

# CSS 배경 스타일

CSS Background

CSS로 요소(Element)에 배경을 지정합니다

# 배경색

background-color

```
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;
```

반복하지 않음

```
background-repeat: repeat;
```

가로세로 바둑판 형식으로 반복

```
background-repeat: repeat-x;
```

가로(x축)으로만 반복

```
background-repeat: repeat-y;
```

세로(y축)으로만 반복

```
}
```

가로로 반복하는 이미지의 경우, 세로로 길고 가로 1px인 이미지를 이용해 배경을 나타낼 수 있으며, 세로반복의 경우 반대로 가로로 길고 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, y축의 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

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

image

repeat

attachement

position

color

지금까지 배운 background관련 속성을 한 번에 줄여쓸 수 있습니다.  
순서대로 image, repeat, attachment, position(x,y), color값을 나타냅니다.



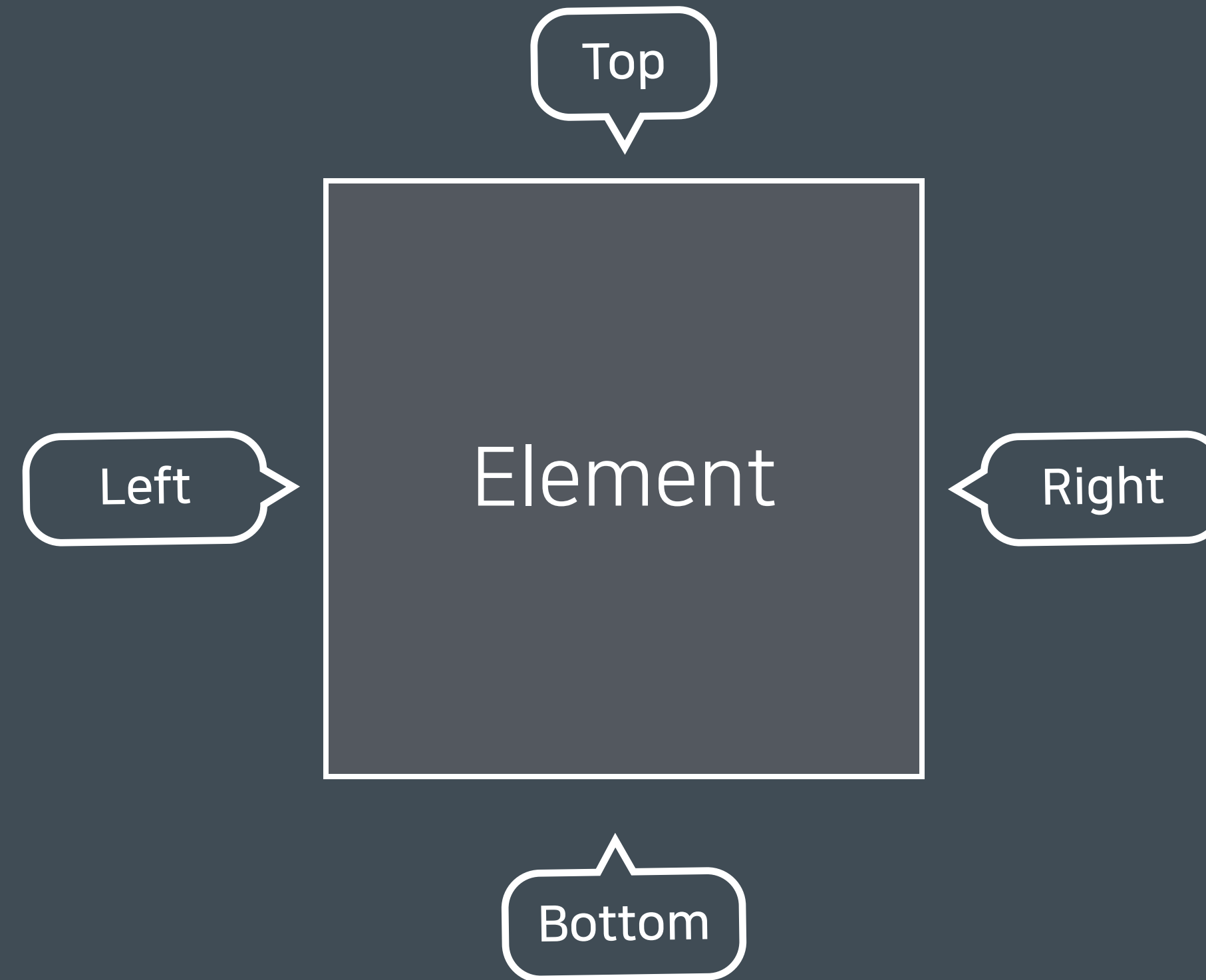
# CSS 테두리 스타일

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

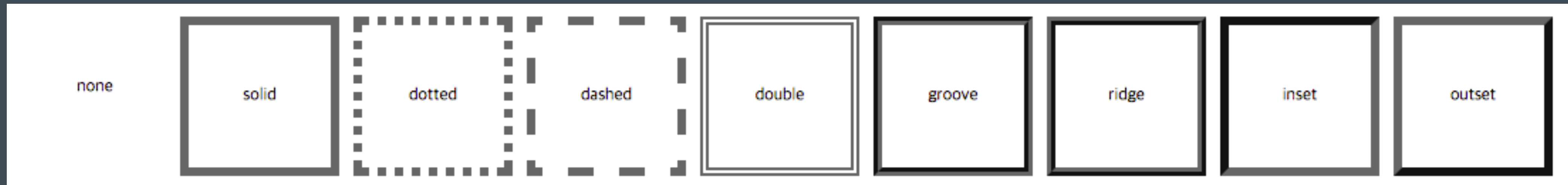
상하 5px, 좌우 10px

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.

