

K-Digital Training 웹 풀스택 과정

CSS

position

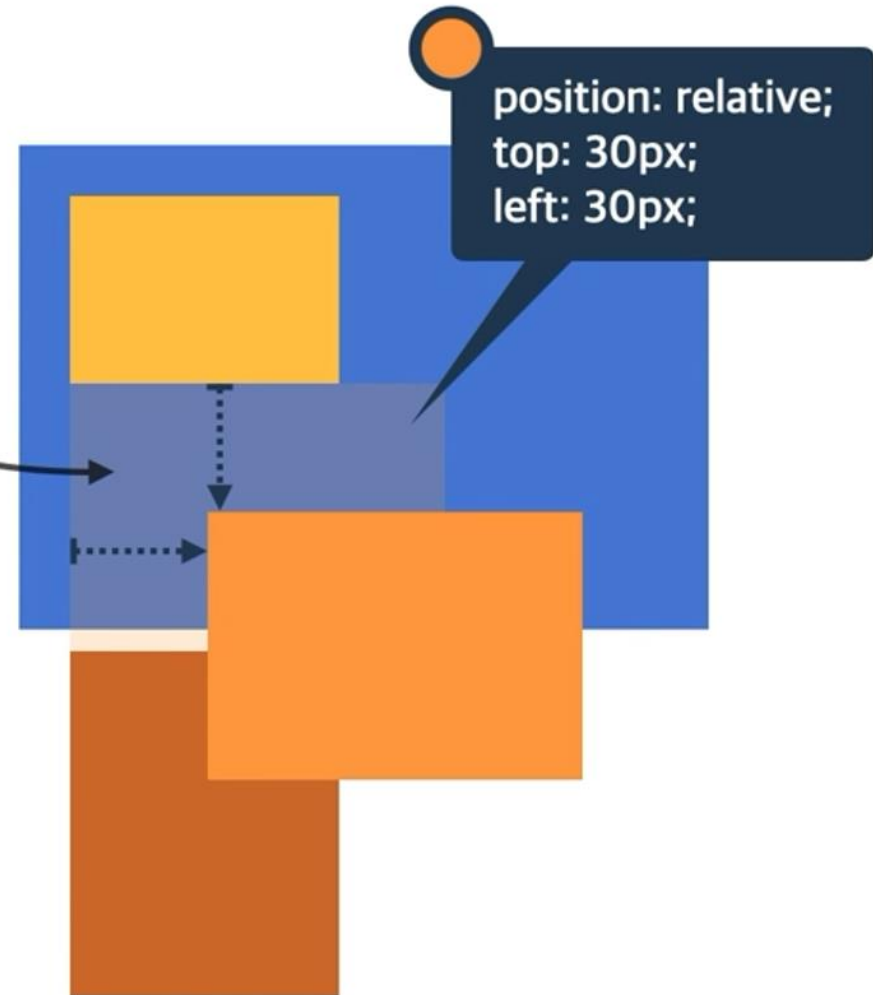
position

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

relative



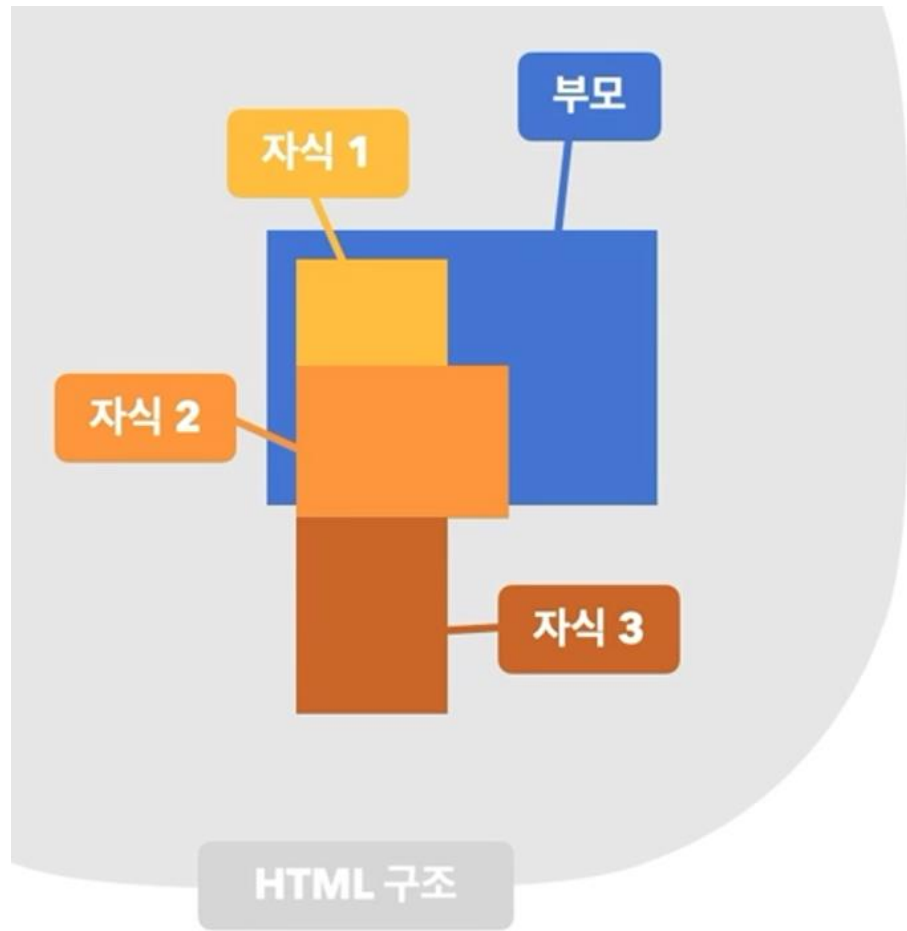
배치 전 자리는
비어 있어요!



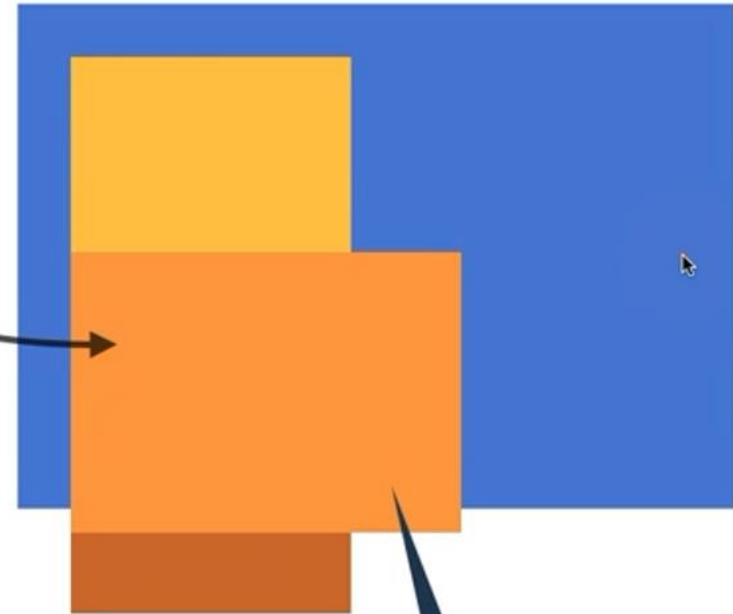
relative

요소 자신을 기준으로 배치!

absolute



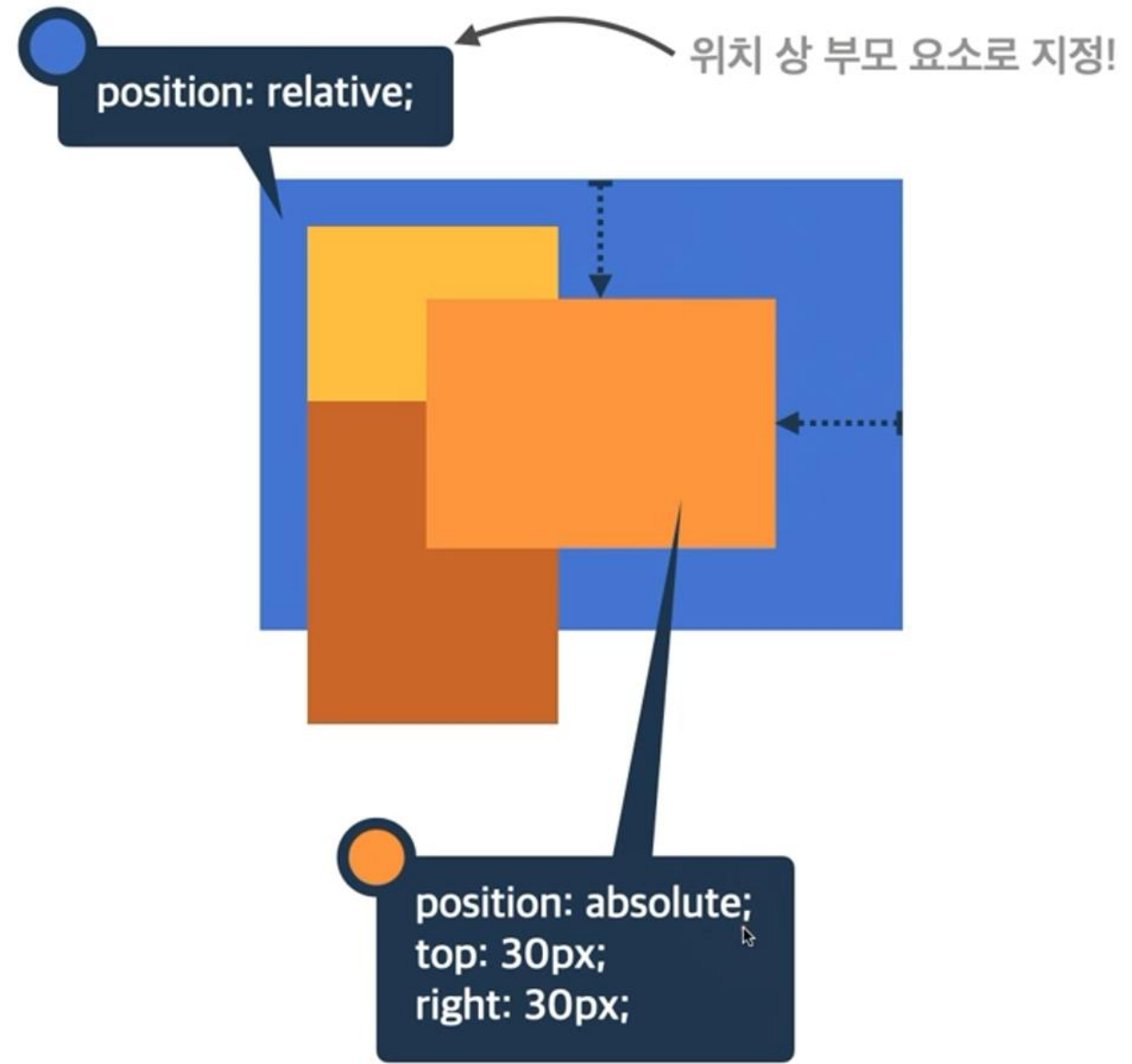
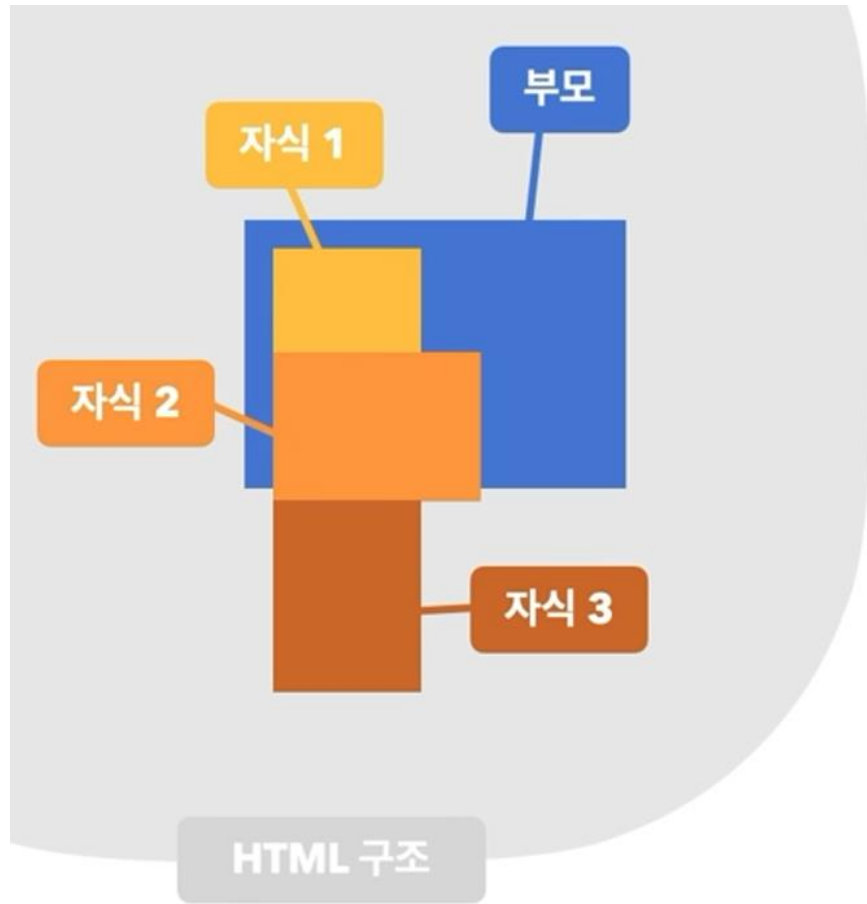
붕~ 뜨면서
요소가 겹쳐요



