

HTML/CSS/Javascript

2025.02.28
김자영 강사

CSS

Cascading Style Sheets

HTML



Markup Language
Content

CSS



Style sheet Language
Presentation

JS



Programming Language
Behavior

CSS

웹 페이지의 디자인과 레이아웃과 같은 스타일 규칙들을 체계적으로 관리하고 적용하는 기술

Cascading

“계단식의”

여러 스타일 규칙이 한 요소에 겹쳐서 적용될 때, 위에서 아래로(또는 계층적으로) 내려오면서 최종 스타일이 결정된다.

Style

“스타일”

웹 페이지의 시각적 표현을 정의한다.
글꼴, 색상, 간격, 레이아웃 등 다양한 시각적 요소를 제어할 수 있다.

Sheets

“문서들”

Sheets는 스타일 규칙이 작성된 문서를 의미하며, 스타일 시트는 HTML 문서와 별도로 관리된다.
HTML 문서에 한 개 이상의 스타일 시트가 적용될 수 있고, 한 개의 시트는 여러 페이지에서 재사용 될 수 있다.

CSS의 주요 기능

텍스트 스타일링	<ul style="list-style-type: none">• 텍스트 속성(크기, 색상, 줄 간격, 폰트 등)을 조정하여 내용의 가독성을 높일 수 있다.
시각적 디자인	<ul style="list-style-type: none">• 배경색, 테두리, 그림자 등의 시각적 효과를 추가하여 웹 페이지의 외관을 개선할 수 있다.
레이아웃 제어	<ul style="list-style-type: none">• 웹 페이지의 전체 레이아웃을 조정하며, 요소들의 크기, 위치, 간격 등을 정밀하게 조정할 수 있다.
반응형 디자인	<ul style="list-style-type: none">• 다양한 디바이스(데스크톱, 태블릿, 모바일)에 따라 다르게 보이도록 웹 페이지를 디자인할 수 있다.

CSS 스타일시트 적용 방법

인라인 스타일 Inline Styles	<ul style="list-style-type: none">• HTML 요소의 style 속성을 사용하여 직접 스타일을 적용하는 방법• 특정 요소에만 스타일을 적용할 때 사용한다.
내부 스타일시트 Internal Stylesheet	<ul style="list-style-type: none">• HTML 문서의 <head> 섹션 내 <style> 태그 안에 CSS를 직접 작성하는 방법• 해당 HTML 문서에만 적용된다.
외부 스타일시트 External Stylesheet	<ul style="list-style-type: none">• 별도의 .css 파일에 스타일을 정의하는 방법• HTML 문서의 <head> 섹션에서 <link> 태그로 연결• 여러 HTML 페이지에서 재사용할 수 있다
@import	<ul style="list-style-type: none">• CSS에서 다른 CSS 파일을 가져오는 방법

각 방식은 상황에 따라 적절히 선택하여 사용한다.
일반적으로 외부 스타일시트를 주로 사용하고, 필요에 따라 다른 방식을 보조적으로 활용한다.

CSS 스타일시트 적용 예시

```
<!DOCTYPE html>
<html lang="ko">
  <head>
    <meta charset="UTF-8" />
    <title>CSS 스타일시트 종류 예시</title>

    <!-- 1. 외부 스타일시트 -->
    <link rel="stylesheet" href="styles.css" />

    <!-- 2. 내부 스타일시트 -->
    <style>
      body {
        font-family: Arial, sans-serif;
      }
      h1 {
        color: navy;
      }
      /* 4. 임포트된 스타일시트 */
      @import url("https://fonts.googleapis.com/css2?family=Roboto&display=swap");
      .imported {
        font-family: "Roboto", sans-serif;
      }
    </style>
```

```
</head>
<body>
  <h1>CSS 스타일시트 예시</h1>

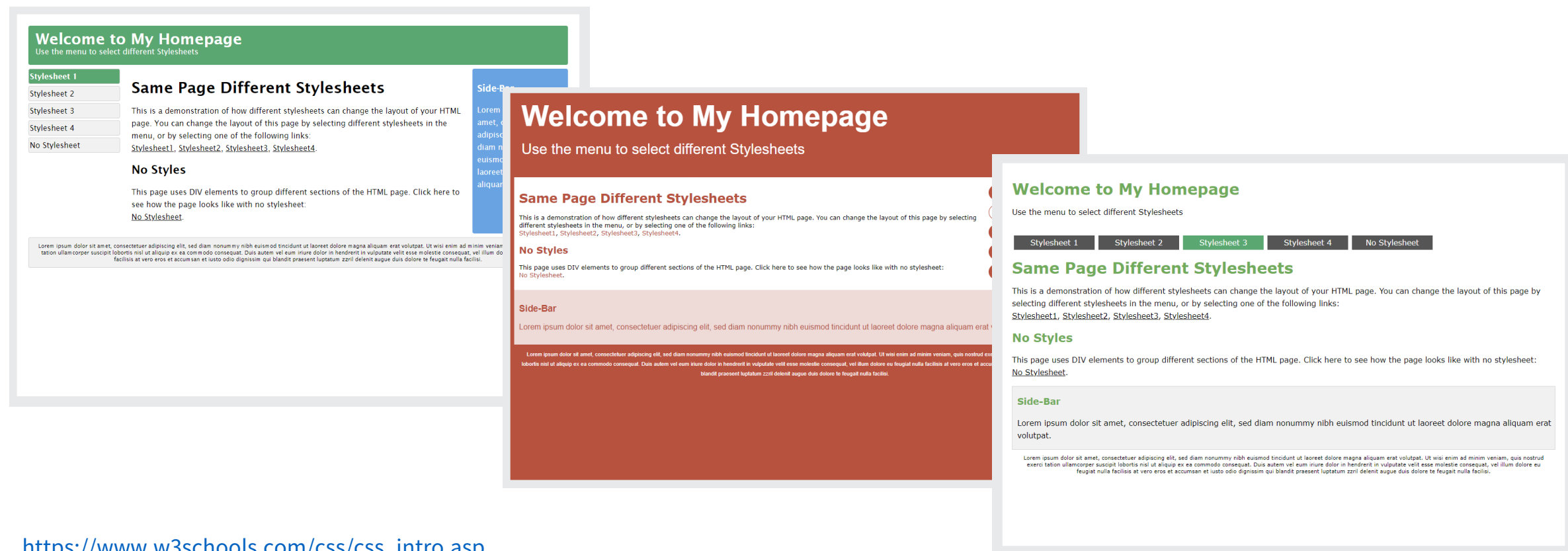
  <!-- 3. 인라인 스타일 -->
  <p style="color: green; font-weight: bold">
    이것은 인라인 스타일을 사용한 문단입니다.
  </p>

  <p class="imported">이 텍스트는 임포트된 Roboto 글꼴을 사용합니다.</p>
</body>
</html>
```

외부 스타일 시트

▪ One HTML Page Multiple Styles

CSS 파일 변경만으로 웹페이지의 디자인/레이아웃을 변경하여, 웹사이트를 다른 분위기로 리뉴얼 할 수 있다.



https://www.w3schools.com/css/css_intro.asp

CSS 문법

html 요소를 선택

선택자{

중괄호{ }

속성1:속성값1;

속성2:속성값2;

속성3:속성값3;

}

콜론(:)

세미콜론(;)

(예시)

```
p {  
  color: red;  
  text-align: center;  
}
```


CSS 선택자의 종류

■ 기본 선택자

선택자 종류	스타일 적용 범위	사용법
전체 선택자 Universal Selector	문서의 모든 요소	*
tag 선택자 Type Selector	특정 태그를 사용한 모든 요소	태그명
class 선택자 Class Selector	동일한 class명을 사용하는 모든 요소	.class명
id 선택자 ID Selector	해당 id를 가진 단일 요소 ID는 페이지에서 유일해야 함	#id명
그룹 선택자 Group Selector	쉼표로 구분된 모든 선택자에 해당하는 요소	선택자1, ..., 선택자n
속성 선택자 Attribute Selector	지정된 속성을 가진 모든 요소	[attribute]
	지정된 속성 값을 가진 모든 요소	[attribute="value"]

CSS 선택자의 종류

■ 속성 선택자

속성선택자	설명
태그[속성]	특정 속성이 있는 요소를 선택
태그[속성=속성값]	특정 속성값이 있는 요소를 선택
태그[속성~=값]	여러 값 중 특정 속성값이 포함된 속성 요소를 선택
태그[속성 =값]	특정 속성값이 포함된 속성 요소를 선택 하이픈(-) 또는 공백으로 구분된 값의 첫 부분을 대상으로 한다.
태그[속성^=값]	특정 속성값으로 시작하는 속성 요소를 선택
태그[속성\$=값]	특정 속성값으로 끝나는 속성 요소를 선택
태그[속성*=값]	일부 속성값이 일치하는 요소를 선택

CSS 선택자의 종류

▪ 고급 선택자

선택자 종류	스타일 적용 범위	사용법
자식 선택자 Child Selector	특정 부모 요소의 직계 자식 요소를 선택	parent > child
후손 선택자 Descendant Selector	특정 조상 요소의 모든 후손 요소를 선택	ancestor descendant
인접 형제 선택자 Adjacent Sibling Selector	특정 요소 바로 다음에 오는 형제 요소를 선택	A + B
일반 형제 선택자 General Sibling Selector	특정 요소 뒤에 위치한 모든 형제 요소를 선택	A ~ B
가상 클래스 선택자 Pseudo-classes	특정 상태에 있는 요소를 선택	:pseudo-class (예) a:hover{} //마우스포인터로 요소를 가리키고 있을 때만 요소를 선택
가상 요소 선택자 Pseudo-elements	요소의 특정 부분에 스타일을 적용	::pseudo-element (예) p::first-line // 요소 내부의 첫번째 줄 선택
복합 선택자 Combinator Selectors	해당 요소 및 클래스를 모두 가진 요소 선택	element.class .class.class .class.class.class 등
부정 선택자 Negation Pseudo-class	지정된 선택자와 일치하지 않는 요소를 선택	:not(selector)

CSS 선택자 연습하기

```
<body>
<div>
  <div>
    <h2>웹프로그래밍 기초</h2>
    <ul>
      <li>HTML</li>
      <li>CSS</li>
      <li>Javascript</li>
    </ul>
  </div>
  <div>
    <h2>참고 사이트</h2>
    <ul>
      <li>
        <a href="https://www.w3.org/" target="_blank">W3C</a>
      </li>
      <li>
        <a href="https://developer.mozilla.org/ko/" target="_blank">MEN web docs</a>
      </li>
      <li>
        <a href="https://www.w3schools.com/" target="_blank">W3C School</a>
      </li>
    </ul>
  </div>
</div>
</body>
```

웹프로그래밍 기초

- HTML
- CSS
- Javascript

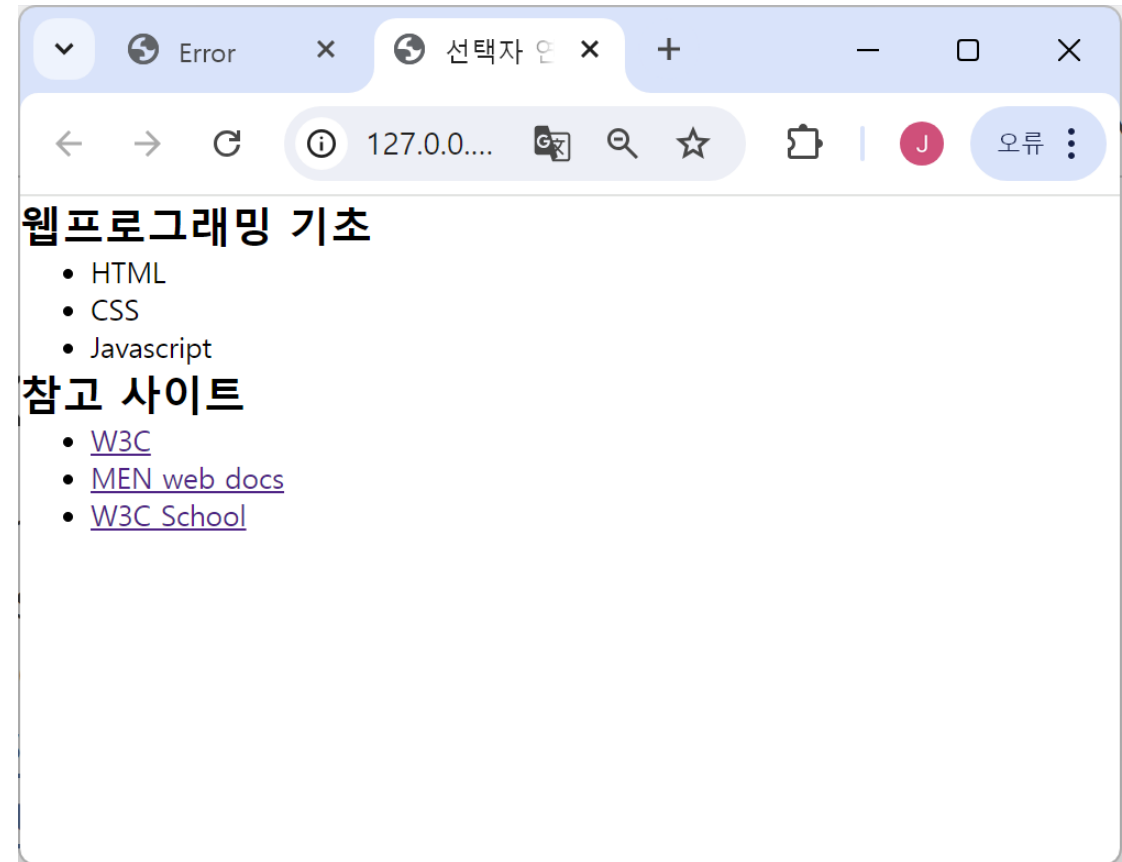
참고 사이트

- [W3C](https://www.w3.org/)
- [MEN web docs](https://developer.mozilla.org/ko/)
- [W3C School](https://www.w3schools.com/)

CSS 선택자 연습하기

■ 전체 선택자

```
* {  
  margin: 0px;  
}
```

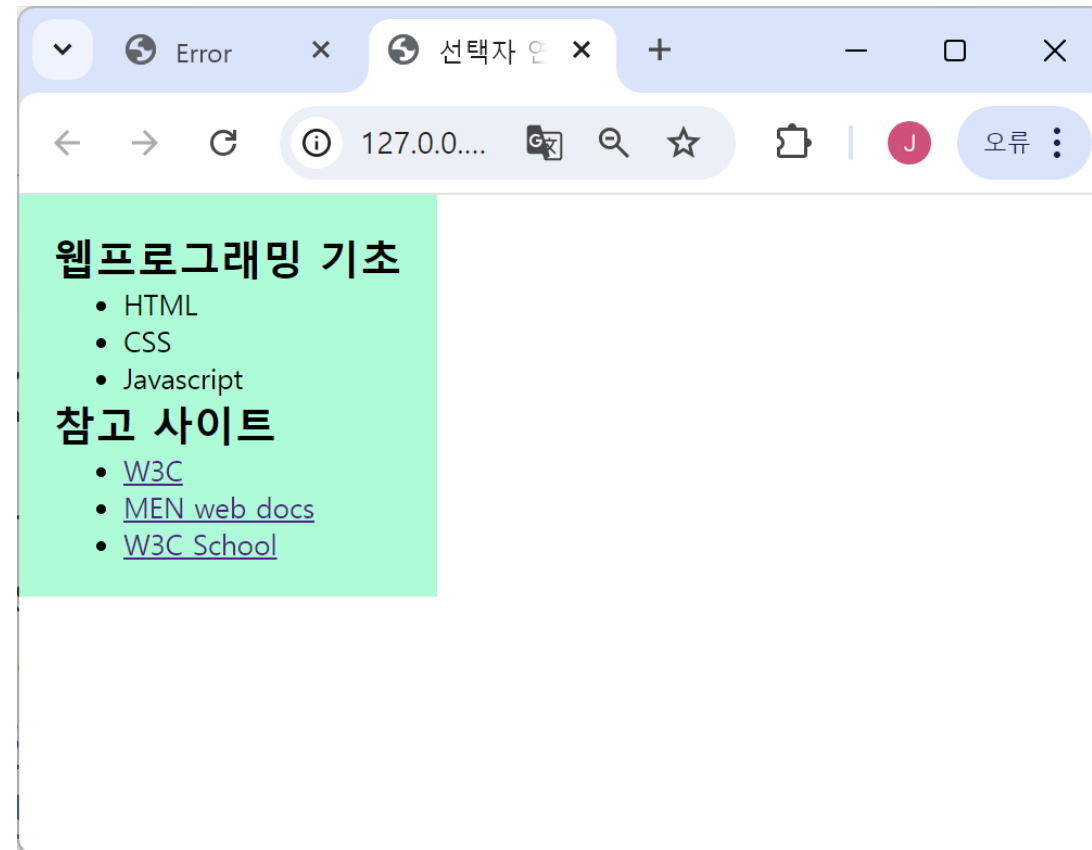


CSS 선택자 연습하기

▪ id 선택자

```
#main-container {  
  background-color: aquamarine;  
  display: inline-block;  
  padding: 20px;  
}
```

```
<div id="main-container">  
  <div>  
    <h2>웹프로그래밍 기초</h2>  
    <ul>  
      <li>HTML</li>  
      <li>CSS</li>  
      <li>Javascript</li>  
    </ul>  
  </div>  
</div>
```

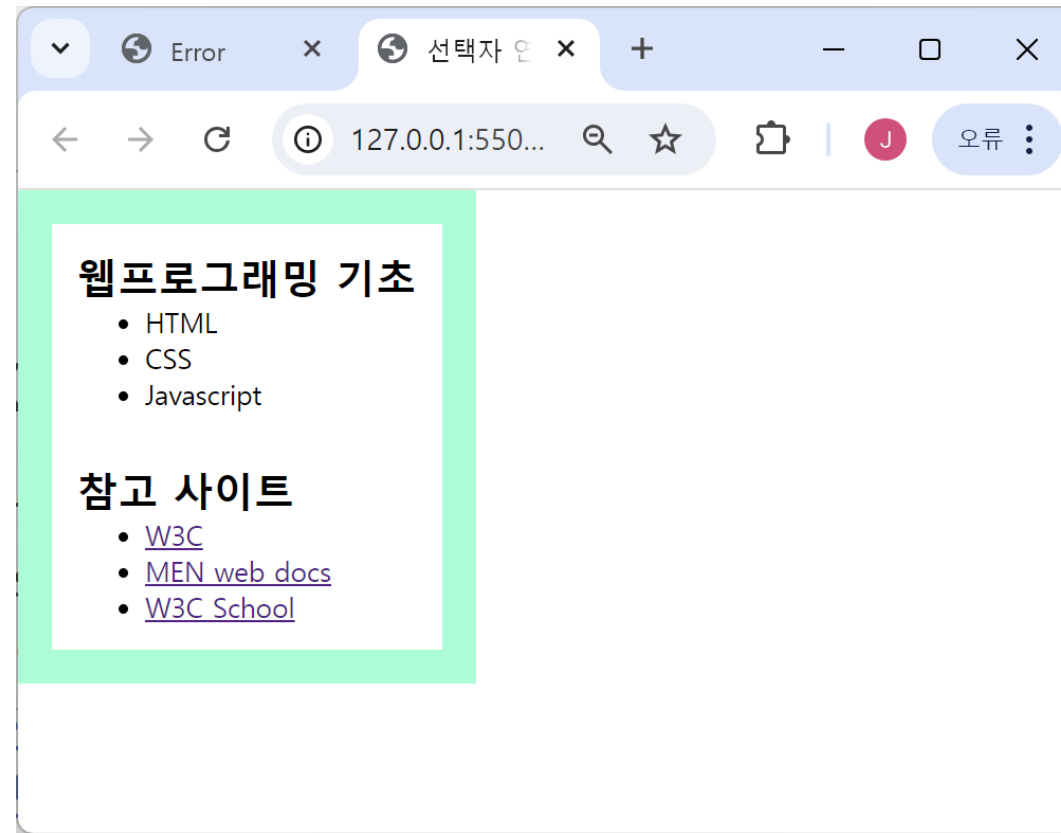


CSS 선택자 연습하기

▪ class 선택자

```
.container {  
  background-color: white;  
  padding: 15px;  
}
```

```
<div id="main-container">  
  <div class="container">  
    <h2>웹프로그래밍 기초</h2>  
    <ul>  
      <li>HTML</li>  
      <li>CSS</li>  
      <li>Javascript</li>  
    </ul>  
  </div>  
  <div class="container">  
    <h2>참고 사이트</h2>  
    <ul>
```



