

CSS & BootStrap 기초

강연자 : 교육팀장 박지호

I 1장 :CSS 기초

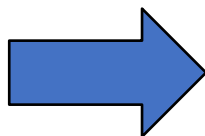
CSS : Cascading Style Sheets

웹에 사용할 스타일들을 명시해둔 명세서.

<p>

이 문단에
폰트는 굴림체
크기는 20px
색은 검정색
을 적용시켜주세요

이 문단에는
폰트는 굴림체
크기는 20px
색은 검정색
을 적용시켜주세요



```
p{  
    font-family : '굴림';  
    font-size : 15px;  
    color : white;  
}
```

✓ 선택자(Selector)란?

```
p {  
  font-family : '굴림';  
  font-size : 15px;  
  color : white;  
}
```

<p> 안녕하세요. </p>

✓ 속성(Property)란?

```
p {  
    font-family : '굴림';  
    font-size : 15px;  
    color : white;  
}
```

✓ 값(Value)란?

```
p {  
    font-family : '굴림';  
    font-size : 15px;  
    color : white;  
}
```

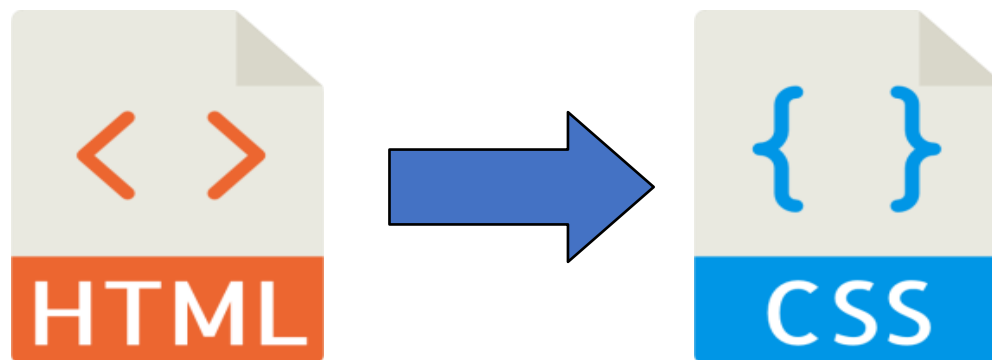
선언 블록
(Declaration block)
선언
(Declaration)

```
p {  
  font-family : '굴림';  
  font-size : 15px;  
  color : white;  
}
```


실습시간

애국가 1절을
폰트는 arial체,
글꼴색상은 하양색,
글꼴크기는 20px,
배경색은 검정색으로 해주세요.

I 2장 : HTML에 CSS를 적용하는 방법



Link style

Embedding style

Inline style

✓ Link style

HTML에 외부에 있는 CSS파일을 불러옴

```
<link rel="stylesheet" href="./main.css">
```

✓ **Embedding style**

HTML의 <head>에 <style>를 이용해 CSS 작성.

✓ **Inline style**

HTML의 요소에 직접 style속성을 이용해 CSS작성.

실습시간.
본인이름을 파랑색
으로 세가지 방법을
써서 표시하시오.

← → ↻ 127.0.0.1:5501/CSS/embedding_style.html ☆ 🌐 ⚙ 119

나는 박지호

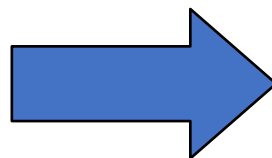
I 3장 :선택자 기초

h1, p, span, div, a, ...

일반적인 태그


```
p {  
    color : red;  
}  
h1{  
    color : red;  
}
```

```
p {  
  color : red;  
}  
h1{  
  color : red;  
}
```



```
p,h1 {  
  color : red;  
}
```

✓ 단순 선택자

타입(Type)

클래스(Class)

아이디(Id)

전체(Universal)

속성(Attribute)

선택자
(Selector)

✓ 타입 선택자(Type Selector)

해당 태그를 가지는 모든 요소에 스타일을 적용.

✓ 아이디 선택자(Id Selector)

해당 ID로 스타일을 적용.

✓ 클래스 선택자(Class Selector)

클래스 이름으로 스타일을 적용

✓ 전체 선택자(Universal Selector)

*을 사용해 모든 요소에 스타일을 적용

✓ 속성 선택자(Attribute Selector)

