HTML/CSS

웹 프로그래밍에서 기본적으로 갖춰야 할 HTML/CSS 관련 지식을 공부합시다

HTML 기본

HTML Basic

웹 페이지의 구조를 다루는 HTML에 대해 배웁니다

HTML은?

Hyper Text Markup Language

마크업 언어

태그 등을 이용하여 문서나 데이터의 구조를 명기하는 언어

“하이퍼 텍스트”

링크를 이용해 웹 페이지를 서로 연결하는 것을 의미

마크업을 이용해 월드 와이드 웹(WWW)의 페이지를 서로 오갈 수 있는 웹 문서를 만드는 언어

웹 표준이란?

W3C(World Wide Web Consortium) Standards

W3C

World Wide Web Consortium의 약자로, 웹 표준과 가이드라인 개발을 목적으로 설립

https://ko.wikipedia.org/wiki/W3C

웹 표준

WWW (World Wide Web)을 서술하고 정의하는 공식 표준 및 규격

https://ko.wikipedia.org/wiki/웹_표준

서로 다른 브라우저/환경에서도 같은 결과를 보여줄 수 있도록
웹 사이트를 만들 때 지켜야 하는 규격을 정해놓은 것

웹 표준 지원정도

W3C(World Wide Web Consortium) Standards

http://html5test.com

HTML의 기본구조

HTML Structure

doctype은 문서 유형을 지정하며,
html은 HTML5 형식임을 의미합니다

```
<!DOCTYPE html>
<html>
<head>
  <title>Document</title>
</head>
<body>

</body>
</html>
```

HTML문서의 시작과 끝을 의미합니다

HTML문서에 관한 기본정보를 포함합니다

브라우저의 제목표시줄에 출력될 내용입니다

문서의 본문에 해당합니다

<와 >로 이루어진 요소를 '태그'라 부릅니다

주석

Comment

```
<!DOCTYPE html>
<html>
<head>
  <title>Document</title>
</head>
<body>
<!-- 이 사이에 넣는 내용은 주석이 됩니다 -->
</body>
</html>
```

자주, 습관적으로 사용하는 것이 좋습니다!

주석은 화면에는 표시되지 않으며, 소스에서만 확인할 수 있습니다.
개발기간이 길어지고, 소스가 많아질수록 작업한 내용을 주석으로 정리해야 나중에 소스를 고치거나 참고할 때 효율성이 증가합니다.

태그의 요소와 속성

Element & Attribute

HTML 태그의 구성

```
<요소 속성="값">또는
<요소 속성="값">내용</요소>
```

스스로 닫는 태그 (담을 내용이 없음)

<열리는태그>와 </닫히는태그>로 이루어진 태그

예제

```
<a href="http://naver.com">네이버 홈으로 가기</a>
<img src="https://www.google.co.kr/images/branding/googlelogo/1x/
googlelogo_color_272x92dp.png">
```

href는 Hyper Reference(참조), src는 Source(출처, 소스)의 줄임말입니다.

태그는 대, 소문자 모두 사용 가능하지만, 가급적이면 소문자를 사용하도록 합니다 (HTML5표준 명세에서 권장)

절대경로와 상대경로

Absolute path & Relative path

절대경로

```
<img src="https://www.google.co.kr/images/branding/googlelogo/1x/
googlelogo_color_272x92dp.png">
```

http://, https://로 시작하는 전체 주소입니다

상대경로

```
<img src="images/sample.png">
```

해당 HTML파일을 기준으로 한 경로입니다

상대경로에서, 하위 폴더로는 "폴더명/파일명"을 사용하며,
상위 폴더로 올라갈때는 "../"으로 시작합니다.
"/"로 시작한다면 가장 최상위 폴더에서부터 시작합니다.

head 태그

문서의 메타데이터 집합

웹 페이지에 직접적으로 보이지 않는 정보를 브라우저에게 제공

```
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <link rel="stylesheet" href="style.css">
  <script src="script.js" charset="utf-8"></script>
  <title>Document</title>
</head>
```

웹 페이지의 인코딩 방식을 정의합니다

IE에서의 렌더링 방식을 최신으로 설정합니다

CSS파일을 연결합니다

JavaScript파일을 연결합니다

문서의 제목을 나타냅니다

여기있는 내용에 더해, 다양한 meta태그가 존재할 수 있습니다.
페이스북에 링크했을때의 제목, 설명, 커버이미지를 보여주는 meta태그나, 검색엔진에서 주로 사용할 내용,
모바일에서의 확대/축소여부 등 여러가지 설정값을 지정할 수 있습니다.
또한, CSS나 JS파일의 링크 역시 head요소 내부에 link, style, script요소를 사용해 나타냅니다.

body 태그

브라우저에 표시될 내용

이곳에 기록한 내용은 사용자에게 보여집니다

```
<body>
  <h1>패스트캠퍼스 웹 프로그래밍 스쿨</h1>
  <p>HTML을 배웁니다</p>
  <blockquote>처음으로 작성한 HTML문서입니다</blockquote>
</body>
```

Atom 편집기

https://atom.io

Atom Packages

패키지 설치법

1. 설정 진입
Cmd + , or Ctrl + ,

2. Install 탭 선택
3. 검색, Install버튼 클릭

설치할 패키지

Emmet
HTML/CSS코딩 생산성을 향상시켜주는 도구

less-than-slash
자동으로 HTML태그를 닫아주는 기능

블록과 인라인

Block, Inline

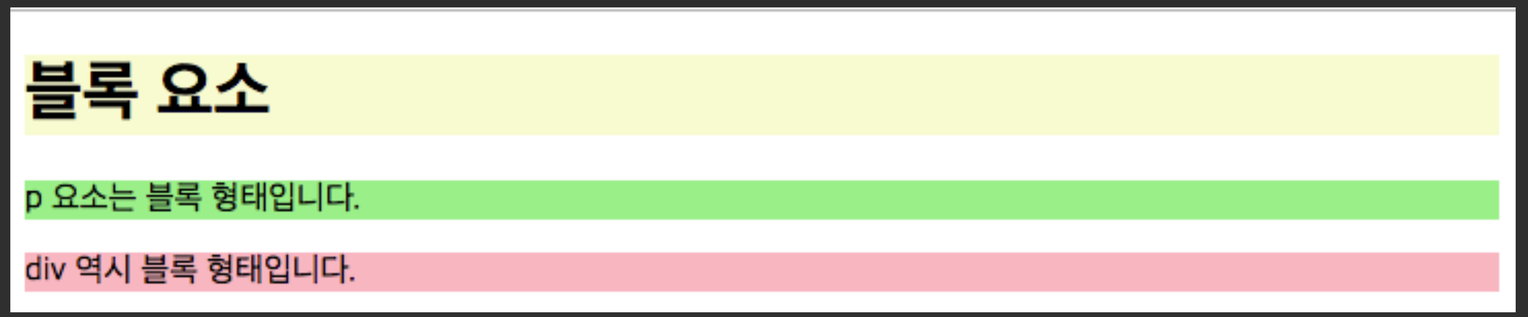
요소의 형태는 두 가지로 나누어집니다

블록 요소

Block Level Elements

줄 바꿈이 일어나는 형태

```
<h1>블록 요소</h1>
<p>p요소는 블록 형태입니다</p>
<div>div요소도 블록 형태입니다</div>
```



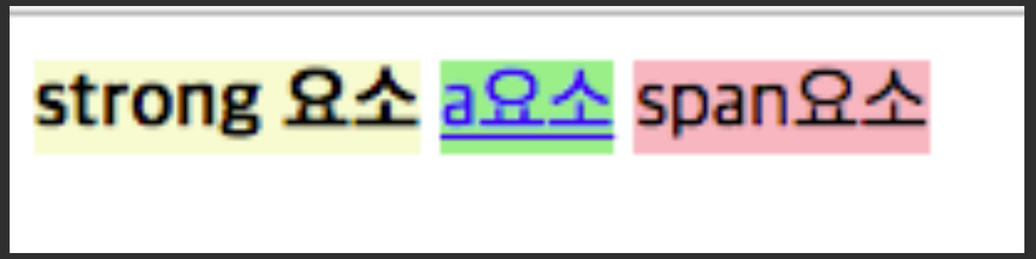
블록 요소는 줄 바꿈이 일어나는 형태이며, 기본적으로 width가 전체 너비의 값을 가집니다.
해당 요소에 배경색(background-color)을 지정하게 되면, 실제로 들어있는 글자 내용과는 별개로 전체 너비 영역만큼을 차지한다는 것을 알 수 있습니다.

인라인 요소

Inline Elements

줄 바꿈이 일어나지 않는 형태

```
<strong>strong 요소</strong>
<a href="">a요소</a>
<span>span요소</span>
```



인라인 요소는 줄바꿈 없이, 기본적으로는 자신의 내용만큼의 가로너비를 가집니다.
블록 요소는 인라인 요소를 포함할 수 있지만, 인라인 요소는 블록 요소를 포함할 수 없습니다.

레이아웃 요소

div, span

```
<div>
  <p>블럭요소 내부에 <span>인라인 요소를 사용합니다</span></p>
</div>
```

div와 span은 오직 Block과 Inline방식의 레이아웃을 구현하는데에 사용합니다.

HTML 태그

Text Tags

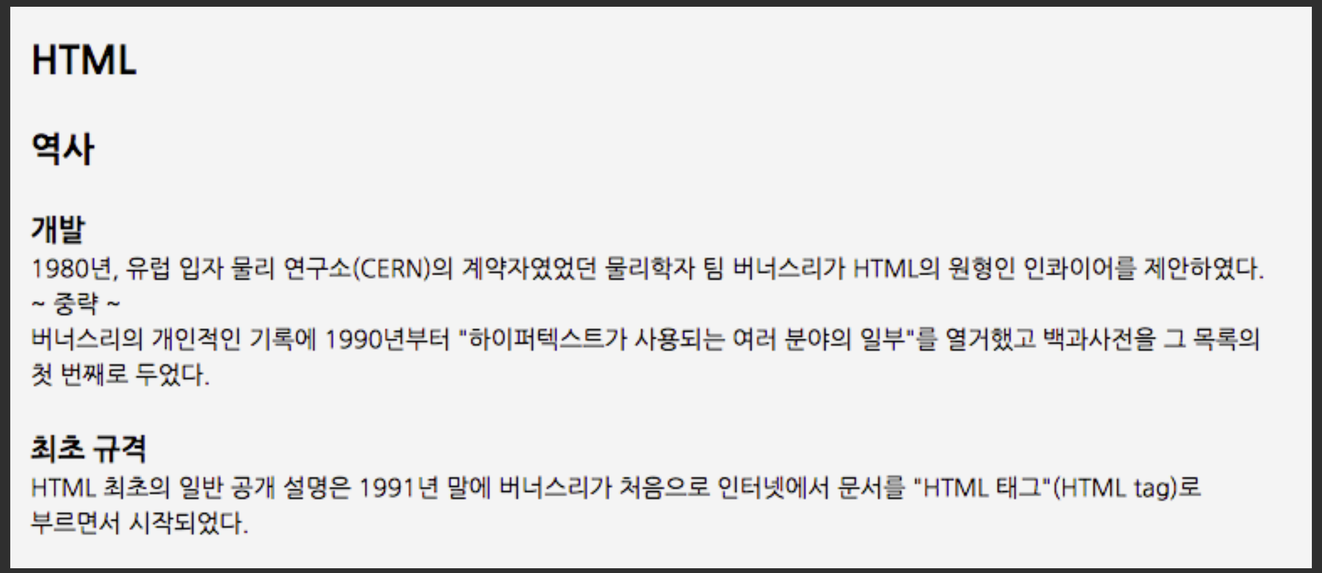
텍스트와 관련된 태그

헤드라인

Headline

웹 페이지의 개요를 나타냄

h1 ~ h6



HTML 코드

```
<h1>HTML</h1>
<h2>역사</h2>
<h3>개발</h3>
<h3>최초규격</h3>
```

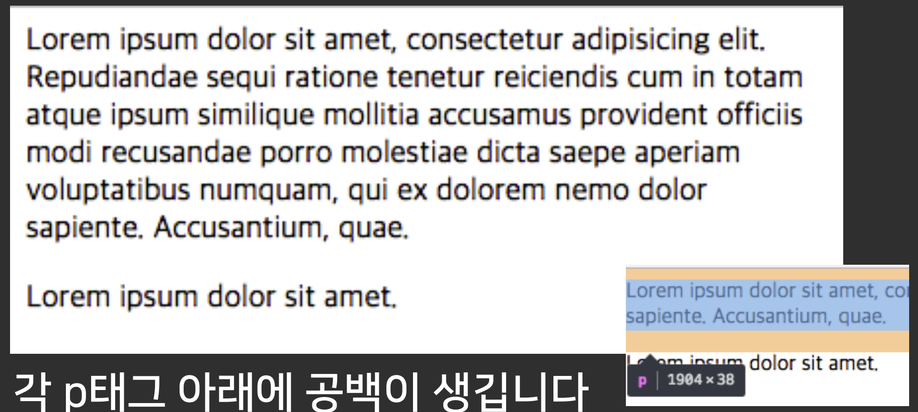
중요도 순으로 개요를 나타낼 때 사용합니다.
학술문서나 검색엔진에서 검색시에 중요하게 사용됩니다.
실제 글자크기등은 CSS에서 만들고자 하는 웹 페이지에 맞춰서 새로 설정하므로, 단계별로 구분할 제목이 있다면 hn태그를 사용하는 것이 좋습니다.

줄 바꾸기

Line breaks

p태그 (Paragraph, 문단)

```
<p>Lorem ipsum dolor sit amet, consectetur
adipisicing elit. Repudiandae sequi ratione
tenetur reiciendis cum in totam atque ipsum
similique quae.</p>
<p>Lorem ipsum dolor sit amet.</p>
```



각 p태그 아래에 공백이 생깁니다

br태그 (Linebreak, 줄 바꾸기)

```
Lorem ipsum dolor sit amet.<br>
Lorem ipsum dolor sit amet, consectetur
adipisicing elit. Tenetur, ab.<br>
```

줄바꿈 간에 여백이 없습니다

그 외

hr태그 (Horizontal rule, 수평선)

```
<hr>
```

blockquote태그 (인용문)

```
<blockquote>인용문 내용</blockquote>
```

pre태그 (Preformatted text, 이미 형식화 된 텍스트)

```
<pre>
def pretag_test():
    val = ‘pretag’
</pre>
```

줄바꿈 없는 텍스트 태그

strong, b태그 (강조, 굵게 표시)

```
<strong>강조할 텍스트</strong>
<b>굵게 표시할 텍스트</b>
```

em, i태그 (문맥상 특정부분 강조, 이탤릭 표시)

```
<em>강조할 텍스트</em>
<i>기울여 표시할 텍스트</i>
```

mark태그 (형광펜 효과)

```
<mark>형광펜으로 그은 효과의 텍스트</mark>
```

HTML 태그

Image, Hyperlink Tags

이미지와 링크 태그

링크

Anchor

a

```
<a href="http://www.naver.com" target="_blank" title="네이버 열기">sdf</a>
```

새 창에서 링크를 열어줍니다

href : 이동할 페이지 주소

target : 링크 걸린 페이지를 여는 방법 (_self, _blank)

title : 마우스를 올렸을 때 보여줄 제목

링크를 클릭한 창의 주소를 바꿉니다

**target에는 추가로 parent, top, <framename>이 있지만, 최근 HTML에서는 Frame레이아웃을 잘 사용하지 않기때문에 설명에서 제외했습니다

이미지

Image

img

```
<img src=“이미지 경로" width="100" height=“200" alt=“이미지 설명">
```

src : 이미지의 경로
width, height : 이미지의 가로/세로 크기(px단위)
alt: 대체 텍스트 (alternative text)

HTML 태그

Data tags

데이터를 나타내는 태그

목록

Ordered List, Unordered List

Ordered List

```
<ol>
  <li>항목</li>
  <li>항목</li>
  <li>항목</li>
  <li>항목</li>
  <li>항목</li>
</ol>
```

Unordered List

```
<ul>
  <li>항목</li>
  <li>항목</li>
  <li>항목</li>
  <li>항목</li>
  <li>항목</li>
</ul>
```

목록 형태로 나타나는 요소들은 ol, ul태그를 사용하여 구현 후, CSS로 제작하고자 하는 디자인에 맞게 스타일을 지정해줍니다.

목록 속성

type, start, reversed

```
<ol type="A" start="3" reversed>
  <li>First</li>
  <li>Second</li>
  <li>Third</li>
</ol>
```

type

값	설명
1	숫자(기본값)
a	영문 소문자
A	영문 대문자
i	로마숫자 소문자
I	로마숫자 대문자

start

시작할 숫자 지정

reversed

역순으로 표시

정의 목록

Description List

정의 목록 태그

dt는 목록 중 개념을 나타냅니다

dd는 해당 개념의 정의를 나타냅니다

```
<dl>
    <dt>HTML</dt>
    <dd>Hyper Text Markup Language</dd>
    <dd>웹 페이지를 구현하는 마크업 언어이다</dd>

    <dt>CSS</dt>
    <dd>Cascading Style Sheet</dd>
    <dd>HTML의 형태를 지정하는 언어이다</dd>
</dl>
```

목록과 정의 목록은 서로 중첩해서 사용이 가능합니다.

테이블 요소

table

표를 만들 수 있는 table요소를 알아봅시다

테이블의 기본 구조

table basic

테이블의 시작

행을 나타냄

th는 테이블의 헤더 셀

td는 일반 셀

```
<table>
  <tr>
    <th>이름</th>
    <th>나이</th>
    <th>점수</th>
  </tr>
  <tr>
    <td>철수</td>
    <td>23세</td>
    <td>70점</td>
  </tr>
  <tr>
    <td>영희</td>
    <td>21세</td>
    <td>89점</td>
  </tr>
</table>
```

두 학생의 성적 비교

이름	나이	점수
철수	23세	70점
영희	21세	89점

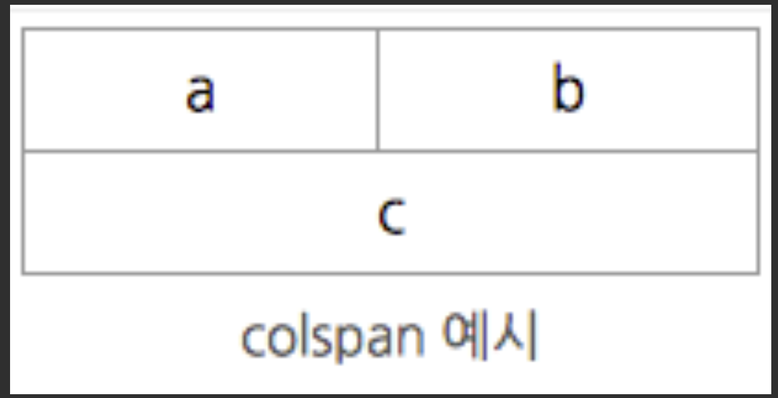
기본적인 도표 예시

셀 병합

colspan

```
<table>
  <tr>
    <td>a</td>
    <td>b</td>
  </tr>
  <tr>
    <td colspan="2">c</td>
  </tr>
</table>
```

병합된 셀 수 만큼 행(tr)안에 셀을 덜 적어 줌



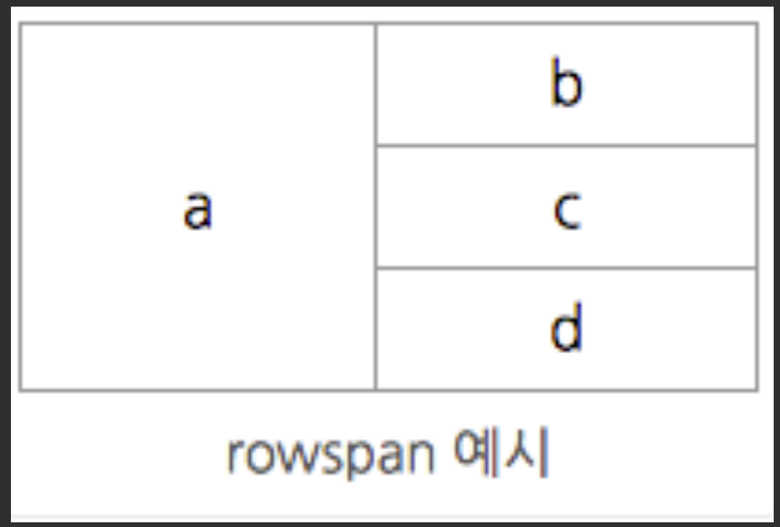
colspan은 가로로 셀을 합칩니다.
즉, 열(col)들을 병합하는 속성을 뜻합니다.
병합은 th나 td에 사용할 수 있습니다.

