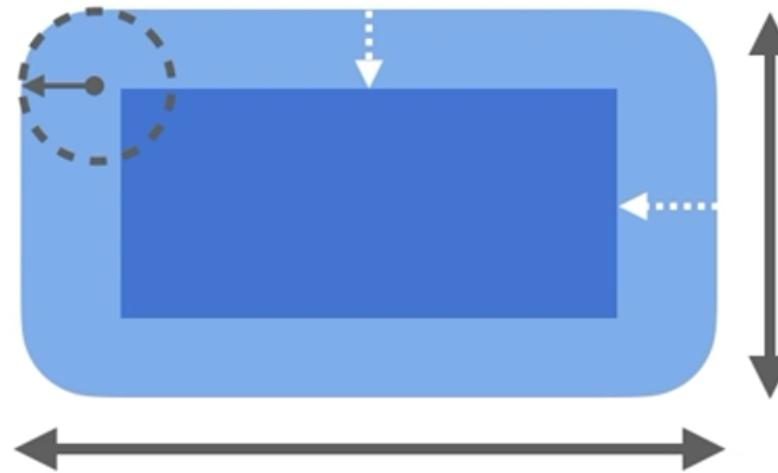


**CSS**



# CSS의 속성

박스 모델  
글꼴, 문자  
배경  
배치  
플렉스(정렬)  
전환  
변환  
띄움  
애니메이션  
그리드  
다단  
필터



# Font

## 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

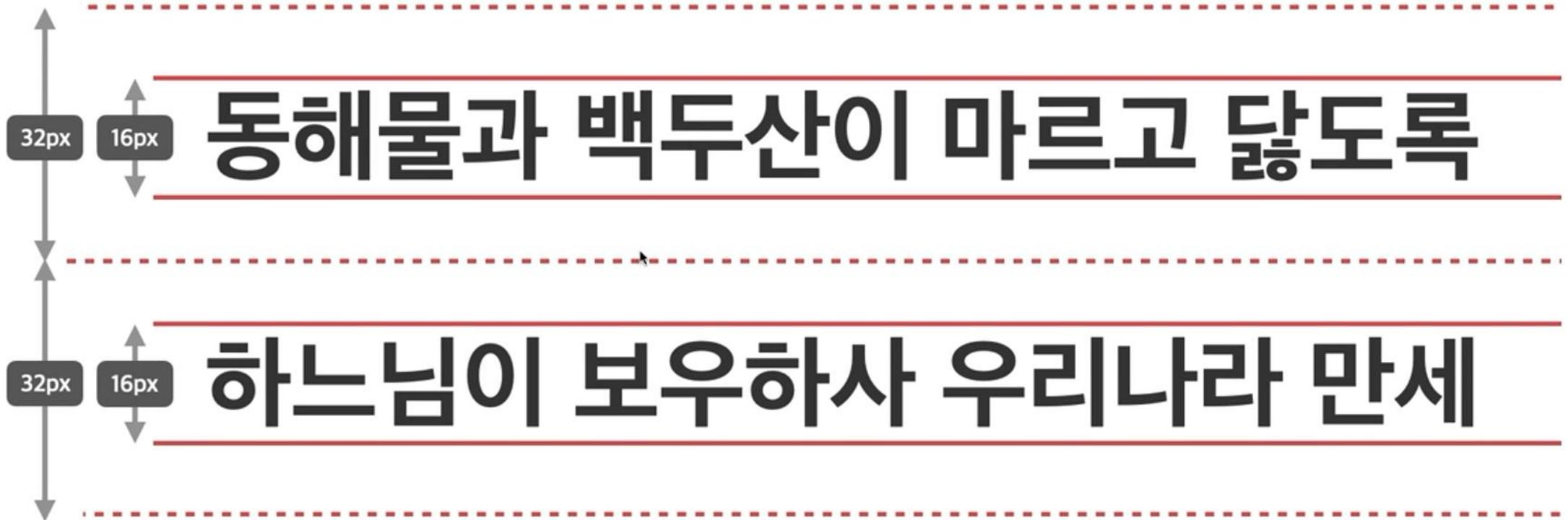
px, em, rem 등의 단위로 지정

%

요소의 글꼴 크기의 비율로 지정

동해물과 백두산이 마르고 닳도록  
하느님이 보우하사 우리나라 만세

```
font-size: 16px;  
line-height: 32px;  
/* line-height: 2; */  
/* line-height: 200%; */
```



```
font-size: 16px; → 2배 차이  
line-height: 32px;  
/* line-height: 2; */  
/* line-height: 200%; */
```

**font-family: 글꼴1, "글꼴2", ... 글꼴계열;**

글꼴(서체) 지정

필수로 작성!

띄어쓰기 등 특수문자가 포함된  
글꼴 이름은 큰 따옴표로 묶어야 합니다~

Hello World!

serif

바탕체 계열

Hello World!

sans-serif

고딕체 계열

Hello World!

monospace

고정너비(가로폭이 동등) 글꼴 계열

*Hello World!*

cursive

필기체 계열

**EMPIRE**

fantasy

장식 글꼴 계열

# 구글 또는 눈누 폰트 사용하기

- 구글 폰트
  - <https://fonts.googleapis.com/?subset=korean>
  - 링크 방식 + 폰트 패밀리
- 눈누 폰트
  - <https://noonnu.cc/>
  - font-face + 폰트 패밀리

# 실습 폰트 적용하기

- 2 개의 <p> 태그를 활용해서 내용을 만들기
- 각각 p 태그에 원하는 폰트를 구글, 눈누에서 받아서 적용하기
- 서체 지정 양식을 잘 작성하여, 구글 폰트 → 눈누 폰트 → 글꼴 계열 순으로 지정
- 구글 폰트 링크 삭제 후, 페이지 확인
- 구글, 누눈 폰트 링크 삭제 후, 페이지 확인

# 문자에 대한 속성

글자의 색상

color

rgb(0,0,0) 검정색

색상 기타 지정 가능한 색상

## 문자의 정렬 방식

# text-align

- ☰ **left** 왼쪽 정렬
- ☰ **right** 오른쪽 정렬
- ☰ **center** 가운데 정렬
- ☰ **justify** 양쪽 정렬

## 문자의 장식(선)

# text-decoration

화면에 출력!

동해물과 백두산이 마르고 닳도록

동해물과 백두산이 마르고 닳도록

동해물과 백두산이 마르고 닳도록

~~동해물과 백두산이 마르고 닳도록~~

**none** 장식 없음

**underline** 밑줄

**overline** 윗줄

**line-through** 중앙 선

박스 모델

# 글자와 상자

요소가 화면에 출력되는 특성, 크게 2가지로 구분.

인라인(Inline) 요소 : 글자를 만들기 위한 요소들.

블록(Block) 요소 : 상자(레이아웃)를 만들기 위한 요소들.

Inline 요소

```
<span>Hello</span>  
<span>World</span>
```

<span></span>

대표적인 인라인 요소!  
본질적으로 아무것도 나타내지 않는,  
콘텐츠 영역을 설정하는 용도.



