

# 과정목표 : CSS의 기본 개념 이해

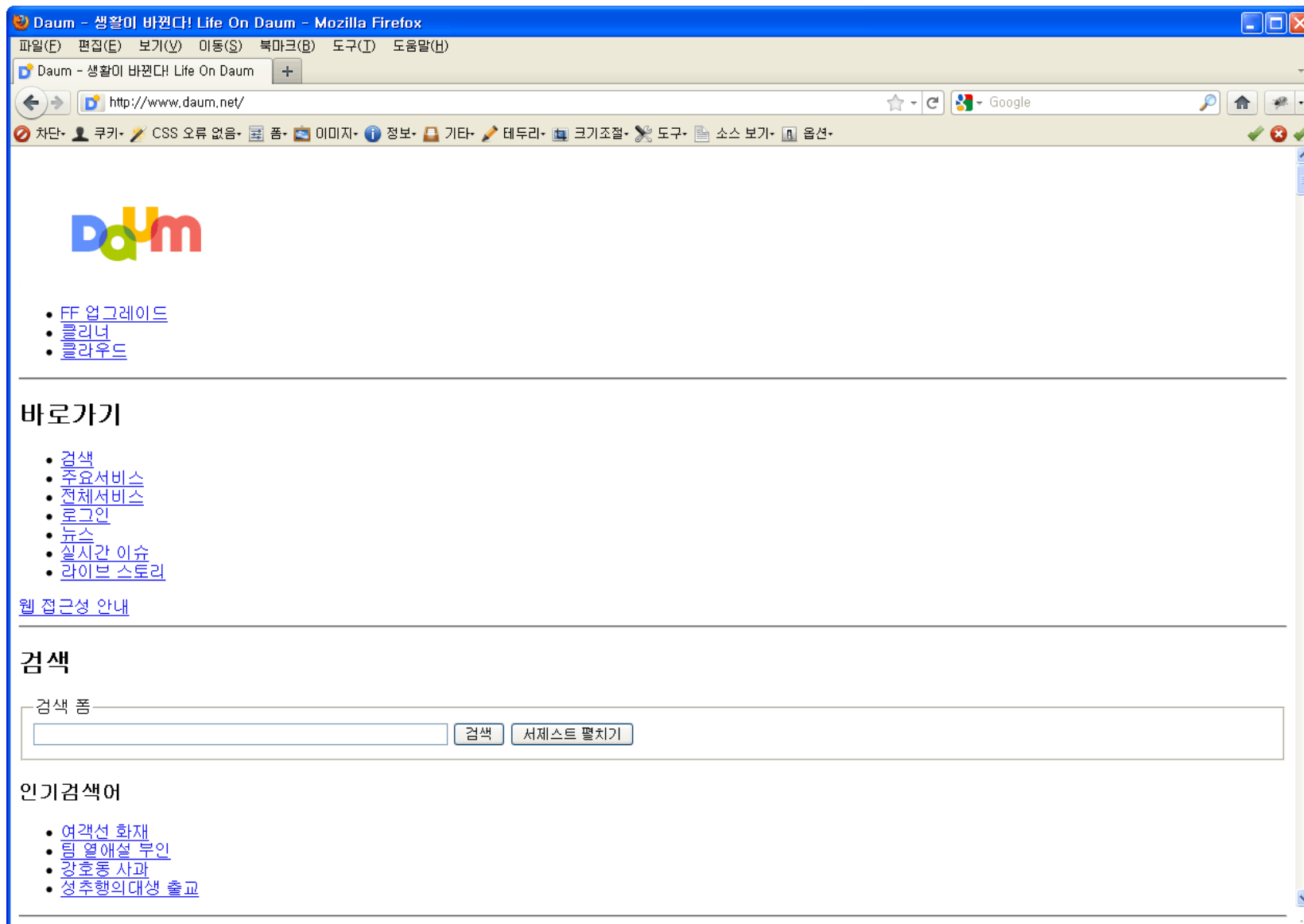
1. CSS 란?
2. CSS 선택자
3. CSS 속성

# 1. CSS 란?

# CSS CASCADING STYLE SHEET

- 스타일 선언이 위에서 아래로 순차적으로 적용이 되고, 마지막에 선언된 스타일이 우선순위 갖음
- 마크업 언어(HTML/XHTML) 가 실제 화면에 표시되는 방법을 기술하는 언어
- HTML문서의 레이아웃과 스타일을 정의한다.
- W3C에서 표준안을 만들어 내고 있으며 1996년 CSS Level 1 이후, 현재 CSS Level 2.1에서 CSS Level 3까지 보급되고 있음

# 1. CSS 란?



# 1. CSS 란?

Daum - 생활이 바뀐다! Life On Daum - Mozilla Firefox

파일(F) 편집(E) 보기(V) 이동(S) 북마크(B) 도구(T) 도움말(H)

Daum - 생활이 바뀐다! Life On Daum

http://www.daum.net/

차단 쿠키 CSS 오류 없음 폰트 이미지 정보 기타 테두리 크기조절 도구 소스 보기 옵션

Firefox 업그레이드 클러니 클라우드 50GB

게임 루리웹 티스토리 키즈맘

Daum

욕주현 눈물 | 김아중 탈세 | 최정윤 열애 | 한현정 공식입장

마이피플 PC! 일반폰도 OK!

카페 메일 블로그 쇼핑 소셜 | 뉴스 증권 부동산 사진 | 영화 뮤직 만화 tv팟 아고라 | 지도 모바일 더보기

보안 1 단계 1 2 3

아이디 or 이메일 ID 저장

비밀번호 로그인

회원가입 아이디 · 비밀번호 찾기

2011. 9. 6 (화)

날씨 환율 운세 문자 TV편성표

코스피 1785.83 +0.00 주가

오늘의 TV - 나는 가수다

버비킴 1위, 중간점검 음원공개

G마켓 추석비 라있다!

뿌로로 3종팩 6,900

트렌드 검색

아이폰 스마트키보드, 놀라운 분강호동, 세무조사로 수십억 추정? 일본간 원빈 김새론, 잘 어울려 최고의 수공예품, 손재주 놀라워 공주의남자 촬영지, 바로 여기였어

뉴스 경제 스포츠 연예 Live 이슈

안철수 출마 결정의 최대 고민은 박원순 | 안으로 적극 내미는孫

아이폰5 800만화소 카메라 누가 공급? | MBC "광우병 보도 사과"

조광래, '9월 7일 골목' 다시 없다 | 욕주현 "결혼설 난 기자랑 원수돼"

펀드소송 항소심도 50억대 배상 판결

北 교란거리 100km이상 전자전 장비 개발중

현금 갈비탕 하루 130그릇씩 숨긴 업주

'백동수' 윤소이, 친부정체 알고 충격

'백두산 호랑이 숲' 조성 | [웹툰]버프더

나니 '최고 선수 꿈..호날두 비교 무의미'

