CSS 복잡한 속성

CSS 복잡한 속성 - position

- static : 정적 위치 지정 방식
- relative : 상대 위치 지정 방식
- absolute : 절대 위치 지정 방식
- fixed : 고정 위치 지정 방식

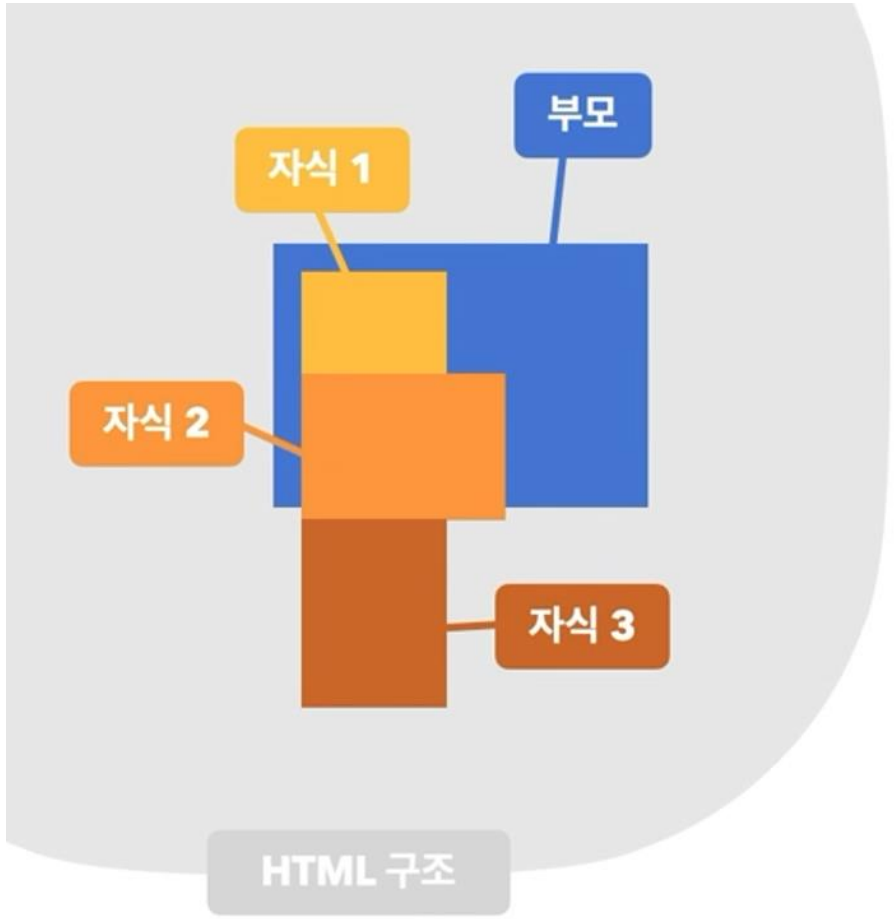
CSS 복잡한 속성 - position

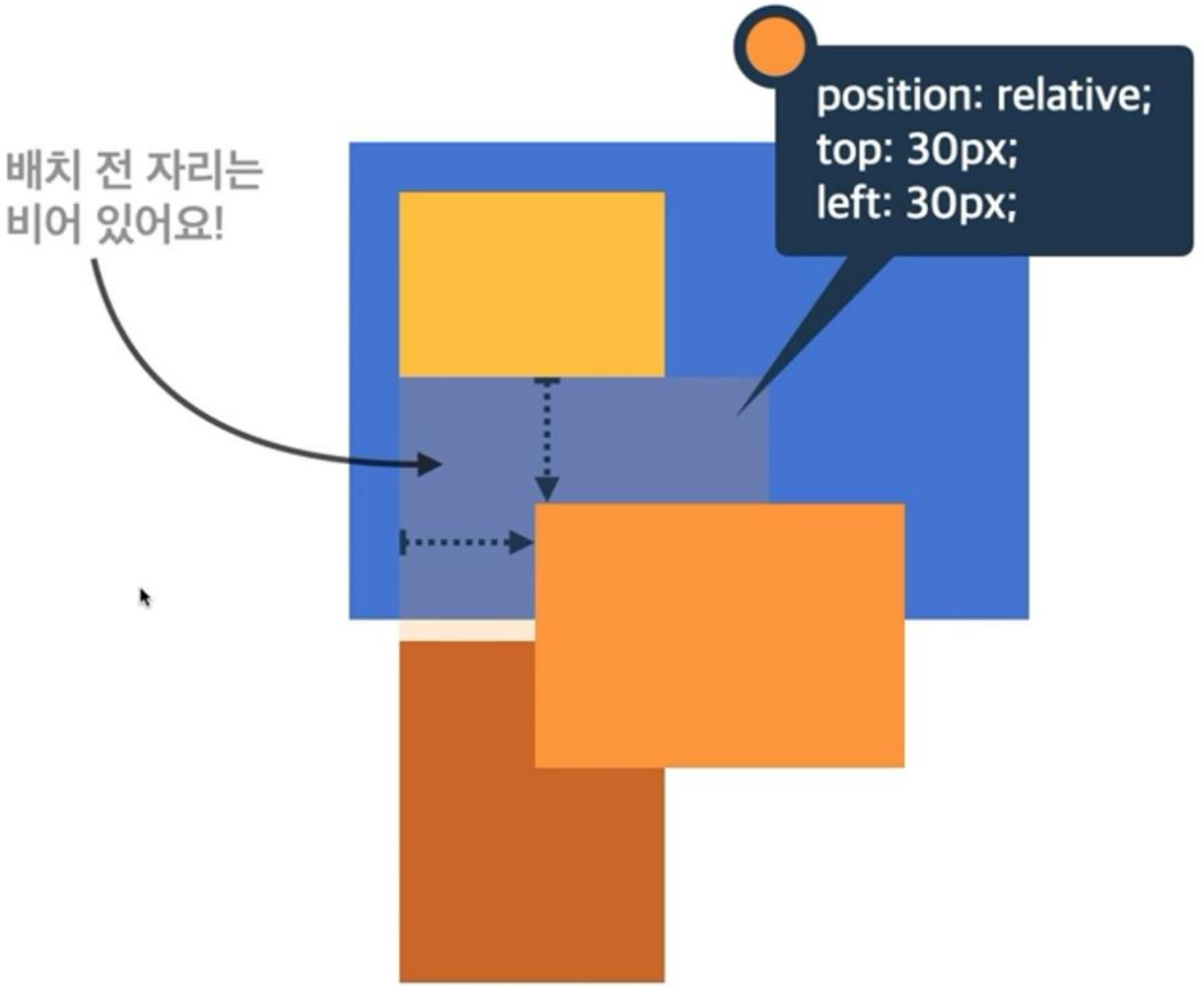
position 속성

Aa 속성	설명
position	HTML 요소의 위치를 결정하는 방식을 설정함.
top	위치가 설정된 조상 요소의 위로부터의 여백을 설정함.
right	위치가 설정된 조상 요소의 오른쪽으로부터의 여백을 설정함.
bottom	위치가 설정된 조상 요소의 아래로부터의 여백을 설정함.
left	위치가 설정된 조상 요소의 왼쪽으로부터의 여백을 설정함.
z-index	겹쳐지는 요소들이 쌓이는 스택(stack)의 순서를 설정함.

(잊지 말자!) position 속성이 정의되어 있어야 사용 가능!

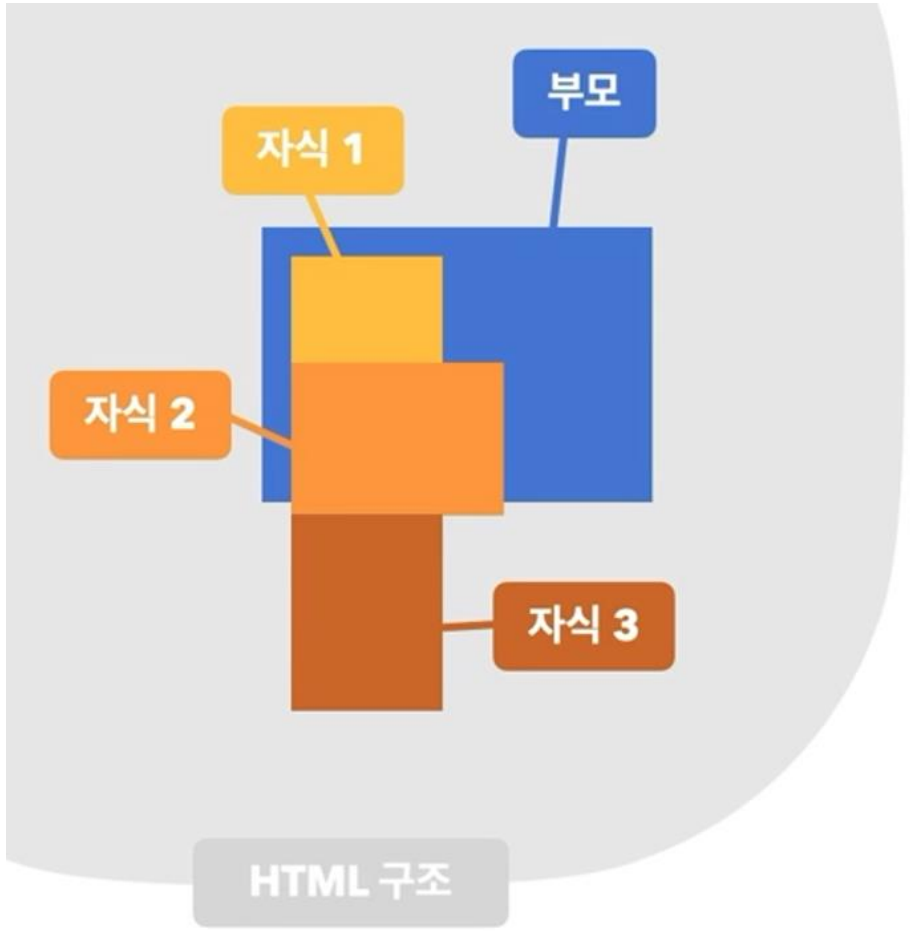
relative





relative 요소 자신을 기준으로 배치!