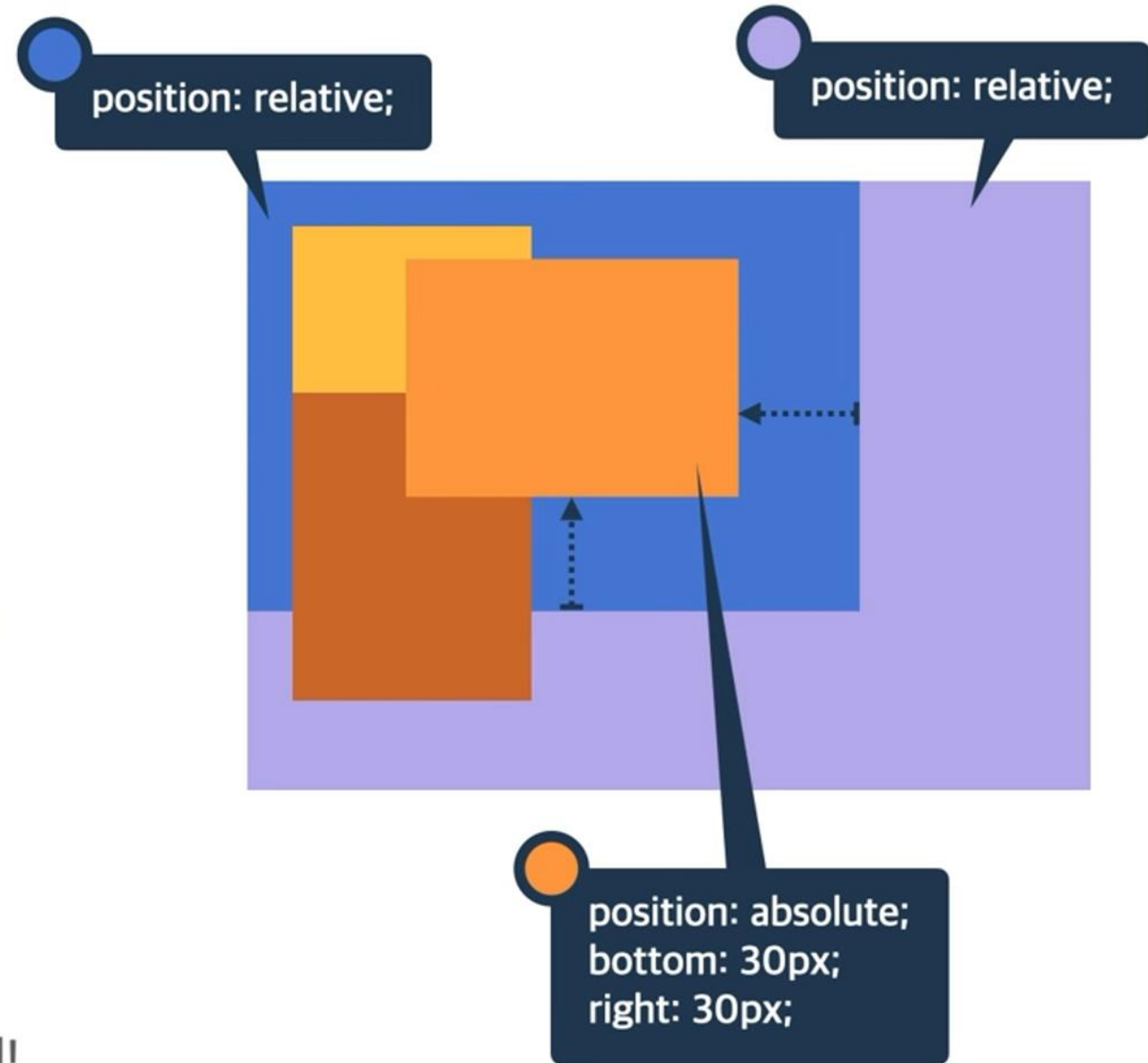
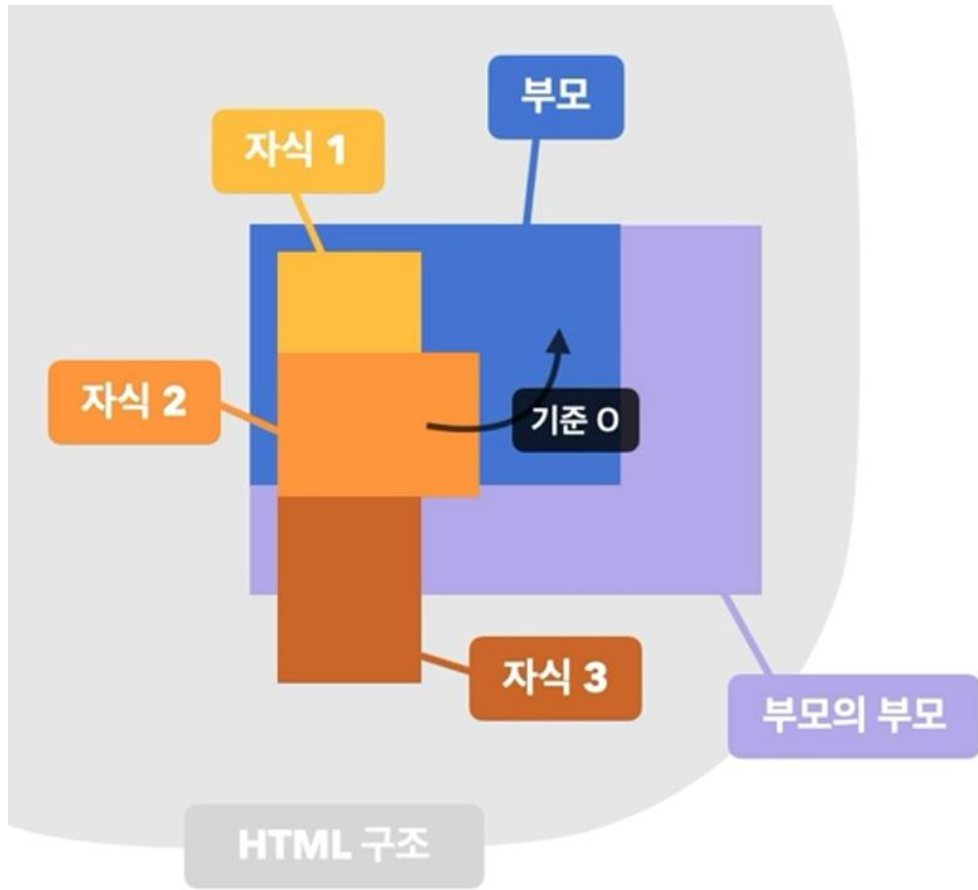
position: absolute;

absolute

위치 상 부모 요소를 기준으로 배치!

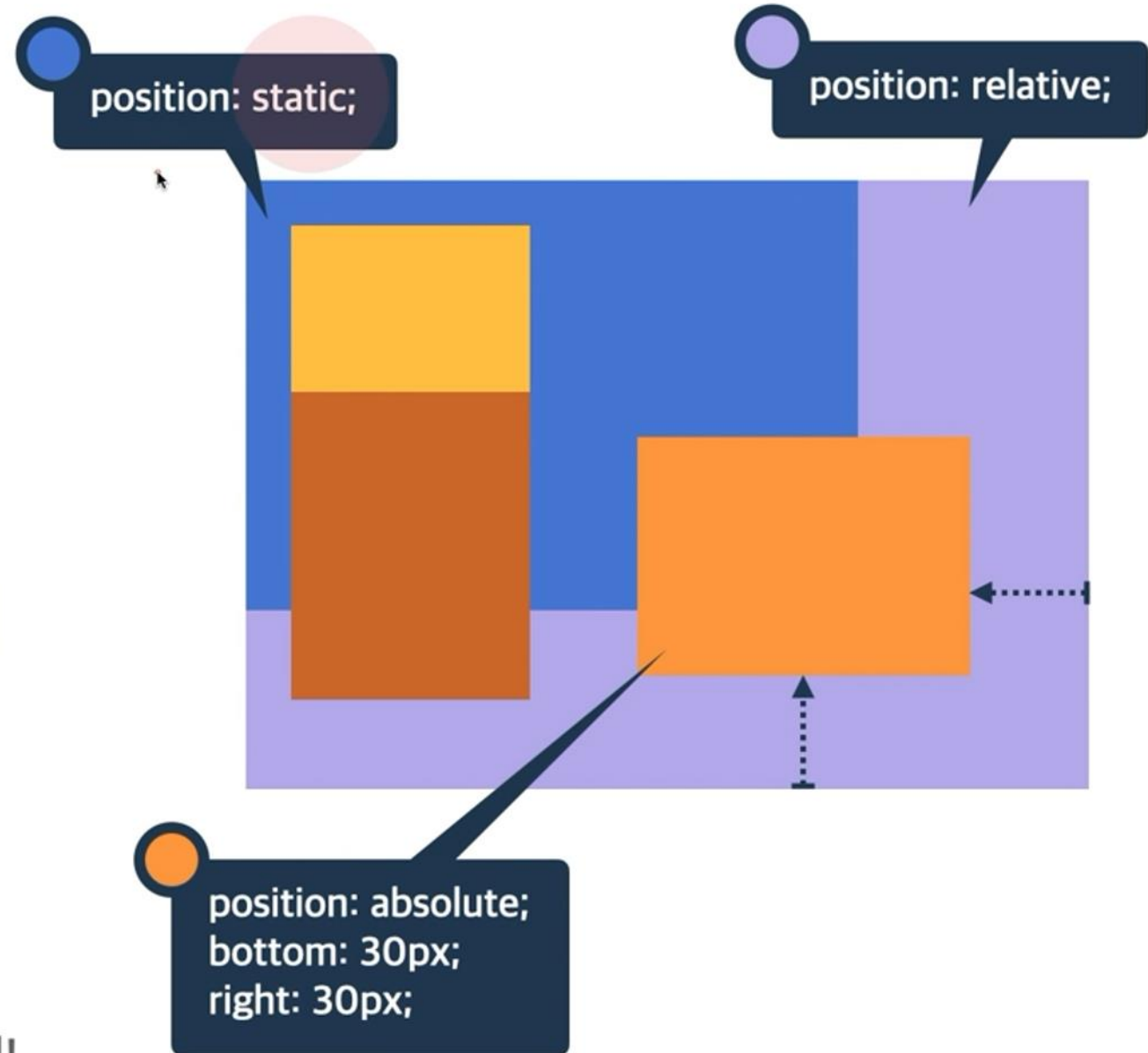
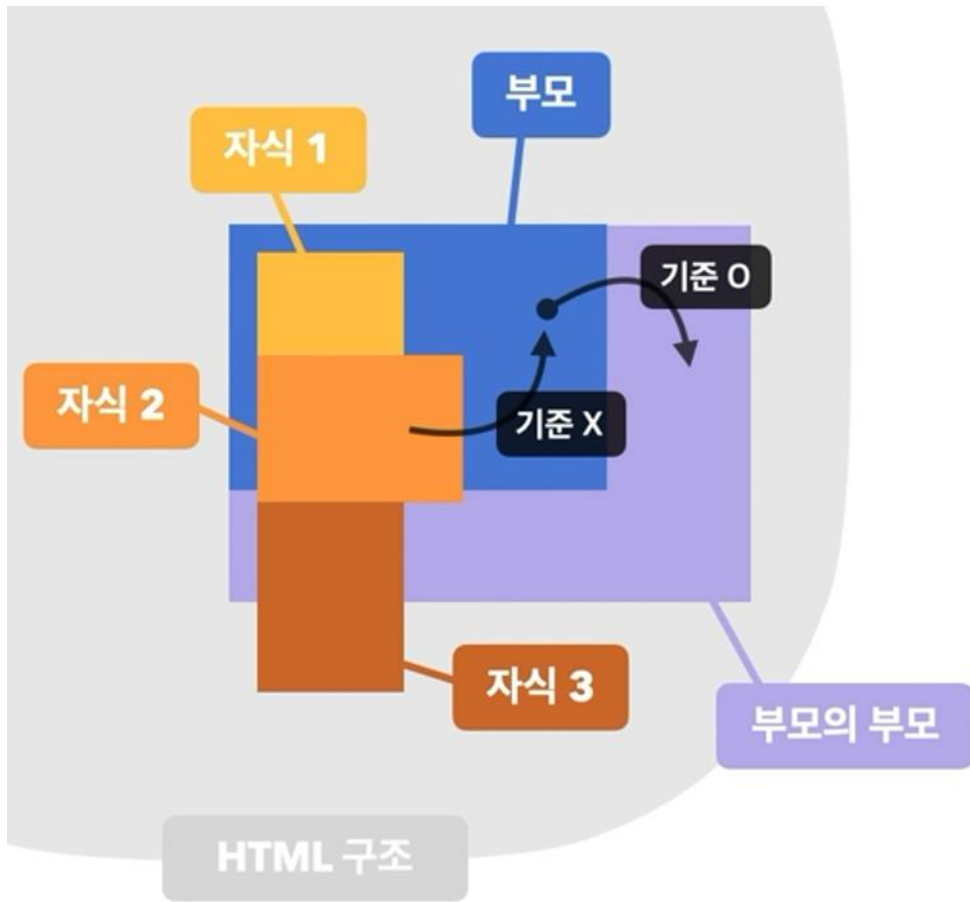