특정 속성을 소유하는 모든 요소에 스타일을 적용

✓ 복합 선택자

```
<section>
```

```
<article>
```

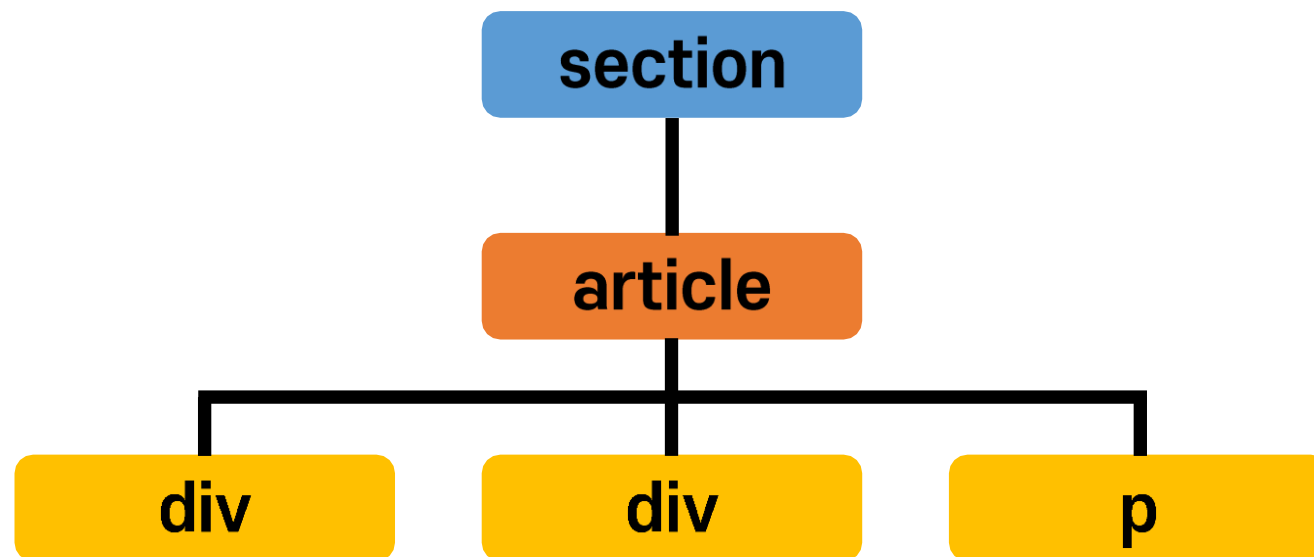
```
<div></div>
```

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

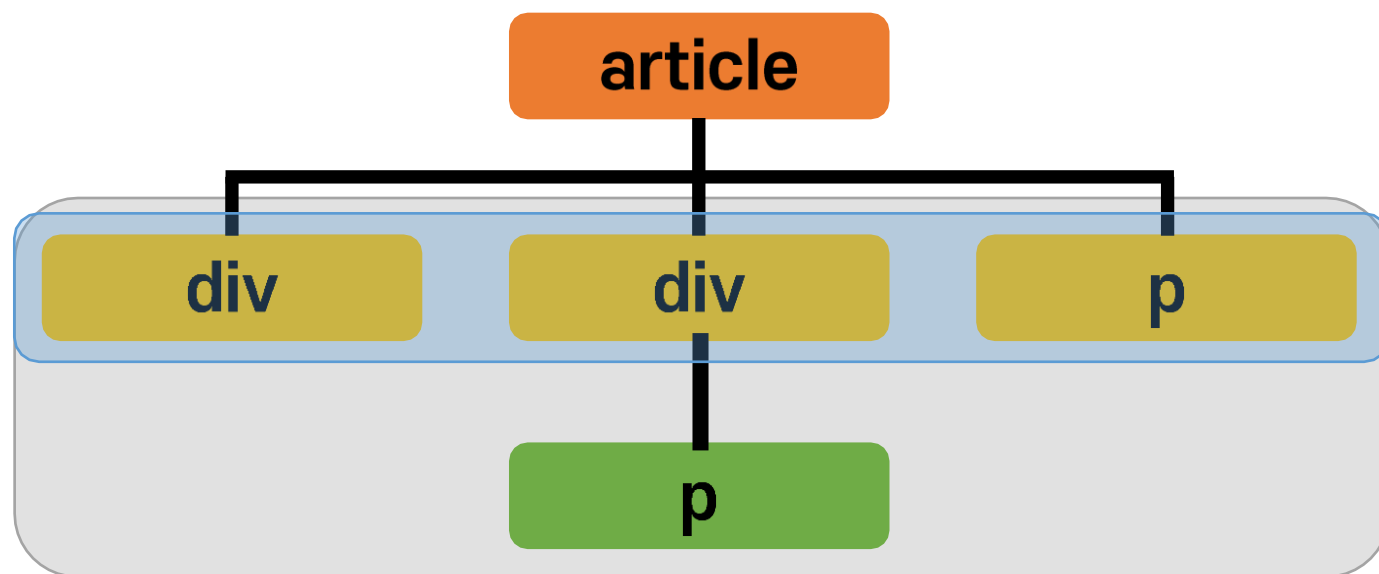
```
<p></p>
```

```
</article>
```

```
</section>
```



✓ 자식 선택자와 후손 선택자

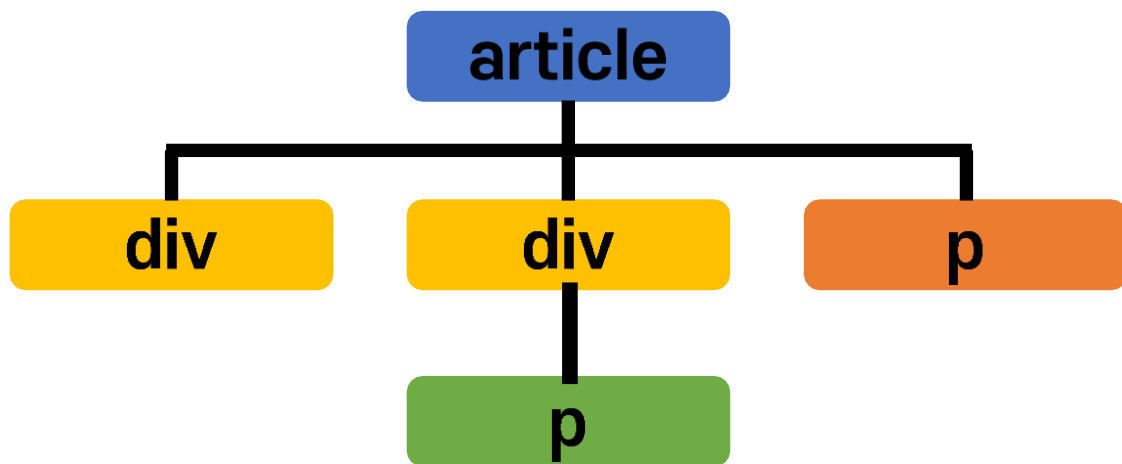


나

자식(Child)

후손(Descendant)

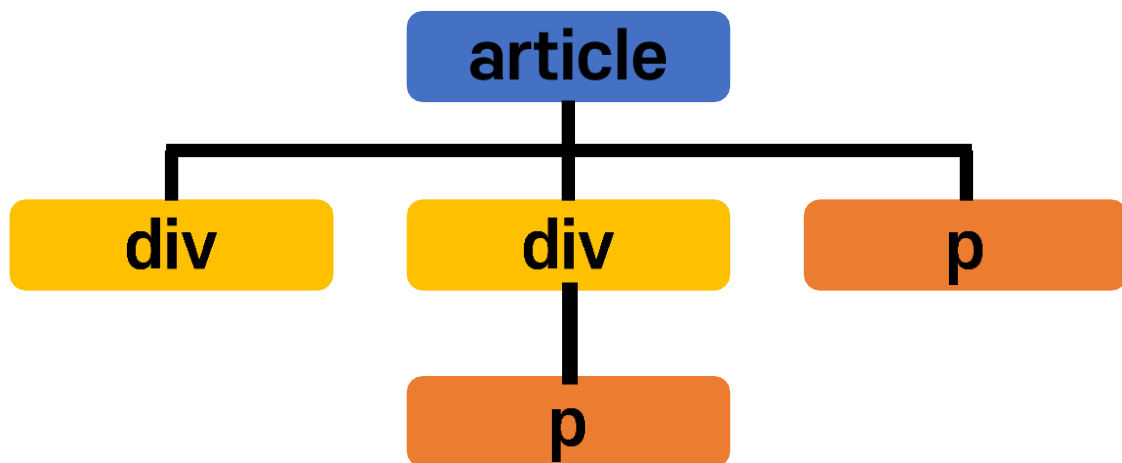
✓ 자식 선택자(Child Selector)



선택자 A > 선택자 B { color : red; }

article > p { color : red; }

✓ 후손 선택자(Descendant Selector)



선택자 A 선택자 B { color : red; }

article p { color : red; }

✓ pseudo 클래스

가상 클래스

```
선택자 : pseudo-class {  
    속성 : 속성 값;  
}
```

color : red; font-size : 20px;

Ⅰ 4장 : 값, 단위, 색

✓ 값(value)란?

```
p {  
    font-family : '굴림';  
    font-size : 15px;  
    color : white;  
}
```

✓ 속성과 값

다양한 속성(Property)

background

font-size

position

border

...

