

Hello,

KDT 웹 개발자 양성 프로젝트

5기!

with



CSS의 속성

박스 모델
글꼴, 문자

배경

배치

플렉스(정렬)

전환

변환

띄움

애니메이션

그리드

다단

필터



박스 모델

inline, block, inline-block



	inline	block	inline-block
기본 넓이	컨텐츠 만큼	부모의 넓이	컨텐츠 만큼
width, height	무시	적용	적용
가로 공간 차지	공유	독점	공유
margin	가로만	가로, 세로 전부 *세로 상쇄	가로, 세로 전부
padding	가로만, *세로는 배경색만	가로, 세로 전부	가로, 세로 전부

inline padding, 세로는 배경색만?



```
<style>
  html {
    font-size: 20px;
    font-weight: bold;
  }
  span {
    color: white;
    background-color: blue;
    padding: 10px;
  }
  div {
    color: white;
    background-color: red;
  }
</style>
```



margin은 크게 남기는 쪽으로 기운다?



- 두 블록 요소의 margin 0이 겹칠 경우 margin의 값이 큰 쪽만 반영



Width, Height

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <title>Inline & Block & Inlineblock</title>
  <style>
    .div1 {
      background-color: hotpink;
      height: 300px;
      margin-bottom: 20px;
    }
    .div2 {
      background-color: lawngreen;
      height: 300px;
      margin-top: 50px;
    }
  </style>
  </head>
  <body>
    <div class="div1"></div>
    <div class="div2"></div>
  </body>
</html>
```





요소의 가로/세로 너비

width, height

기본값
(요소에 이미 들어있는 속성의 값)

- auto 브라우저가 너비를 계산
- 단위 px, em, vw 등 단위로 지정

요소가 커질 수 있는 최대 가로/세로 너비

max-width, max-height

- none 최대 너비 제한 없음
- auto 브라우저가 너비를 계산
- 단위 px, em, vw 등 단위로 지정



요소가 작아질 수 있는 최소 가로/세로 너비

min-width, min-height

- 0 최소 너비 제한 없음
- auto 브라우저가 너비를 계산
- 단위 px, em, vw 등 단위로 지정

Margin

요소의 외부 여백(공간)을 지정하는 단축 속성

가로(세로) 너비가 있는 요소의 가운데 정렬에 활용해요!

margin

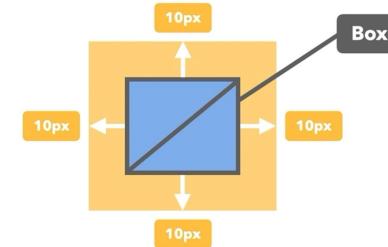
- 0 외부 여백 없음
- auto 브라우저가 여백을 계산
- 단위 px, em, vw 등 단위로 지정
- % 부모 요소의 가로 너비에 대한 비율로 지정

음수를 사용할 수 있어요!



`margin: 10px;`

`top, right, bottom, left`



Margin, 단축 속성 정리

`margin: top, right, bottom, left ;`
`margin: top, bottom left, right ;`
`margin: top left, right bottom ;`
`margin: top right bottom left ;`



Margin, 개별 속성 정리

margin-top
margin-bottom
margin-left
margin-right



Padding



요소의 내부 여백(공간)을 지정하는 단축 속성

padding

0 내부 여백 없음

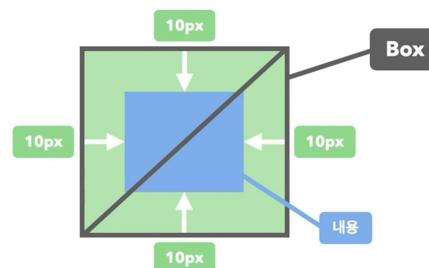
단위 px, em, vw 등 단위로 지정

% 부모 요소의 가로 너비에 대한 비율로 지정

요소의 크기가 커져요!

`padding: 10px;`

`top, right, bottom, left`



Padding, 단축 속성 정리

`padding: top, right, bottom, left ;`

`padding: top, bottom left, right ;`

`padding: top left, right bottom ;`

`padding: top right bottom left ;`

Padding, 개별 속성 정리



**padding-top
padding-bottom
padding-left
padding-right**

Border



요소의 크기가 커져요!

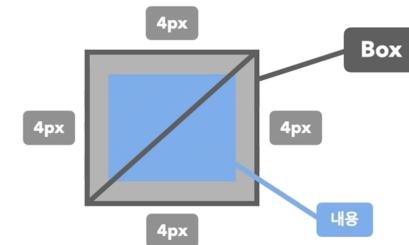
border: 선-두께 선-종류 선-색상;

요소의 테두리 선을 지정하는 단축 속성

border-width border-style border-color

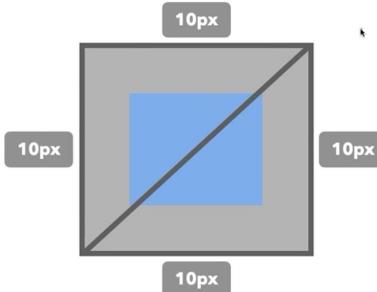
`border: 4px solid black;`

border-width border-style border-color



`border: 10px solid black;`

`border-width` `border-style` `border-color`



요소 테두리 선의 두께

border-width

`medium` 중간 두께

`thin` 얇은 두께

`thick` 두꺼운 두께

`단위` px, em, % 등 단위로 지정

요소 테두리 선의 종류



border-style

`none` 선 없음

`solid` 실선 (일반 선)

`dotted` 점선

`dashed` 파선

`double` 두 줄 선

`groove` 홈이 파여있는 모양

`ridge` 솟은 모양 (groove의 반대)

`inset` 요소 전체가 들어간 모양

`outset` 요소 전체가 나온 모양

요소 테두리 선의 색상을 지정하는 단축 속성

border-color