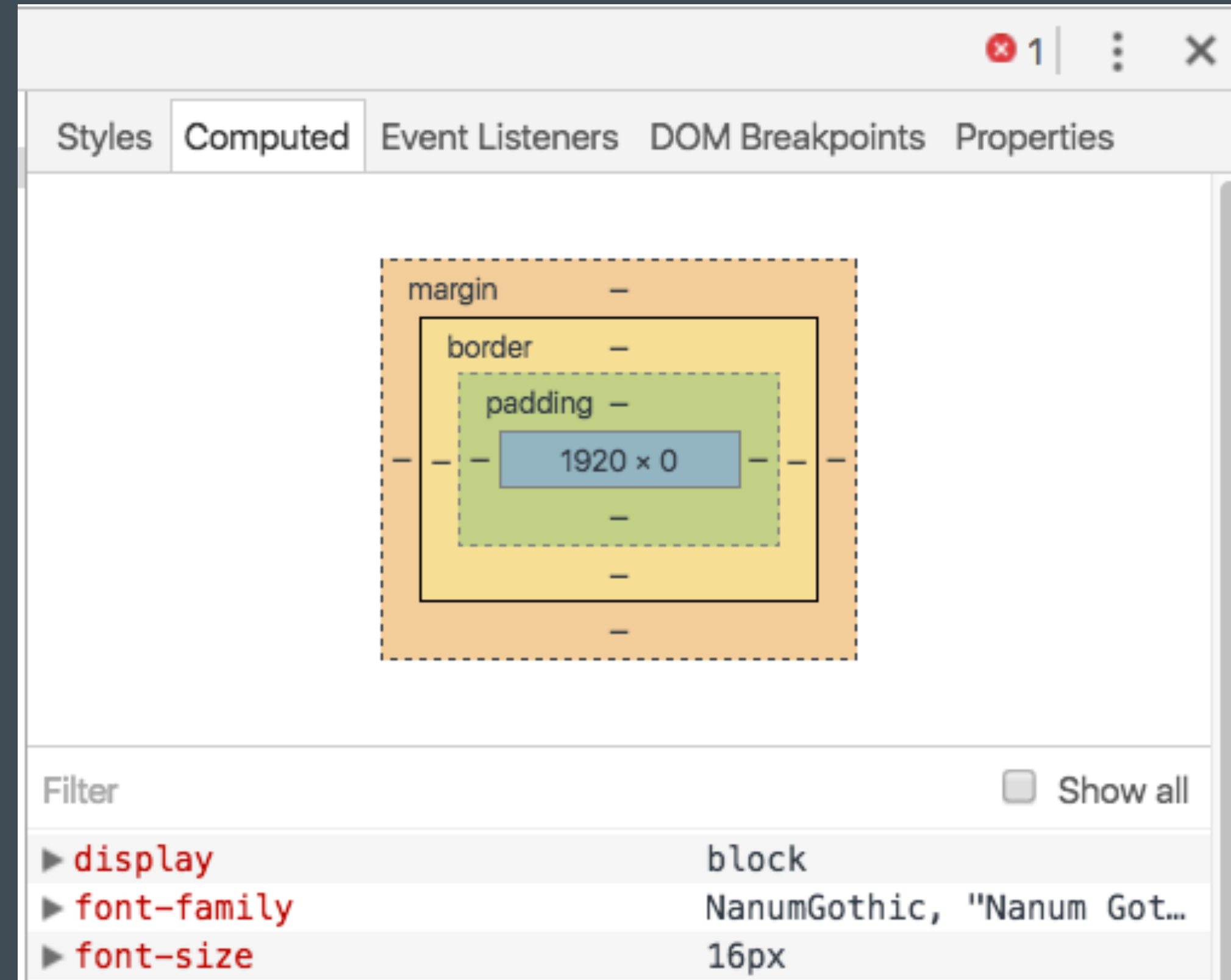
# CSS 박스 모델

CSS Box Model

CSS로 요소를 구성하는 박스에 대해 알아봅니다

# 박스모델?

CSS요소를 이루는 형태



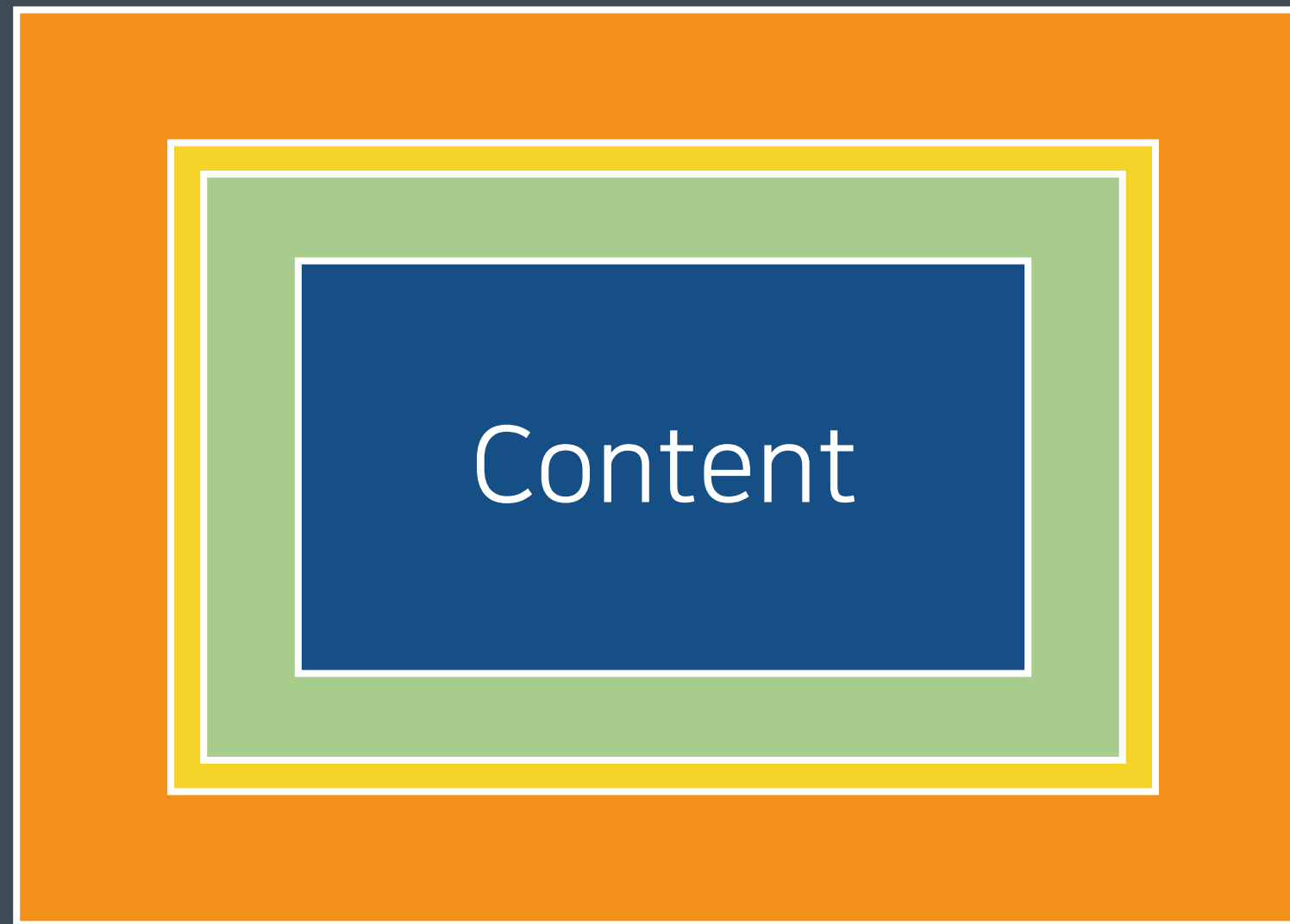
개발자 도구에서 CSS요소의 박스 모델을 확인 가능

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 요소는 가로마진만 가질 수 있습니다

Inline Element

Block Element

블록 요소는 가로/세로 마진을 모두 가집니다

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

# 바깥 여백 (마진)

margin

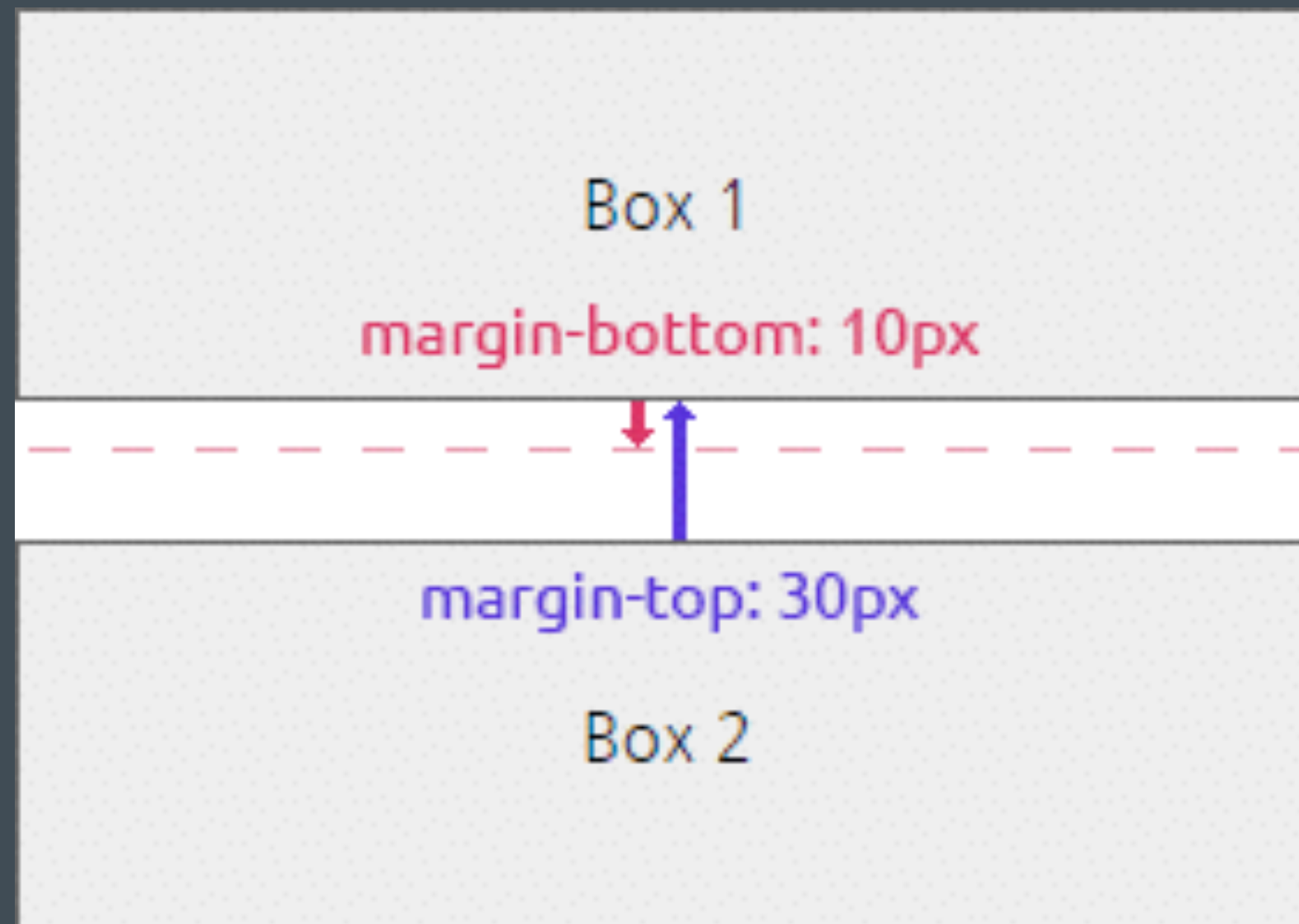
```
div {  
    margin-top: 10px;  
    margin-bottom: 30px;  
    margin: 10px 0;  
    margin: 40px 20px 30px 50px;  
}
```

