

CSS Part.1

Front-end 웹 개발 개론

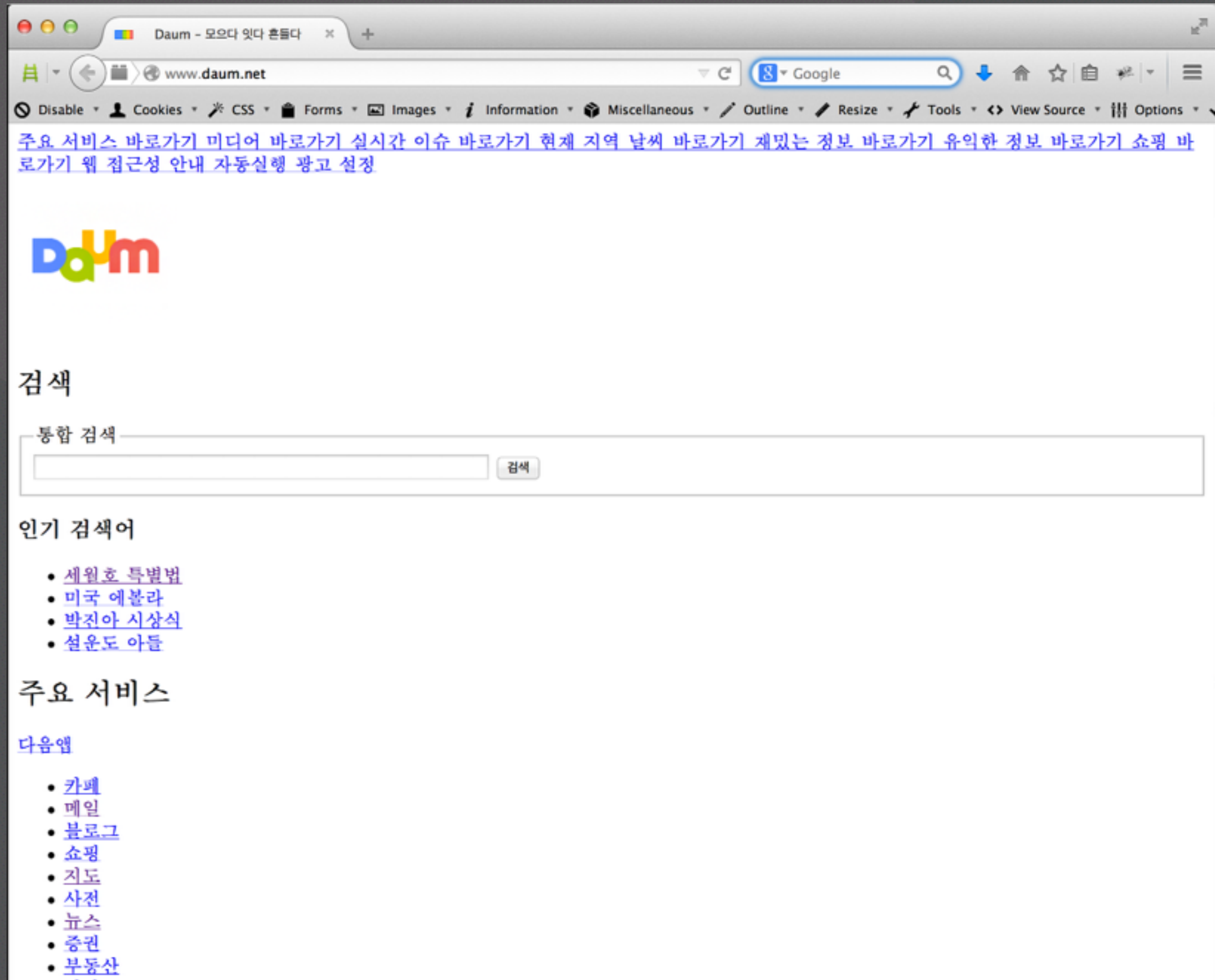
Daum Service 웹 표준 개발팀 홍윤표

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 2.1이후 CSS는 모듈로 분리 되어 각 모듈별로 버전업이 진행중에 있음

CSS (Cascading Style Sheet)



CSS (Cascading Style Sheet)



CSS 선언방식 - inline 방식

HTML Tag에 직접 style 속성 값을 지정

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

CSS 선언방식 - internal 방식

HTML 문서 내부에 style tag로 선언

head tag안에 넣는것이 정석이지만, 상황에 따라 body안에도 선언이 가능

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


CSS 선언방식 - External 방식

CSS를 따로 파일로 저장하여 불러오는 방식

파일을 여러개 불러올 수 있기 때문에 공통/자주 사용되는 부분을 따로 저장하여 사용하기에 용이함

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


CSS 수치 단위

em, % (상대단위)