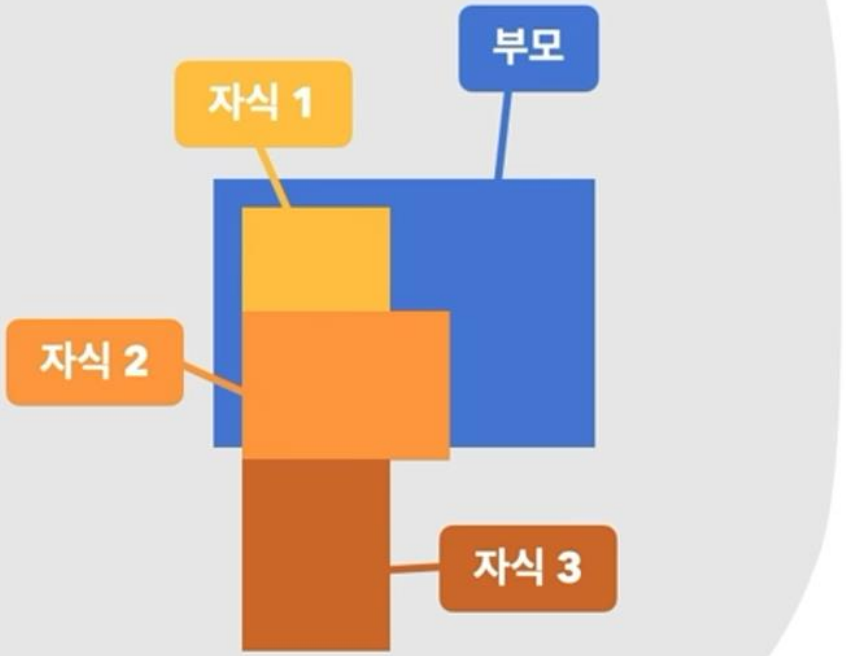
absolute

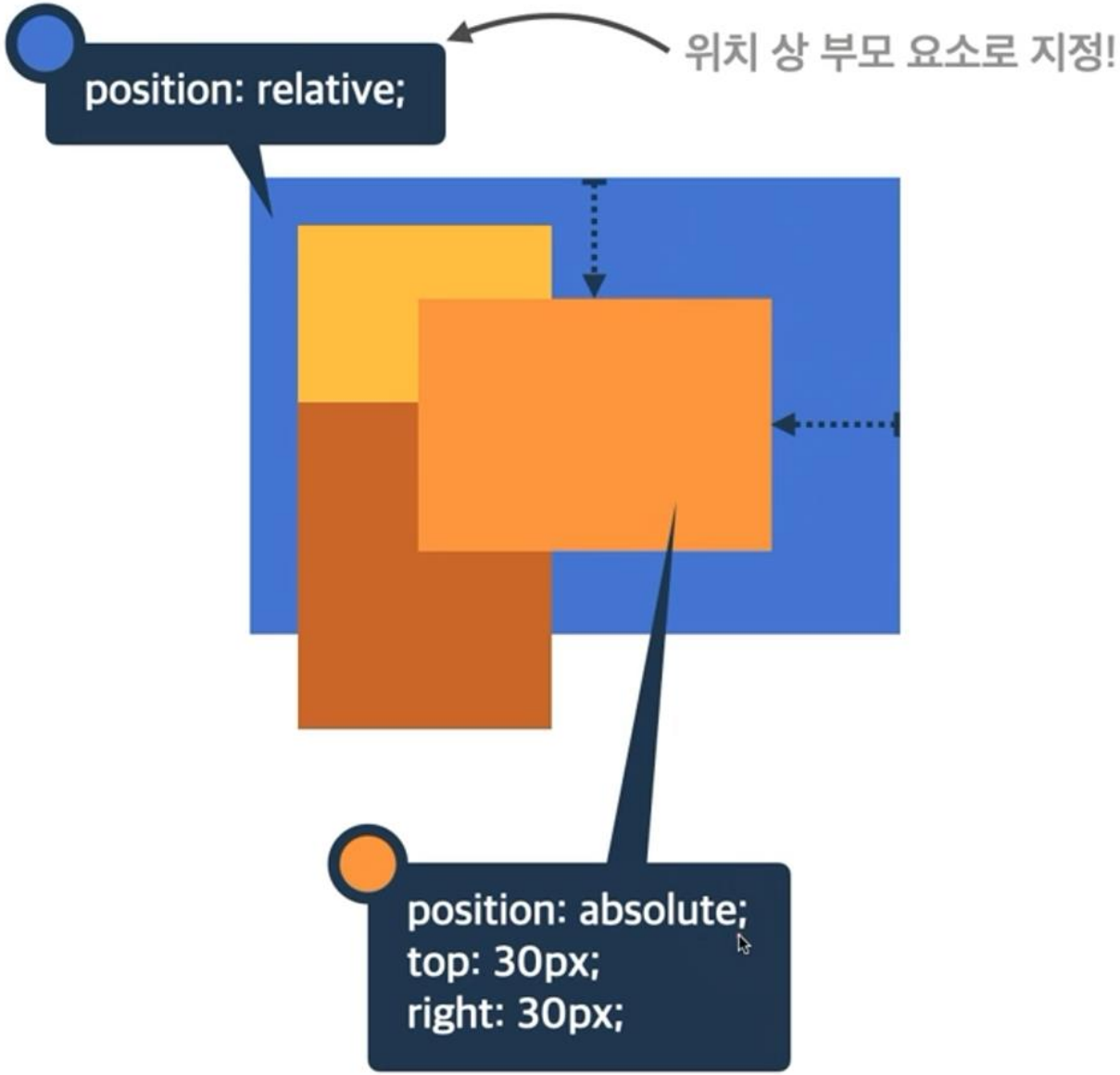


붕~ 뜨면서
요소가 겹쳐요

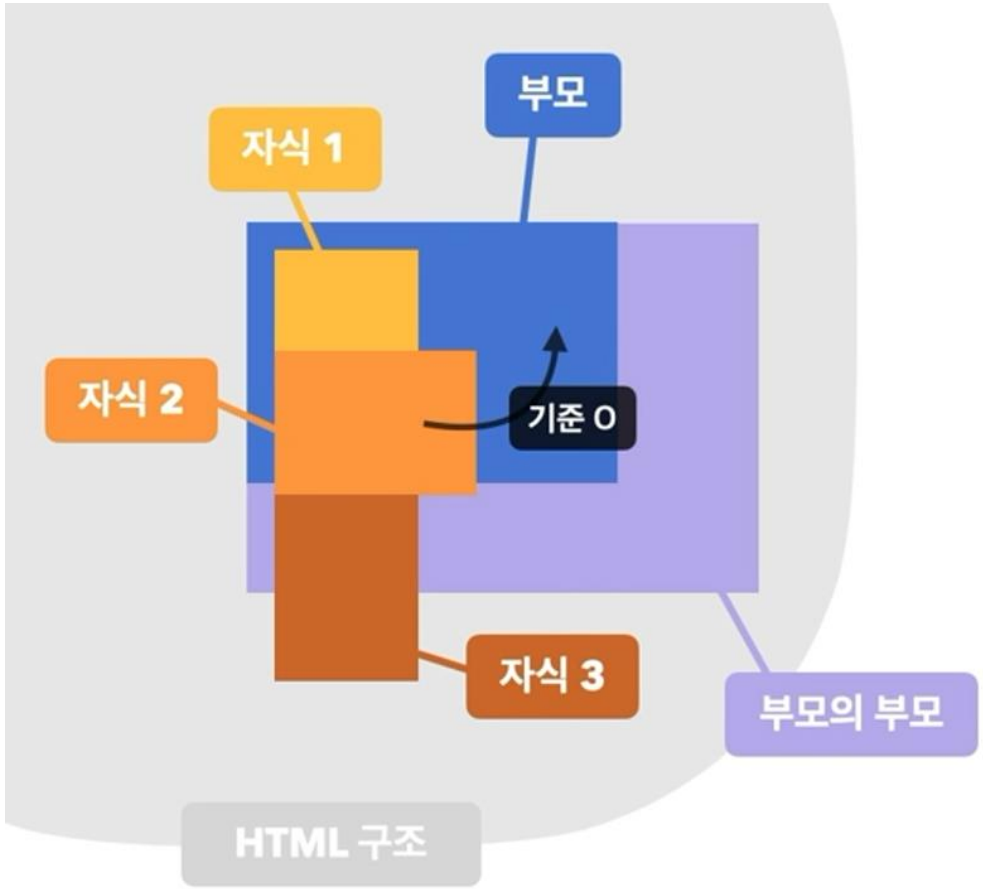
position: absolute;

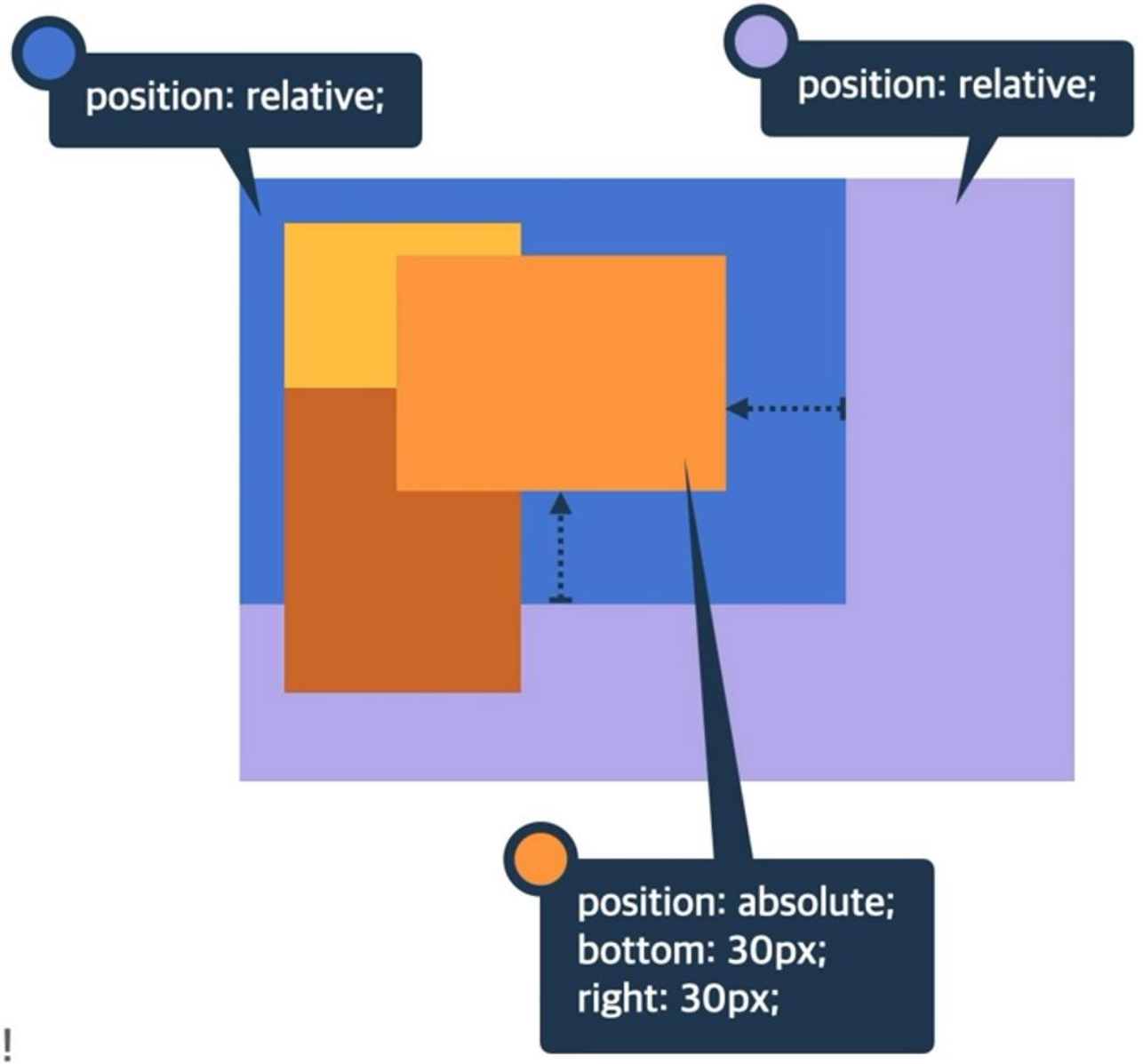
absolute 위치 상 부모 요소를 기준으로 배치!



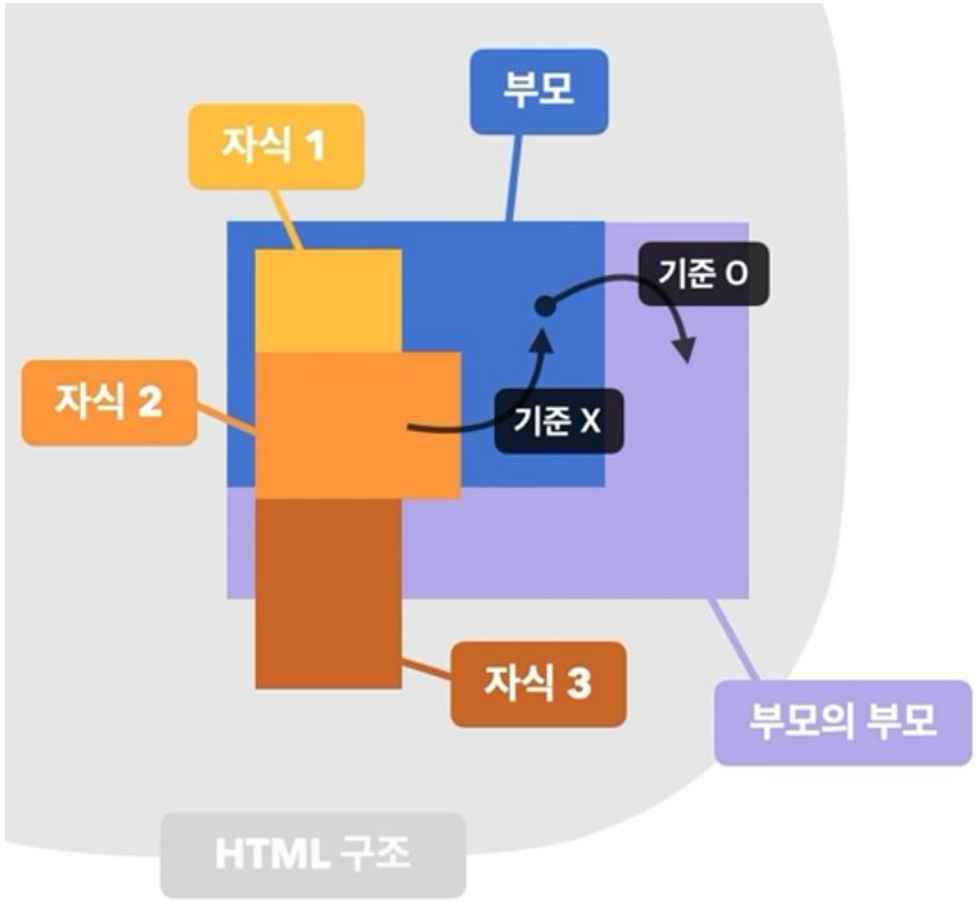


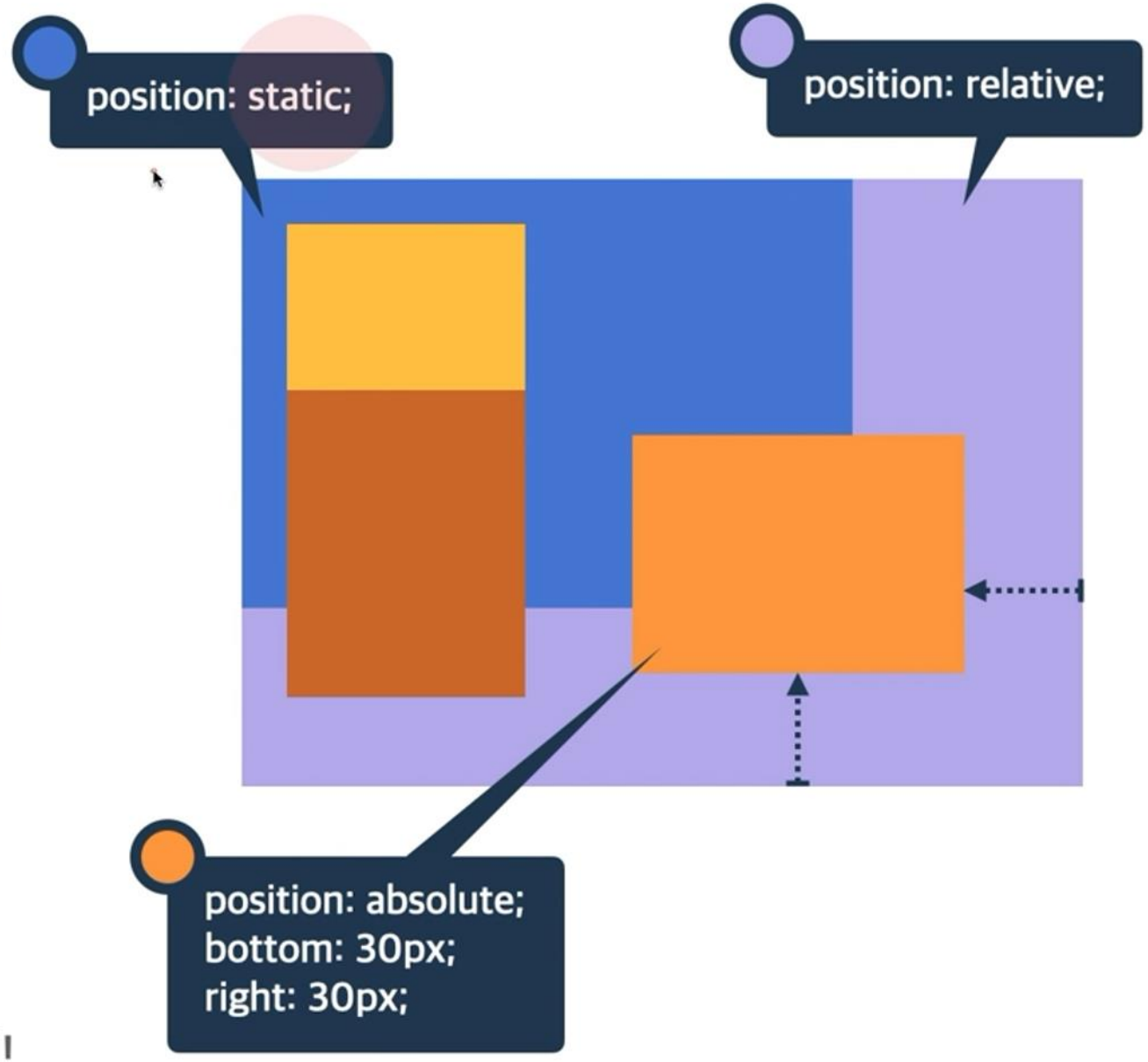
absolute 위치 상 부모 요소를 기준으로 배치!