셀 병합

rowspan

```
<table>
  <tr>
    <td rowspan="3">a</td>
    <td>b</td>
  </tr>
  <tr>
    <td>c</td>
  </tr>
  <tr>
    <td>d</td>
  </tr>
</table>
```

병합된 셀 수 만큼 다음 행(tr)의 셀(td)개수가 줄어듬



rowspan 예시

rowspan은 세로로 셀을 합칩니다.

세로로 합쳐진 셀의 수에 해당하는 다음 행(tr)의 셀 개수는 하나씩 줄어들게 됩니다.

colgroup

하나 이상의 열을 그룹지음

HTML

```
<table>
  <caption></caption>
  <colgroup>
    <col />
    <col />
    <col />
  </colgroup>
  <!-- 또는 -->
  <colgroup span="3"> </colgroup>
</table>
```

CSS

```
table > colgroup {
  background: #f3f3f3;
  border-right: 3px double #333;
}
```

참가자별 점수표

이름	나이	성별	100M 달리기	윗몸 일으키기
홍길동	22세	남	15.25	29
황진이	20세	여	16.12	41
참가자 평균			15.7	35

colgroup의 사용 예시

colgroup을 사용하면, 특정 '열' 또는, 특정 열의 '그룹'에 쉽게 속성을 줄 수 있습니다.

행의 구조화

thead

```
<table>
  <thead>
    <tr>
      <th>이름</th>
      <th>나이</th>
      <th>성별</th>
      <th>성적</th>
      <th>메모</th>
    </tr>
  </thead>
</table>
```

colgroup이 열을 그룹화한다면, thead, tbody, tfoot은 행을 그룹화합니다.
이중 thead는 열의 제목을 나타냅니다.

행의 구조화

tbody

tbody는 본문에 해당하는 영역입니다.
thead와 tfoot은 table에서 한 번만 선언될 수 있으나,
tbody는 여러번 선언되어 행을 그룹화 할 수 있습니다.

```
<table>
  <tbody>
    <tr>
      <th>홍길동</th>
      <th>22세</th>
      <th>남자</th>
      <th>98점</th>
      <th>잘생김</th>
    </tr>
  </tbody>
  <tbody>
    <tr>
      <th>김춘향</th>
      <th>19세</th>
      <th>여자</th>
      <th>99점</th>
      <th>예쁨</th>
    </tr>
  </tbody>
</table>
```

행의 구조화

tfoot

```
<table>
  <tfoot>
    <tr>
      <td colspan="3">평균</td>
      <td>87</td>
      <td></td>
    </tr>
  </tfoot>
</table>
```

도표 하단을 나타냅니다.
일반적으로 도표 전체의 합계나 결과를 표시하는 경우가 많습니다.

Form 요소

form

데이터를 입력하거나 전송할 때 사용하는 HTML요소에 대해 알아봅니다

form 요소

form

form태그

```
<form action="" method="get">
  <label for="username">ID</label>
  <input type="text" name="username">
</form>
```

form은 브라우저(클라이언트)에서 서버로 데이터를 전송하기 위해 사용하는 태그입니다.

form - method

method 속성

method

폼에서 서버로 데이터를 전송하는 방식을 결정

GET

URL에 데이터를 담아 전달

POST

URL과는 별도로 데이터를 전달

아이디/패스워드와 같은 중요한 정보는 GET방식으로 전달하지 않습니다.

form - action

action 속성

action

form에서 데이터를 전송할 URL

input 태그

input types

input의 type들

```
<input type="text" id="username">
<input type="password" id="password">
<input type="radio" id="radio">
<input type="checkbox" id="checkbox">
<input type="button">
<input type="file" id="file">
<input type="submit">
<input type="reset">
<input type="hidden" id="hidden" value="hiddenValue">
```

데이터 전송
폼 초기화

input요소는 값을 가지거나, form에 영향을 줍니다

input 태그

input 주요 속성들

input의 속성들

```
<input type="text" value="disabled" disabled>
<input type="text" value="readonly" readonly>
<input type="text" required>
<input type=“text" placeholder=“공백은 안됩니다">
<input type="text" size="3">
<input type="text" maxlength="10">
<input type="checkbox" checked="checked">
<input id="radio1" type="radio" name="agree" checked="checked">
<input id="radio2" type="radio" name="agree">
```

같은 이름을 가진 radio요소는 동시에 체크되지 않습니다

input요소는 값을 가지거나, form에 영향을 줍니다

label 태그

폼 요소에 레이블을 붙임

label내부에 표현

label태그 내에 폼 요소를 삽입

```
<label>ID <input type="text"></label>
```

label과 별도로 표현

label태그와 별도로

```
<label for="username">Username</label>
<input type="text" id="username">
```

for속성과 form요소의 id를 연결시키면 label이 정확히 해당 form요소를 가리키게 됩니다

select 태그

select

select 태그

```
<select name="number" id="select-id">
  <option value="1">First</option>
  <option value="2">Second</option>
  <option value="3">Third</option>
  <option value="4">Fourth</option>
</select>
```

select태그는 여러개의 주어진 값 중 일부를 선택하는 역할을 합니다

select 태그

select 태그의 속성

multiple 속성

```
<select multiple="multiple">
  <option value="apple">Apple</option>
  <option value="banana">Banana</option>
  <option value="orange">Orange</option>
</select>
```

기본적으로 select요소는 option의 값을 한 개만 선택할 수 있습니다.
multiple속성을 가질 경우, ctrl (맥의 경우 cmd), shift키를 이용해 여러개의 값을 선택할 수 있습니다.

select 태그

select 요소의 속성

size 속성

```
<select size="2">
  <option value="apple">Apple</option>
  <option value="banana">Banana</option>
  <option value="orange">Orange</option>
</select>
```

한 번에 option을 몇 개 보여줄지 정합니다.

optgroup 태그

select 요소의 option을 그룹지어줌

```
<select>
  <optgroup label="Fruits">
    <option value="apple">Apple</option>
    <option value="banana">Banana</option>
    <option value="orange">Orange</option>
  </optgroup>
  <optgroup label="Colors">
    <option value="red">Red</option>
    <option value="blue">Blue</option>
    <option value="green">Green</option>
  </optgroup>
</select>
```

option을 그룹화하여 보여줍니다

button 태그

button types

button 요소

```
<button type="submit">submit type button</button>
<button type="reset">reset type button</button>
<button type="button">button type button</button>
```

button요소는 input요소의 같은 type을 대체할 수 있습니다.

fieldset, legend 태그

fieldset, legend

fieldset의 제목을 나타냅니다

form요소들을 담는
틀을 나타냅니다

fieldset과 legend 요소

```
<fieldset>
  <legend>Login</legend>
  <label>username : <input type="text"></label>
  <label>password : <input type="password"></label>
</fieldset>
```

legend요소는 fieldset의 첫 번째 자식으로 사용해야합니다.
fieldset은 다른 fieldset을 중첩해서 자식으로 가질 수 있습니다.

클래스와 아이디 속성

class, id

네이밍

첫 글자는 알파벳으로 시작
두 번째부터는 알파벳, 숫자, -, _를 사용 가능
대소문자를 구분

클래스와 아이디의 차이

id(아이디)는 페이지에서 딱 한번반 선언 가능, 요소의 unique한 특성을 나타냄
class(클래스)는 여러번 사용 가능, 범용적인 부분을 나타냄

ID와 Class의 적절한 사용 예

```
<div class="chapter" id="chapter1">
  <h2>HTML</h2>
  <p>HTML강의를  시작해봅시다</p>
</div>

<div class="chapter" id="chapter2">
  <h2>CSS</h2>
  <p>Chapter2는 CSS입니다!</p>
</div>

<div class="chapter" id="chapter3">
  <h2>JavaScript</h2>
  <p>JavaScript는 야무님이죠!</p>
</div>
```

class와 id의 이름은 짧게짓기보다는, 길더라도 해당 요소의 의미에 적합하게 짓는 것이 더 좋습니다.

색상

Color

```
<p style="color: #333;">Color #333</p>
<p style="color: DarkGreen;">Color DarkGreen</p>
```

색상은 Hex code를 사용한 #000000 ~ #FFFFFF까지의 값과,
HTML규격에 미리 정의된 ColorName을 사용할 수 있습니다.

HTML 색상표(2) color name을 이용한 색상표

Color Name	Color HEX	Color
AliceBlue	#F0F8FF	
AntiqueWhite	#FAEBD7	
Aqua	#00FFFF	
Aquamarine	#7FFFD4	
Azure	#F0FFFF	
Beige	#F5F5DC	
Bisque	#FFE4C4	
Black	#000000	
BlanchedAlmond	#FFEBCD	
Blue	#0000FF	
BlueViolet	#8A2BE2	
Brown	#A52A2A	
BurlyWood	#DEB887	
CadetBlue	#5F9EA0	
Chartreuse	#7FFF00	
Chocolate	#D2691E	
Coral	#FF7F50	
CornflowerBlue	#6495ED	
Cornsilk	#FFF8DC	
Crimson	#DC143C	
Cyan	#00FFFF	
DarkBlue	#00008B	
DarkCyan	#008B8B	
DarkGoldenRod	#B8860B	
DarkGray	#A9A9A9	
DarkGrey	#A9A9A9	
DarkGreen	#006400	
DarkKhaki	#BDB76B	
DarkMagenta	#8B008B	
DarkOliveGreen	#556B2F	
Darkorange	#FF8C00	
DarkOrchid	#9932CC	
DarkRed	#8B0000	

http://www.homejjang.com/03/color_name.php
http://www.w3schools.com/colors/colors_names.asp

배운내용을 복습하며 문서를 작성해봅시다

form-signup.html

CSS 기본

CSS Basic

CSS란 무엇이며, 어떻게 사용하는 것인지?

CSS란?

Cascading Style Sheet

마크업 언어(HTML)가 실제 표시되는 방법을 기술하는 언어

레이아웃과 스타일을 정의할 때 사용

HTML과 CSS를 분리해서 사용해야 하는 이유는?

과거의 HTML

혼돈! 파괴! 망~~가~~각!

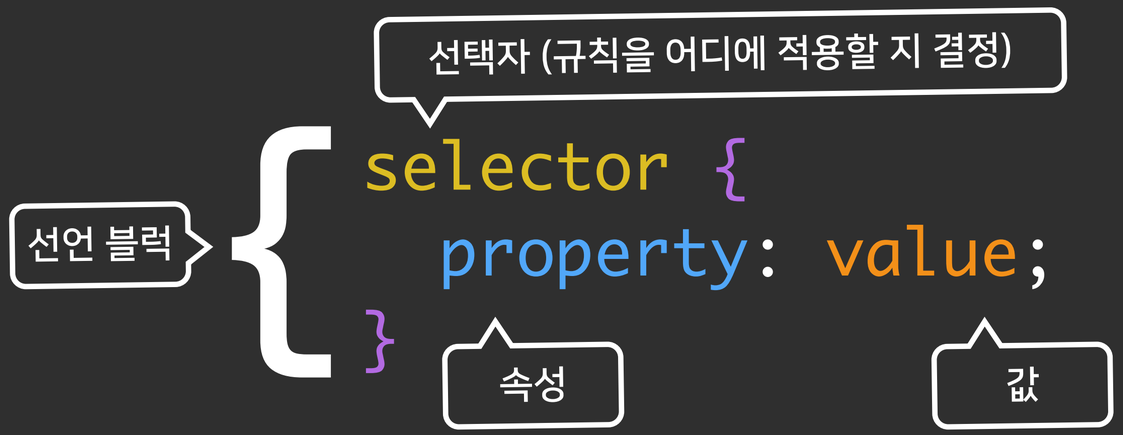
```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html;charset=UTF-8">
  </head>
  <body bgcolor="Azure">
    <font size="4" face="arial" color="DarkBlue">Chaos! Destruction! Oblivion!</font>
    <hr size="1" width="100%" color="DarkCyan">
    <p font-size="20px">CSS가 없는 혼돈의 세계입니다</p>
    <br>
    <hr align="right" width="40%" color="DarkSlateGray">
    <p align="right">여러분은 이렇게 사용하시면 안됩니다</p>
  </body>
</html>
```

HTML에는 스타일을 제외한, 문서의 구조만이 명확히 나타나야 함

과거 브라우저에서 사용하던 font, center등의 태그는 HTML5에서는 더 이상 사용하지 않습니다
http://www.w3schools.com/html/html5_intro.asp

CSS문법

CSS Syntax Basic



CSS문법

CSS Syntax Basic - Example

```
#body-title {
  font-size: 14px;
  font-weight: bold;
  color: DarkSlateGrey;
}
```

HTML Color Names
http://www.w3schools.com/colors/colors_names.asp

CSS사용법

내부 스타일 시트 (Internal Style Sheet)

```
<html lang="en">
<head>
  <style type="text/css">
    #body-title {
      font-size: 14px;
      font-weight: bold;
      color: DarkSlateGrey;
    }
  </style>
</head>
<body>
  <p id="body-title">Internal Style Sheet</p>
</body>
</html>
```

head 안쪽, style태그 내부에 작성

CSS사용법

인라인 스타일 시트 (Inline Style Sheet)

```
<html lang="en">
<head>
</head>
<body>
  <p id=“body-title" style="font-size:14px; font-weight: bold; color: DarkSlateGrey”>Inline Style Sheet</p>
</body>
</html>
```

사용할 요소의 style속성에 정의

인라인 스타일은 내용과 스타일이 분리되지 않으므로 권장되지 않습니다

CSS사용법

외부 스타일 시트 (External Style Sheet)

```
<html lang="en">
<head>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <p id="body-title">External Style Sheet</p>
</body>
</html>
```

link태그를 사용, href속성에 경로를 입력

link태그를 사용하여 외부 CSS파일을 HTML문서에 연결합니다

개발자 도구

Developer tools

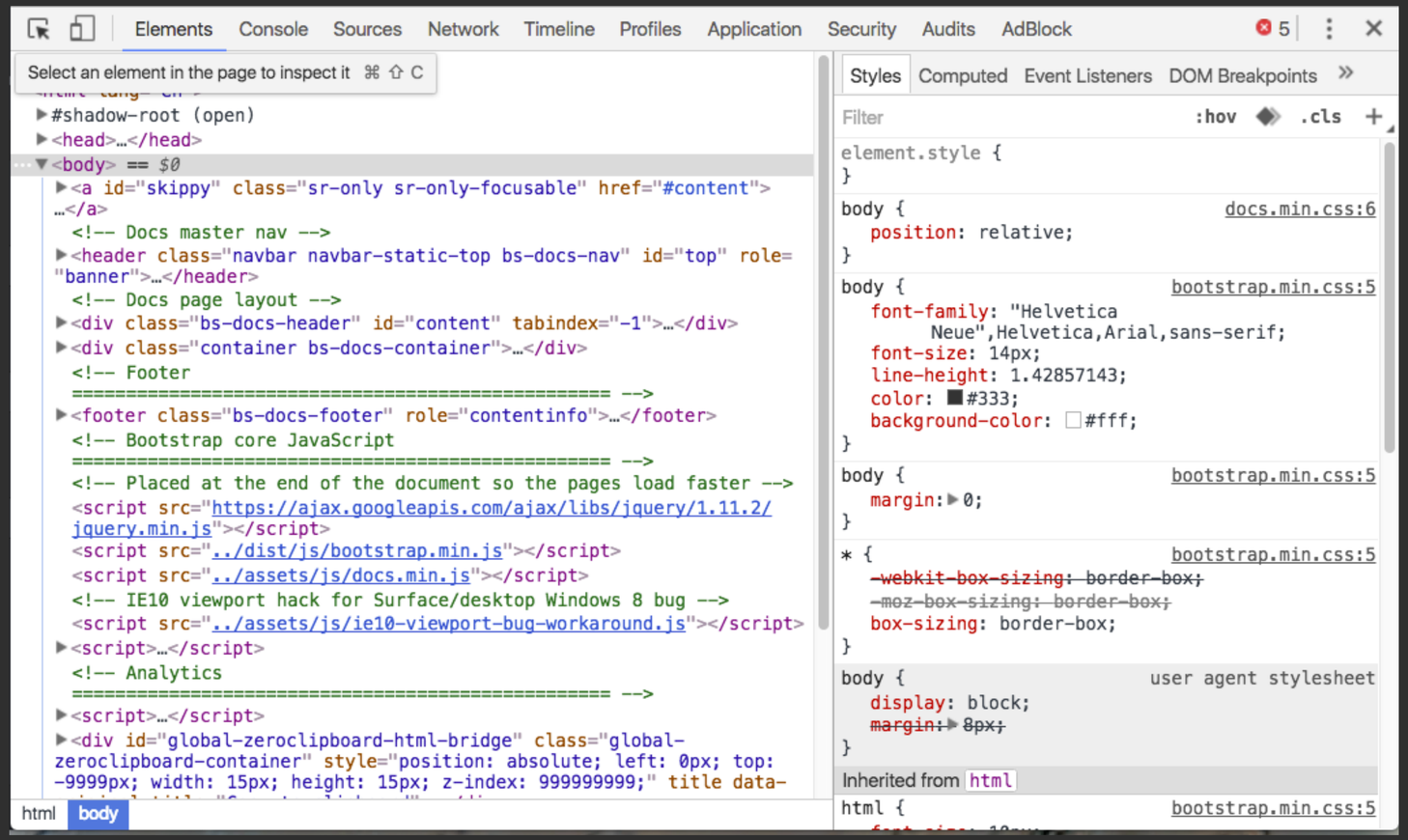
브라우저에서 HTML/CSS요소를 디버깅하는 법에 대해 알아봅니다

요소(Elements)

IE의 경우 DOM 탐색기 또는 HTML, 파이어폭스의 경우 검사기(Inspector)

F12 or Ctrl + Shift + I
Cmd + Alt + I

현재 렌더링 된
HTML요소들을 보여줍니다



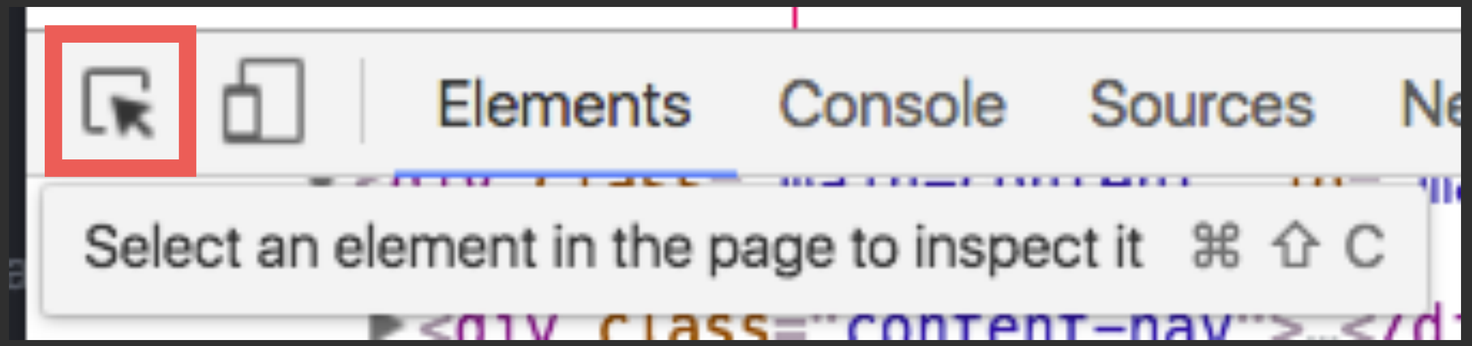
선택한 요소에 대한
CSS값을 보여줍니다

우측에서 직접 CSS속성을 추가/변경/삭제하며 실시간으로 결과를 확인할 수 있습니다.
좌측에서 직접 HTML요소를 추가/변경하는것도 가능합니다.