1 욕주현 현미 NEW

2 고대익대생 출교

3 김지민

4 임실군수

5 부실대학명단

6 왕갈비탕

7 탈세

8 배우자 출산휴가

9 김조한 캔커피 NEW

10 기생수

11 박정현 요트 NEW

12 설봉호 화재

FUN 영화 뮤직 만화 tv팟 웹툰 책

오늘의 소셜쇼핑 11

< 8,900

엘리자베스아덴 8종 55% TMON

AK물 LG패션 오가게 iSTYLE24

오늘만쏜다 주방초특가

머디서션! 너무메쁘다

지도도나 리더 78%

스왈은반지 1위 팔레트

명품스타일 1위 쇼핀물

맥도날드 빅맥+1

청바지1위~인기쇼핀물

필아트스탈 꼭필요해

신발의계절 단히루5 9%

명품미시옷 가을캐지트

간금재입고 키엘대용량

가을+신상 에버서너리

CUCKOO

게임 아고라 미즈넷 재테크

## CSS 선언방식

- in-line 방식

### in-line 방식

```
.  
.   
.   
<body>  
  <strong style="font-size:14px; color:blue;">안녕하세요</strong>  
  <span style="color:red;">만나서 반갑습니다~</span>  
</body>  
</html>
```

- HTML Tag에 직접 style 속성 값을 지정

## CSS 선언방식

- internal 방식

### Internal 방식

```
<html>
<head>
  <style type="text/css">
    h1 { font-size: 20px; }
    strong { font-size: 16px; }
  </style>
</head>
.
```

- HTML 문서 내부에 style tag로 선언
- head tag안에 넣는것이 정석이지만, 상황에 따라 body안에도 선언이 가능

## CSS 선언방식

- external 방식

### External 방식

```
<html>
<head>
  <link rel="stylesheet" href="common.css" type="text/css" />
  <link rel="stylesheet" href="main.css" type="text/css" />
</head>
.
.
.
```

- CSS를 따로 파일로 저장하여 불러오는 방식
- 파일을 여러개 불러올 수 있기 때문에 공통/자주 사용되는 부분을 따로 저장하여 사용하기에 용이함



## CSS 수치 단위

### em, % (상대단위)

- 브라우저의 기본 글꼴을 기준.  $1em = 100\%$
- 브라우저 메뉴의 글꼴 크기 변화에 따라 유동적임

### px, pt (절대단위)

- px (픽셀,  $1px = 1dot$ ) – 모니터의 1 픽셀
- pt (포인트,  $1pt = 1/72$  인치)

- 브라우저의 기본 글꼴을 기준으로  $1em = 12pt = 16px = 100\%$
- 상대단위를 사용하면 접근성이 향상되나, 단위 환산의 어려움과 스타일 수정 시 다양한 환경에서 일관적이지 못한 화면을 나타낼 문제가 있음.
- 디자인 제작시 기본 단위가 px로 이루어져, 다른 단위보다 빠르게 대응 할 수 있는 px을 사용

## 2. CSS 선택자

# SELECTOR

- HTML에 스타일을 적용할 부분을 선택 하게끔 해주는 역할
- 선택자의 대표적인 종류는 다음과 같음
  1. 타입(엘리먼트/단순) 선택자
  2. 하위 선택자
  3. ID, CLASS 선택자
  4. 유사클래스 선택자
  5. 전체 선택자

### 타입 선택자

- HTML Tag 를 지정하여 선언
- 동일한 속성을 , (콤마) 로 구분하여 선언 가능

#### 타입 선택자

```
h1 {font-size:20px;}
```

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

```
a {color:blue;}
```

```
p {text-decoration:underline;}
```

```
h1, h2 {color:blue;}
```

### 하위 선택자

- 특정 엘리먼트 하위의 엘리먼트를 지정할 때 사용
- 공백(space) 으로 구분함
- 뒤에 나올 ID, CLASS 선택자와도 함께 사용 가능

#### 하위 선택자

```
ul li {height:30px;}
```

```
ul li a {color:blue;}
```

```
table tr td {text-align:center;}
```

### ID, CLASS 선택자

- 엘리먼트에 사용자가 직접 id, class 선택자 이름을 지정
- id는 # 으로 구분
- class는 . 으로 구분

#### ID, Class 선택자

```
#bodyContent {width:100%; height:500px;}
```

```
.clr {clear:both;}
```

```
.datetime {font-size:10px;}
```

```
#divNews .tab1 {width:50px;height:50px;}
```

```
#divNews ul {background-color:red;}
```

### 유사클래스 선택자

- 해당 엘리먼트의 **상태에 따라** 구분짓는 선택자로 HTML문서상엔 없으나 CSS에선 존재하는것 처럼 작성
- IE6에선 a (Anchor) Tag에만 지정이 가능하며 그 외 브라우저는 지원여부가 조금씩 다름

#### 유사클래스 선택자

```
a:link {color:blue;text-decoration:none;} /* 방문하지 않은 링크 */  
a:visited {color:red;text-decoration:none;} /* 방문했던 링크 */  
a:hover, a:active {color:black;text-decoration:underline;} /* 마우스 오버시와 클릭시 */
```

### 전체 선택자

- 모든 Tag에 대한 속성을 지정 할 수 있으나, 사용빈도가 매우 낮음
- 선언은 \* 로 선언

#### 전체 선택자

```
* {color:red;}  
div * {font-size:10px;}
```



### 그 외 선택자

- 자식(child), 인접(Adjacent Sibling), 속성(Attribute)선택자가 추가로 있으나, IE6에서 지원을 하지 않는 관계로 널리 사용되고 있지는 않음

#### 그 외 선택자

/\* 자식 선택자 \*/

```
div > a {font-size:20px;}
```

/\* 인접 선택자 \*/

```
h1 + div {background-color:blue;}
```

/\* 속성 선택자 \*/

```
[class="box"] {background-color:red;}
```

### 선택자 우선순위

- 스타일 선언이 위에서 아래로 순차적으로 적용이 되고,  
마지막에 선언된 스타일이 우선순위를 갖는 특징에 대해서 알아보시다.

#ex1.html

```
#btnLink {color:red;}
```

```
#btnLink {color:blue;}
```

```
#btnLink {color:#000;}
```

## 2. CSS 선택자

- 같은 선택자가 아닌 다른 선택자로 중복 선언을 한다면?

#ex2.html

```
<style type="text/css">
  .tit_name { color:red; }
  h3 { color:green; }
  #titName { color:blue; }
</style>

<h3 id="titName" class="tit_name" style="color:yellow;">타이틀 제목</h3>
```

선택자에 대한 지정순위

in-line > id > class > type

### 선택자 강제 우선순위

- 선택자에 대한 지정순위를 무시하고 사용자의 스타일이 우선적으로 적용되게 함
- style 뒤에 공백(space) 후 !important 를 추가

#### #ex2.html 이어서

```
<style type="text/css">
  .tit_name { color:red; }
  h3 { color:green !important; }
  #titName { color:blue; }
</style>

<h3 id="titName" class="tit_name" style="color:yellow;">타이틀 제목</h3>
```

# 3. CSS 속성

## FONT

- 글자와 관련된 스타일을 지정할 수 있는 속성
  1. font-style : normal / *italic* / *oblique* ;
  2. font-weight : normal / **bold** ;
  3. font-size : 12px ;
  4. line-height (줄간격) : 12px;
  5. font-family (글꼴 서체) : gulim ;

#### Font 속성 축약형 #font.html

```
span { font : italic bold 12px/12px gulim; }
```

## COLOR

- 글자의 색상을 변경하는 속성

### color 속성

```
color : red;  
color : rgb(255,0,0);  
color : #FF0000;  
color : #F00; (축약)
```

## TEXT-INDENT

- 글자를 들여쓰거나 내여쓸때 사용

### text-indent 속성

```
text-indent : 10px ;  
text-indent : -10px ;
```

## TEXT-DECORATION

- 글자의 윗줄, 중간줄, 밑줄 등을 적용시에 사용

### text-decoration 속성

```
text-decoration : none;  
text-decoration : underline;  
text-decoration : line-through;  
text-decoration : overline;
```

## TEXT-ALIGN

- 글자의 정렬 기준을 지정

### text-align 속성

```
text-align : left;  
text-align : right;  
text-align : center;  
text-align : justify;
```



## 실습 #1 (FONT\_이름.HTML)

- 아래와 같은 스타일이 표현되도록 CSS속성을 사용하여 적용해 봅시다.