다양한 값(Value)

12px

left

border-box

solid

...

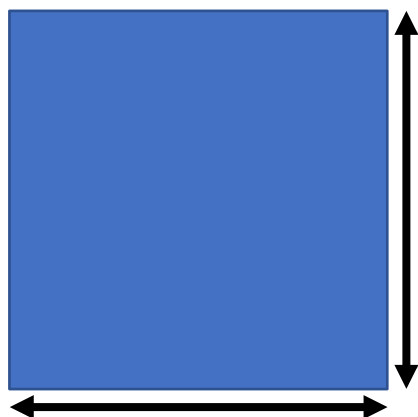
숫자값

키워드

색

CSS 값의 종류

✓ 숫자값과 백분율



너비
(Width)

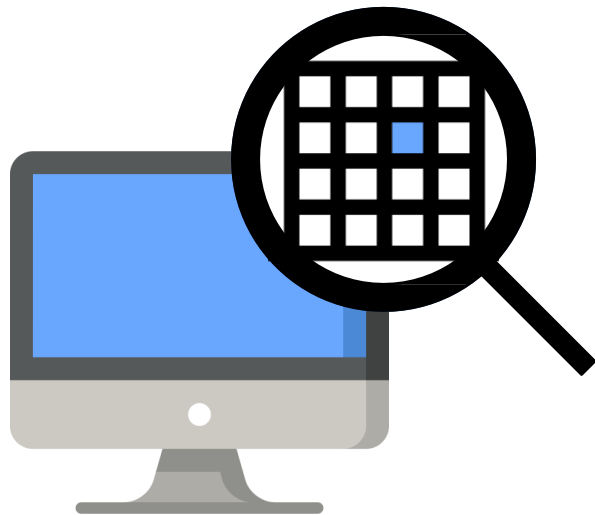
높이
(Height)

글자

글자 크기
(Font-size)

16px 단위
(Units)

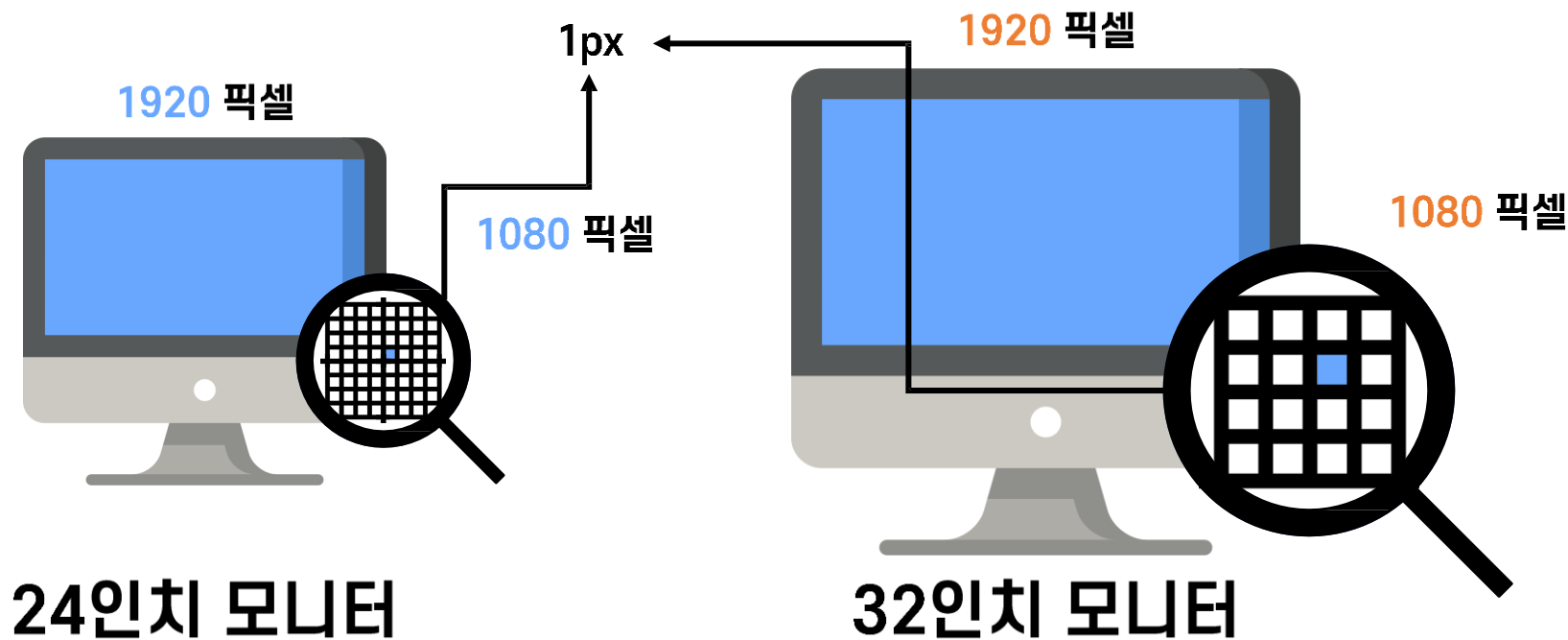
수많은 단위가 존재
But, 웹에서 주로 쓰는 px, em, rem 위주



화소
(1px)

총 픽셀 수 = 가로 픽셀 X 세로 픽셀

✓ 1px = ?