요소(Elements)

IE의 경우 DOM 탐색기 또는 HTML, 파이어폭스의 경우 검사기(Inspector)

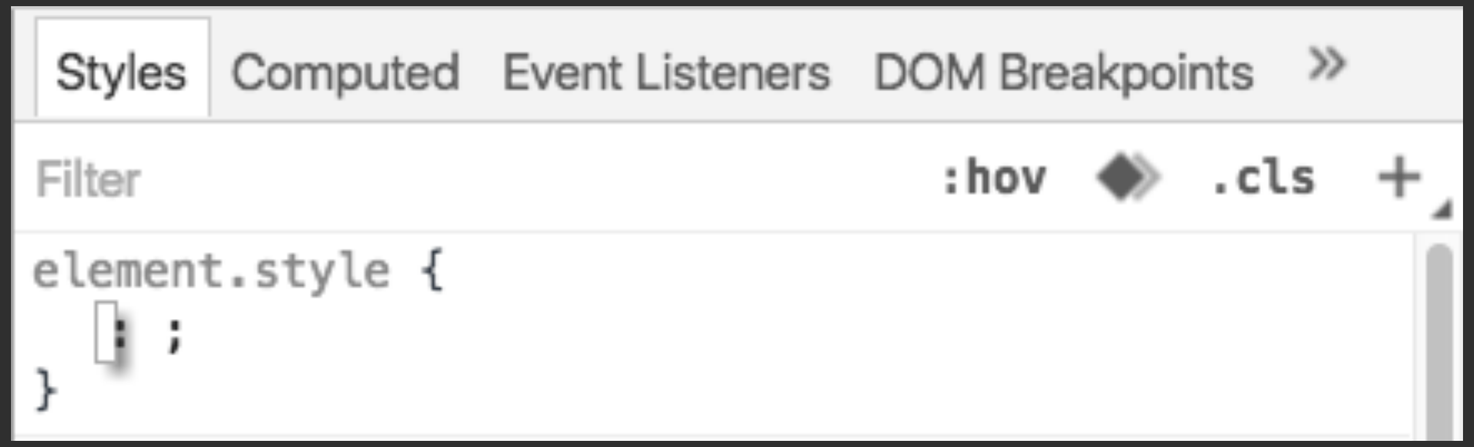
요소 선택
cmd + shift + c



자주 사용하는 단축키이며
외워두시는게 좋습니다

요소(Elements) - 스타일(Styles)

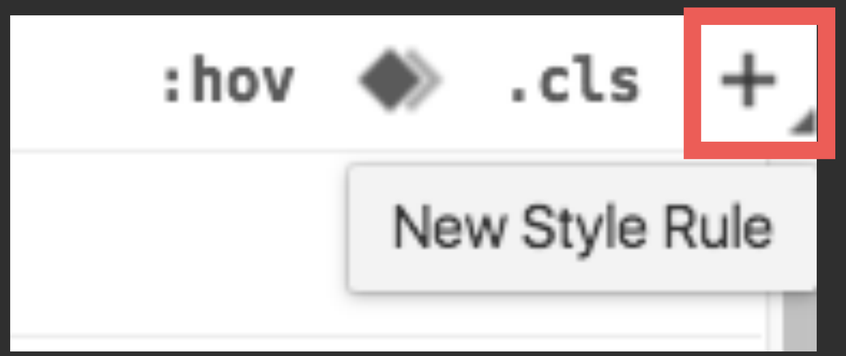
IE의 경우 DOM 탐색기 또는 HTML, 파이어폭스의 경우 검사기(Inspector)

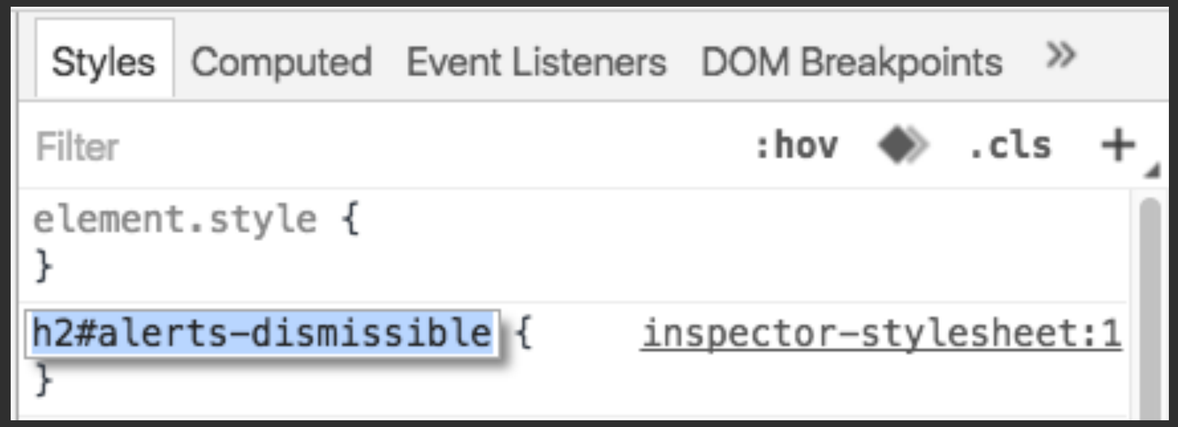


element.style에 넣은 값은 Inline스타일로 적용됩니다.

요소(Elements) - 스타일(Styles)

IE의 경우 DOM 탐색기 또는 HTML, 파이어폭스의 경우 검사기(Inspector)

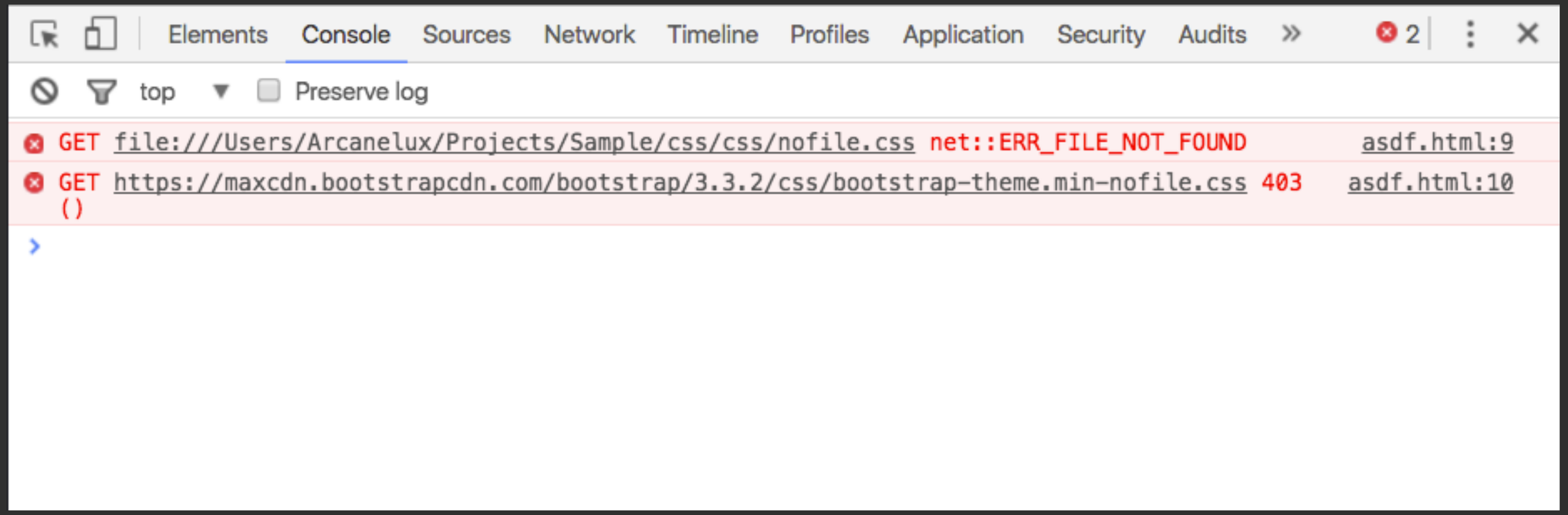




New Style Rule버튼을 누르면, 현재 요소를 특정하는 선택자를 자동으로 생성해줍니다.

콘솔(Console)

실시간으로 내부의 자바스크립트를 테스트하거나, 로그를 출력해줍니다

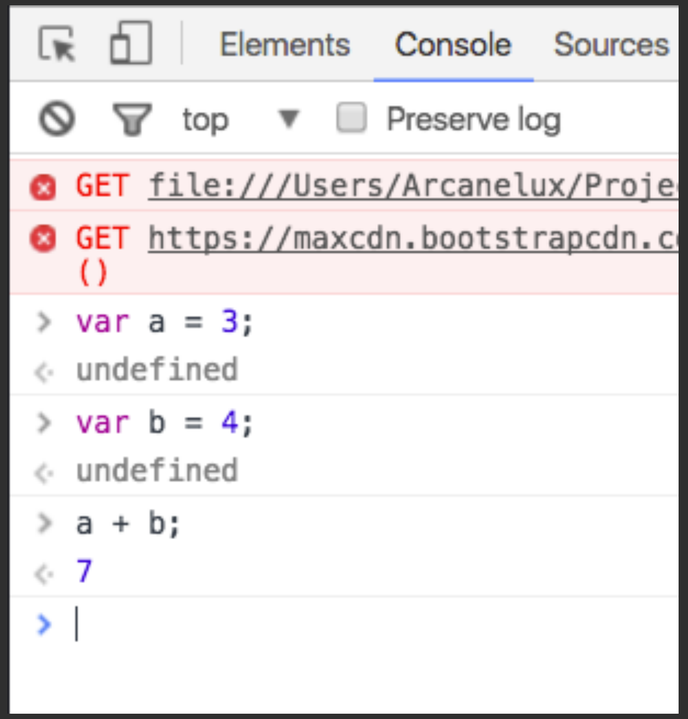


링크로 불러오려는 파일이 없는 경우, 이곳에 에러메세지가 표시됩니다.

css파일이 적용이 안되거나, 이미지파일이 출력되지 않을 경우 먼저 Console화면을 확인해보는것이 좋습니다.

콘솔(Console)

실시간으로 내부의 자바스크립트를 테스트하거나, 로그를 출력해줍니다



파이썬 인터프리터와 비슷하게, 현재 브라우저에서 자바스크립트를 실행할 수 있습니다.
HTML페이지에서 자바스크립트 파일을 로드한 경우, 로드한 모듈도 사용 가능합니다.

CSS 선택자

CSS Selector

CSS를 적용할 요소

CSS선택자

Universal Selector (전체 선택자)

별표 기호 (Asterisk)

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

HTML페이지 내부의 모든 요소에 같은 CSS속성을 적용합니다.
따라서 margin이나 padding값을 초기화하는 등, 기본값을 정할 때 주로 사용합니다.
다만 문서의 모든 요소를 읽기 때문에 페이지 로딩시간이 길어질 수 있으니 자주 사용하는 것은 좋지 않습니다.

CSS선택자

Tag Selector (태그 선택자)

HTML 태그명

HTML 태그명

```
h1 {
  color: red;
}
p {
  color: gray;
  margin-bottom: 10px;
}
```

해당하는 모든 HTML태그 요소를 지정합니다.

CSS선택자

Class Selector (클래스 선택자)

CSS

마침표 기호

```css
.section {
  color: #333;
  margin-bottom: 40px;
}
p.section-title {
  font-size: 18px;
}
p.section-content {
  font-size: 13px;
  line-height: 13px;
  color: #999;
}
```

앞에 TAG명 입력 가능

HTML

```html
<body>
  <div class="section">
    <p class="section-title">Lorem ipsum dolor sit amet.</p>
    <p class="section-content">Lorem ipsum dolor sit amet</p>
  </div>

  <div class="section">
    <p class="section-title">Lorem ipsum dolor sit amet.</p>
    <p class="section-content">Lorem ipsum dolor sit amet</p>
  </div>
</body>
```

CSS에서는 마침표 기호로 나타내며, HTML에서는 주어진 값을 class속성값으로 가진 요소를 선택합니다.
마침표 앞에 태그를 붙여주면 범위는 지정한 태그에 한합니다.

CSS선택자

ID Selector (ID 선택자)

CSS

기호

```css
#index-title {
  font-size: 18px;
}
p#index-description {
  font-size: 12px;
  color: #999;
}
```

앞에 TAG명 입력 가능

HTML

```html
<body>
  <h3 id="index-title">Lorem ipsum dolor sit.</h3>
  <p id="index-description">Lorem ipsum doloro?</p>
</body>
```

CSS에서는 #기호로 나타내며, HTML에서는 주어진 값을 id속성값으로 가진 요소를 선택합니다.
HTML에서 id값은 오직 하나만 존재해야 합니다.
클래스 선택자와 같이 앞에 TAG명을 입력할 수 있습니다.
ID선택자의 우선순위가 Class선택자의 수선순위보다 높으므로, 같은 속성에 서로 다른 값을 지정할 경우 ID선택자의 값이 적용됩니다

CSS선택자

Group Selector (그룹 선택자)

CSS

```css
#index-title {
  font-size: 18px;
}
p#index-description {
  font-size: 12px;
  color: #999;
}
#index-title, #index-description {
  text-align: center;
}
```

둘 이상 요소에 같은 스타일 적용

HTML

```html
<body>
  <h3 id="index-title">Lorem ipsum dolor sit.</h3>
  <p id="index-description">Lorem ipsum doloro?</p>
</body>
```

여러 선택자에 같은 스타일을 적용하는 경우, 쉼표로 구분해 선택자를 나열해 사용합니다

CSS선택자

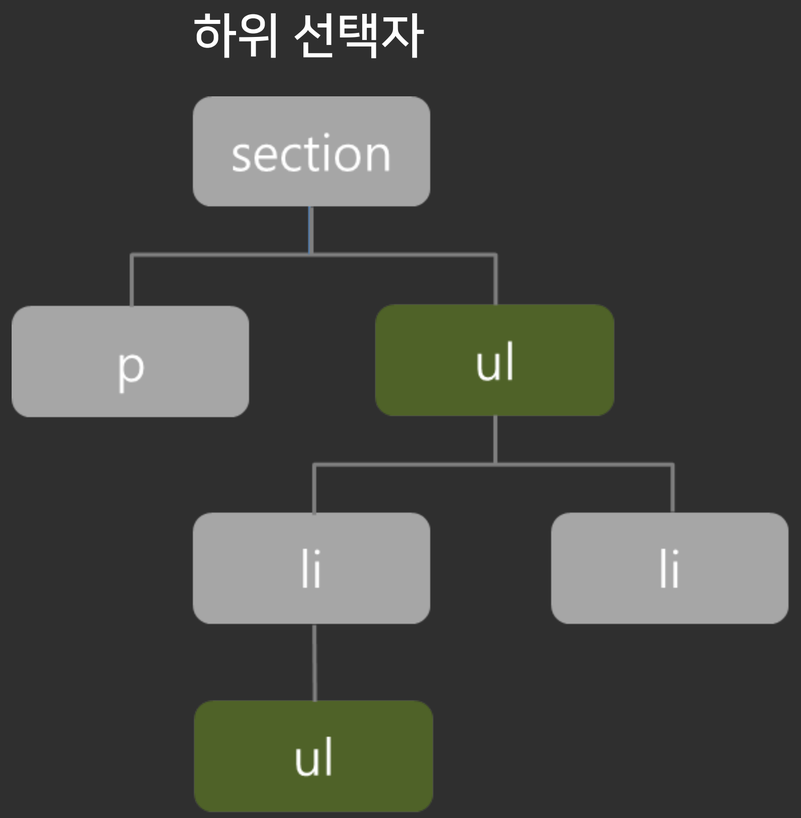
Combinator Selector (복합 선택자)

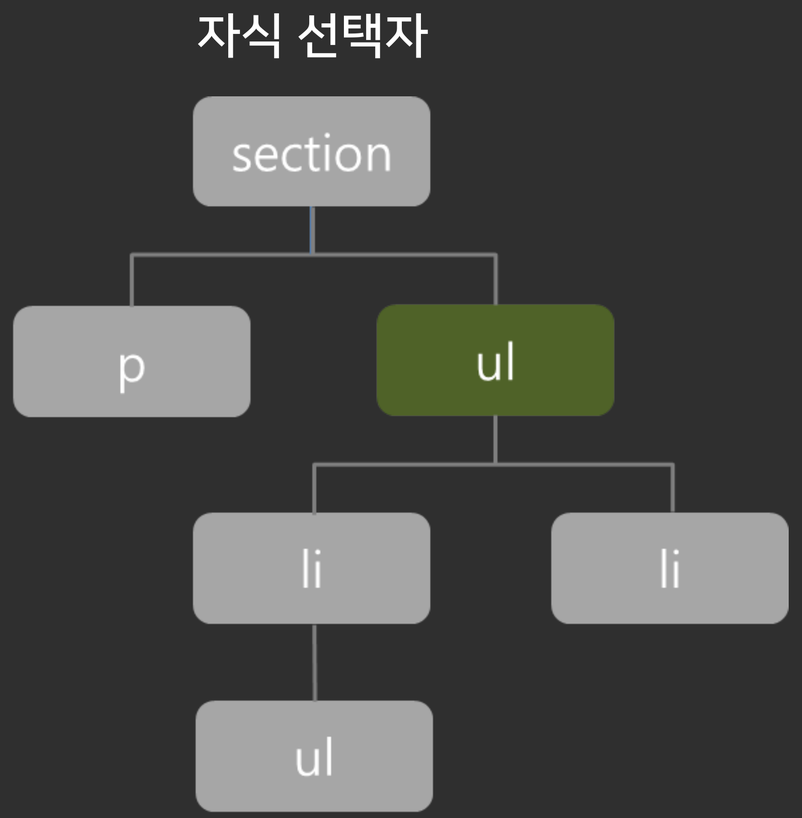
하위선택자와 자식선택자 (Descendant and Child)

하위 선택자 (descendant selector)

```
section ul {
  border: 1px solid black;
}
section > ul {
  border: 1px solid black;
}
```

자식 선택자 (Child selector)





포함 관계를 가지는 태그들 사이에서, 포함하는 요소는 '부모 요소', 포함되는 요소는 '자식 요소'라고 합니다.
하위 선택자는 부모요소에 포함된 '모든' 하위 요소를 지정하며,
자식 선택자는 부모요소의 '바로 아래' 자식 요소만을 지정합니다.