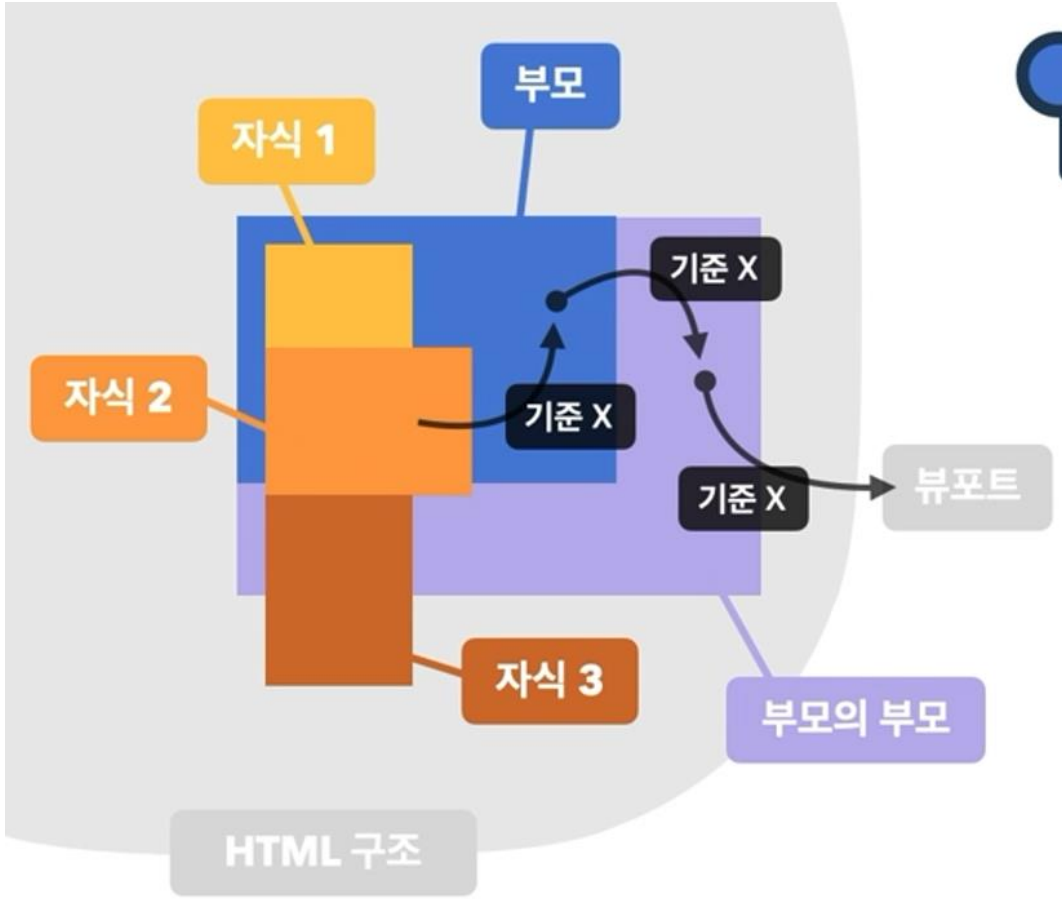


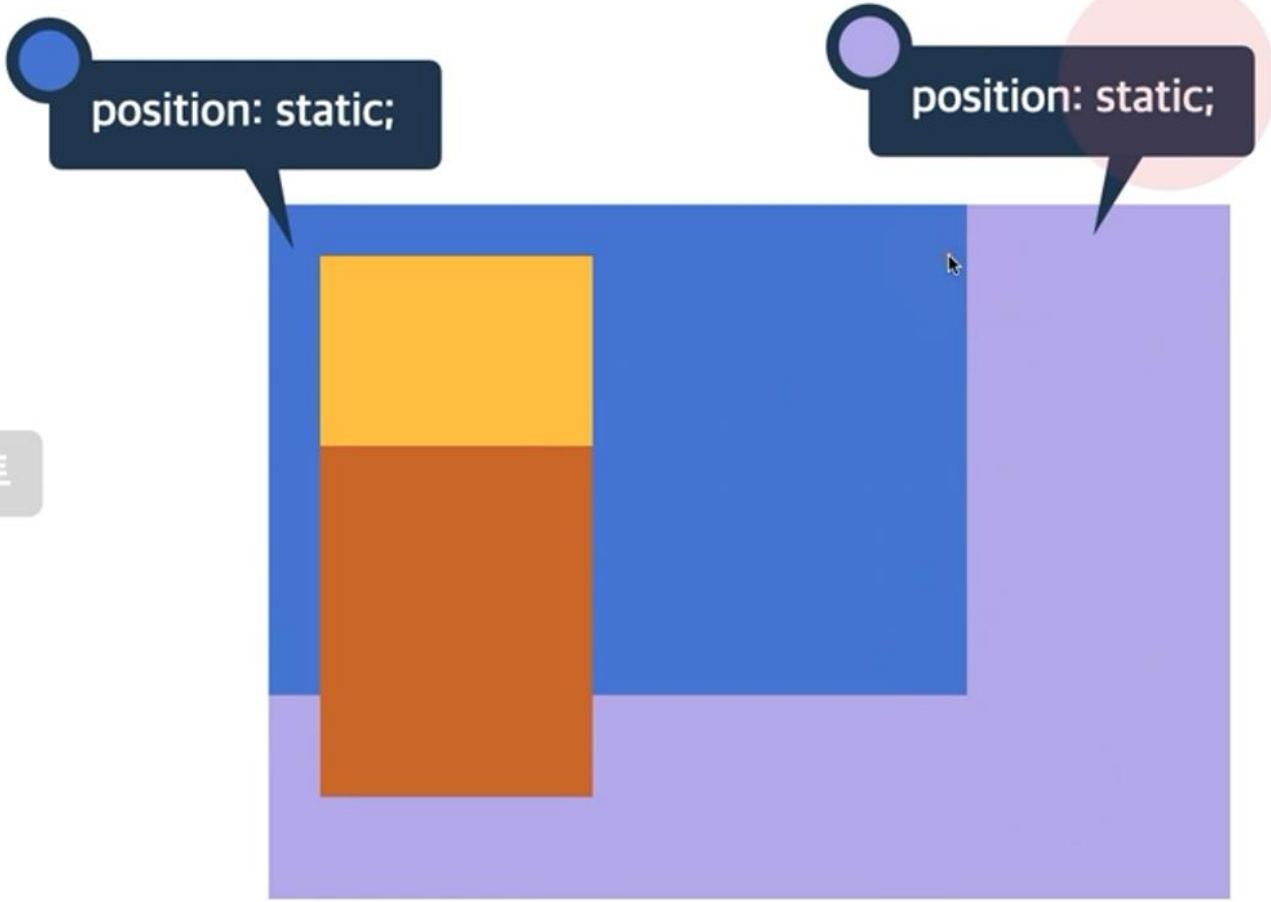
absolute 위치 상 부모 요소를 기준으로 배치!





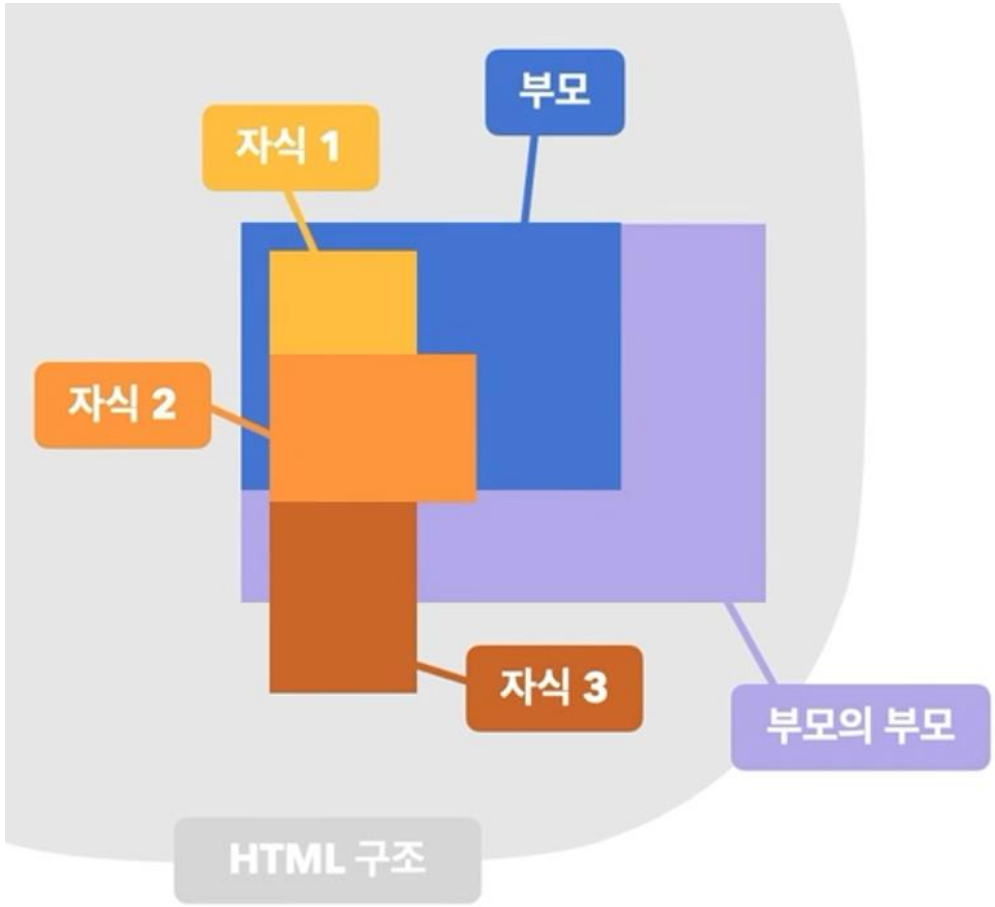
absolute 위치 상 부모 요소를 기준으로 배치!

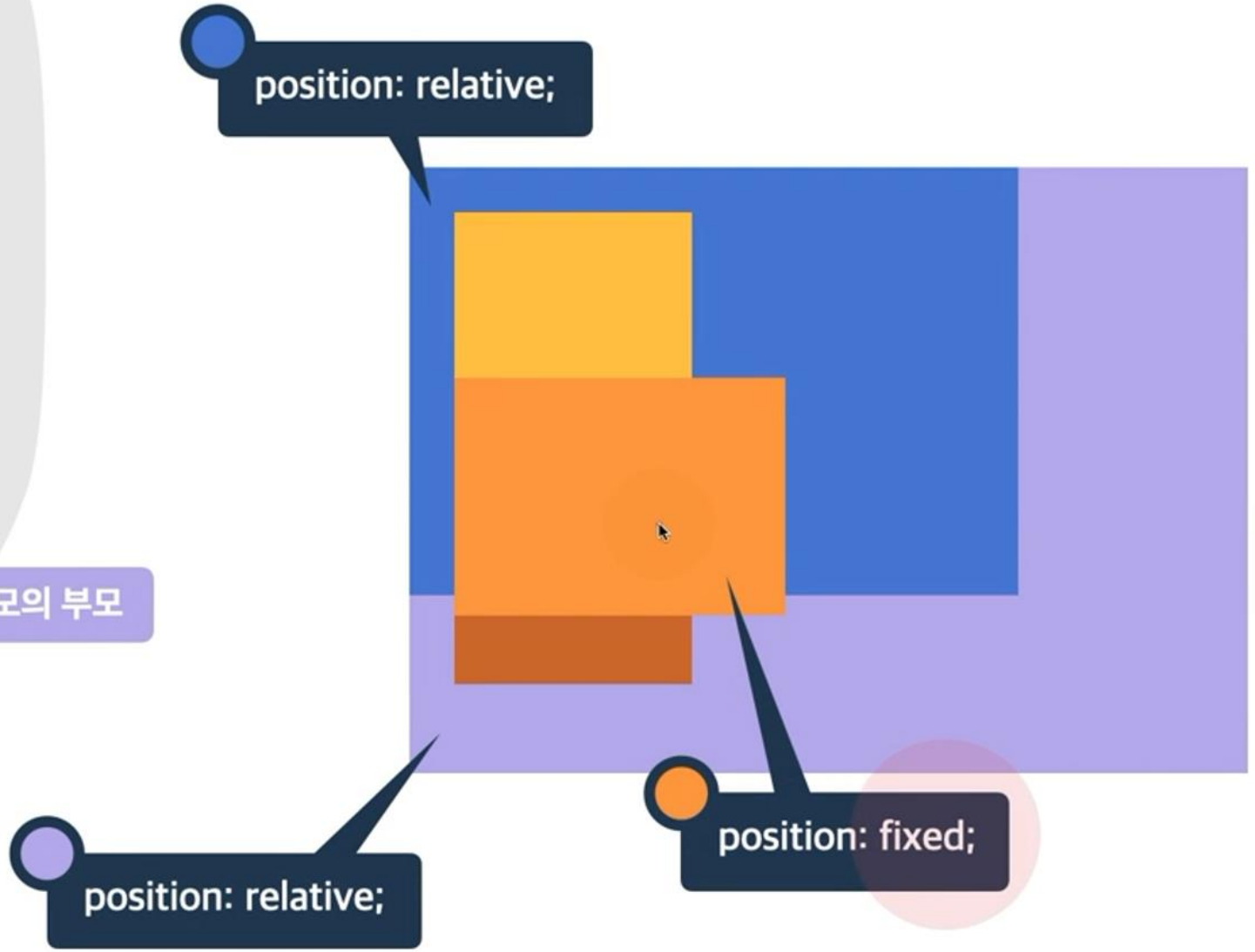




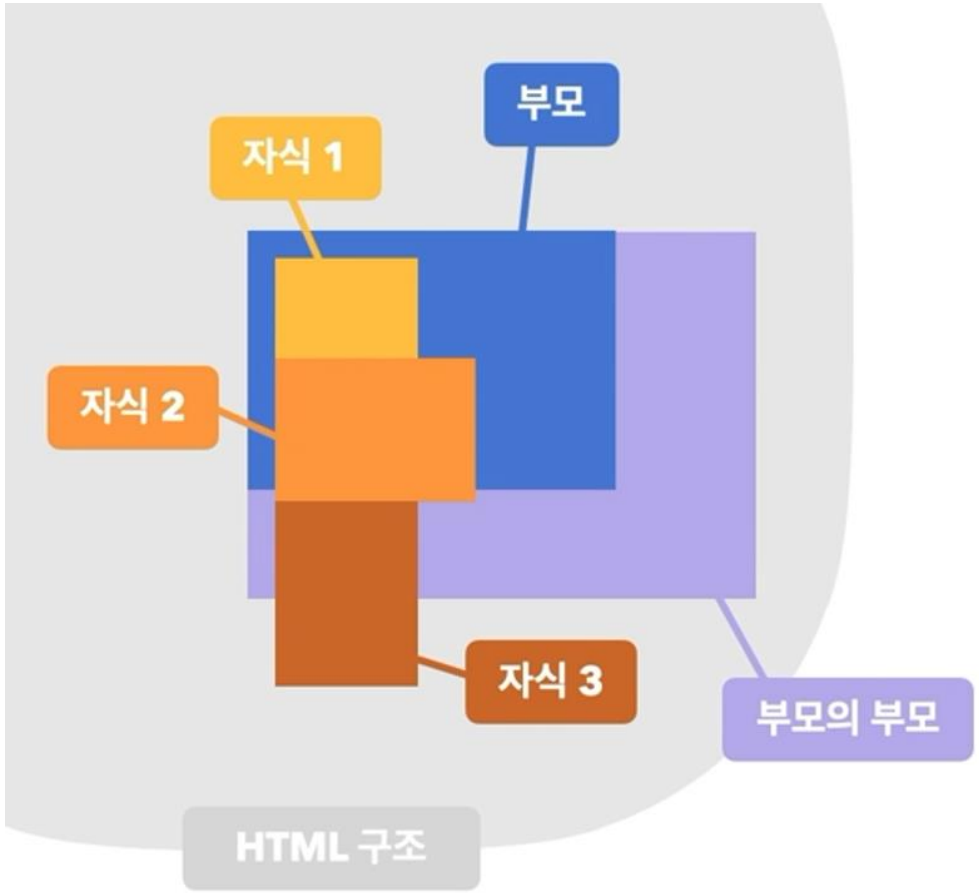
absolute 위치 상 부모 요소를 기준으로 배치!

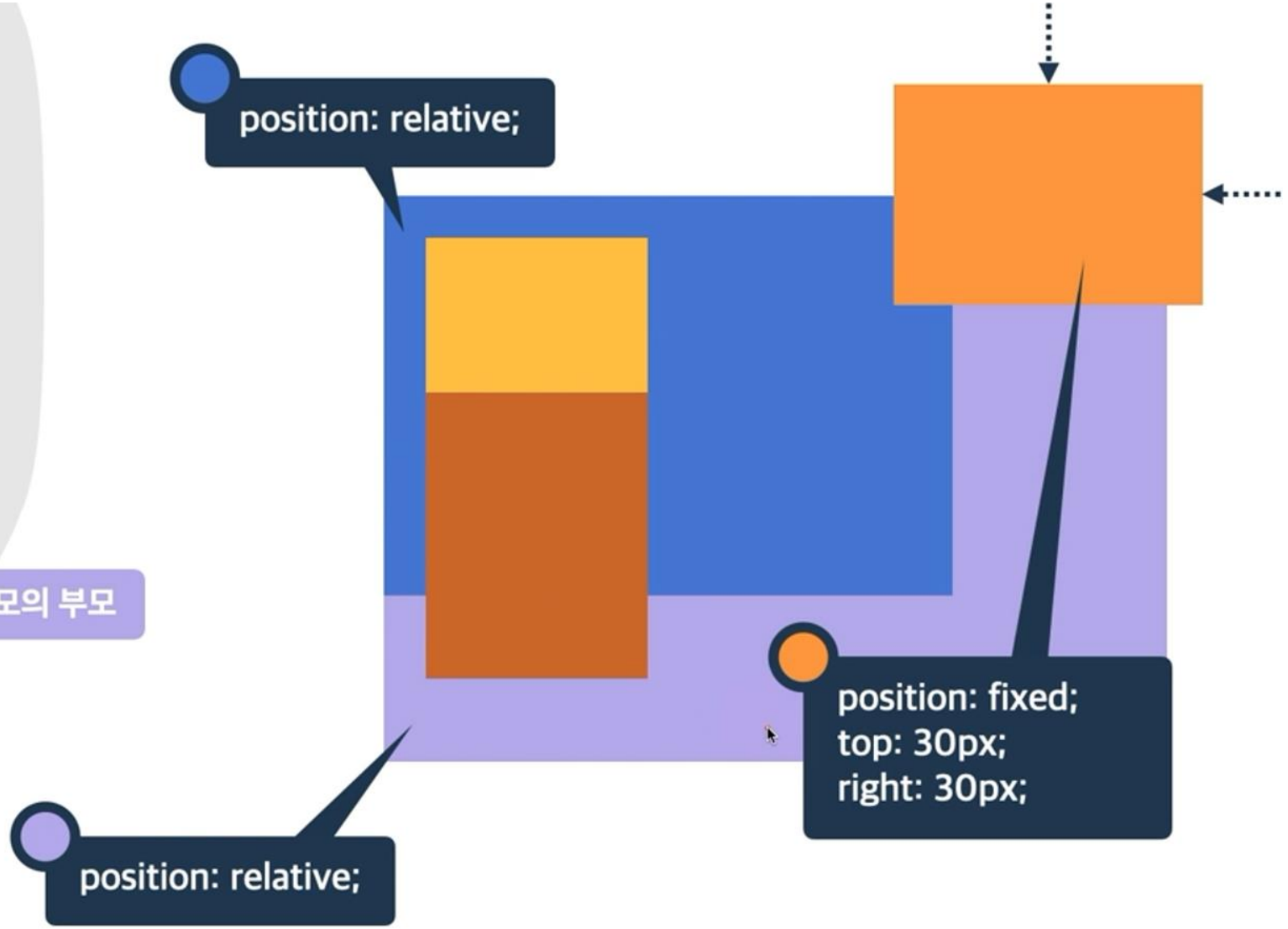
fixed





fixed 뷰포트(브라우저)를 기준으로 배치!





fixed 뷰포트(브라우저)를 기준으로 배치!