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

```
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

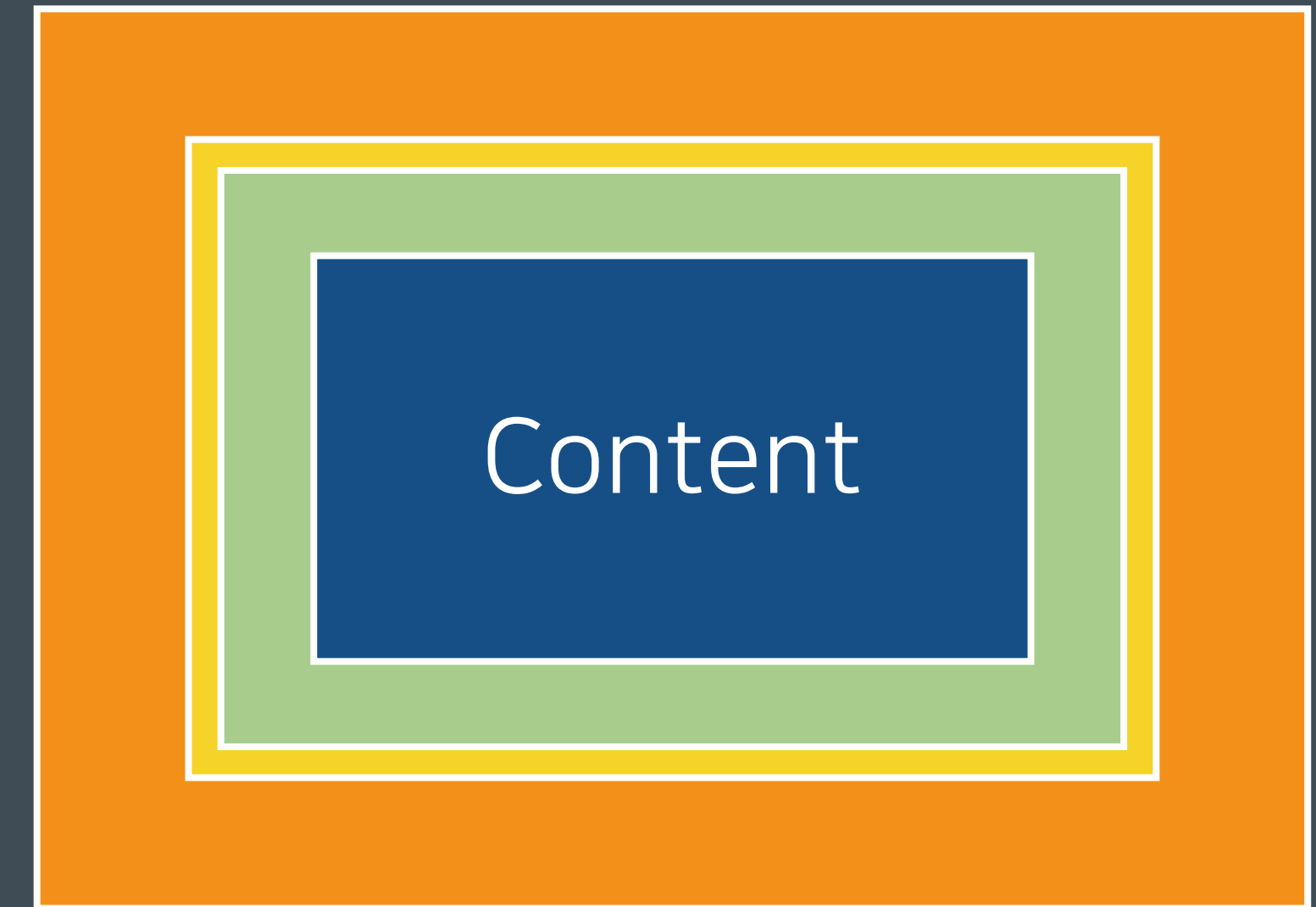
width, height

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

# 가로, 세로

width, height

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



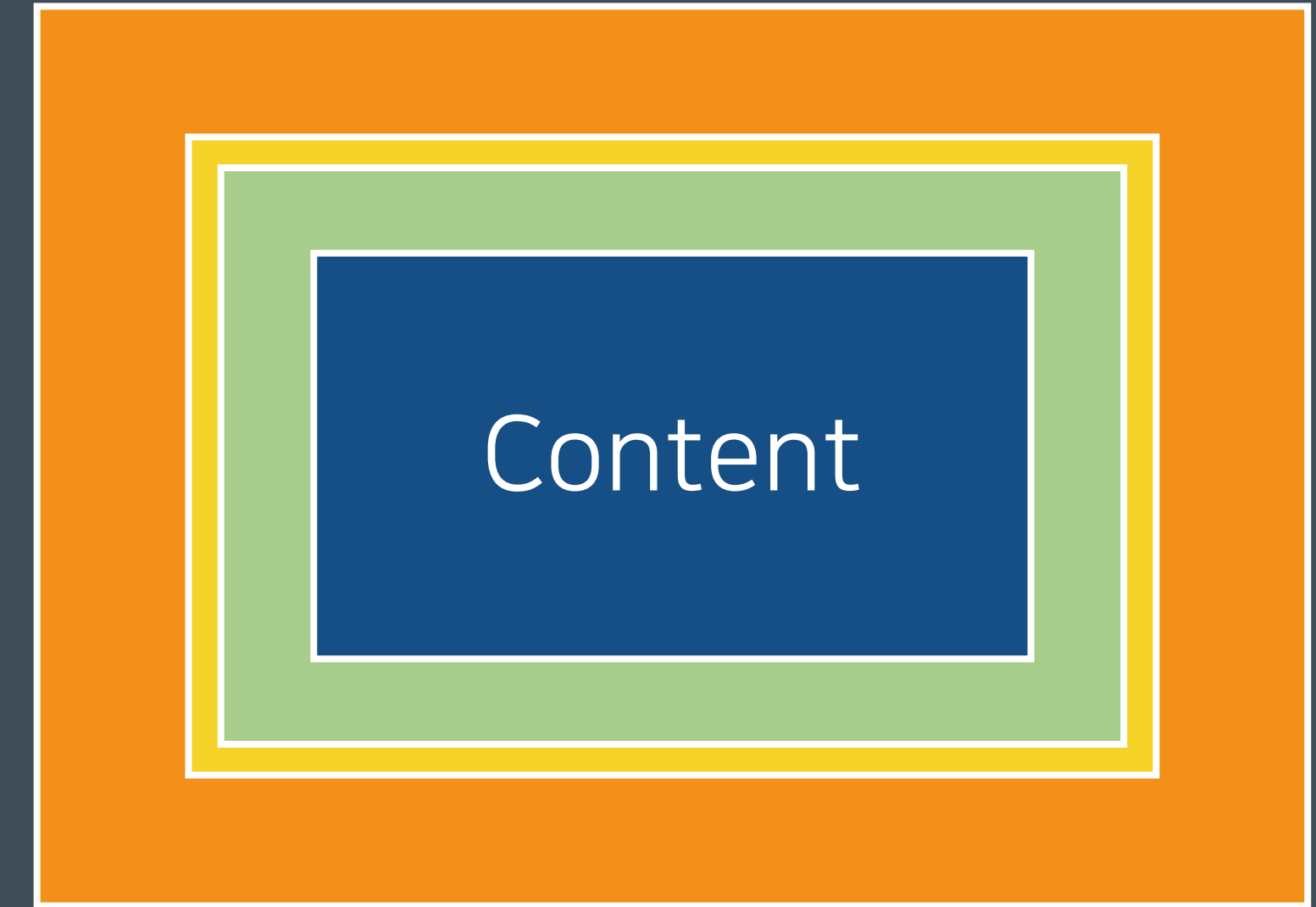
요소의 총 가로길이  
 $200\text{px}(\text{가로}) + (10\text{px} + 1\text{px} + 15\text{px}) \times 2$   
 $= 252\text{px}$

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  $200\text{px} - (10\text{px} + 1\text{px} + 15\text{px}) \times 2$   
 $= 148\text{px}$

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;  
}
```