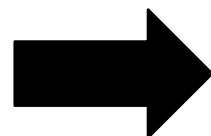


총 2,073,600 픽셀로 동일

1px = 1/96 in
1px -> 절대 길이

✓ em & rem

em
rem



상대적인 길이

✓ **em & rem**

em

: 현재 스타일이 지정된 요소의 font-size기준

rem

: 최상위 요소의 font-size기준

px? em? rem?

✓ %(퍼센트)

상대 길이, 보통 이미지나 레이아웃의
너비나 높이를 지정할 때 씬

✓ 색상

```
background-color: #52A2E4;
```

Hex code

```
color: blue;
```

```
background-color: rgb(189, 221, 247);
```

RGB

5장 : 텍스트와 관련된 프로퍼티

Aa

폰트

✓ 폰트와 관련된 프로퍼티

Font-size

Font-family

Font-style

Font-weight

✓ font-family

```
font-family: Georgia, 'Times New Roman', Times, serif;
```

Google Fonts
(<https://fonts.google.com/>)

✓ font-style

```
font-style: ;
```

- italic
- normal
- oblique
- inherit
- initial
- unset

주로 사용!
순서대로 이탤릭체, 기본, 기울이기

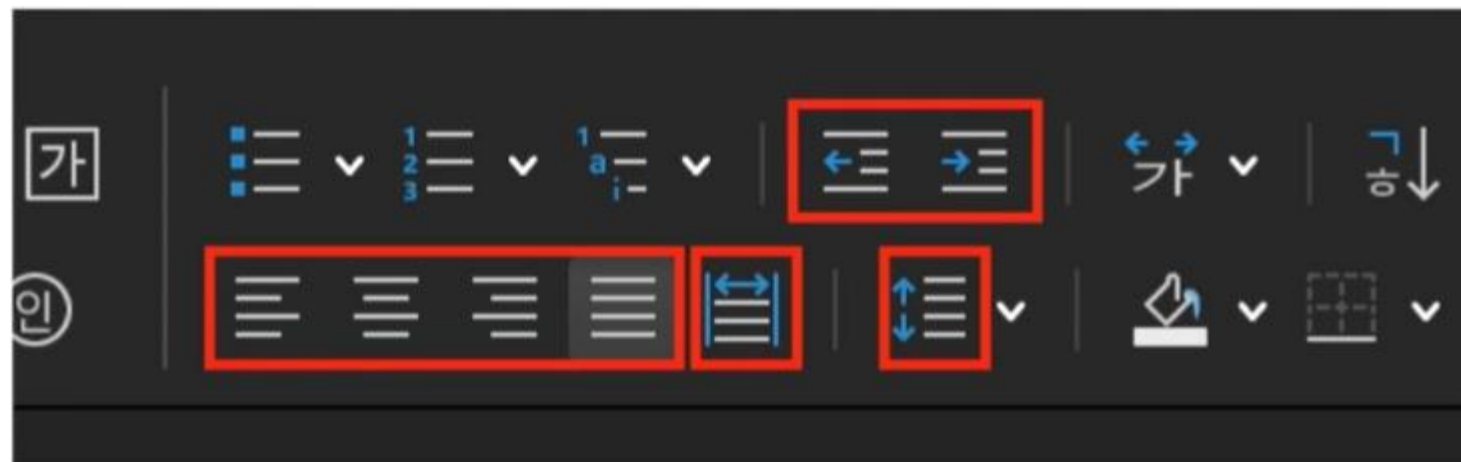
✓ font-weight

```
font-weight: ;
```

- 500
- 600
- 700
- 800
- 900
- bold
- bolder
- lighter
- normal
- inherit
- initial
- unset

100
...
400(normal)
...
700(bold)
...
900

✓ 텍스트 정렬과 관련된 속성

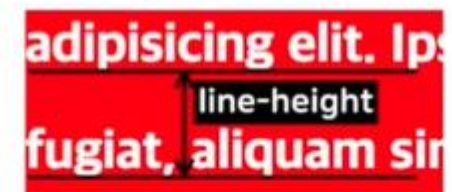


text-align
: 텍스트를 좌, 우, 중앙 정렬함

```
text-align: center;
line-height: 15px;
letter-spacing: 15px;
```