요소 쌓임 순서(Stack order)

어떤 요소가 사용자와 더 가깝게 있는지(위에 쌓이는지) 결정

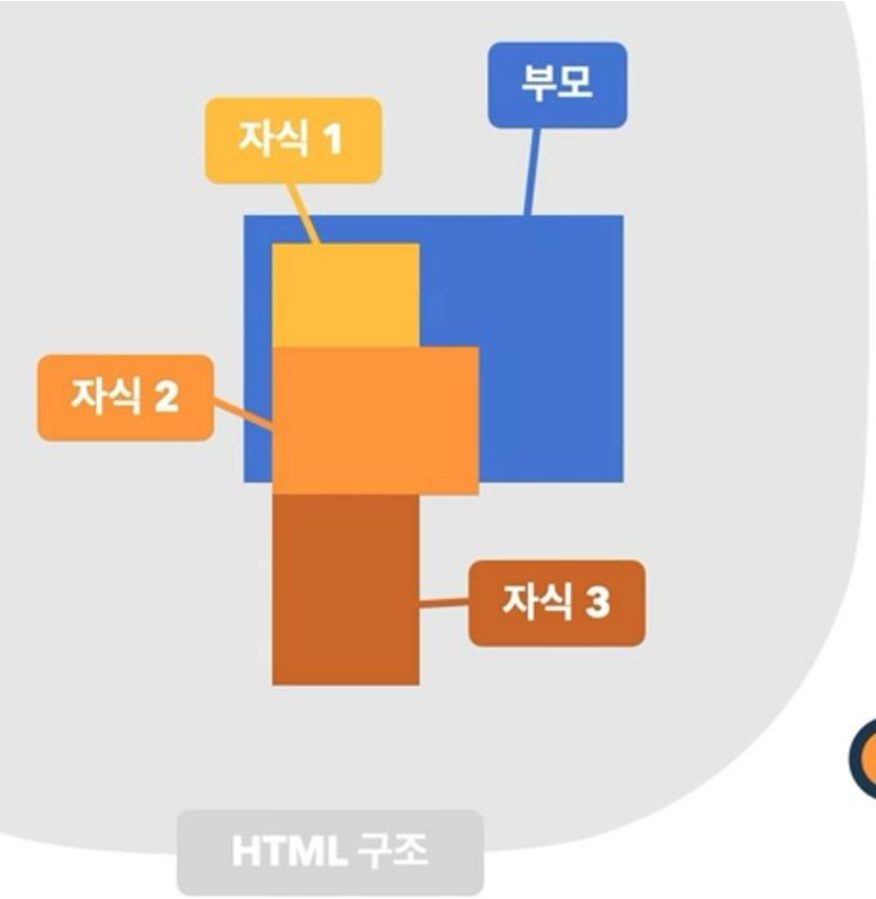
1. 요소에 position 속성의 값이 있는 경우 위에 쌓임.(기본값 static 제외)
2. 1번 조건이 같은 경우, z-index 속성의 숫자 값이 높을 수록 위에 쌓임.
3. 1번과 2번 조건까지 같은 경우, HTML의 다음 구조일 수록 위에 쌓임.

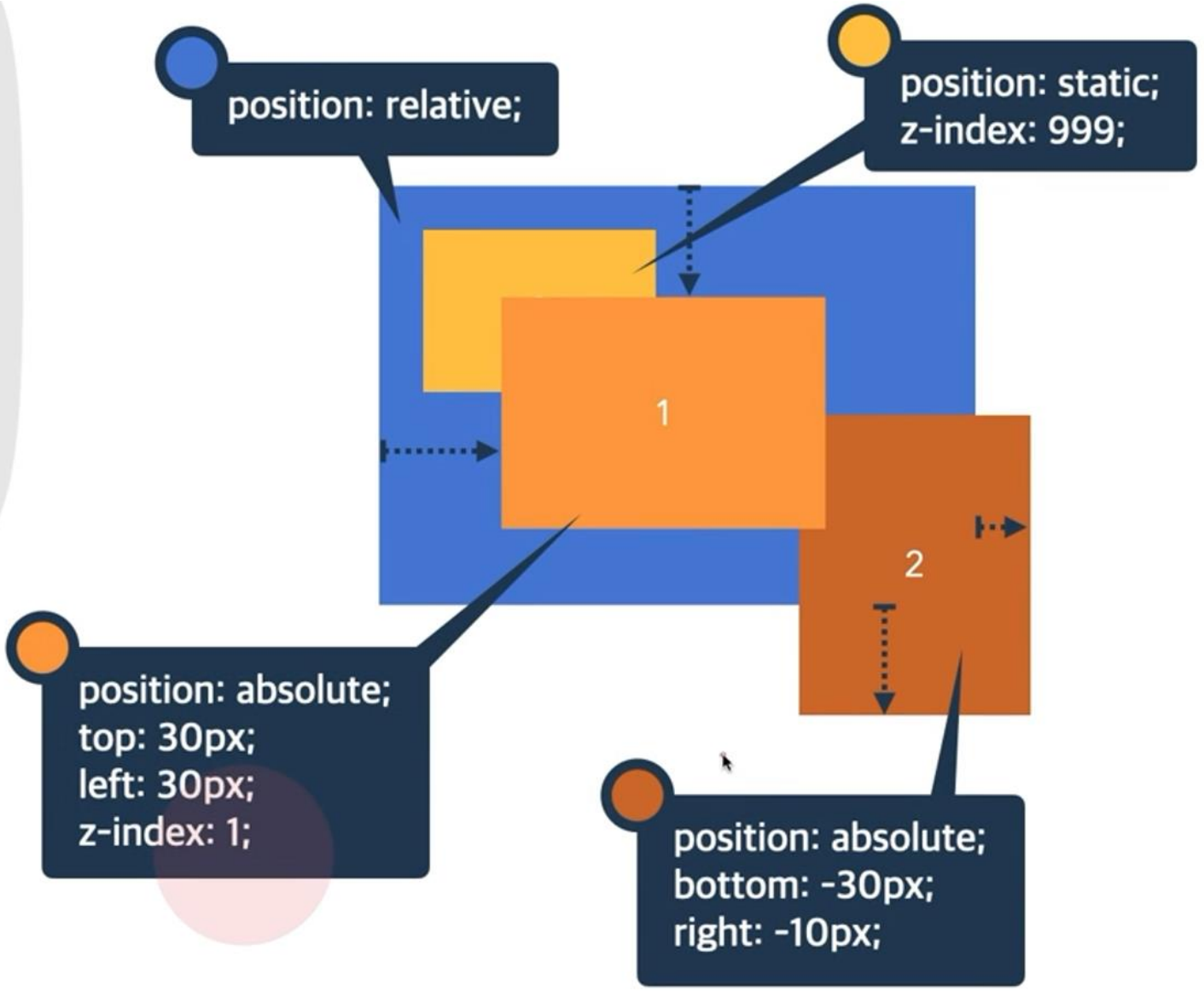
요소의 쌓임 정도를 지정

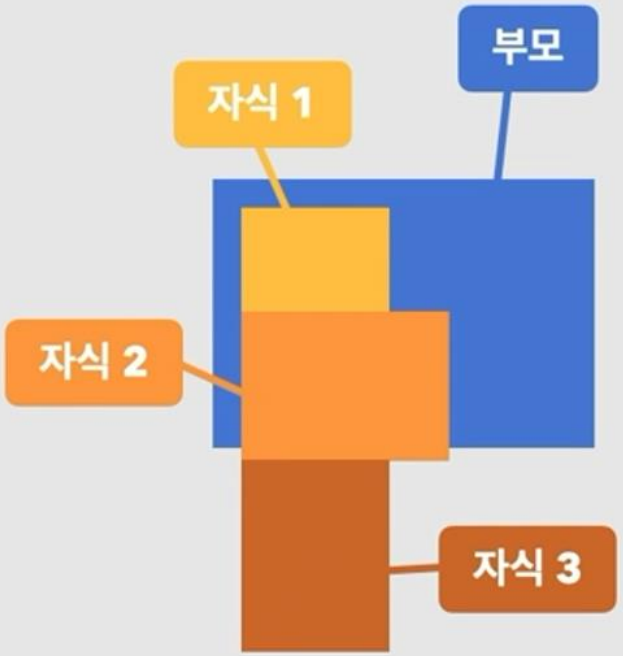
z-index

auto 부모 요소와 동일한 쌓임 정도

숫자 숫자가 높을 수록 위에 쌓임


부모
자식 1
자식 2
자식 3
HTML 구조

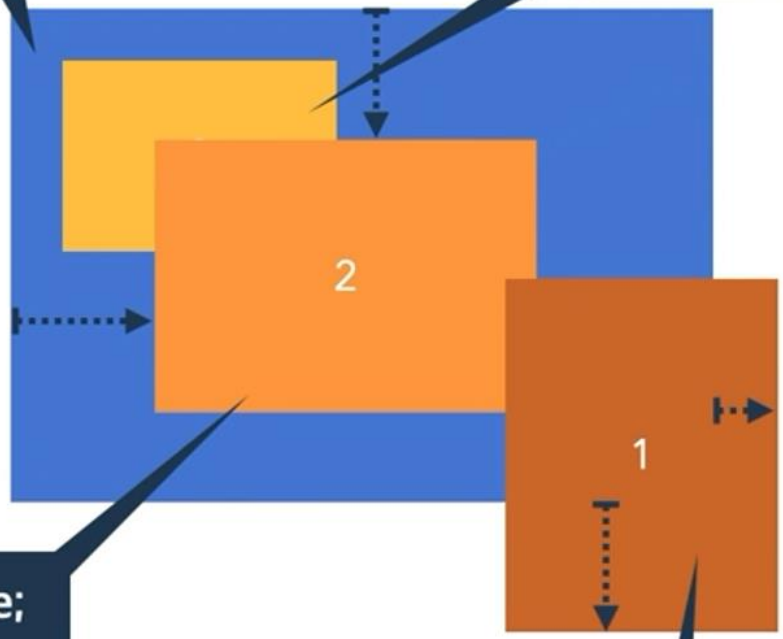

position: relative;
position: static;
z-index: 999;
1
2
position: absolute;
top: 30px;
left: 30px;
z-index: 1;
position: absolute;
bottom: -30px;
right: -10px;



HTML 구조

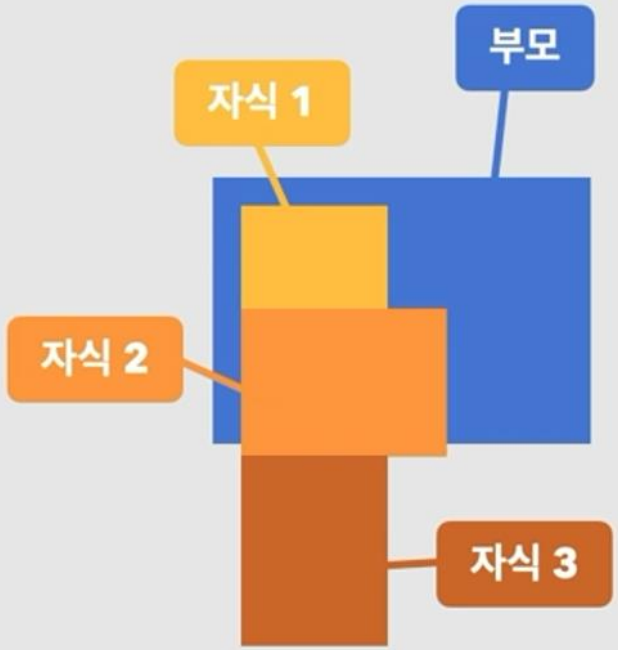
position: relative;

position: static;
z-index: 999;



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

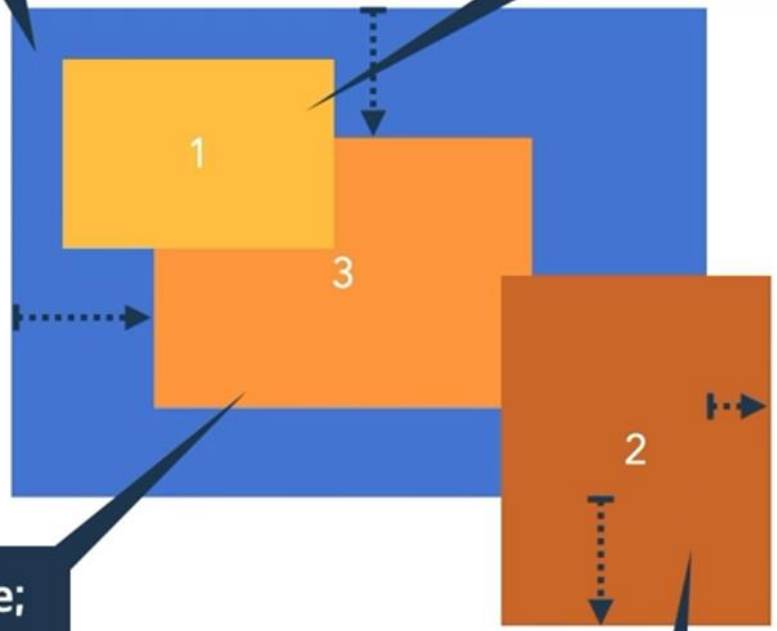
position: absolute;
bottom: -30px;
right: -10px;
z-index: 2;



HTML 구조

position: relative;

position: relative;
z-index: 999;



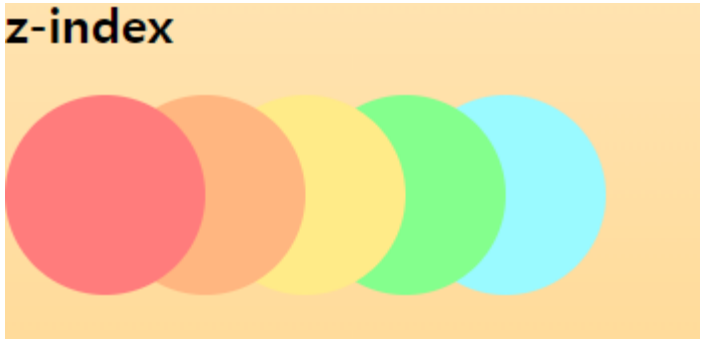
position: absolute;
top: 30px;
left: 30px;
z-index: 1;

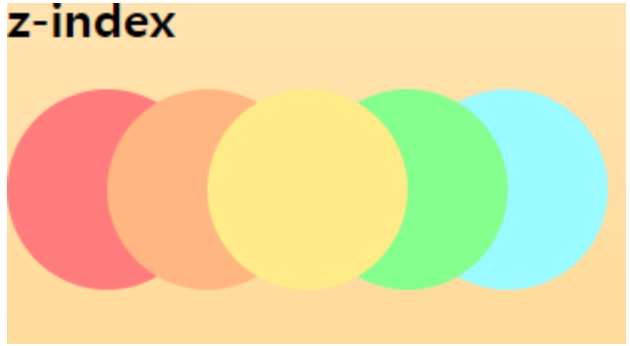
position: absolute;
bottom: -30px;
right: -10px;
z-index: 2;

z-index

```
.parent {
        position: relative;
}

.circle {
        width: 100px;
        height: 100px;
        border-radius: 50%;
        position: absolute;
}

.circle1 {
        background-color: #9afaff;
        top: 0;
        left: 200px;
}

.circle2 {
        background-color: #84ff8d;
        top: 0;
        left: 150px;
}

.circle3 {
        background-color: #ffeb88;
        top: 0;
        left: 100px;
}

.circle4 {
        background-color: #ffb680;
        top: 0;
        left: 50px;
}

.circle5 {
        top: 0;
        left: 0;
        background-color: #ff7c7c;
}
```

attr3.css





attr3.html

```
<div class="parent">
        <div class="circle circle1"></div>
        <div class="circle circle2"></div>
        <div class="circle circle3"></div>
        <div class="circle circle4"></div>
        <div class="circle circle5"></div>
</div>
```

과제. position sticky 에 대해 알아오기!