`black` 검정색

`색상` 선의 색상

`transparent` 투명



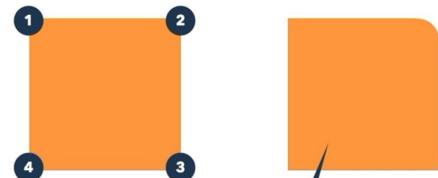
Border-radius

요소의 모서리를 둥글게 깎음

border-radius

0 둥글게 없음

단위 px, em, vw 등 단위로 지정



border-radius: 0 10px 0 0;



Box-sizing



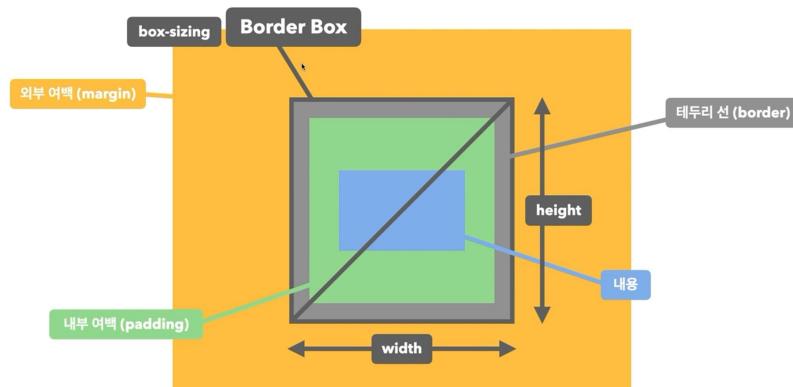
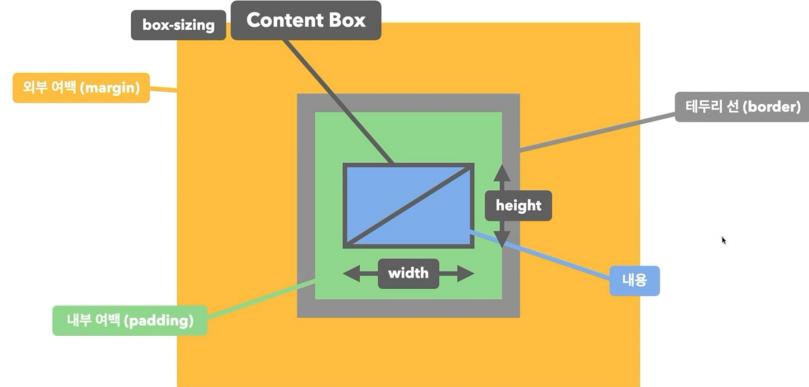


요소의 크기 계산 기준을 지정

box-sizing

content-box 요소의 내용(content)으로 크기 계산

border-box 요소의 내용 + padding + border로 크기 계산



Overflow



요소의 크기 이상으로 내용이 넘쳤을 때, 보여짐을 제어하는 단축 속성

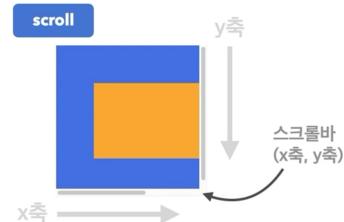
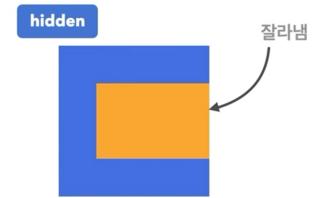
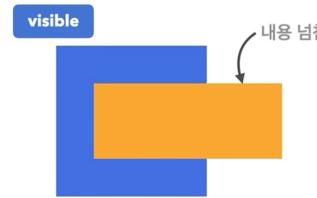
overflow

visible 넘친 내용을 그대로 보여줌

hidden 넘친 내용을 잘라냄

scroll 넘친 내용을 잘라냄, 스크롤바 생성

auto 넘친 내용이 있는 경우에만 잘라내고 스크롤바 생성



요소의 크기 이상으로 내용이 넘쳤을 때, 보여짐을 제어하는 개별 속성들

overflow-x

overflow-y

Display





요소의 화면 출력(보여짐) 특성

각 요소에 이미 지정되어 있는 값

따로 지정해서 사용하는 값

display

block	상자(레이아웃) 요소
inline	글자 요소
inline-block	글자 + 상자 요소
flex	플렉스 박스 (1차원 레이아웃)
grid	그리드 (2차원 레이아웃)
none	보여짐 특성 없음, 화면에서 사라짐
기타	table, table-row, table-cell 등..



Opacity



요소 투명도

opacity

1 불투명

0~1 0부터 1 사이의 소수점 숫자



Visibility

요소를 숨기는 3가지 방법!

- **opacity**(투명하게 만들기)

- 모습만 숨기는 방법 / 속성 남음 / 자리 차지

- **visibility : hidden**

- 모습과 속성을 숨기는 방법 / 자리 차지

- **display : none**

- 그냥 없애 버리는 방법 / 자리도 사라짐



Font

글자의 기울기

font-style

normal

기울기 없음

italic

이탤릭체

oblique

기울어진 글자



글자의 두께(가중치)

font-weight

normal, 400 기본 두께

bold, 700 두껍게

bolder 상위(부모) 요소보다 더 두껍게

lighter 상위(부모) 요소보다 더 얇기

100 ~ 900 100단위의 숫자 9개,
normal과 bold 이외 두께



글자의 크기



font-size

16px 기본 크기

단위 px, em, rem 등 단위로 지정

% 부모 요소의 폰트 크기에 대한 비율

smaller 상위(부모) 요소보다 작은 크기

larger 상위(부모) 요소보다 큰 크기

xx-small ~ xx-large 가장 작은 크기 ~ 가장 큰 크기까지,
7단계의 크기를 지정

한 줄의 높이, 행간과 유사

line-height

normal 브라우저의 기본 정의를 사용

숫자 요소의 글꼴 크기의 배수로 지정

단위 px, em, rem 등의 단위로 지정

% 요소의 글꼴 크기의 비율로 지정



문자에 대한 속성



글자의 색상

color

rgb(0,0,0) 검정색

색상 기타 지정 가능한 색상



문자의 정렬 방식

text-align

- left** 원쪽 정렬
- right** 오른쪽 정렬
- center** 가운데 정렬
- justify** 양쪽 정렬



문자의 장식(선)

text-decoration

화면에 출력!

- none** 장식 없음
- underline** 밑줄
- overline** 윗줄
- line-through** 중앙 선



동해물과 백두산이 마르고 닳도록 하느
님이 보우하사 우리나라 만세 무궁화 삼천리 화
려 강산 대한 사람 대한으로 길이 보전하세

들여쓰기(50px)

문자 첫 줄의 들여쓰기
을수를 사용할 수 있어요!
반대는 내어쓰기(outdent)입니다.

text-indent

- 0** 들여쓰기 없음
- 단위** px, em, rem 등 단위로 지정
- %** 요소의 가로 너비에 대한 비율



배경



요소의 배경 색상

background-color

transparent 투명함

색상 지정 가능한 색상



요소의 배경 이미지 삽입

background-image

none 이미지 없음

url("경로") 이미지 경로



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



x축 y축



요소의 배경 이미지 크기

background-size

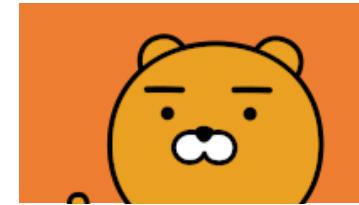
auto 이미지의 실제 크기

단위 px, em, rem 등 단위로 지정

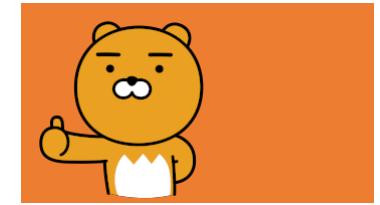
cover 비율을 유지, 요소의 더 넓은 너비에 맞춤

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



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

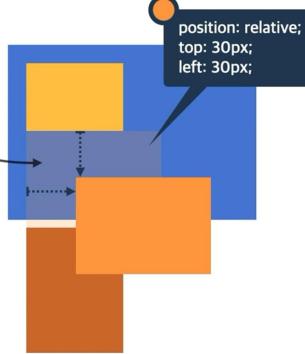
단위 px, em, rem 등 단위로 지정

relative





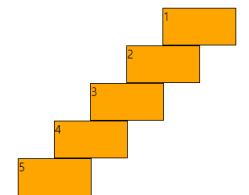
배치 전 자리는
비어 있어요!



`position: relative;
top: 30px;
left: 30px;`

relative 요소 자신을 기준으로 배치!

실습, 동일한 것 만들기!

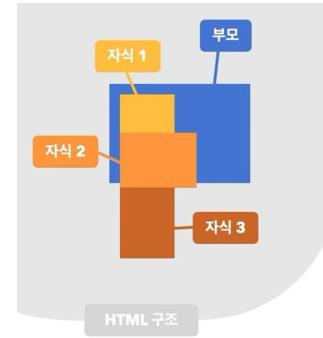