1. 오늘 점심은 어떤 메뉴일까?

2. 오늘 점심은 어떤 메뉴일까?♪

3. 오늘 점심은 어떤 메뉴일까?

4. 오늘 점심은 어떤 메뉴일까?

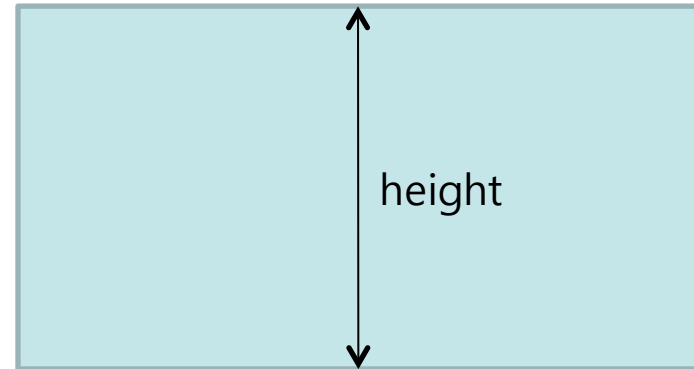
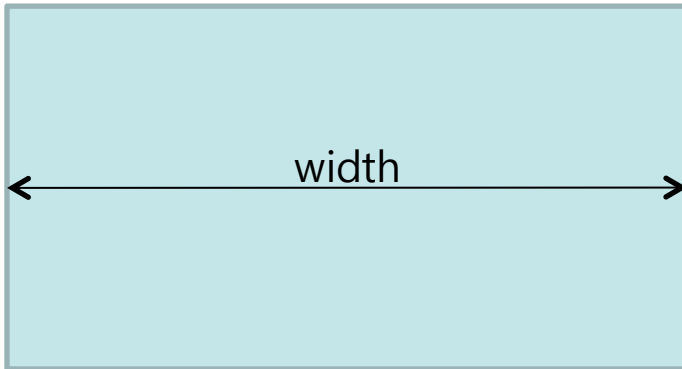
5. 오늘 점심은 어떤 메뉴일까?

6. 오늘 점심은 어떤 메뉴일까?

- 조건
  - 기본 <p> tag 사용
  - in-line 방식 스타일 사용
- 1. 굵은 돋움 , 크기: 14px , 색상: blue
- 2. 이탤릭체 굴림 , 크기: 16px , 색상: #E8334D
- 3. 밑줄이 있는 돋움 , 크기: 12px
- 4. 들여쓰기가 10px 되어있는 돋움 , 크기: 15px
- 5. 줄간격이 50px 인 굵은 돋움 , 크기: 16px , 색상: green
- 6. 우측정렬이 되어있는 밑줄이 있는 굵은 돋움 , 크기: 20px

## WIDTH, HEIGHT

- Block 요소의 가로, 세로 사이즈를 지정



### width, height

width : 150px ;

width : 50% ;

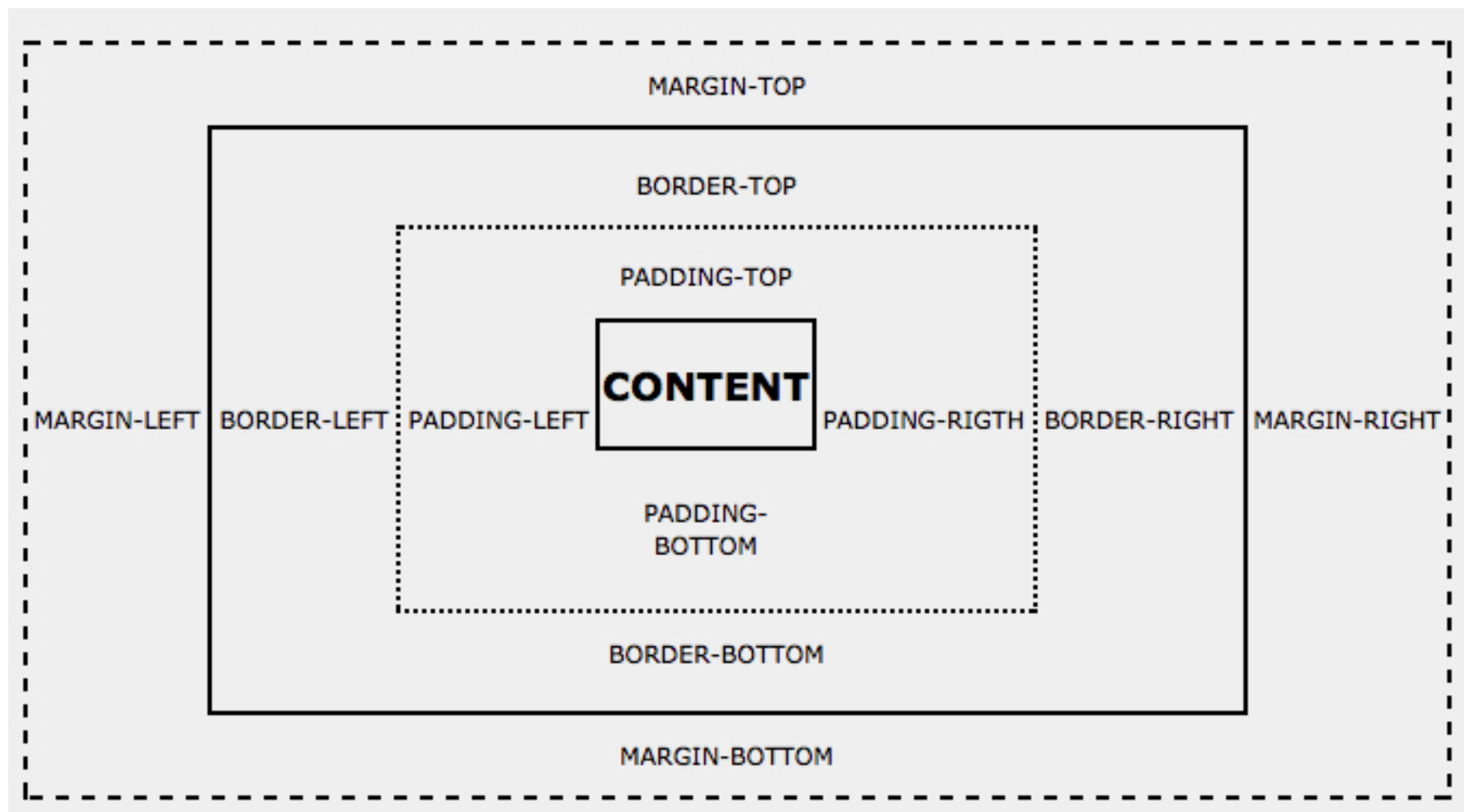
height : 300px ;

height : 100% ;

