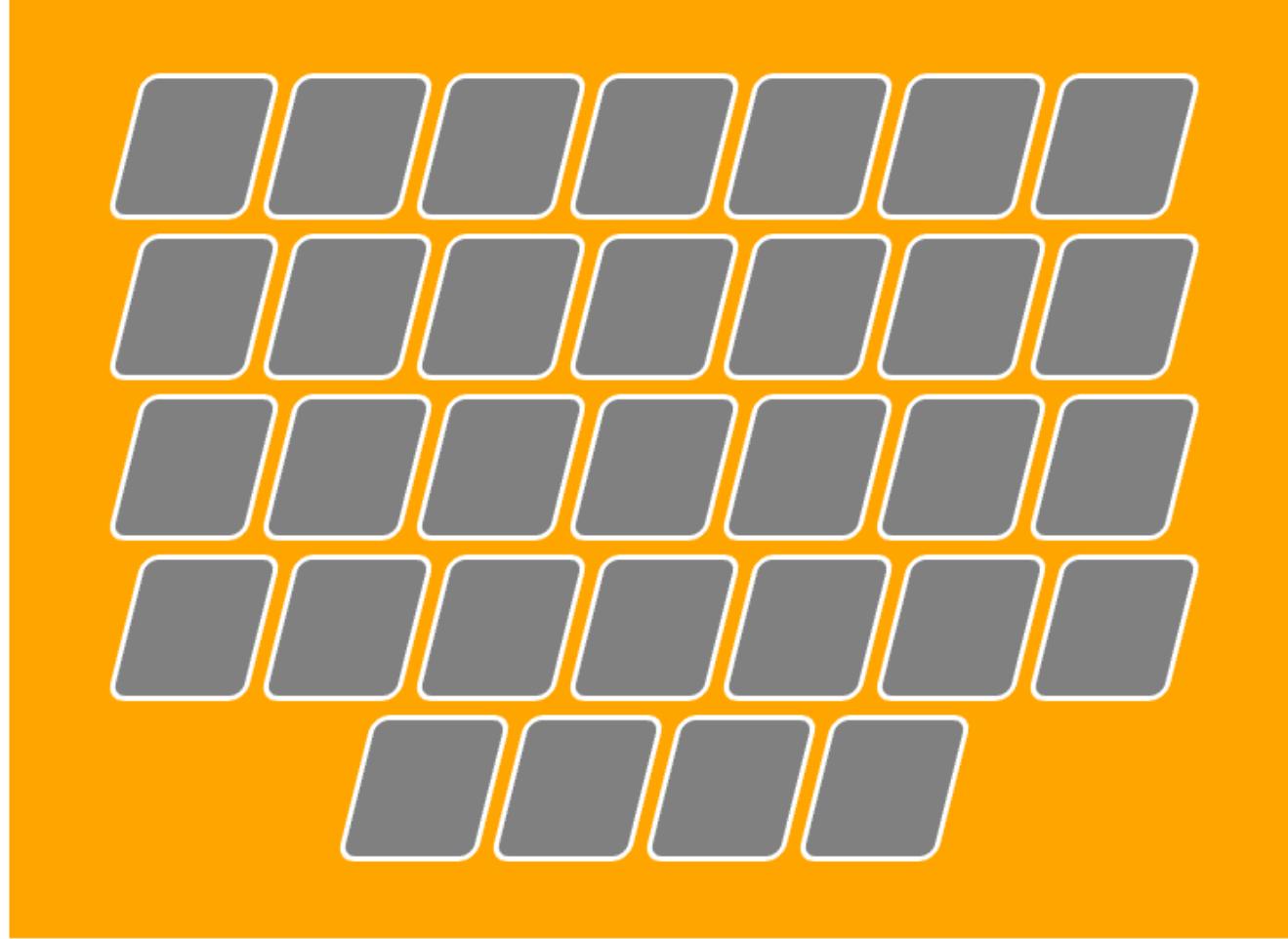


```
<div class="container">
  <div class="heroes">
    <div class="hero">
      <div class="image"></div>
    </div>
    <div class="hero">
      <div class="image"></div>
    </div>
  </div>
</div>
```

- 그리고 영웅은 32명입니다!
- Emmet 사용하기!

```
div.container>div.heroes>div.hero*32>div.image
```

실습, 캐릭터 선택 창 레이아웃 짜기!



실습, 캐릭터 선택 창 레이아웃 짜기!



- Heroes 클래스
 - div 의 가로 넓이는 700px
- Hero 클래스
 - 크기 : 80px 90px
 - 마진 : 4px
 - 테두리 : 3px solid white
 - 박스 기울임 : -14deg
 - Border-radius : 10px



Transform

적용

캐릭터에 hover 시, 선택창이 커지게



- Transform 적용 시작
- 커져야 하니까? → scale()
- 어 그런데 그냥 사각형이 되네요?
 - Transition 의 속성 이해하기!
 - 변하는 속성 2개를 전부 지정
 - 시간은 transform 0.1s / 색상(배경, 테두리)은 0.5s
 - 확대 시 옆의 요소 보다 쌓임 순서가 높아야 하므로 z-index: 1





```
.container .heroes .hero:hover {  
background-color: orange;  
border-color: orange;  
transform: skew(-14deg) scale(1.3);  
transition: transform 0.1s, background-color 0.5s, border-color 0.5s;  
z-index: 1;  
}
```





백그라운드 이미지 넣기

캐릭터 이미지 넣기!