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

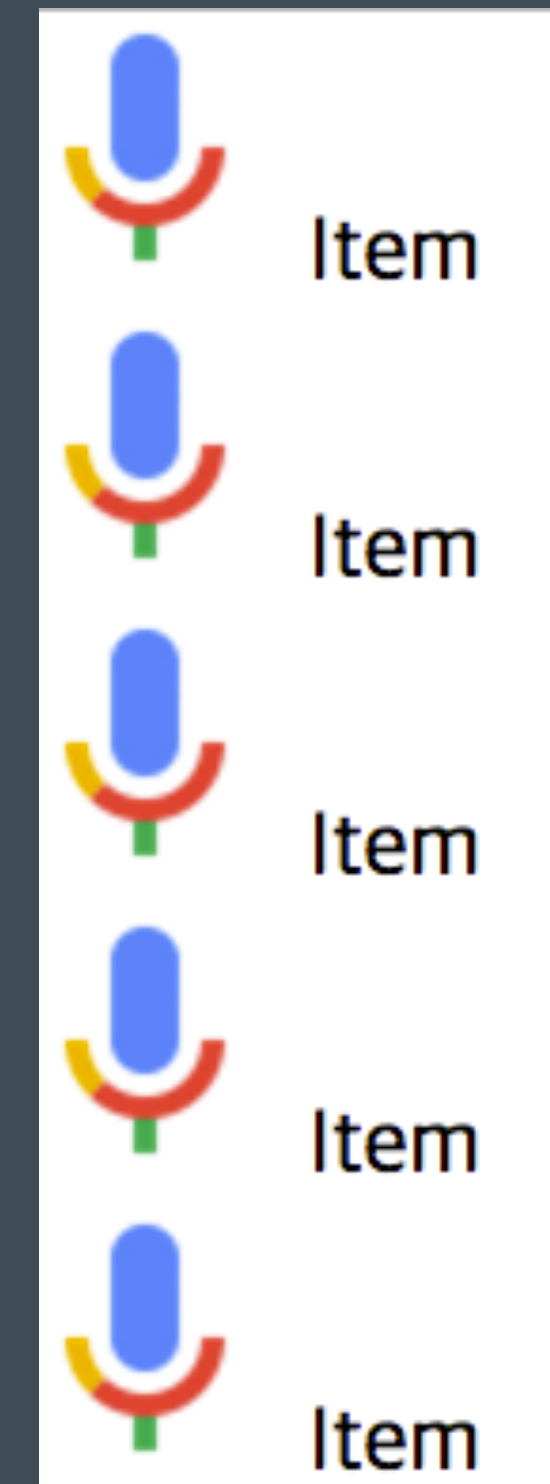
## 적용 예시

- disc 형태
  - circle 형태
  - square 형태
- 
1. decimal
  - b. lower-alpha
  - C. upper-alpha
  - iv. lower-roman
  - V. upper-roman

# 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 {
```

블릿 타입

이미지를 사용할 경우

블릿 위치 내부/외부 여부

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

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

# 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

a	b
c	

colspan 예시

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

a	b
	c
	d

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.

# 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;
}
```

A	B	C
1	2	3
4	5	6

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

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

A	B	C
1	2	3
4	5	6

border-spacing 적용

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

# 빈 셀의 표시

empty-cells

```
<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>
```

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

A	B	C
1		
4	5	

empty-cells에 show 적용

A	B	C
1		
4	5	

empty-cells에 hide 적용

\*\*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 블록 속성을 지니게 됩니다

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

기존에 블록 요소가 아닌 요소를 블록 속성을 갖도록 합니다.

# 화면 표시방법

display: inline

반대로, div가 display프로퍼티에 inline속성을 받으면  
인라인 요소로 취급됩니다

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

기존에 인라인 요소가 아닌 요소를 인라인 속성을 갖도록 합니다.



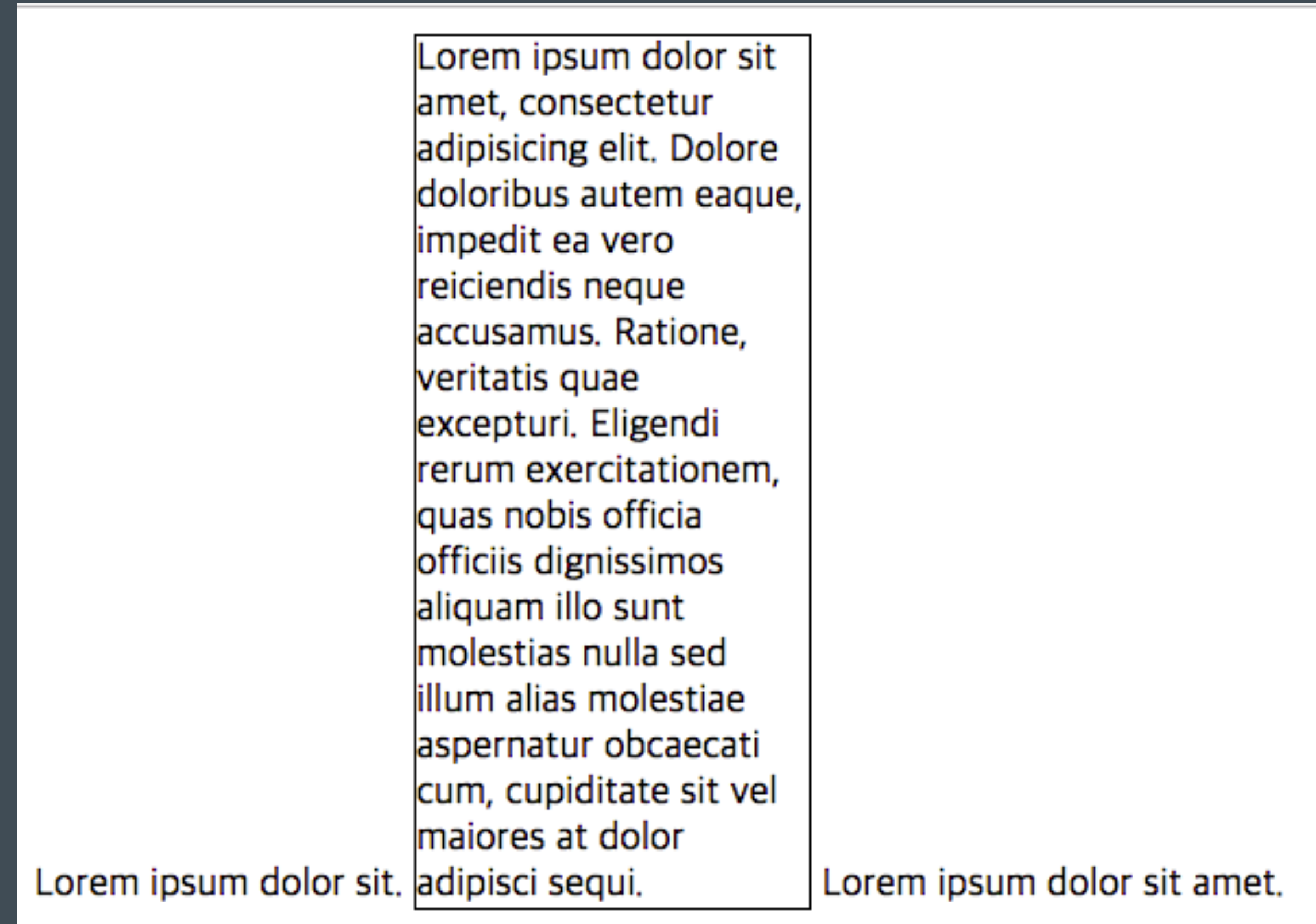
# 화면 표시방법

display: inline-block

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

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

인라인요소에 높이와 상/하 패딩, 마진값을 줄 때 사용합니다.



display 프로퍼티에 inline-block속성을 가진 span

# 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

**visible**

이 속성 값을 사용한 요소는 안에 콘텐츠들이 박스를 벗어나더라도 그대로

**scroll**

스크롤이 항상

**hidden**

당신은 이 박스를 벗어나는 내용을 볼 수 없습니다. 왜냐

**auto**

내용이 짧으면 스크롤 안보임.

**auto**

하지만 내부의 콘텐츠가 박스를 벗어나기 시작하면

높이가 고정되어있고, 내용이 길면 스크롤 할 부분에서는 auto를 사용합니다.