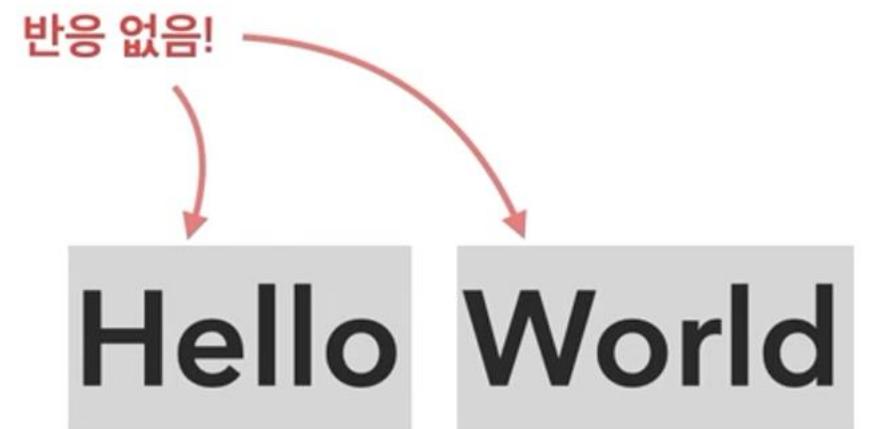
Hello World

요소가 수평으로 쌓임

요소의 가로 너비를 지정하는 CSS 속성

```
<span style="width: 100px;">Hello</span>
<span style="height: 100px;">World</span>
```

요소의 세로 너비를 지정하는 CSS 속성



```
<span>Hello</span>  
<span>World</span>
```

<span></span>  
대표적인 인라인 요소!  
본질적으로 아무것도 나타내지 않는,  
콘텐츠 영역을 설정하는 용도.

auto

포함한 콘텐츠 크기만큼 자동으로 줄어듬!

포함한 콘텐츠 크기만큼  
자동으로 줄어듬!

auto

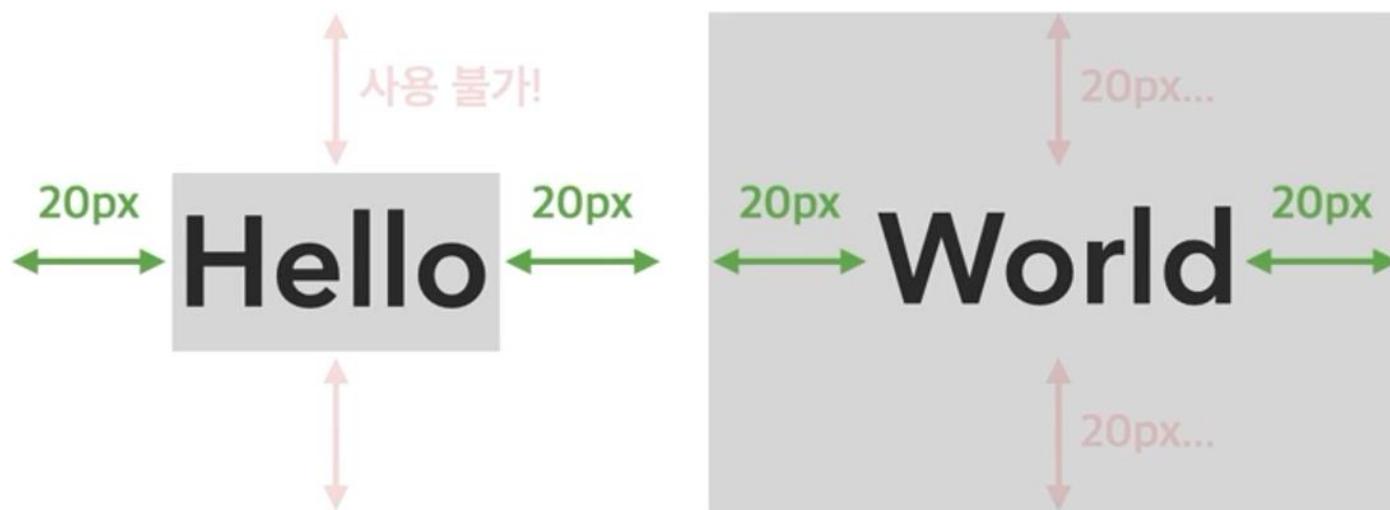
Hello World

요소의 외부 여백을 지정하는 CSS 속성

```
<span style="margin: 20px 20px;">Hello</span>
```

```
<span style="padding: 20px 20px;">World</span>
```

요소의 내부 여백을 지정하는 CSS 속성



인라인 요소

불가!

블록 요소

```
<span><div></div></span>
```

```
<span><span></span></span>
```

가능!

Block 요소

```
<div>Hello</div>  
<div>World</div>
```

```
<div></div>
```

대표적인 블록 요소!  
본질적으로 아무것도 나타내지 않는,  
콘텐츠 영역을 설정하는 용도.

Hello  
World

요소가  
수직으로 쌓임

```
<div>Hello</div>  
<div>World</div>
```



```
<div></div>
```

대표적인 블록 요소!  
본질적으로 아무것도 나타내지 않는,  
콘텐츠 영역을 설정하는 용도.

auto

부모 요소의 크기만큼 자동으로 늘어남!

```
Hello  
World
```

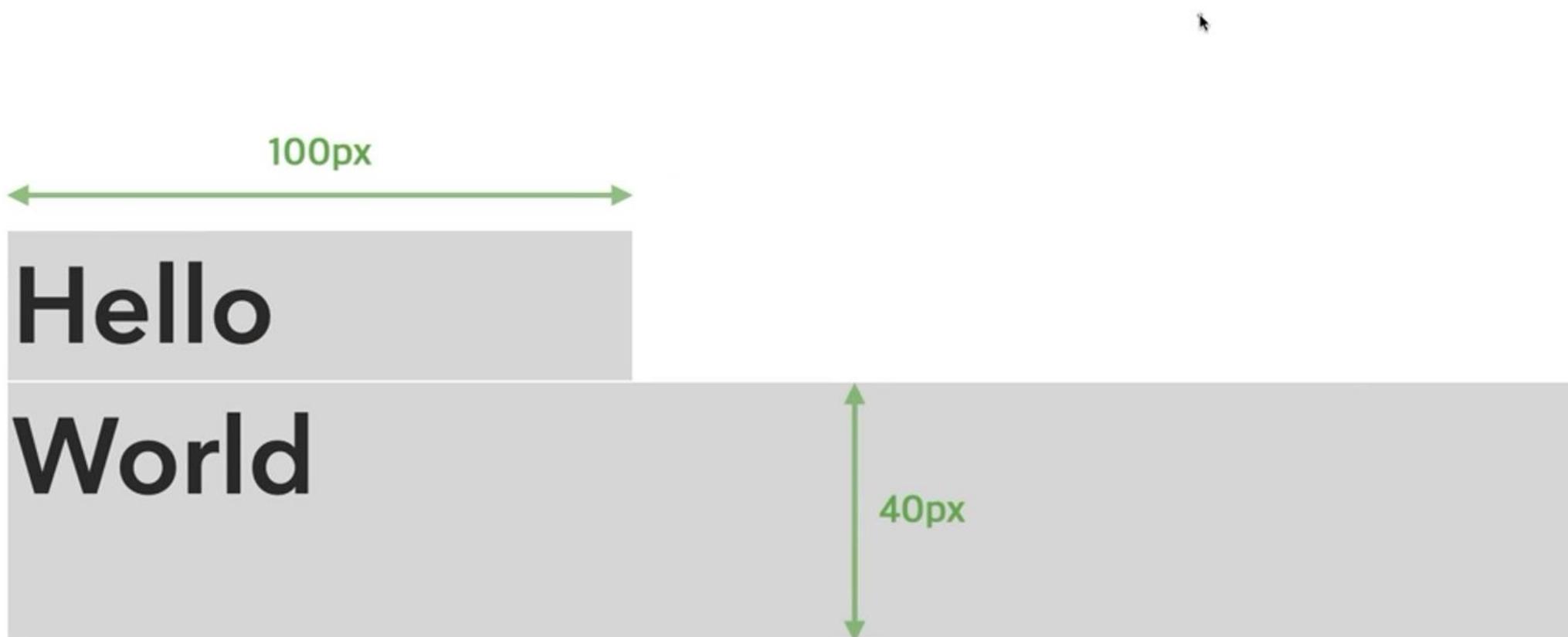
포함한 콘텐츠 크기만큼  
자동으로 줄어듬!