```

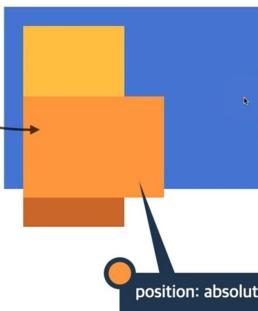
div {
  position: relative;
  width: 100px;
  height: 50px;
  background-color: orange;
  border: 1px solid black;
}
  
```



absolute

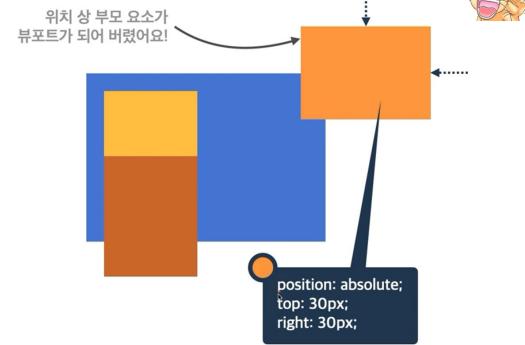


붕~ 뜨면서
요소가 겹쳐요

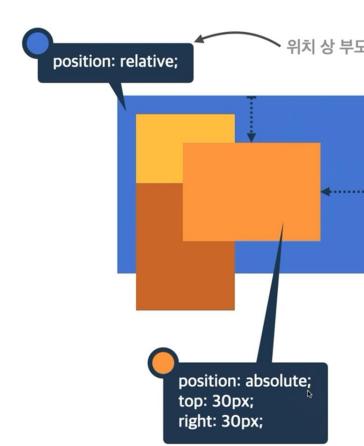


`position: absolute;`

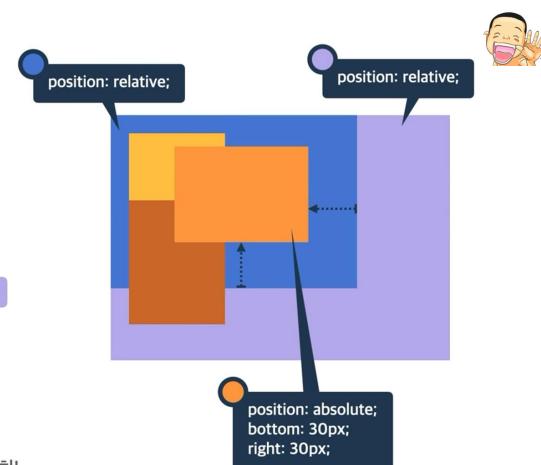
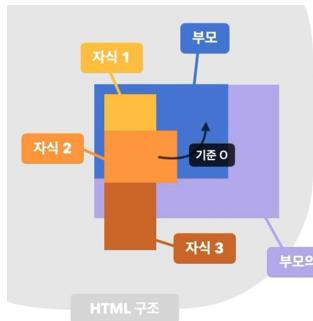
absolute 위치 상 부모 요소를 기준으로 배치!



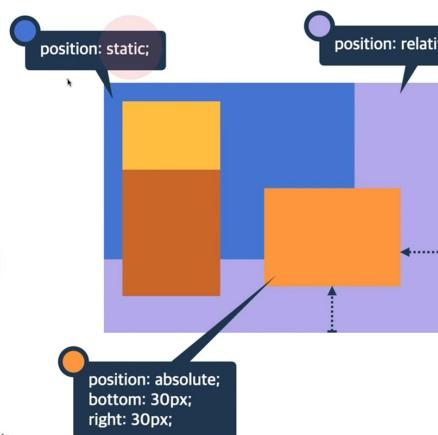
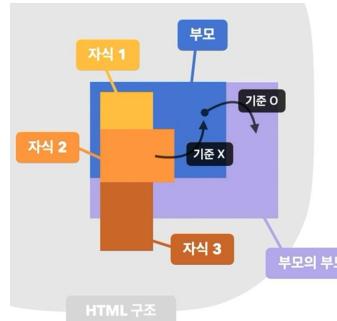
absolute 위치 상 부모 요소를 기준으로 배치!



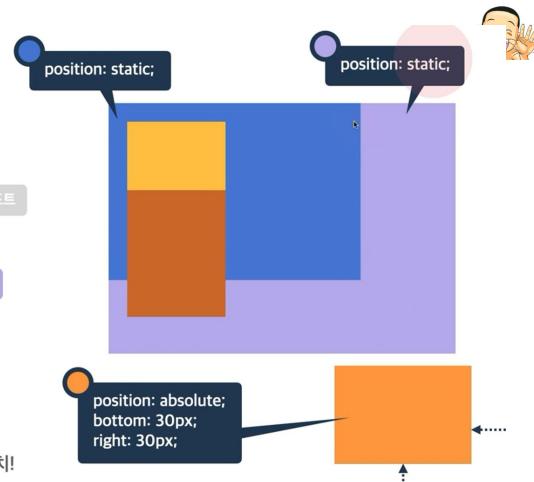
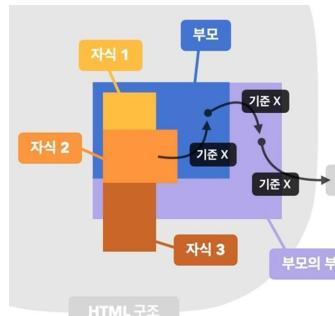
absolute 위치 상 부모 요소를 기준으로 배치!



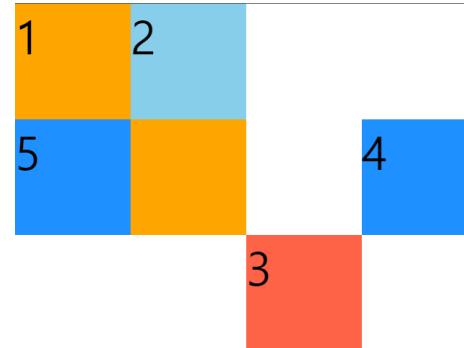
absolute 위치 상 부모 요소를 기준으로 배치!



absolute 위치 상 부모 요소를 기준으로 배치!



실습, 동일한 것 만들기!

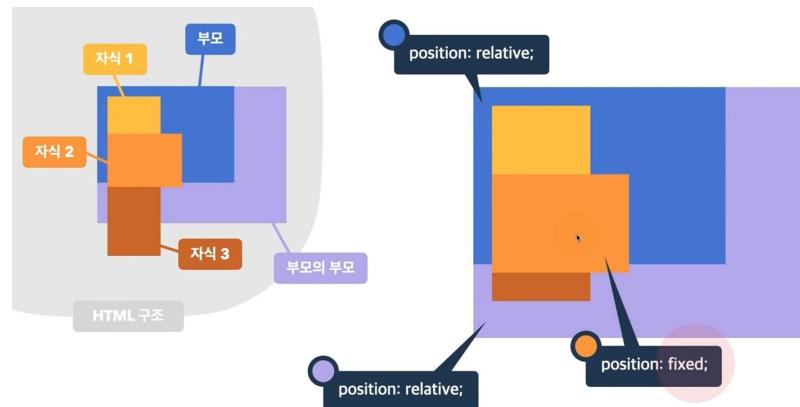


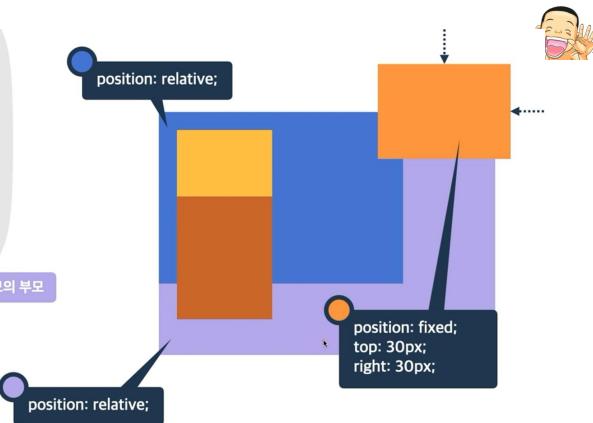
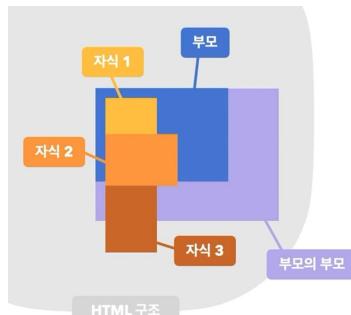
```