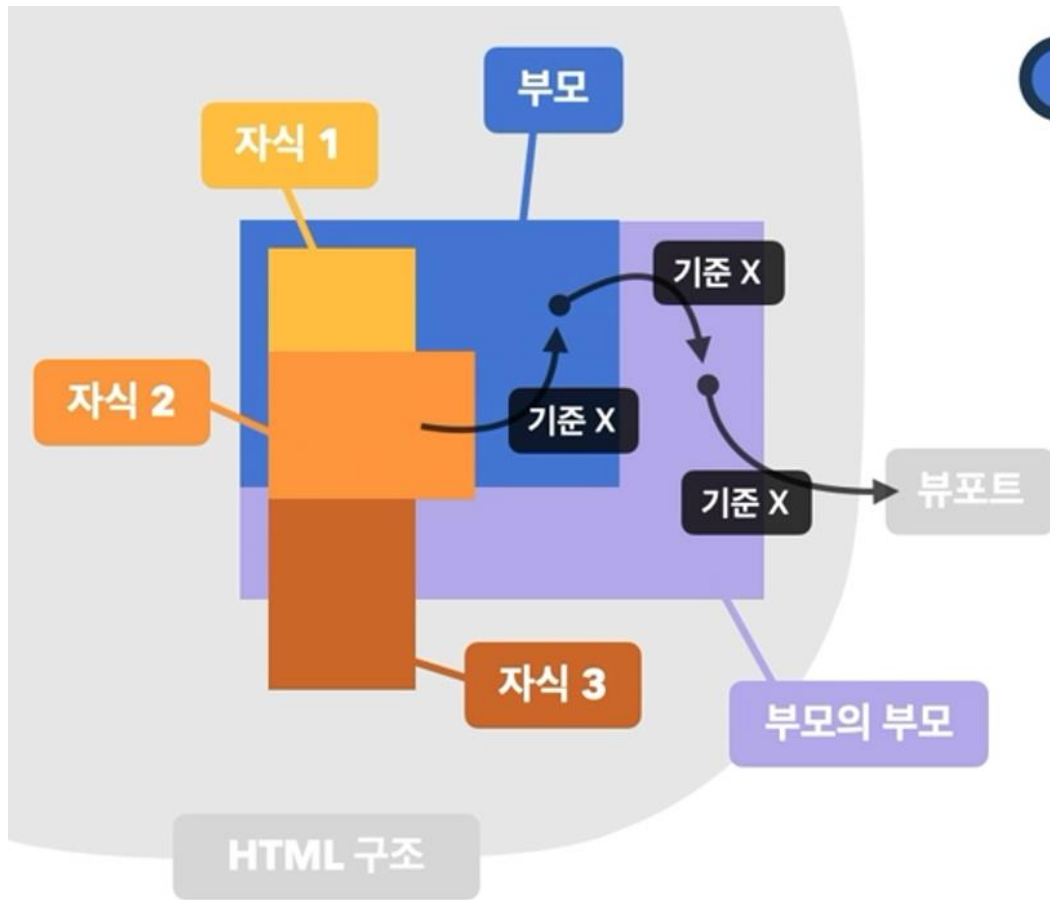
absolute 위치 상 부모 요소를 기준으로 배치!



absolute

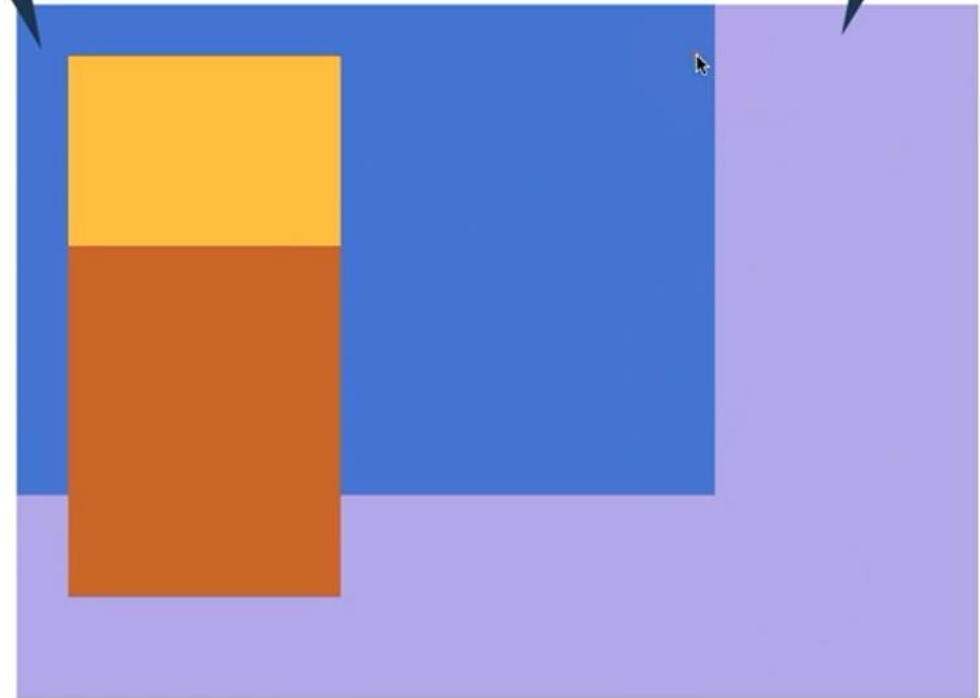
위치 상 부모 요소를 기준으로 배치!


absolute
위치 상 부모 요소를 기준으로 배치!

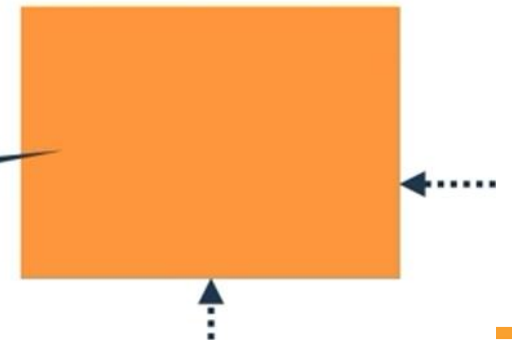


position: static;

position: static;



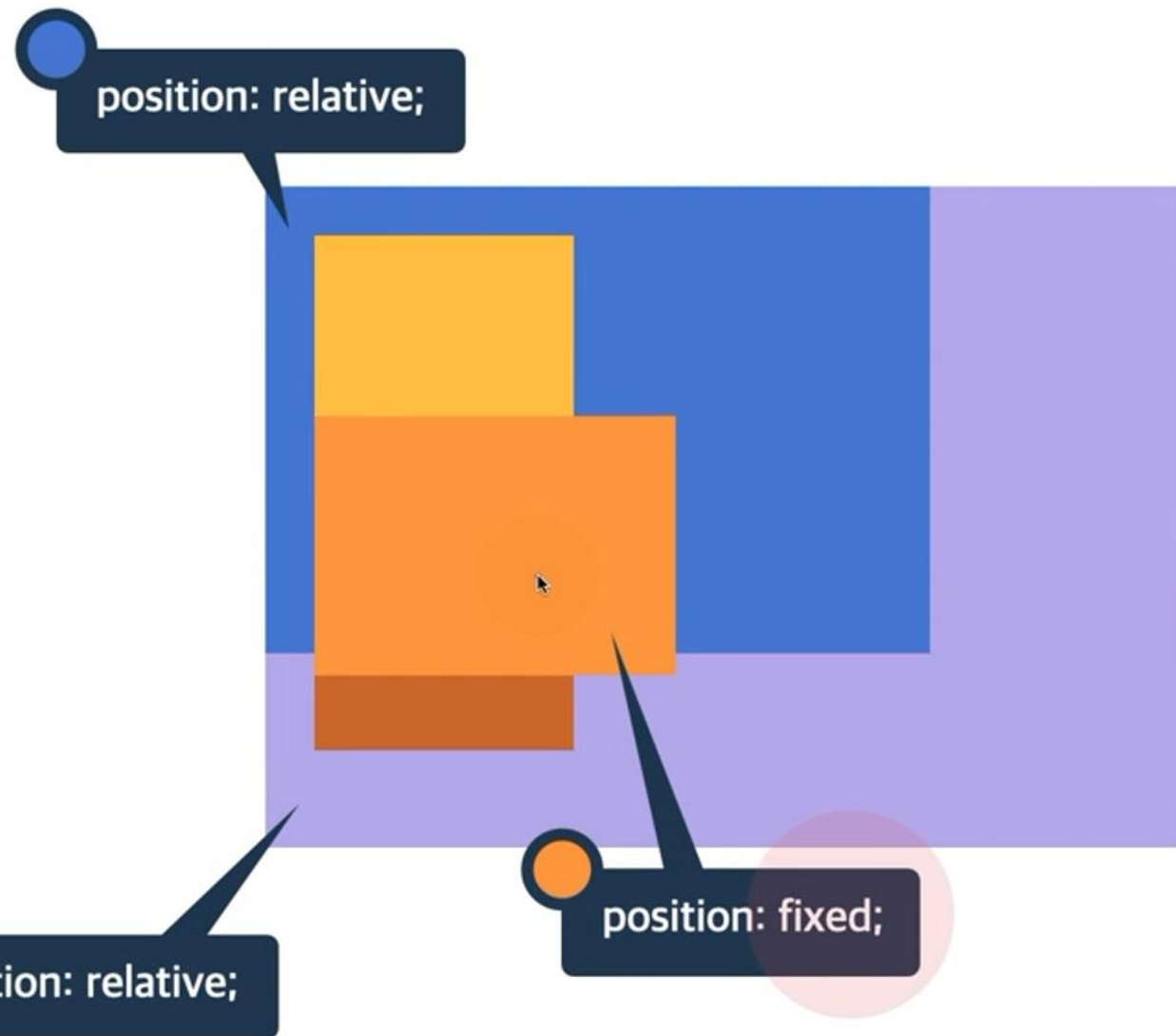
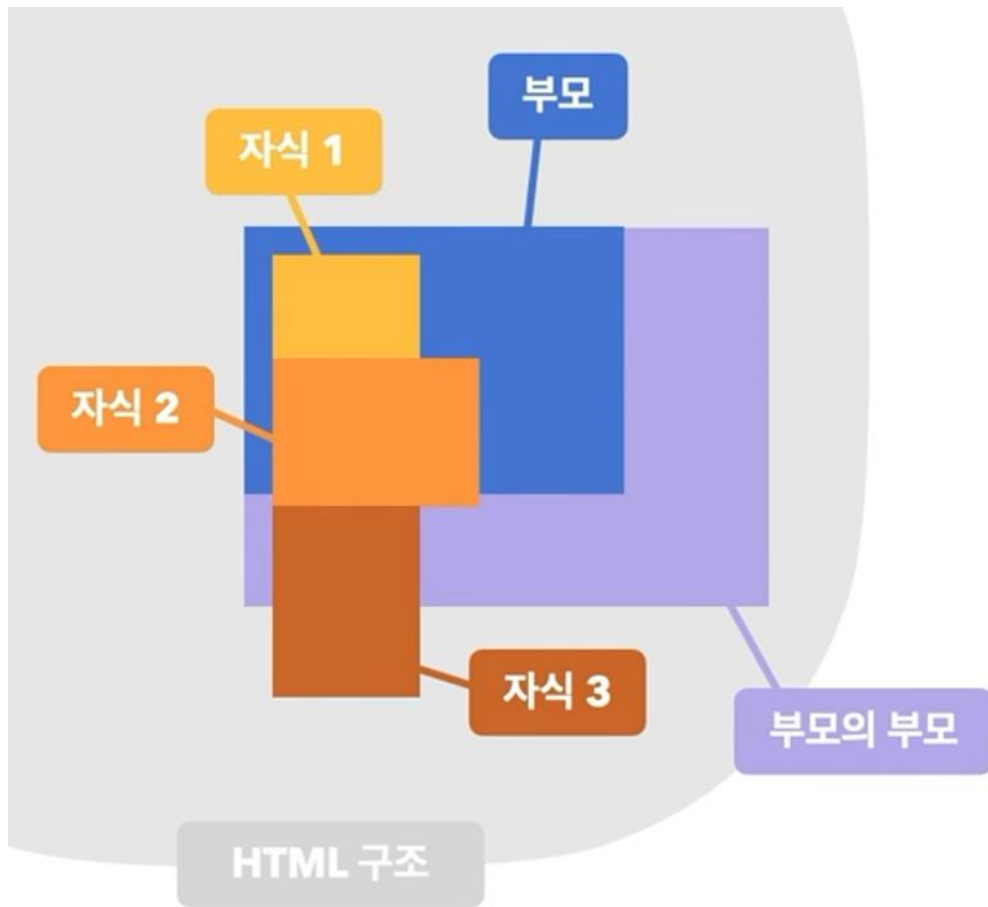
position: absolute;
bottom: 30px;
right: 30px;