auto

요소의 가로 너비를 지정하는 CSS 속성

```
<div style="width: 100px;">Hello</div>
<div style="height: 40px;">World</div>
```

요소의 세로 너비를 지정하는 CSS 속성

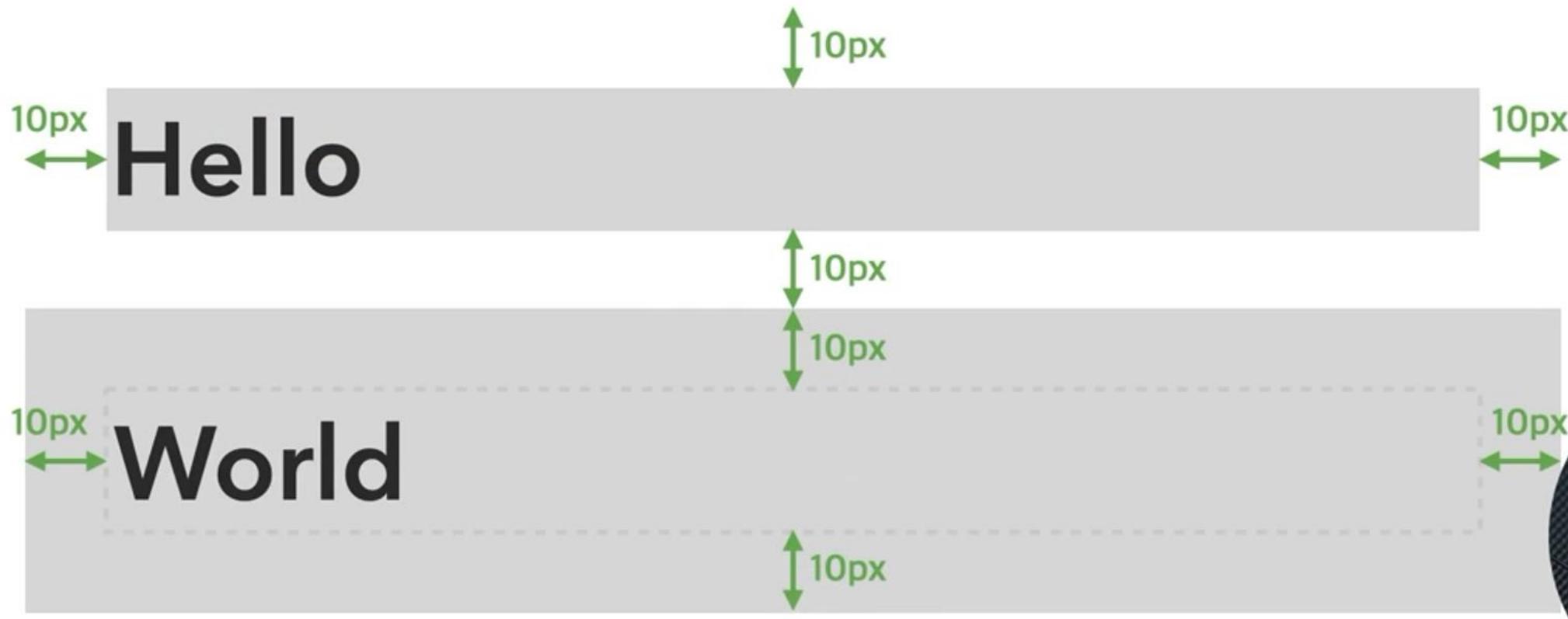


요소의 외부 여백을 지정하는 CSS 속성

```
<div style="margin: 10px;">Hello</div>
```

```
<div style="padding: 10px;">World</div>
```

요소의 내부 여백을 지정하는 CSS 속성



블록 요소

<div><div></div></div>

가능!

<div><span></span></div>

가능!

인라인 요소

# inline, block

**inline**



**block**



# Inline-block?

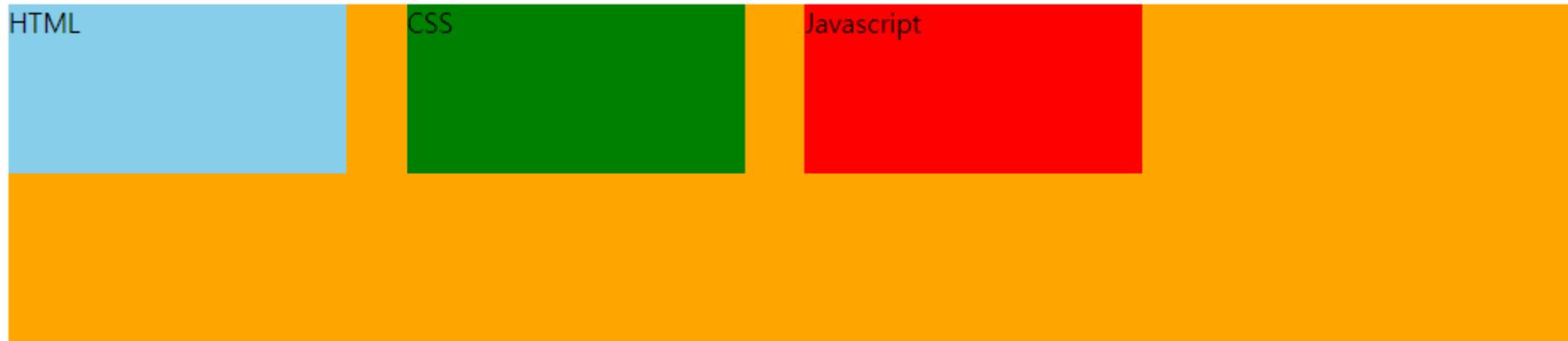


- Block 요소가 수직으로만 쌓이는 문제점을 해결하고자 나온 개념
- 디자인이 중요시 되면서 블록을 가로로 배열하는 경우가 많이 생겼는데, span 태그는 inline 요소라서 해당 구역을 꾸미기가 어려움!
- 따라서, 둘의 특징을 합쳐서 만든 inline-block!