body {
  margin: 0px;
  font-size: 4rem;
}
div {
  position: absolute;
}
  
```

- 1은 20 x 20 rem
- 2 ~ 5는 10 x 10 rem

fixed





fixed 뷰포트(브라우저)를 기준으로 배치!

실습, 동일한 것 만들기!

```
body {
  height: 3000px;
}

div {
  position: fixed;
  width: 150px;
  height: 200px;
  background-image: url("");
  border: 1px solid black;
}
```



실습, 동일한 것 만들기!



- 라이언이 스크롤에도 따라 다닐 수 있도록 만들어 주세요!
- Div 에 CSS 속성과 값을 추가하여 위와 동일하게 구현해 주세요!

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.

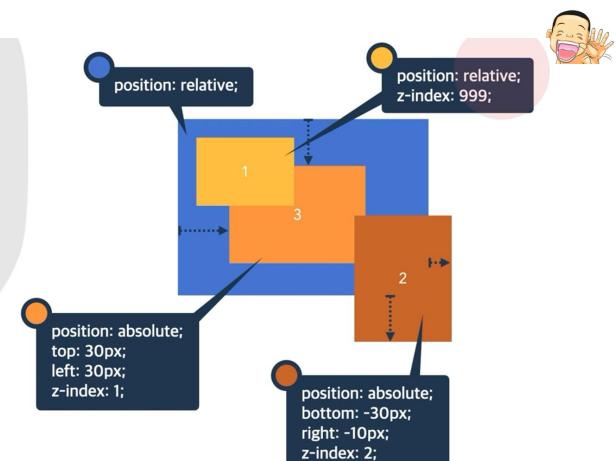
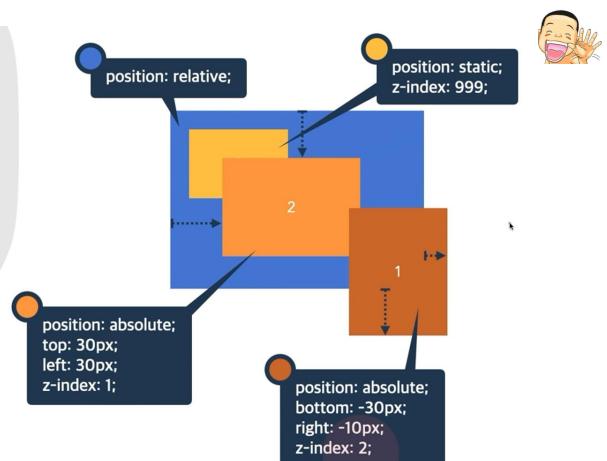
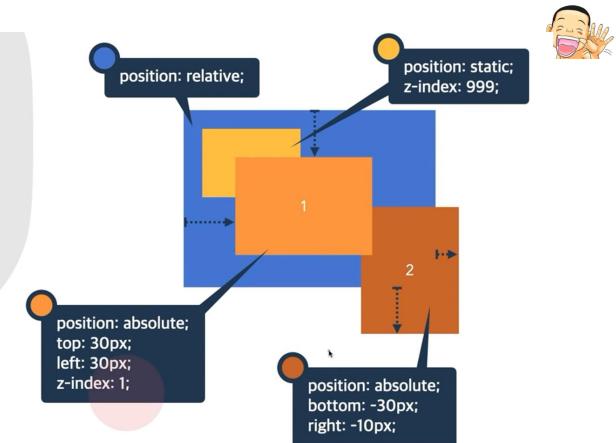




요소의 쌓임 정도를 지정

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;
```