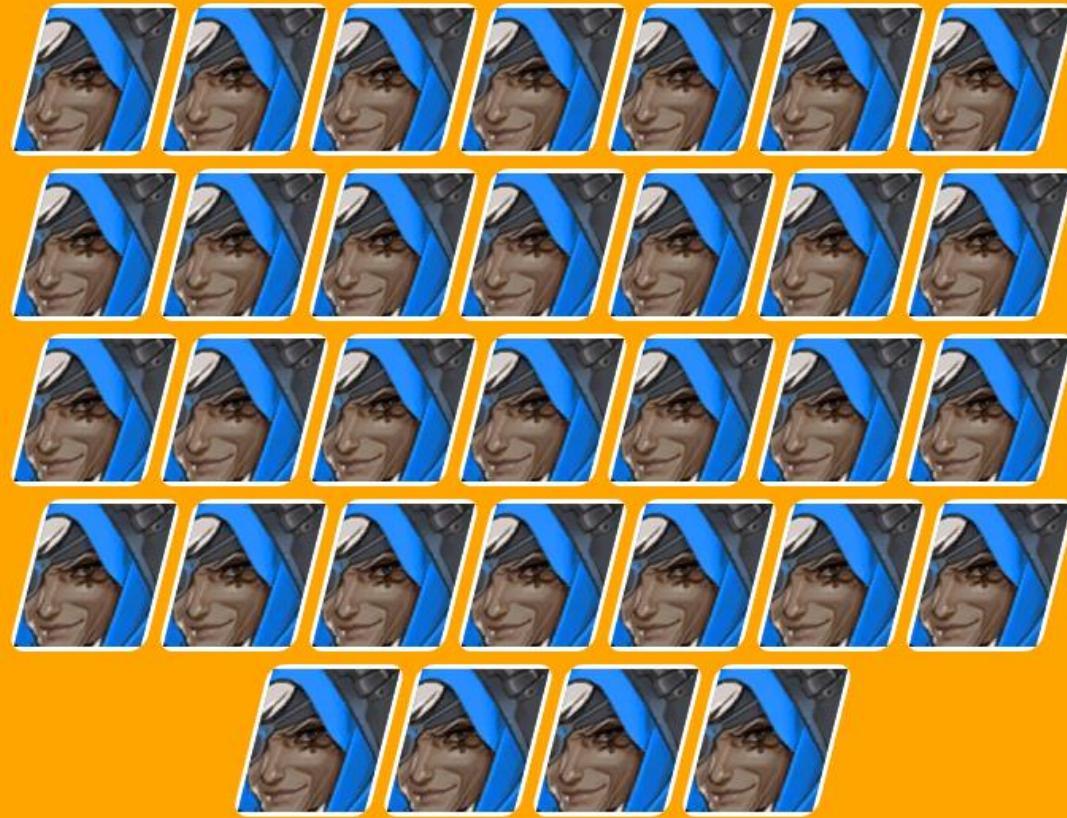


- 모든 이미지는 `images` 폴더에 순서대로 있습니다!
- `Hero01.png` 파일을 `image` 클래스에 백그라운드 이미지로 넣어 봅시다!

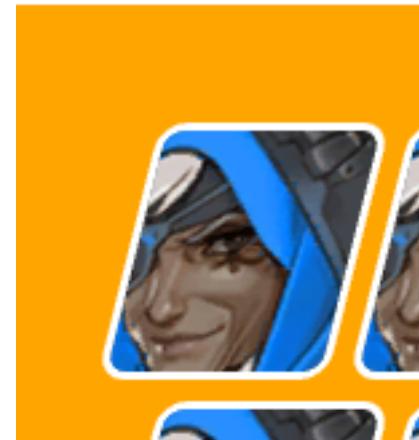
```
.container .heroes .hero .image {  
    background-image: url("./images/hero1.png");  
    width: 100%;  
    height: 100%;  
    background-position: center;  
    background-repeat: no-repeat;  
}
```



- 이미지가 빠져 나오네요!?



```
.container .heroes .hero {  
    width: 80px;  
    height: 90px;  
    margin: 4px;  
    border: 3px solid white;  
    transform: skew(-14deg);  
    background-color: gray;  
    border-radius: 10px;  
    overflow: hidden;  
}
```



캐릭터 이미지 넣기!



- 일단 이미지의 크기부터 조절 합시다!
 - `Background-size : cover`
- `Image` 클래스는 어떤 모양일까요? 확인해 봅시다!



- 기울어서 사진이 이상하게 보이니까, `image` 클래스에서 `skew` 속성을 다시 돌려 봅시다! `transform: skew(14deg)`

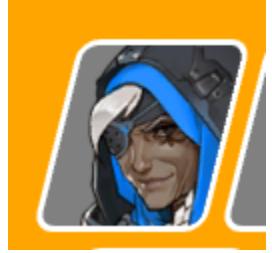
캐릭터 이미지 넣기!



- 이미지는 방향이 제대로 되었지만 기울임이 사라져서 캐릭터 선택창 보다 좁아 졌네요!? → 넓이 값을 더 크게 해줍시다!

```
width: 140%;
```

캐릭터 이미지 넣기!



- 왼쪽에 공간이 뜨는 것 같으니, 위치를 이동!

```
transform: skew(14deg) translateX(-10px);
```



실습, 다른 캐릭터 이미지도 적용



- 각각 캐릭터 창에 순서대로 이미지를 적용해 주세요!
- 배우신 선택자를 잘 활용하여서 적용하시면 됩니다! 😊



실습, 배경 이미지 및 로고 삽입!



- 전체 배경을 적용해 봅시다!
 - 배경 이미지는 bg.png 입니다!
- 로고도 삽입해 봅시다!
 - 로고는 img 태그로 삽입을 해주세요!
 - 로고 이미지는 logo_overwatch.png 입니다!
 - 로고는 중앙 정렬이 되어야 합니다!
 - 로고의 최대 크기는 300px 이고 뷰포트에 따라서 무한히 작아질 수 있습니다!



OVERWATCH