CSS 복잡한 속성 - transform

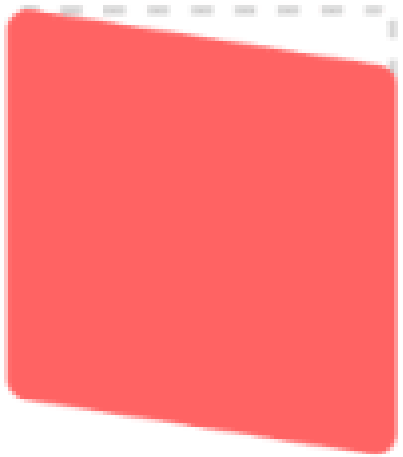
요소에 이동(translate), 회전(rotate), 확대축소(scale), 비틀기(skew) 효과를 부여하기 위한 함수를 제공

- **skew** : 기울기
- **scale** : 확대
- **rotate** : 회전
- **translate** : 이동

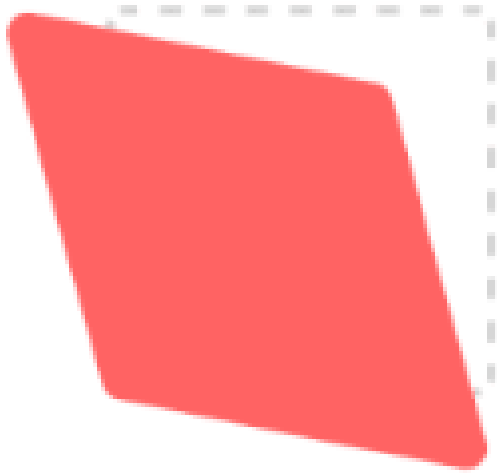
CSS 복잡한 속성 - transform

- skew : 기울기

transform: skewX(10deg) ;



transform: skewY(10deg) ;



transform: skew(10deg, 10deg) ;

사진 출처: https://dev.to/kunaal438/css-transform-complete-guide-on-css-transform-everything-you-need-for-good-developer-841

CSS 복잡한 속성 - transform

- rotate : 회전

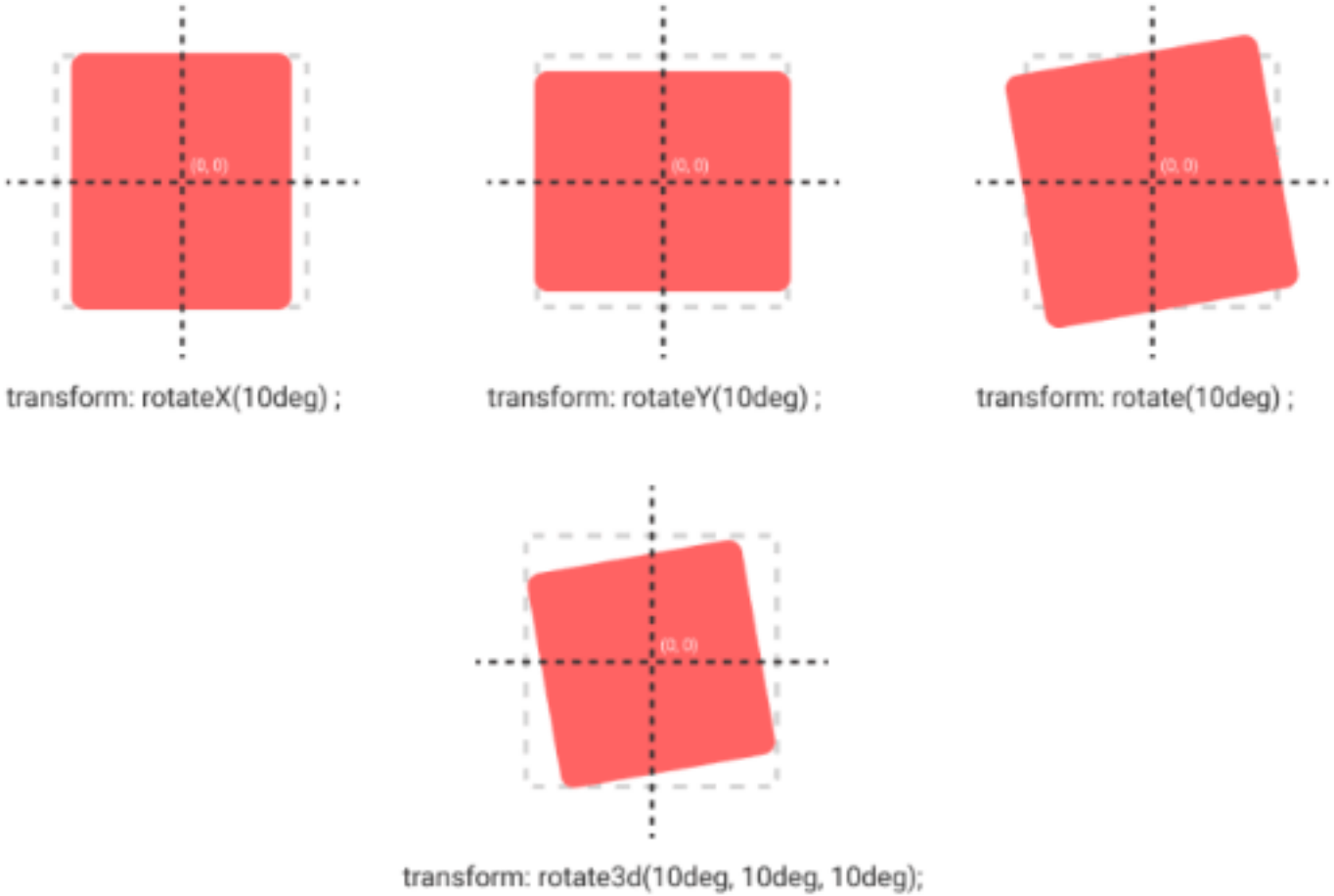


사진 출처: https://dev.to/kunaal438/css-transform-complete-guide-on-css-transform-everything-you-need-for-good-developer-841

CSS 복잡한 속성 - transform

- translate : 이동

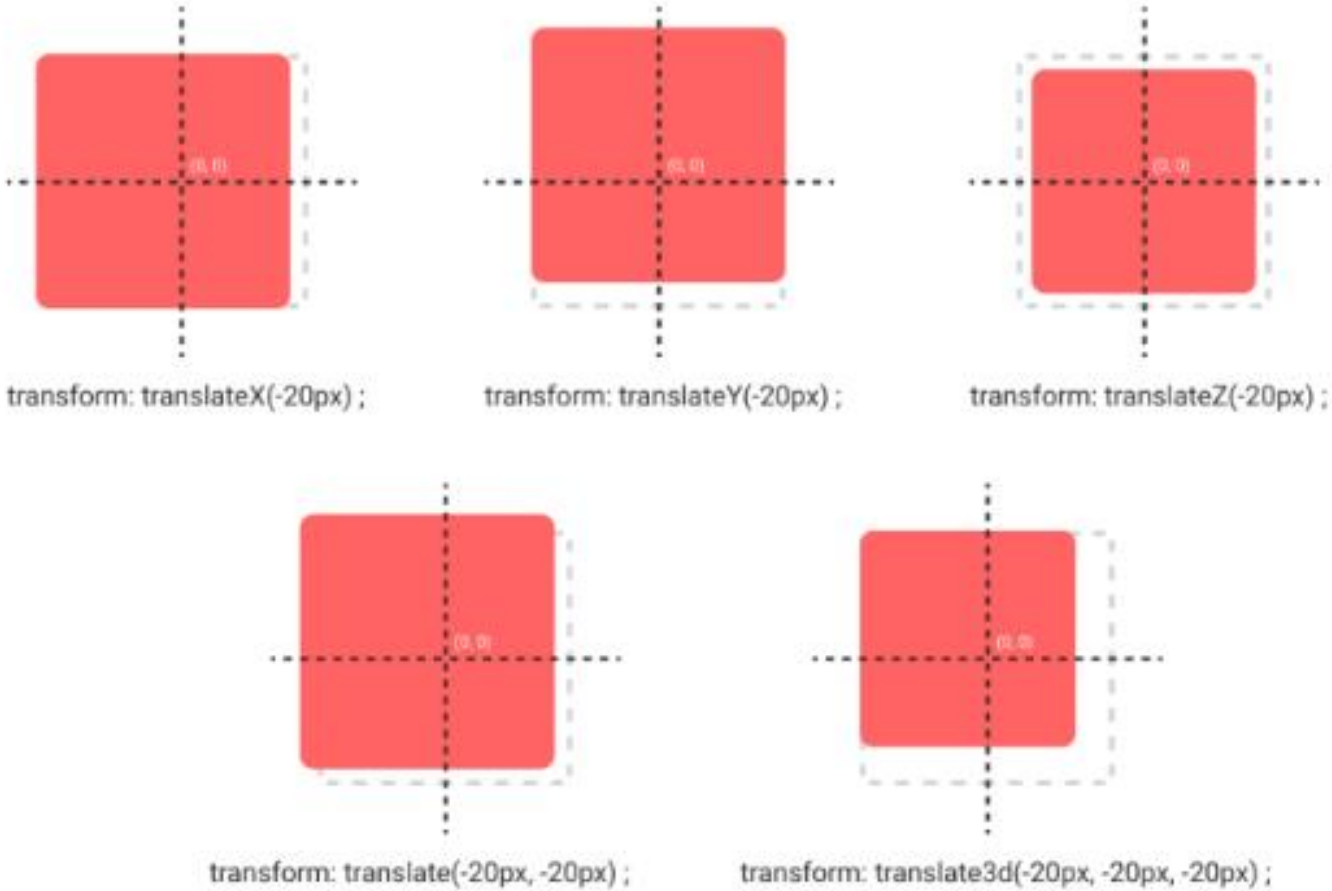
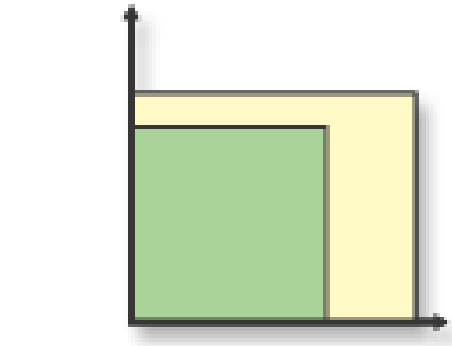
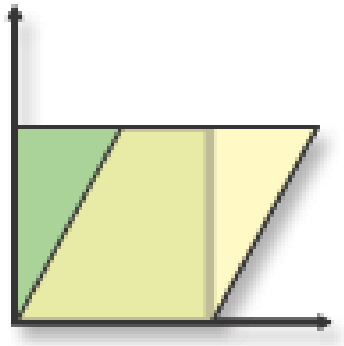
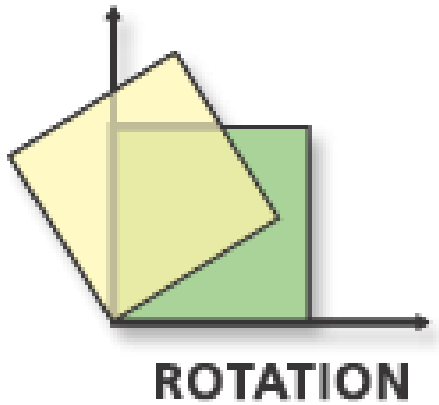


사진 출처: https://dev.to/kunaal438/css-transform-complete-guide-on-css-transform-everything-you-need-for-good-developer-841

CSS 복잡한 속성 - transform







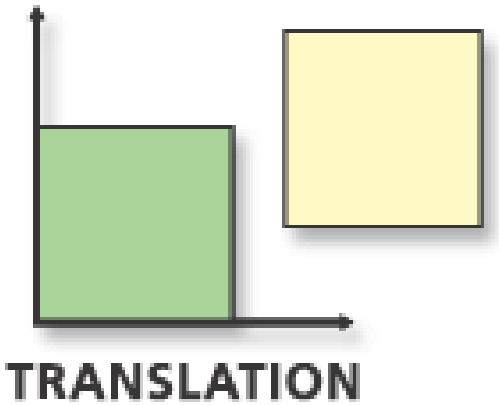
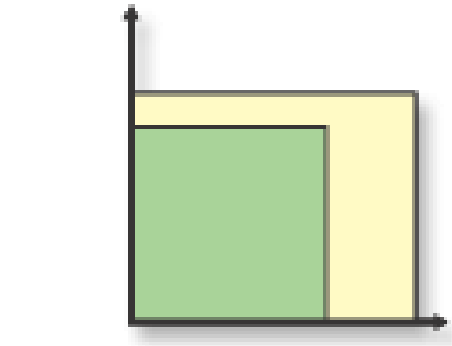
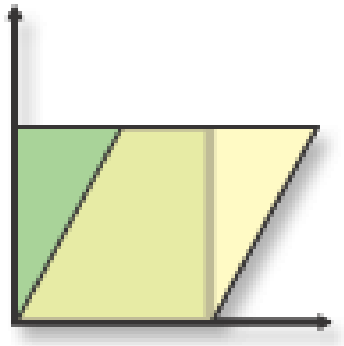
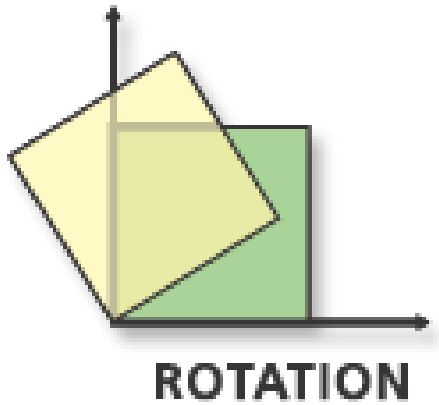


사진 출처: https://desktop.arcgis.com/en/arcmap/10.6/tools/coverage-toolbox/how-transform-works.htm

CSS 복잡한 속성 - transform







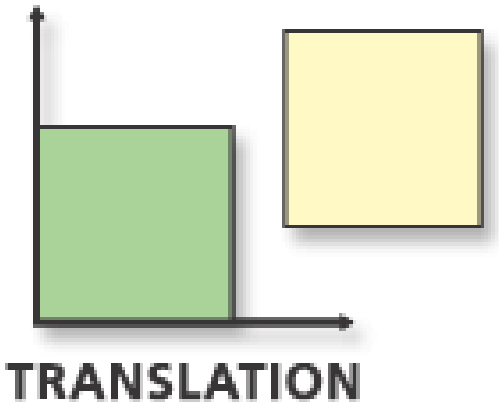


사진 출처: https://desktop.arcgis.com/en/arcmap/10.6/tools/coverage-toolbox/how-transform-works.htm

요소의 변환 효과

transform: 변환함수1 변환함수2 변환함수3 ... ;

transform: 원근법 이동 크기 회전 기울임;