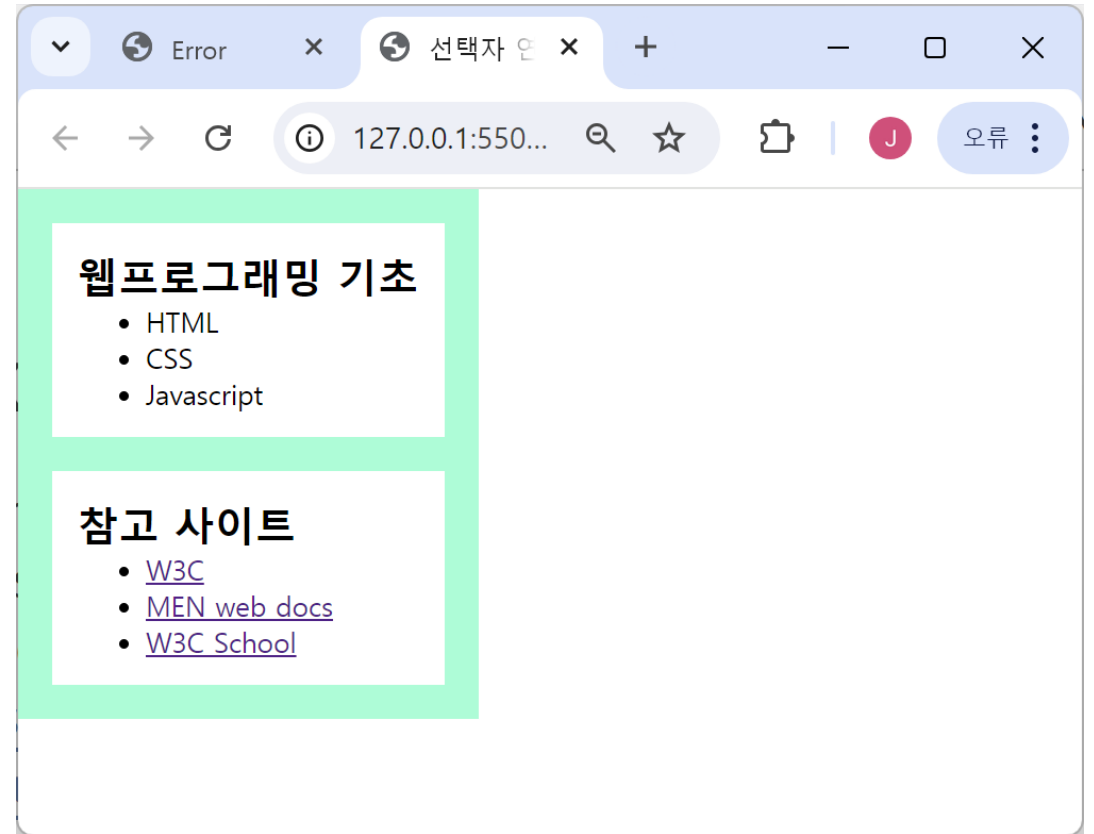
CSS 선택자 연습하기

▪ class 선택자

```
.margin-bottom20 {  
  margin-bottom: 20px;  
}
```

```
<div id="main-container">  
  <div class="container margin-bottom20">  
    <h2>웹프로그래밍 기초</h2>  
    <ul>  
      <li>HTML</li>  
      <li>CSS</li>  
      <li>Javascript</li>  
    </ul>  
  </div>  
  <div class="container">  
    <h2>참고 사이트</h2>  
    <ul>
```

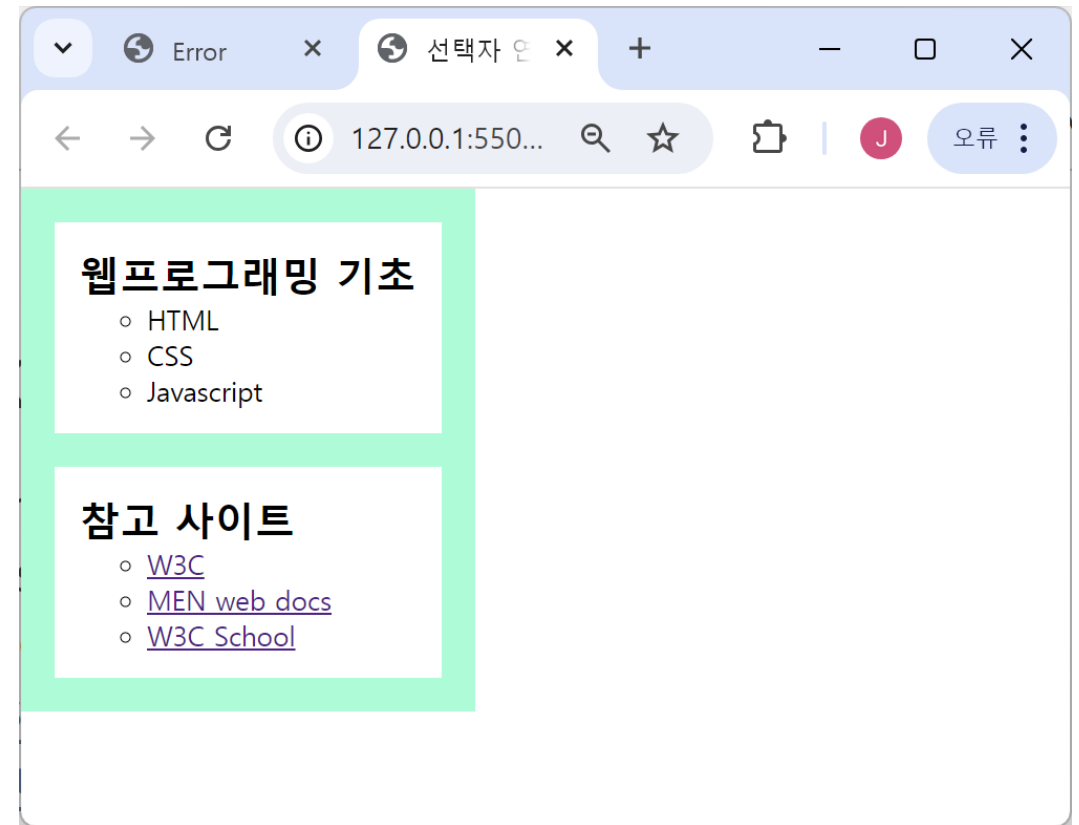
하나의 요소에
여러 개의 클래스를
공백으로 구분하여 적용.



CSS 선택자 연습하기

▪ type 선택자

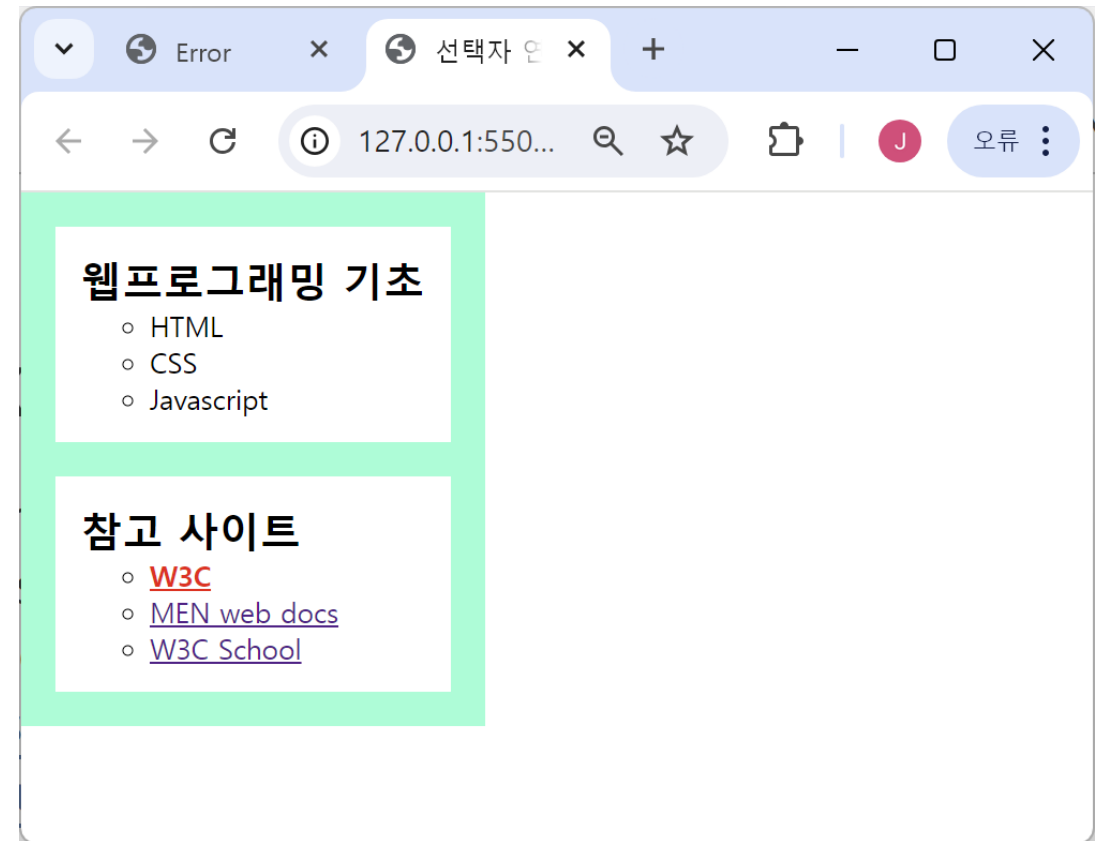
```
ul {  
  list-style-type: circle;  
}
```



CSS 선택자 연습하기

■ 속성 선택자

```
[href="https://www.w3.org/"]  
{  
  font-weight: bold;  
  color: red;  
}
```

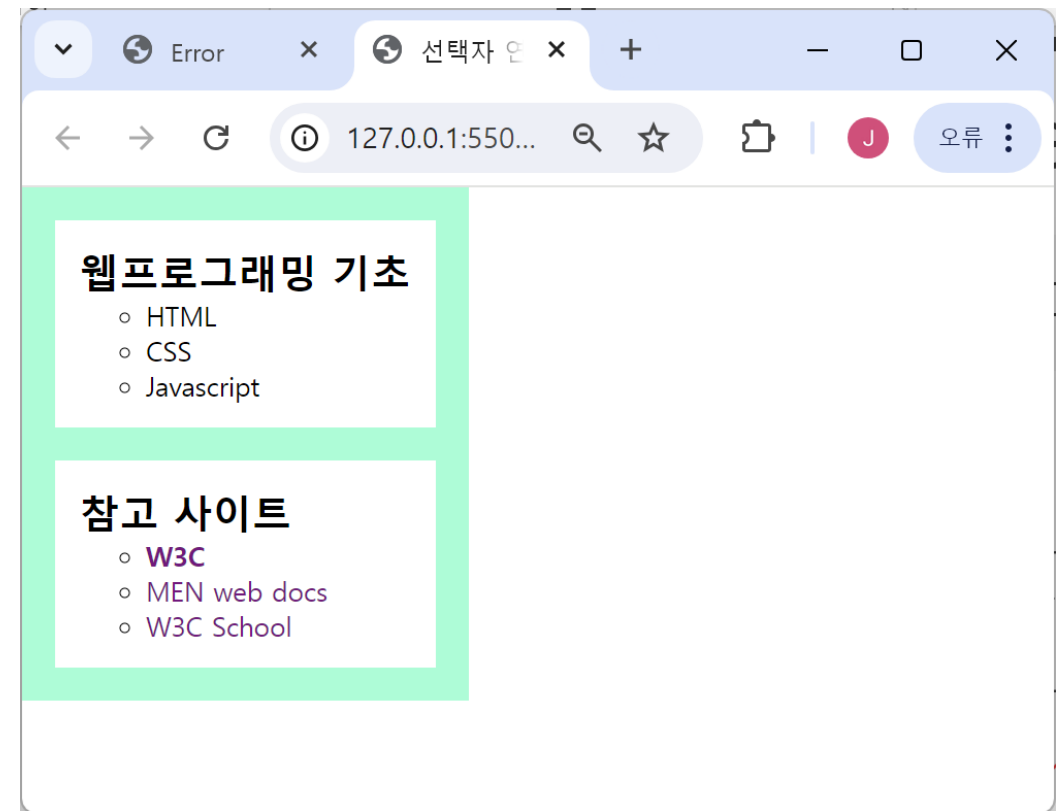


CSS 선택자 연습하기

가상클래스 선택자

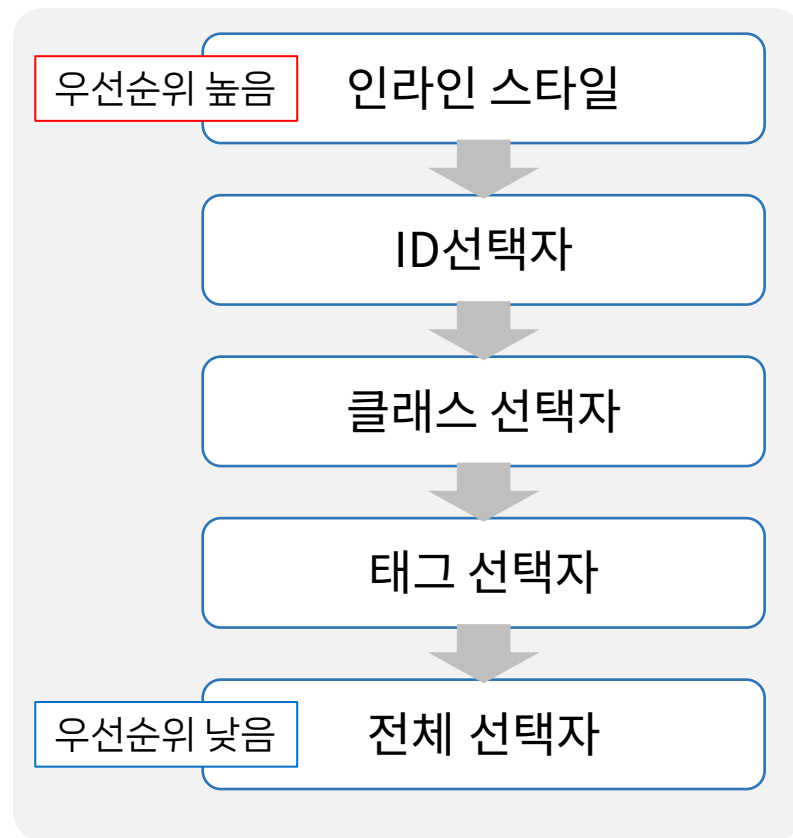
※ HTML 문서의 특정 요소가 특정 상태에 있을 때 스타일을 적용하기 위해 사용되는 선택자

```
a:link { 링크가 걸려있을 때  
  text-decoration: none;  
}  
a:visited { 이미 방문했을 때  
  color: purple;  
}  
a:hover { 마우스 올렸을 때  
  background-color: lightyellow;  
}  
a:active { 클릭하는 순간  
  background-color: yellow;  
}  
div:hover {  
  background-color: lightgray;  
}
```



CSS 적용 매커니즘1 - 선택자 우선순위

- 동일한 요소에 여러 스타일이 적용될 때, 다음과 같은 순서에 따라 우선순위가 평가된다.



[참고]
!important 키워드를 이용하여
우선순위를 무시하고 스타일을 강제 적용하도록 할 수 있다.
하지만 사용 권장하지는 않음

CSS 적용 매커니즘1 - 선택자 우선순위

- 동일한 요소에 여러 스타일이 적용될 때, 다음과 같은 순서에 따라 우선순위가 평가된다.

```
<body>
  <h1>CSS선택자 우선순위</h1>
  <p class="myclass" id="myid" style="color: violet">1. inline 선택자</p>
  <p class="myclass" id="myid">2. id선택자</p>
  <p class="myclass">3. class 선택자</p>
  <p class="myclass">4. 태그 선택자</p>
  <p class="myclass">5. 전체 선택자</p>
</body>
```

무슨색?

무슨색?

무슨색?

무슨색?

```
<style>
  * {
    color: blue;
  }
  p {
    color: red;
    font-size: 1.5em;
  }
  .myclass {
    color: gray;
  }
  #myid {
    color: green;
  }
</style>
```

[참고] !important 키워드

우선순위를 무시하고 스타일을 강제 적용하도록 할 수 있지만 사용 권장하지 않음

※ 본 강의 자료는 본인 학습용으로만 사용가능하며 외부 유출을 금합니다.

CSS 적용 매커니즘2 – cascade

- 동일한 우선 순위를 갖는 두 규칙이 적용될 때, CSS 에서 마지막에 나오는 규칙이 적용된다.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>cascade</title>
    <style>
      h1 {
        color: red;
      }
      h1 {
        color: blue;
      }
    </style>
  </head>
  <body>
    <h1>나는 무슨색일까요?</h1>
  </body>
</html>
```

무슨색?

CSS 적용 매커니즘3 - 스타일 속성 상속

- 부모 요소에서 설정된 일부 CSS 속성 값은 자식 요소에 상속된다.

상속되는 속성 : 텍스트 관련 속성

- color: 텍스트 색상
- font-family: 글꼴
- font-size: 글꼴 크기
- font-style: 글꼴 스타일 (이탤릭 등)
- font-weight: 글꼴 두께
- line-height: 줄 간격
- text-align: 텍스트 정렬
- text-indent: 텍스트 들여쓰기
- text-transform: 텍스트 대소문자 변환
- visibility: 요소 가시성
- 등

• Initial

- 상속되는 속성을 명시적으로 상속받지 않도록 설정

상속되지 않는 속성 : 레이아웃 관련 속성

- margin: 외부 여백
- padding: 내부 여백
- border: 테두리
- width: 너비
- height: 높이
- display: 요소의 표시 방식
- position: 위치 지정 방식
- background: 배경 관련 속성들
- 등

• Inherit

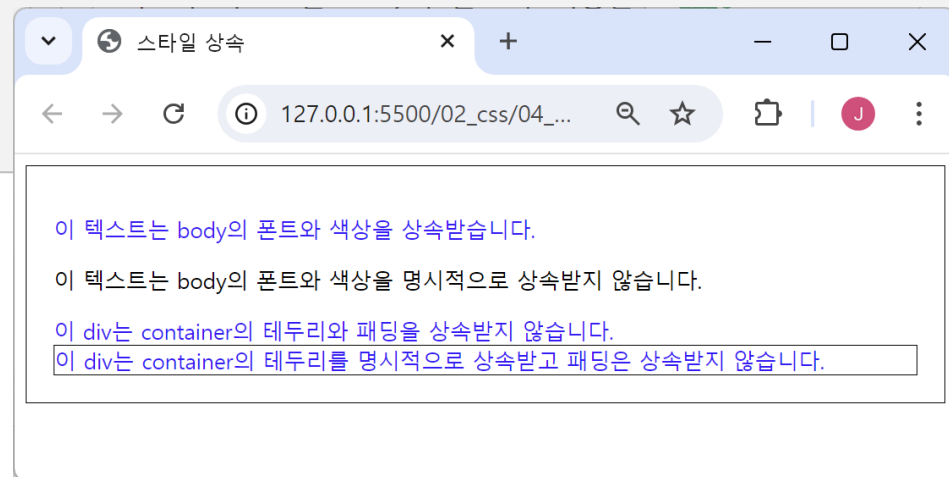
- 상속되지 않는 속성을 명시적으로 상속받도록 설정

CSS 적용 매커니즘3 - 스타일 속성 상속

- 부모 요소에서 설정된 일부 CSS 속성 값은 자식 요소에 상속된다.

```
<style>
  body {
    color: blue;
  }
  .container {
    border: 1px solid black;
    padding: 20px;
  }
  .inherit {
    border: inherit; /* 명시적으로 상속 */
  }
  .initial {
    color: initial; /* 명시적으로 상속받지 않음 */
  }
</style>
```

```
<body>
  <div class="container">
    <p>이 텍스트는 body의 폰트와 색상을 상속받습니다.</p>
    <p class="initial">
      이 텍스트는 body의 폰트와 색상을 명시적으로 상속받지 않습니다.
    </p>
    <div>이 div는 container의 테두리와 패딩을 상속받지 않습니다.</div>
    <div class="inherit">
      이 div는 container의 테두리를 명시적으로 상속받고 패딩은 상속받지
      않습니다.
    </div>
  </div>
</body>
```



color

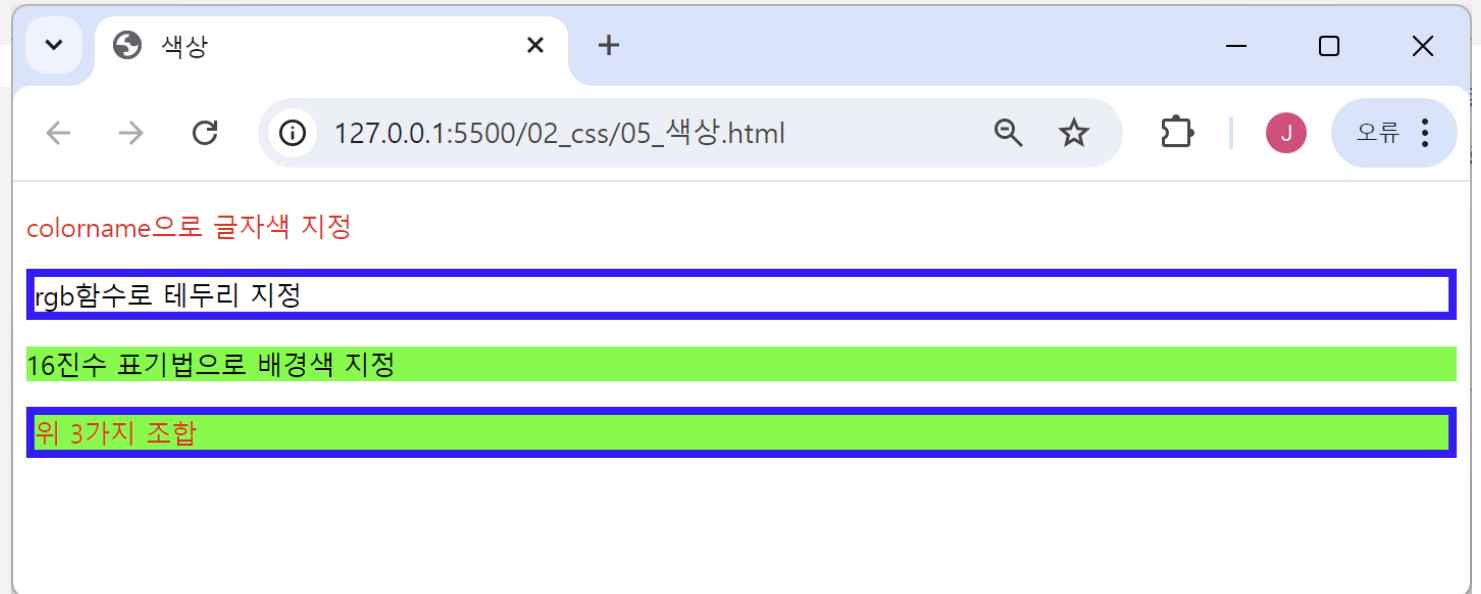
색 지정 방법	사용 예시
색상 이름	<ul style="list-style-type: none">• 색상이름• https://developer.mozilla.org/en-US/docs/Web/CSS/named-color
16진수 표기법	<ul style="list-style-type: none">• #RRGGBB
rgb, rgba	<ul style="list-style-type: none">• rgb(red(0~255) , green(0~255) , blue색상값(0~255))• rgb(red(%) , green(%) , blue색상값(%))• rgba(red(0~255) , green(0~255) , blue(0~255), 투명도(0~1))• rgba(red(%) , green(%) , blue(%), 투명도(%))
hsl, hsla hue saturation lightness	<ul style="list-style-type: none">• hsl(색조(0~360), 채도(%), 명도(%))• hsla(색조(0~360), 채도(%), 명도(%), 투명도(%))

color

- 글자, 테두리, 배경에 color 적용하기