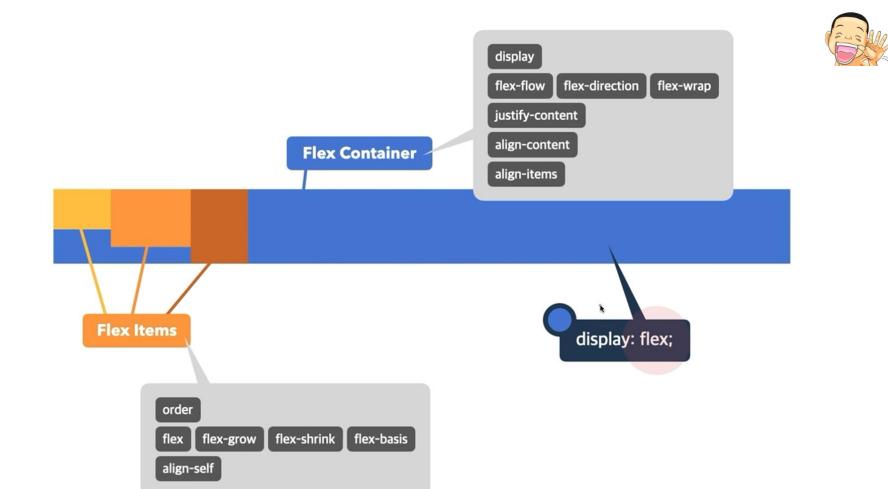
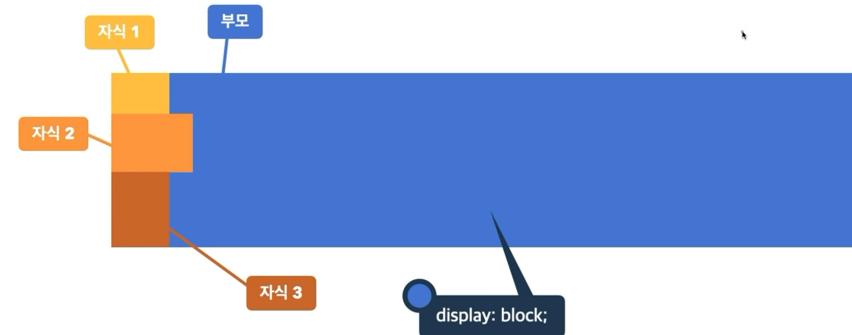


실습, 애벌레 만들기



Flex





Flex
Container 속성



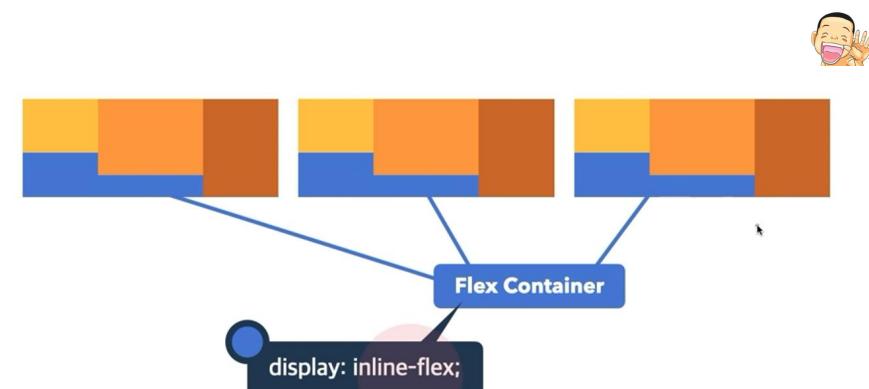
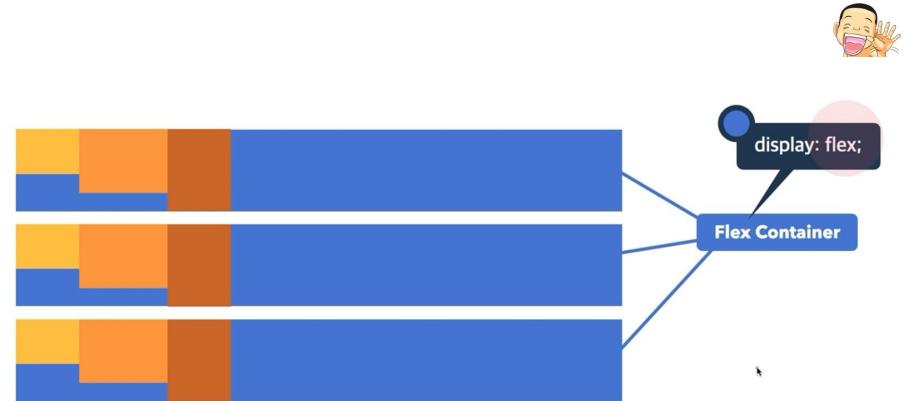


Flex Container의 화면 출력(보여짐) 특성

display

flex 블록 요소와 같이 Flex Container 정의

inline-flex 인라인 요소와 같이 Flex Container 정의



주 축을 설정

flex-direction

row 행 축 (좌 => 우)

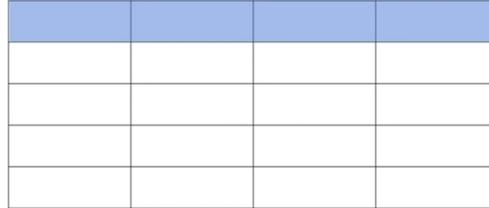
row-reverse 행 축 (우 => 좌)

column 열 축 (위 => 아래)

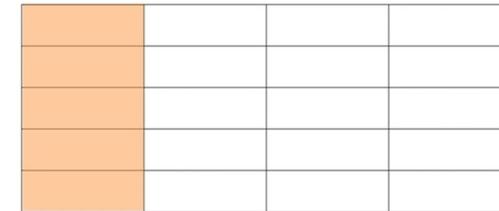
column-reverse 열 축 (아래 => 위)



행, Row



열, Column



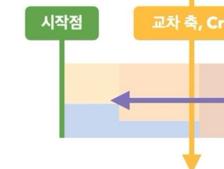
display: flex;
flex-direction: row;