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 효과를 사용할 속성 이름을 지정

transition-property

all 모든 속성에 적용

속성이름 전환 효과를 사용할 속성 이름 명시

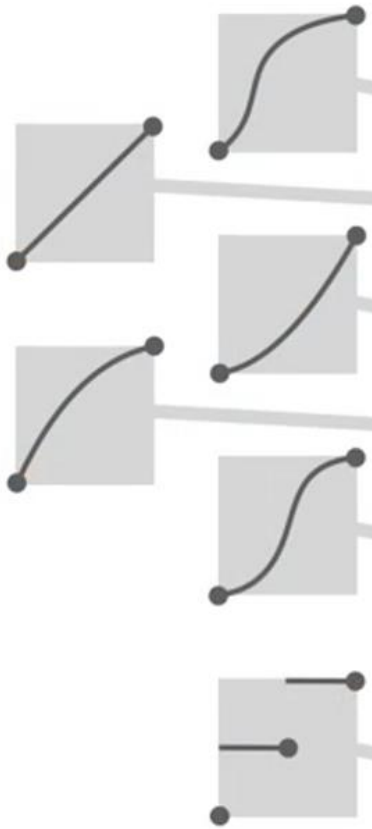
전환 효과의 지속시간을 지정

transition-duration

0s	전환 효과 없음
시간	지속시간(s)을 지정

전환 효과의 타이밍(Easing) 함수를 지정

transition-timing-function



- **ease** 느리게 - 빠르게 - 느리게 = cubic-bezier(0.25, 0.1, 0.25, 1)
- **linear** 일정하게 = cubic-bezier(0, 0, 1, 1)
- **ease-in** 느리게 - 빠르게 = cubic-bezier(0.42, 0, 1, 1)
- **ease-out** 빠르게 - 느리게 = cubic-bezier(0, 0, 0.58, 1)
- **ease-in-out** 느리게 - 빠르게 - 느리게 = cubic-bezier(0.42, 0, 0.58, 1)
- **cubic-bezier(n,n,n,n)** 자신만의 값을 정의(0~1)
- **steps(n)** n번 분할된 애니메이션

전환 효과가 몇 초 뒤에 시작할지 대기시간을 지정

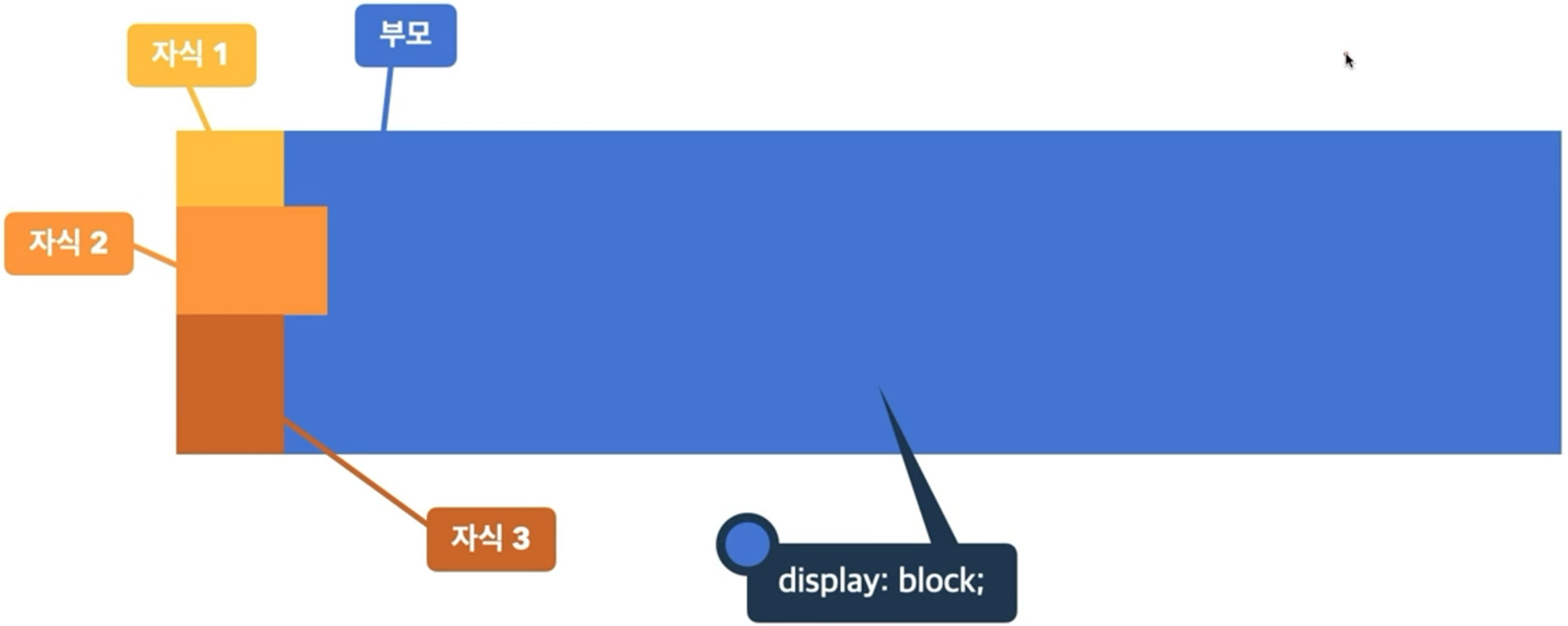
transition-delay

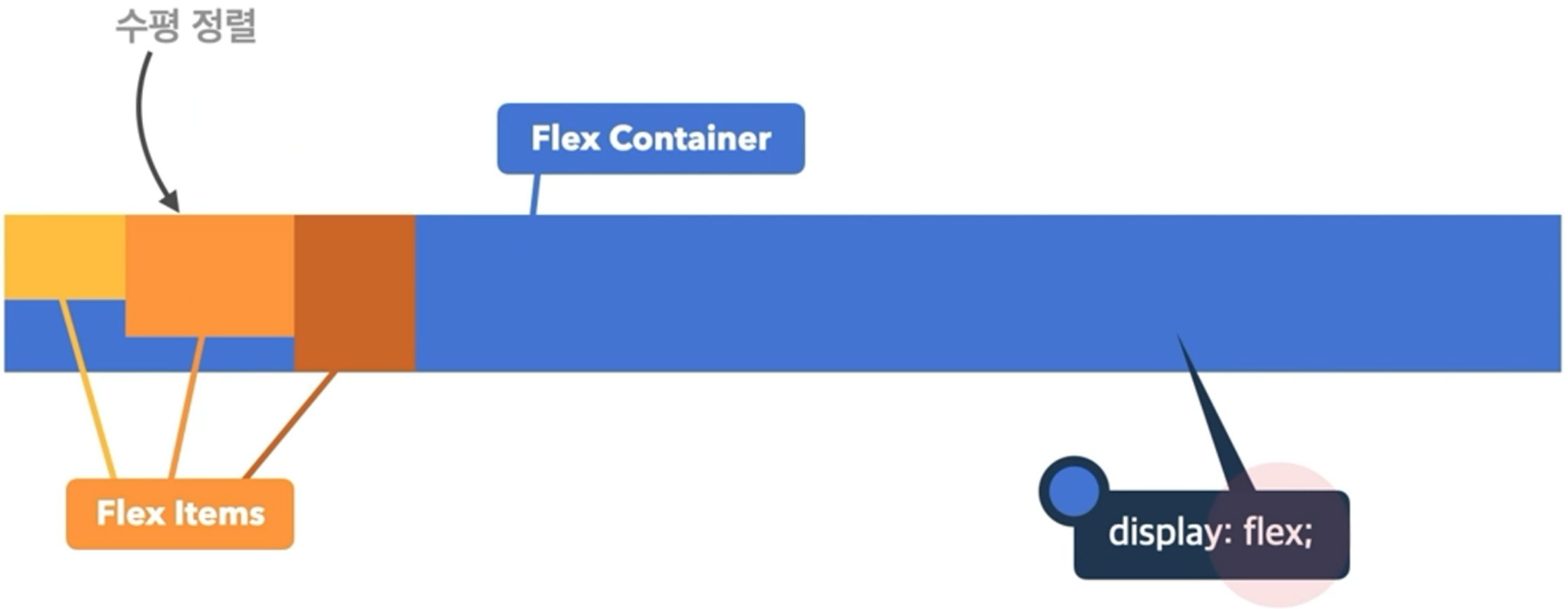
0s	대기시간 없음
시간	대기시간(s)을 지정

CSS 복잡한 속성 - display

- inline / block
- inline-block
- flex

Flex


부모
자식 1
자식 2
자식 3
display: block;

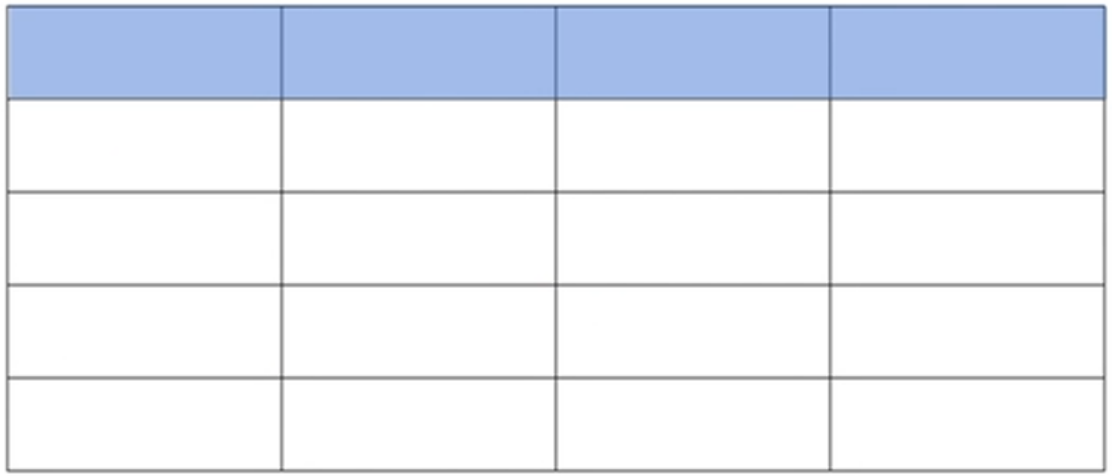

수평 정렬
Flex Container
Flex Items
display: flex;

주 축을 설정

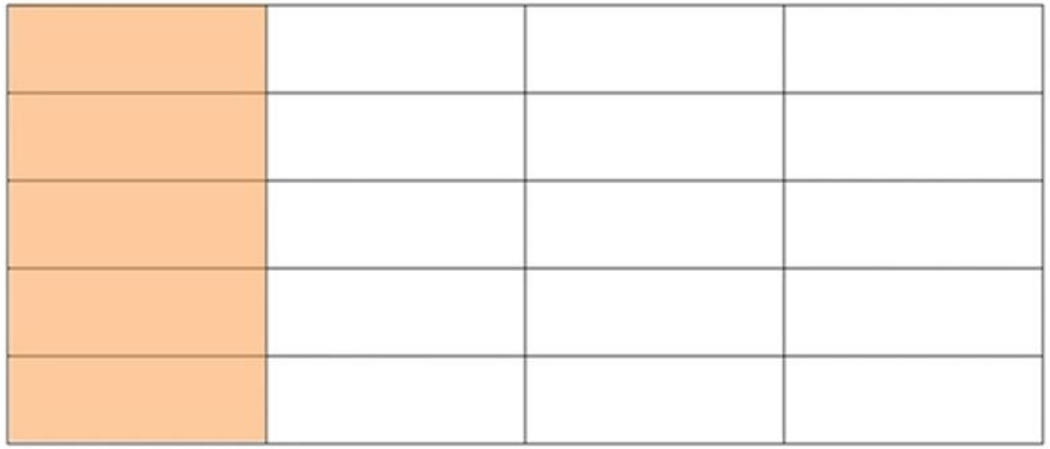
flex-direction

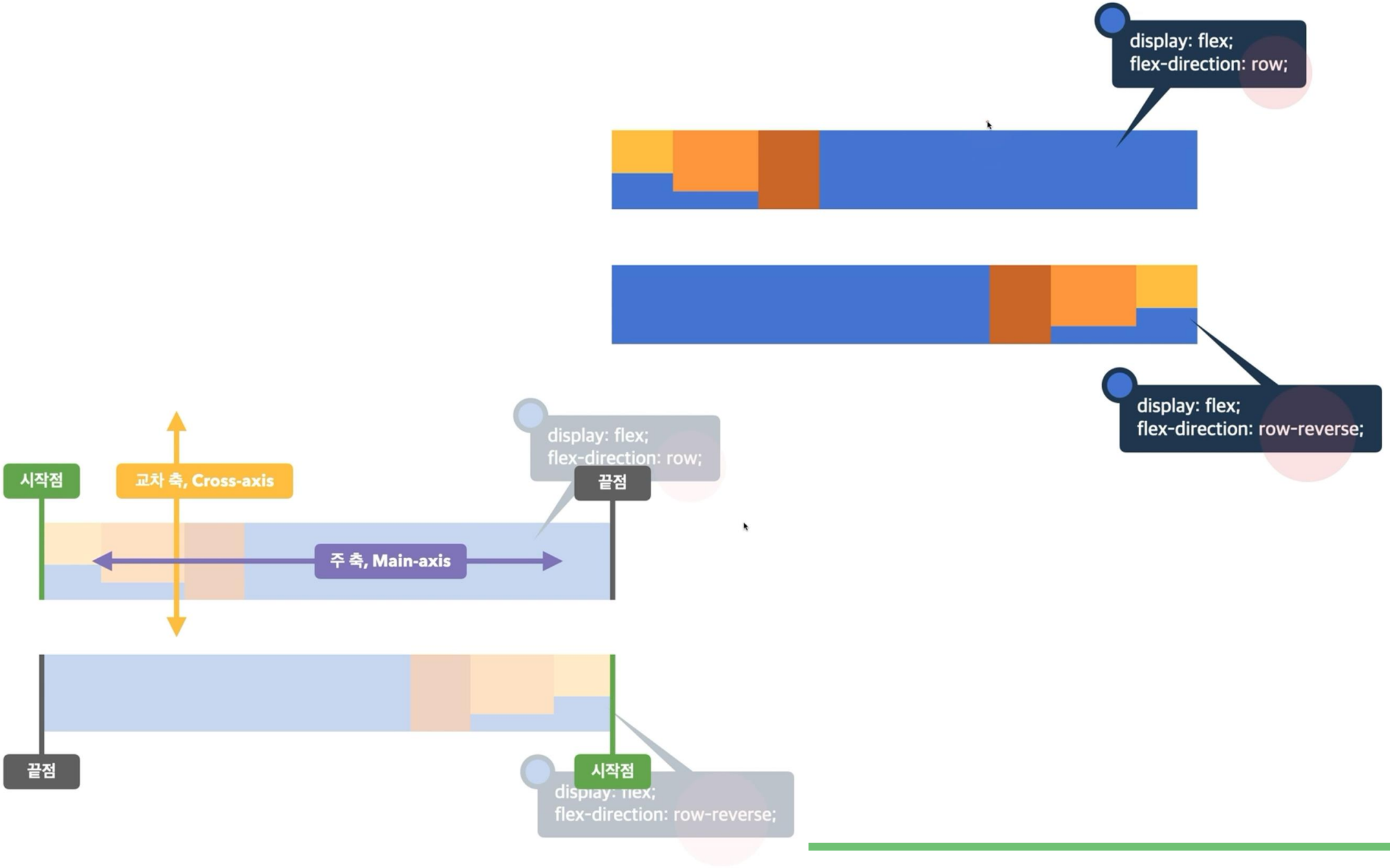
- `row` 행 축 (좌 => 우)
- `row-reverse` 행 축 (우 => 좌)
- `column` 열 축 (위 => 아래)
- `column-reverse` 열 축 (아래 => 위)