```
<style>
  .font {
    color: red;
  }
  .line {
    border-color: rgb(0, 0, 255);
    border-style: solid;
    border-width: 5px;
  }
  .background {
    background-color: #00ff00;
  }
</style>
```

```
<body>
  <p class="font">colorname으로 글자색 지정</p>
  <p class="line">rgb함수로 테두리 지정</p>
  <p class="background">16진수 표기법으로 배경색 지정</p>
  <p class="font line background">위 3가지 조합</p>
</body>
```



color

■ 색을 선택하기 위한 유용한 사이트

사이트	주요 특징
Adobe Color https://color.adobe.com/	<ul style="list-style-type: none">색상 휠을 사용해 조화를 이루는 색상 조합을 쉽게 찾을 수 있다.다양한 색상 규칙(예: 보색, 삼색 등)을 바탕으로 팔레트를 생성할 수 있다.
Coolors https://coolors.co/	<ul style="list-style-type: none">빠르게 색상 팔레트를 생성하고, 조정하며, 저장할 수 있다.
ColorHunt https://colorhunt.co/	<ul style="list-style-type: none">커뮤니티 기반 팔레트사용자가 만든 수많은 색상 팔레트 통하여 트렌트를 반영할 수 있다.
ColorSpace https://mycolor.space/	<ul style="list-style-type: none">단일 색상을 지정하면 어울리는 색 조합으로 팔레트 생성
Paletton https://paletton.com/	<ul style="list-style-type: none">색상 휠을 이용해 조화를 이루는 색상 팔레트를 만들 수 있는 도구
Flat UI Colors 2 https://flatuicolors.com/	<ul style="list-style-type: none">플랫 디자인 색상, 국가별 팔레트
Material Design Color Tool https://material.io/resources/color/	<ul style="list-style-type: none">Google의 Material Design 가이드라인을 기반으로 색상 팔레트를 생성할 수 있는 사이트
Colormind http://colormind.io/	<ul style="list-style-type: none">AI 기반 팔레트 생성, 이미지에서 색상 추출, 부트스트랩 UI 미리보기

텍스트 스타일링

- 글꼴 스타일
- 텍스트 레이아웃 스타일

글꼴 스타일

font- family	글꼴 종류. 기본적으로 사용자 시스템의 글꼴을 이용하여 웹 브라우저 화면에 표시한다. 글꼴 스택이나 웹폰트를 사용할 수 있다.
font- size	글꼴 크기 px, em, rem
font- style	이탤릭체 표시 Normal, italic , oblique
font- weight	글자 굵기 normal, bold , lighter, bolder, 100~900
font	한 줄에서 모든 폰트 속성을 설정할 때 사용

font-family

```
body {  
  font-family: "Times New Roman", Times, serif;  
}
```

Generic Font Family

- 우선순위가 가장 높은 글꼴부터 가장 낮은 글꼴까지 하나 이상의 글꼴 목록을 지정한다.
- 글꼴 이름에 공백이 포함되는 경우 따옴표로 묶는다.
- 글꼴 이름을 2개 이상 지정하면, 첫번째 글꼴이 없을 경우 순서대로 다음 글꼴을 찾아 사용한다.
- 지정한 폰트를 사용할 수 있다는 보장이 없으므로, 마지막에는 Generic Font Family(폰트 계열)를 지정하여 해당 계열의 폰트를 사용할 수 있도록 한다.

font-family

▪ Generic Font Family (폰트 계열)

- 특정 폰트 이름이 아닌, 글꼴의 스타일이나 성격을 나타내는 더 광범위한 분류
- 웹에서 사용되는 다섯가지 Generic Font Family

구분	특징	주요폰트
serif	• 글자에 세리프(serif)라고 하는 작은 장식이 있는 글꼴	Times New Roman, Georgia
sans-serif	• 글자에 세리프가 없는, 깔끔하고 현대적인 느낌을 주는 글꼴	Arial, Verdana
monospace	• 모든 글자가 동일한 폭을 차지하는 글꼴	Courier New, Lucida Console
cursive	• 필기체 스타일의 글꼴	<i>Lucida Handwriting, Comic Sans MS,</i>
fantasy	• 장식적이고 독특한 스타일의 글꼴	impact , Papyrus

font-family

```
<style>
  .p1 {
    font-family: "Times New Roman", Times, serif;
  }

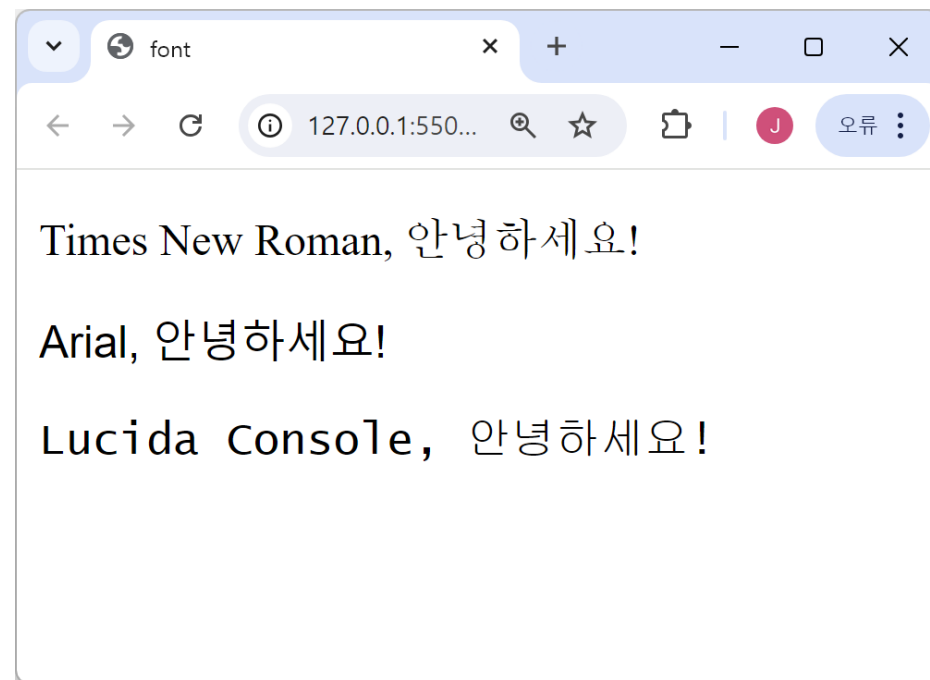
  .p2 {
    font-family: Arial, Helvetica, sans-serif;
  }

  .p3 {
    font-family: "Lucida Console", "Courier New", monospace;
  }
</style>
</head>
<body>
  <p class="p1">Times New Roman, 안녕하세요!</p>
  <p class="p2">Arial, 안녕하세요!</p>
  <p class="p3">Lucida Console, 안녕하세요!</p>
</body>
```

[참고]

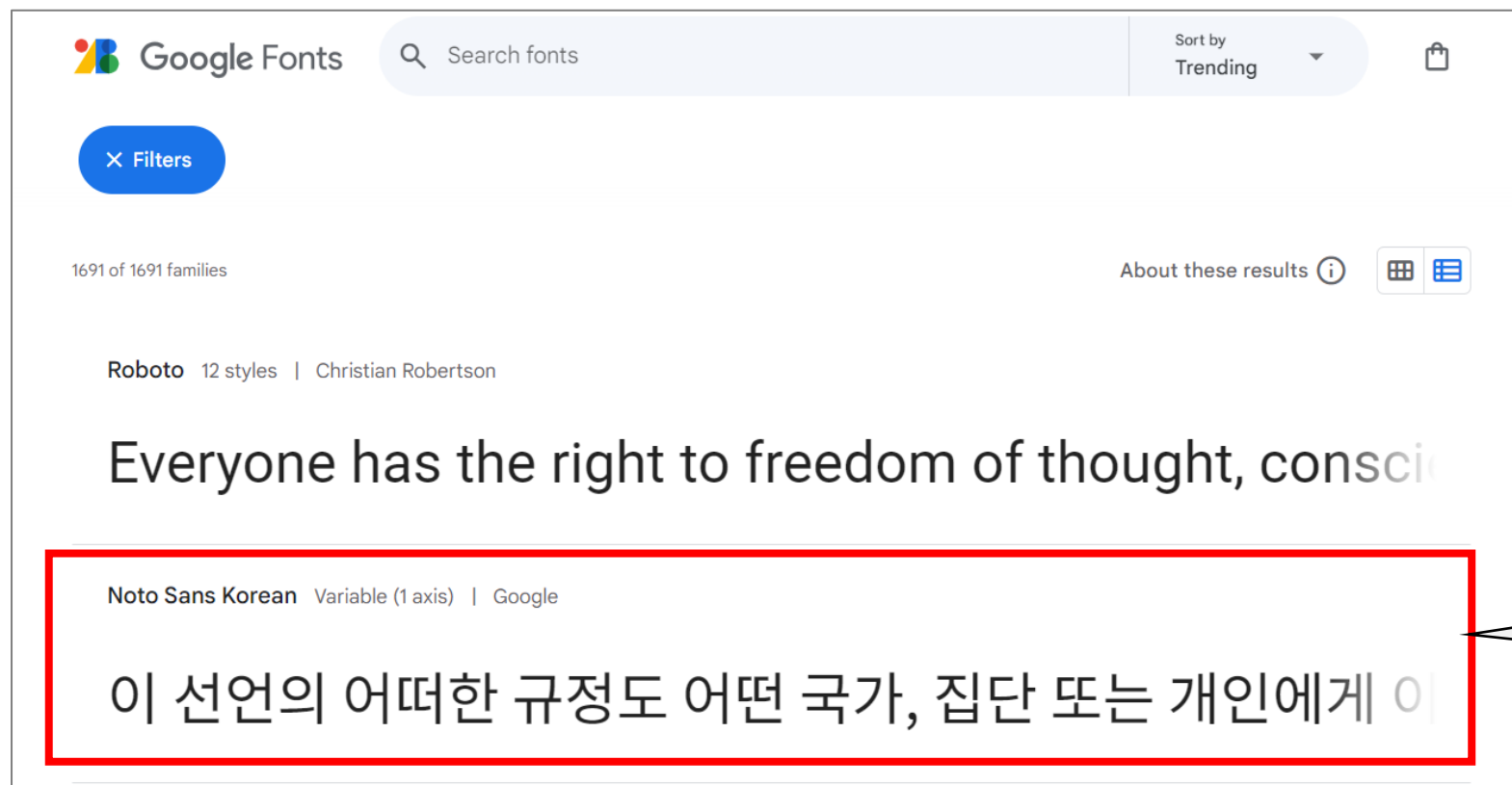
https://www.w3schools.com/css/css_font.asp

https://www.w3schools.com/css/css_font_websafe.asp



구글 웹폰트 사용하기

■ 구글 폰트(<https://fonts.google.com/>)



구글 웹폰트 사용하기

■ 폰트 가져오기

The screenshot shows the Google Fonts interface. At the top, there's a search bar with 'Search Noto fonts' and a shopping bag icon. Below the search bar are tabs for 'Specimen', 'Type tester', 'Glyphs', and 'About & license'. The 'Get font' button is highlighted with a red box. The main content area shows 'Noto Sans Korean' as the selected font family. Below this, it says '1 font family selected'. There are buttons for 'Share', 'Remove all', and 'Download all (1)'. The 'Get embed code' button is also highlighted with a red box. The text '이 선언의 어떠한 규정도 어떤 국가, 조' is visible at the bottom of the font preview area.

구글 웹폰트 사용하기

■ 폰트 가져오기

Web

Android

iOS

Flutter

☐ <link>

☒ @import

Embed code in the <head> of your html

```
<style>  
@import url('https://fonts.googleapis.com/css2?family=Noto+Sans+KR:wght@100..900&display=swap');  
</style>
```

Copy code

Noto Sans KR: CSS class for a variable style

```
// <uniquifier>: Use a unique and descriptive class name  
// <weight>: Use a value from 100 to 900  
  
.noto-sans-kr-<uniquifier> {  
  font-family: "Noto Sans KR", sans-serif;  
  font-optical-sizing: auto;  
  font-weight: <weight>;  
  font-style: normal;  
}
```

Copy code

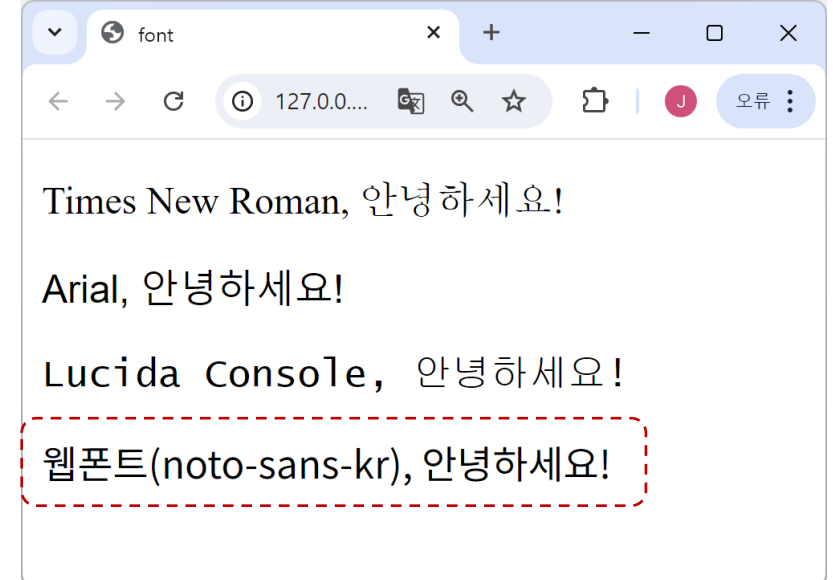
구글 웹폰트 사용하기

■ 웹문서에 스타일 넣기

main.css (모든 페이지에서 공통으로 사용되는 css)

```
@import url("https://fonts.googleapis.com/css2?family=Noto+Sans+KR:wght@100..900&display=swap");  
.noto-sans-kr {  
  font-family: "Noto Sans KR", sans-serif;  
}
```

```
<link rel="stylesheet" href="main.css" />  
</head>  
<body>  
  <p class="p1">Times New Roman, 안녕하세요!</p>  
  <p class="p2">Arial, 안녕하세요!</p>  
  <p class="p3">Lucida Console, 안녕하세요!</p>  
  <p class="noto-sans-kr">웹폰트(noto-sans-kr), 안녕하세요!</p>  
</body>
```



font-size

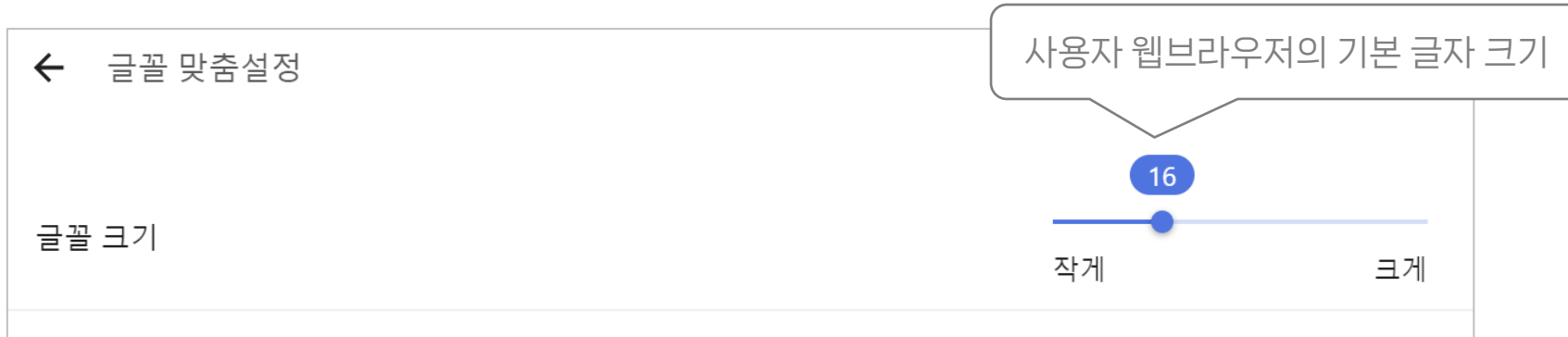
단위	설명
px	<ul style="list-style-type: none">디스플레이 화면의 1pixel 기준.디스플레이 해상도나 화면 크기에 따라 다르게 보일 수 있다.
em	<ul style="list-style-type: none">부모 요소의 글자 크기에 대한 상대적인 글자 크기
rem	<ul style="list-style-type: none">root(html) 요소의 글자 크기에 대한 상대적인 글자 크기
%	<ul style="list-style-type: none">부모 요소의 글자 크기에 대한 상대적인 글자 크기
키워드	<ul style="list-style-type: none">웹브라우저의 기본 글자 크기에 대한 상대적인 글자 크기xx-small < x-small < small < medium < large < x-large < xx-large

웹브라우저의 기본 글자 크기

font-size

■ 내 브라우저의 글꼴 설정 확인하기

크롬브라우저기준 : 설정>모양>글꼴맞춤설정



※ 웹브라우저의 기본 글자 크기는 사용자가 바꿀 수 있다.

font-size

```
<style>
  html {
    font-size: 20px;
  }
  .div1 {
    border: 1px solid black;
    font-size: 30px;
  }
  .p4 {
    font-size: 15px;
  }
  .p5 {
    font-size: 0.5em;
  }
  .p6 {
    font-size: 0.5rem;
  }
  .p7 {
    font-size: 50%;
  }
  .p8 {
    font-size: medium;
  }
</style>
```

```
<div class="div1">
  저의 글자 크기는 30px입니다.
  <p class="p4">저의 글자 크기는 15px입니다.</p>
  <p class="p5">저의 글자 크기는 0.5em입니다.</p>
  <p class="p6">저의 글자 크기는 0.5rem입니다.</p>
  <p class="p7">저의 글자 크기는 50%입니다.</p>
  <p class="p8">저의 글자 크기는 medium입니다.</p>
</div>
```

저의 글자 크기는 30px입니다.

저의 글자 크기는 15px입니다.

저의 글자 크기는 0.5em입니다.

저의 글자 크기는 0.5rem입니다.

저의 글자 크기는 50%입니다.

저의 글자 크기는 medium입니다.

저는 몇 px일까요?

저는 몇 px일까요?

저는 몇 px일까요?

저는 몇 px일까요?

font-style

■ 이탤릭체 지정

단위	설명
normal	<ul style="list-style-type: none">일반 글자 모양
<i>italic</i>	<ul style="list-style-type: none">이탤릭체 (모양도 변형됨)대부분의 세리프 폰트와 일부 산세리프 폰트에서 제공
oblique	<ul style="list-style-type: none">텍스트 기울임 (기계적으로 기울이기만 함)일부 산세리프 폰트에서 제공

font-weight

■ 글꼴의 두께 지정

키워드로 지정	• <code>Lighter</code> < <code>normal</code> < <code>bold</code> < <code>bolder</code>
숫자로 지정	• <code>100</code> < <code>200</code> < <code>300</code> < <code>400</code> < <code>500</code> < <code>600</code> < <code>700</code> < <code>800</code> < <code>900</code>

font-weight

```
<div class="div1">  
  <p>저는 일반 글자모양입니다.</p>  
  <p style="font-weight: lighter">저는 lighter입니다.</p>  
  <p style="font-weight: bold">저는 bold입니다.</p>  
  <p style="font-weight: bolder">저는 bolder입니다.</p>  
  <p style="font-style: italic">저는 italic입니다.</p>  
  <p style="font-style: oblique">저는 oblique입니다.</p>  
</div>
```

저는 일반 글자모양입니다.

저는 lighter입니다.

저는 **bold**입니다.

저는 **bolder**입니다.

저는 *italic*입니다.

저는 *oblique*입니다.

font

■ 폰트 축약 기법