letter-spacing
: 자간

line-height
: 문장사이 간격을 조정



line간의 height X

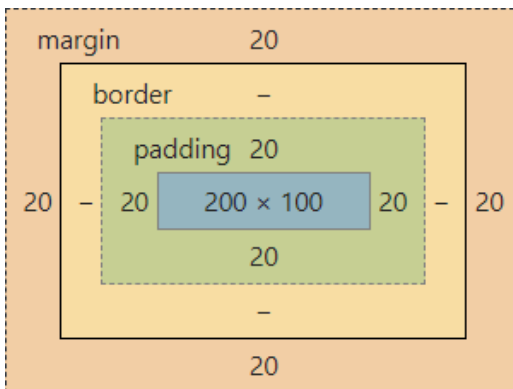


line의 height O

I 6장 : 박스모델 개념

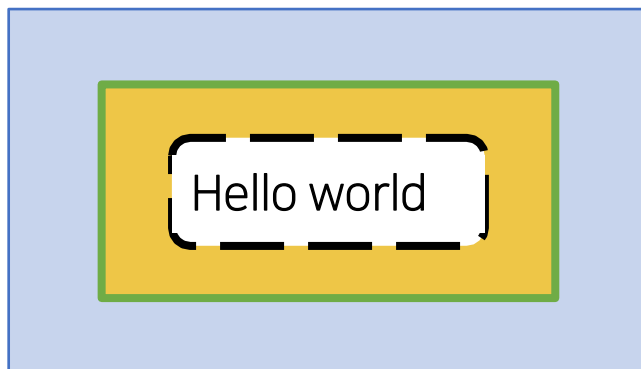


마진(Margin)
: 보더 밖의 여백



패딩(Padding)
: 콘텐츠와 보더 사이의 여백

내용(Content)
: 태그 속 실제 내용



경계선(Border)
: 콘텐츠를 감싸는 테두리

✓ **Border**

border-style

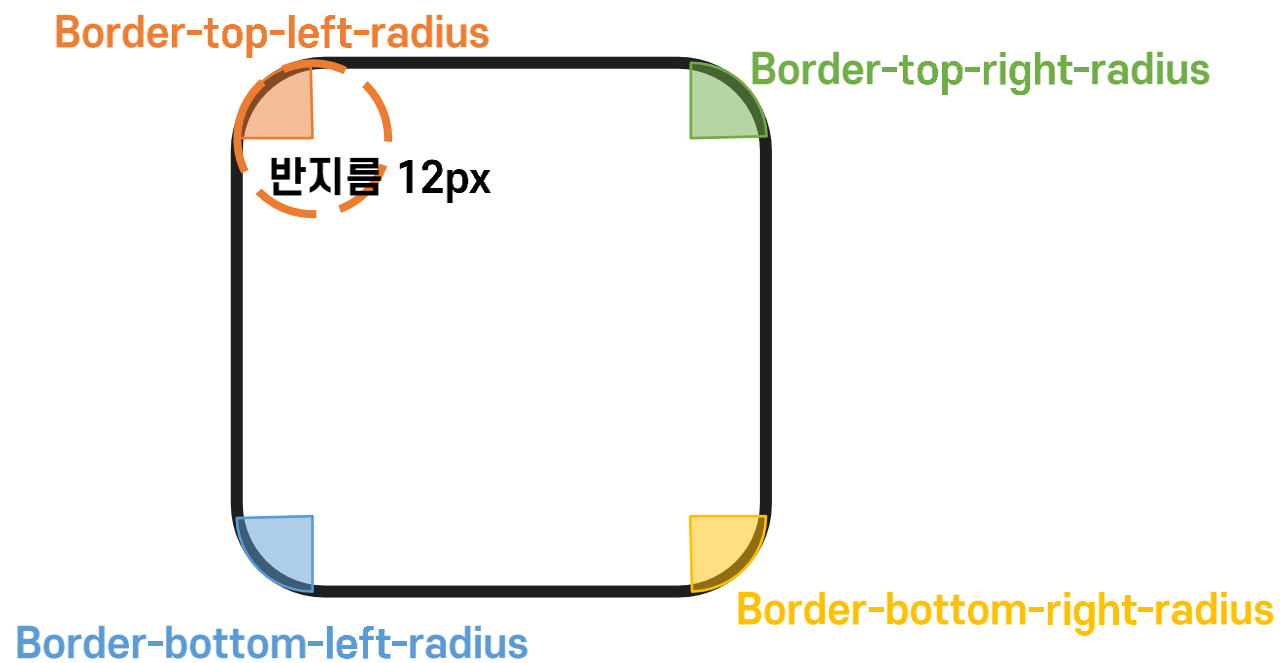
border-width

border-color

border-radius

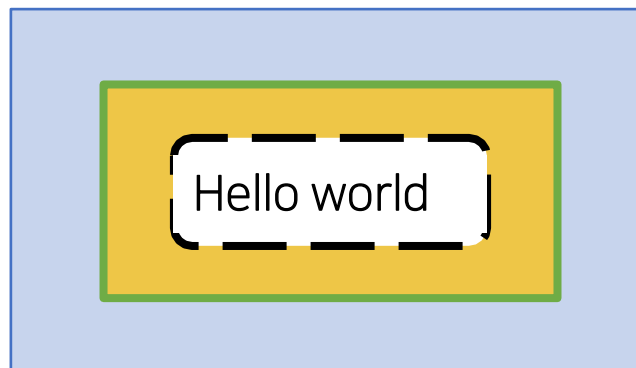


```
#inner{  
    width: 200px;  
    height: 100px;  
    margin: 20px;  
    padding: 20px;  
    border-style: dashed double dotted solid;  
    border-width: 6px;  
    border-color: red blue yellow green;  
}
```



✓ Padding과 Margin

마진(Margin)
: 보더 밖의 여백

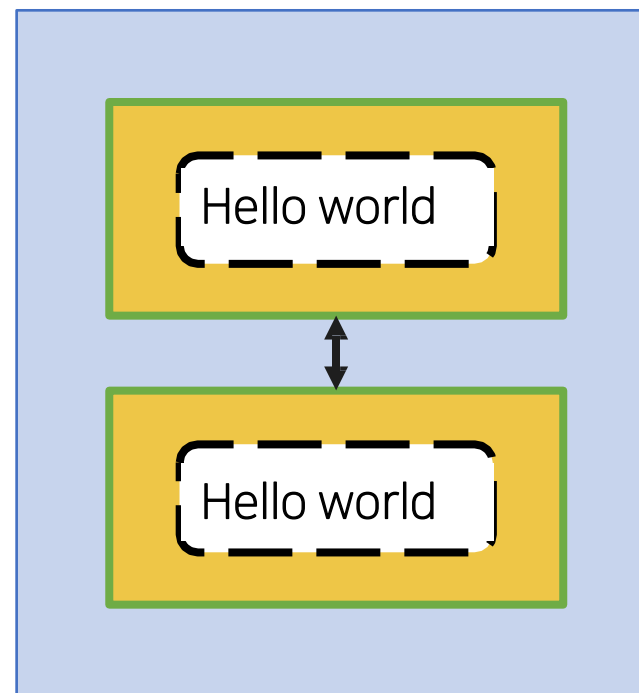
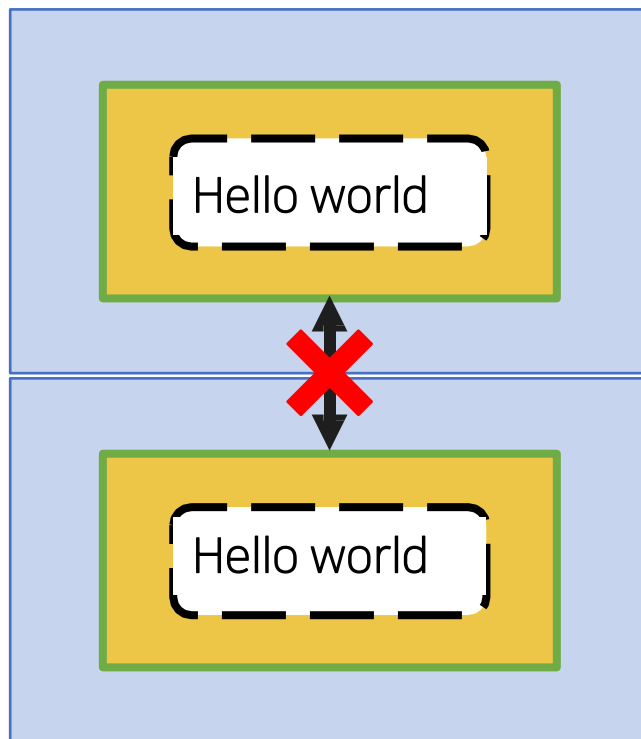