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



시작점



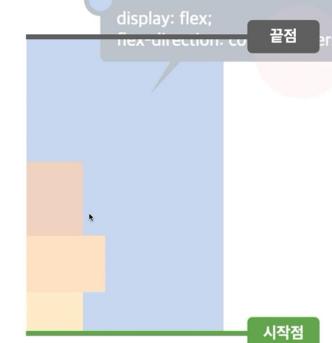
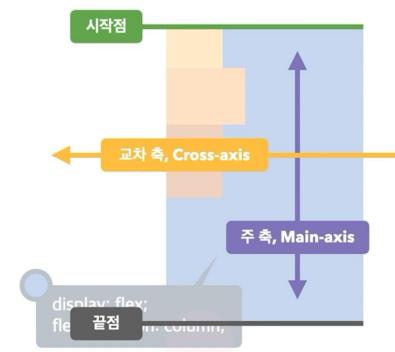
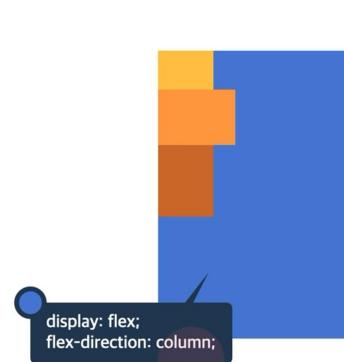
주 축, Main-axis

display: flex;
flex-direction: row;