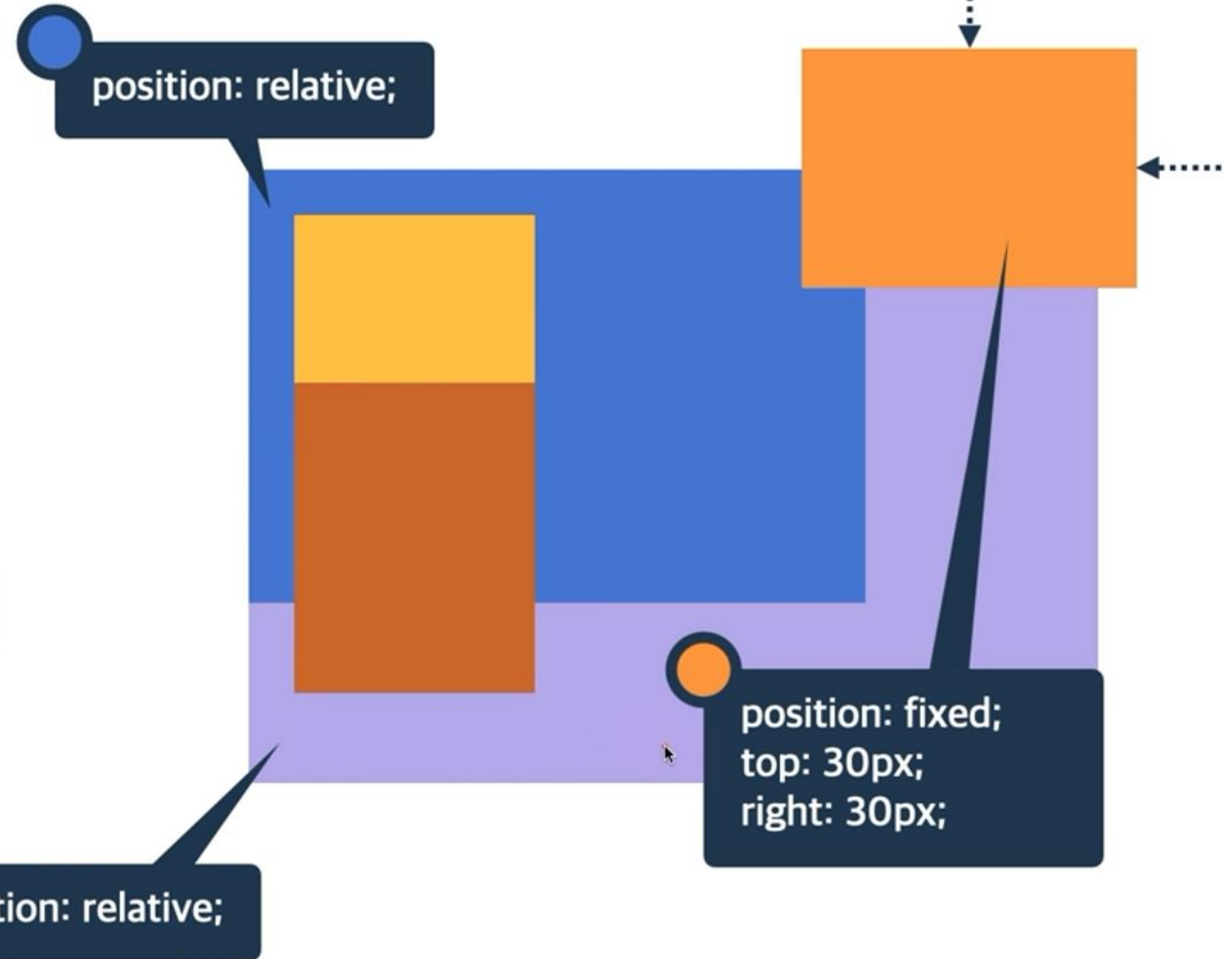
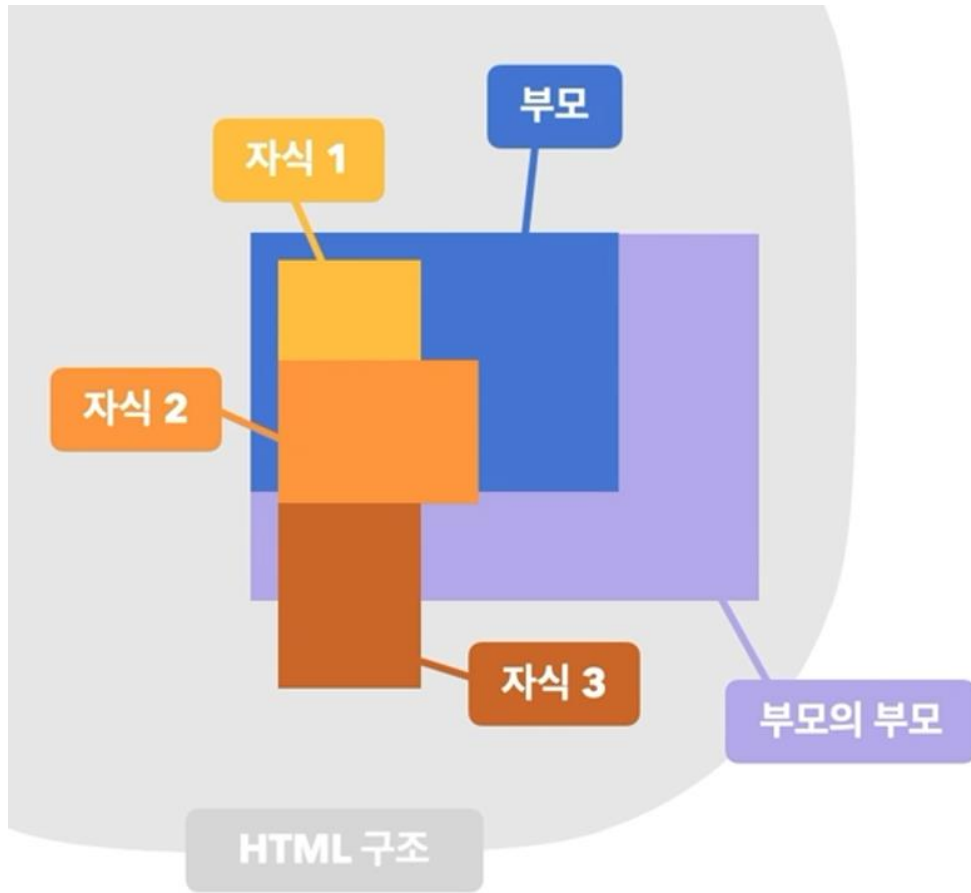
absolute