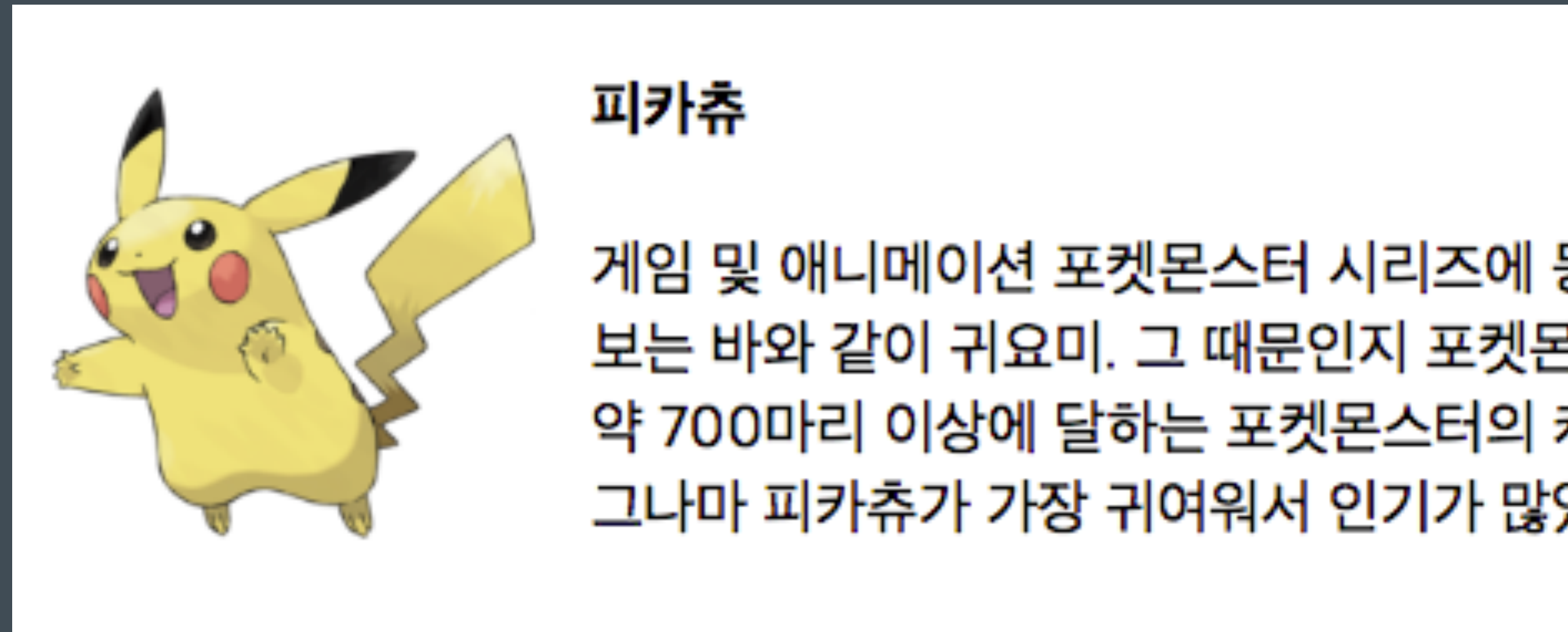
# CSS float속성

CSS float

CSS로 요소를 띄웁니다

# float속성

float: 띄우다



float속성은 중간에 이미지를 넣은 단락을 만들고자 하는 경우에도 사용가능합니다.