```
.p10 {  
  font: italic bold 50px Arial;  
}
```

- 폰트 속성을 한 줄에 지정한다.
- Font-style font-weight font-size font-family 순으로 지정한다.
- Font-size와 font-family는 필수로 지정해야 한다.

텍스트 레이아웃 스타일

스타일 속성	설명	값
text-align	텍스트 정렬	left, right, center, justify(양쪽맞춤)
text-decoration	밑줄, 윗줄, 취소선	none, <u>underline</u> , <u>overline</u> , line-through
text-shadow	그림자 효과	가로거리(px) 세로거리(px) 번짐정도(px) 색상
letter-spacing	글자 간격 조절	px, em
word-spacing	단어 간격 조절	px, em
line-height	줄 간격	px, 배수, %(font-size 기준)

미션

- [어린왕자] 전문의 일부를 CSS스타일로 꾸미기
 - div로 제목, 작가 등 구간 나누고, 각각의 구간에 다른 스타일 지정
 - 제목 : 가운데정렬, 폰트사이즈 크게, 볼드체, 그림자효과 등
 - 작가 : 오른쪽정렬, 폰트사이즈 작게, 이탤릭체 등
 - 본문 : 양쪽정렬, 글자간격, 단어간격, 줄간격 등
 - 특정 글자 강조하고 꾸미기
 - 특정글자에 밑줄, 특정글자에 백그라운드컬러 등...
 - 이미지 추가하기
 - 코끼리를 삼키고 있는 보아뱀
 - 그림 2호

어린왕자

생똥쥐베리

여섯 살 때 나는 원시림에 관한 이야기가 있는 '실제 경험담'이라는 책에서 멋진 그림 하나를 보았다. 그것은 코끼리를 집어삼키고 있는 보아뱀 그림이었다. 위의 그림이 그것을 복사한 것이다. 그 책에는 이렇게 써어 있었다. "보아뱀은 먹이를 씹지 않고 통째로 삼킨다. 그리고 나서 몸을 움직일 수 없게 되어 여섯 달 동안 잠을 자면서 먹이를 소화시킨다." 나는 정글의 모험들에 대해 깊이 생각해 보았다. 그리고 나는 색연필로 나의 첫 번째 그림을 그리는 데 성공했다. 나의 그림 제1호였다. 그것은 이렇게 생겼다. 나는 나의 걸작을 어른들에게 보여주며 그림을 보고 무서워하는지 물어보았다. 어른들은 "모자가 무서워 보이긴 하네."라고 대답했다. 내 그림은 모자가 아니었다. 그것은 코끼리를 소화하고 있는 보아뱀이었다. 그래서 나는 어른들이 이해할 수 있도록 보아뱀의 속을 그렸다. 어른들은 항상 설명이 필요하다. 나의 그림 제2호는 이렇게 생겼다.

- 어린왕자 전문의 앞부분에서 발췌한 내용입니다. -

CSS 박스모델

- CSS 박스모델의 개념
- 블록레벨요소, 인라인레벨요소
- 박스모델 디자인

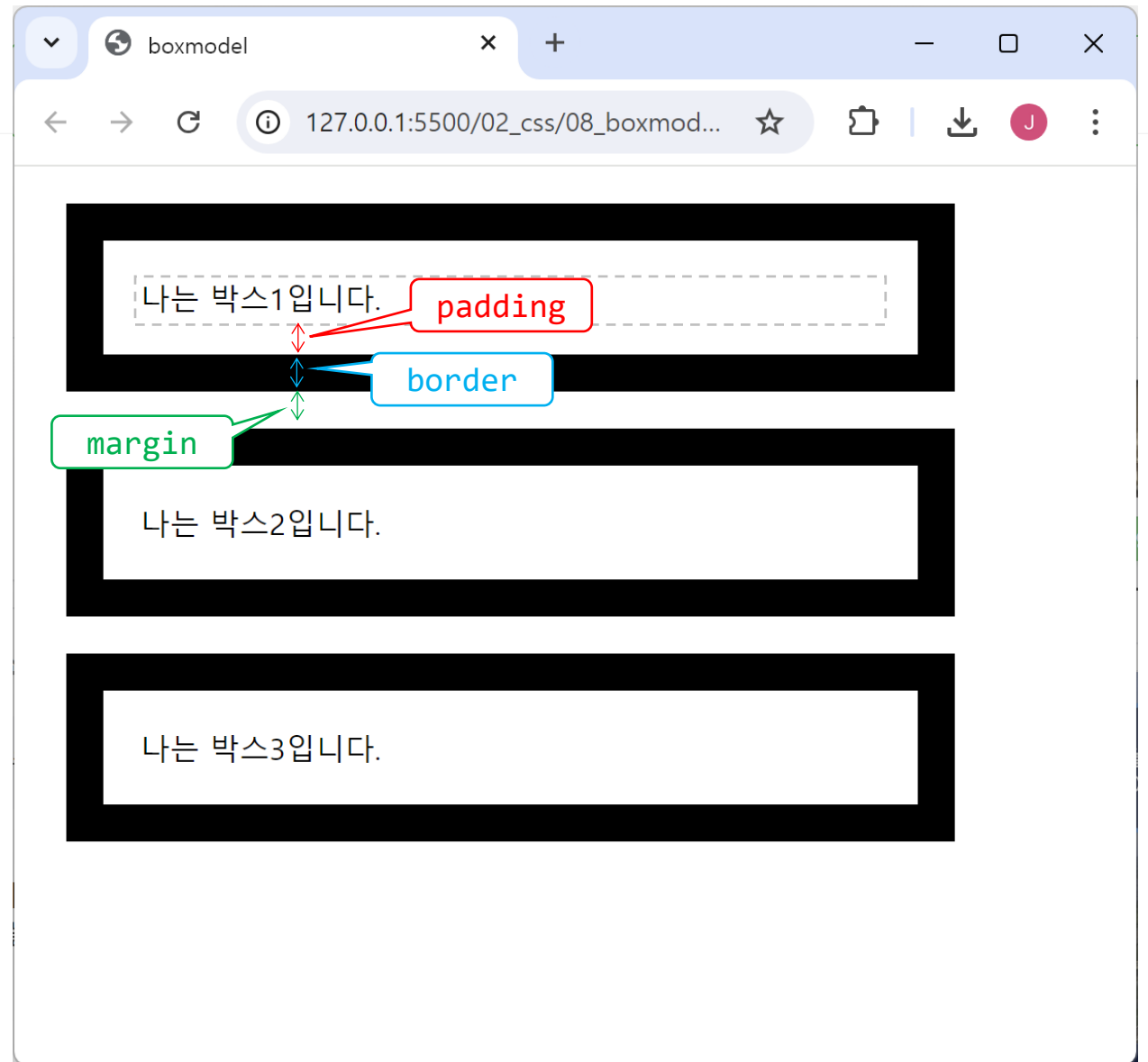
박스 모델(box model)

- 웹페이지의 모든 요소들을 사각형 박스로 나타내는 방법
 - 웹브라우저는 각 요소들을 박스(box)로 간주하고 화면에 배치한다
- 박스모델의 구성요소
 - content : 내용
 - padding : 콘텐츠 주변을 채운다.
 - border : 패딩과 콘텐츠를 둘러싸는 테두리
 - margin : 테두리 바깥쪽 영역



박스 모델(box model)

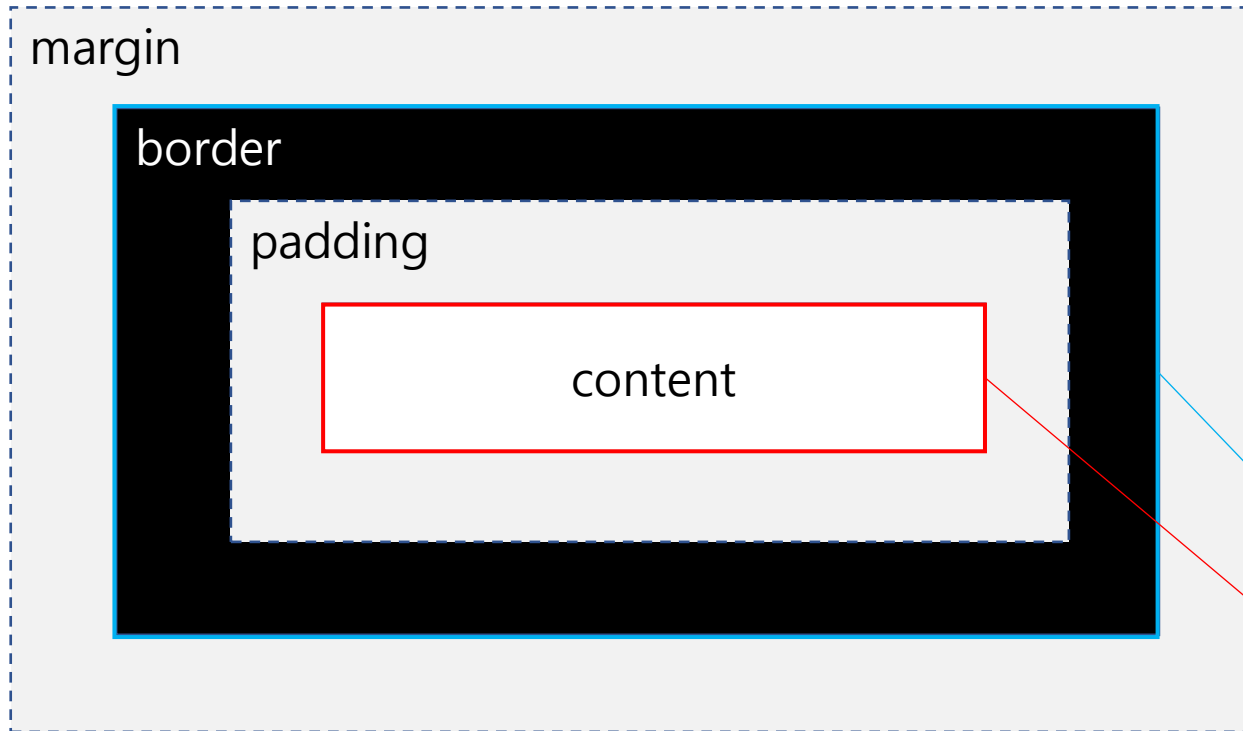
```
<style>
  .box1,
  .box2,
  .box3 {
    width: 400px;
    border: 20px solid black;
    padding: 20px;
    margin: 20px;
  }
</style>
</head>
<body>
  <div class="box1">나는 박스1입니다.</div>
  <div class="box2">나는 박스2입니다.</div>
  <div class="box3">나는 박스3입니다.</div>
</body>
```



박스 스타일

스타일 속성		설명	값
box-sizing		width, height 지정 기준	border-box, content-box(디폴트)
box-shadow		박스에 그림자 효과 지정	수평거리, 수직거리, 흐림정도, 번짐정도, 색상, inset
border	border-style	박스의 테두리 모양 지정	none, hidden, Solid, dotted, dashed, double, grove, inset, outset, ridge
	border-width	박스 테두리 두께 지정	top right, bottom, left 별도 지정 가능
	border-color	박스 테두리 색상 지정	top right, bottom, left 별도 지정 가능
	border-radius	박스 테두리 둥글게	top right, bottom, left 별도 지정 가능
padding		박스 테두리 내부 여백	top right, bottom, left 별도 지정 가능
margin		박스 테두리 외부 여백	top right, bottom, left 별도 지정 가능

box-sizing



단위	설명
px	고정값
%	부모 요소 기준 상대적 크기
auto	콘텐츠 양에 따라 자동 결정

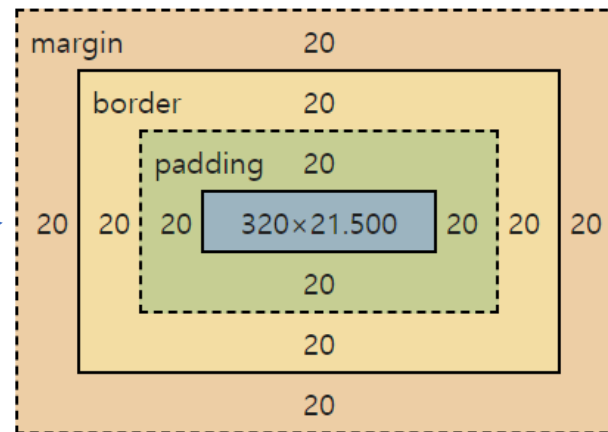
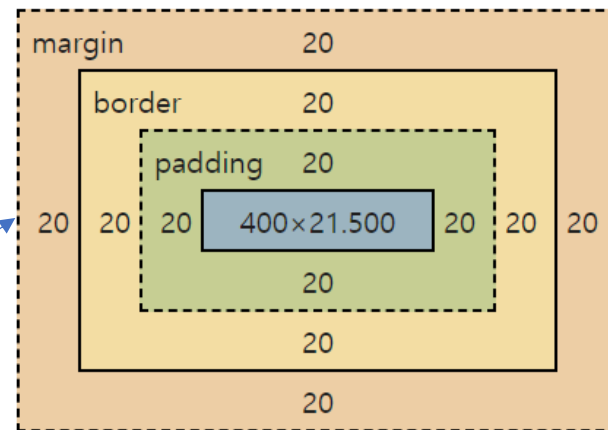
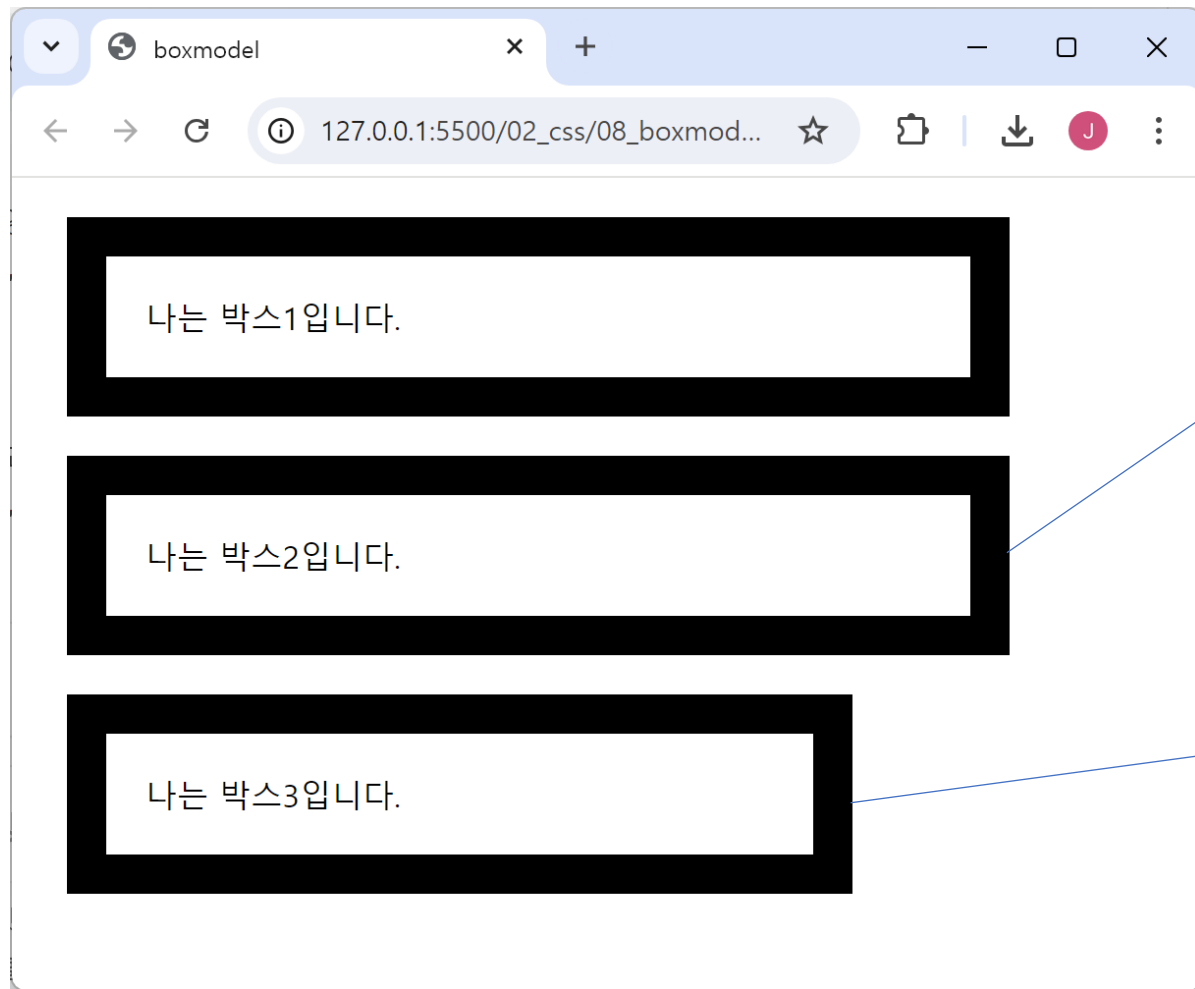
종류	설명
border-box	테두리까지 포함하여 너비값 지정 (권장)
content-box	콘텐츠 영역만으로 너비값 지정 (default)

box-sizing

```
<style>
.box1,
.box2,
.box3 {
  width: 400px;
  border: 20px solid black;
  padding: 20px;
  margin: 20px;
}
.box3 {
  box-sizing: border-box;
}
</style>
```



box-sizing



box-shadow

■ 박스에 그림자 지정하기 위한 옵션

아래 옵션을 순서대로 작성한다.

종류	설명
수평거리(필수)	양수 : 오른쪽 그림자, 음수:왼쪽 그림자
수직거리(필수)	양수: 아래쪽 그림자, 음수:위쪽 그림자
흐림정도	0을 기준으로 하여 값이 커질수록 부드러운 그림자 표시(음수x)
번짐정도	양수 : 그림자 퍼짐, 음수 : 그림자 축소
색상	그림자 색상
inset	키워드를 설정하면 안쪽 그림자로 그림

box-shadow

```
.box4 {  
  box-sizing: border-box;  
  width: 400px;  
  height: 200px;  
  border: 1px solid black;  
  margin: 20px;  
  padding: 20px;  
  box-shadow: 20px 20px 5px #eee;  
}
```



박스모델의 테두리 설정

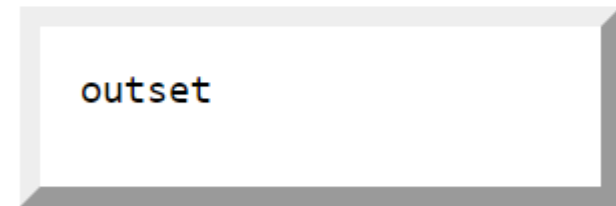
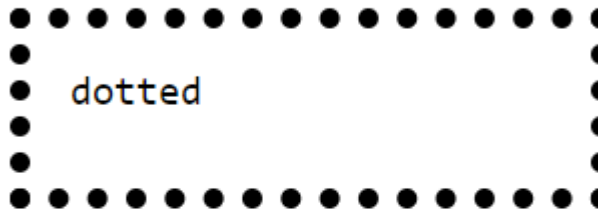
속성	설명
border-style	테두리 스타일 설정
border-width	테두리 두께 설정
border-color	테두리 색상 설정
border	위 세가지 속정을 한번에 설정

border-style

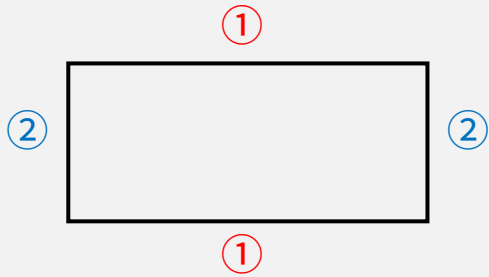
`none`(테두리없음.기본값)



`hidden`

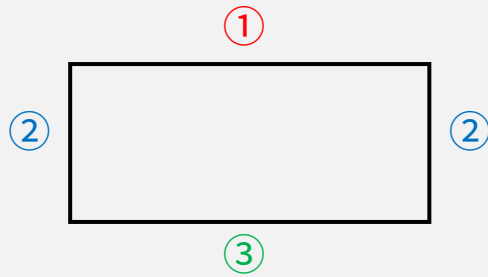


border-width



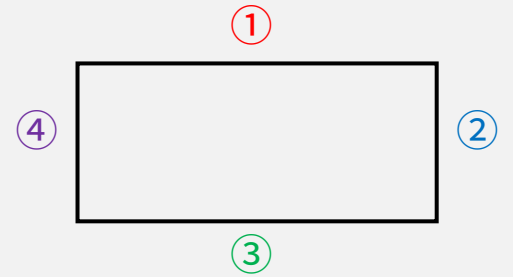
```
border-width: 1px 5px
```

box-width



```
border-width: 5px 3px 1px
```

box-width



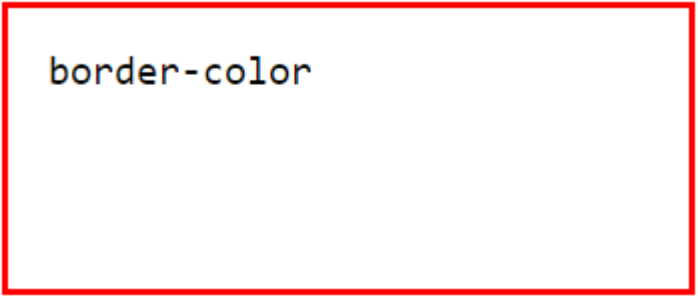
```
border-width: 1px 5px 10px 15px;
```

box-width

border-color

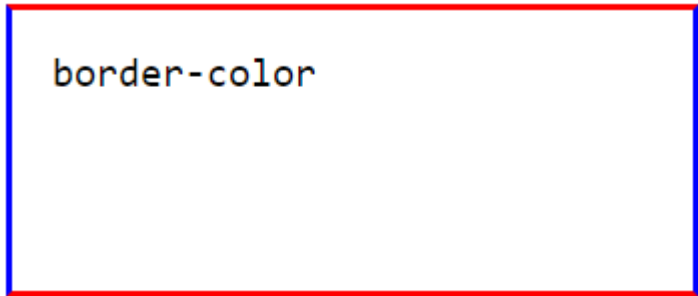
```
border-color: red;
```

border-color



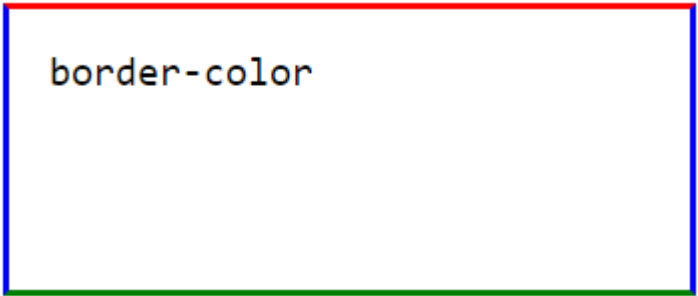
```
border-color: red blue;
```

border-color



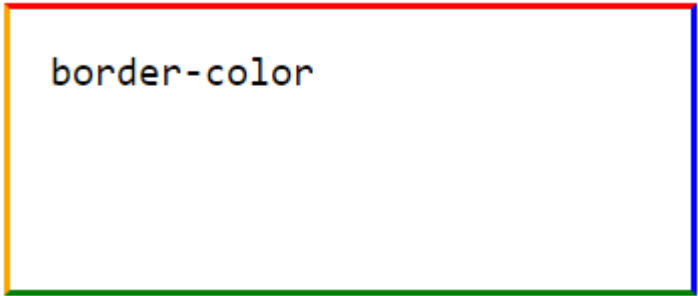
```
border-color: red blue green;
```

border-color