# inline, block, inline-block

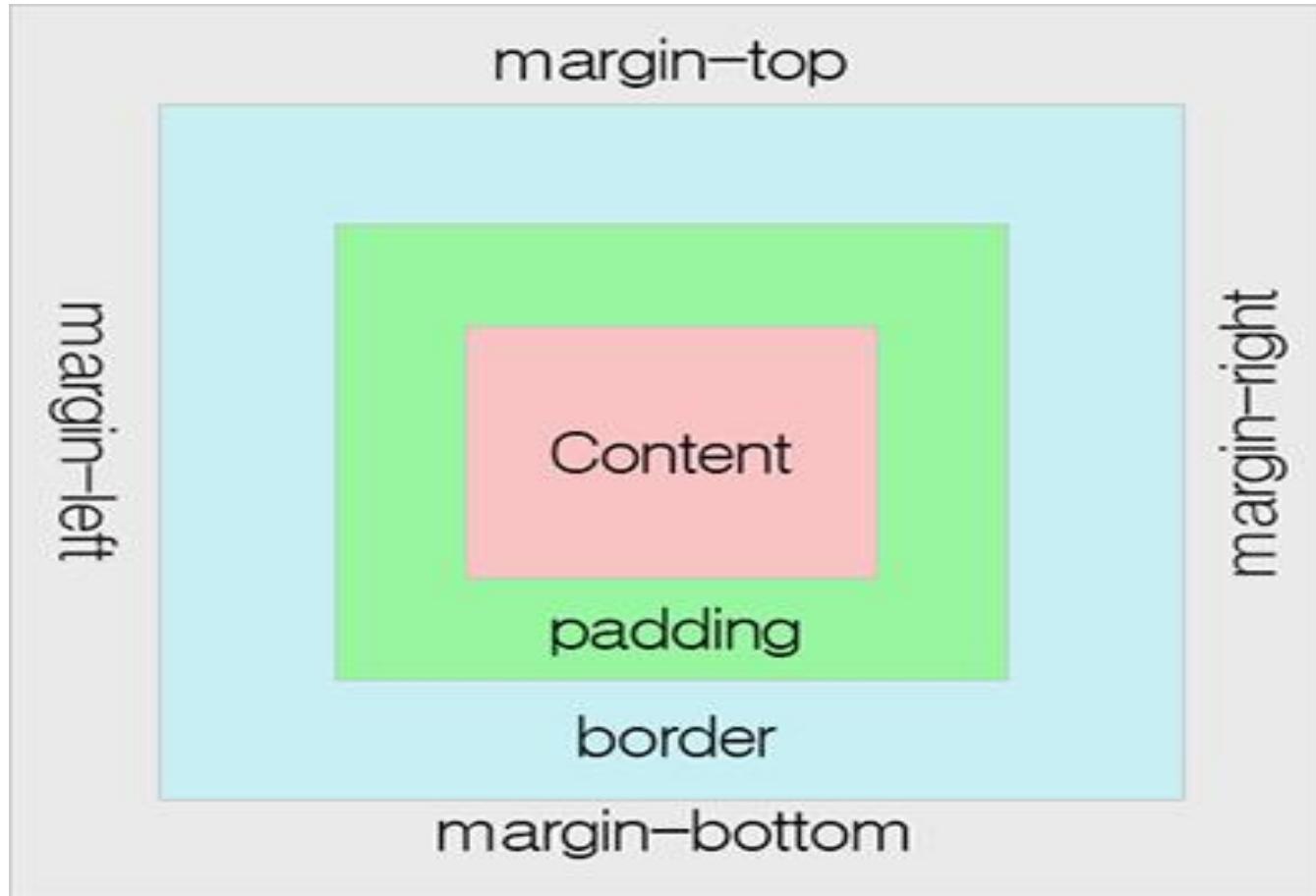
	inline	block	inline-block
기본 넓이	컨텐츠 만큼	부모의 넓이	컨텐츠 만큼
width, height	무시	적용	적용
가로 공간 차지	공유	독점	공유
margin	가로만	가로, 세로 전부 *세로 상쇄	가로, 세로 전부
padding	가로만, *세로는 배경색만	가로, 세로 전부	가로, 세로 전부

# 실습 아래 그림처럼 만들기!



1. 오렌지 박스의 크기는 width : auto, height : 200px;
2. 안의 요소의 크기는 200px 100px
3. 각각 요소의 간격은 30px

# margin & padding



margin : 다른 요소와의 거리, 바깥쪽 여백  
padding: border 와 content 간의 안쪽 여백

## 요소의 외부 여백(공간)을 지정하는 단축 속성

가로(세로) 너비가 있는 요소의  
가운데 정렬에 활용해요!

# margin

음수를 사용할 수 있어요!

0

외부 여백 없음

auto

브라우저가 여백을 계산

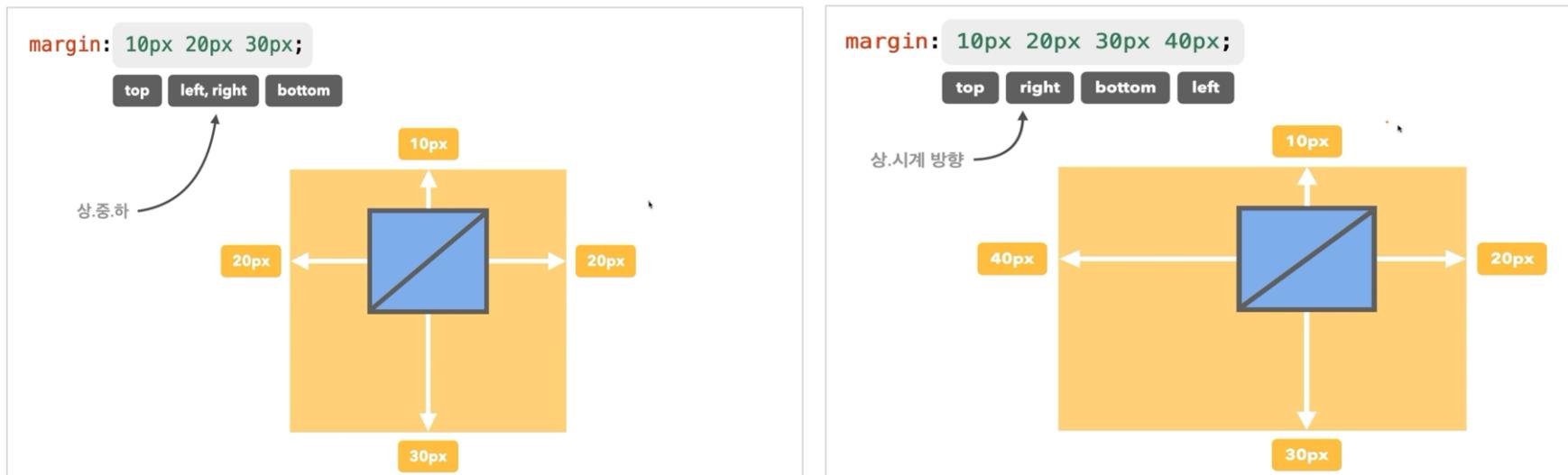
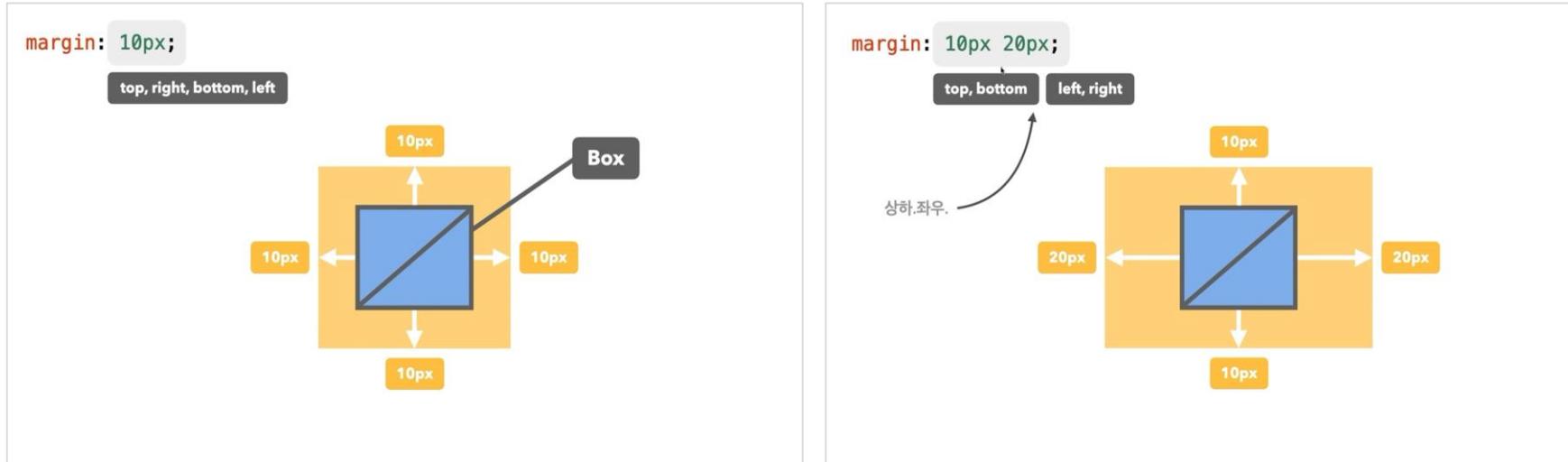
단위