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

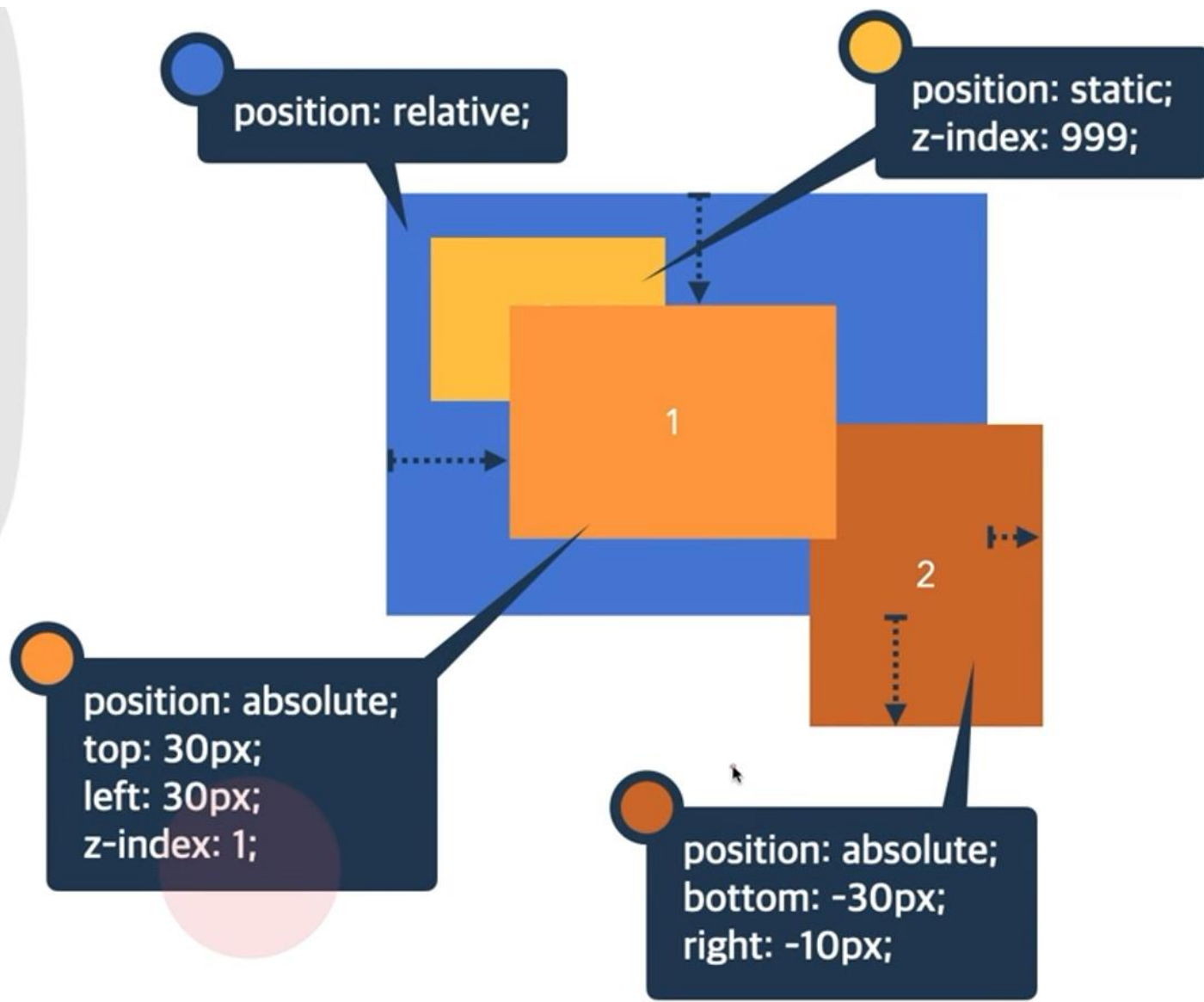
z-index

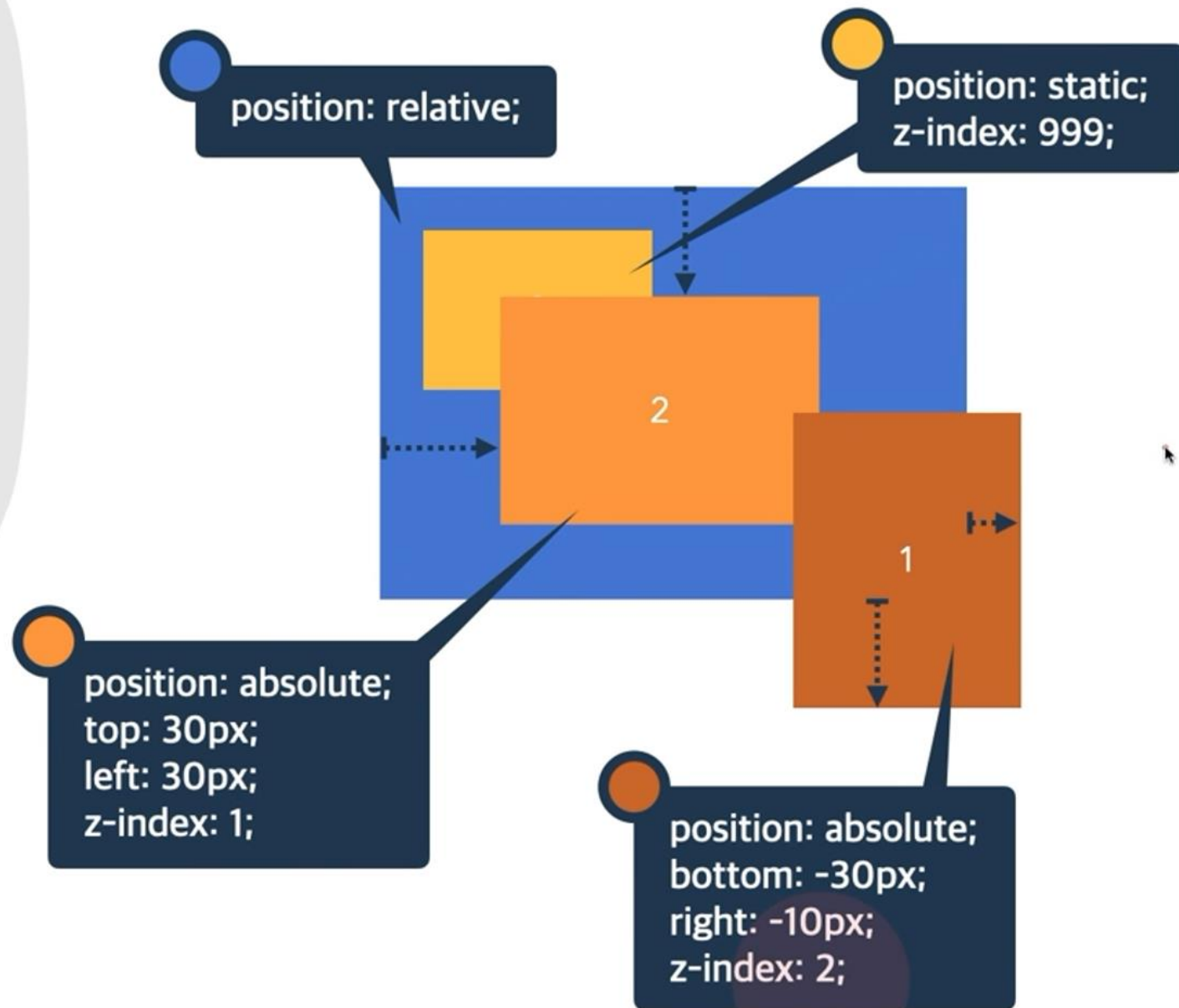
auto

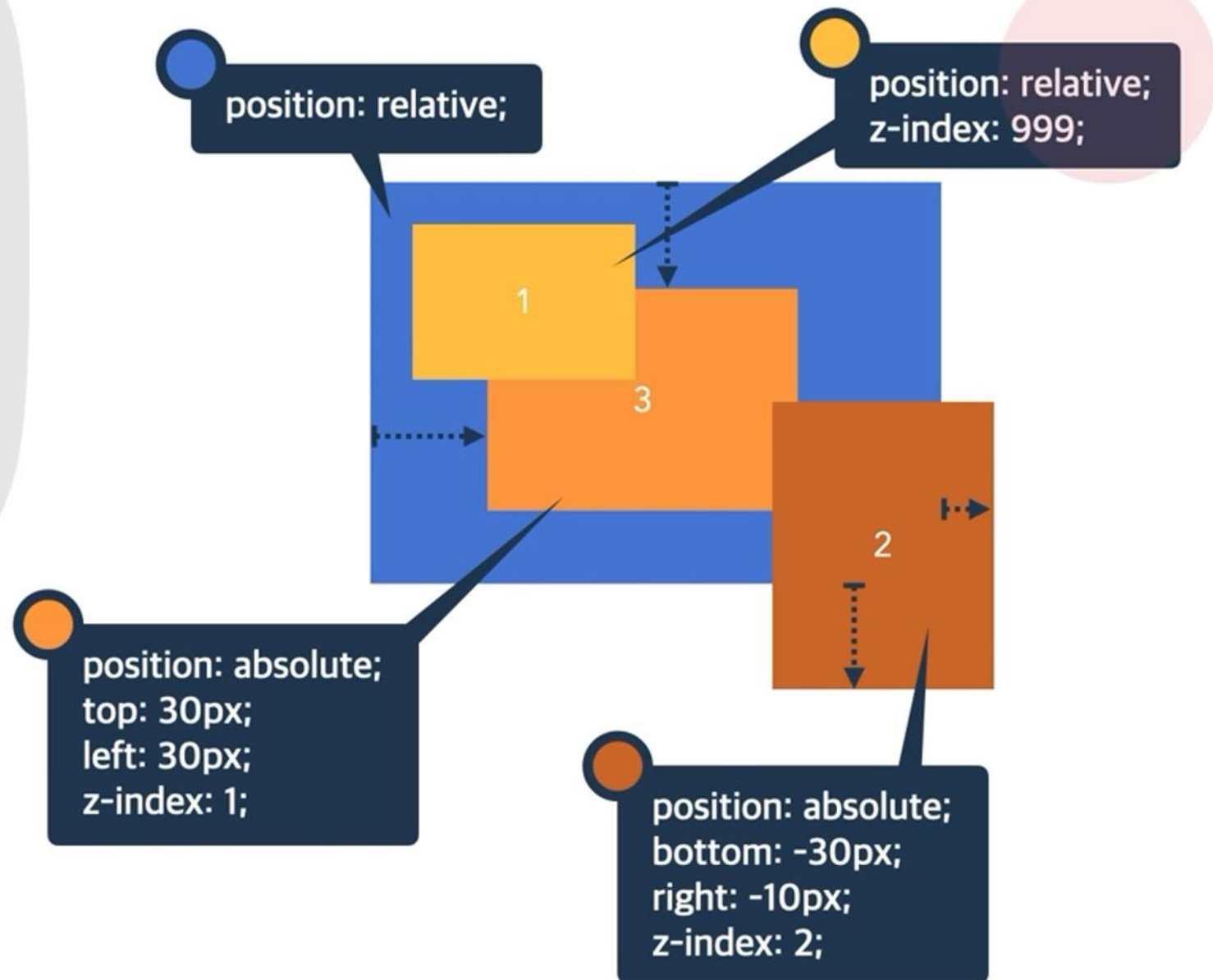
부모 요소와 동일한 쌓임 정도

숫자

숫자가 높을 수록 위에 쌓임







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;
```

z-index

```
.parent {
    position: relative;
}

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

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

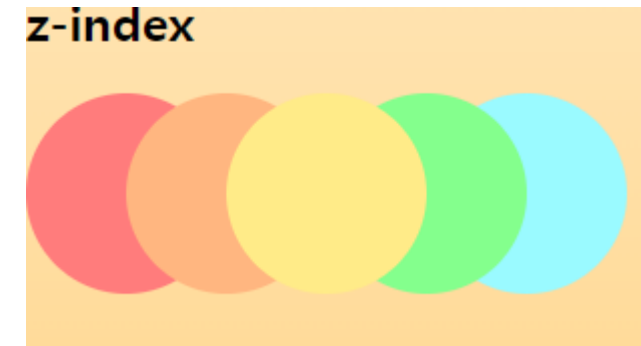
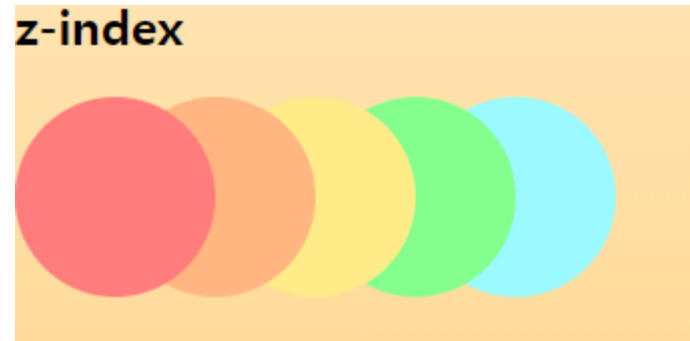
.circle2 {
    background-color: #84ff8d;
    top: 0;
    left: 150px;
}

.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>
```

배경

요소의 배경 색상

background-color

transparent

투명함

색상

지정 가능한 색상

그라데이션 넣기!

- 그라데이션 컬러를 백그라운드 요소로 삽입 가능
- background : linear-gradient()
 - 색상1 | 색상2 : 상하로 색 2개 지정
 - 방향 | 색상1 | 색상2 : 해당 방향으로 색상 2개 지정(ex, 90deg blue red)
 - 방향 | 색상1 | 색상1의 비중 | 색상2 : 색상 1의 비중을 % 로 지정
 - 방향 | 색상1 | 색상2 | 색상3 : 색상 3개 사용
- <https://developer.mozilla.org/en-US/docs/Web/CSS/gradient/linear-gradient>



```
<!DOCTYPE html>
<html lang="ko">
<head>
  <title>linear gradient</title>
  <style>
    div {
      width: 300px;
      height: 200px;
      margin-bottom: 30px;
    }

    .grad-1 {
      background: linear-gradient(grey, tomato);
    }

    .grad-2 {
      background: linear-gradient(90deg, grey, tomato);
    }

    .grad-3 {
      background: linear-gradient(90deg, grey, 75%, tomato);
    }

    .grad-4 {
      background: linear-gradient(grey, tomato, gold);
    }

  </style>
</head>
<body>
  <div class="grad-1"></div>
  <div class="grad-2"></div>
  <div class="grad-3"></div>
  <div class="grad-4"></div>
</body>
</html>
```

요소의 배경 이미지 삽입

background-image

none

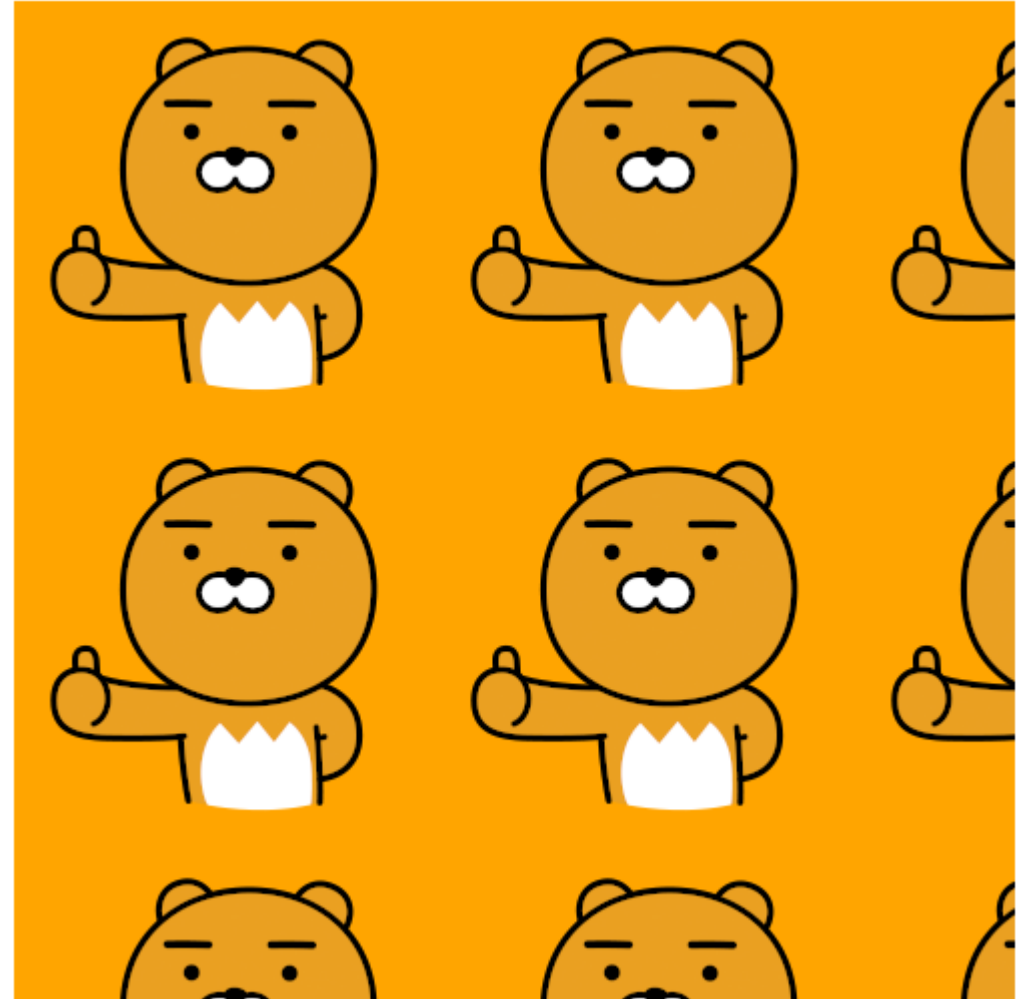
이미지 없음

url("경로")

이미지 경로

백그라운드 컬러 위에 이미지!