끝점

끝점

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



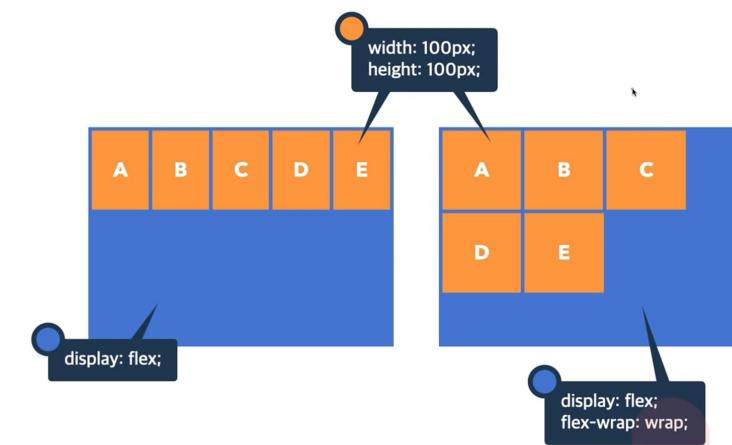
Flex Items 묶음(줄 바꿈) 여부

flex-wrap

nowrap 묶음(줄 바꿈) 없음

wrap 여러 줄로 묶음

wrap-reverse wrap의 반대 방향으로 묶음



 주 축의 정렬 방법

justify-content

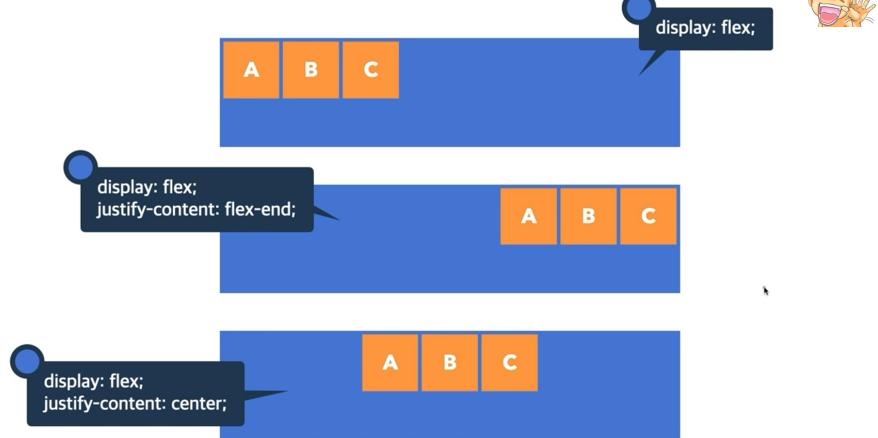
flex-start Flex Items를 시작점으로 정렬

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



 교차 축의 여러 줄 정렬 방법

align-content

stretch Flex Items를 시작점으로 정렬

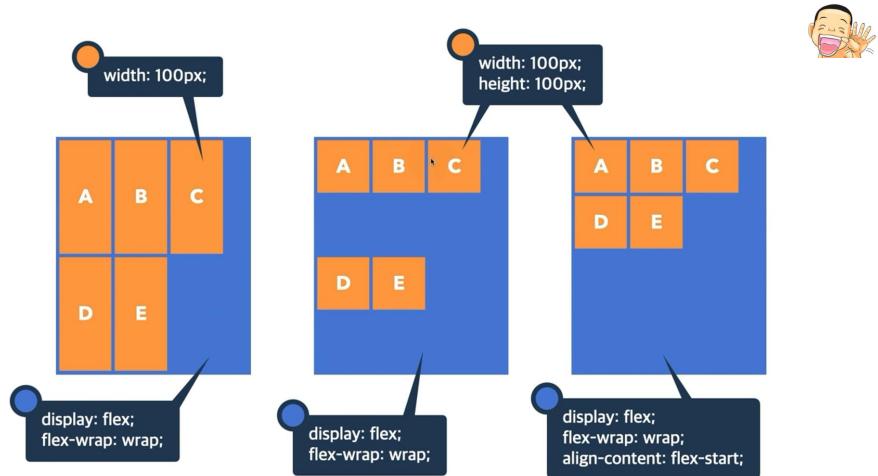
flex-start Flex Items를 시작점으로 정렬

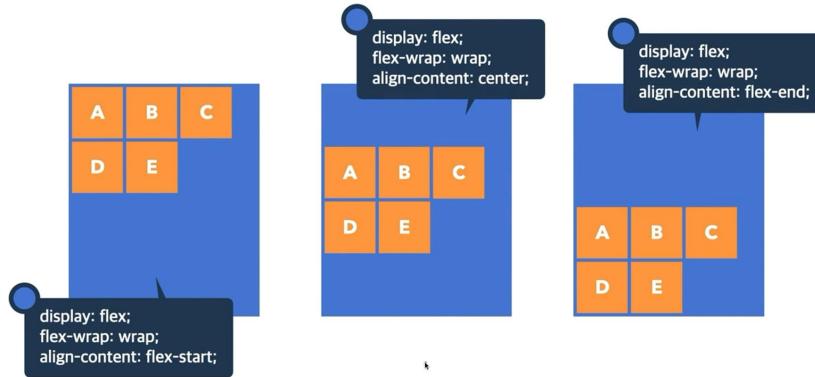
flex-end Flex Items를 끝점으로 정렬

center Flex Items를 가운데 정렬

space-between 각 Flex Item 사이를 균등하게 정렬

space-around 각 Flex Item의 외부 여백을 균등하게 정렬

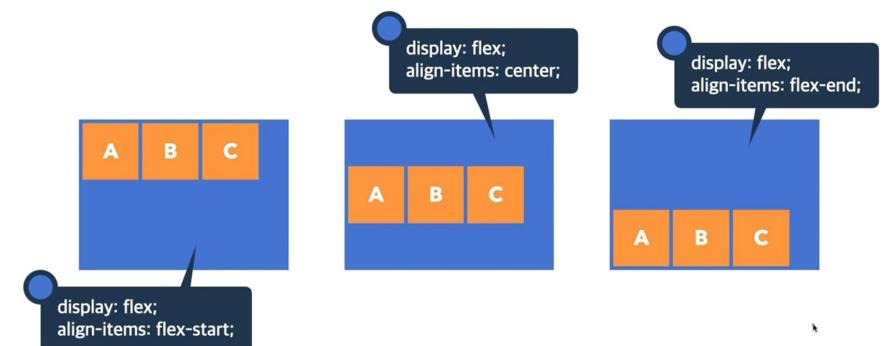
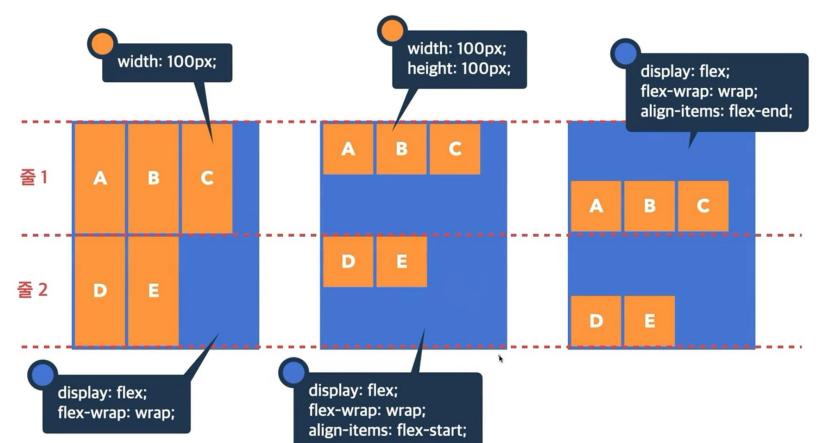




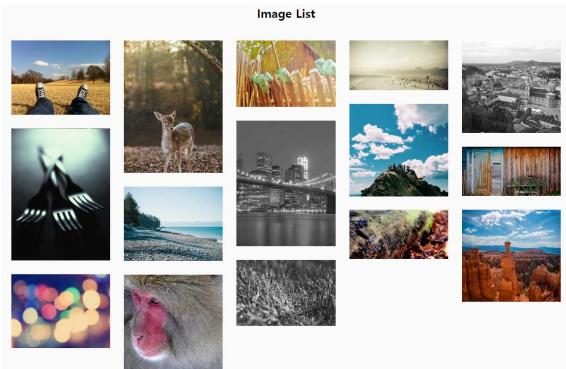
교차 축의 한 줄 정렬 방법

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



Flex 사용 예시



Flex Item 속성

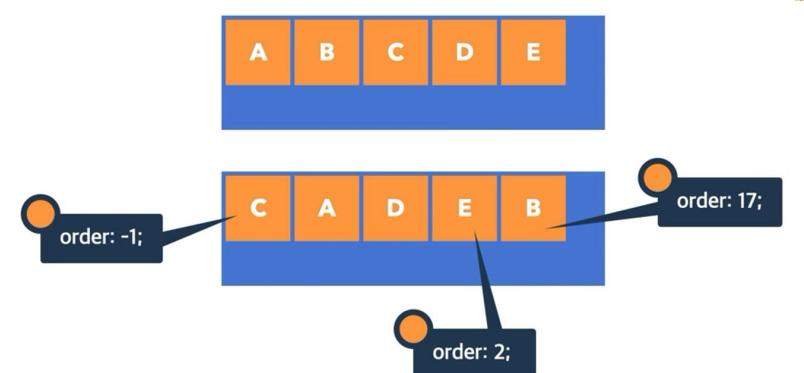


Flex Item의 순서

order

0 순서 없음

숫자 숫자가 작을 수록 먼저





Flex Item의 증가 너비 비율

flex-grow

0 증가 비율 없음

숫자 증가 비율



flex-grow: 1;



flex-grow: 1;



flex-grow: 2;

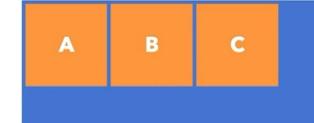


Flex Item의 감소 너비 비율

flex-shrink

1 Flex Container 너비에 따라 감소 비율 적용

숫자 감소 비율



flex-shrink: 0;



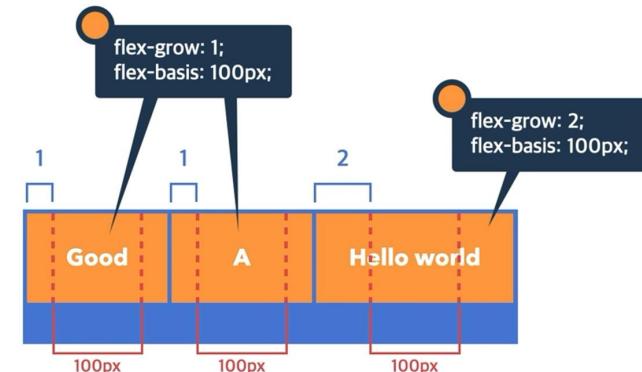
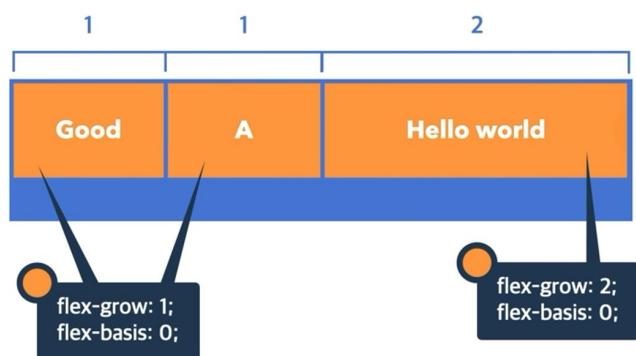
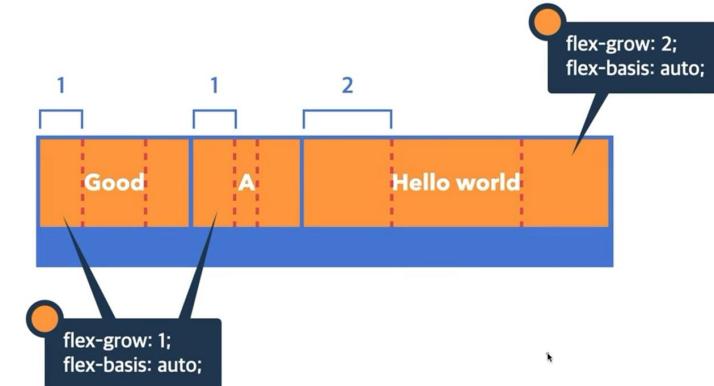