```
border-color: red blue green orange;
```

border-color



border-color

```
border-top-color: red;
```


border-color

```
border-top-color: red;  
border-right-color: blue;
```

border-color

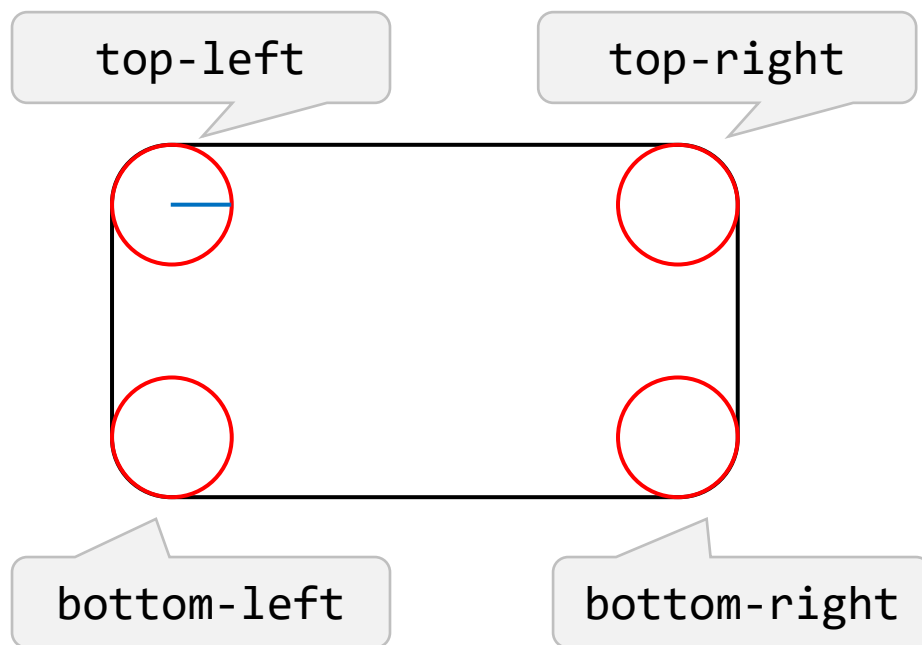
border

```
border: solid green 3px;
```



border

border-radius



속성	설명
크기	반지름의 크기를 px, em의 단위와 함께 수치로 표현
백분율	현재 요소의 크기를 기준으로 비율(%),로 지정

border-radius

```
border-radius: 20px;
```



```
border-radius: 40px 20px;
```



```
border-radius: 20px 40px 100px;
```



```
border-radius: 20px 40px 60px 80px;
```



border-radius

```
border-radius: 50%;
```



```
border-top-left-radius: 20px;  
border-top-right-radius: 20px;
```



```
border-radius: 40px / 20px;
```

가로 반지름

세로 반지름



여백 조절하기



padding	콘텐츠 영역과 border간 여백 조정 (border 안쪽) <ul style="list-style-type: none">paddingpadding-toppadding-bottompadding-leftpadding-right
margin	요소와 요소 간 간격 조절 (border 바깥) <ul style="list-style-type: none">marginmargin-topmargin-bottommargin-leftmargin-right

padding

```
padding: 10px;
```



```
padding: 10px 20px;
```



```
padding: 10px 20px 30px;
```

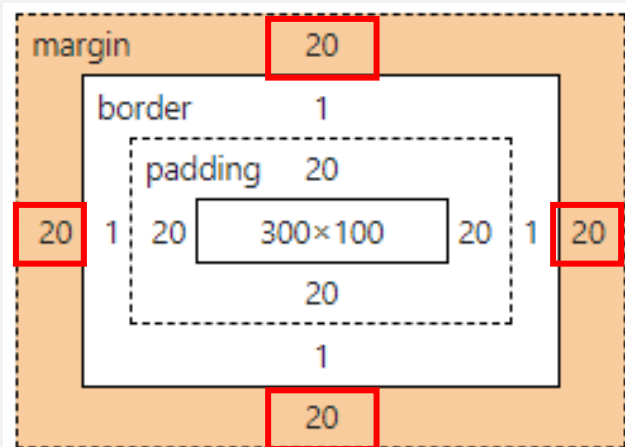


```
padding: 10px 20px 30px 40px;
```

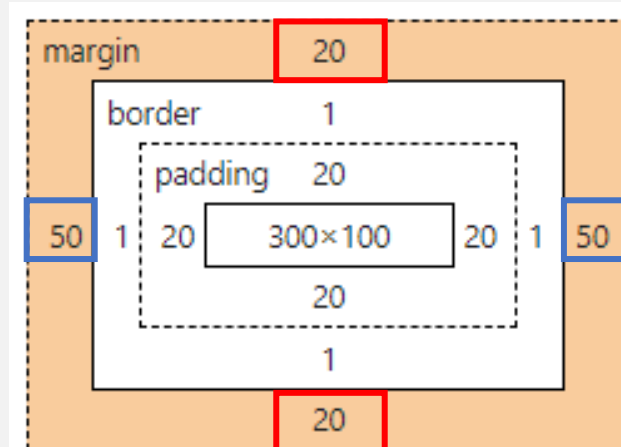


margin

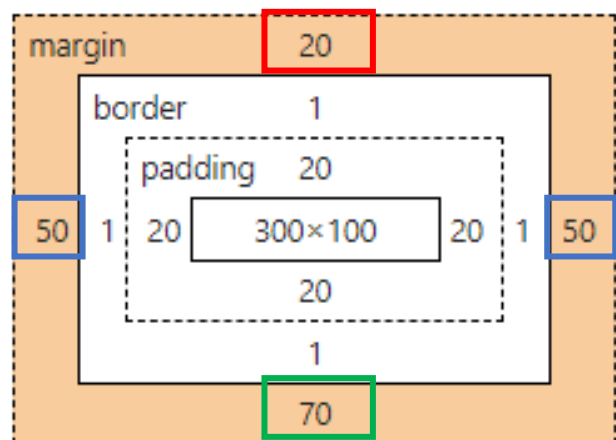
`margin: 20px;`



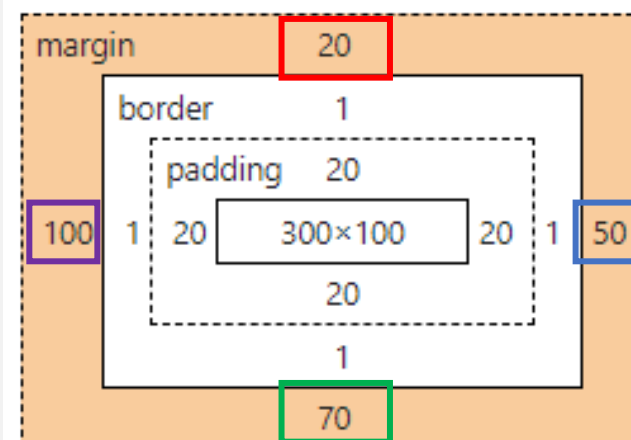
`margin: 20px 50px;`



`margin: 20px 50px 70px;`

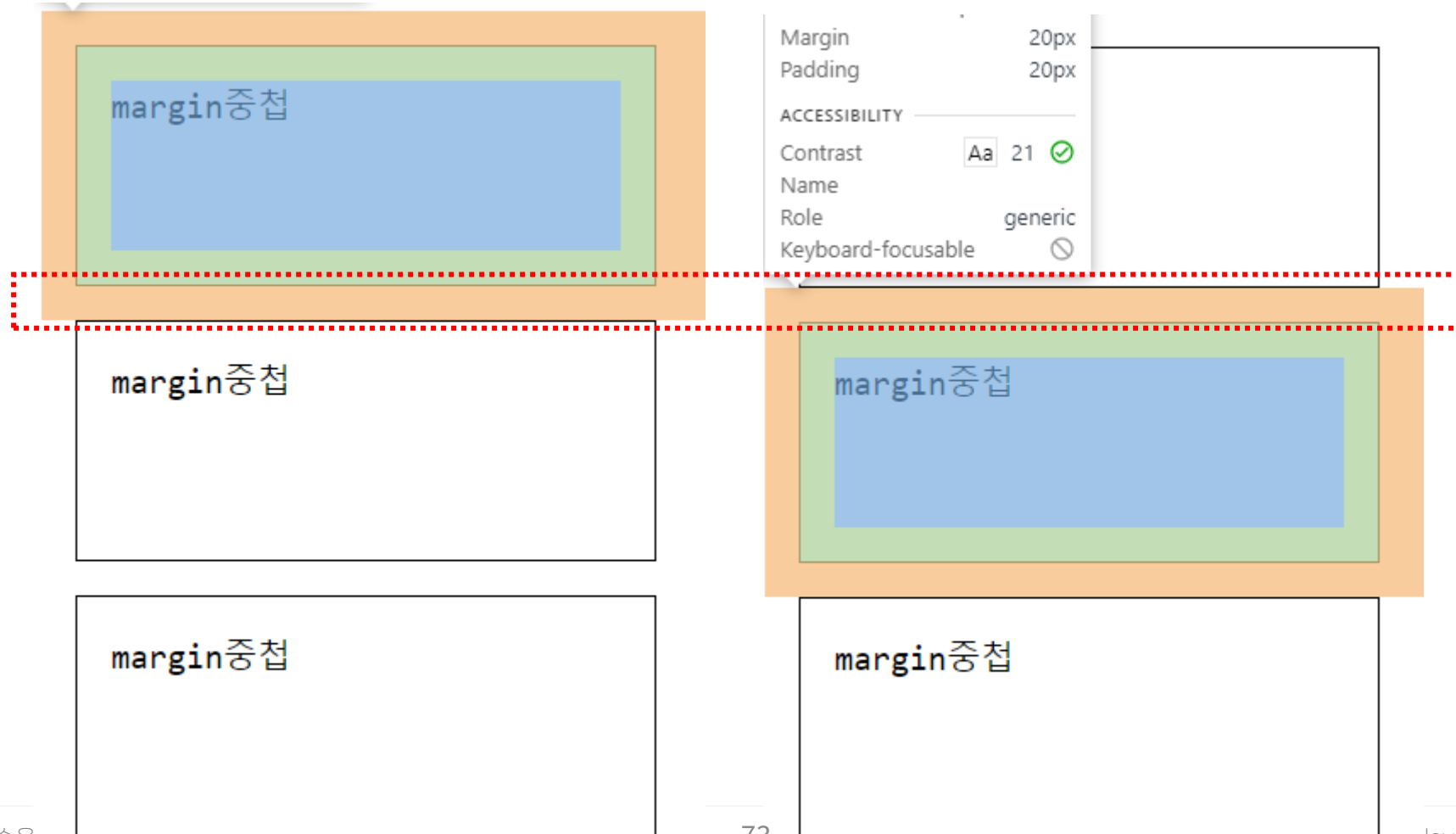


`margin: 20px 50px 70px 100px;`

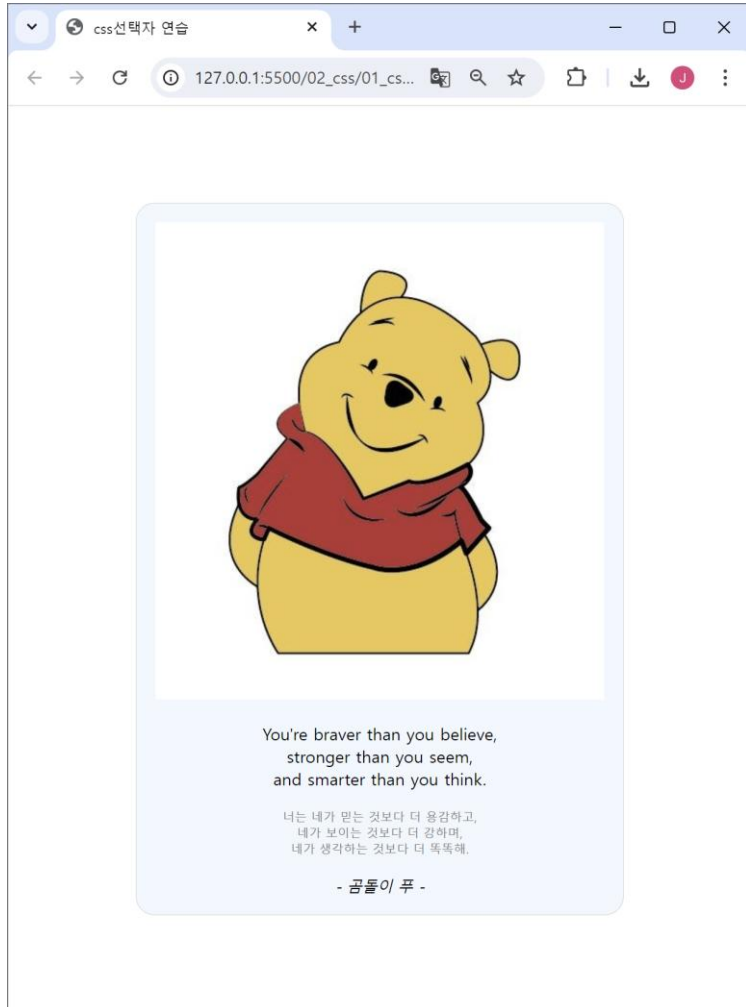


margin 중첩

박스모델 요소를 세로로 배치할 때 margin 중첩을 고려하여 디자인 한다.



명언 카드 만들기



- 명언, 이미지 준비
- div로 구분하여 전체 틀 만들기
- 콘텐츠 넣기
- 테두리, 배경색 등 넣기
- margin, padding 조정하기
- 폰트 꾸미기 등

목록 스타일

속성명	설명
<code>list-style-type</code>	bullet의 모양 지정 (ol, ul 공통 사용)
<code>list-style-position</code>	들여쓰기 (inside, outside)
<code>list-style-image</code>	bullet 이미지 지정
<code>list-style</code>	목록 속성을 한번에 표시

list-style-type

ol, ul 공통 적용

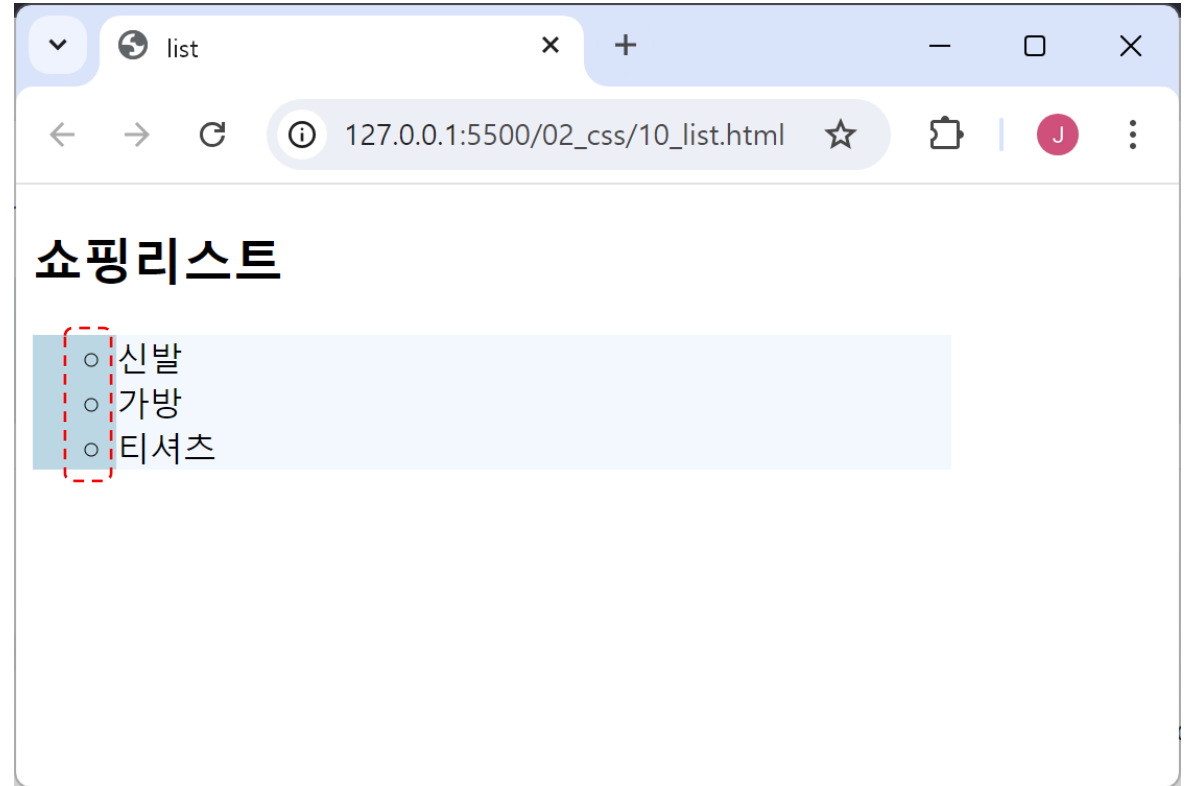
종류	모양
disc	●
circle	○
square	■

- HTML
- CSS
- Javascript

종류	모양
decimal	1,2,3,...
decimal-leading-zero	01,02,...
lower-roman	i, ii, iii
upper-roamn	I, II, III, ...
lower-alpha, lower-latin	a,b,c,...
upper-alpha, upper-latin	A,B,C,...
none	※ 표시하지 않음

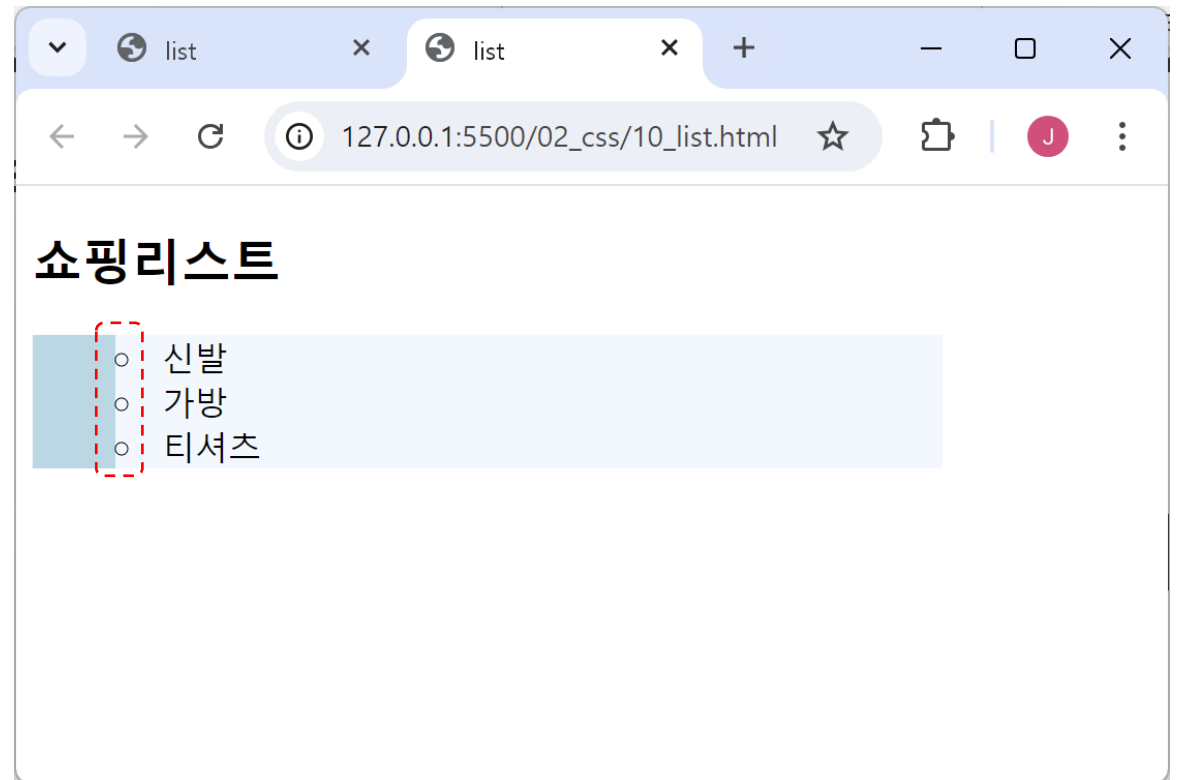
list-style-type