```
<style>
  div {
    background-color: orange;
    background-image: url("");
    width: 500px;
    height: 500px;
  }
</style>
```



요소의 배경 이미지 반복

background-repeat

repeat

이미지를 수직, 수평 반복

repeat-x

이미지를 수평 반복

repeat-y

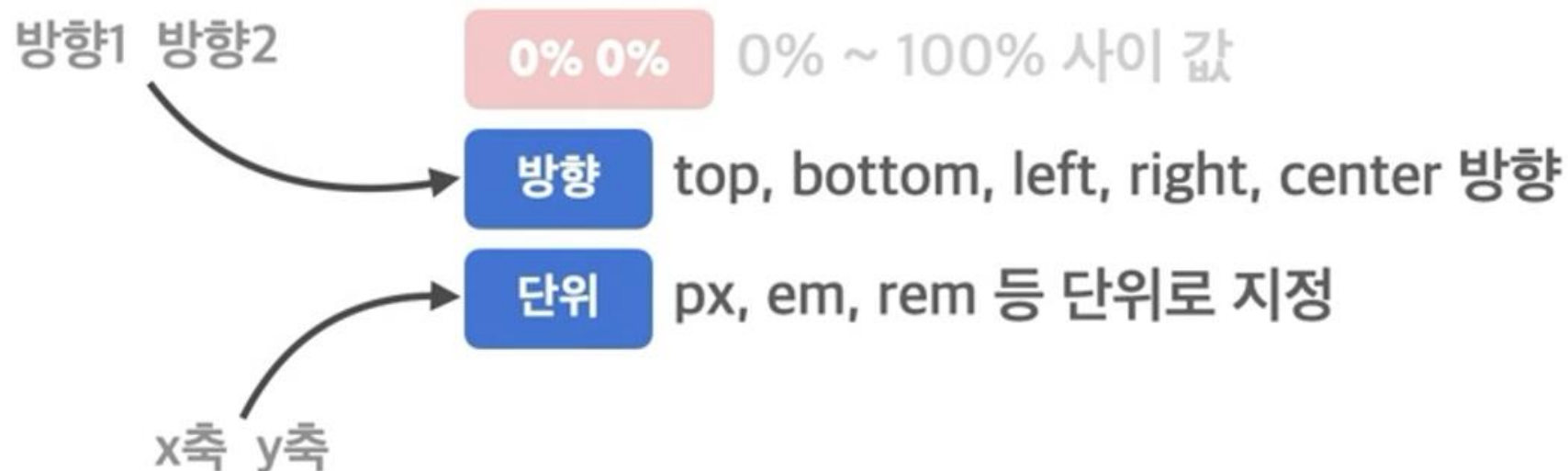
이미지를 수직 반복

no-repeat

반복 없음

요소의 배경 이미지 위치

background-position



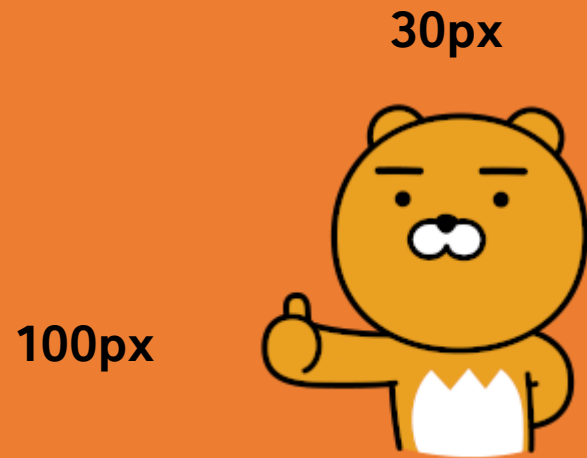
Background-position: top right;



Background-position: center;



Background-position: 100px 30px;



요소의 배경 이미지 크기

background-size

auto

이미지의 실제 크기

단위

px, em, rem 등 단위로 지정

cover

비율을 유지, 요소의 더 넓은 너비에 맞춤

contain

비율을 유지, 요소의 더 짧은 너비에 맞춤

Background-size: cover;



Background-size: contain;



요소의 배경 이미지 스크롤 특성

background-attachment

scroll

이미지가 요소를 따라서 같이 스크롤

fixed

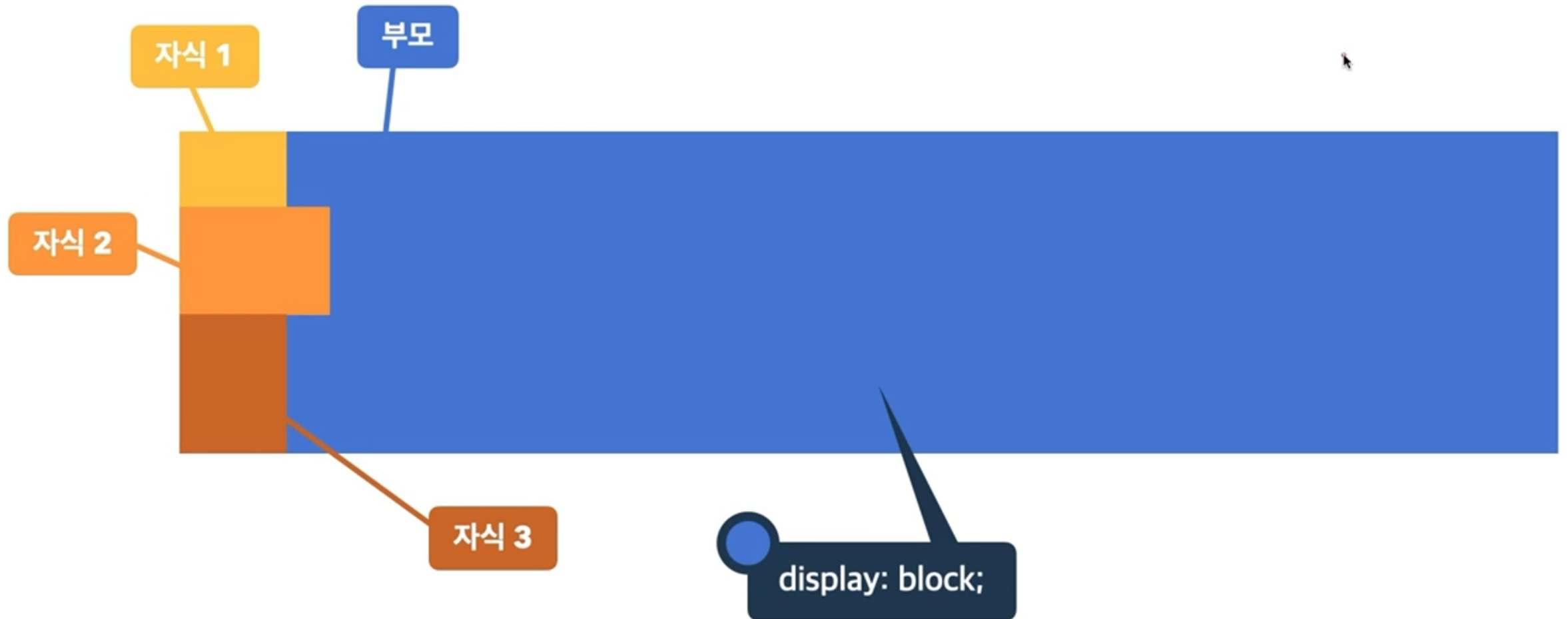
이미지가 뷰포트에 고정, 스크롤 X

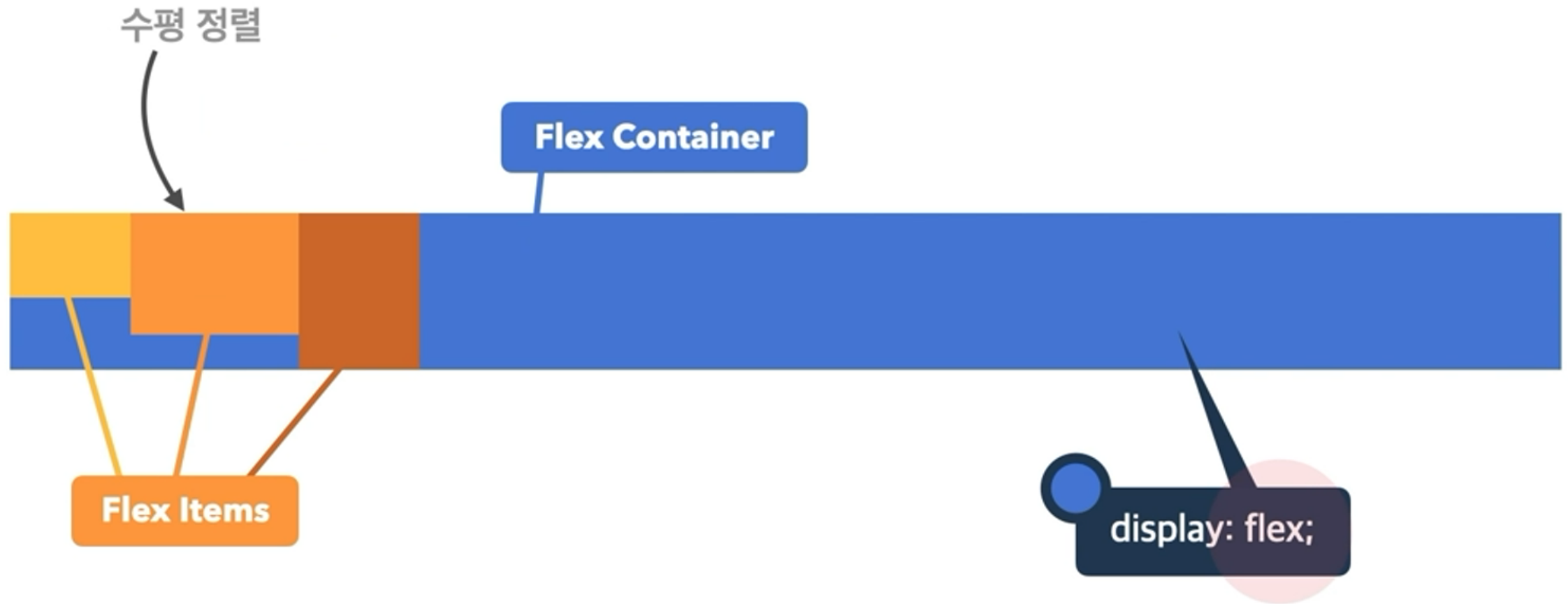