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

px, pt (절대단위)

- px (픽셀, $1px = 1\text{dot}$) – 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

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

Type Selector

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

```
h1 {font-size:20px;}  
ul {margin:0px;}  
a {color:blue;}  
p {text-decoration:underline;}  
  
h1, h2 {color:blue;}
```


Descendant Selector

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

```
ul li {height:30px;}  
ul li a {color:blue;}  
table tr td {text-align:center;}
```

id, class Selector

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

```
#bodyContent {width:100%; height:500px;}  
.datetime {font-size:10px;}  
  
#divNews    .tab1 {width:50px;height:50px;}  
#divNews    ul {background-color:red;}
```

pseudo class Selector

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

```
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 */
```



Universal Selector

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

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


Combinator Selector + ETC

- 자식(child), 인접(Adjacent Sibling), 속성(Attribute)선택자가 추가로 있음
- Internet Explorer 6에서는 미지원되는 문제점이 존재함
- 그외에도 다양한 Selector 들이 존재함
- <http://quirksmode.org/css/selectors/>

```
/* 자식 선택자 */  
div > a {font-size:20px;}  
  
/* 인접 선택자 */  
h1 + div {background-color:blue;}  
  
/* 속성 선택자 */  
[class="box"] {background-color:red;}
```

Combinator Selector + ETC

- 자식(child), 인접(Adjacent Sibling), 속성(Attribute)선택자가 추가로 있음
- Internet Explorer 6에서는 미지원되는 문제점이 존재함
- 그외에도 다양한 Selector 들이 존재함
- <http://quirksmode.org/css/selectors/>

```
/* 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.

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

선택자 우선순위

같은 선택자가 아닌 다른 선택자로 중복 선언을 한다면?

```
<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>
```


선택자 우선순위

같은 선택자가 아닌 다른 선택자로 중복 선언을 한다면?

```
<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

```
<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>
```

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

```
<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>
```

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 내어쓸때 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

글자를 들여쓰거나 내어쓸때 사용

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?

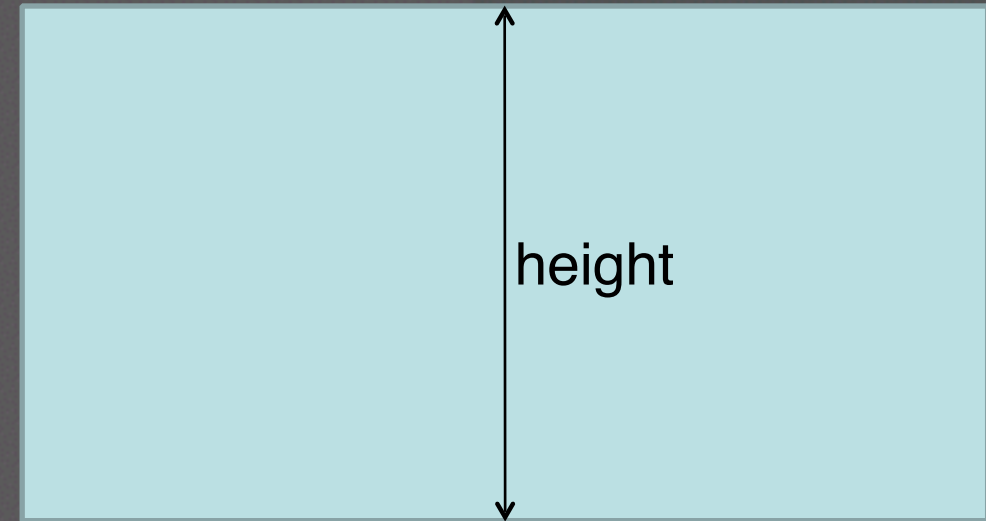
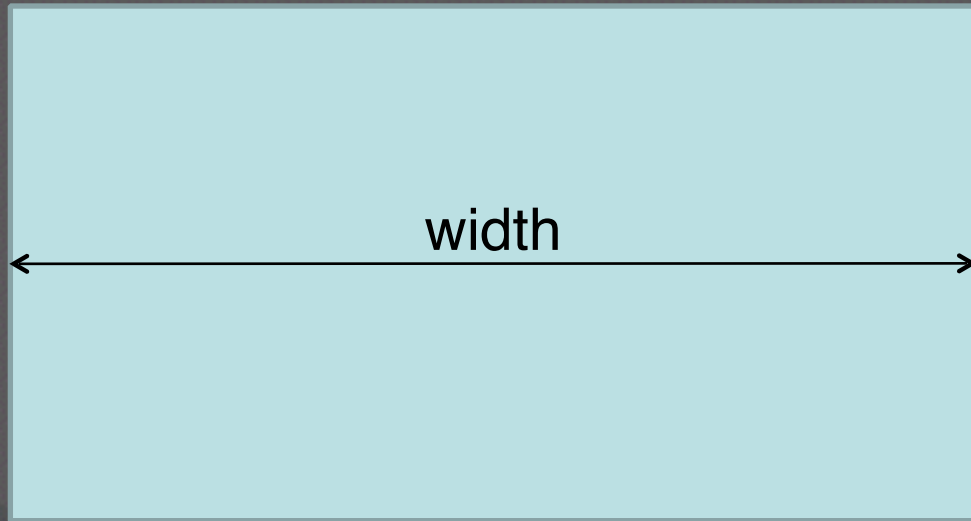
조건