```
<style>
  ol {
    background-color: lightblue;
    width: 400px;
    list-style-type: circle;
  }
  li {
    background-color: aliceblue;
  }
</style>
</head>
<body>
  <h2>쇼핑리스트</h2>
  <ol>
    <li>신발</li>
    <li>가방</li>
    <li>티셔츠</li>
  </ol>
</body>
```



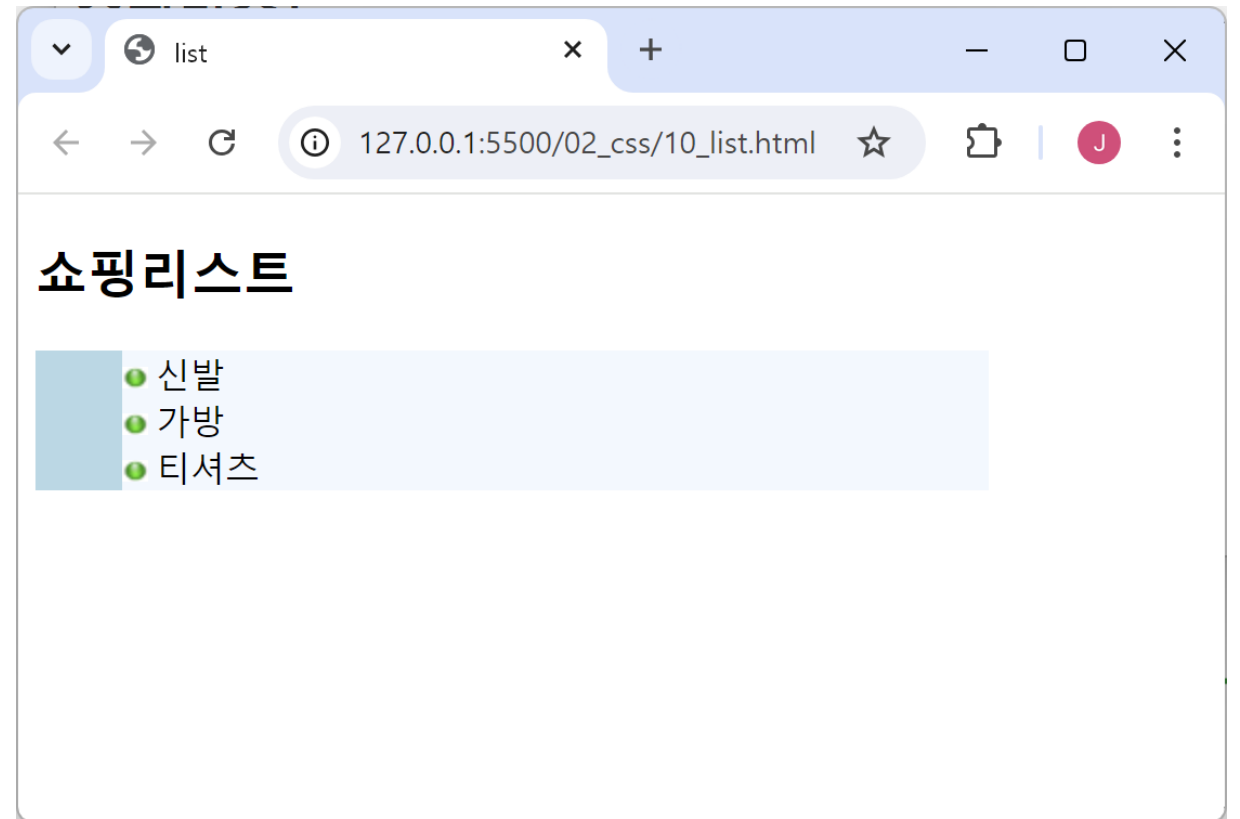
list-style-position

```
<style>
  ol {
    background-color: lightblue;
    width: 400px;
    list-style-type: circle;
    list-style-position: inside;
  }
  li {
    background-color: aliceblue;
  }
</style>
</head>
<body>
  <h2>쇼핑리스트</h2>
  <ol>
    <li>신발</li>
    <li>가방</li>
    <li>티셔츠</li>
  </ol>
</body>
```



list-style-image

```
<style>
  ol {
    background-color: lightblue;
    width: 400px;
    list-style-type: circle;
    list-style-position: inside;
    list-style-image: url("images/dot.png");
  }
  li {
    background-color: aliceblue;
  }
</style>
</head>
<body>
  <h2>쇼핑리스트</h2>
  <ol>
    <li>신발</li>
    <li>가방</li>
    <li>티셔츠</li>
  </ol>
</body>
```



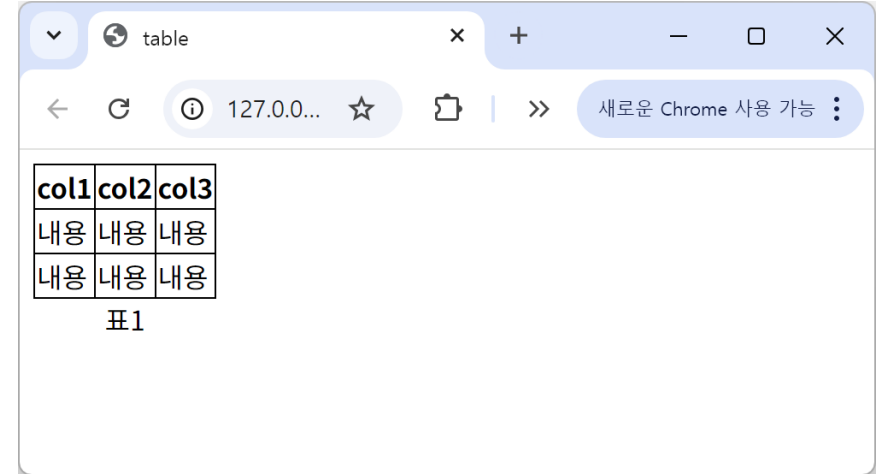
테이블 스타일

스타일 속성	설명	속성값
border	표 테두리	선두께 선종류 선색상 순서대로 적어준다.
border-collapse	표와 셀 테두리 합치기	collapse separate
caption-side	표 제목의 위치 설정	top bottom

테이블 스타일

```
<style>
  table,
  th,
  td {
    border: 1px black solid;
    border-collapse: collapse;
    caption-side: bottom;
  }
</style>
```

```
<table>
  <caption>
    표1
  </caption>
  <tr>
    <th>col1</th>
    <th>col2</th>
    <th>col3</th>
  </tr>
  <tr>
    <td>내용</td>
    <td>내용</td>
    <td>내용</td>
  </tr>
  <tr>
    <td>내용</td>
    <td>내용</td>
    <td>내용</td>
  </tr>
</table>
```



반응형 웹과 미디어 쿼리

- ✓ 반응형 웹의 개념
- ✓ 뷰포트
- ✓ 미디어쿼리.

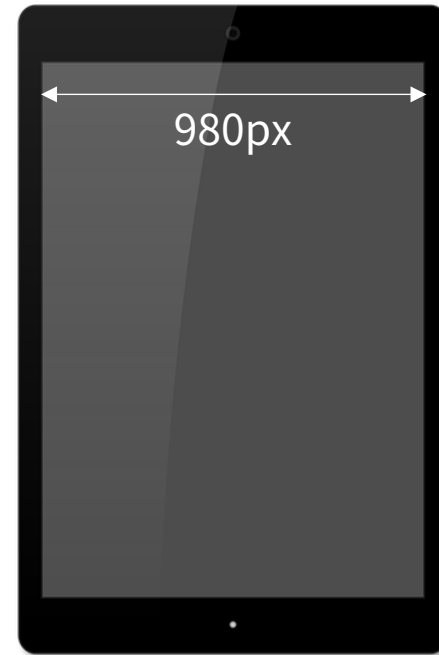
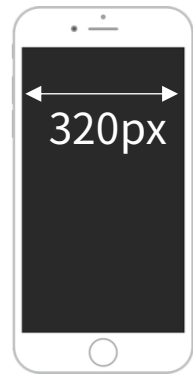
반응형 웹

화면의 크기에 반응하는 반응형 웹



뷰포트 (viewport)

사용자의 화면에서 웹 페이지가 실제로 표시되는 영역



뷰포트(viewport) 지정

<head></head> 사이에 <meta> 태그를 이용하여 지정한다.

```
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
```

속성	설명
width	뷰포트 너비(px 또는 device-width)
height	뷰포트 높이(px 또는 device-height)
user-scalable	확대/축소 가능 여부(yes, no)
initial-scale	초기 확대/축소 값(1~10)

뷰포트(viewport) 단위

화면의 크기에 기반하여 동적으로 계산되는 상대적인 측정 단위

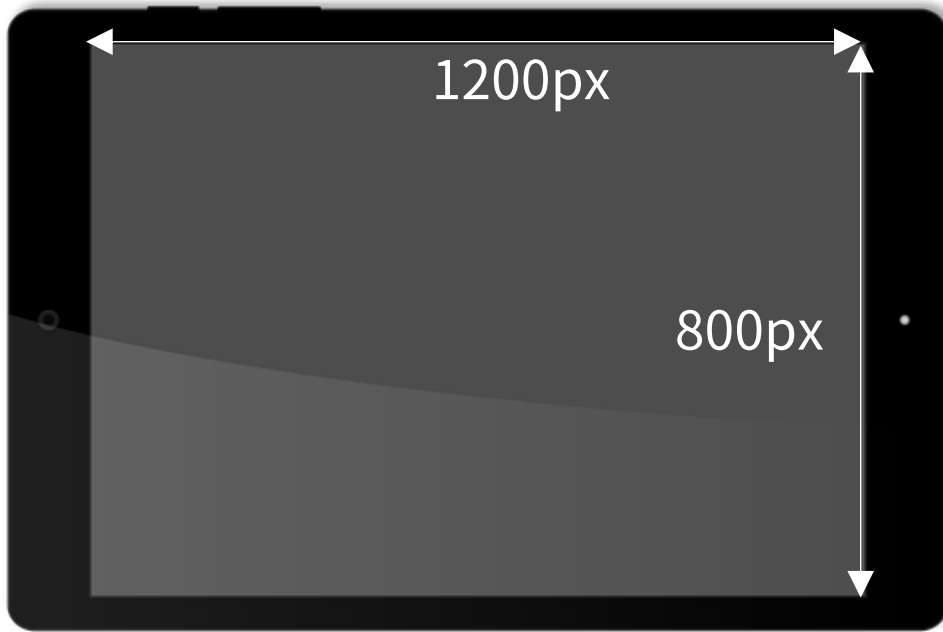
단위	설명
vw	(viewport width) 1vw는 뷰포트 너비의 1%에 해당
vh	(viewport height) 1vh는 뷰포트 높이의 1%에 해당
vmin	(viewport minimum) 1vmin은 뷰포트 너비, 높이 중 작은 값의 1%
max	(viewport maximum) 1vmax는 뷰포트 너비, 높이 중 큰 값의 1%

뷰포트(viewport) 단위



- 1 vw = px
- 1 vh = px

뷰포트(viewport) 단위

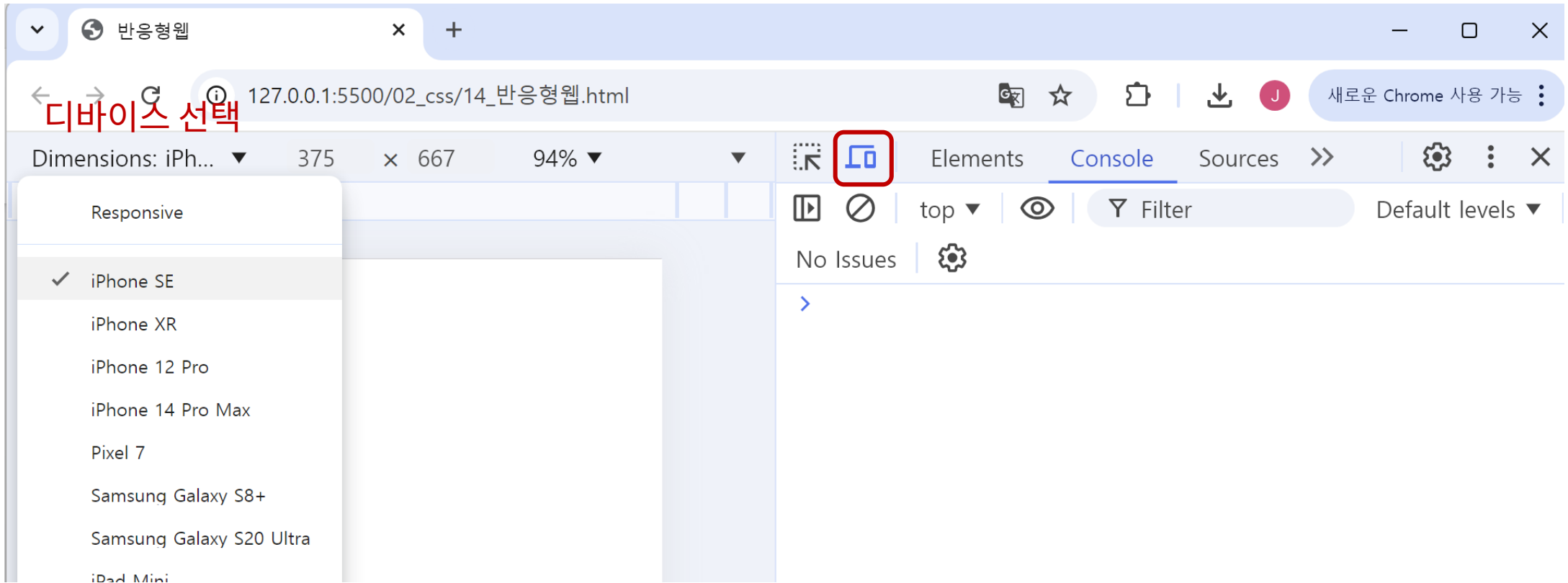


- 1 vw = px
- 1 vh = px

뷰포트 기준 폰트사이즈

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>반응형웹</title>
    <style>
      div {
        font-size: 5vw;
      }
    </style>
  </head>
  <body>
    <div>Hello!</div>
  </body>
</html>
```

디바이스 크기에 따른 웹페이지 확인하기



미디어 쿼리

■ 사이트에 접속하는 장치 및 조건에 따라 다른 스타일시트를 사용하는 방법

@media [only | not] 미디어유형 [and 조건]

미디어 유형	설명	미디어 유형	설명
all	모든 미디어 유형	braille	점자 표시 장치
screen	PC, 모바일 스크린	handheld	손에 들고 다니는 장치(pad 등)
print	인쇄장치	projection	프로젝터
tv	TV	tty	디스플레이 기능이 제한된 장치
aural	음성합성장치	embossed	점자 프린터

미디어 쿼리

- 사이트에 접속하는 장치에 따른 스타일시트 작성

```
@media screen{  
  
/*화면용 스타일 작성*/  
  
}
```

미디어 쿼리

■ 뷰포트의 크기에 따른 스타일시트 작성

```
@media screen and (min-width:1440px){  
  
/뷰포트 너비가 최소 1440px인(1440px 이상인) 화면용 스타일*/  
  
}
```

종류	설명
width	너비
height	높이
min-width	최소너비
min-height	최소높이
max-width	최대너비
max-height	최대높이

미디어 쿼리

■ 화면 회전 속성에 따른 스타일시트 작성

```
@media screen and (min-width:812px) and (orientation:portrait){  
  
/*뷰포트 너비가 최소 812px인(812px 이상인) 세로모드 화면용 스타일*/  
  
}
```

종류	설명
portrate	세로모드
landscape	가로모드

미디어 쿼리

■ 화면 크기에 따라 색깔 바꾸기

```
<style>
  div {
    font-size: 5vw;
  }
  body {
    background-color: yellow;
  }
  @media screen and (min-width: 768px) {
    body {
      background-color: blue;
    }
  }
  @media screen and (min-width: 1024px) {
    body {
      background-color: red;
    }
  }
</style>
```

일문서 display 만들기

block	<ul style="list-style-type: none">• 항상 새로운 줄에서 시작• width, height 설정 가능• margin, padding 모든 방향 적용 가능
inline	<ul style="list-style-type: none">• 줄 바꿈 없이 흐름대로 배치• width, height 설정 불가• margin, padding 좌우만 적용 가능
inline-block	<ul style="list-style-type: none">• 줄 바꿈 없이 흐름대로 배치(inline)• width, height 설정 가능(block)• margin, padding 모든 방향 적용 가능(block)

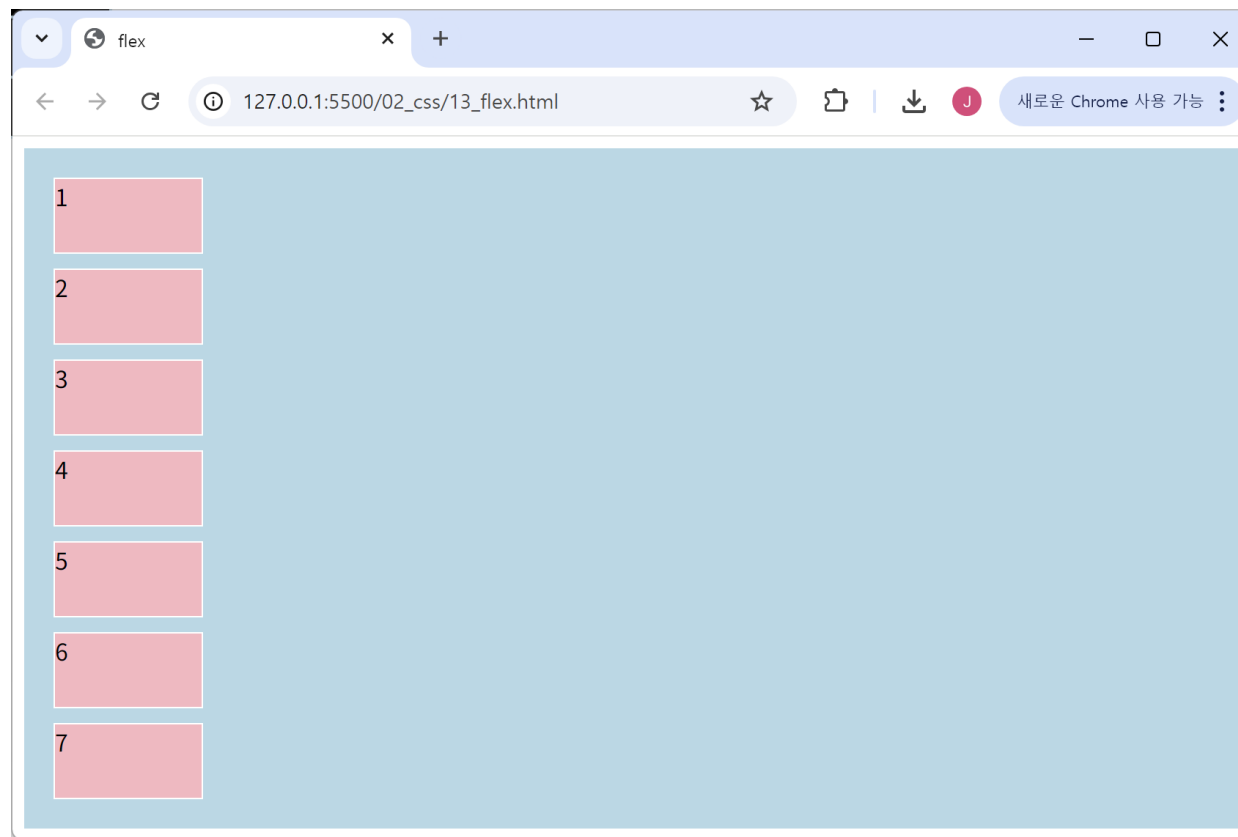
flex	<ul style="list-style-type: none">• 유연한 레이아웃 구성• 1차원적 레이아웃• 자식 요소의 크기와 순서를 쉽게 제어
grid	<ul style="list-style-type: none">• 2차원 레이아웃 구성• 행과 열을 동시에 제어• 복잡한 레이아웃에 적합
none	<ul style="list-style-type: none">• 요소를 완전히 숨김• 공간도 차지하지 않음

display

```
<style>
  #container {
    background-color: lightblue;
    padding: 10px;
  }
  .flexbox {
    background-color: lightpink;
    border: 1px solid white;
    width: 100px;
    height: 50px;
    margin: 10px;
  }
</style>
```

```
<div id="container">
  <div class="flexbox">1</div>
  <div class="flexbox">2</div>
  <div class="flexbox">3</div>
  <div class="flexbox">4</div>
  <div class="flexbox">5</div>
  <div class="flexbox">6</div>
  <div class="flexbox">7</div>
</div>
```

<div></div>는 블록레벨요소라서 줄바꿈이 되어 배치되었다.
줄바꿈이 되지 않도록 배치하려면?