px, em, vw 등 단위로 지정

%

부모 요소의 가로 너비에 대한 비율로 지정

# 단축속성



# 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

**margin-top**

**margin-bottom**

**margin-left**

**margin-right**

요소의 내부 여백(공간)을 지정하는 단축 속성

# padding

요소의 크기가 커져요!

0

내부 여백 없음

단위

px, em, vw 등 단위로 지정

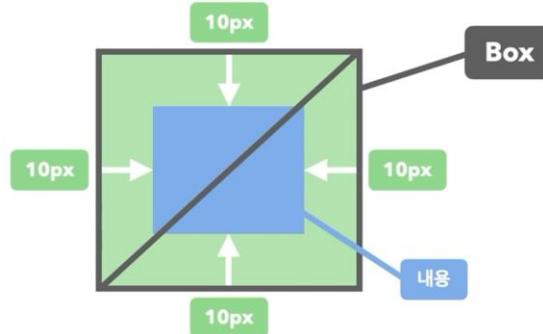
%

부모 요소의 가로 너비에 대한 비율로 지정

# Padding 단축 속성

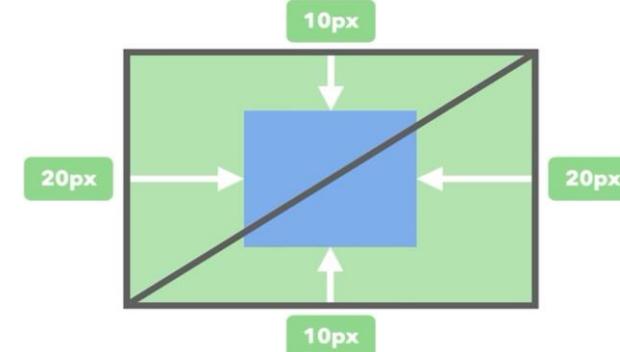
padding: 10px;

top, right, bottom, left



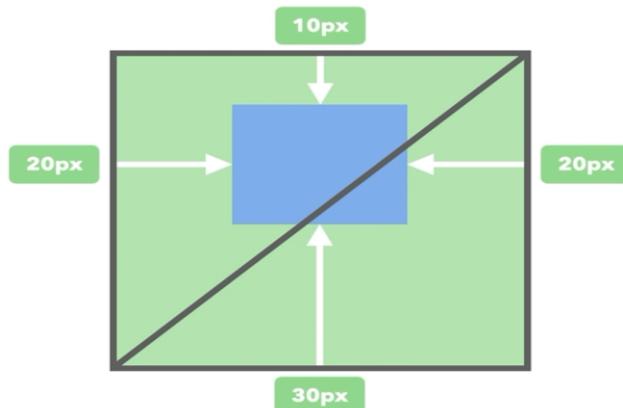
padding: 10px 20px;

top, bottom | left, right



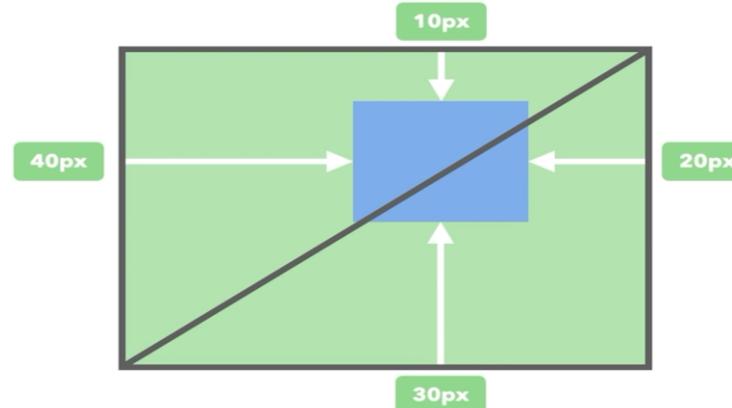
padding: 10px 20px 30px;

top | left, right | bottom



padding: 10px 20px 30px 40px;

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

**padding-top**

**padding-bottom**

**padding-left**

**padding-right**

# Width, Height

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

표현 단위

# 단위

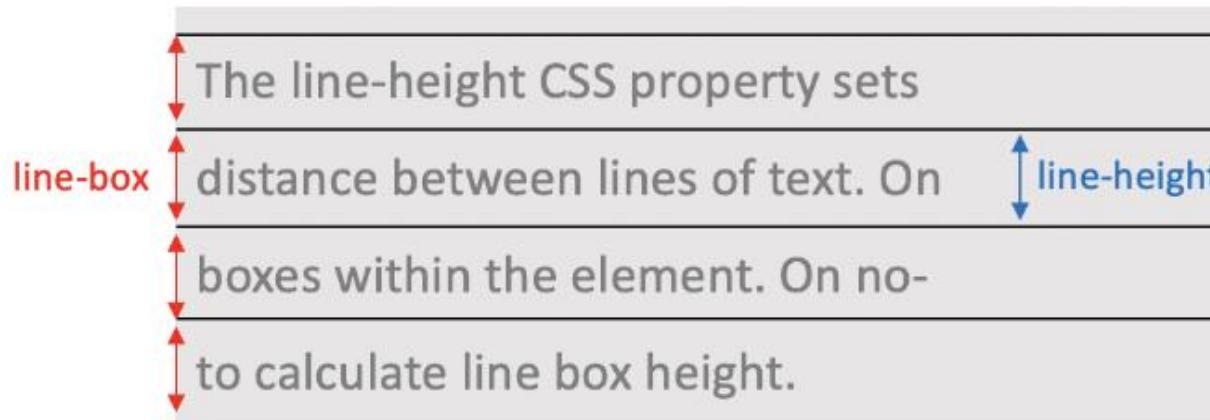
<b>px</b>	픽셀
<b>%</b>	상대적 백분율
<b>em</b>	요소의 글꼴 크기
<b>rem</b>	루트 요소(html)의 글꼴 크기
<b>vw</b>	뷰포트 가로 너비의 백분율
<b>vh</b>	뷰포트 세로 너비의 백분율

# Calc()

- 사용자가 원하는 크기 값을 계산하여 적용
- Ex
  - Calc(100vh - 20vw)
  - Calc(1920px - 10vw)

# line-height

- 영역 요소 내부 컨텐츠 글자의 줄 높이
- Box model의 크기 단위 적용 가능(font 에도 적용 가능!)
- 컨텐츠가 1줄인 경우 box height 와 line height 를 동일하게 하면 세로 중앙 정렬 효과!