CSS선택자

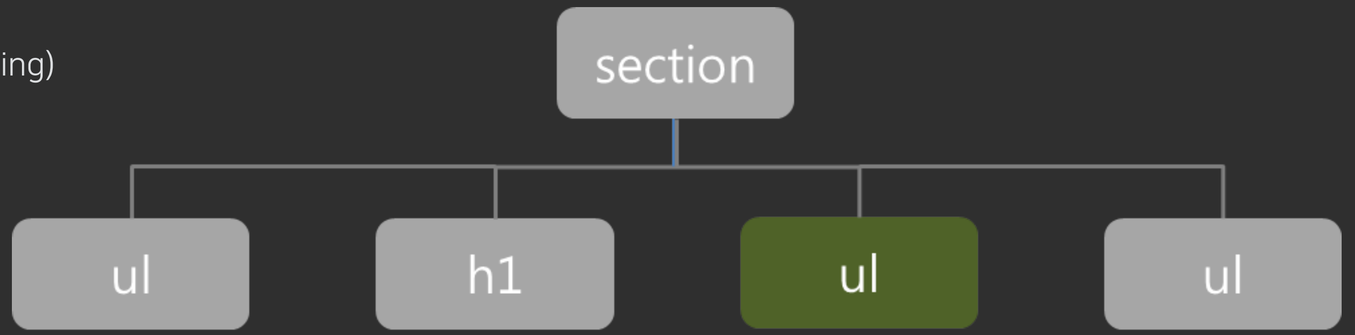
Combinator Selector (복합 선택자)

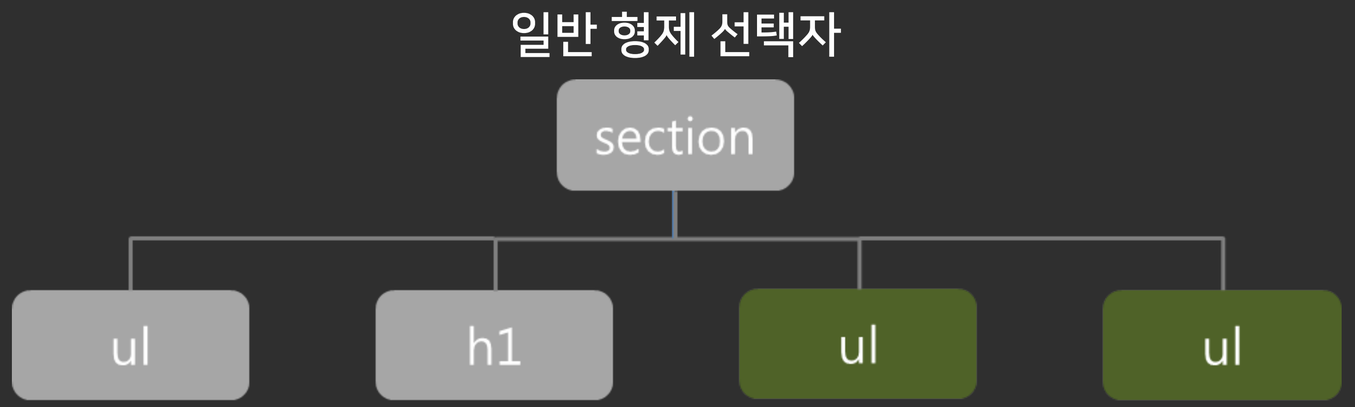
인접 형제 선택자와 일반 형제 선택자 (Adjacent Sibling & General Sibling)

인접 형제 선택자

```
h1 + ul {
  background: Azure;
  color: DarkBlue;
}
h1 ~ ul {
  background: Azure;
  color: DarkBlue;
}
```

일반 형제 선택자





같은 부모 요소를 가지는 요소들은 '형제 관계' 라고 부릅니다.
이 때, 먼저 나오는 요소를 '형 요소', 나중에 나오는 요소는 '동생 요소'라 합니다. (부모 요소 내부에서 보다 윗 줄에 쓰여진 것을 의미합니다)

두 선택자 모두 형 요소에는 적용되지 않으며, 인접 형제 선택자는 조건을 충족하는 '첫 번째' 동생 요소만을 지정하며,
일반 형제 선택자는 조건을 충족하는 '모든' 동생 요소를 지정합니다.

CSS선택자

속성 선택자 (Attribute Selector)

태그 내의 속성에 따름

패턴	의미	예제
E[attr]	'attr'속성이 포함된 요소 E	<E attr>Lorem</E>
E[attr="val"]	'attr'속성의 값이 'val'인 요소 E	<E attr="val">Lorem</E>
E[attr~="val"]	'attr'속성의 값에 'val'이 포함되는 요소 E (공백으로 분리된 값이 일치해야 함)	<E attr="val">Lorem</E> <E attr="item val">Lorem</E> <E attr="item-val">Lorem</E>
E[attr|="val"]	'attr'속성의 값에 'val'이 포함되거나 (공백분리), 'val-'로 시작하는 요소 E	<E attr="val">Lorem</E> <E attr="val-num3">Lorem</E> <E attr="value">Lorem</E>
E[attr^="val"]	'attr'속성의 값이 'val'로 시작하는 요소 E	<E attr="val">Lorem</E> <E attr="value">Lorem</E>
E[attr$="val"]	'attr'속성의 값이 'val'로 끝나는 요소 E	<E attr="val">Lorem</E> <E attr="item-val">Lorem</E>
E[attr*="val"]	'attr'속성의 값에 'val'이 포함되는 요소 E (공백이나 Dash(-)에 영향받지 않음)	<E attr="val">Lorem</E> <E attr="many itemval">Lorem</E> <E attr="item-val">Lorem</E>

CSS선택자

가상 클래스 선택자 (Pseudo-Classes Selector), 가상 엘리먼트 선택자 (Pseudo-Elements Selector)

HTML소스에는 존재하지 않지만 필요에 의해 가상의 선택자를 지정

패턴	의미
E:link	방문하지 않은 링크 E
E:visited	방문한 링크 E
E:active	E 요소에 마우스 클릭 또는 키보드 엔터가 눌린 동안
E:hover	E 요소에 마우스가 올라가 있는 동안
E:focus	E 요소에 포커스가 머물러 있는 동안
E::first-line	E 요소의 첫 번째 라인
E::first-letter	E 요소의 첫 번째 문자
E::before	E 요소의 시작 지점에 생성된 요소
E::after	E 요소의 끝 지점에 생성된 요소

CSS 우선순위

CSS Cascading

스타일 적용의 우선순위를 알아봅니다

CSS스타일 적용 우선순위

특정도(specify)값이 높은 순서대로 적용됩니다

특정도는 가장 구체적인 값을 의미합니다

특정도 계산식

스타일	특정도
Inline (인라인 스타일)	A
ID Selector (ID 선택자)	B
Class Selector (클래스 선택자)	C
TAG Selector (태그 선택자)	D

특정도는 CSS구문에서
해당하는 스타일의 수 * 특정도 값을 모두 더한 값입니다.

예제)

CSS 구문	특정도
p { color: gray; }	1D
p:first-line { color: black; }	2D
.wrap { color: black; }	1C
p.wrap { color: black; }	1C,1D
p.wrap>.item { color: black; }	2C,1D
#wrap { color: black; }	1B
p#wrap { color: black; }	1B,1D
<!— HTML —> Example	1A

클래스 선택자가 3개 존재하며, 태그선택자가 있는 CSS구문이라면

ex) p.wrap.item>.active

클래스선택자의 갯수(3) * 클래스선택자의 특정도(10) + 태그선택자의 개수(1) * 태그 선택자의 특정도(1) = 31의 특정도를 가지게 됩니다.

우선순위의 흑마법

인라인 스타일과 important

from 위키백과 : 흑마술은 전통적으로 악의적, 이기적 목적을 위한 초자연적인 힘의 이용을 일컫는다.

특정도 계산식

스타일	특정도
important	Absolute
Inline (인라인 스타일)	A
ID Selector (ID 선택자)	B
Class Selector (클래스 선택자)	C
TAG Selector (태그 선택자)	D

CSS

```css
p {
  font-size: 30px !important;
  color: green !important;
}
```

HTML

```html
<p style="font-size: 10px; color: green;">important는 모든걸 무시하지</p>
```

important 쓴 것을 잊어버리면 나중에 왜 안되는지 헤매게 됨

!important값은 해당하는 요소에 지정된 어떤 특정도도 무시하고 가장 우선순위로 적용됩니다.
이는 테스트시에는 유용할 수 있지만, 추후 유지보수나 전체적 스타일 흐름을 방해하기 때문에 이용하지 않는 것이 좋습니다.

CSS 서체

CSS Typography

CSS로 서체에 스타일을 지정합시다

서체 색상

Color

```
h1 {
  color: #ff0000;
}
```

color속성은 글자 색을 지정합니다.
HEX코드, rgb(), rgba(), 색상명이 올 수 있습니다.

서체 지정

Font family

```
body {
  font-family: “돋움”, dotum, “굴림”, gulim, arial, helvetica, sans-serif;
}
```

웹 페이지를 방문한 사용자가 없을 경우를 대비해, 다양한 서체들을 선언해놓습니다.
순서대로 해당 서체가 없을 경우 다음 서체가 사용자의 컴퓨터에 설치되어 있는지 확인 후, 순서 중 가장 먼저 찾은 폰트를 사용하게 됩니다.

서체의 종류

Font types

Serif

글자에 꺾쇠가 붙어있는 서체, 인쇄물에 많이 사용
영어 서체에는 Georgia, Times New Roman
한글 서체에는 바탕체, 궁서체, 명조체가 있다

Python, 파이썬

Monospace

고정 폭 서체
글자들 간 구분이 쉬워야 하는
프로그래밍 코드를 나타낼 때 주로 사용

Python, 파이썬

Sans-Serif

Sans는 “없음”을 뜻하는 프랑스어.
Serif가 없는 글시체를 말함
영어 서체에는 Arial, Helvetica
한글 서체에는 돋움, 굴림, 나눔 고딕등이 있다.

Python, 파이썬

Cursive

커브가 많이 들어간 서체
필기체를 말함
가독성이 떨어지니 일부분에만 사용하는 것이 좋다

Python, 파이썬

글자 크기

Font size

```
body {
  font-size: 14px;
}
h1 {
  font-size: 28px(2em);
}
```

** em은 부모 요소로부터의 비율입니다.

부모(body)의 font-size가 14px이며, h1의 font-size가 2em이면 14px * 2.0 = 28px이 적용됩니다.

글자 스타일

Font style

```
p {
  font-style: italic;
}
```

italic과 oblique는 둘 다 기울임꼴을 나타내지만, italic은 별도의 기울어진 폰트가 있을 경우 해당 폰트를 사용합니다.
inherit은 상위 요소의 font-style을 물려받아 나타냅니다.

글자 굵기

Font weight

```
p {
  font-weight: bold;
}
p {
  font-weight: 700;
}
```

lighter
normal
bold
bolder
inherit

<400
400
700
>700

lighter와 bolder는, 해당 서체의 Lighter또는 Bolder서체가 있어야 표현되며,
해당하는 서체가 없을 경우 normal, bold와 동일 굵기로 나타납니다.

줄 간격

Line height

숫자만 입력할 경우 해당 font-size 에 곱한 값이 줄 간격이 됩니다

```
p {
  line-height: 1.5;
}
```

** font-size가 px로 고정되어 있다면, line-height에도 고정된 px을 사용할 수 있습니다.

문자 꾸미기

text-decoration

```
p.item1 {
  text-decoration: none;
}
p.item2 {
  text-decoration: underline;
}
p.item3 {
  text-decoration: overline;
}
p.item4 {
  text-decoration: line-through;
}
```

밑줄

윗줄

취소선

문자 정렬

text-align

```
p.item1 {
  text-align: left;
}
p.item2 {
  text-align: center;
}
p.item3 {
  text-align: right;
}
p.item4 {
  text-align: justify;
}
```

양쪽정렬

** justify는 우측 끝 부분을 깔끔하게 양쪽 정렬해주지만, 줄의 내부 간격이 뒤틀리므로 잘 사용하지 않습니다.

문자 들여쓰기

text-indent

-값일 경우, 내어쓰기

```
p {
  text-indent: 1em;
}
```

대소문자 변환

text-transform

```
p.item1 {
  text-transform: capitalize;
}
p.item2 {
  text-transform: uppercase;
}
p.item3 {
  text-transform: lowercase;
}
```

첫 글자만 대문자로

모두 대문자로

모두 소문자로

당연히 영문에만 적용됩니다.

자간

letter-spacing

```
p {
  letter-spacing: 5px;
}
```

각 글자간의 간격

단어 간격

word-spacing

글자가 아닌, 단어간의 간격을 조절합니다

```
p {
  word-spacing: 5px;
}
```

각 단어간의 간격

요소간 수직 정렬

verticla-align

박스 내에서의 수직 정렬이 아닌, 나란히 오는 인라인 요소간의 정렬

```html
<div>
  <strong>baseline : </strong>
  <span class="ori">align</span>
  <span class="vitem item1">Text</span>
</div>

<div>
  <strong>sub : </strong>
  <span class="ori">align</span>
  <span class="vitem item2">Text</span>
</div>

<div>
  <strong>super : </strong>
  <span class="ori">align</span>
  <span class="vitem item3">Text</span>
</div>

<div>
  <strong>top : </strong>
  <span class="ori">align</span>
  <span class="vitem item4">Text</span>
</div>

<div>
  <strong>text-top : </strong>
  <span class="ori">align</span>
  <span class="vitem item5">Text</span>
</div>

<div>
  <strong>bottom : </strong>
  <span class="ori">align</span>
  <span class="vitem item6">Text</span>
</div>

<div>
  <strong>text-bottom : </strong>
  <span class="ori">align</span>
  <span class="vitem item7">Text</span>
</div>

<div>
  <strong>middle : </strong>
  <span class="ori">align</span>
  <span class="vitem item8">Text</span>
</div>
```

```css
.ori {
  font-size: 20px;
}
.vertical-item {
  font-size: 10px;
}
.item1 {
  vertical-align: baseline;
}
.item2 {
  vertical-align: sub;
}
.item3 {
  vertical-align: super;
}
.item4 {
  vertical-align: top;
}
.item5 {
  vertical-align: text-top;
}
.item6 {
  vertical-align: bottom;
}
.item7 {
  vertical-align: text-bottom;
}
.item8 {
  vertical-align: middle;
}
```

줄 바꿈 설정

text-align

```
p.item1 {
  white-space: nowrap;
}
p.item2 {
  white-space: pre;
}
p.item3 {
  white-space: pre-line;
}
p.item4 {
  white-space: pre-wrap;
}
```

줄 바꿈이 없습니다

줄 바꿈, 띄어쓰기, 공백등 모든것이 보여지며,
박스를 벗어나도 줄 바꿈이 일어나지 않습니다

줄 바꿈만 보여주며, 띄어쓰기는 무시합니다
자동으로 줄바꿈이 됩니다

줄 바꿈과 띄어쓰기를 모두 보여주며
자동으로 줄바꿈이 됩니다

CSS 배경 스타일

CSS Background

CSS로 요소(Element)에 배경을 지정합니다

배경색

background-color

```css
div {
  background-color: #eee;
  background-color: #efefef;
  background-color: rgb(230, 222, 120);
  background-color: rgba(230, 22, 120, 0.4);
}
```

HEX Code (#RRGGBB or #RGB)

rgb(0~255값으로 표현)

alpha(투명도)를 조절

배경 이미지

background-image

```
div {
  background-image: url('../images/icon.png');
}
```

이미지의 주소는 상대경로, 절대경로 모두 사용가능합니다.

배경 이미지 반복

background-repeat

```
div {
  background-repeat: no-repeat;
  background-repeat: repeat;
  background-repeat: repeat-x;
  background-repeat: repeat-y;
}
```

반복하지 않음

가로세로 바둑판 형식으로 반복

가로(x축)으로만 반복

세로(y축)으로만 반복

가로로 반복하는 이미지의 경우, 세로로 길고 가로 1px인 이미지를 이용해 배경을 나타낼 수 있으며,
세로반복의 경우 반대로 가로로길고 세로가 1px인 이미지를 이용해 배경을 나타낼 수 있습니다.

배경 이미지 반복

background-repeat

세로로 긴 이미지를
repeat-x를 이용해
가로로 반복하면
배경화면이 됩니다
(그라데이션 효과)

배경 이미지 위치

background-position

```
div {
  background-position: 50% 16px;
  background-position: center bottom;
}
```

center는 가운데 (x,y모두 가능)
left, right는 x축
top, bottom은 y축에만 사용

삽입된 이미지의 좌표를 정해줍니다.
두 개의 값을 받으며, 각각 x축, y축의 값을 가집니다.
각각은 양의값을 가질경우 x축은 우측, y축은 하단으로 이동합니다. (음수를 가질경우 반대로 좌측, 상단으로 이동합니다)
%값을 사용할 경우, 이미지 사이즈를 기준으로 움직입니다. (x축이 100%일 경우, 이미지의 너비만큼 우측에서 보이게 됨)
left, center, right, top, bottom으로 x, y측의 0%, 100%, 가운데 값을 사용할 수 있습니다.

배경 이미지 고정

background-attachment

```
div {
  background-attachment: local;
  background-attachment: scroll;
  background-attachment: fixed;
}
```

요소 안 내용에 고정
(스크롤할 때 이미지가 같이 움직임)

요소에 고정 (스크롤에 무관)

background-position좌표를 웹 페이지 화면 전체기준으로 합니다

local값을 가질 경우, 요소의 왼쪽 상단을 기준으로 이미지를 표현하며, 스크롤 시 이미지가 같이 움직입니다.
scroll값을 가질 경우, 요소의 왼쪽 상단을 기준으로 이미지를 표현하며, 스크롤 시 이미지가 함께 움직이지 않고 고정됩니다.
fixed값을 가질 경우, 요소와 관계없이 웹 페이지 전체 화면을 기준으로 이미지가 표시되며, 스크롤 시 함께 움직이지 않고 고정됩니다.

배경 속기법

background

image repeat attachement position color

```
div {
  background: url('images/sample01.png') no-repeat scroll right 50% #eee;
}
```

지금까지 배운 background관련 속성을 한 번에 줄여쓸 수 있습니다.

순서대로 image, repeat, attachment, position(x,y), color값을 나타냅니다.

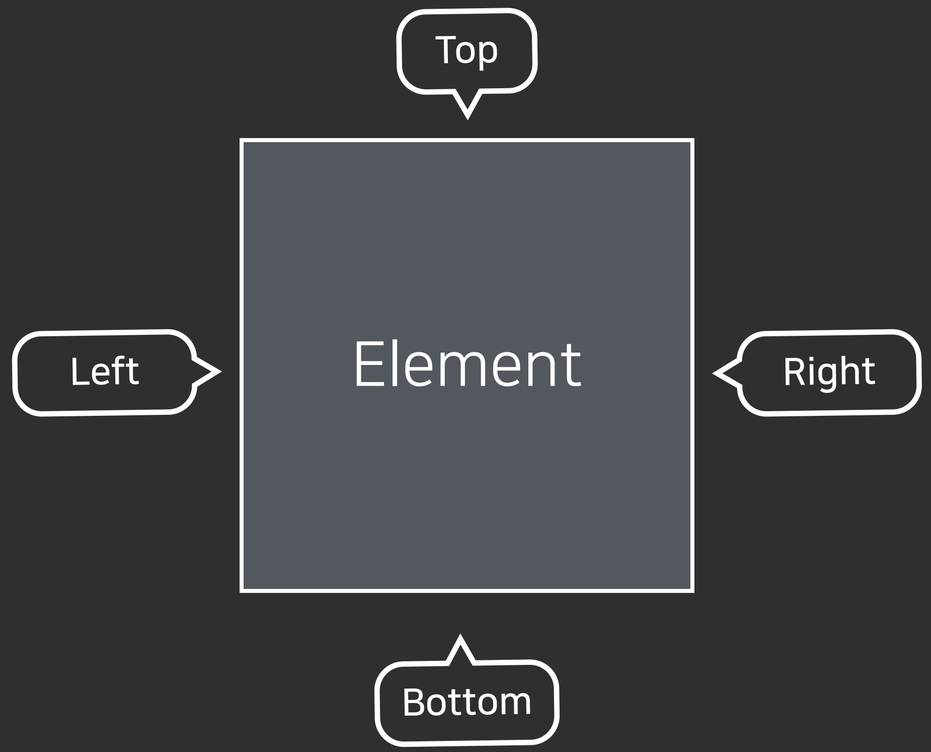
CSS 테두리 스타일

CSS Border

CSS로 요소(Element)에 테두리를 지정합니다

요소의 방향

Element’s direction



요소의 상하좌우 속성을 정의할 때, 순서는 시계방향으로 진행됩니다.
상단부터 시계방향으로, Top -> Right -> Bottom -> Left방향으로 값을 정한다고 기억하시면 됩니다.