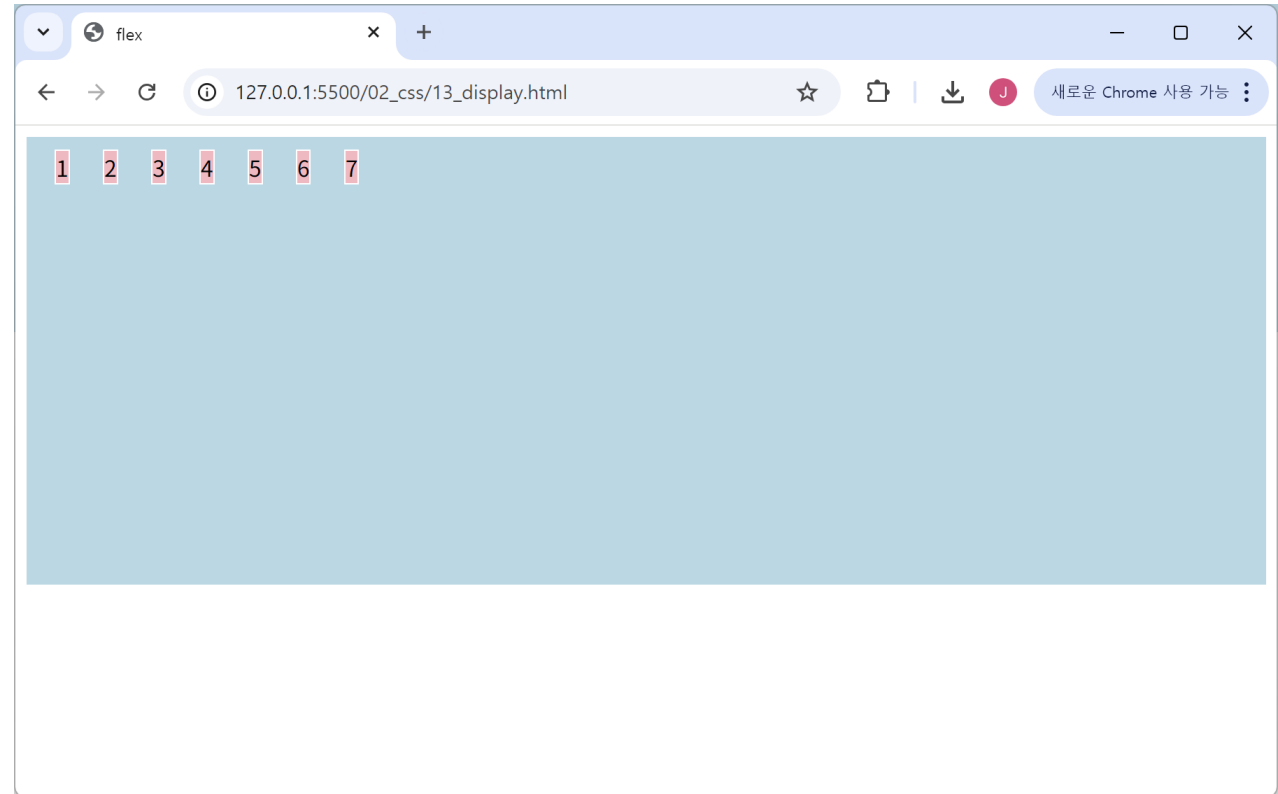
display:inline;

```
<style>
  #container {
    background-color: lightblue;
    padding: 10px;
  }
  .box {
    background-color: lightpink;
    border: 1px solid white;
    width: 100px;
    height: 50px;
    margin: 10px;
    display: inline;
  }
</style>
```

```
<div id="container">
  <div class="box">1</div>
  <div class="box">2</div>
  <div class="box">3</div>
  <div class="box">4</div>
  <div class="box">5</div>
  <div class="box">6</div>
  <div class="box">7</div>
</div>
```

inline 요소

- 줄바꿈이 되지 않고, 콘텐츠 만큼의 자리를 차지한다.
- width, height 설정할 수 없다.



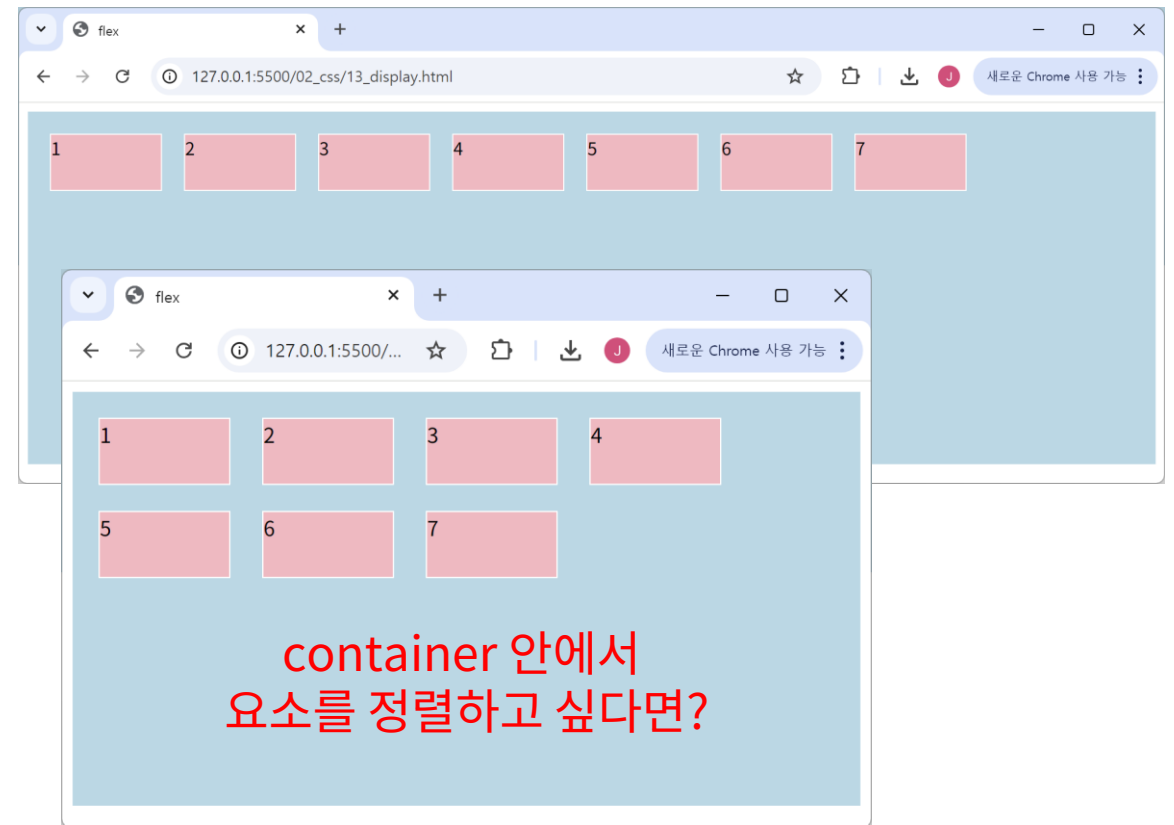
display:inline-block;

```
<style>
  #container {
    background-color: lightblue;
    padding: 10px;
  }
  .box {
    background-color: lightpink;
    border: 1px solid white;
    width: 100px;
    height: 50px;
    margin: 10px;
    display: inline-block;
  }
</style>
```

```
<div id="container">
  <div class="box">1</div>
  <div class="box">2</div>
  <div class="box">3</div>
  <div class="box">4</div>
  <div class="box">5</div>
  <div class="box">6</div>
  <div class="box">7</div>
</div>
```

Inline-block 요소

- 블록레벨 요소의 특성을 가지지만, 줄바꿈 되지 않는다.
- 화면 크기에 따라 요소가 재배치 된다.



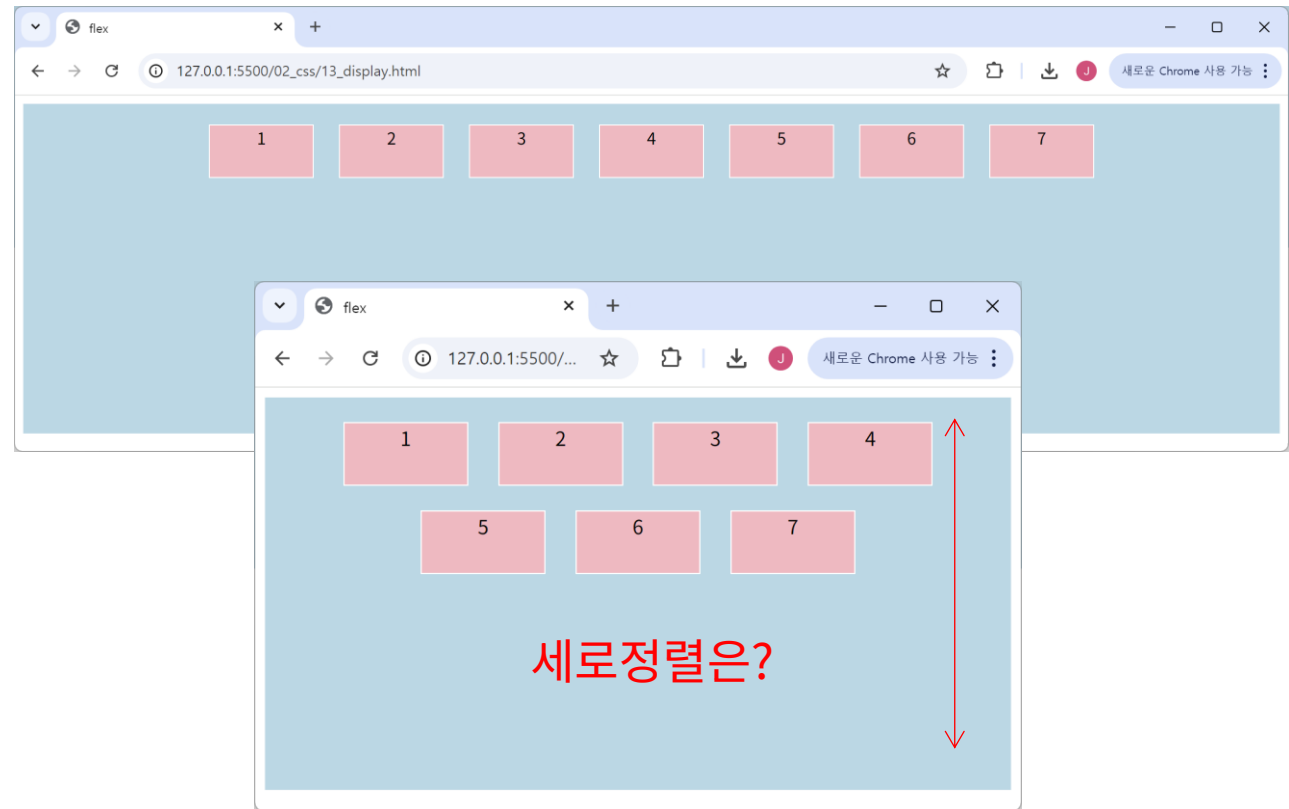
display:inline-block;

```
<style>
  #container {
    background-color: lightblue;
    padding: 10px;
    text-align: center;
  }
  .box {
    background-color: lightpink;
    border: 1px solid white;
    width: 100px;
    height: 50px;
    margin: 10px;
    display: inline-block;
  }
</style>
```

```
<div id="container">
  <div class="box">1</div>
  <div class="box">2</div>
  <div class="box">3</div>
  <div class="box">4</div>
  <div class="box">5</div>
  <div class="box">6</div>
  <div class="box">7</div>
</div>
```

Inline-block 요소

- 블록레벨 요소의 특성을 가지지만, 줄바꿈 되지 않는다.
- 화면 크기에 따라 요소가 재배치 된다.



display: flex;

display: flex; 속성을 지정하면,
해당 요소는 **flex** 컨테이너가 되며 그 직계자식 요소들은 **flex** 아이템이 된다.

- 플렉스 아이템들은 컨테이너 내에서 자동으로 크기가 조정된다.
- 주요 속성
 - ✓ **justify-content** : 주 축(main axis)에 따라 아이템들을 정렬
 - ✓ **align-items** : 교차 축(cross axis)에 따라 아이템들을 정렬
 - ✓ **flex-wrap**: 아이템들이 컨테이너를 넘어갈 때 줄 바꿈 여부 결정

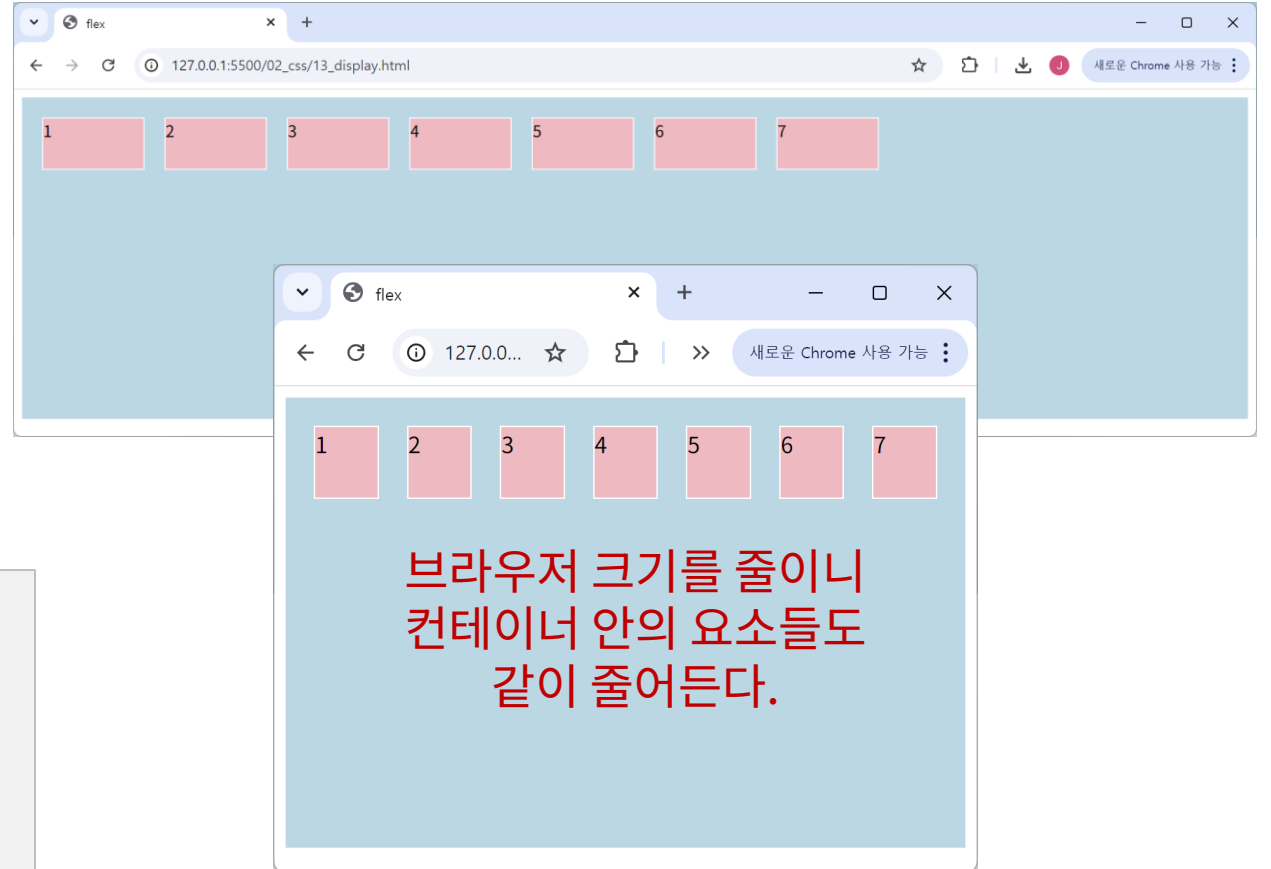
display: flex;

```
<style>
  #container {
    background-color: lightblue;
    padding: 10px;
    display: flex;
  }
  .box {
    background-color: lightpink;
    border: 1px solid white;
    width: 100px;
    height: 50px;
    margin: 10px;
  }
</style>
```

flex 컨테이너

flex 아이tem

```
<div id="container">
  <div class="box">1</div>
  <div class="box">2</div>
  <div class="box">3</div>
  <div class="box">4</div>
  <div class="box">5</div>
  <div class="box">6</div>
  <div class="box">7</div>
</div>
```

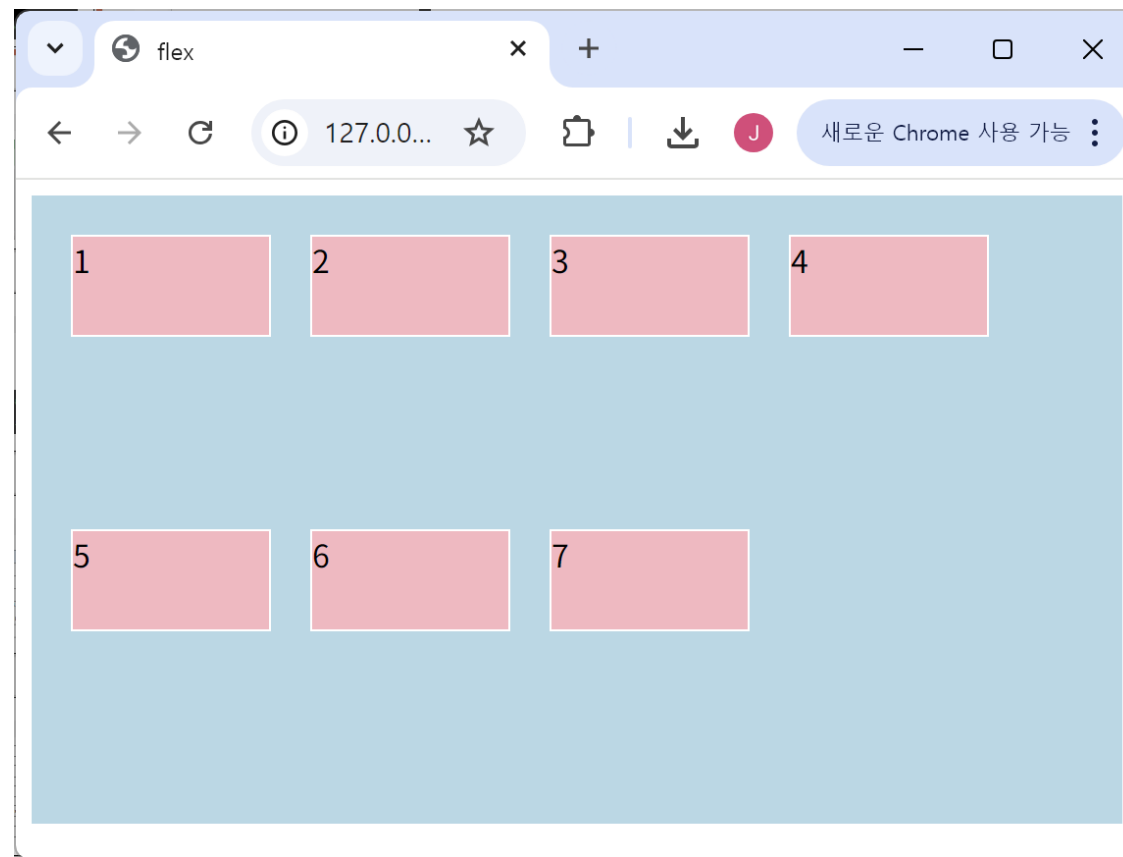


플렉스 아이tem들은
컨테이너 내에서 자동으로 크기가 조정된다.

display: flex;

```
<style>
  #container {
    background-color: lightblue;
    padding: 10px;
    display: flex;
    flex-wrap: wrap; /*자식요소 줄바꿈*/
  }
  .box {
    background-color: lightpink;
    border: 1px solid white;
    width: 100px;
    height: 50px;
    margin: 10px;
  }
</style>
```

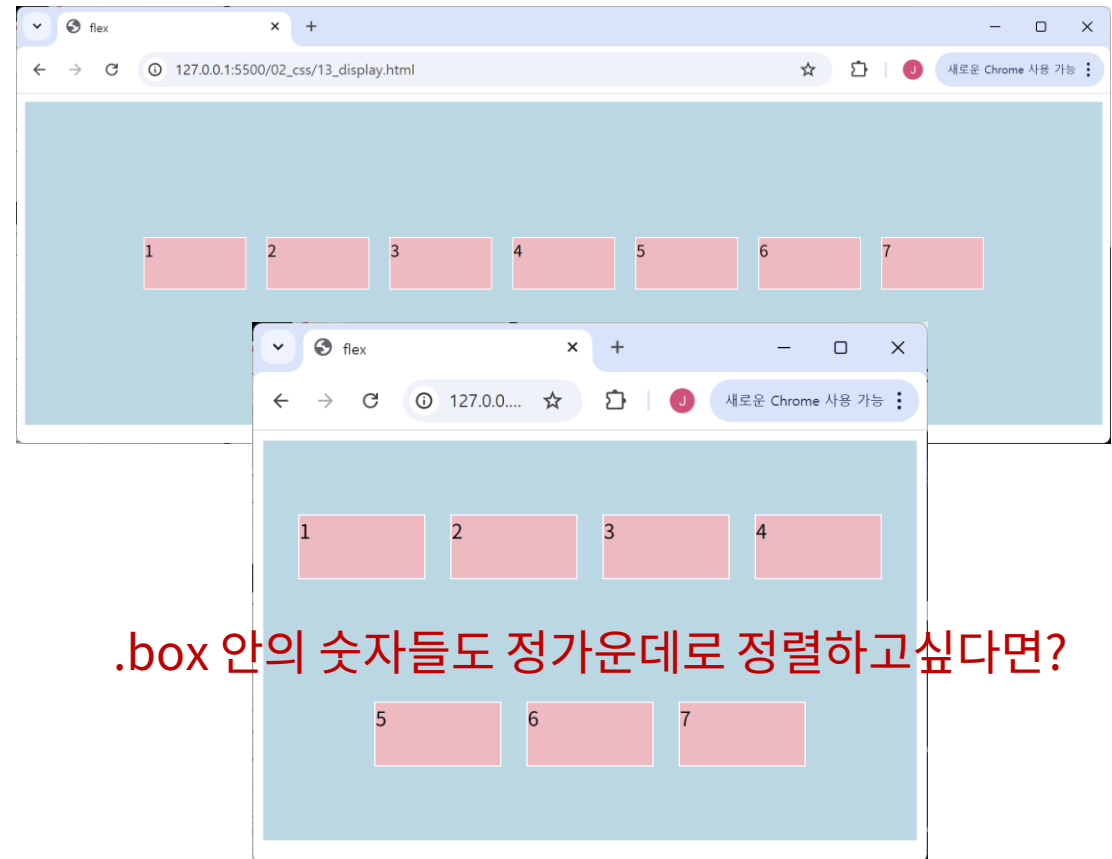
```
<div id="container">
  <div class="box">1</div>
  <div class="box">2</div>
  <div class="box">3</div>
  <div class="box">4</div>
  <div class="box">5</div>
  <div class="box">6</div>
  <div class="box">7</div>
</div>
```



display: flex;

```
<style>
  #container {
    background-color: lightblue;
    padding: 10px;
    display: flex;
    flex-wrap: wrap;
    justify-content: center; /*자식요소 가로정렬*/
    align-items: center; /*자식요소 세로정렬*/
  }
  .box {
    background-color: lightpink;
    border: 1px solid white;
    width: 100px;
    height: 50px;
    margin: 10px;
  }
</style>
```

```
<div id="container">
  <div class="box">1</div>
  <div class="box">2</div>
  <div class="box">3</div>
  <div class="box">4</div>
  <div class="box">5</div>
  <div class="box">6</div>
  <div class="box">7</div>
</div>
```