# 실습 calc 및 line-height 사용하기

- 현재 화면 넓이의 1/10인 길이를 변으로 하는 div 정사각형 만들기, 배경색은 예쁘게!
- div 안에는 KDT 라는 컨텐츠를 삽입하고 세로 중앙 정렬 하기
- 새로운 html 파일을 만들어서 하기!

# Border

요소의 크기가 커져요!

**border: 선-두께 선-종류 선-색상;**

**border-width**

요소의 테두리 선을 지정하는 단축 속성

**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

# Border-width, 단축 속성 정리

border-width: `top, right, bottom, left` ;

border-width: `top, bottom` `left, right` ;

border-width: `top` `left, right` `bottom` ;

border-width: `top` `right` `bottom` `left` ;

## 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

# Border-style, 단축 속성 정리

border-style: `top, right, bottom, left` ;

border-style: `top, bottom` `left, right` ;

border-style: `top` `left, right` `bottom` ;

border-style: `top` `right` `bottom` `left` ;

요소 테두리 선의 색상을 지정하는 단축 속성

# border-color

black 검정색

색상 선의 색상

transparent 투명

# Border-color, 단축 속성 정리

border-color: top, right, bottom, left ;

border-color: top, bottom left, right ;

border-color: top left, right bottom ;

border-color: top right bottom left ;

요소의 테두리 선을 지정하는 기타 속성들

**border-방향**

**border-방향-속성**

**border-top:** 두께 종류 색상;  
**border-top-width:** 두께;  
**border-top-style:** 종류;  
**border-top-color:** 색상;

**border-right:** 두께 종류 색상;

**border-right-width:** 두께;

**border-right-style:** 종류;

**border-right-color:** 색상;

# 실습 테두리 그리기

- Border 속성으로 테두리 적용하기 x 4개(각각 클래스 부여)
- Border-width 속성으로 4 방향 테두리 두께 변경하기
- Border-style 속성으로 4 방향 테두리 스타일 변경하기
- Border-color 속성으로 4 방향 테두리 색상 변경하기
- Border-방향 속성으로 모든 방향의 테두리 스타일이 다른 div 완성하기

# Border-radius

요소의 모서리를 둥글게 깎음

# border-radius

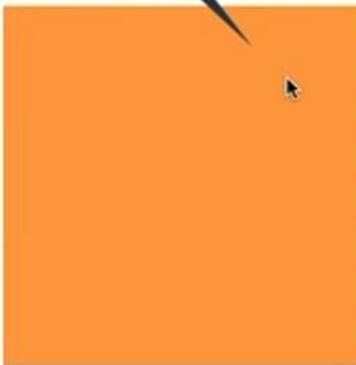
0

둥글게 없음

단위

px, em, vw 등 단위로 지정

`border-radius: 0;`

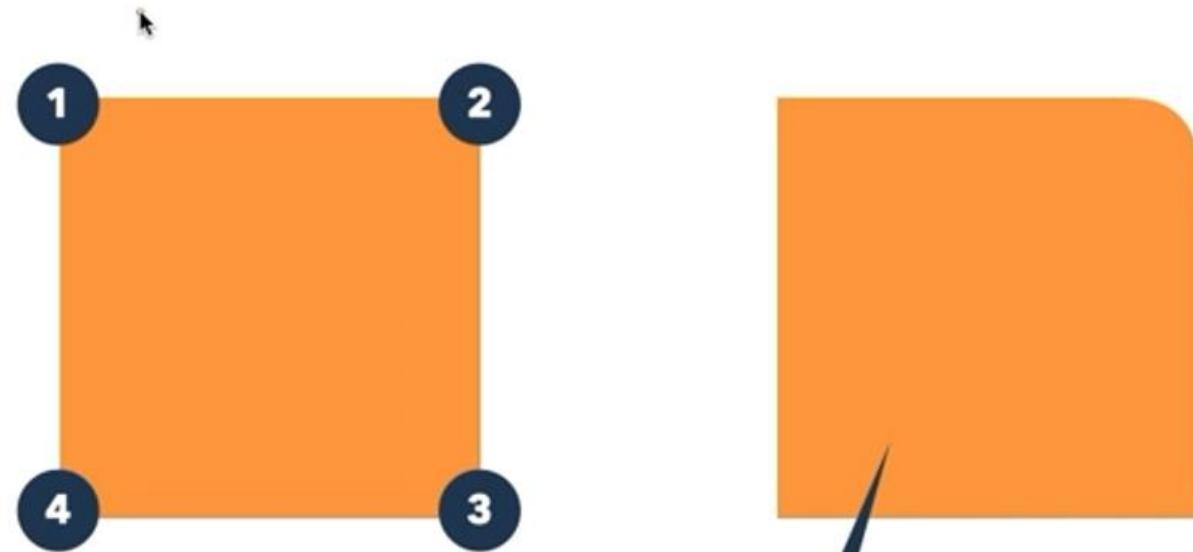


`border-radius: 10px;`



`border-radius: 0 10px 0 0;`





border-radius: 0 10px 0 0;

# 실습 border-radius

- 방금 전 실습인 테두리 그리기에 border-radius 를 모든 방향에 다르게 적용 하기!