테두리를 구성하는 요소

선 굵기 (border-width)

```
div {
  border-width: 3px;
  border-top-width: 4px;
  border-width: 3px 4px 5px 6px;
  border-width: 5px 10px;
}
```

상하좌우 모두 3px

상단만 4px

상 우 하 좌 순서대로 3,4,5,6px

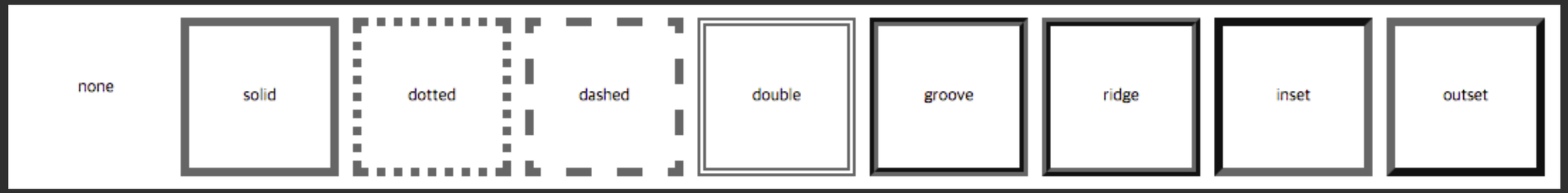
상하 5px, 좌우 10px

값을 2개만 적을 경우, 첫 번째 값은 상&하의 값, 두 번째 값은 좌&우의 값을 나타냅니다.

테두리를 구성하는 요소

선 형태 (border-style)

```
div {
  border-style: solid;
  border-top-style: double;
  border-style: solid double dashed dotted;
}
```



solid 실선
dotted 점선
dashed 바느질선 형태의 점선

double 이중선
groove 입체적으로 보여줌
ridge groove와 반대방향으로 선이 돌출

inset 요소 전체가 안으로 들어가 보임
outset 요소 전체가 바깥으로 나와 보임

테두리를 구성하는 요소

선 색상 (border-color)

```
div {
  border-color: #aaa;
  border-color: red blue green yellow;
}
```

지금까지와 마찬가지로 Hex code, rgb, rgba, ColorName 전부 사용 가능합니다.

테두리 속성 속기법

border

```
div {
  border: 1px solid red;
}
```

border의 속기법은 모든 변에 동일한 값만 적용 가능합니다.
(각 변에 다른 값을 주고 싶을 경우, 각 속성(width, style등)에 4가지 값을 입력하거나, border-top-<property>에 값을 입력해야 합니다.

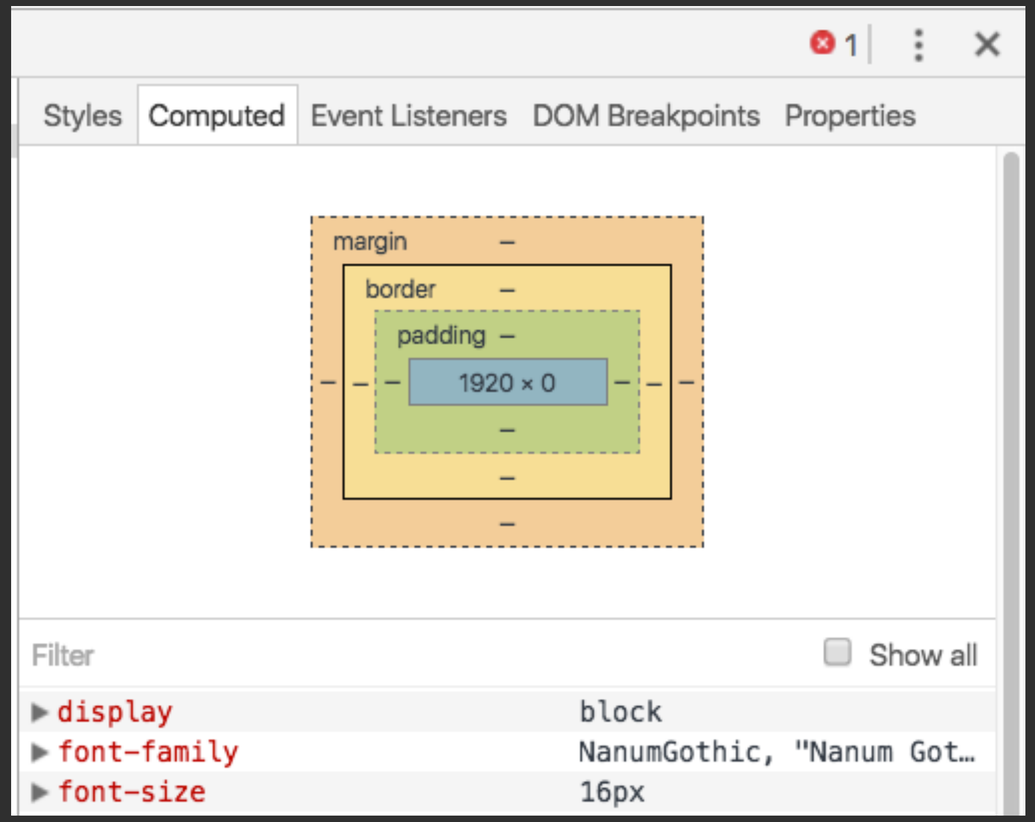
CSS 박스 모델

CSS Box Model

CSS로 요소를 구성하는 박스에 대해 알아봅니다

박스모델?

CSS요소를 이루는 형태

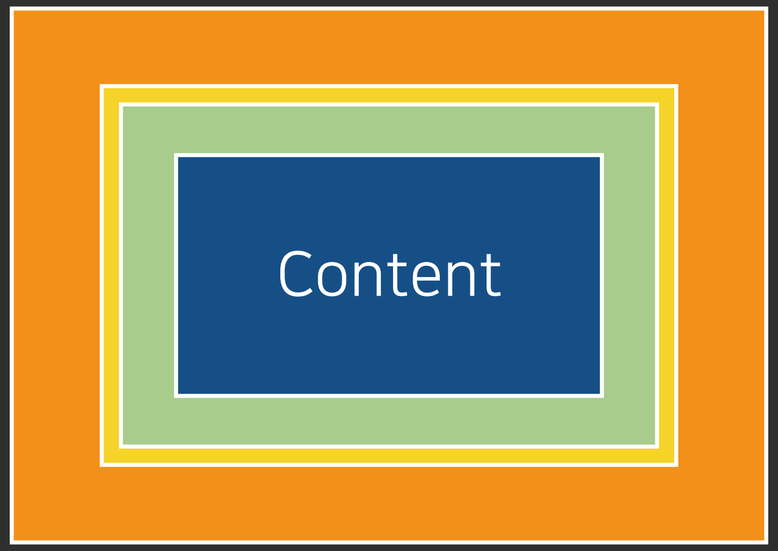


개발자 도구에서 CSS요소의 박스 모델을 확인 가능

CSS요소는 박스 형태를 이루며,
박스는 콘텐츠, 패딩, 보더, 마진의 4가지로 이루어집니다

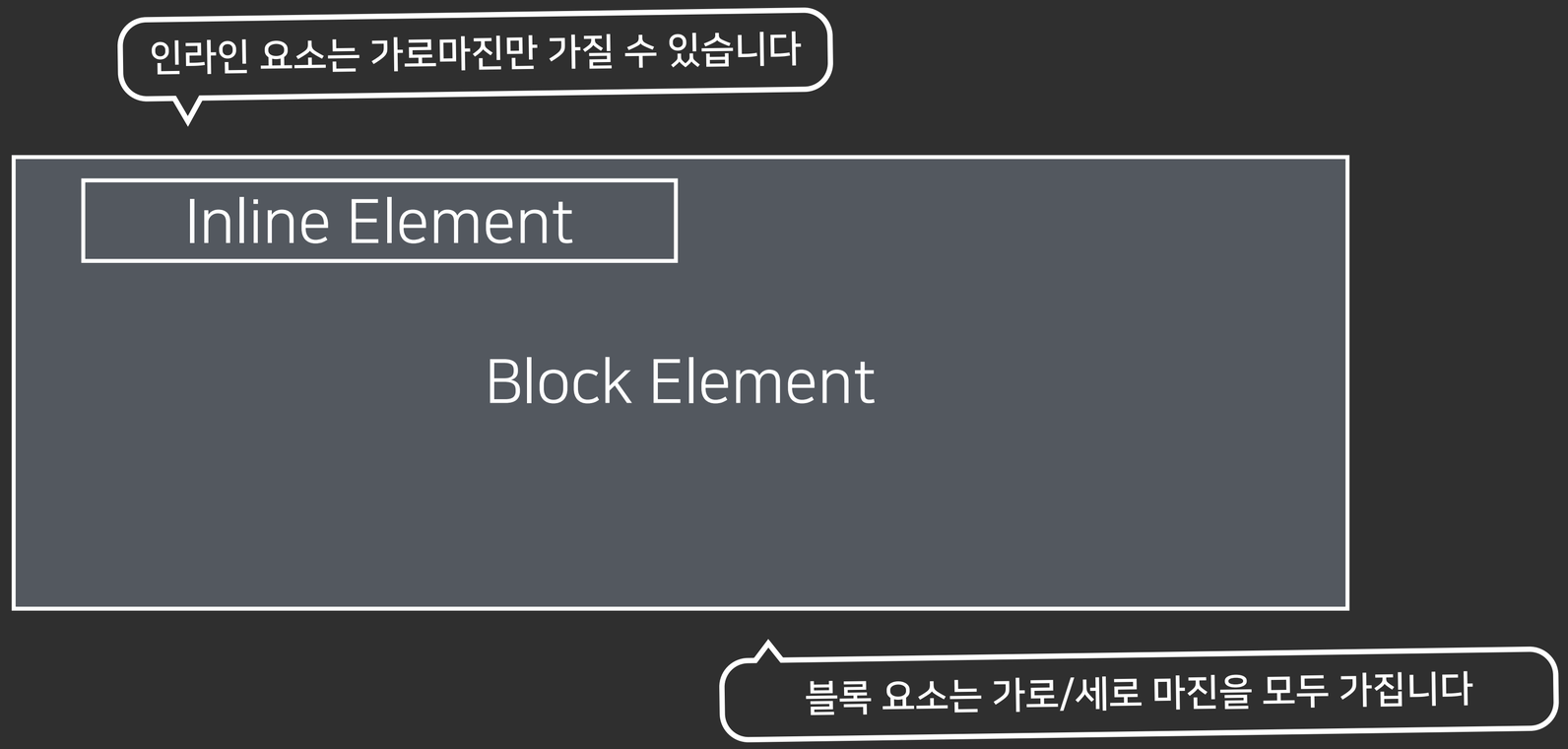
박스모델?

CSS요소를 이루는 형태



CSS요소는 박스 형태를 이루며,
박스는 콘텐츠, 패딩, 보더, 마진의 4가지로 이루어집니다

블록 요소와 인라인 요소의 차이



인라인 요소의 길이/높이는 지정이 불가능합니다 (내용에 자동으로 맞추어짐)

바깥 여백 (마진)

margin

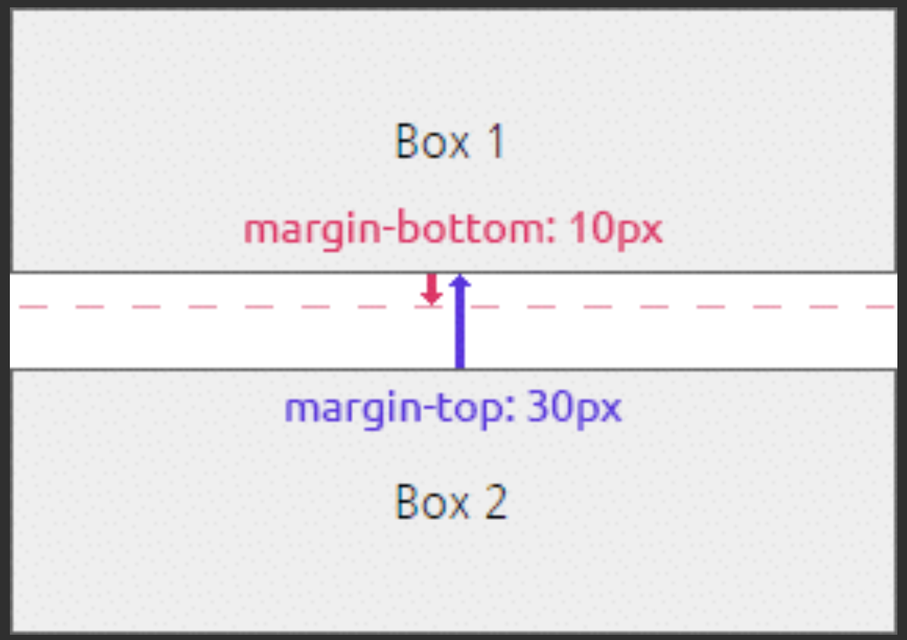
```css
div {
  margin-top: 10px;
  margin-bottom: 30px;
  margin: 10px 0;
  margin: 40px 20px 30px 50px;
}
```

지금까지와 마찬가지로 Hex code, rgb, rgba, ColorName 전부 사용 가능합니다.

마진 겹침

margin collapse

```
<div style="margin-bottom:10px;">Box 1</div>
<div style="margin-top:30px;">Box 2</div>
```



두 블록요소의 마진이 서로 겹칠 경우, 해당하는 마진값이 더해지는 것이 아니라 둘 중 큰 값만이 적용됩니다.
(세로가 아닌 가로에서는 해당 현상이 없습니다)
따라서 서로 위/아래로 겹치는 마진값을 준 경우, 한 쪽에만 값을 몰아주거나, padding을 활용하는 방식으로 해결해야 합니다.

내부 여백 (패딩)

padding

```css
div {
  padding-top: 10px;
  padding-bottom: 10px;
  padding: 10px 0;
  padding: 10px 20px 30px 40px;
}
```

padding과 margin을 구분하는 가장 쉬운법은,
padding과 margin을 준 요소에 background-color를 지정한 후 개발자 모드에서 해당 요소가 차지하는 공간을 확인하는 것입니다.

가로, 세로

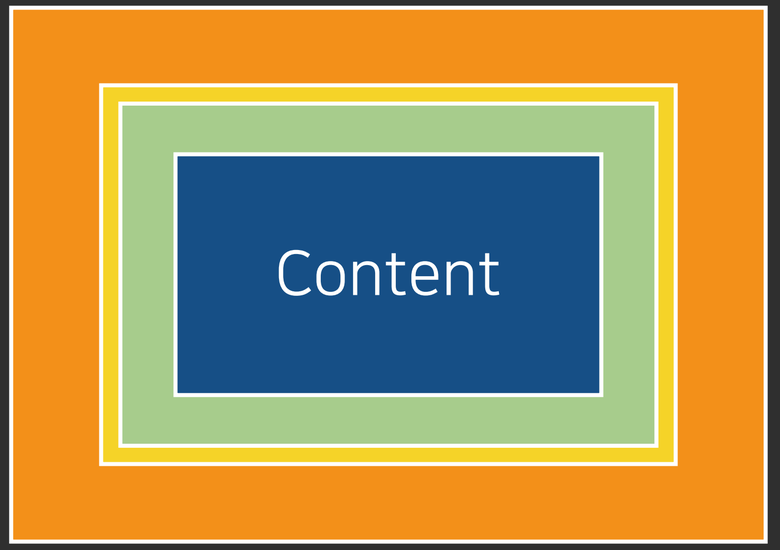
width, height

```
div {
  width: 100px;
  height: 50px;
}
```

가로, 세로

width, height

```
div {
  width: 200px;
  height: 50px;
  padding: 10px;
  border: 1px solid black;
  margin: 15px;
}
```



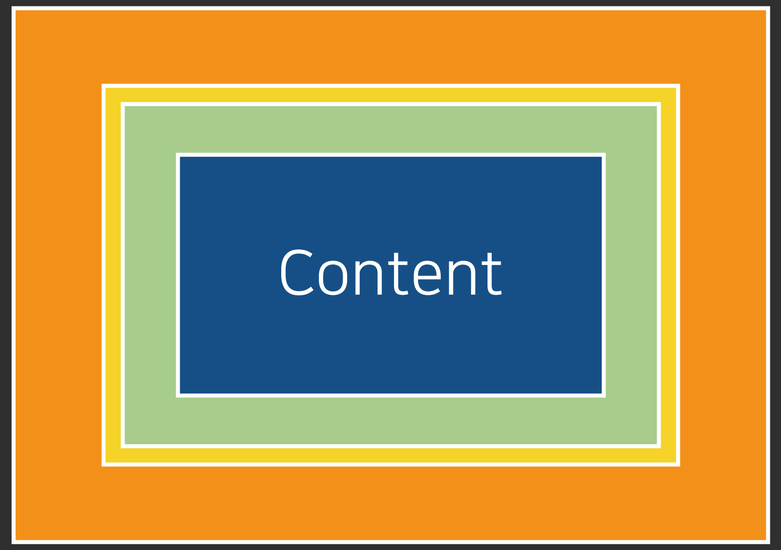
요소의 총 가로길이
200px(가로) + (10px + 1px + 15px) x 2
= 252px

padding과 margin을 구분하는 가장 쉬운법은,
padding과 margin을 준 요소에 background-color를 지정한 후 개발자 모드에서 해당 요소가 차지하는 공간을 확인하는 것입니다.

box-sizing

박스 모델의 크기를 지정합니다

```
div {
    width: 200px;
    height: 50px;
    padding: 10px;
    border: 1px solid black;
    margin: 15px;
    box-sizing: border-box;
}
```



요소의 총 가로길이 200px 고정
Content의 가로길이는 200px - (10px + 1px + 15px) x 2
= 148px

box-sizing에 border-box를 지정하면, 요소의 width값에 맞추어 padding, border, margin값을 제외한 width가 새로 적용됩니다.

CSS 리스트 스타일

CSS List Style

CSS로 리스트에 스타일을 지정합니다

리스트 앞 Bullet타입 설정

list-style-type

```
ul {
  list-style-type: disc;
  list-style-type: circle;
  list-style-type: square;

  list-style-type: decimal;
  list-style-type: lower-alpha;
  list-style-type: upper-alpha;
  list-style-type: lower-roman;
  list-style-type: upper-roman;
}
```

적용 예시

- disc 형태
- circle 형태
- square 형태

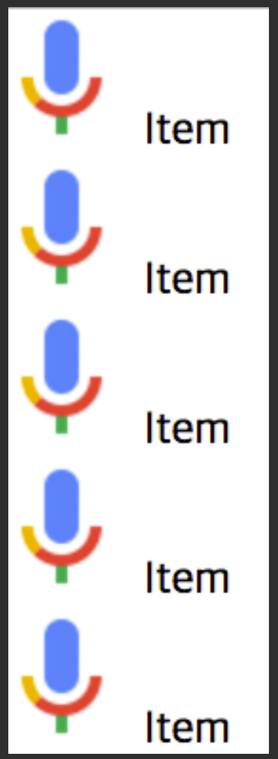
1. decimal
b. lower-alpha
C. upper-alpha
iv. lower-roman
V. upper-roman

disc, circle, square는 Unordered List에 어울리는 속성이며,
decimal, alpha, roman은 Ordered List에 어울리는 속성입니다.