Flex Item의 공간 배분 전 기본 너비

flex-basis

auto 요소의 Content 너비

단위 px, em, rem 등 단위로 지정



실습, flex로 헤더 만들기!



메뉴1

메뉴2

메뉴3

메뉴4

• <https://heartfelt-stroopwafel-c50d55.netlify.app/>

• 위의 사이트와 똑같은 헤더를 만들어 주세요!

• 질문 대환영입니다 😊

• 꼼수는 피해주세요 ;)

```
<style>
  ul {
    list-style: none;
    padding-left: 0px;
  }
</style>
```

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 x 100) 요소를 클릭하면, 오른쪽(300 x 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;



2D 변환 함수

px

`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)

deg

`rotate(degree)` 회전(각도)

`skew(x, y)` 기울임(x축, y축)

`skewX(x)` 기울임(x축)

`skewY(y)` 기울임(y축)

`matrix(n,n,n,n,n)` 2차원 변환 효과



`transform: translateY(40px);`

`transform: translate(40px, 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 변화 함수



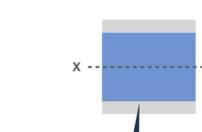
`transform: rotate(45deg);`



`transform: rotateY(45deg);`



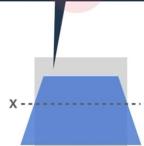
`transform: rotateX(45deg);`



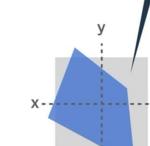
원근법 함수는
제일 앞에 작성해야 합니다!!

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

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



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



하위 요소를 관찰하는 원근 거리를 지정

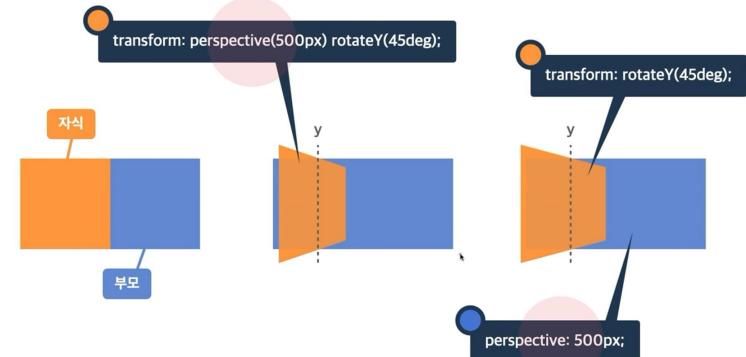
perspective

단위 px 등 단위로 지정



perspective 속성과 함수 차이점

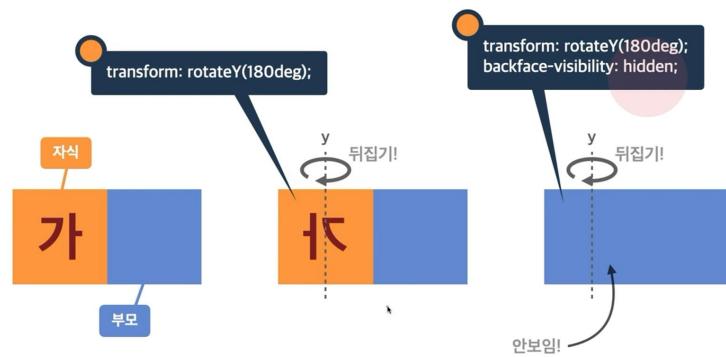
속성 / 함수	적용 대상	기준점 설정
<code>perspective: 600px;</code>	관찰 대상의 부모	<code>perspective-origin</code>
<code>transform: perspective(600px)</code>	관찰 대상	<code>transform-origin</code>



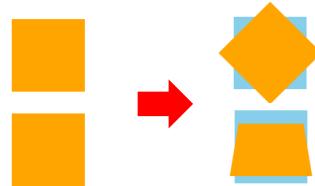
3D 변환으로 회전된 요소의 뒷면 숨김 여부

backface-visibility

- visible 뒷면 보임
- hidden 뒷면 숨김



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?





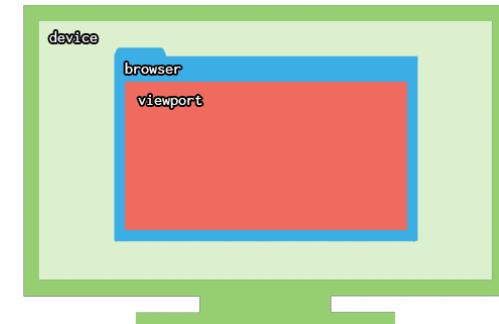
<https://youtu.be/bdWnD6VqLAs>

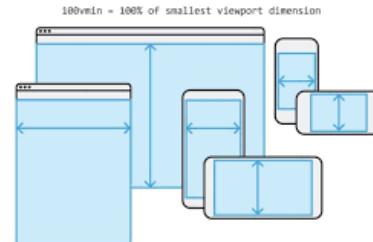
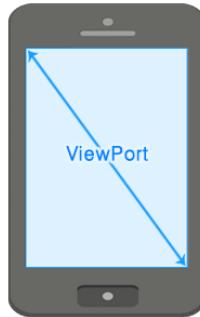


Viewport!



- 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?





@media (미디어 쿼리)

- 서로 다른 미디어 타입(디바이스 화면)에 따라 별도의 CSS를 지정하게 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 500px를 기준으로 500px 이상일 경우 <body> 태그의 배경색이 하늘색, 300px 이하일 경우 배경색이 오렌지
- 화면 높이 300px를 기준으로 300px 이하일 경우 <p> 태그의 폰트 사이즈를 변경 하고, 500px 이상일 경우 배경색이 초록색



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