# Box-sizing

요소의 크기 계산 기준을 지정

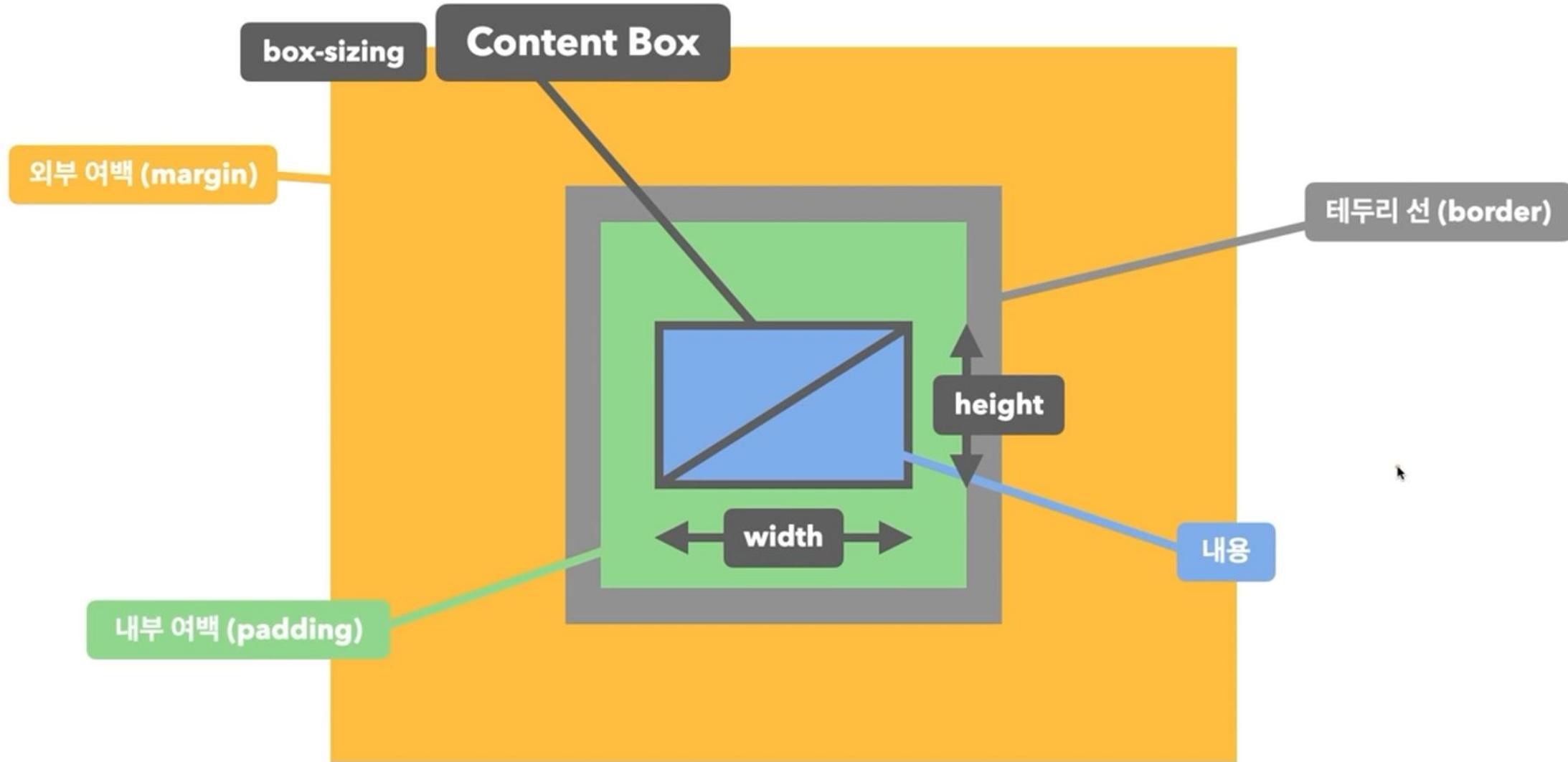
# box-sizing

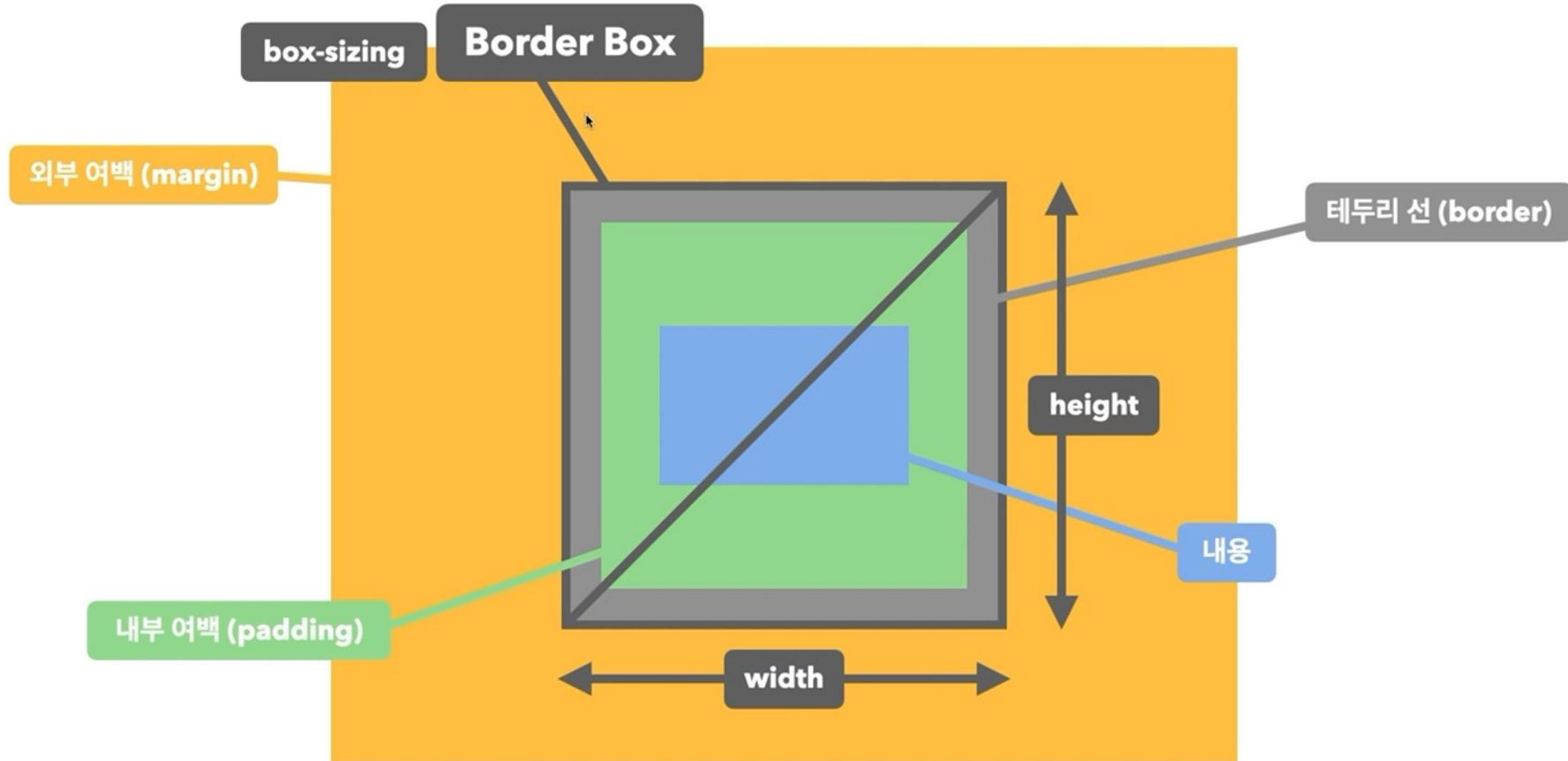
**content-box**

요소의 내용(content)으로 크기 계산

**border-box**

요소의 내용 + padding + border로 크기 계산



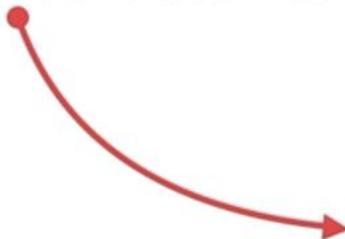


# 실습 크기 비교하기

- Width 100px, height 100px, padding 20px, border 15px solid red, 배경 오렌지인 div 2개 선언
- 하나는 content-box, 하나는 border-box로 속성을 주고 둘의 크기를 비교

## 요소의 화면 출력(보여짐) 특성

각 요소에 이미 지정되어 있는 값



**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..

# display

# Opacity

## 요소 투명도

# opacity

1

불투명

0~1

0부터 1 사이의 소수점 숫자

opacity: 0.07;



opacity: 1;



opacity: 0.4;

opacity: 0.7;

# 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

## *attr1.css*

### **display:none, visibility:hidden, opacity:0**



```
.d-none {  
    /* display: none; */  
    background-color: red;  
}  
.invisible {  
    /* visibility: hidden; */  
    background-color: orangered;  
}  
  
.opacity-0 {  
    /* opacity: 0; */  
    background-color: tomato;  
}
```