패딩(Padding)
: 콘텐츠와 보더 사이의 여백

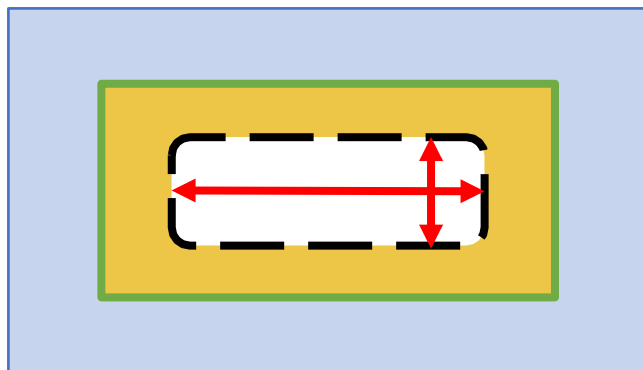
경계선(Border)
: 콘텐츠를 감싸는 테두리

Border를 경계로,
안은 패딩, 밖은 마진

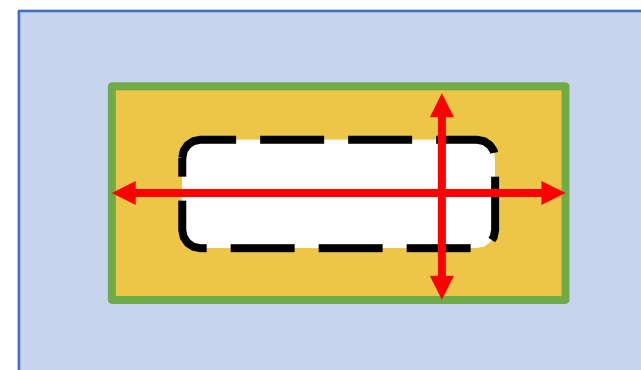
비슷한 두 요소... 차이는?
패딩은 요소 내에 포함!
마진은 요소 내에 미포함!



*** 마진 상쇄(Margin Collapse)**
: 두 마진이 공존하지 않고,
큰 마진을 따라 상쇄되는 것



box-sizing: **content-box**



box-sizing: **border-box**;

box-sizing: content-box;

width(height) = content size

box-sizing: border-box;

width(height) = content size + padding + border

7장 : 위치와 관련된 프로퍼티



✓ display

: 요소가 보여지는 방식을 지정.

```
div{  
  display: block;  
}
```

```
span{  
  display: inline;  
}
```

✓ **display** : block;

<div>, <h1>~<h6>, <header>, <section>

```
<div>
```

```
<h1>수노를 소개합니다.</h1>
```

```
<p>안녕하세요 저는 수노입니다.</p>
```

```
</div>
```

✓ **display** : inline;

<a>, ,

display : block;

: width, height, margin, padding 가능!

display : inline;

: width, height, margin-top, margin-bottom 불가능!

✓ **display** : inline-block;

: width, height, margin-top, margin-bottom 가능

✓ **position**

: 요소의 위치를 정의

1. static
2. relative
3. absolute
4. fixed

Z-index

실습시간

간단한 웹사이트 모음을
앞서 배운내용을 적용해보자.
(디자인 구림주의!)

flexbox

flex container(부모 요소) ← display: flex; 추가



flex container

flex-direction

flex-wrap

justify-content

align-items

align-content

flex item

flex-grow

flex-shrink

flex-basis

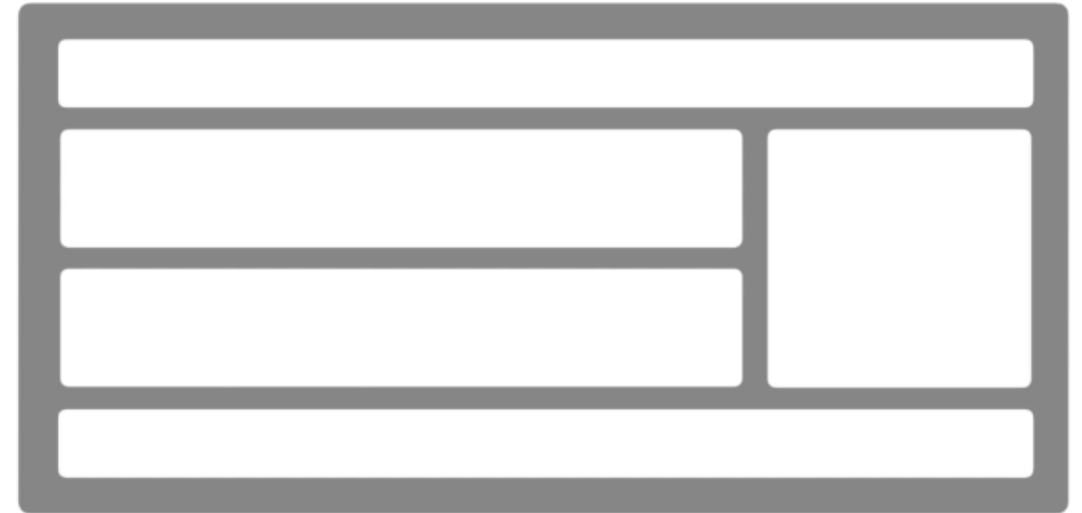
flex



가로방향



세로방향



웹 레이아웃

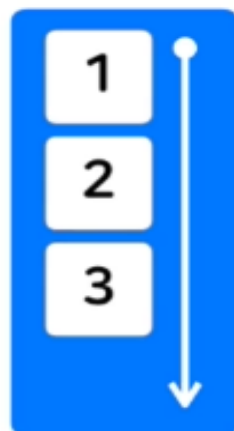
✓ flex-direction



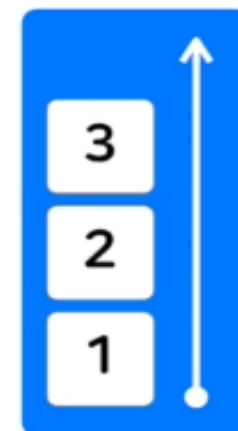
flex-direction: row;



flex-direction: row-reverse;