local

요소 내 스크롤 시 이미지가 같이 스크롤

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)

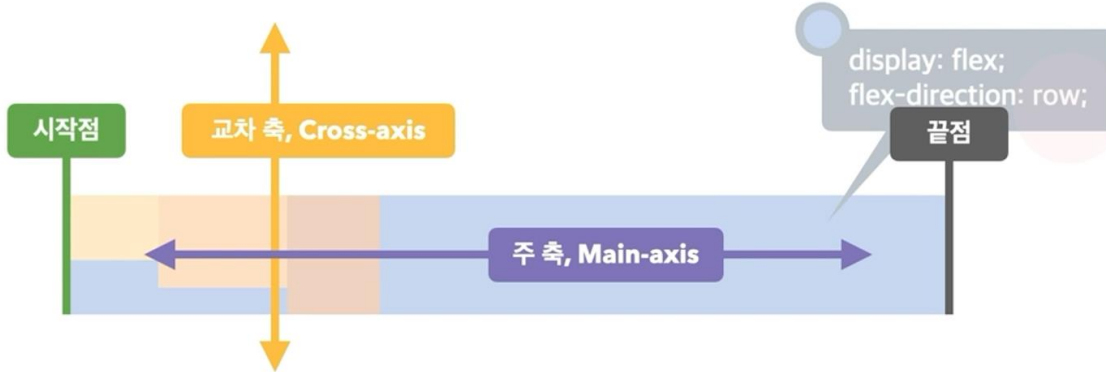
행, Row

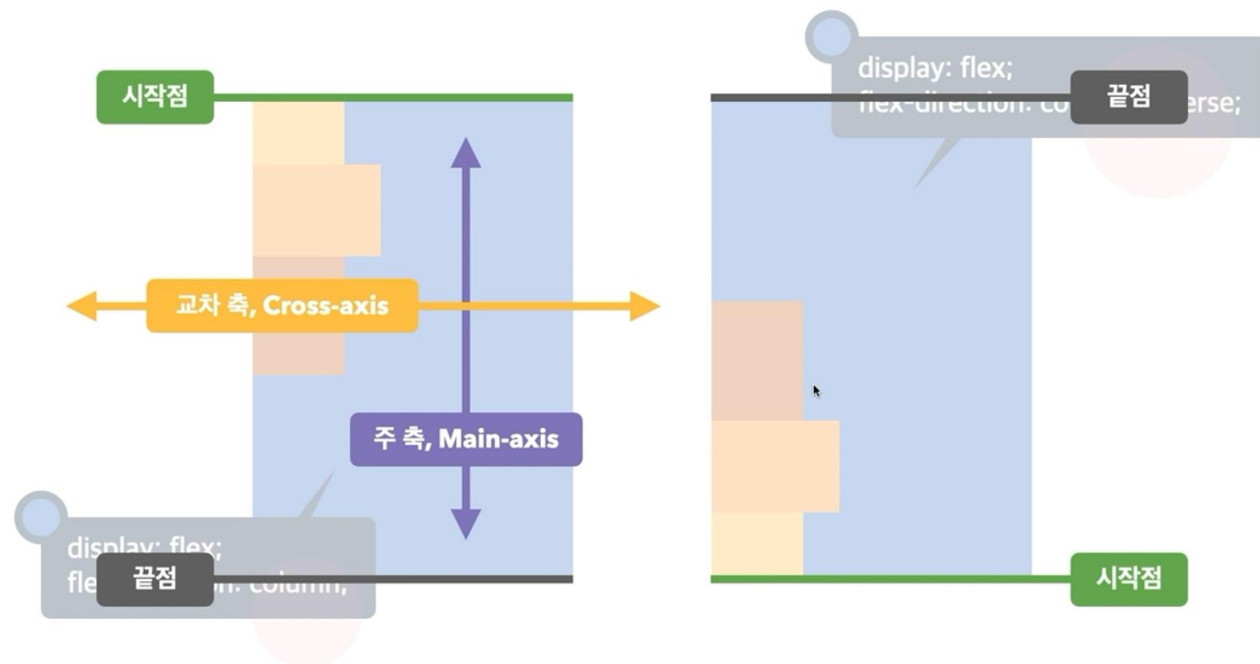
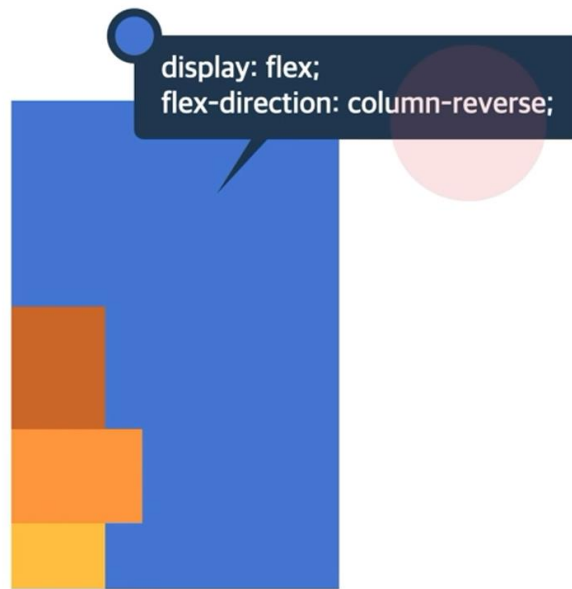
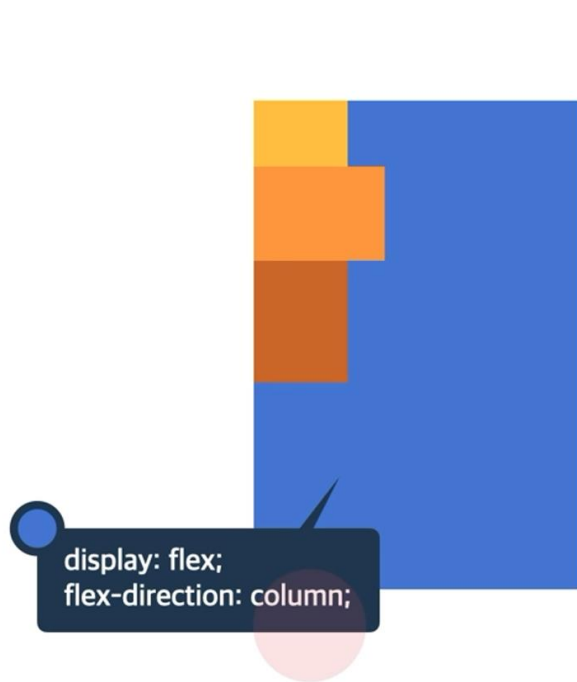
열, Column

display: flex;
flex-direction: row;



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





Flex Items 묶음(줄 바꿈) 여부

flex-wrap

nowrap

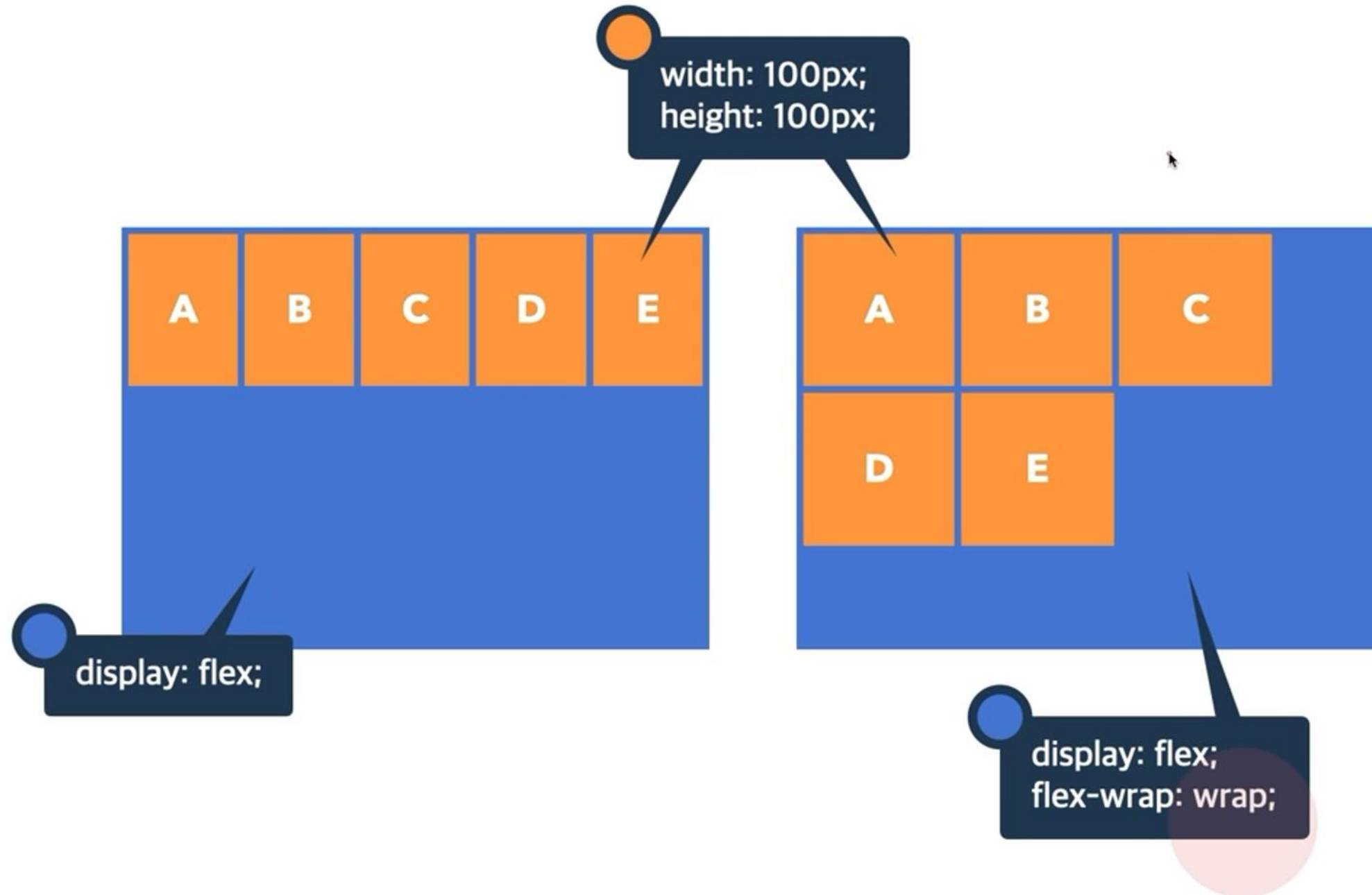
묶음(줄 바꿈) 없음

wrap

여러 줄로 묶음

wrap-reverse

wrap의 반대 방향으로 묶음



실습6. flex-wrap

- 500px 500px 컨테이너 선언
- 100px 100px 아이템 div 8개 선언
- 컨테이너 안에 한 줄이 되도록 설정
- 컨테이너 안에 두 줄이 되도록 설정

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

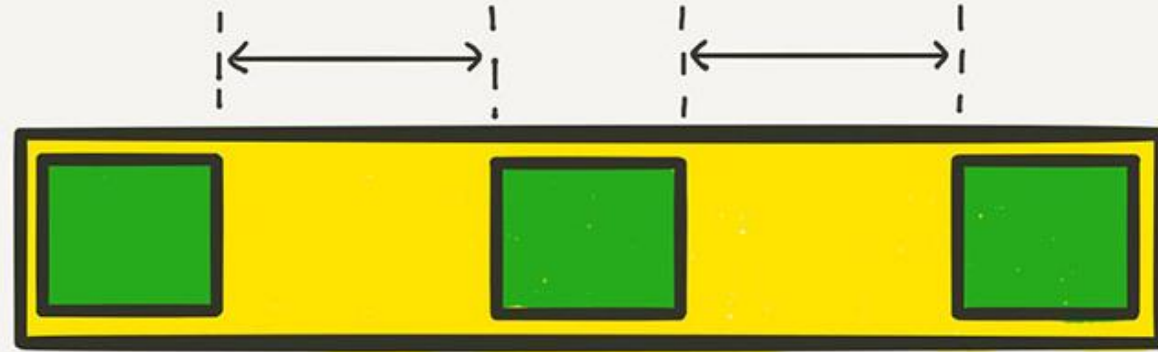


display: flex;

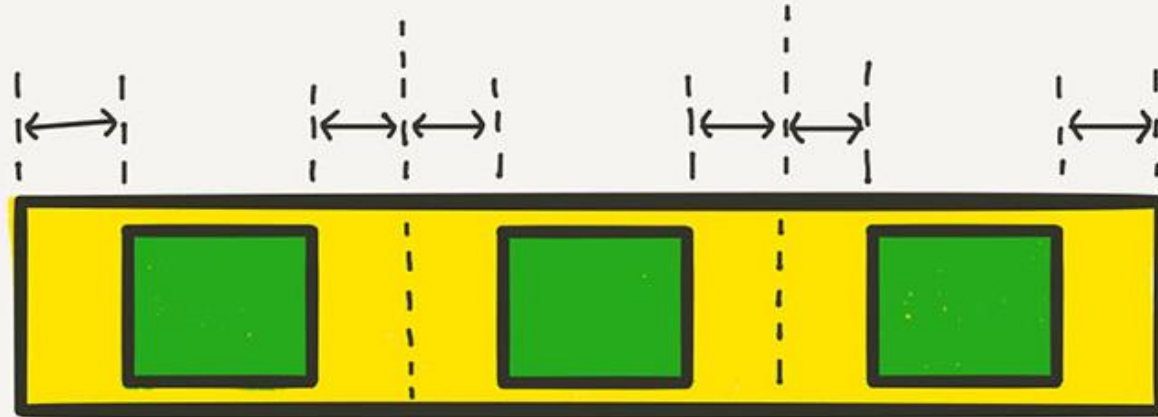
display: flex;
justify-content: flex-end;

display: flex;
justify-content: center;