리스트 블릿에 이미지 지정

list-style-image

```
ul {
  list-style-image: url('images/mic.png');
}
```



padding과 margin을 구분하는 가장 쉬운법은,
padding과 margin을 준 요소에 background-color를 지정한 후 개발자 모드에서 해당 요소가 차지하는 공간을 확인하는 것입니다.

리스트 블릿 위치 지정

list-style-position

```
ul {
  list-style-position: inside;
  list-style-position: outside;
}
```

적용 예시

- outside는 기본적인 리스트 형태입니다.
  이렇게 줄 바꿈이 일어나도 블릿과 일정한 여백을 유지합니다.
  - 반면 inside는 본문 내에 위치하게 됩니다.
  때문에 이렇게 줄 바꿈이 일어나면 마치 본문의 일부처럼 보일 수 있습니다.

참고로 위의 리스트의 여백은 ul 요소의 padding으로 조절할 수 있습니다.

리스트 스타일 속기법

list-style

```
ul {
  list-style: square url('images/mic.png') inside;
}
```

블릿 타입

이미지를 사용할 경우

블릿 위치 내부/외부 여부

**블릿타입과 이미지 주소를 동시에 넣을 경우, 이미지를 찾지 못하면 지정한 블릿타입을 사용합니다.

CSS의 테이블 스타일

CSS Table style

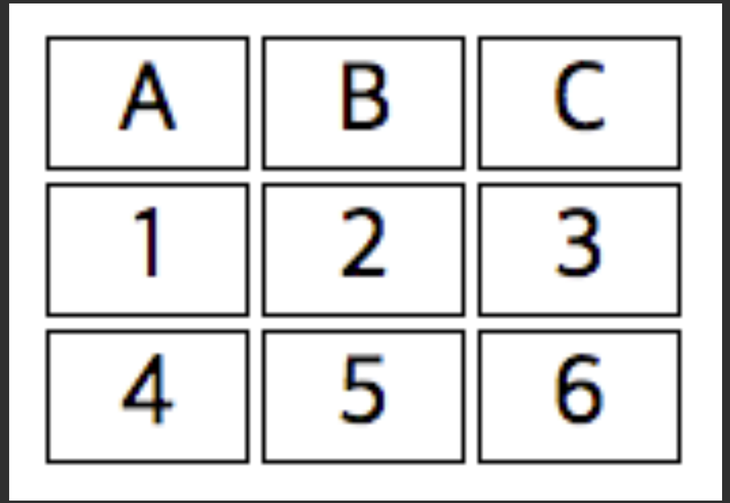
CSS로 테이블을 테이블에 스타일을 지정합니다

테두리 합치기

border-collapse

```
<table>
  <tr>
    <td>A</td>
    <td>B</td>
    <td>C</td>
  </tr>
  <tr>
    <td>1</td>
    <td>2</td>
    <td>3</td>
  </tr>
  <tr>
    <td>4</td>
    <td>5</td>
    <td>6</td>
  </tr>
</table>
```

```
tr, td {
  border: 1px solid black;
  width: 30px;
  text-align: center;
}
```



tr, td에 테두리를 준 경우

테두리 합치기

border-collapse

```
<table>
  <tr>
    <td>A</td>
    <td>B</td>
    <td>C</td>
  </tr>
  <tr>
    <td>1</td>
    <td>2</td>
    <td>3</td>
  </tr>
  <tr>
    <td>4</td>
    <td>5</td>
    <td>6</td>
  </tr>
</table>
```

```
table {
  border-collapse: collapse;
}
tr, td {
  border: 1px solid black;
  width: 30px;
  text-align: center;
}
```

A	B	C
1	2	3
4	5	6

border-collapse 적용

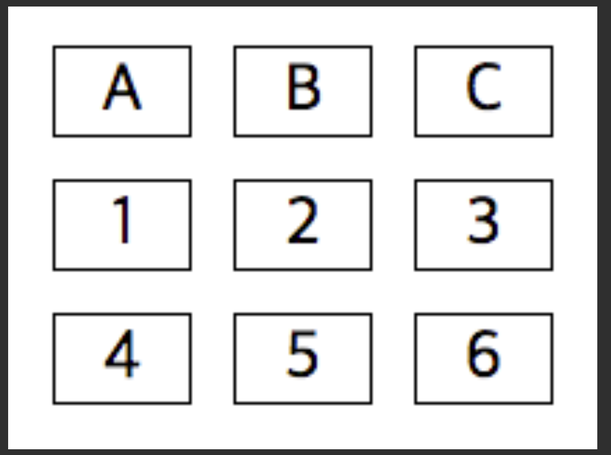
테이블의 셀간 공백을 합쳐서 없애는 속성입니다.
해당속성은 오직 table요소에만 사용 가능합니다.

테이블 셀 간격

border-spacing

```html
<table>
  <tr>
    <td>A</td>
    <td>B</td>
    <td>C</td>
  </tr>
  <tr>
    <td>1</td>
    <td>2</td>
    <td>3</td>
  </tr>
  <tr>
    <td>4</td>
    <td>5</td>
    <td>6</td>
  </tr>
</table>
```

```css
table {
  border-spacing: 10px;
}
tr, td {
  border: 1px solid black;
  width: 30px;
  text-align: center;
}
```



border-spacing 적용

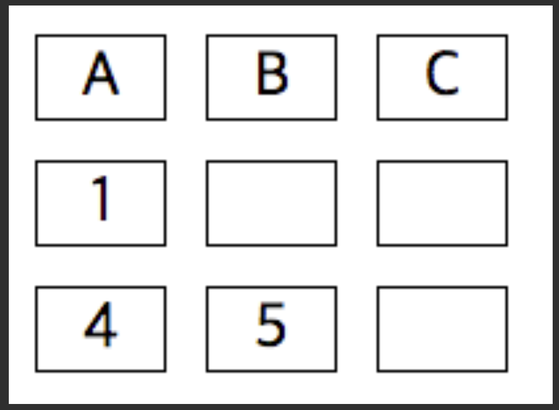
**셀 간 간격을 지정할 때는, border-collapse가 'collapse'값이면 적용되지 않습니다.

빈 셀의 표시

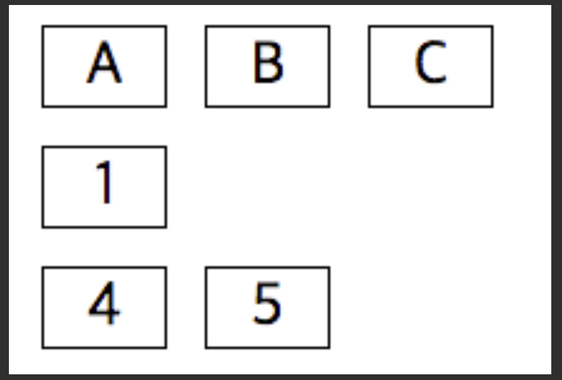
empty-cells

```html
<table>
  <tr>
    <td>A</td>
    <td>B</td>
    <td>C</td>
  </tr>
  <tr>
    <td>1</td>
    <td></td>
    <td></td>
  </tr>
  <tr>
    <td>4</td>
    <td>5</td>
    <td></td>
  </tr>
</table>
```

```css
table {
  border-spacing: 10px;
}
tr, td {
  border: 1px solid black;
  width: 30px;
  text-align: center;
  empty-cells: hide;
}
```



empty-cells에 show적용



empty-cells에 hide적용

**셀 간 간격을 지정할 때는, border-collapse가 'collapse'값이면 적용되지 않습니다.

테이블 레이아웃

table-layout

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Laudantium, neque.	Lorem.

테이블의 기본 설정은 내용이 긴 쪽이 더 늘어납니다
table-layout: auto;

테이블 레이아웃

table-layout

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Laudantium, neque.	Lorem.

table-layout의 속성을 fixed로 설정하면, 셀 길이가 일정하게 유지됩니다.
(이 때, table에는 width property가 설정되어있어야 합니다)

배운내용을 복습하며 문서를 작성해봅시다
https://namu.wiki/w/박보영

CSS의 화면 표시 속성

CSS display

CSS로 요소가 어떤 모습으로 보일지 결정합니다

화면 표시방법

display: block

span은 원래 인라인속성이지만,
display 프로퍼티에 block값을 받으면 블럭 속성을 지니게 됩니다

```
span {
  display: block;
}
```

기존에 블럭 요소가 아닌 요소를 블럭 속성을 갖도록 합니다.

화면 표시방법

display: inline

반대로, div가 diplay프로퍼티에 inline속성을 받으면
인라인 요소로 취급됩니다

```
div {
  display: inline;
}
```

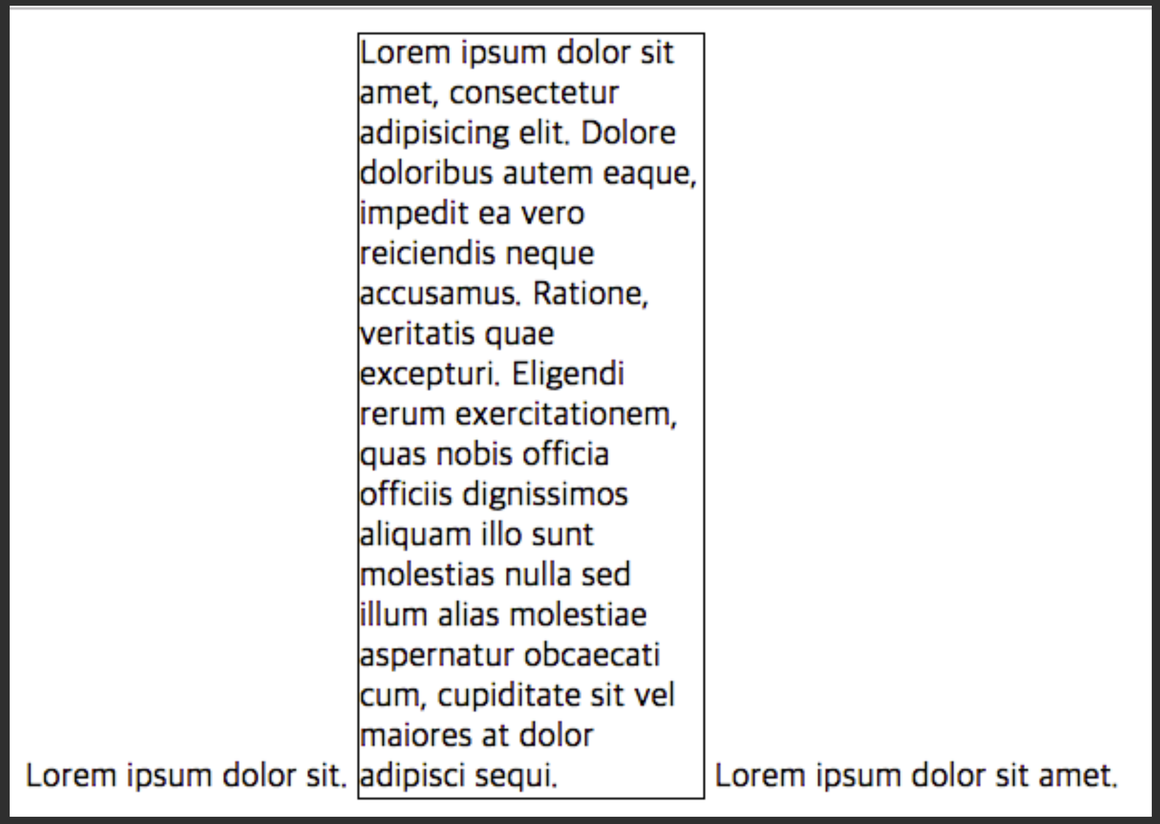
기존에 인라인 요소가 아닌 요소를 인라인 속성을 갖도록 합니다.

화면 표시방법

display: inline-block

기본적으로 inline요소처럼 취급되지만,
block요소와 같이 높이 및 상/하값을 가질 수 있게됩니다

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



display 프로퍼티에 inline-block속성을 가진 span

인라인요소에 높이와 상/하 패딩, 마진값을 줄 때 사용합니다.

화면 표시방법

display: none

화면에 완전히 보이지 않게 됩니다

```
div {
  display: none;
}
```

해당 요소의 하위 요소들도 전부 보이지 않게 되며, 공간도 차지하지 않습니다.

화면 표시방법

visibility

화면에 공간은 차지하나, 투명해집니다

```
div {
  visibility: hidden;
  visibility: visible;
}
```

display: none;은 화면에서 차지하던 공간조차도 전부 없어지며,
visibility: hidden;은 화면에서 차지하던 공간은 그대로인채 투명해진다고 생각하시면 됩니다.

화면 넘침 표시방법

overflow

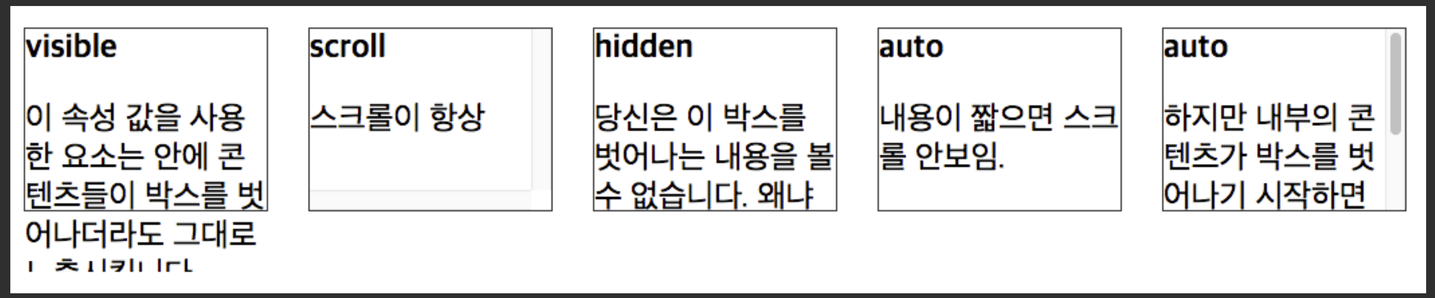
```
div {
  overflow: hidden;
  overflow: visible;
  overflow: auto;
  overflow: scroll;
}
```

넘친 콘텐츠를 전부 숨깁니다

영역 밖으로 콘텐츠가 나간 모습이 보입니다

콘텐츠가 넘칠 경우, 스크롤바가 생성됩니다

콘텐츠가 넘치지 않아도 항상 스크롤바가 있습니다



높이가 고정되어있고, 내용이 길면 스크롤 할 부분에서는 auto를 사용합니다.

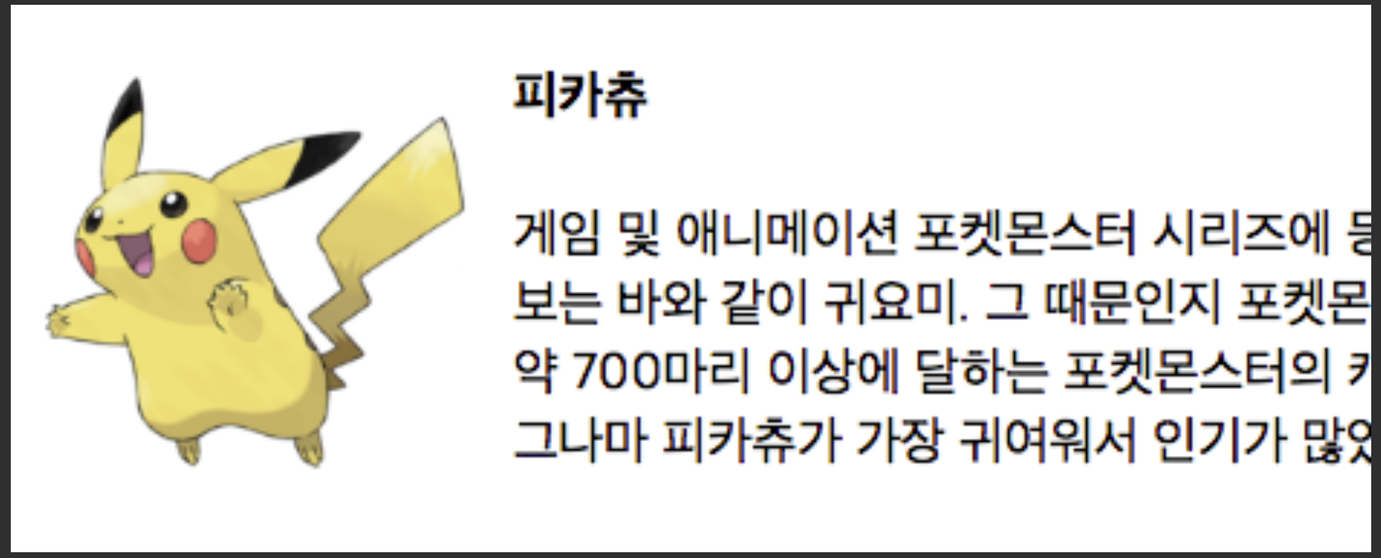
CSS float속성

CSS float

CSS로 요소를 띄웁니다

float속성

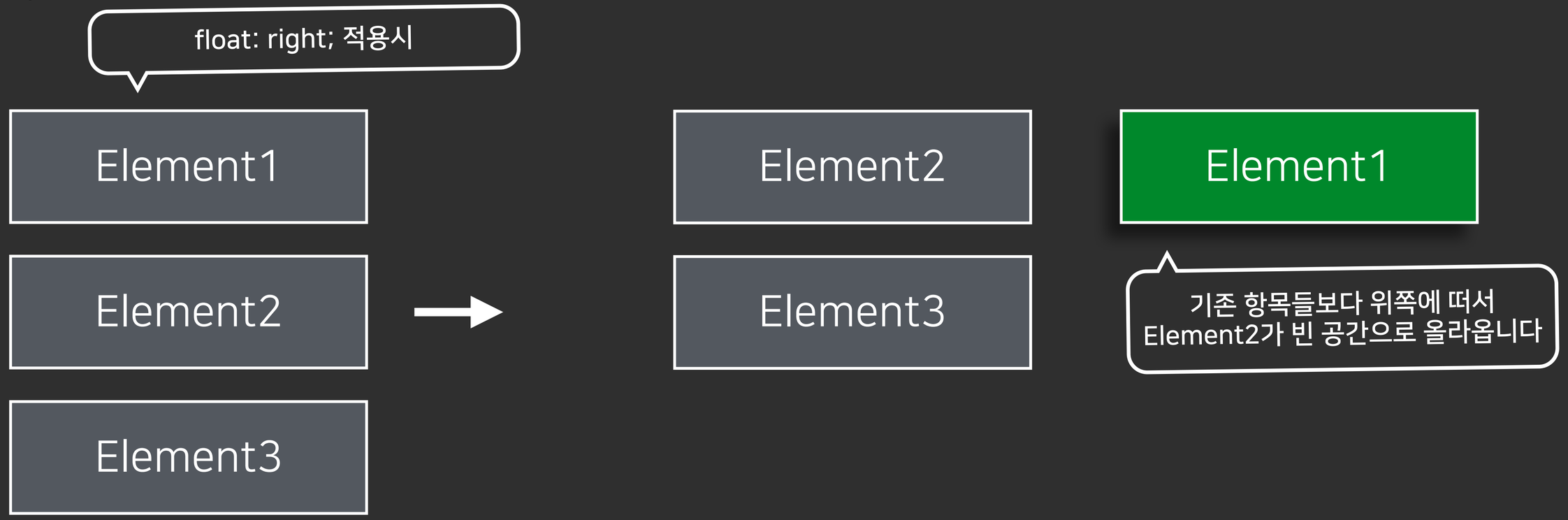
float: 띄우다



float속성은 중간에 이미지를 넣은 단락을 만들고자 하는 경우에도 사용가능합니다.