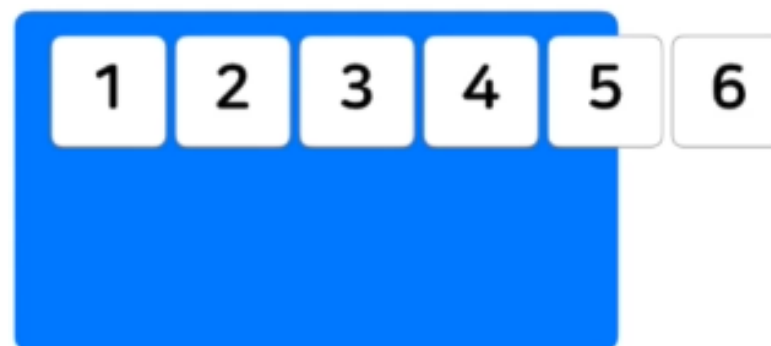


flex-direction
: column;

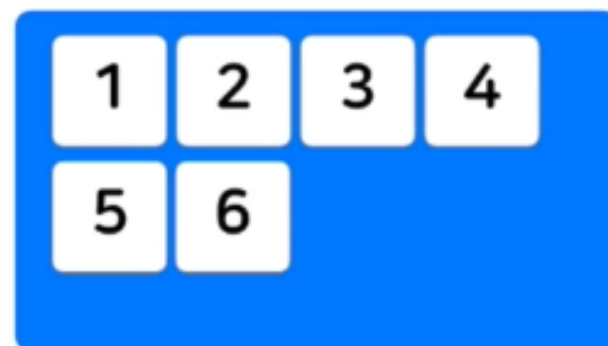


flex-direction
: column-reverse;

✓ flex-wrap



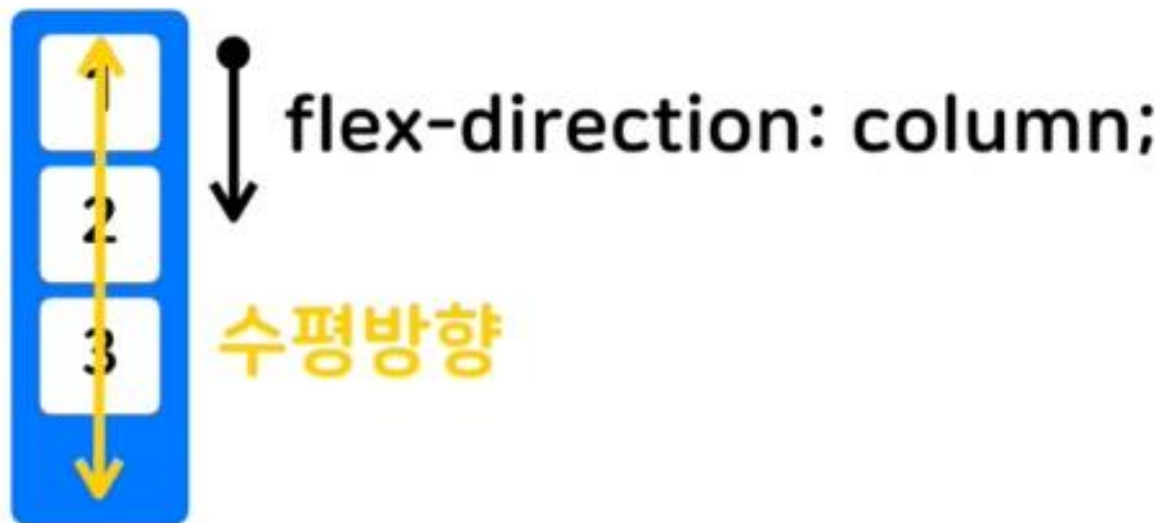
flex-wrap
: nowrap;



flex-wrap
: wrap;

✓ Justify-content

● → flex-direction: row;





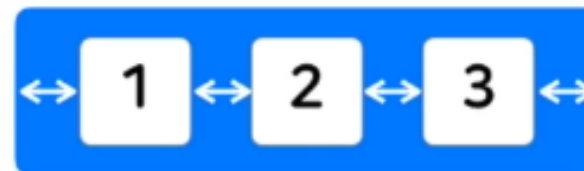
`justify-content: flex-start;`
(기본값)



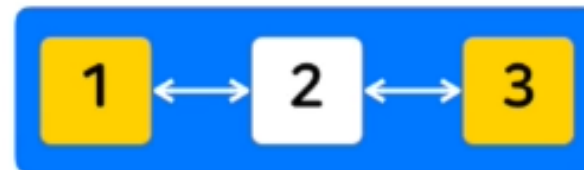
`justify-content: center;`



`justify-content: flex-end;`



`justify-content
: space-around;`



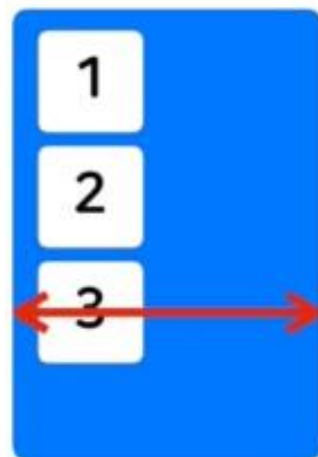
`justify-content
: space-between;`

✓ align-items

● → flex-direction: row;

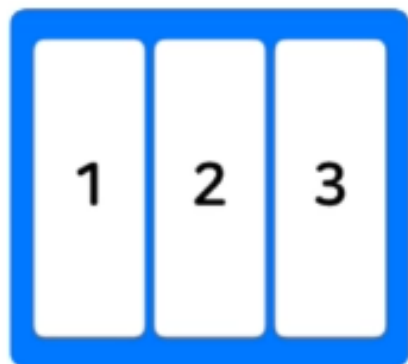


수직방향

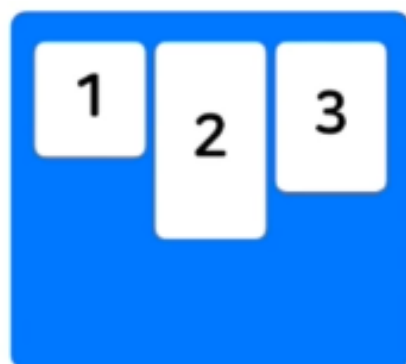


● ↓ flex-direction: column;