## MARGIN, PADDING

- 엘리먼트의 바깥여백(margin)과 안쪽 여백(padding)을 지정





## MARGIN, PADDING

#### 1. margin

- margin-top
- margin-right
- margin-bottom
- margin-left

#### 2. padding

- padding-top
- padding-right
- padding-bottom
- padding-left

#### margin, padding 속성 #box.html

```
margin-top: 10px;
```

```
margin-right: 40px;
```

```
margin : 10px 40px 20px 10px ;
```

```
padding : 20px ;
```

```
margin : 20px 10px;
```

```
margin : 20px 10px 30px;
```

- 축약형의 순서는 위, 오른쪽, 아래, 왼쪽 시계방향으로 인식

## MARGIN 병합

- 세로 방향으로 마진을 지정한 두개의 서로 다른 요소가 수직 방향으로 접해 있을 때, 두 요소 사이의 마진 간격은 두 요소의 마진값 중 큰값으로 병합

### #ex3.html

```
<style type="text/css">
body {margin:0; padding:0;}
.margin_box {width:100%; height:30px; margin:20px 0 50px; background-color:#dcdcdc;
}
</style>

<div class="margin_box"> </div>
<div class="margin_box"> </div>
<div class="margin_box"> </div>
<div class="margin_box"> </div>
```

### 3. CSS 속성

## MARGIN 을 이용한 BLOCK BOX 의 중앙 정렬

- Block box는 margin 좌우 속성값을 auto로 지정해 중앙에 위치시킬 수 있음
- 단, 정렬하려는 Block 요소에 width를 지정해야 함

#### #ex4.html

```
<style type="text/css">
    .box1 {width:200px;height:100px;margin:0 auto;background-color:red;}
</style>

<div class="box1"> </div>
```

## BORDER

- 엘리먼트 바깥에 선을 그려줌(외곽선)
- 방향에 따라 스타일을 따로 지정가능
- width, height 값과는 별개로 엘리먼트의 padding 바깥쪽에 영역을 차지

#### 1. border

- border-width
- border-style
- border-color

#### 2. border-방향

- border-방향-width
- border-방향-style
- border-방향-color

#### 3. border-style 속성

- solid (실선)
- dashed (끊어진 실선)
- dotted (점선)

#### border 축약형 #ex5.html

```
border : 1px solid red ;
```

```
border : 1px dotted #0000FF ;
```



## BACKGROUND

- 엘리먼트 영역에 색상, 이미지를 채우는 속성
  1. **background-color** : #fafafa ;
    - 색상이름/색상코드
    - transparent(투명)
  2. **background-image** : url(http://... .gif) ;
  3. **background-repeat** : no-repeat ;
    - repeat, no-repeat, repeat-x, repeat-y
  4. **background-position** : 0px 10px ;
    - 0px 0px (left,top 기준)
    - left(right), top(bottom), center

#### background 축약형 #ex6.html

```
background : #FF0000 url(daum.gif) no-repeat 0px 10px ;
```

```
background : red url(daum.gif) repeat-y left top ;
```

```
background : url(daum.gif) repeat-x 0px 0px ;
```

### 3. CSS 속성

## 실습 #2 (BOX\_이름.HTML)

- 아래와 같은 스타일이 표현되도록 CSS속성을 사용하여 적용해 봅시다.
- box는 div태그를 사용
- CSS는 in-line방식의 스타일 사용

### 설명

- 박스의 크기는 가로 300px, 세로 300px, 배경색은 red
- 박스의 외곽선은 3px 두께의 실선이고 외곽선 색은 #000
- 박스는 가운데 정렬
- 박스의 안쪽 여백은 위로 100px, 좌측 20px
- 박스안에 '본인 이름' 을 적어주세요

완성된 박스는 외곽선을 포함한 크기가 가로 300px, 세로 300px 이 맞을까요?

## DISPLAY

- 문서 내에서 엘리먼트가 가지고 있는 고유한 렌더링 속성을 변경
- inline요소를 block요소로, block요소를 inline요소로 변경가능
- 그 외에 화면에 표시되지 않게 할 수 있음

### display 속성 #ex7.html

```
display : none ;  
display : block ;  
display : inline ;  
display : inline-block ;
```

- inline-block : inline요소처럼 배치되면서 block요소의 속성(width, height, margin, padding)들을 사용할 수 있는 속성

## OVERFLOW

- block요소의 자식 콘텐츠가 부모요소의 사이즈를 변화시킬 때 숨김 / 스크롤바 표시여부를 지정
- 부모의 사이즈가 지정 되어 있을 경우에 사용가능

### overflow 속성 #ex8.html

`overflow : visible ;` (기본)

`overflow : hidden ;`

`overflow : scroll ;`

`overflow : auto ;`

- `overflow : scroll` 가로, 세로에 대한 속성을 따로 지정 할 수 있다.
  - `overflow-x`
  - `overflow-y`

## FLOAT

- 엘리먼트를 block형태로 부유시킬때 사용 (display:block; 을 안해주어도~)
- block형태로 특정방향(left, right)으로 정렬

### float 속성

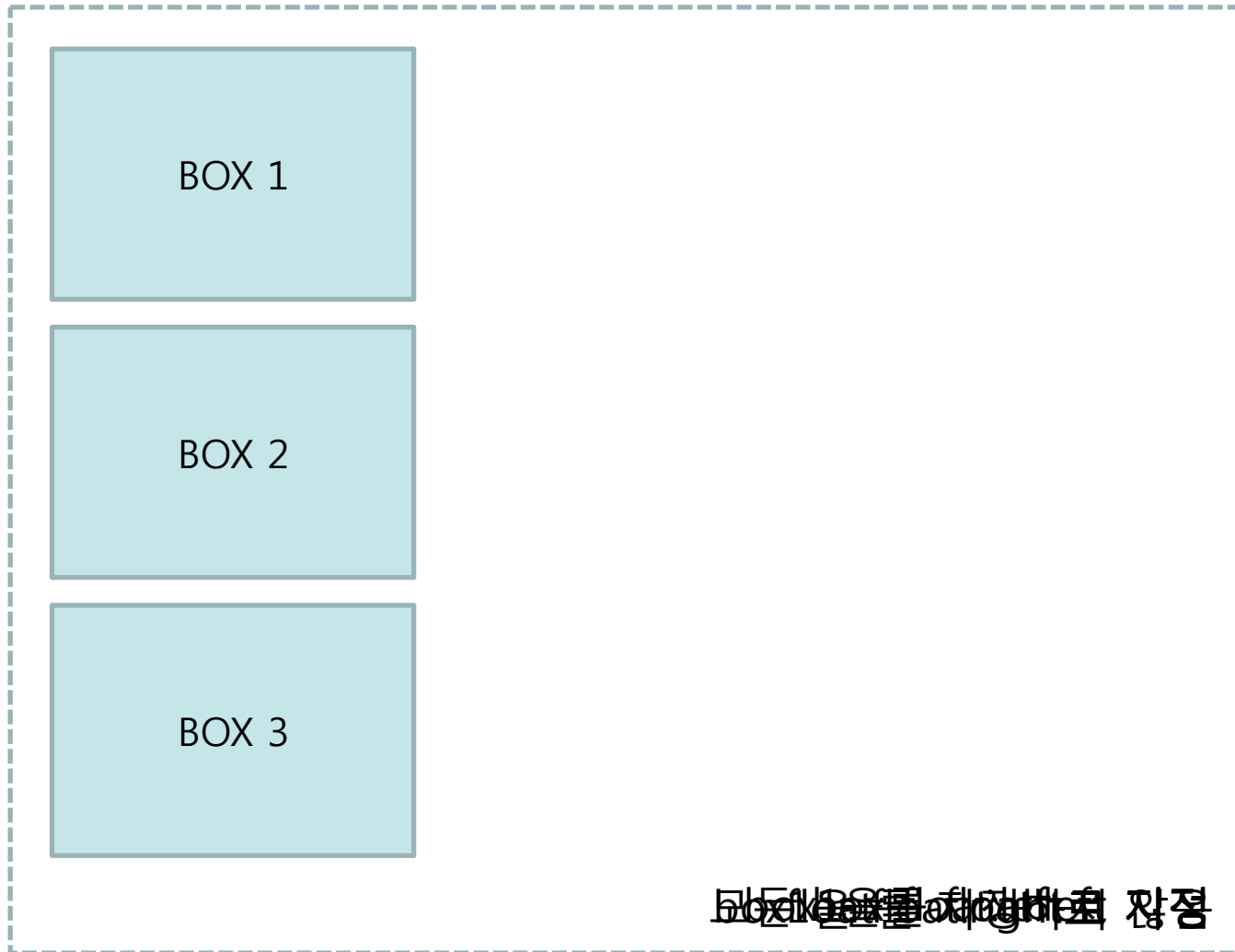
float : inherit ; (기본 : 밖의 float속성을 상속받음)