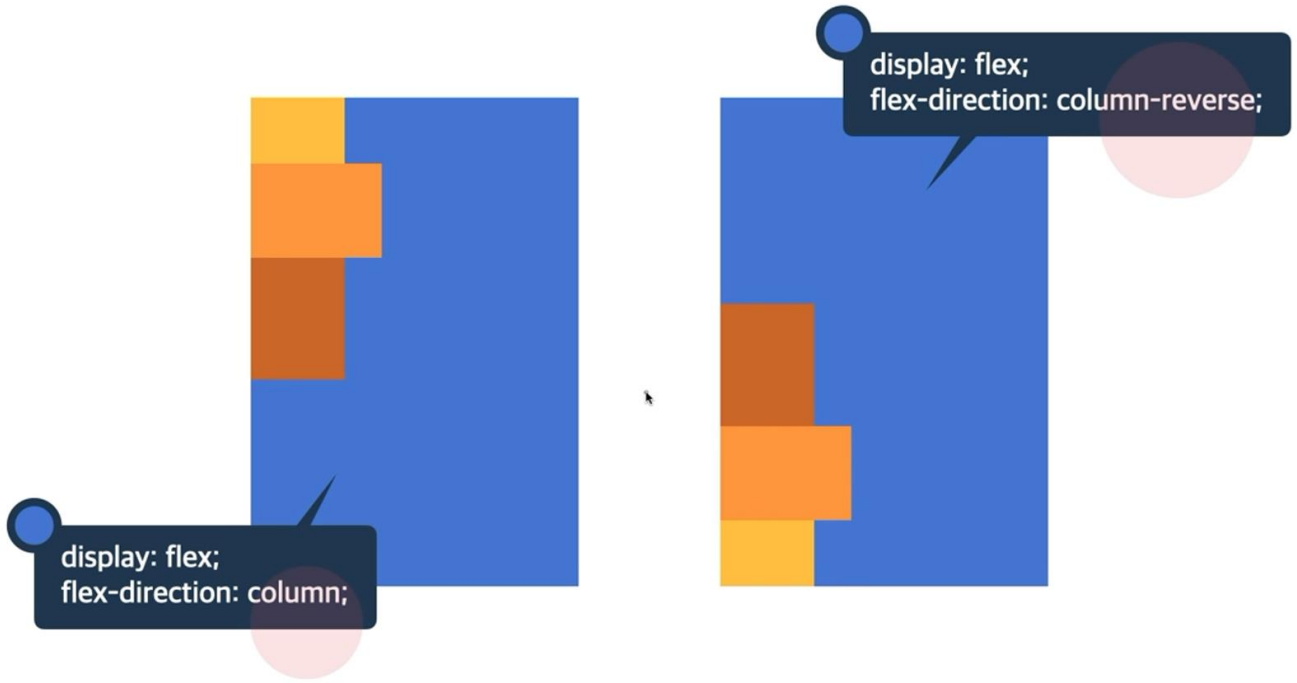
행, Row

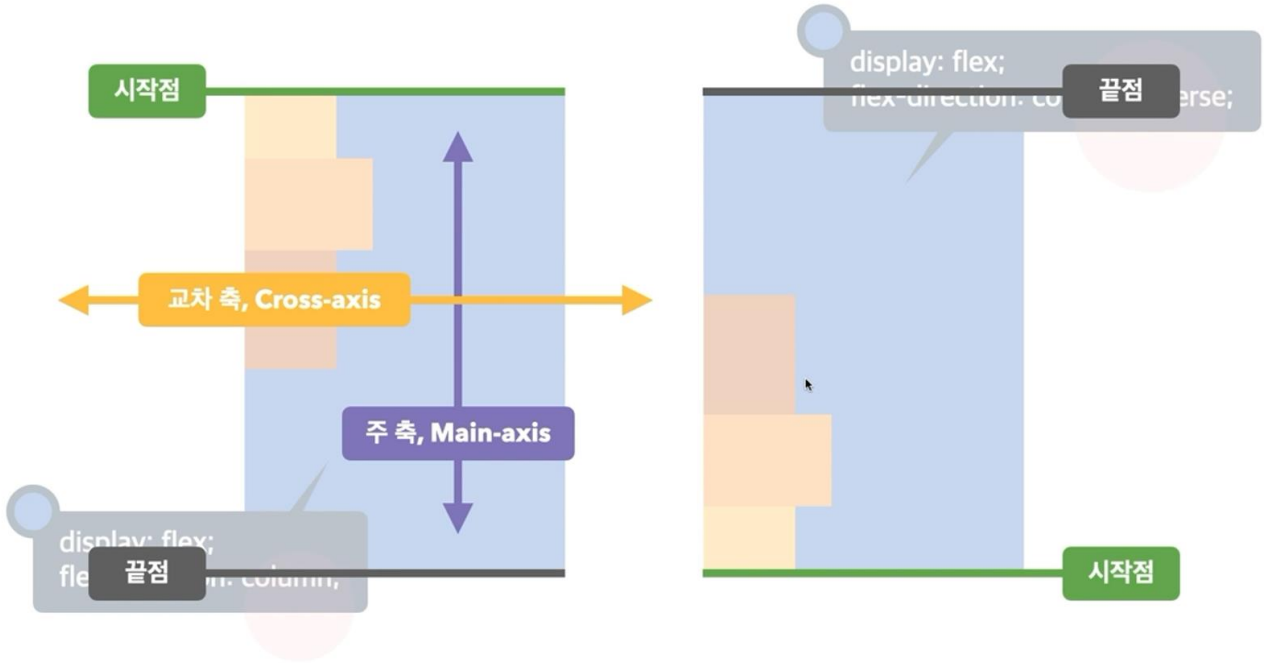


열, Column




display: flex;
flex-direction: row;
display: flex;
flex-direction: row-reverse;
시작점
교차 축, Cross-axis
주 축, Main-axis
끝점
display: flex;
flex-direction: row;
끝점
시작점
display: flex;
flex-direction: row-reverse;


display: flex;
flex-direction: column-reverse;
display: flex;
flex-direction: column;


시작점
교차 축, Cross-axis
주 축, Main-axis
끝점
display: flex;
끝점
시작점

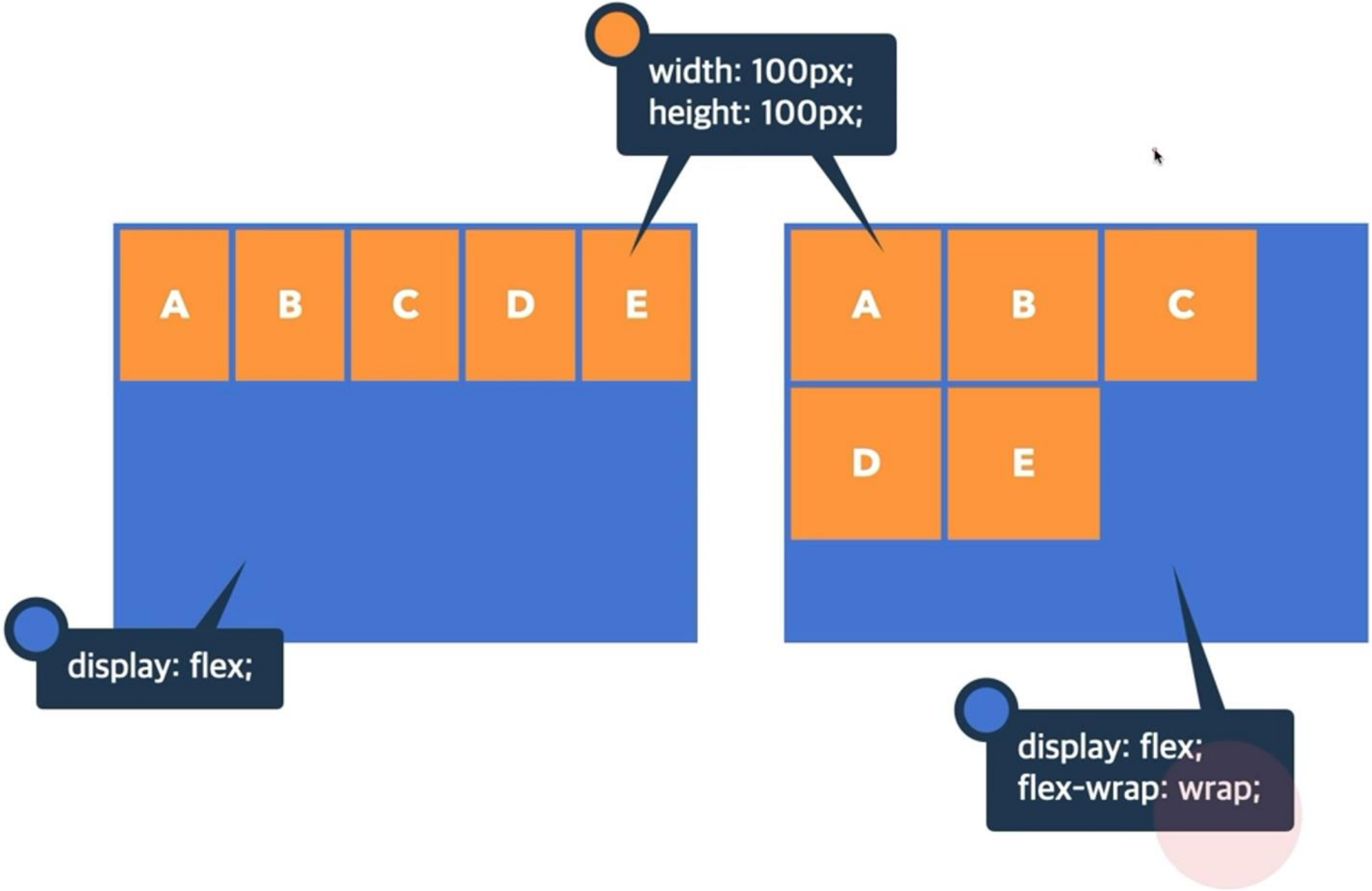
Flex Items 묶음(줄 바꿈) 여부

flex-wrap

nowrap 묶음(줄 바꿈) 없음

wrap 여러 줄로 묶음

wrap-reverse wrap의 반대 방향으로 묶음


width: 100px;
height: 100px;
A
B
C
D
E
display: flex;
A
B
C
D
E
display: flex;
flex-wrap: wrap;

주 축의 정렬 방법

justify-content

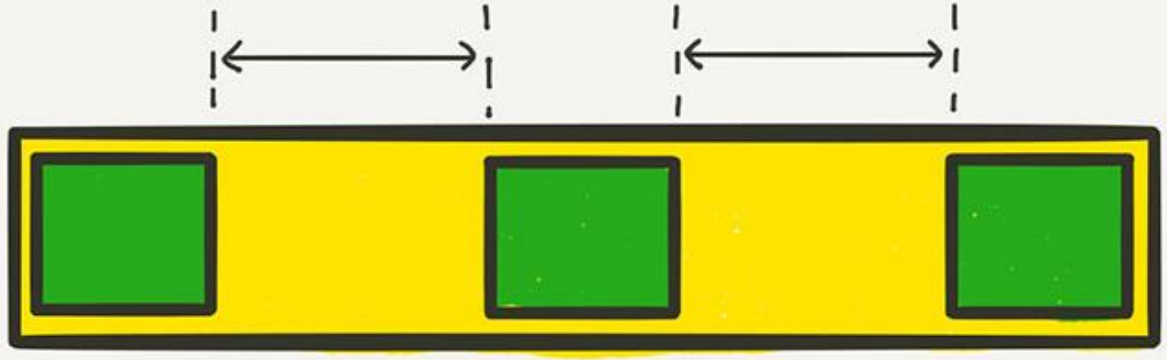
flex-start	Flex Items를 시작점으로 정렬
flex-end	Flex Items를 끝점으로 정렬
center	Flex Items를 가운데 정렬
space-between	각 Flex Item 사이를 균등하게 정렬
space-around	각 Flex Item의 외부 여백을 균등하게 정렬

display: flex;

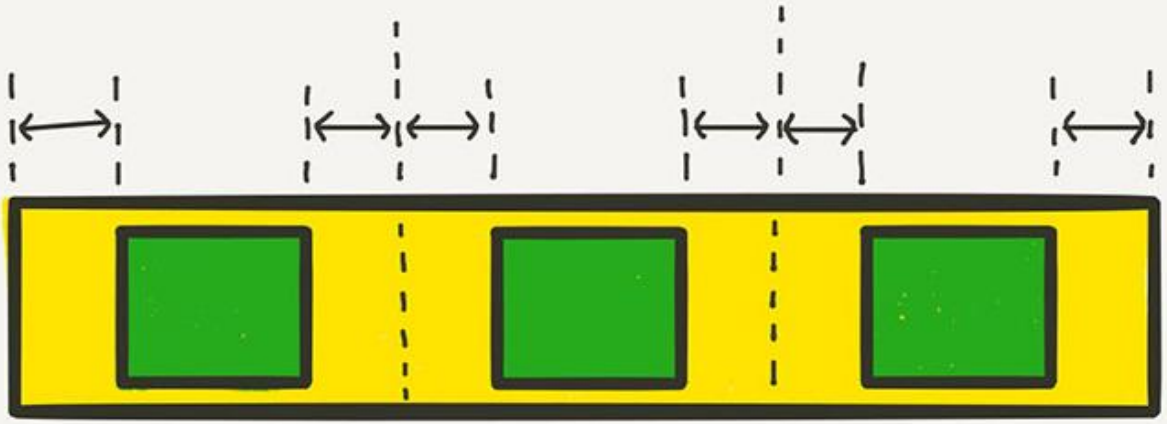
display: flex;
justify-content: flex-end;

display: flex;
justify-content: center;