## *attr1.html*

```
<div class="square">original</div>  
<div class="square d-none">d-none</div>  
<div class="square invisible">invisible</div>  
<div class="square opacity-0">opacity-0</div>  
<div class="square">original</div>
```

# **실습** 요소 사라지게 하기

- div 요소의 크기를 200px 200px 배경 파란색으로 설정
- 해당 div에 마우스가 올라가면 요소가 사라지도록 만들기

# Overflow

요소의 크기 이상으로 내용이 넘쳤을 때, 보여짐을 제어하는 단축 속성

# overflow

**visible**

넘친 내용을 그대로 보여줌

**hidden**

넘친 내용을 잘라냄

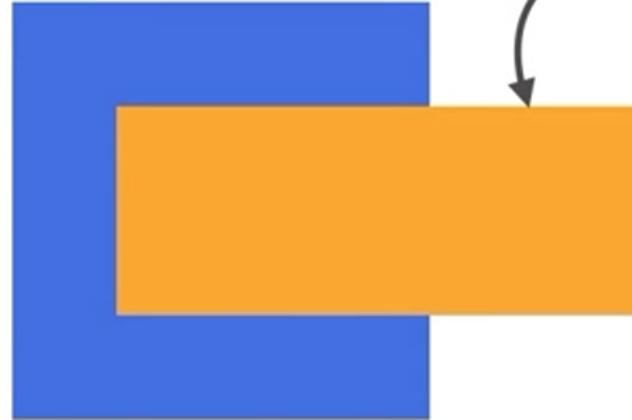
**scroll**

넘친 내용을 잘라냄, 스크롤바 생성

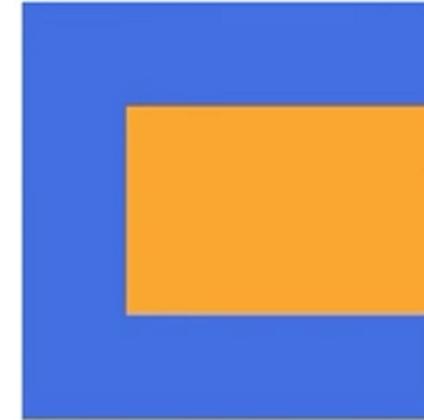
**auto**

넘친 내용이 있는 경우에만 잘라내고 스크롤바 생성

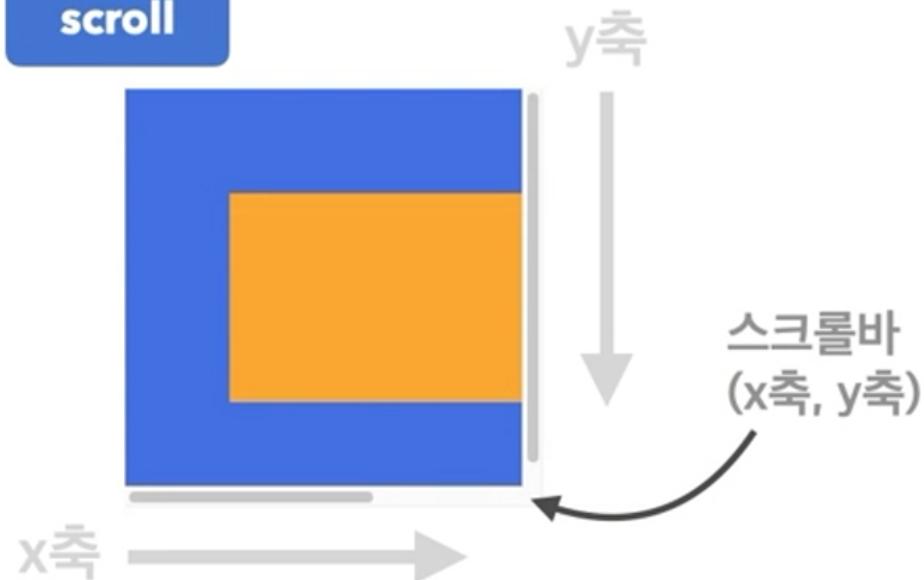
**visible**



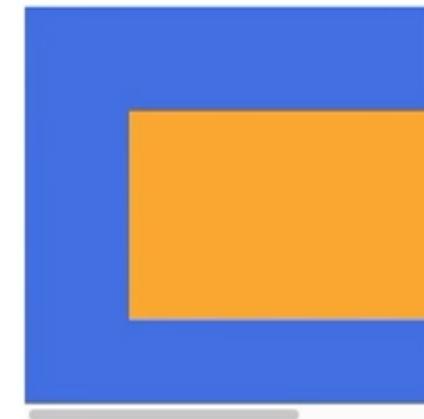
**hidden**



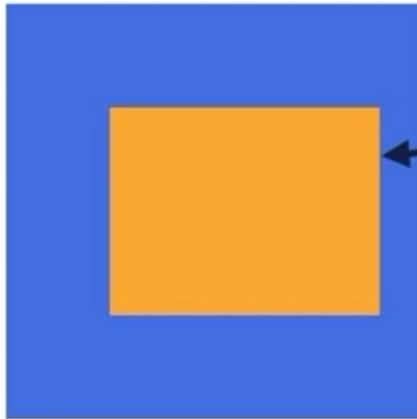
**scroll**



**auto**

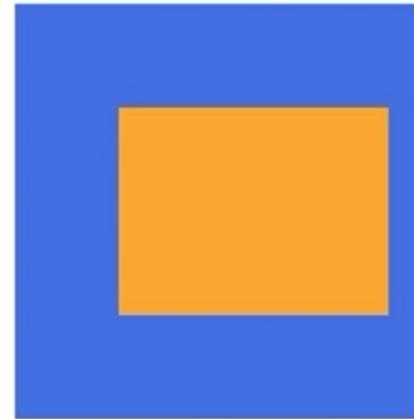


**visible**

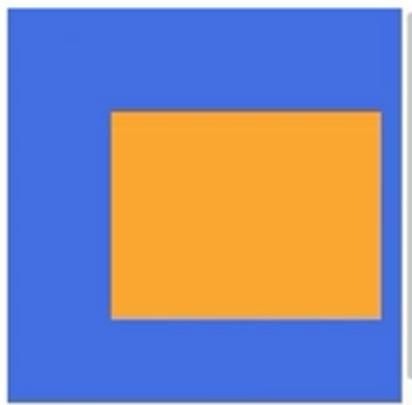


내용 넘치지 않음!

**hidden**

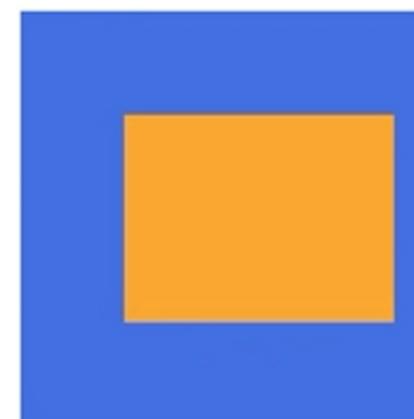


**scroll**



y축  
스크롤바  
(x축, y축)

**auto**



x축

요소의 크기 이상으로 내용이 넘쳤을 때, 보여짐을 제어하는 개별 속성들

**overflow-x**

**overflow-y**