float속성

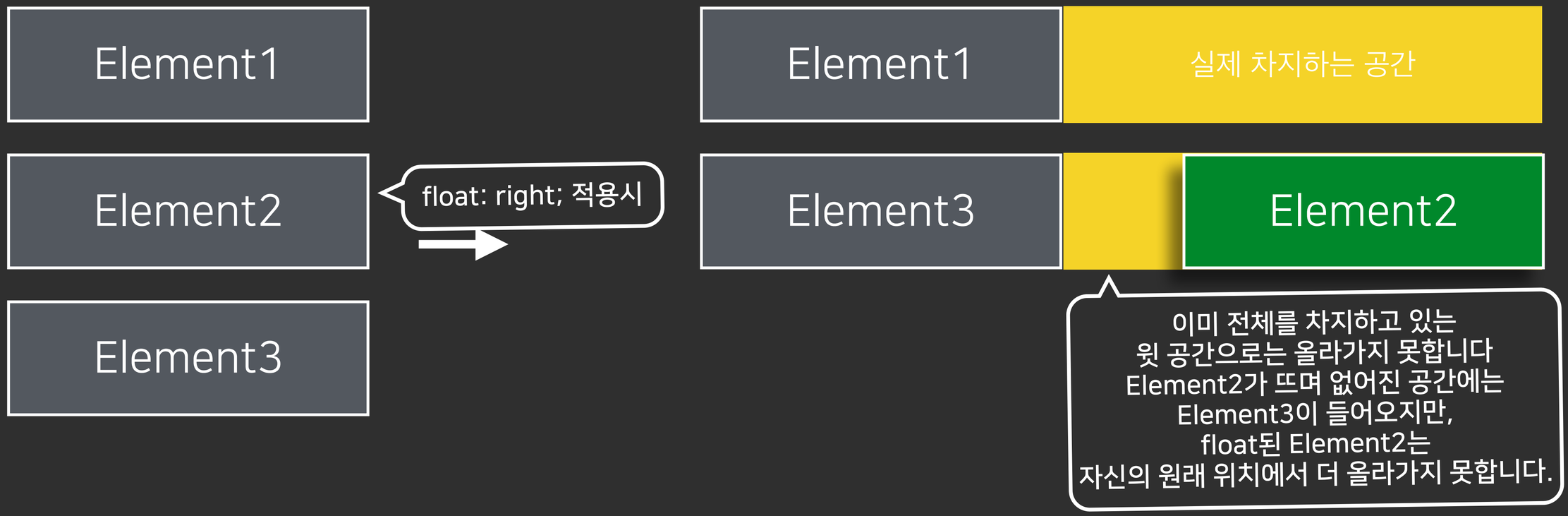
float: 띄우다



float를 사용하면 해당 요소를 문서의 흐름과 별개로 취급하여, 왼쪽이나 오른쪽으로 띄워줄 수 있습니다.

float속성

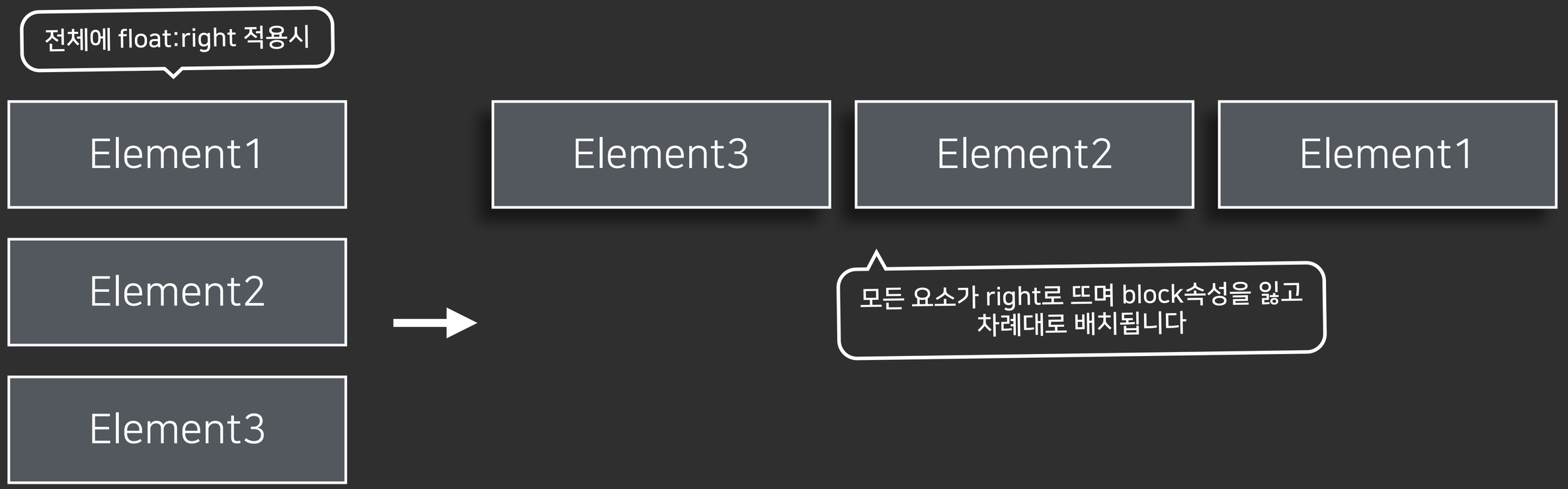
float: 띄우다



Element1이 block요소라면, 해당 요소가 가로 너비를 차지하므로 그 아래쪽에서 우측으로 가게됩니다

float속성

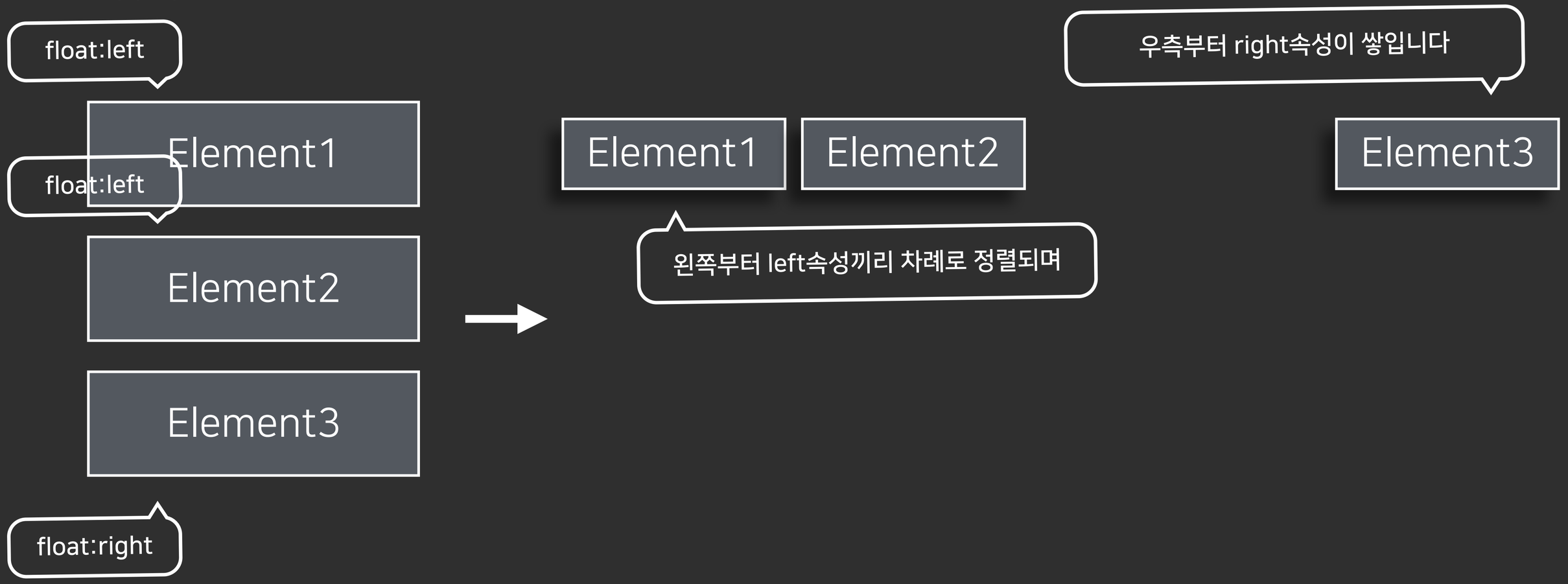
float: 띄우다



Element1이 block요소라면, 해당 요소가 가로 너비를 차지하므로 그 아래쪽에서 우측으로 가게됩니다

float속성

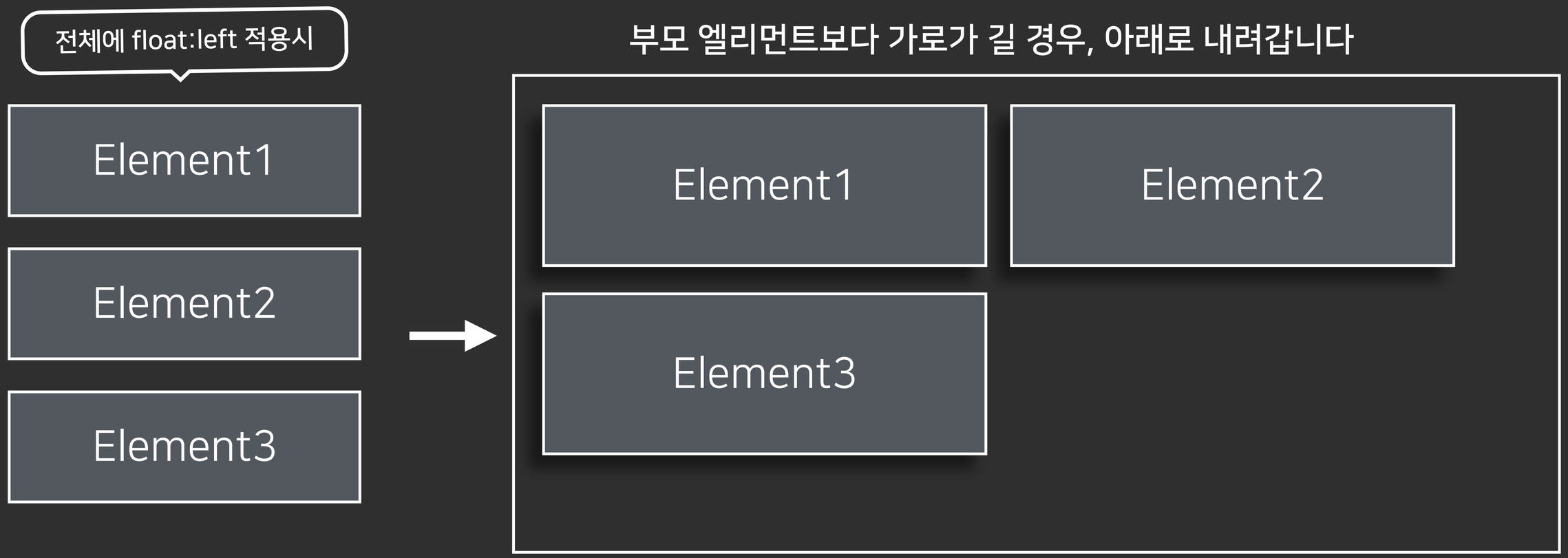
float: 띄우다



Element1이 block요소라면, 해당 요소가 가로 너비를 차지하므로 그 아래쪽에서 우측으로 가게됩니다

float속성

float: 띄우다



Element1이 block요소라면, 해당 요소가 가로 너비를 차지하므로 그 아래쪽에서 우측으로 가게됩니다

clear 속성

float요소와 겹치는 경우, float속성을 해제합니다

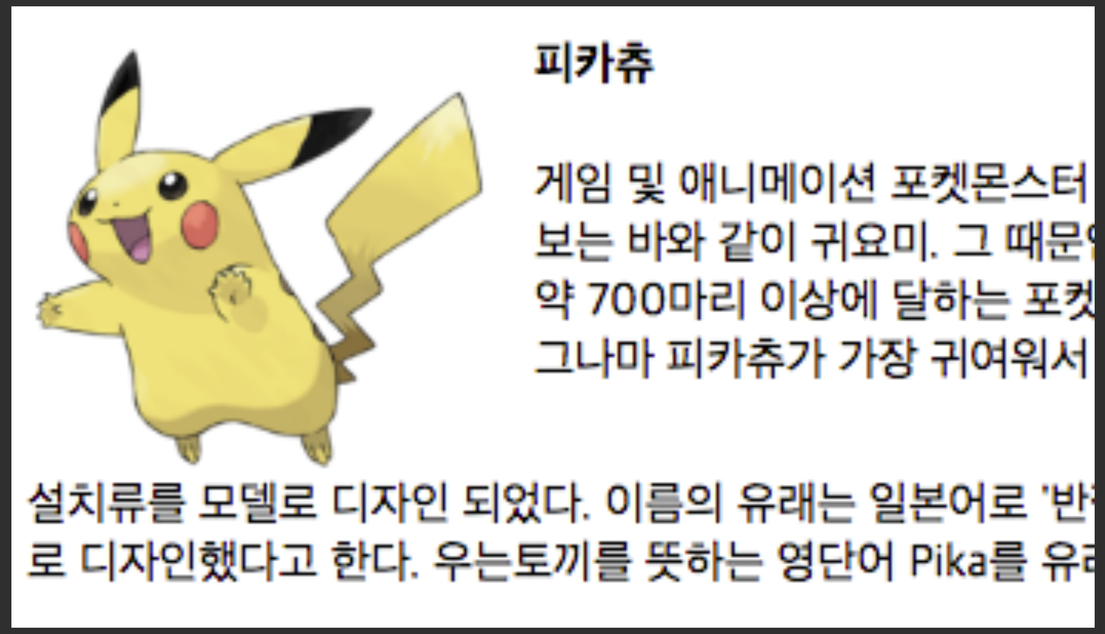
float:left

두 번째 문단은 이미지의 float에 관계없이 block처리되어 아래로 내려가도록 하려면?

clear 속성

float요소와 겹치는 경우, float속성을 해제합니다

float:left



아래쪽 요소에 {clear: left;} 지정

clear 속성

float요소와 겹치는 경우, float속성을 해제합니다

```
p {
  clear: both;
  clear: left;
  clear: right;
}
```

left, right모두 해제

CSS float 레이아웃

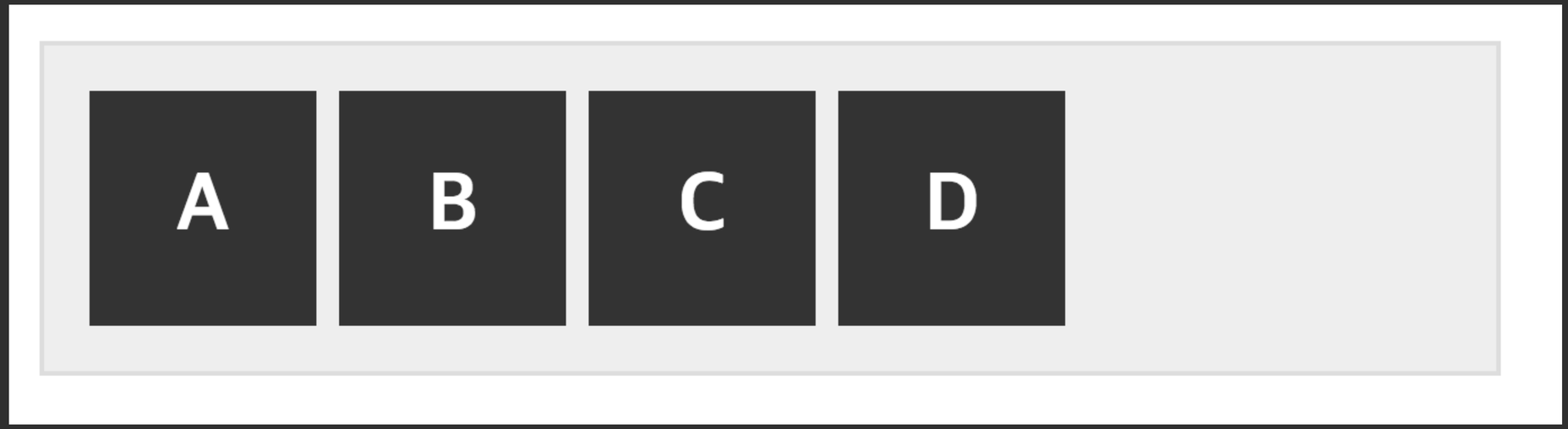
CSS float layout

CSS의 float속성으로 레이아웃을 구현합니다

float 레이아웃

float layout

지금까지는 없었지만, 앨범이나 슬라이드 등에서 자주 쓰이는 레이아웃입니다



CSS로 레이아웃을 구현할 때는 float속성을 자주 사용합니다.

float 레이아웃

float layout

CSS

```css
.float-frame {
  width: 300px;
  background-color: #eee;
  border: 1px solid #ddd;
  padding: 10px;
}

.float-unit {
  width: 50px;
  background: #333;
  color: #fff;
  font-size: 18px;
  font-weight: bold;
  text-align: center;
  padding: 15px 0;
  margin-right: 5px;
}
```

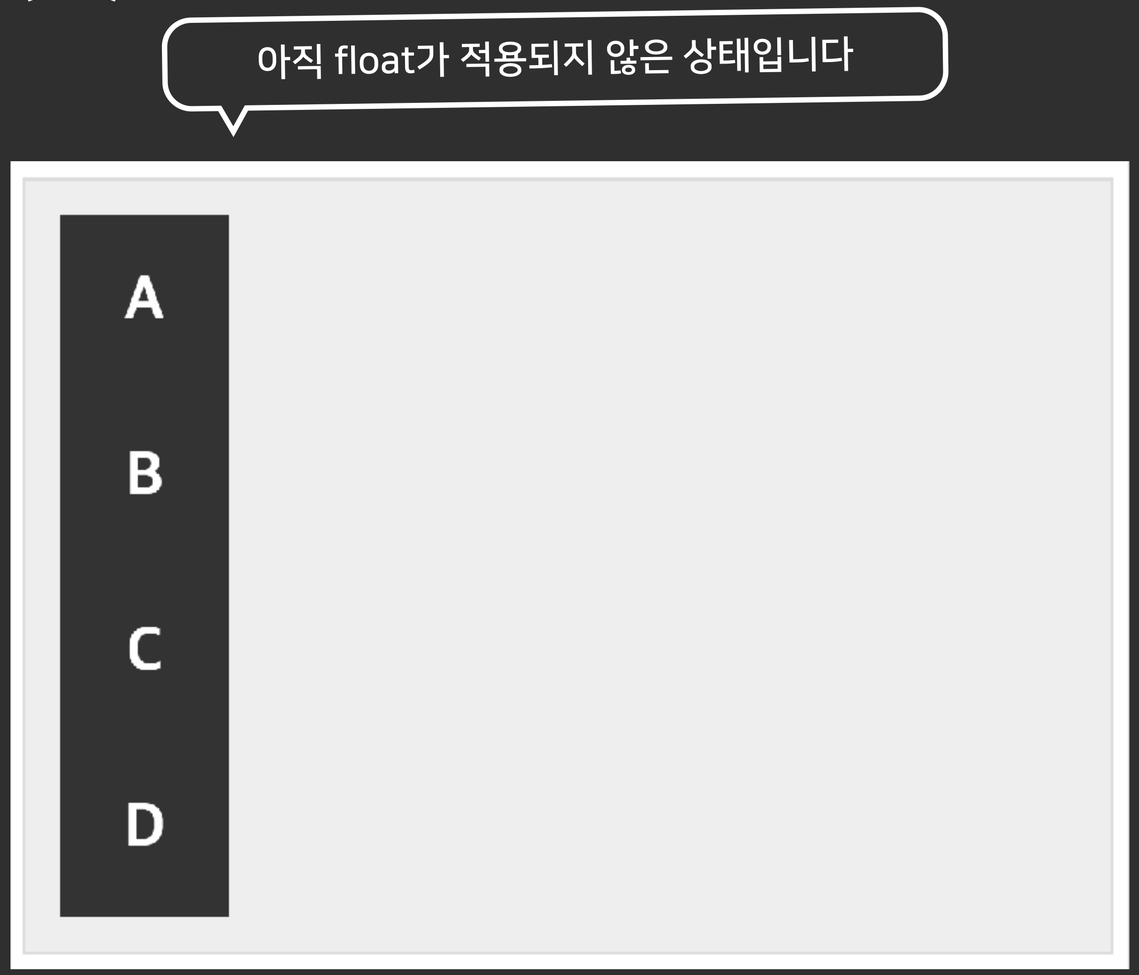
HTML

```html
<div class="float-frame">
  <div class="float-unit">A</div>
  <div class="float-unit">B</div>
  <div class="float-unit">C</div>
  <div class="float-unit">D</div>
</div>
```