float : left ;

float : right ;

float : none ;

## FLOAT



## CLEAR

- float 를 해제할 때 사용
- 이전 엘리먼트의 float값을 상속받아 같은 방향으로 정렬되는 현상을 막기 위해 사용

### Clear 속성

`clear : inherit ;` (기본)

`clear : left ;` (왼쪽으로 float된 엘리먼트의 하단에 위치)

`clear : right ;` (오른쪽으로 float된 엘리먼트의 하단에 위치)

`clear : both ;` (모든 float의 하단에 위치 – 개념상 float상속을 받지 않음)

## POSITION

- 엘리먼트의 상대적인 위치에 대한 속성
- float과는 다르게 왼쪽, 오른쪽 정렬하는 형태가 아닌, 위치하는 곳에 대한 값(오프셋)을 지정

### position 속성

`position : static ;` (기본, offset을 가지지 않음)

`position : relative ;` (상대적, 원래 위치를 기준)

`position : absolute ;` (절대적, 부모위치를 기준)

`position : fixed ;` (화면에서 고정, ie6 지원 불가)

`top : 10px ;`

`left : 20px ;`

`bottom : 10px ;`

`right : 20px ;`



## POSITION

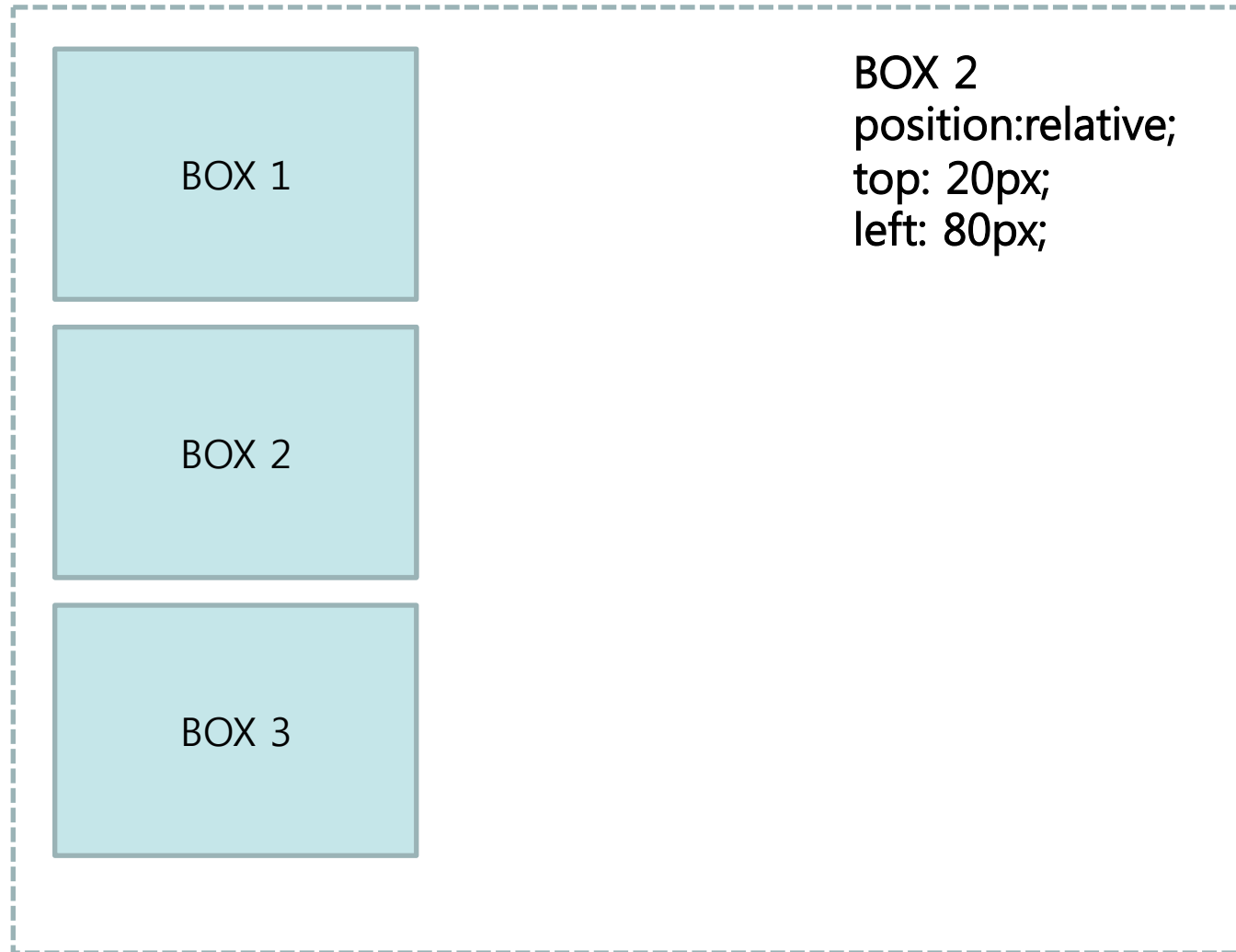
`position : static;`

BOX 1

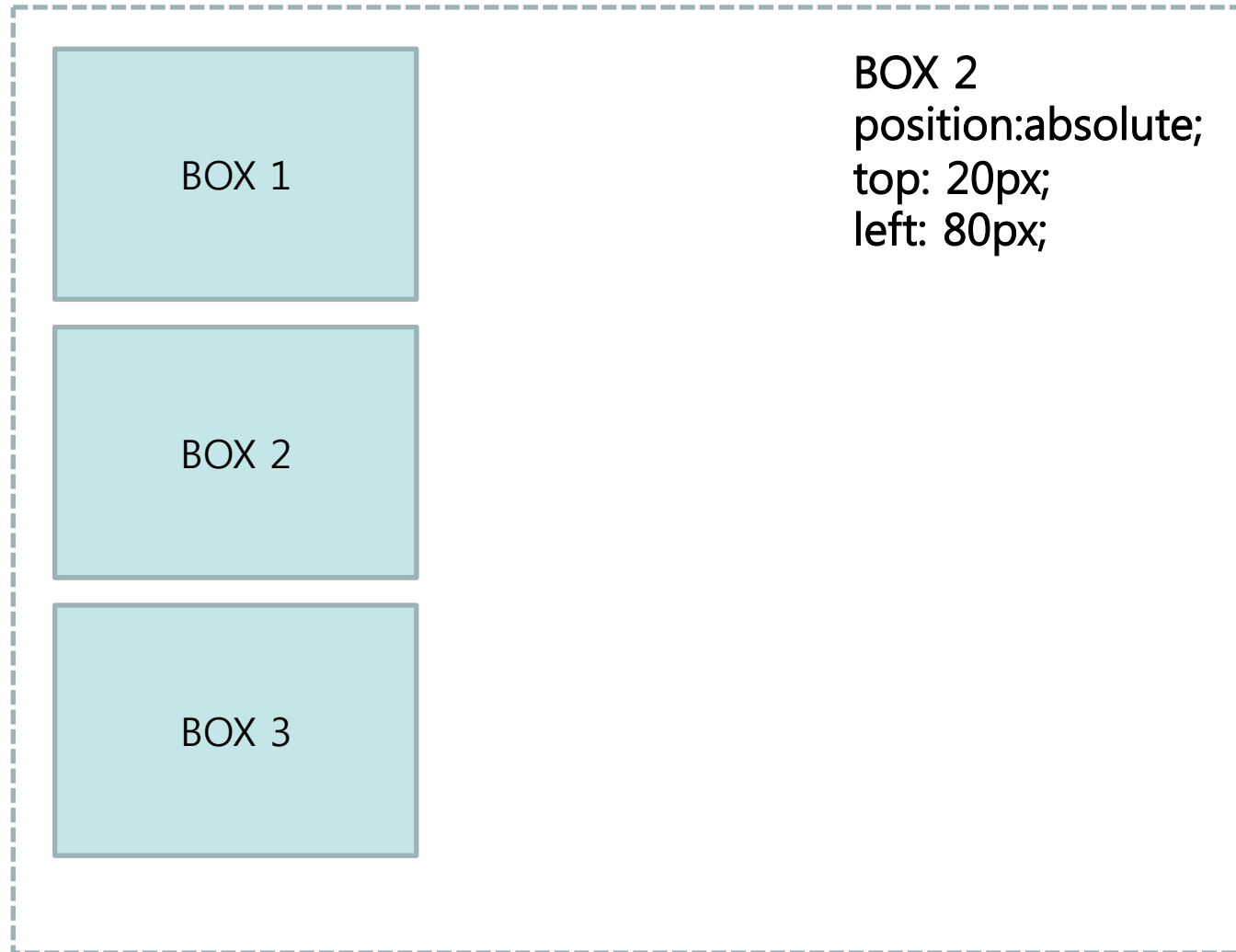
BOX 2

BOX 3

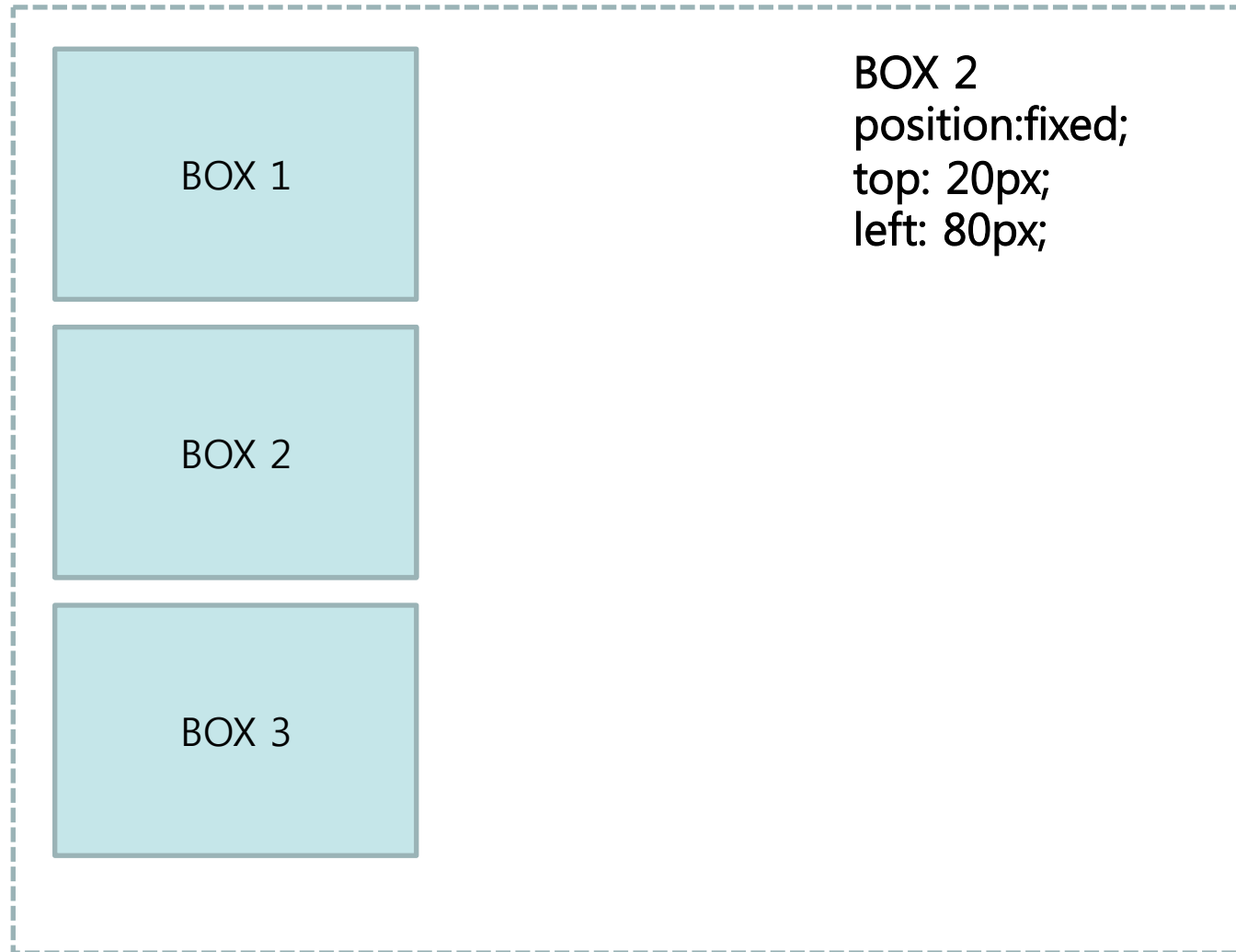
## POSITION



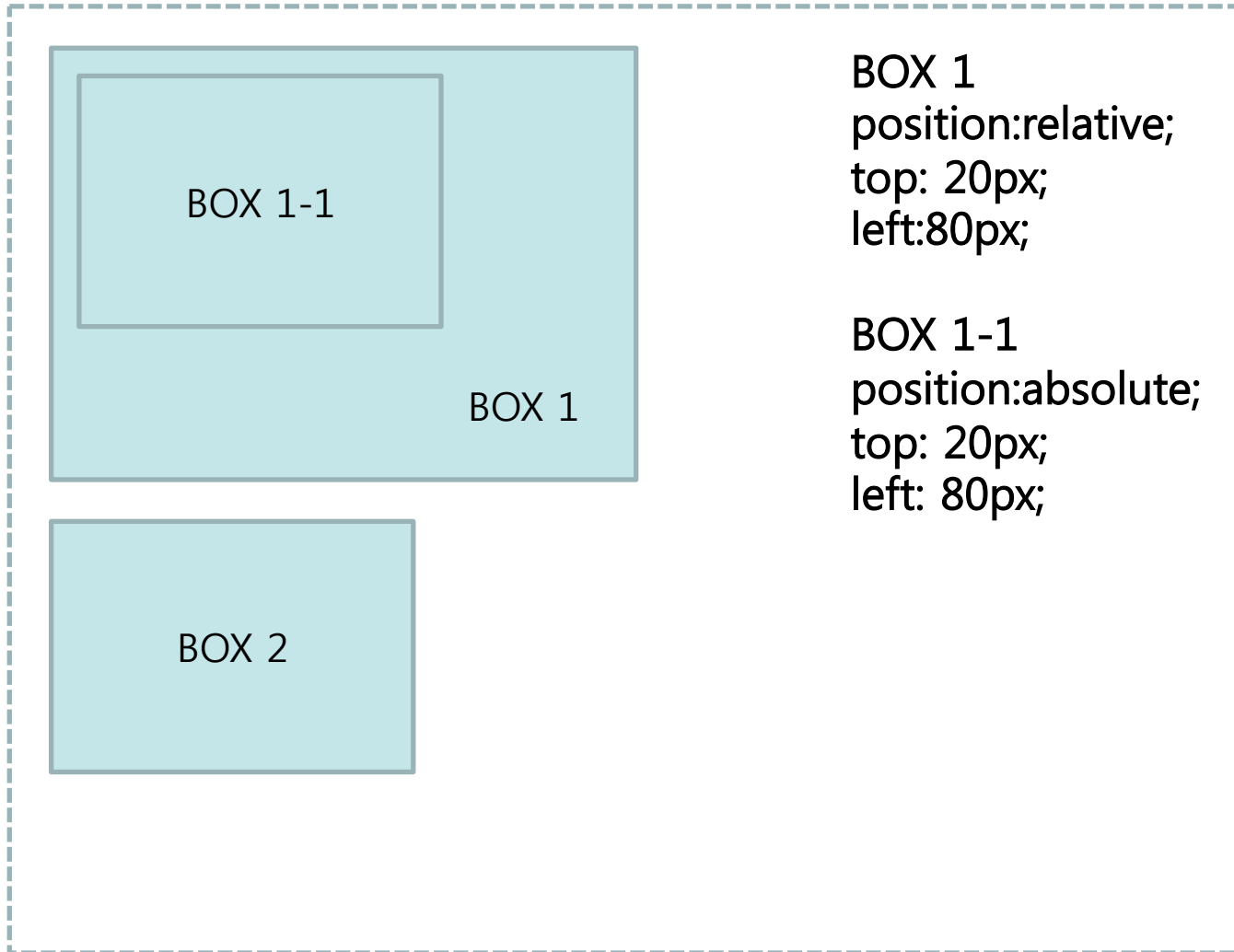
## POSITION



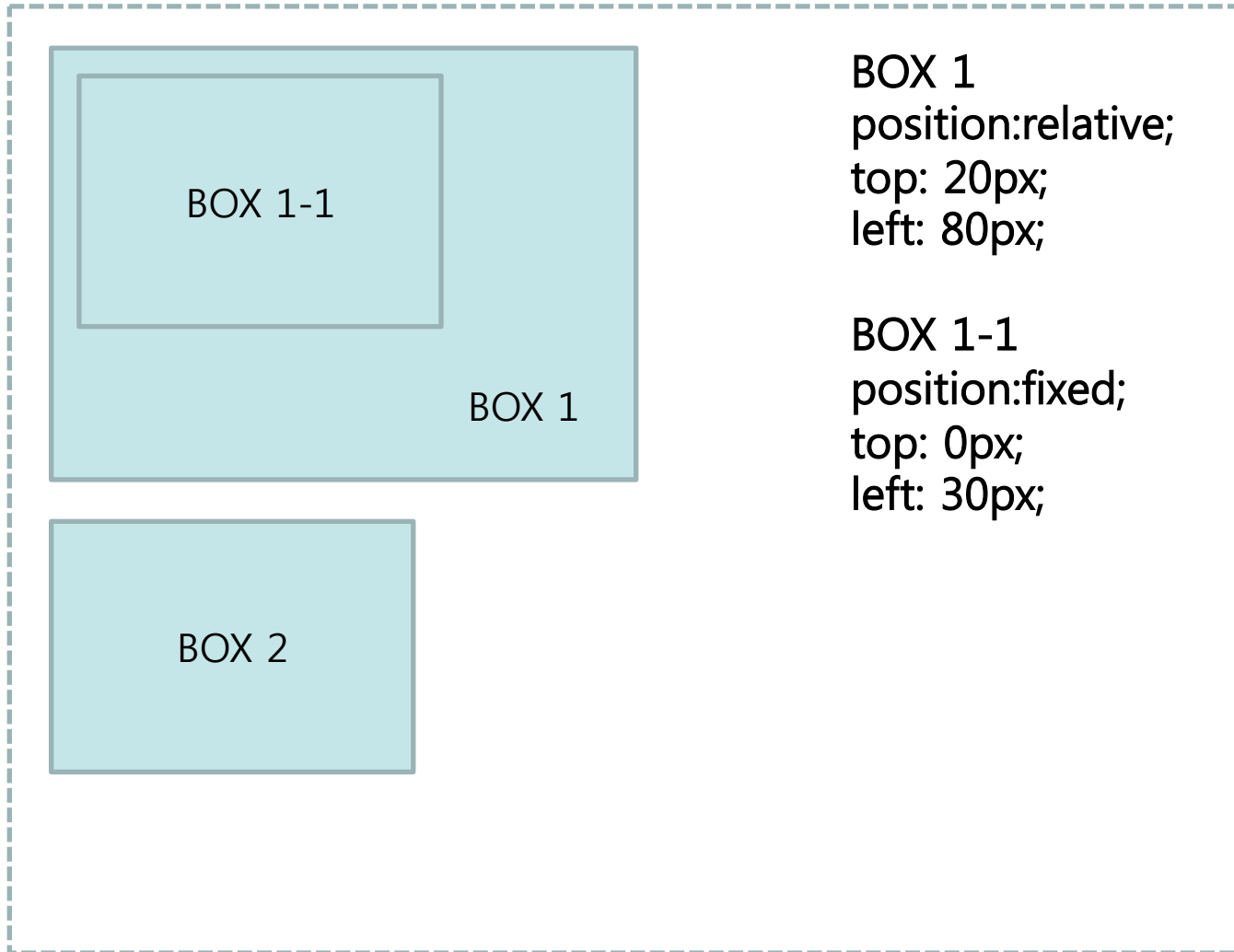