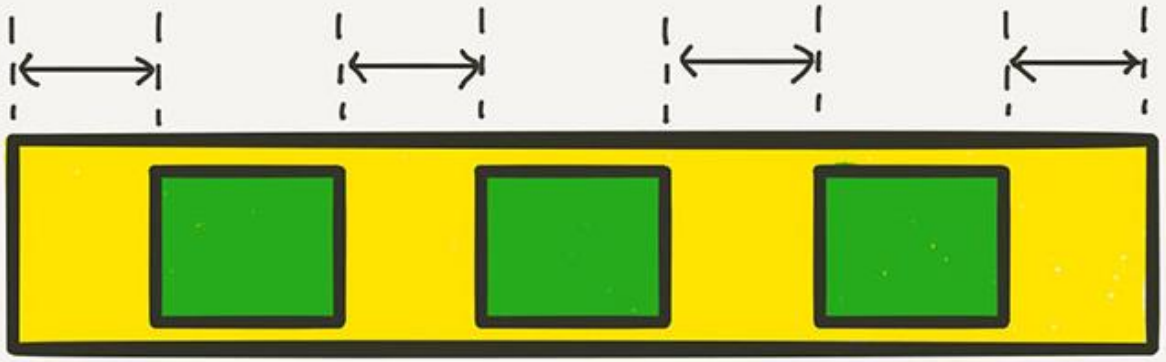
space-between



space-around



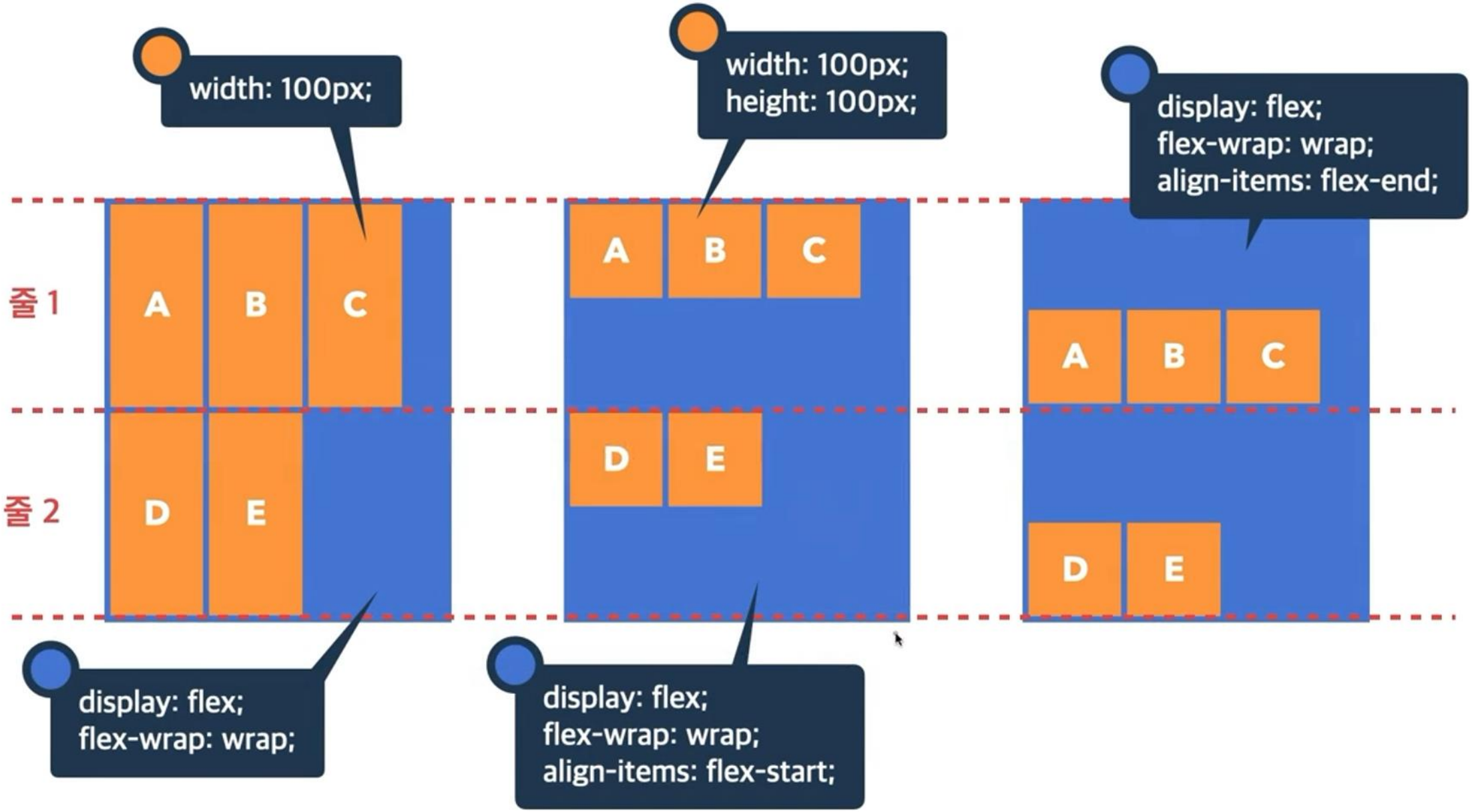
space-evenly

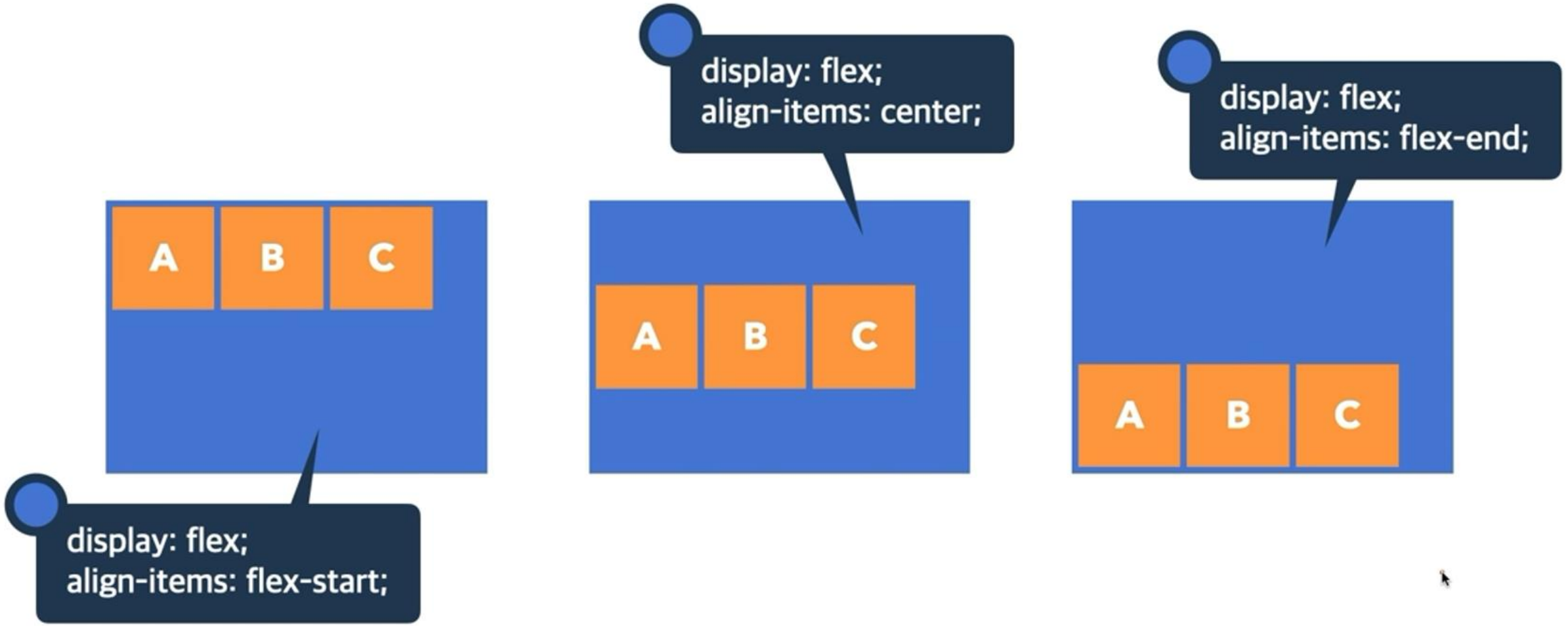


교차 축의 한 줄 정렬 방법

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

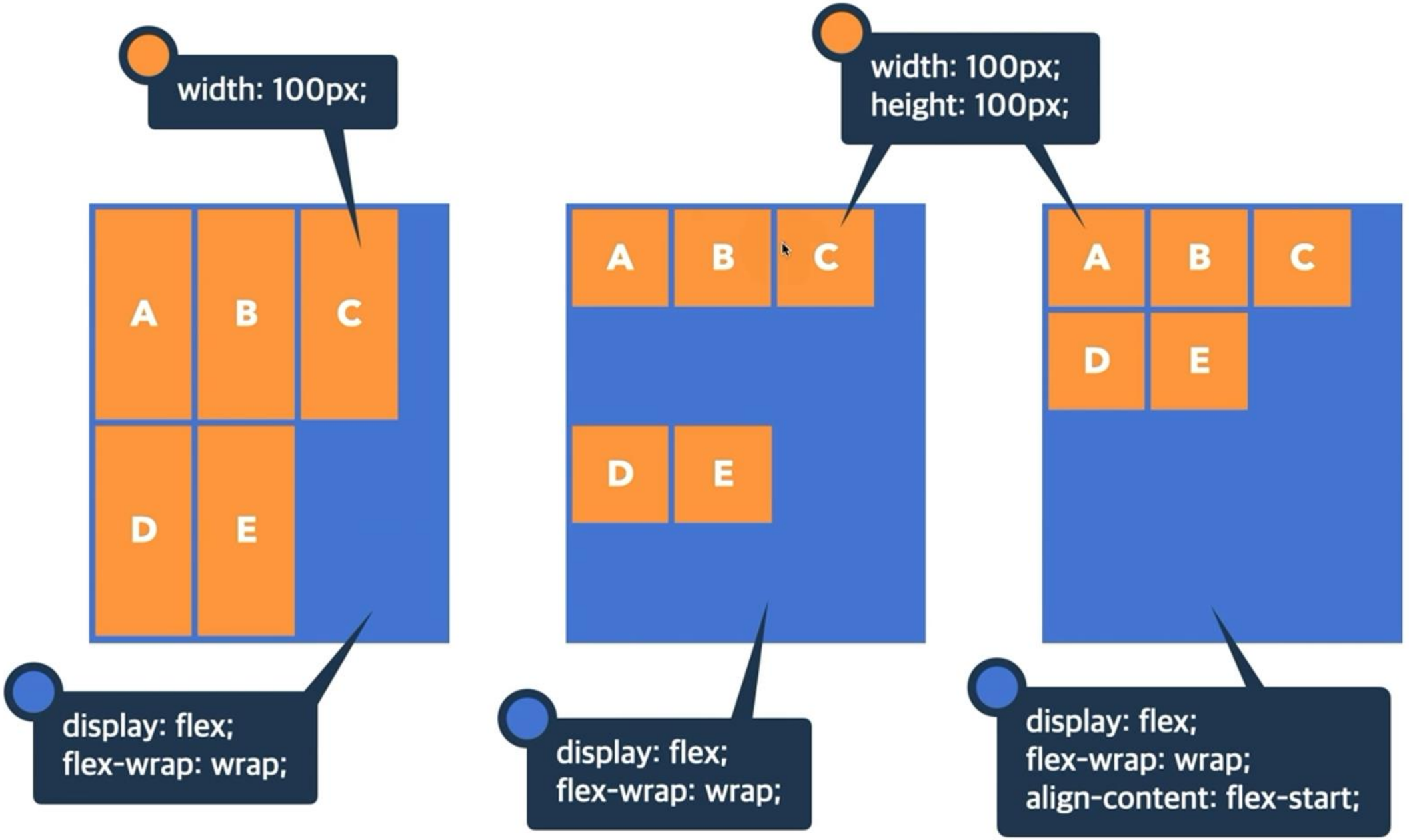

width: 100px;
줄 1
A
B
C
줄 2
D
E
display: flex;
flex-wrap: wrap;
width: 100px;
height: 100px;
A
B
C
D
E
display: flex;
flex-wrap: wrap;
align-items: flex-start;
display: flex;
flex-wrap: wrap;
align-items: flex-end;
A
B
C
D
E

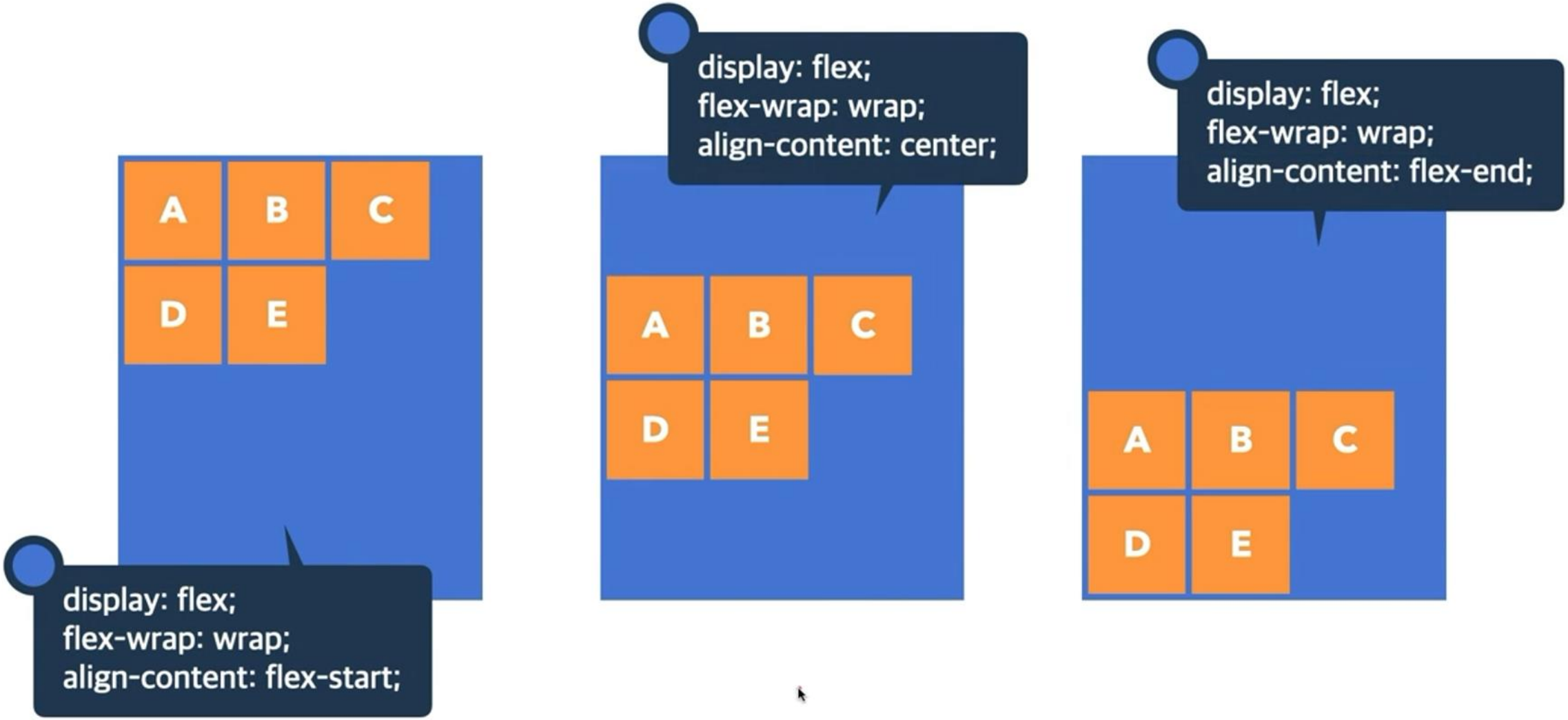

display: flex;
align-items: center;
display: flex;
align-items: flex-end;
A
B
C
A
B
C
A
B
C
display: flex;
align-items: flex-start;

교차 축의 여러 줄 정렬 방법

align-content

stretch	Flex Items를 시작점으로 정렬
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


width: 100px;
A
B
C
D
E
display: flex;
flex-wrap: wrap;
width: 100px;
height: 100px;
A
B
C
D
E
display: flex;
flex-wrap: wrap;
A
B
C
D
E
display: flex;
flex-wrap: wrap;
align-content: flex-start;


A
B
C
D
E
display: flex;
flex-wrap: wrap;
align-content: flex-start;
display: flex;
flex-wrap: wrap;
align-content: center;
display: flex;
flex-wrap: wrap;
align-content: flex-end;

Flex! 참고 사이트-1분 코딩

https://studiomeal.com/archives/197

CSS 복잡한 속성 - animation

애니매이션을 나타내는 css 스타일

+

애니매이션의 중간 상태를 나타내는 키 프레임

@keyframes 로 설정되는 값 (to ~ from, 0% ~ 100%)

@keyframes

- CSS의 애니메이션 효과를 개발자가 직접 지정하는 기능
- 애니메이션의 중간 지점마다 CSS 속성 값을 지정하여 세밀하게 애니메이션 조절하는 기능
- 키프레임을 변수에 선언하고 해당 변수를 CSS 에서 불러와서 사용할 수 있다!

- Keyframes 로 설정되는 값
  - to/from
  - 0% ~ 100%

Animation 속성

- 이름 : keyframes 로 지정한 애니메이션 이름
- 지속시간 : 애니메이션 지속 시간
- 진행형태 : 애니메이션이 진행되는 형태(시간 함수)
  - ease / linear 등등
- 반복횟수 : 반복되는 횟수를 지정, 소수점 가능, infinite

Animation 속성

- animation-name (이름)
- animation-duration (지속 시간)
- animation-delay (지연 시간)
- animation-iteration-count (반복 횟수)
- animation-timing-function (반복 형태)
- animation-direction (애니메이션 방향)

```
animation-timing-function: ease;
animation-timing-function: ease-in;
animation-timing-function: ease-out;
animation-timing-function: ease-in-out;
animation-timing-function: linear;
```

단축 속성 : name duration timing-function delay iteration-count direction

```
animation: ani-name 2s ease-in 5s Infinite alternate;
```

CSS 복잡한 속성 - animation

- Animation-timing-function

linear	처음 속도와 마지막 속도가 일정합니다.
ease	처음엔 천천히 시작하여 빨라지고 마지막에 다시 느려집니다.
ease-in	천천히 시작되어 정상 속도가 됩니다.
ease-out	마지막에 천천히 속도를 줄여 끝납니다.
ease-in-out	천천히 시작되어 정상 속도가 되었다가 마지막에 느려집니다.