# float속성

float: 띄우다

float: right; 적용시

Element1

Element2

Element3



Element2

Element3

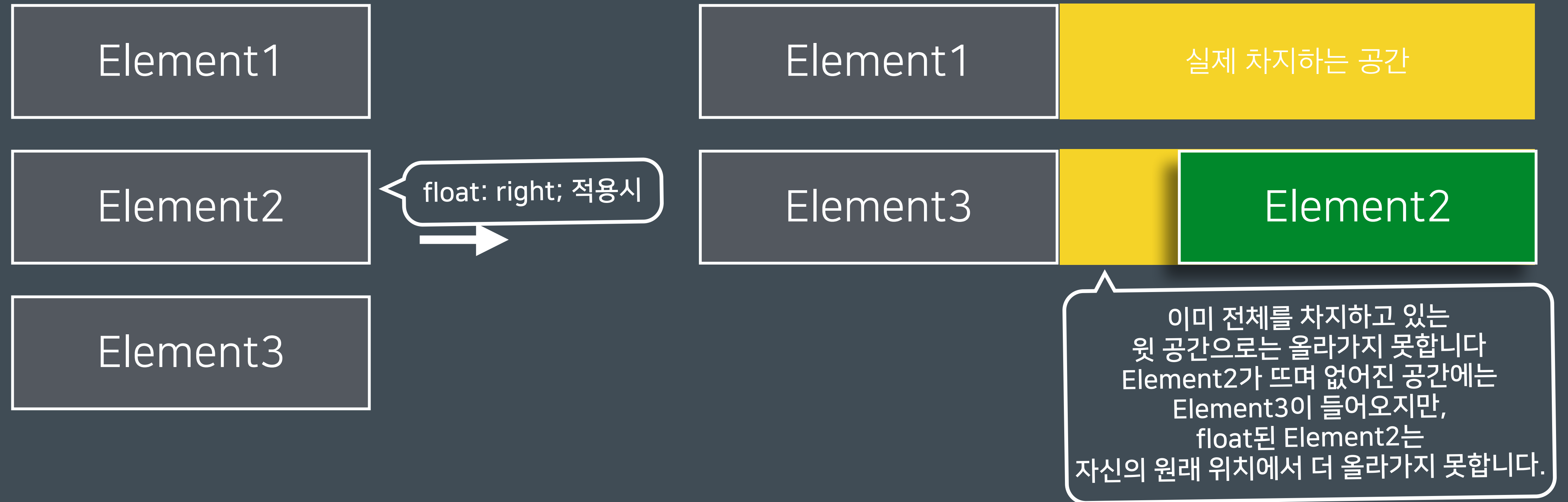
Element1

기존 항목들보다 위쪽에 떠서  
Element2가 빈 공간으로 올라옵니다

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

전체에 float:left 적용시

Element1

Element2

Element3



Element1

Element1

Element1

모든 요소가 left로 띄며 block속성을 잃고  
차례대로 배치됩니다

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

# float속성

float: 띄우다

float:left

float:left

Element1

Element2

Element3

float:right

우측부터 right속성이 쌓입니다

Element1

Element1

Element1

왼쪽부터 left속성끼리 차례로 정렬되며

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

# float속성

float: 띄우다

전체에 float:left 적용시

Element1

Element2

Element3



부모 엘리먼트보다 가로가 길 경우, 아래로 내려갑니다

Element1

Element2

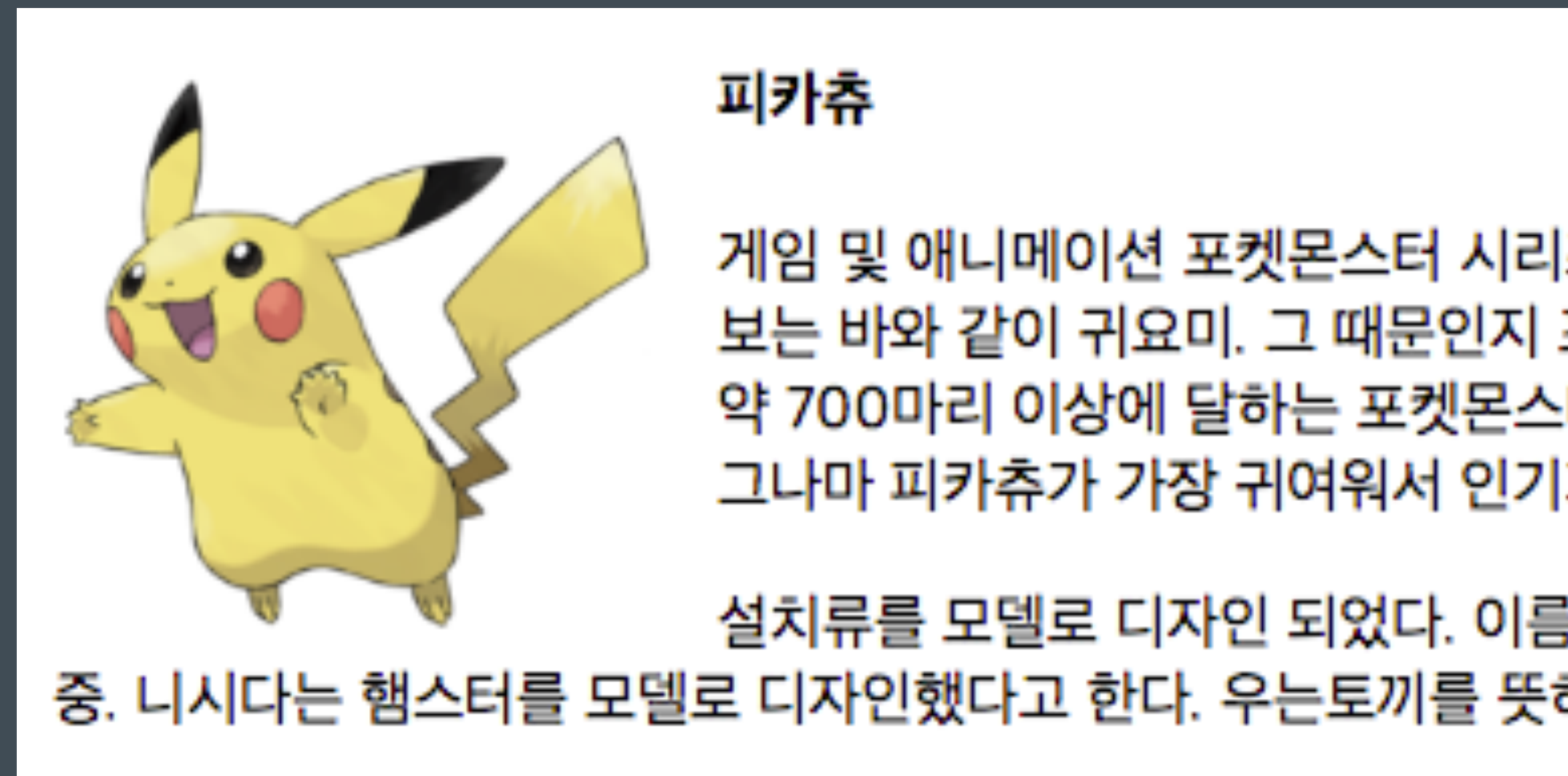
Element3

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

# clear 속성

float요소와 겹치는 경우, float속성을 해제합니다

float:left

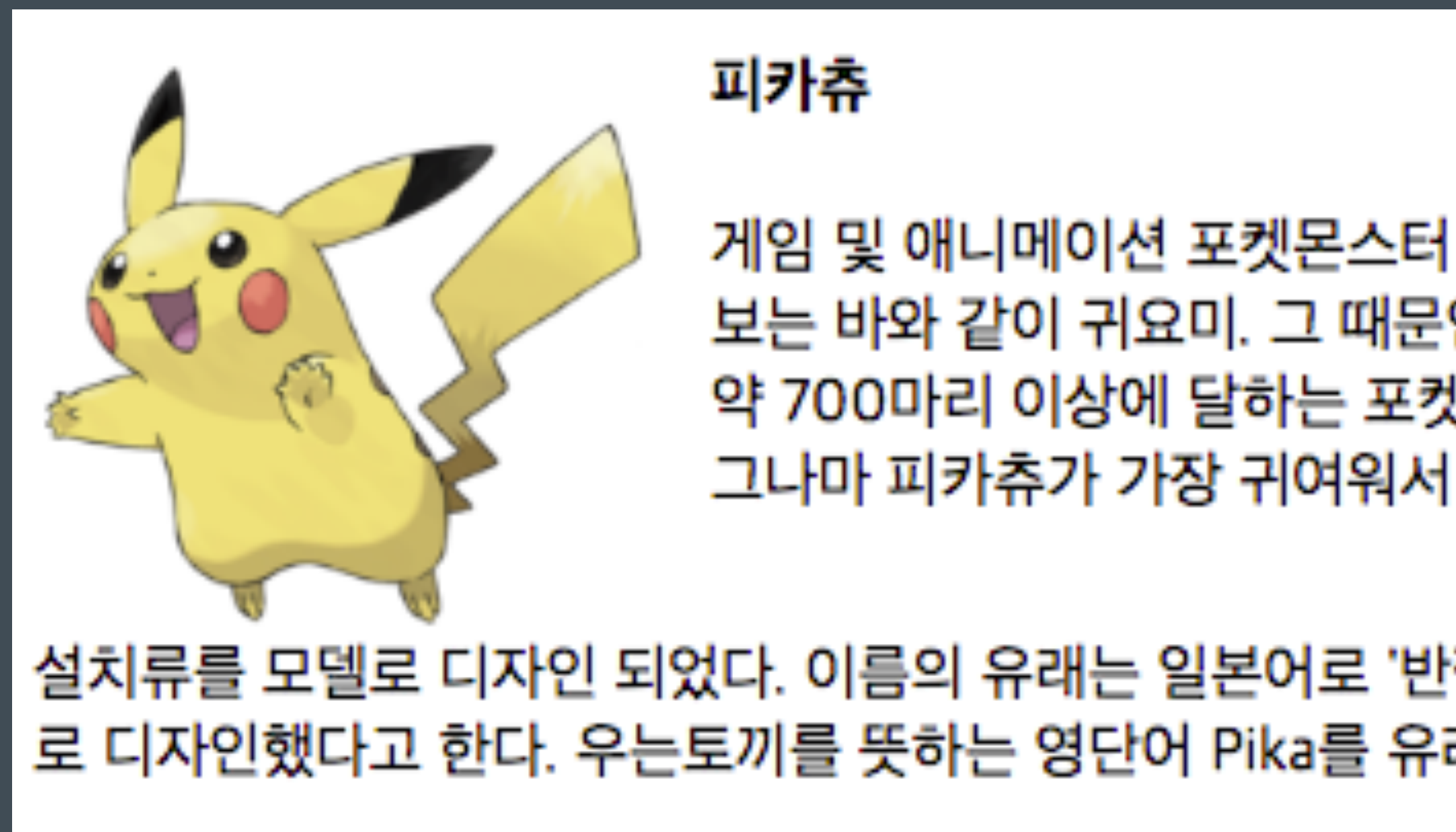


두 번째 문단은 이미지의 float에 관계없이 block처리되어 아래로 내려가도록 하려면?