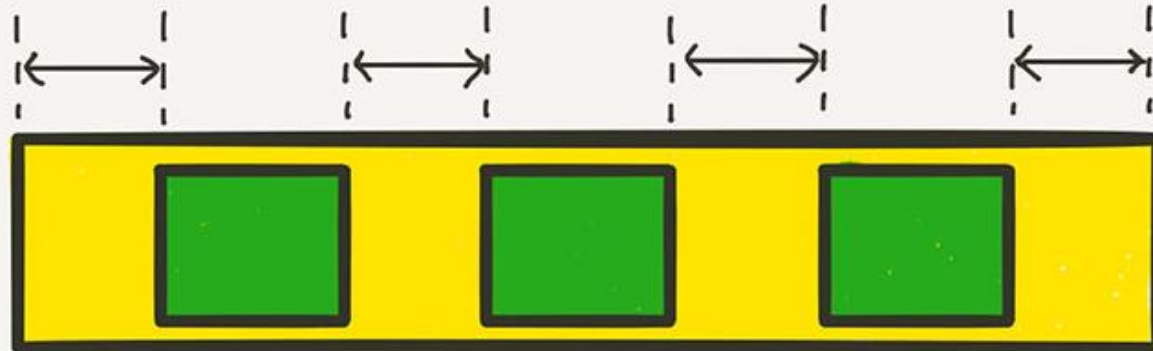
space-between



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

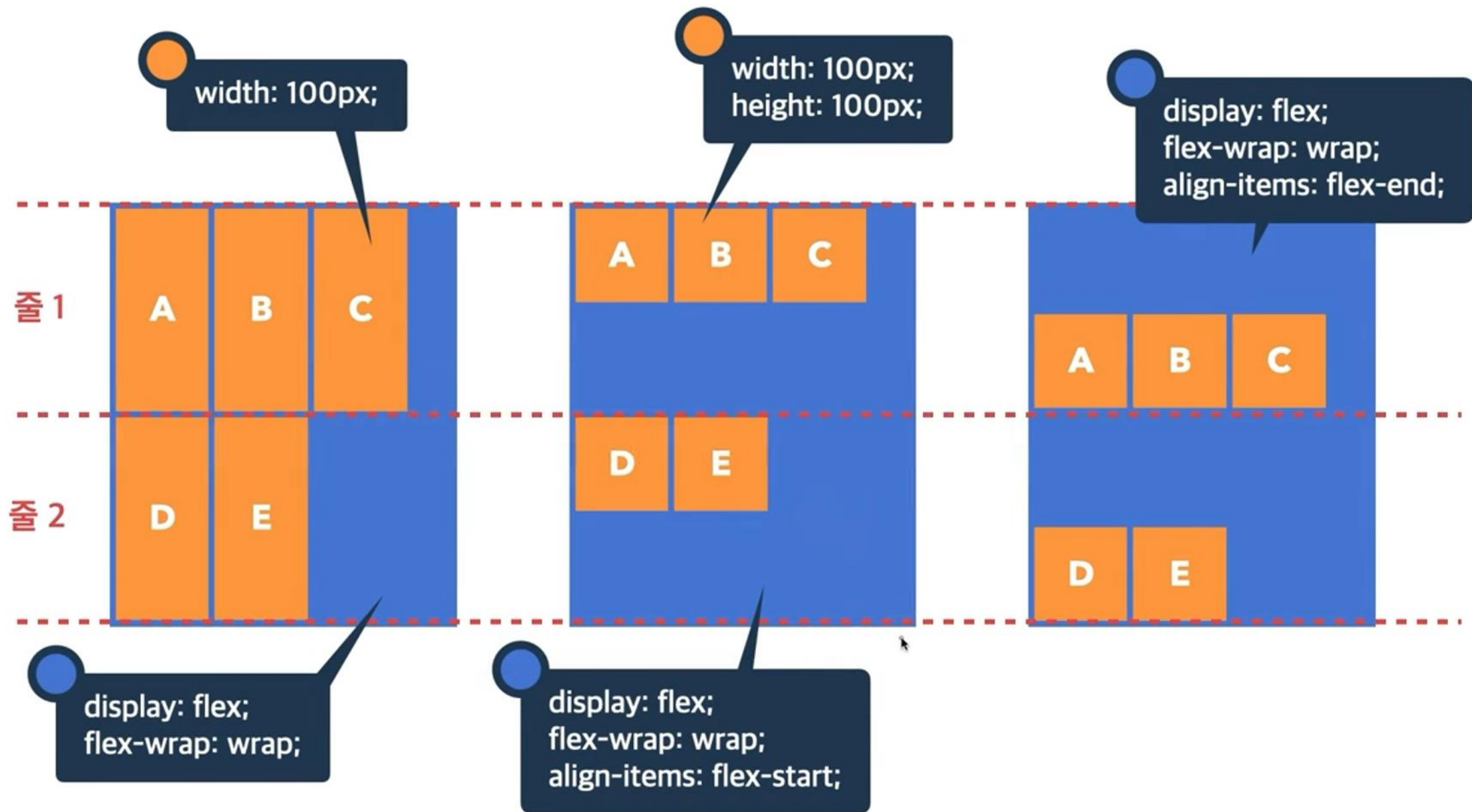
Flex Items를 각 줄의 끝점으로 정렬

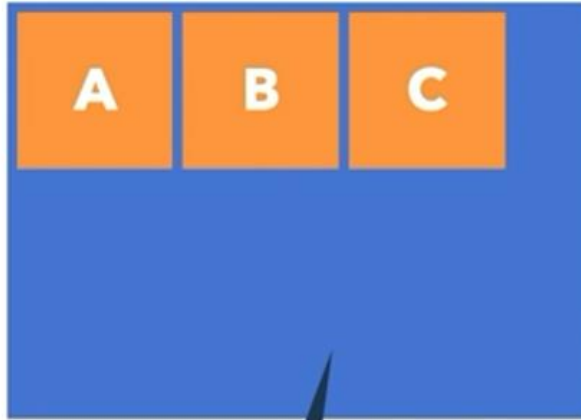
center

Flex Items를 각 줄의 가운데 정렬

baseline

Flex Items를 각 줄의 문자 기준선에 정렬





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



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



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

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

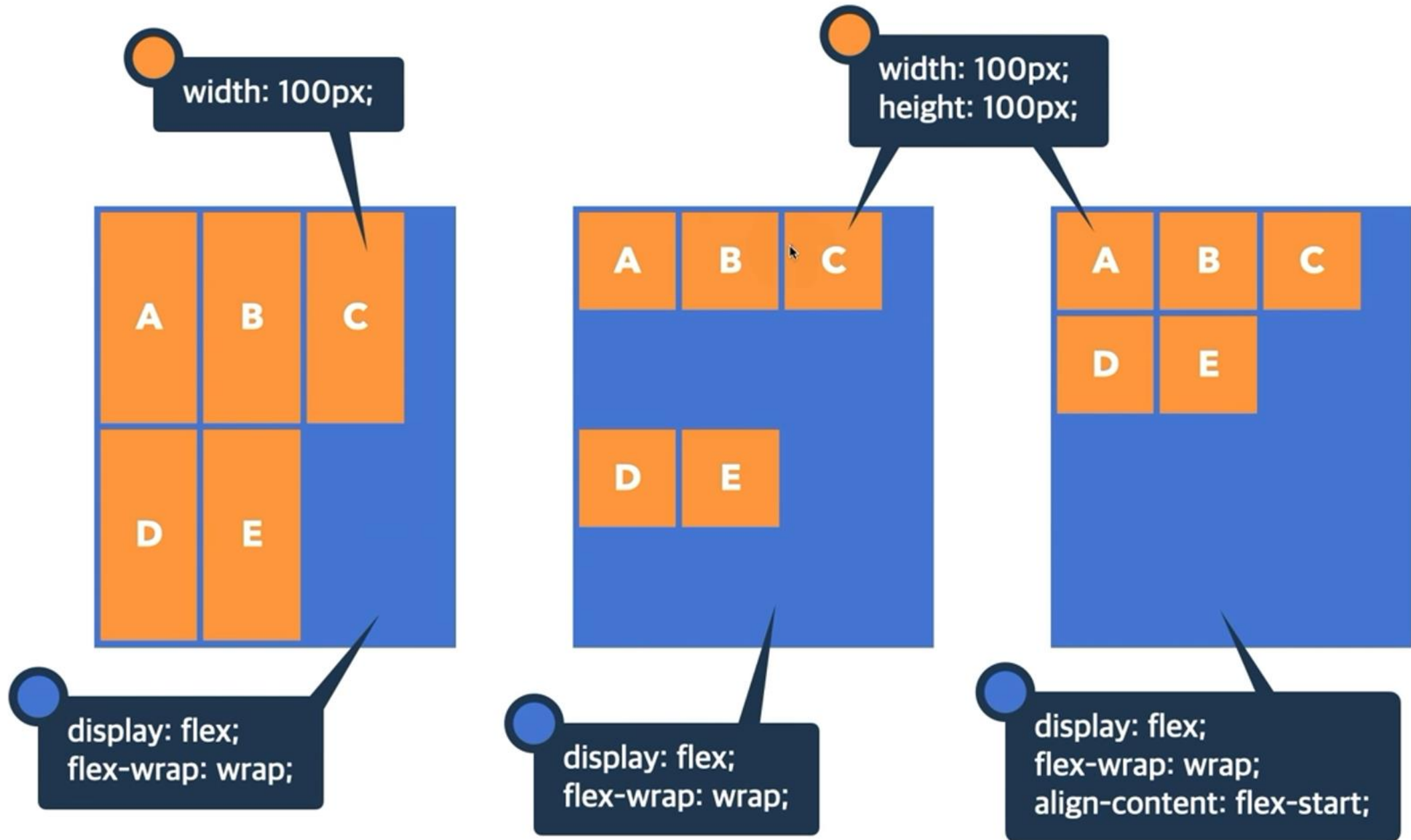
Flex Items를 가운데 정렬

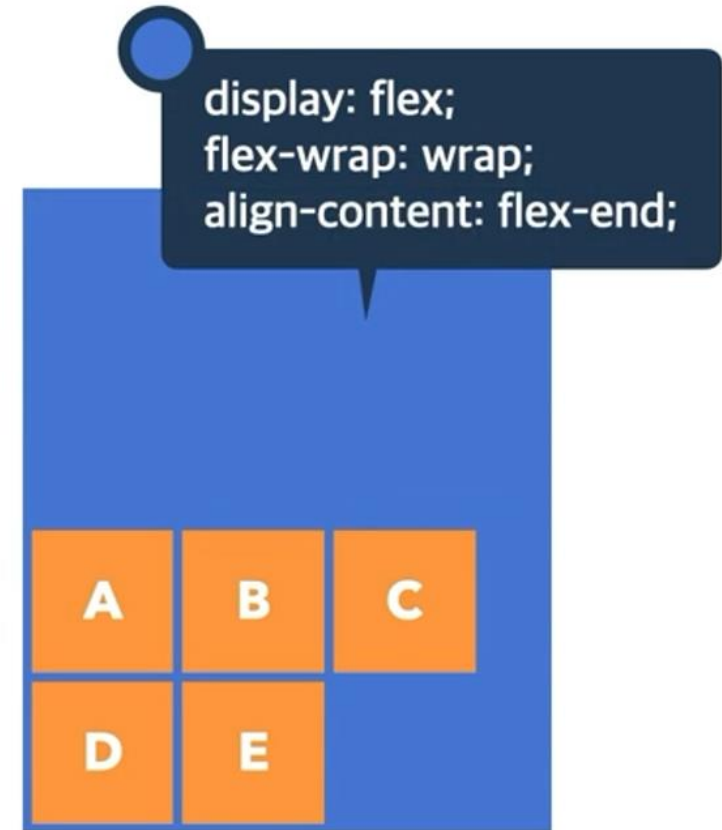
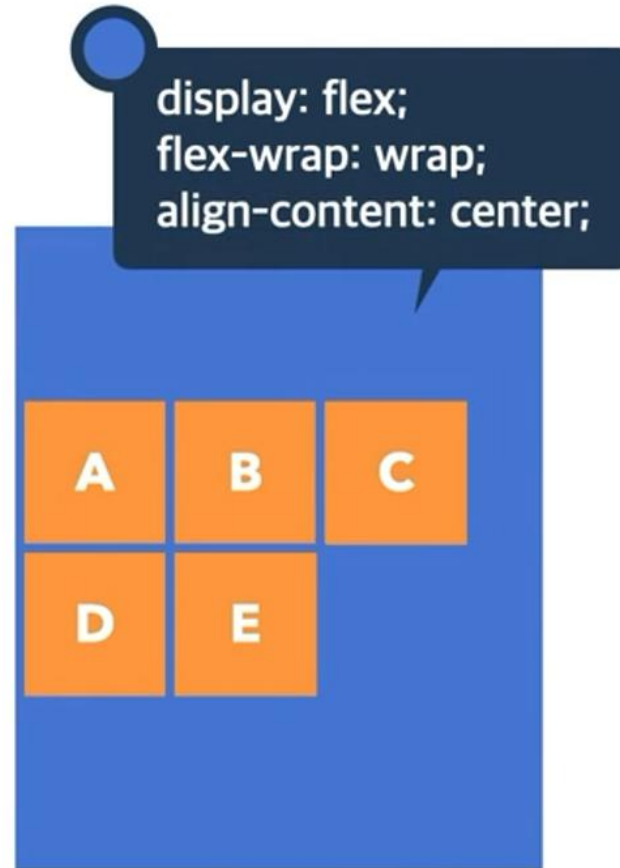
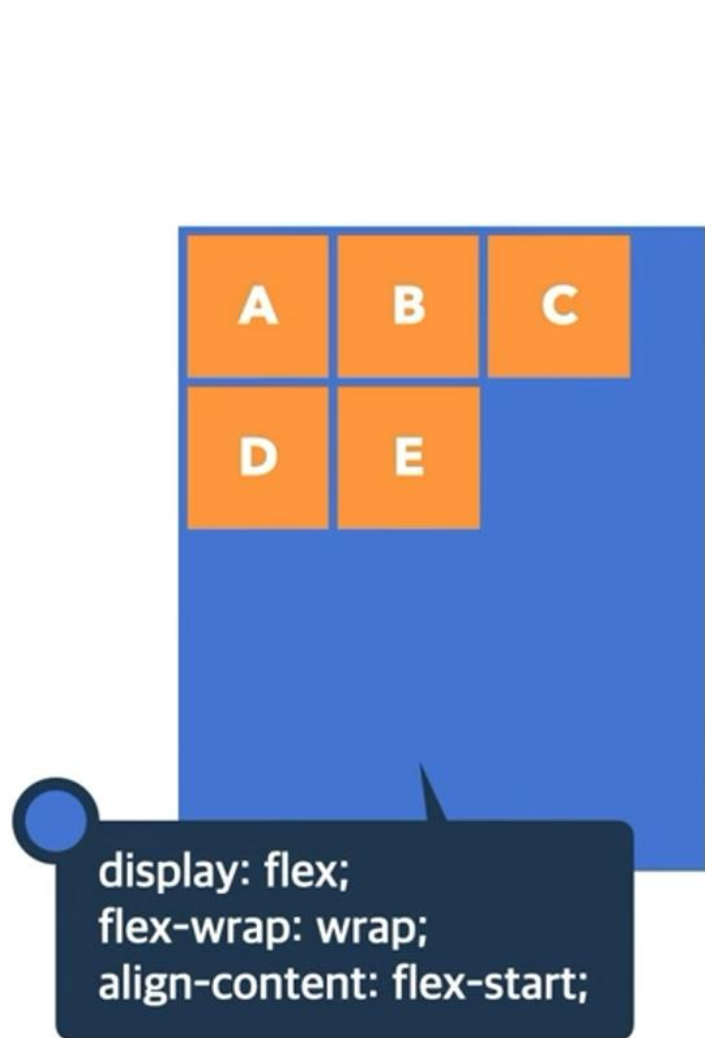
space-between

각 Flex Item 사이를 균등하게 정렬

space-around

각 Flex Item의 외부 여백을 균등하게 정렬





Flex! 참고 사이트-1분 코딩

<https://studiomeal.com/archives/197>