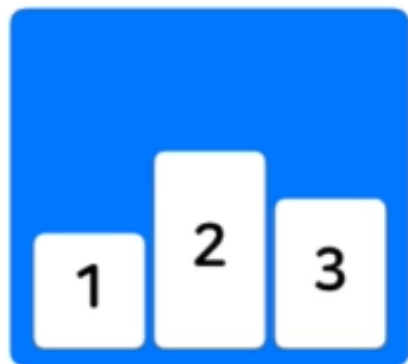
수직방향



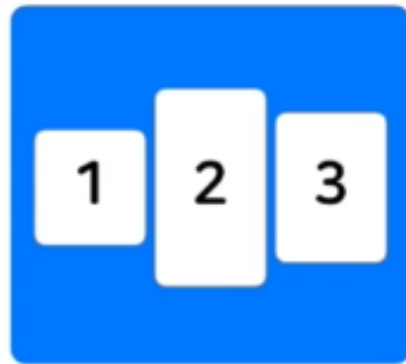
align-items
: stretch;(기본값)



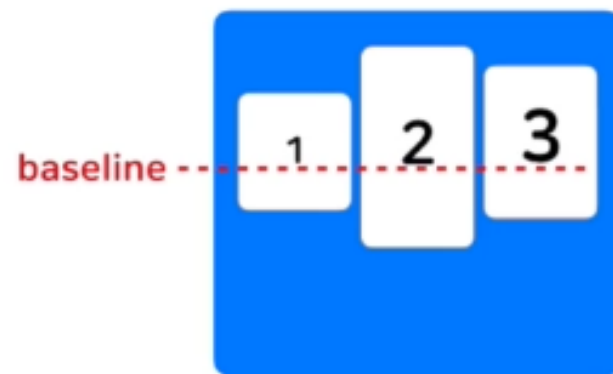
align-items
: flex-start;



align-items
: flex-end;

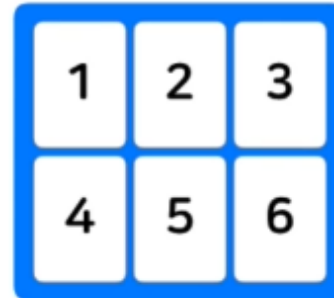


align-items
: center;



align-items
: **baseline**;

✓ align-content



align-content
: stretch;(기본값)



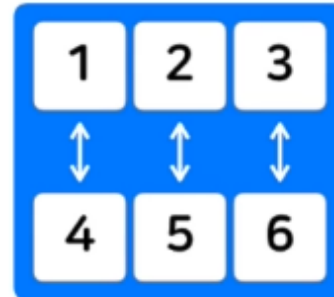
align-content
: flex-start;



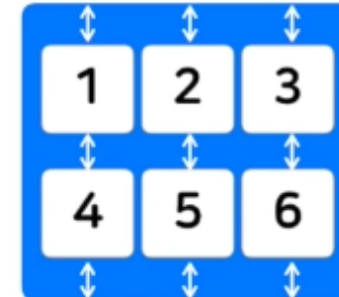
align-content
: flex-end;



align-content
: center;

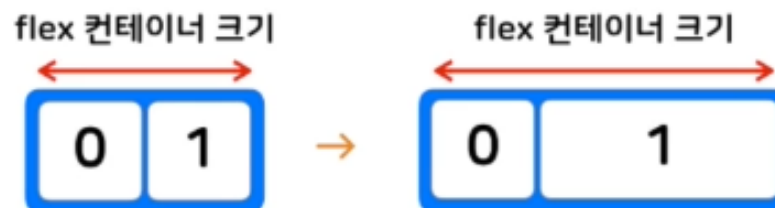
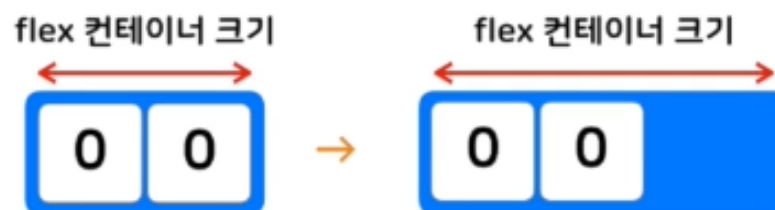
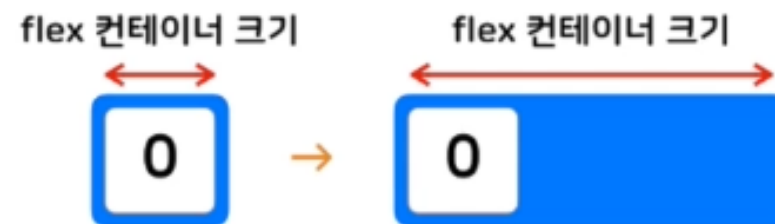


align-content
: space-between;



align-content
: space-around;

✓ Flex-grow

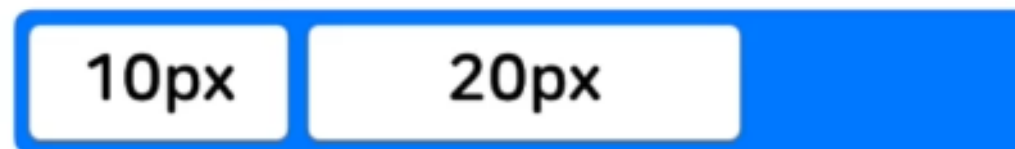


✓ Flex-shrink



✓ Flex-basis

flex-basis: 10px;



flex-basis: 20px;

✓ **Flex**

flex: 1 0 auto; => flex-grow

flex: 0 1 auto; => flex-shrink

flex: 1 1 auto; => 둘 다

flex: 1; => 둘 다

실습시간

간단한 자기소개 블로그를
만들어보자!

I 8장 : 상속

모든 CSS 프로퍼티가 상속되는 것은 아님

상속 여부 아래 링크에서 확인 가능

<https://www.w3.org/TR/CSS21/propidx>

or

margin : inherit;

👉 CSS 적용 우선 순위

1. 중요도
2. 명시도
3. 선언 순서

1. 중요도

- 1) <head> 태그 내의 <style> 태그
- 2) <head> 태그 내의 <style> 태그 내의 @import
문
- 3) <link> 태그로 연결된 CSS
- 4) <link> 태그로 연결된 CSS 내의 @import문
- 5) 브라우저 디폴트 스타일 시트

2. 명시도

- 1) !important
- 2) 인라인 스타일
- 3) 아이디 선택자
- 4) 클래스, 속성, 가상 클래스 선택자
- 5) 태그 선택자
- 6) 전체 선택자
- 7) 상속

선언 순서

나중에 선언된 스타일 우선 적용

| Bootstrap