CSS로 레이아웃을 구현할 때는 float속성을 자주 사용합니다.

float 레이아웃

float layout



검은 span요소에 float를 적용합니다.

float 레이아웃

float layout

CSS

```css
.float-frame {
  width: 300px;
  background-color: #eee;
  border: 1px solid #ddd;
  padding: 10px;
}

.float-unit {
  width: 50px;
  background: #333;
  color: #fff;
  font-size: 18px;
  font-weight: bold;
  text-align: center;
  padding: 15px 0;
  margin-right: 5px;
  float: left;
}
```

HTML

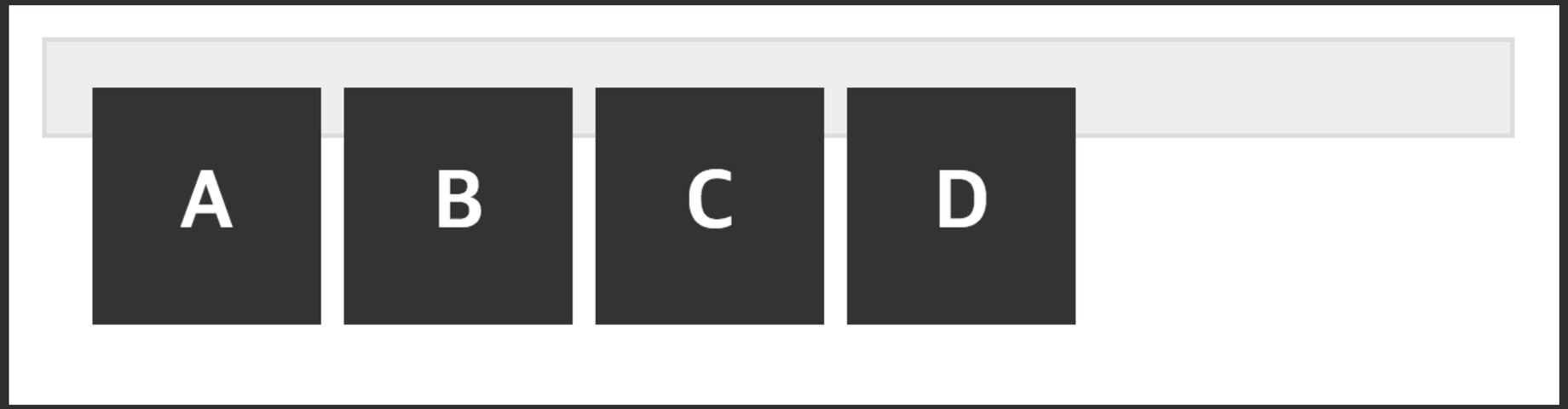
```html
<div class="float-frame">
  <div class="float-unit">A</div>
  <div class="float-unit">B</div>
  <div class="float-unit">C</div>
  <div class="float-unit">D</div>
</div>
```

float속성을 추가합니다

float 레이아웃

float layout

항목들은 float:left;를 따라 잘 배열됐지만, 부모요소의 height가 인식되지 않습니다



float 레이아웃 - 임의의 요소 삽입

float를 해제할 임의의 요소 삽입

CSS

```css
.float-frame {
  width: 300px;
  background-color: #eee;
  border: 1px solid #ddd;
  padding: 10px;
}

.float-unit {
  width: 50px;
  background: #333;
  color: #fff;
  font-size: 18px;
  font-weight: bold;
  text-align: center;
  padding: 15px 0;
  margin-right: 5px;
  float: left;
}
```

HTML

```html
<div class="float-frame">
  <div class="float-unit">A</div>
  <div class="float-unit">B</div>
  <div class="float-unit">C</div>
  <div class="float-unit">D</div>
  <br style="clear: both;">
</div>
```

float 레이아웃 - overflow를 사용

부모에 overflow를 설정

CSS

```css
.float-frame {
  width: 300px;
  background-color: #eee;
  border: 1px solid #ddd;
  padding: 10px;
  overflow: auto;
  overflow: hidden;
}

.float-unit {
  width: 50px;
  background: #333;
  color: #fff;
  font-size: 18px;
  font-weight: bold;
  text-align: center;
  padding: 15px 0;
  margin-right: 5px;
  float: left;
}
```

HTML

```html
<div class="float-frame">
  <div class="float-unit">A</div>
  <div class="float-unit">B</div>
  <div class="float-unit">C</div>
  <div class="float-unit">D</div>
</div>
```

overflow:hidden은 overflow:visible때는 인식하지 못하던 float요소와, 자식요소의 margin값을 인식하게 됩니다.

float 레이아웃 - 부모에도 float를 설정

부모에도 float를 설정

CSS

```css
.float-frame {
  width: 300px;
  background-color: #eee;
  border: 1px solid #ddd;
  padding: 10px;
  float: left;
}

.float-unit {
  width: 50px;
  background: #333;
  color: #fff;
  font-size: 18px;
  font-weight: bold;
  text-align: center;
  padding: 15px 0;
  margin-right: 5px;
  float: left;
}
```

HTML

```html
<div class="float-frame">
  <div class="float-unit">A</div>
  <div class="float-unit">B</div>
  <div class="float-unit">C</div>
  <div class="float-unit">D</div>
</div>
```

overflow:hidden은 overflow:visible때는 인식하지 못하던 float요소와, 자식요소의 margin값을 인식하게 됩니다.

CSS 포지션

CSS position

CSS의 요소의 위치를 설정합니다

position

요소의 위치를 지정합니다

기본값

static

static과 동일하게 작동

relative

문서 전체를 기준으로 위치

fixed

자신과 가장 가까운
‘position’이 ‘static’이 아닌 부모를 기준
(없을 경우 본문(body)기준)

absolute

position

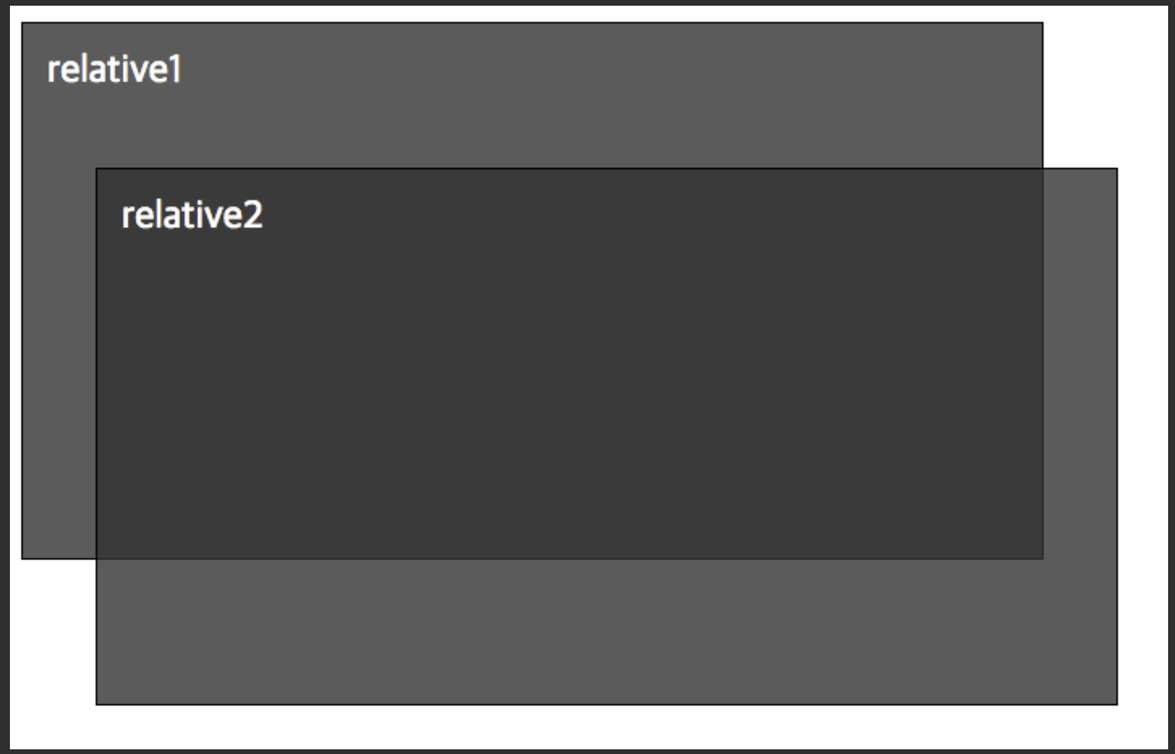
relative

HTML

```
<div class="relative1">
  relative1
  <div class="relative2">relative2</div>
</div>
```

CSS

```
div {
  width: 400px;
  height: 200px;
  padding: 10px;
  border: 1px solid black;
  color: white;
  background-color: rgba(50,50,50,0.8);
}
.relative1 {
  position: relative;
}
.relative2 {
  position: relative;
  top: 30px;
  left: 20px;
}
```



relative 포지션은 자신의 위치를 기준으로 정렬됩니다

position

fixed

HTML

```html
<div class="fixed">fixed element</div>
```

CSS

```css
.fixed {
  position: fixed;
  width: 100px;
  height: 50px;
  background-color: blue;
  right: 10px;
  bottom: 10px;
}
```

fixed포지션은 뷰포트(표시영역)를 기준으로 정렬됩니다

position

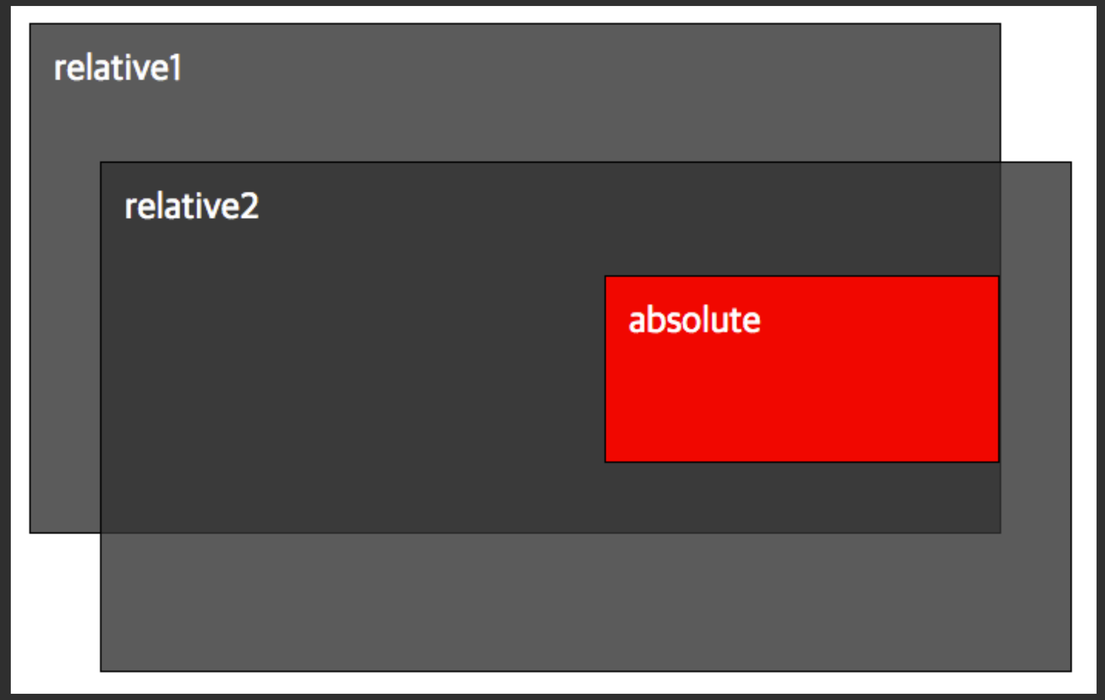
absolute

HTML

```
<div class="relative1">
  relative1
  <div class="relative2">relative2</div>
  <div class="absolute">absolute</div>
</div>
<div class="fixed">fixed element</div>
```

CSS

```
.absolute {
  position: absolute;
  width: 150px;
  height: 60px;
  background-color: red;
  color: white;
  right: 0px;
  bottom: 30px;
}
```



absolute포지션은 static이 아닌
가장 가까운 부모를 기준으로 정렬됩니다

CSS 가운데 정렬

CSS Center positioning

CSS를 사용해서 레이아웃을 가운데 정렬하는 법을 배웁니다

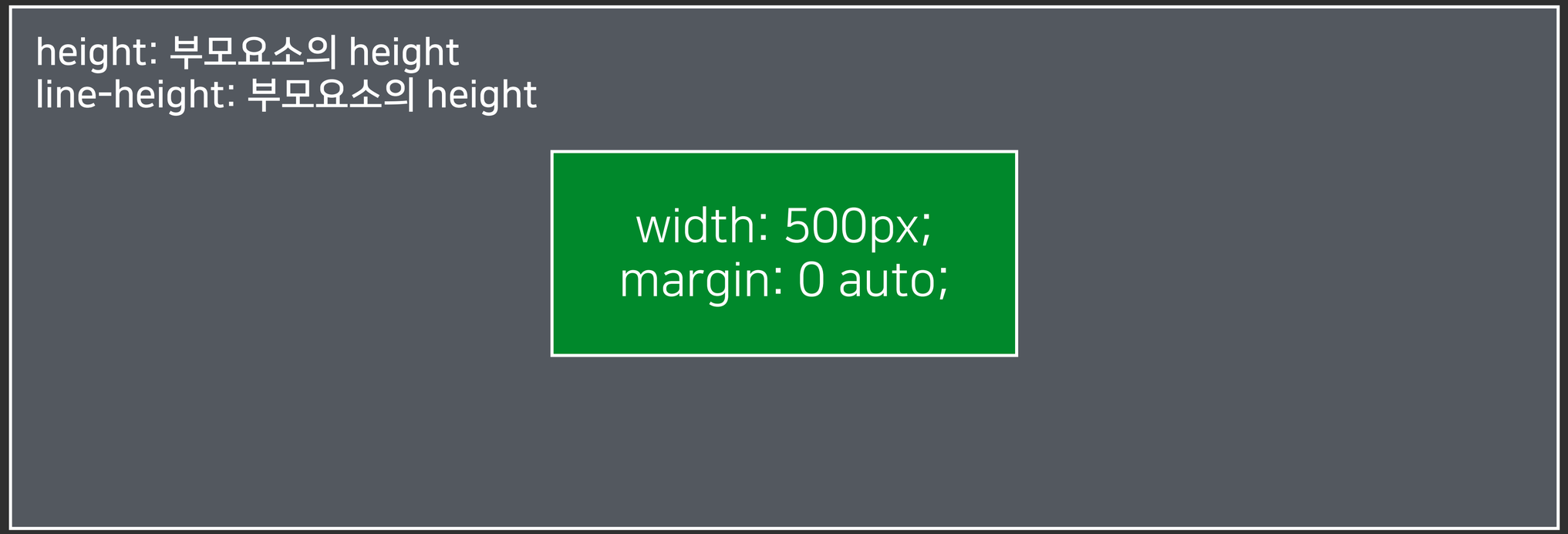
가로 가운데 정렬

부모의 가운데로 정렬하는 법

전체 크기가 정해져 있지 않을 경우, 내용의 width만 지정한 후 좌/우 여백은 auto로 같게 처리해줍니다

가로/세로 가운데 정렬

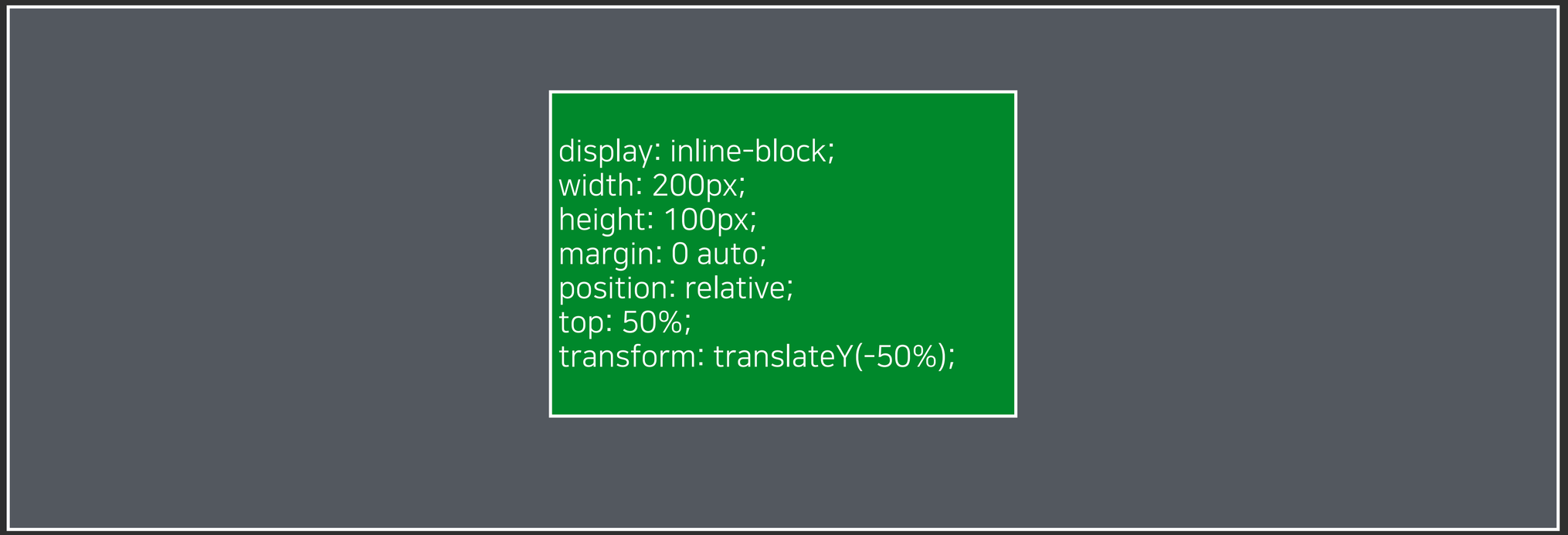
부모의 가로/세로 가운데 정렬하는 법



부모요소의 height와 line-height의 값이 같을 경우, 내부의 요소들은 세로 가운데로 정렬됩니다

가로/세로 가운데 정렬

부모의 가로/세로 가운데 정렬하는 법



요소를 inline-block속성으로 변경한 뒤, 상단에서 50%만큼 내린 후 자신의 높이의 -50%만큼을 다시 위로 올립니다.