# 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

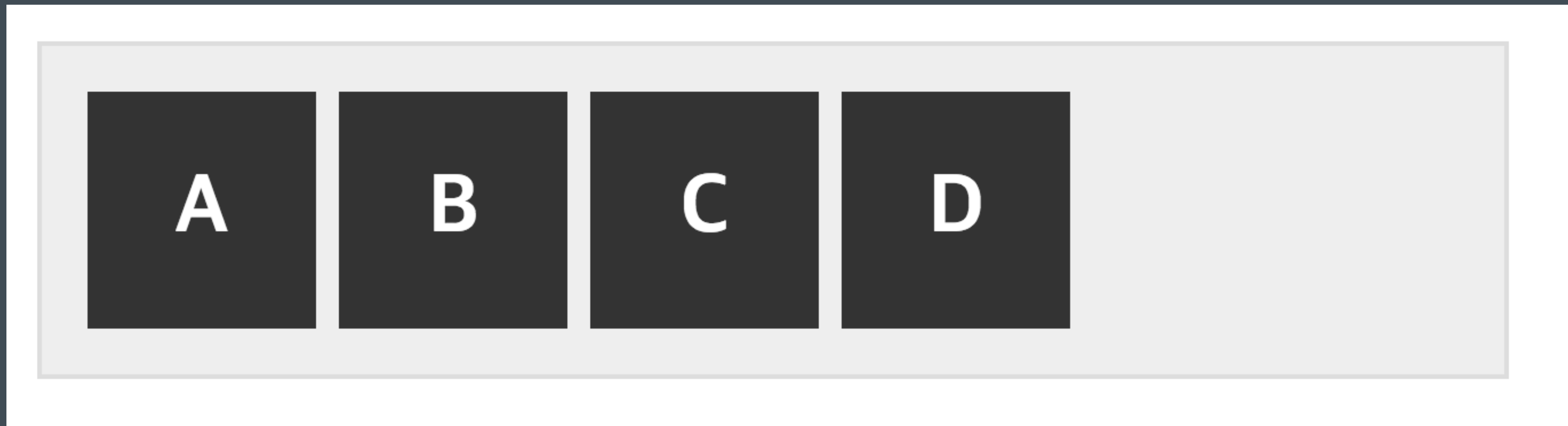
CSS float layout

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

```
.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

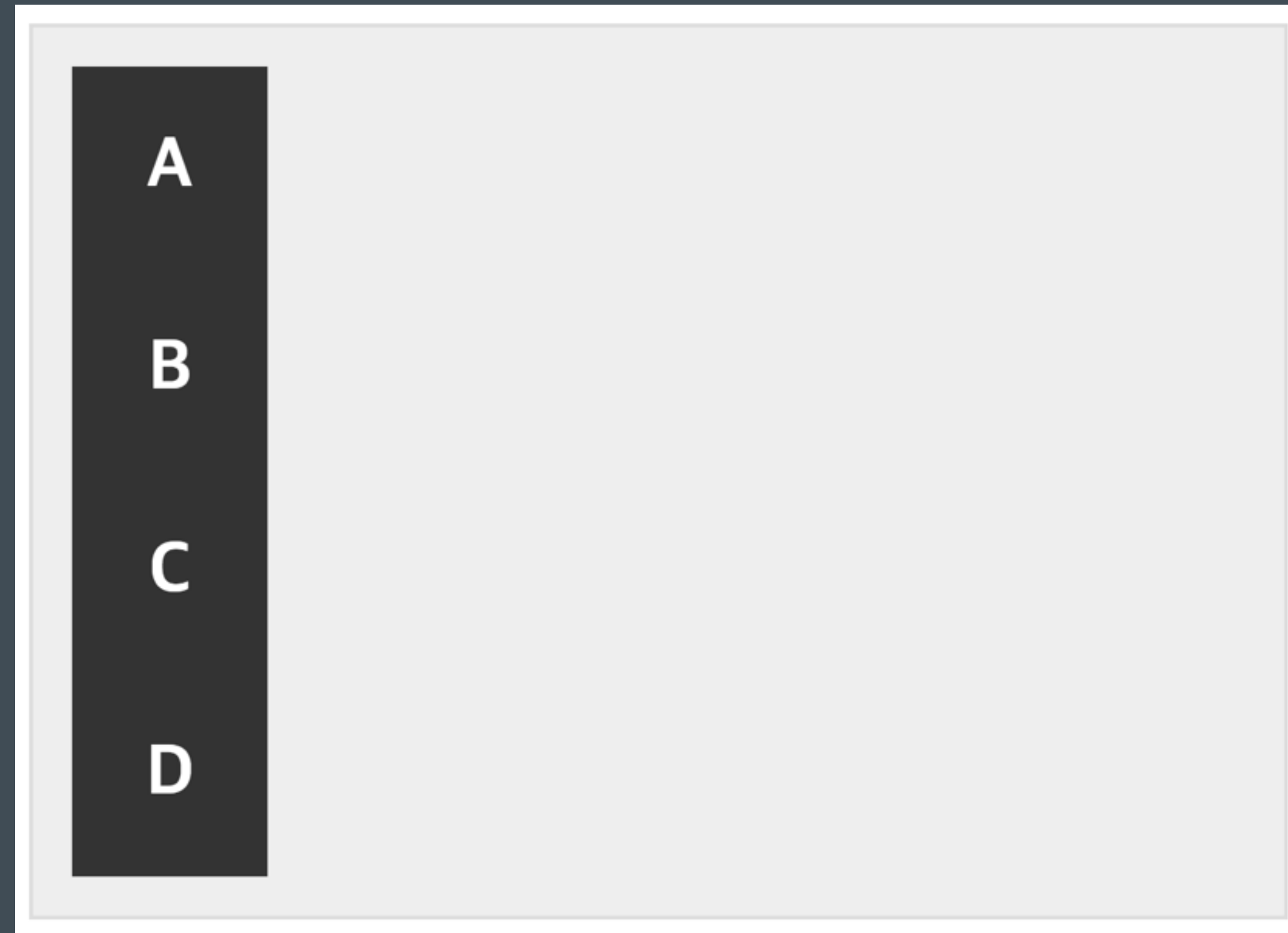
```
<div class="float-frame">  
  <div class="float-unit">A</div>  
  <div class="float-unit">B</div>  
  <div class="float-unit">C</div>  
  <div class="float-unit">D</div>  
</div>
```

CSS로 레이아웃을 구현할 때는 float속성을 자주 사용합니다.

# float 레이아웃

float layout

아직 float가 적용되지 않은 상태입니다



검은 span요소에 float를 적용합니다.

# float 레이아웃

float layout

## 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

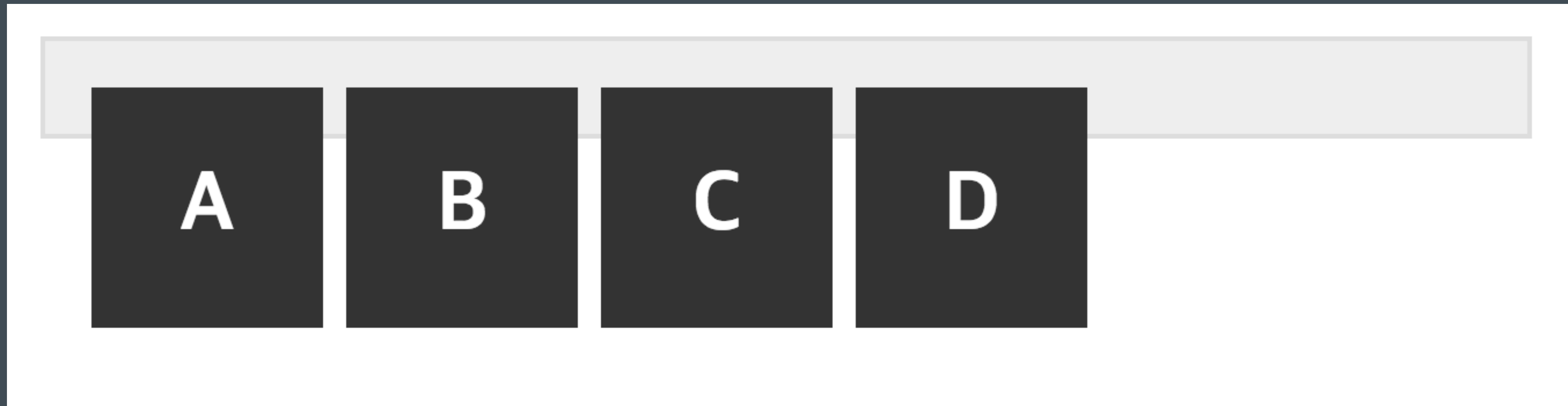
```
<div class="float-frame">  
  <div class="float-unit">A</div>  
  <div class="float-unit">B</div>  
  <div class="float-unit">C</div>  
  <div class="float-unit">D</div>  
</div>
```

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

```
.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

```
<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

```
.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

```
<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