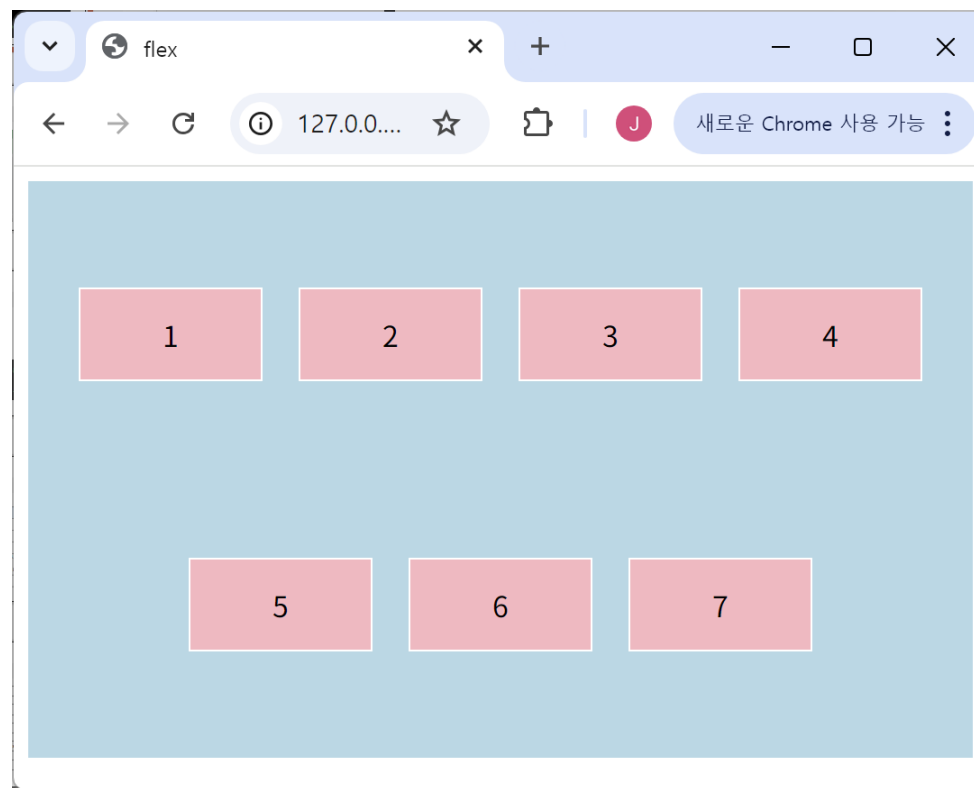


.box 안의 숫자들도 정가운데로 정렬하고싶다면?

display: flex;

```
<style>
  #container {
    background-color: lightblue;
    padding: 10px;
    display: flex;
    flex-wrap: wrap;
    justify-content: center; /*가로정렬*/
    align-items: center; /*세로정렬*/
  }
  .box {
    background-color: lightpink;
    border: 1px solid white;
    width: 100px;
    height: 50px;
    margin: 10px;
    display: flex;
    justify-content: center;
    align-items: center;
  }
</style>
```

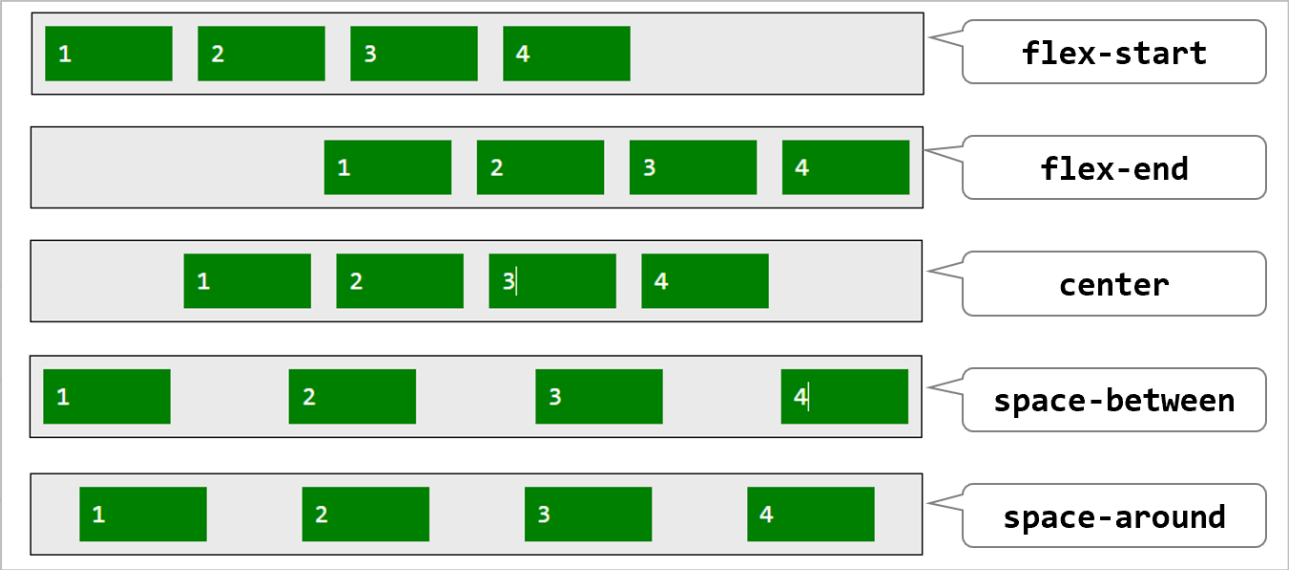
box안의 숫자들을 정 가운데로 정렬



display: flex;

justify-content

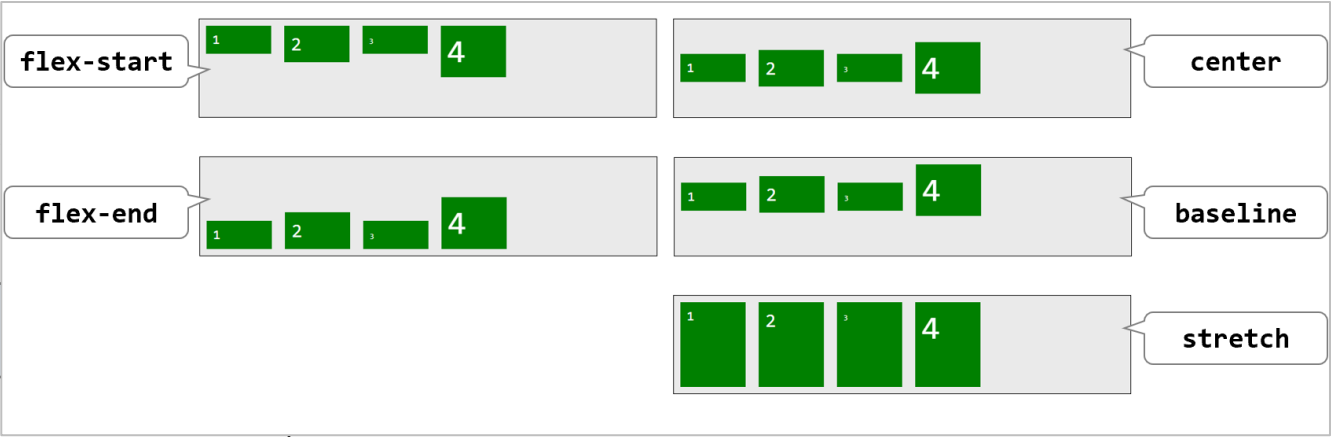
종류	설명
flex-start	시작점에 맞추어 배치
flex-end	끝점에 맞추어 배치
center	중앙에 맞추어 배치
space-between	첫번째 항목의 시작, 끝 항목을 끝점에 배치 나머지 항목은 그 사이에 같은 간격으로 배치
space-around	모든 항목을 같은 간격으로 배치



display: flex;

align-items

종류	설명
flex-start	시작점에 맞추어 배치
flex-end	끝점에 맞추어 배치
Center	중앙에 맞추어 배치
baseline	교차축의 문자 기준에 맞추어 배치
stretch	항목을 늘려 가득 차게 배치



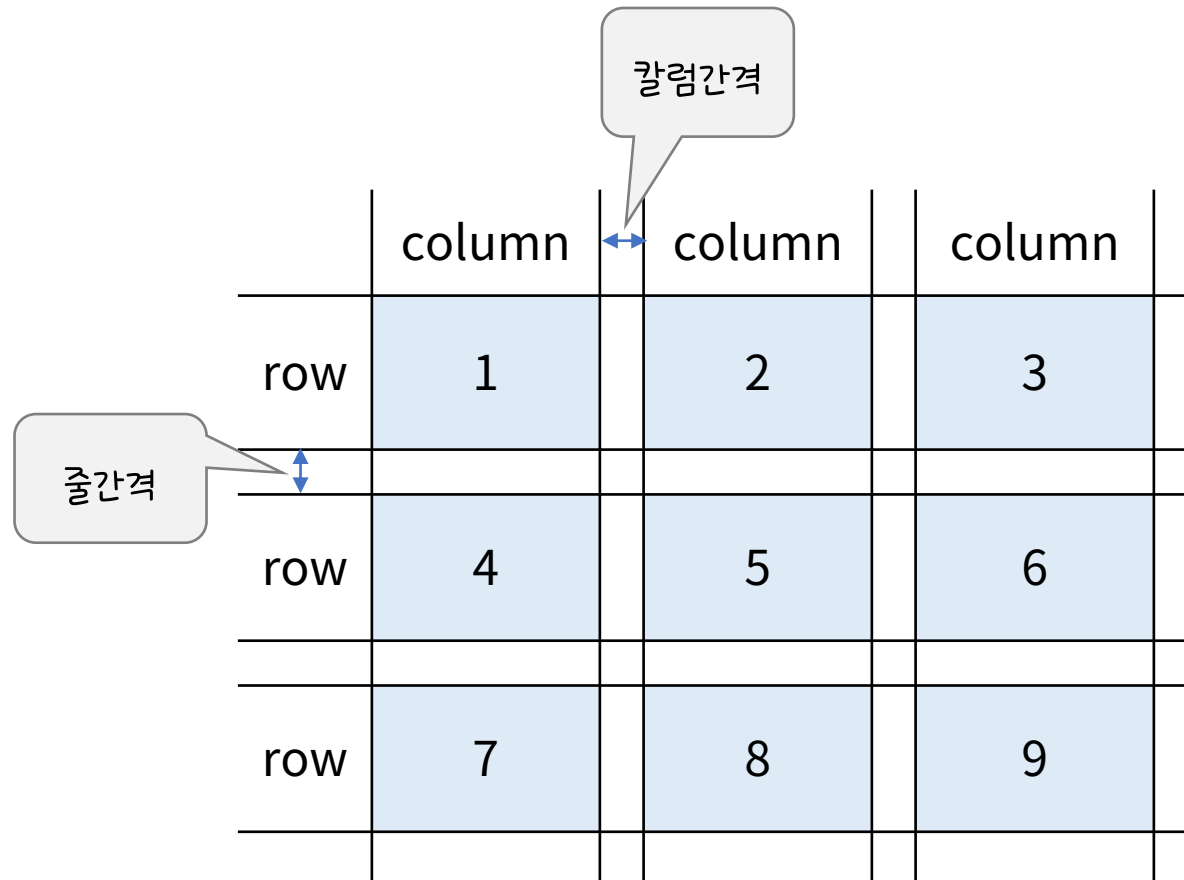
미션

- 배경 이미지가 화면에 가득 차도록 한다.
- 창의 크기에 따라 글씨의 크기가 바뀐다.
- 글씨는 항상 정 가운데에 위치한다.



```
display:grid;
```

row와 column으로 웹 화면을 구성하는 방법



display:grid;

■ grid layout 항목 배치

관련 속성	설명
grid-template-rows	줄 지정 (단위 : px, fr)
grid-template-column	칼럼 지정 (단위 : px, fr)

fr:상대적 크기

display: grid;

■ grid layout 항목 배치

```
<style>
  #container {
    background-color: lightblue;
    padding: 10px;
    display: grid;
    grid-template-columns: 300px 300px 300px;
    grid-template-rows: 100px 100px 100px;
  }
  .box {
    background-color: lightpink;
    border: 1px solid white;
    display: flex;
    justify-content: center;
    align-items: center;
  }
</style>
```

```
<div id="container">
  <div class="box">1</div>
  <div class="box">2</div>
  <div class="box">3</div>
  <div class="box">4</div>
  <div class="box">5</div>
  <div class="box">6</div>
  <div class="box">7</div>
</div>
```



display: grid;

■ grid layout 항목 배치

```
<style>
  #container {
    background-color: lightblue;
    padding: 10px;
    display: grid;
    grid-template-columns: repeat(3, 300px);
    grid-template-rows: repeat(3, 100px);
  }
  .box {
    background-color: lightpink;
    border: 1px solid white;
    display: flex;
    justify-content: center;
    align-items: center;
  }
</style>
```

```
<div id="container">
  <div class="box">1</div>
  <div class="box">2</div>
  <div class="box">3</div>
  <div class="box">4</div>
  <div class="box">5</div>
  <div class="box">6</div>
  <div class="box">7</div>
</div>
```



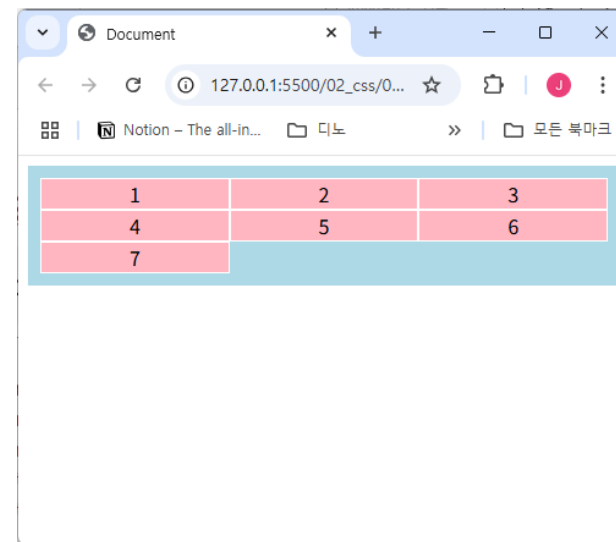
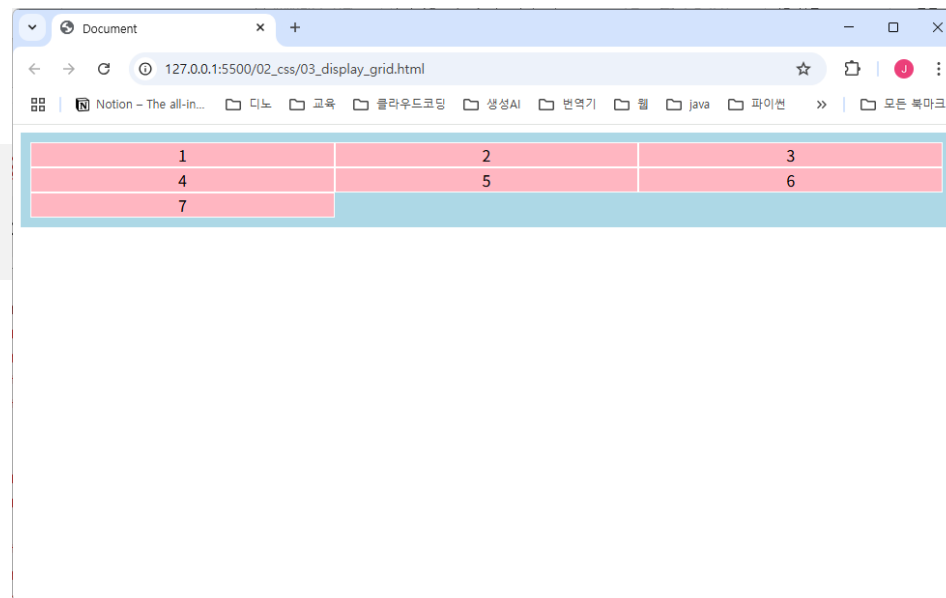
display: grid;

■ grid layout 항목 배치

```
<style>
  #container {
    background-color: lightblue;
    padding: 10px;
    grid-template-columns: repeat(3, 1fr);
    grid-template-rows: repeat(3, 1fr);
  }
  .box {
    background-color: lightpink;
    border: 1px solid white;
    display: flex;
    justify-content: center;
    align-items: center;
  }
</style>
```

비율

```
<div id="container">
  <div class="box">1</div>
  <div class="box">2</div>
  <div class="box">3</div>
  <div class="box">4</div>
  <div class="box">5</div>
  <div class="box">6</div>
  <div class="box">7</div>
</div>
```



display: grid;

■ 항목 관련 지정

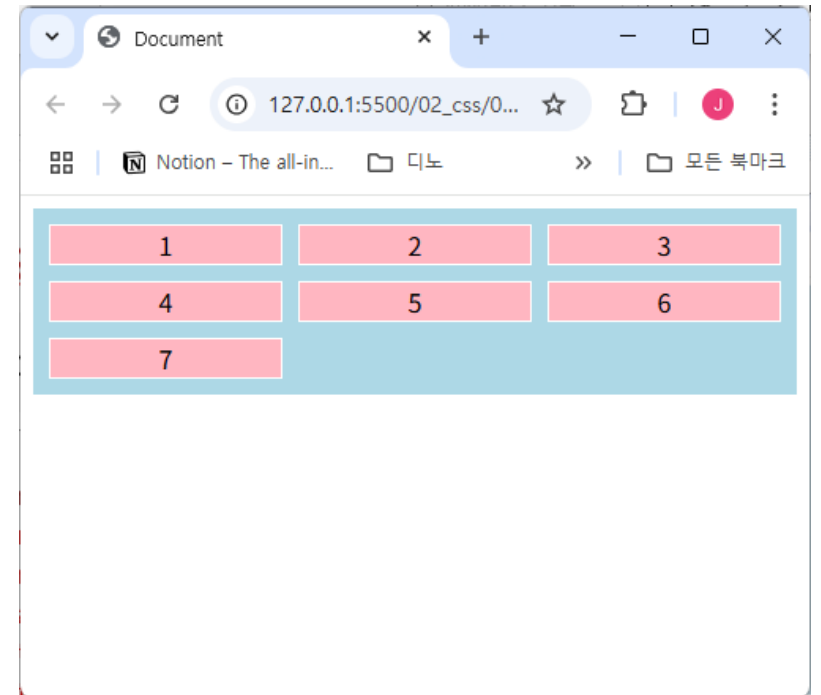
관련 속성	설명
grid-column-gap	칼럼 사이 간격 지정
grid-row-gap	줄 사이 간격 지정
grid-gap	칼럼, 줄 사이 간격 한번에 지정

display: grid;

■ grid layout 항목 배치

```
<style>
  #container {
    background-color: lightblue;
    padding: 10px;
    display: grid;
    grid-template-columns: repeat(3, 1fr);
    grid-template-rows: repeat(3, 1fr);
    grid-gap: 10px;
  }
  .box {
    background-color: lightpink;
    border: 1px solid white;
    display: flex;
    justify-content: center;
    align-items: center;
  }
</style>
```

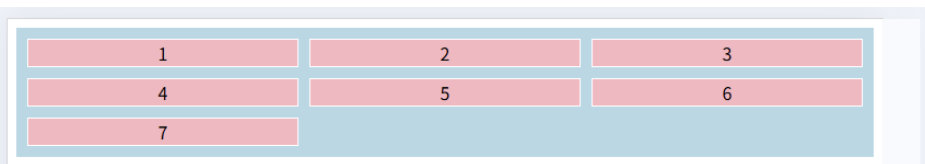
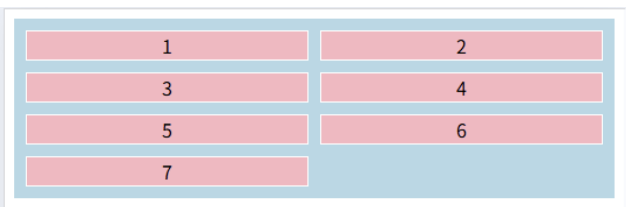
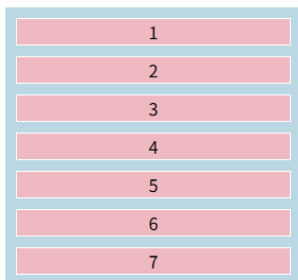
```
<div id="container">
  <div class="box">1</div>
  <div class="box">2</div>
  <div class="box">3</div>
  <div class="box">4</div>
  <div class="box">5</div>
  <div class="box">6</div>
  <div class="box">7</div>
</div>
```



display: grid;

■ 반응형 웹 적용

모바일(480px기준) → 컬럼수 1개
태블릿(768px기준) → 컬럼수 2개
pc(768px기준) → 컬럼수 3개



※ 본 강의 자료는 본인 학습용으로만 사용가능하며 외부 유출을 금합니다.

```
<style>
  #container {
    background-color: lightblue;
    padding: 10px;
    display: grid;
    grid-template-columns: 1fr;
    grid-gap: 10px;
  }
  .box {
    background-color: lightpink;
    border: 1px solid white;
    display: flex;
    justify-content: center;
    align-items: center;
  }
</style>
```

```
@media screen and (min-width: 480px) {
  #container {
    grid-template-columns: 1fr;
  }
}
```

```
@media screen and (min-width: 768px) {
  #container {
    grid-template-columns: 1fr 1fr;
  }
}
```

```
</style>
```

미션

반응형 웹과 그리드 레이아웃을 이용한 반응형 앱 만들기