## POSITION



## POSITION



## POSITION



## Z-INDEX

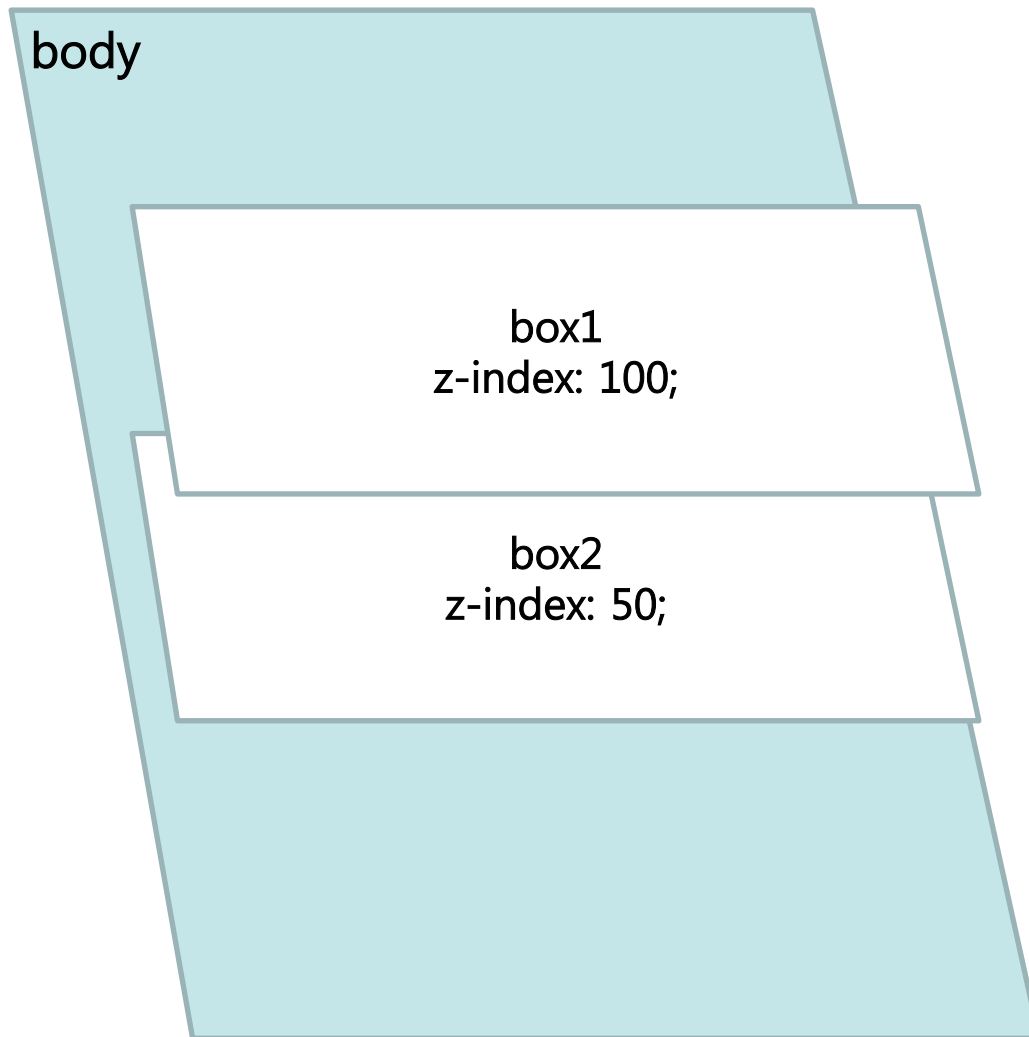
- 엘리먼트의 입체적인 순서를 정의
- 수치가 작을수록 아래쪽에 있다는 것을 의미

### z-index 속성

```
z-index : 3 ;  
z-index : 10000 ;  
z-index : 400 ;
```

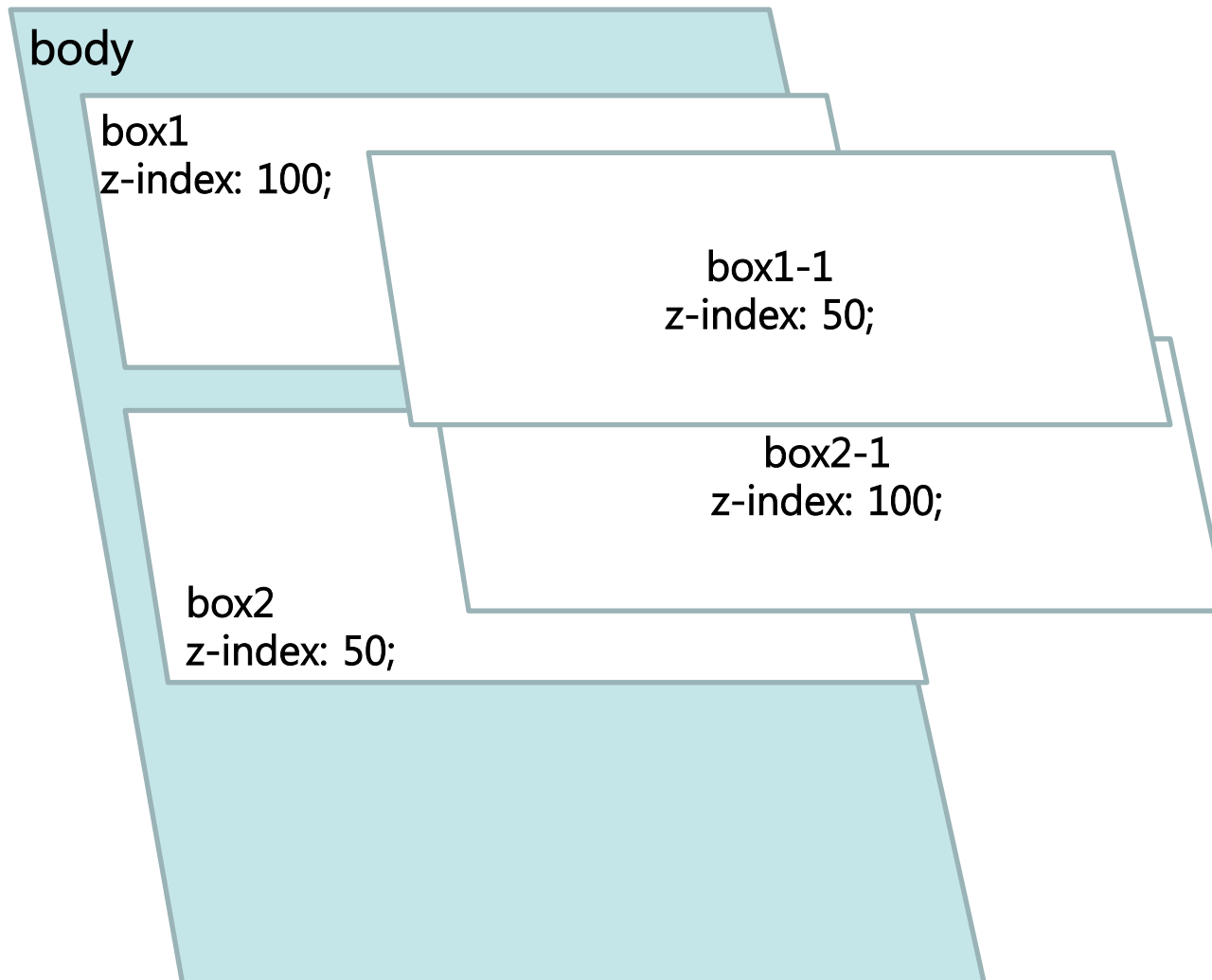
- box안의 엘리먼트의 z-index가 높아도 상위(부모)엘리먼트끼리의 수치가 낮으면 아래쪽에 위치함

## Z-INDEX





## Z-INDEX



감사합니다