```
.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

```
<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

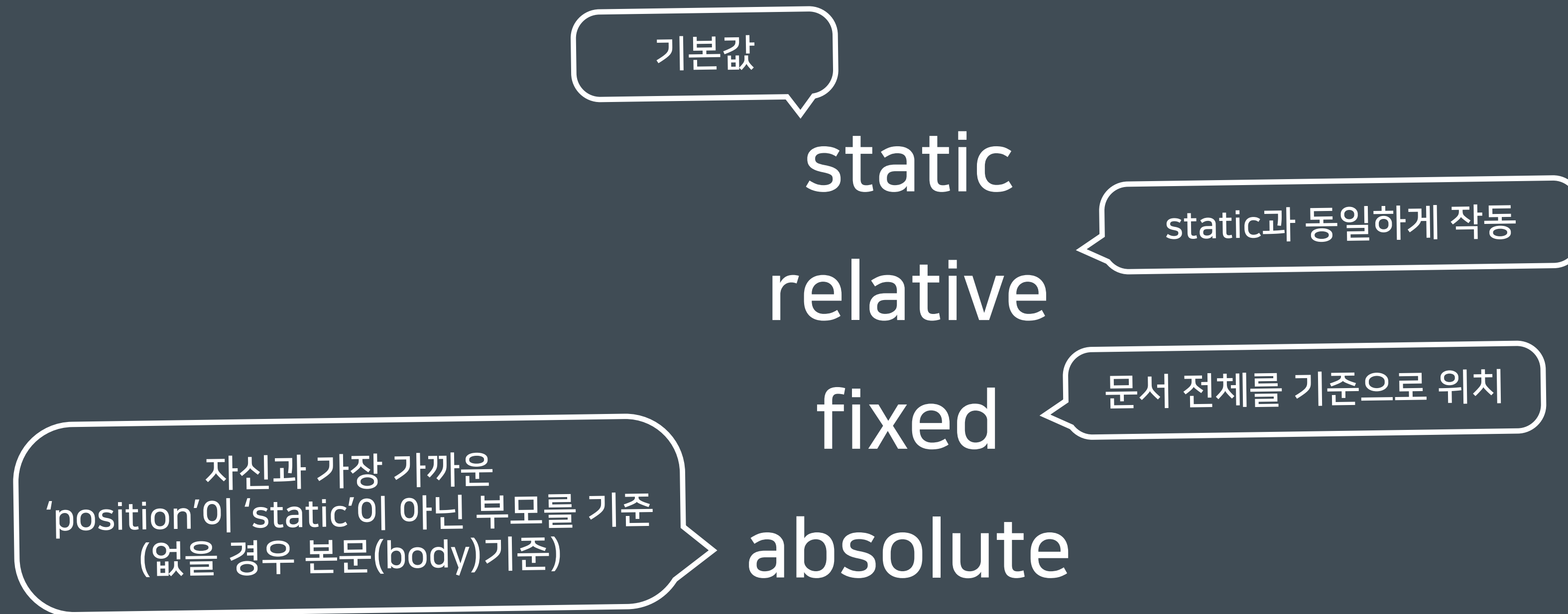
CSS position

CSS의 요소의 위치를 설정합니다



# position

요소의 위치를 지정합니다



# 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

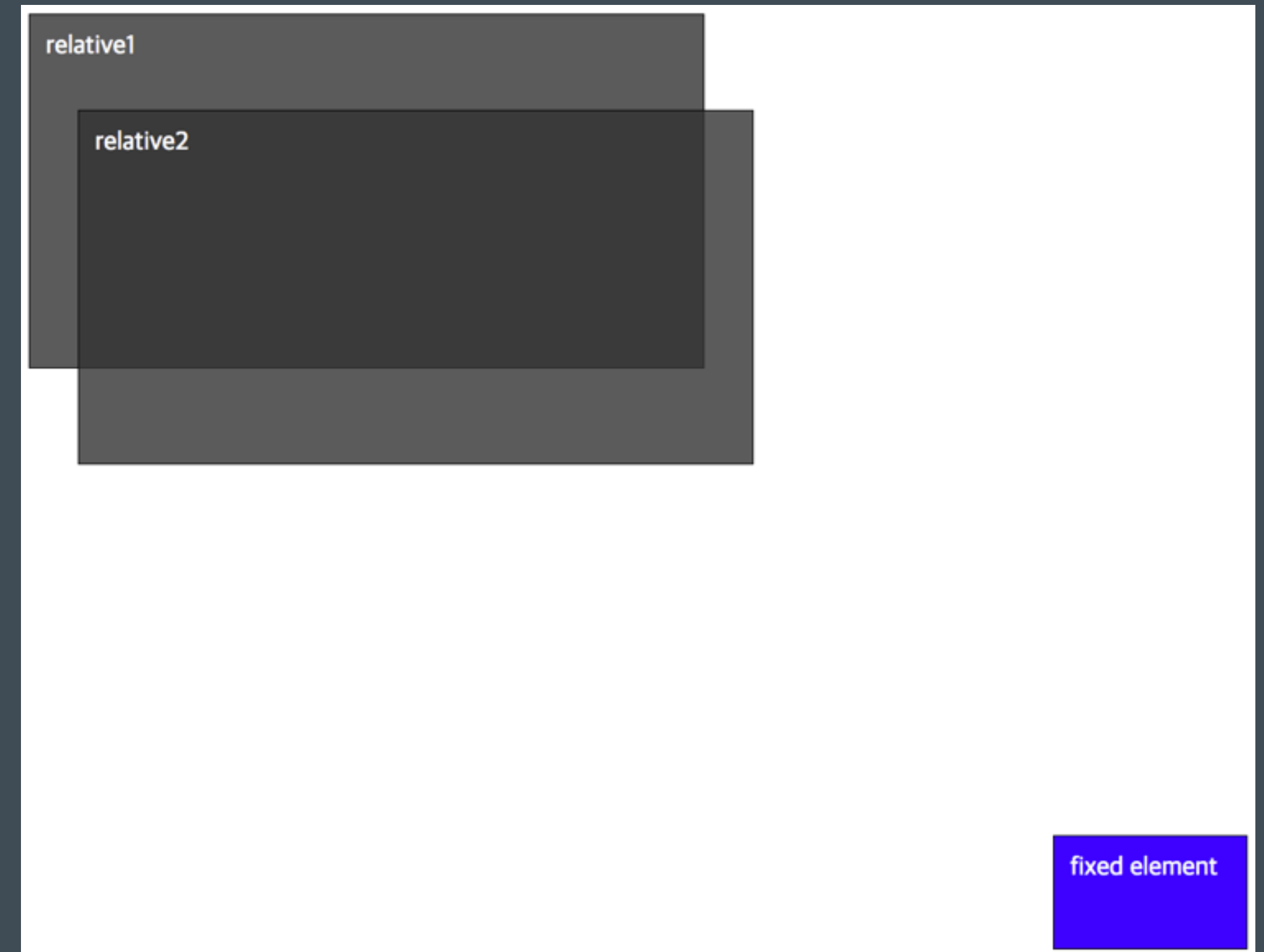
# position

fixed

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

CSS

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



fixed포지션은 뷰포트(표시영역)를 기준으로 정렬됩니다

# position

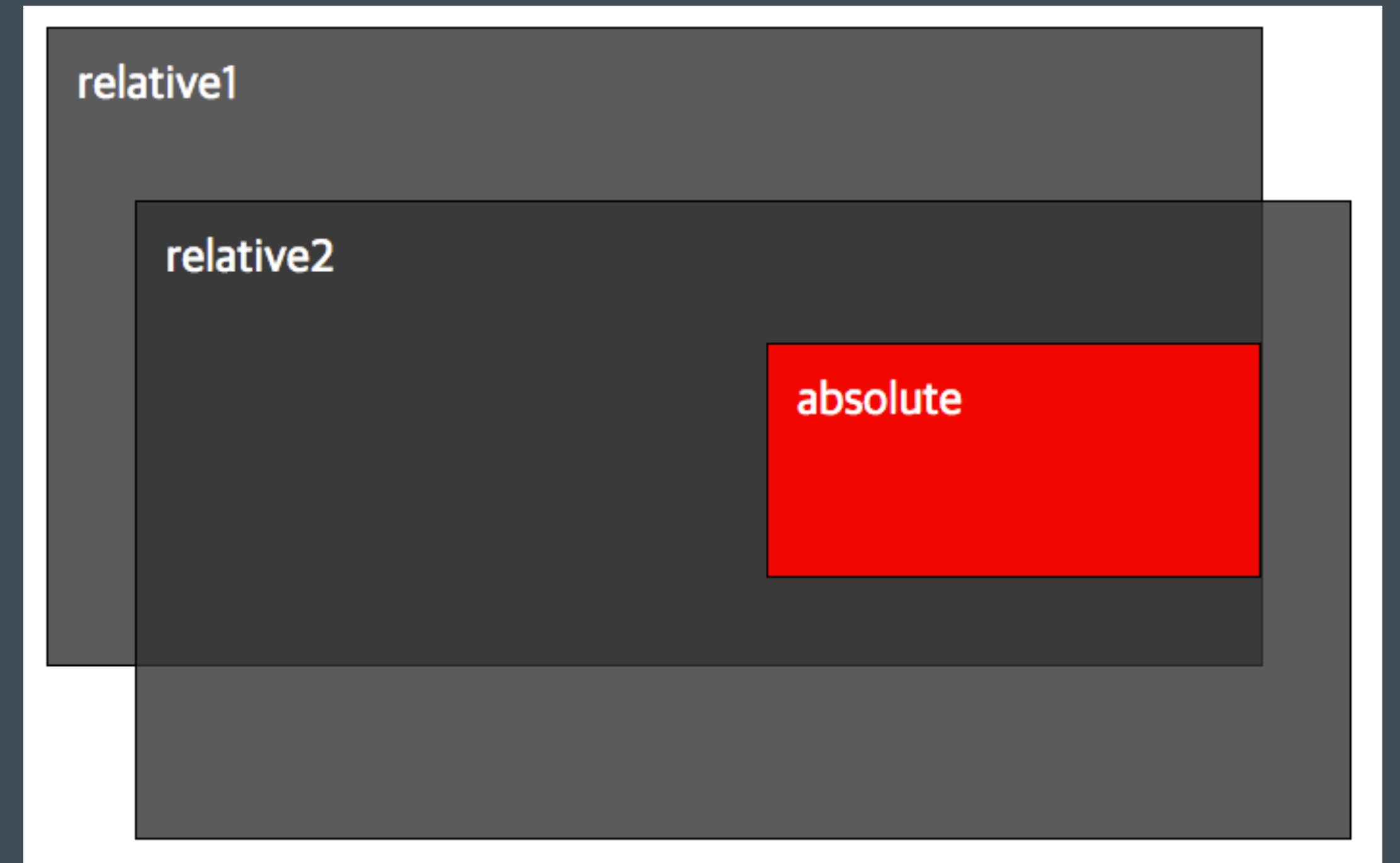
absolute

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

height: 부모요소의 height  
line-height: 부모요소의 height

width: 500px;  
margin: 0 auto;

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

# 가로/세로 가운데 정렬

부모의 가로/세로 가운데 정렬하는 법

```
display: inline-block;  
width: 200px;  
height: 100px;  
margin: 0 auto;  
position: relative;  
top: 50%;  
transform: translateY(-50%);
```

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



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