- 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

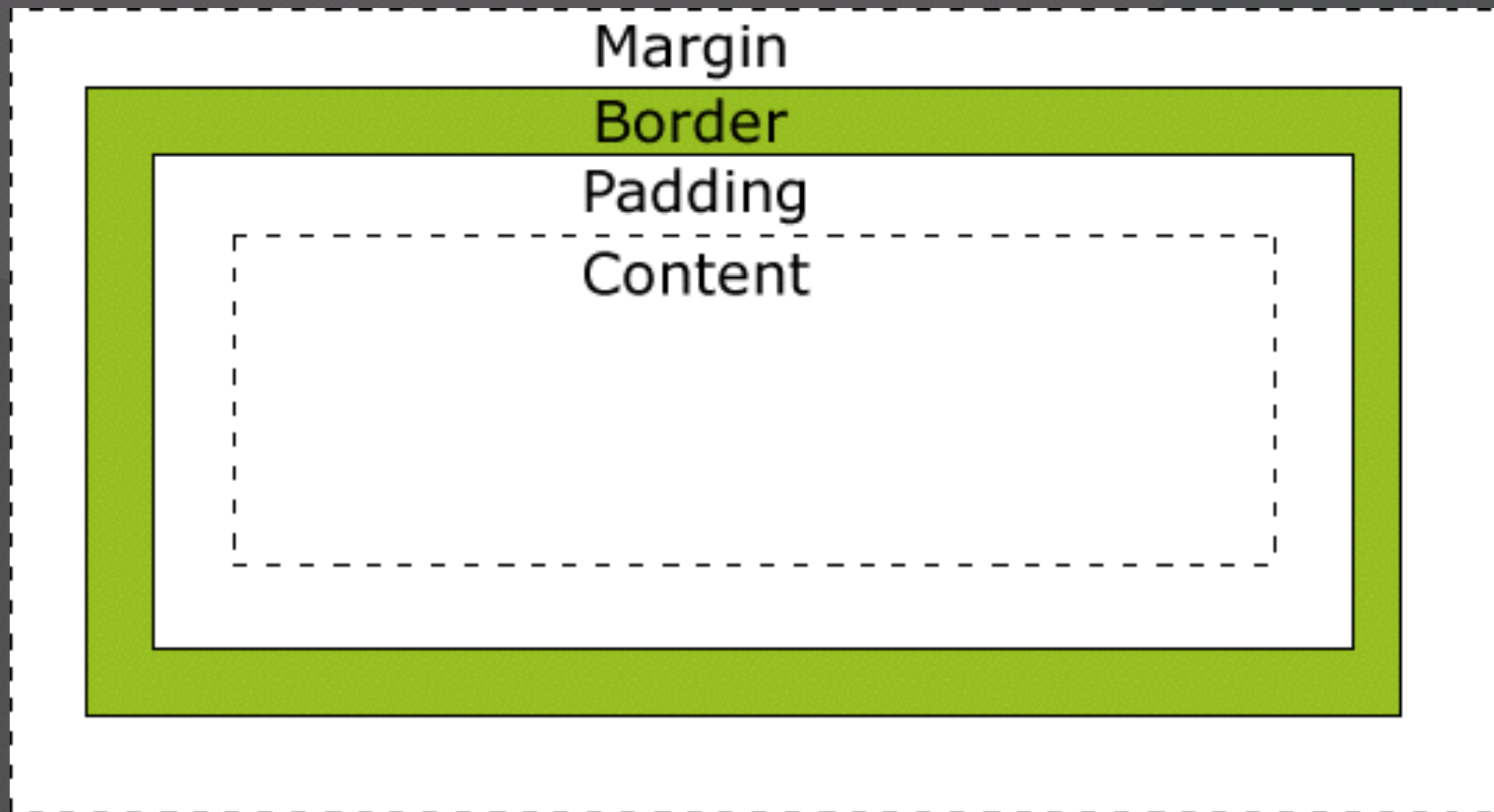


width, height

```
width : 150px ;  
width : 50% ;  
height : 300px ;  
height : 100% ;
```

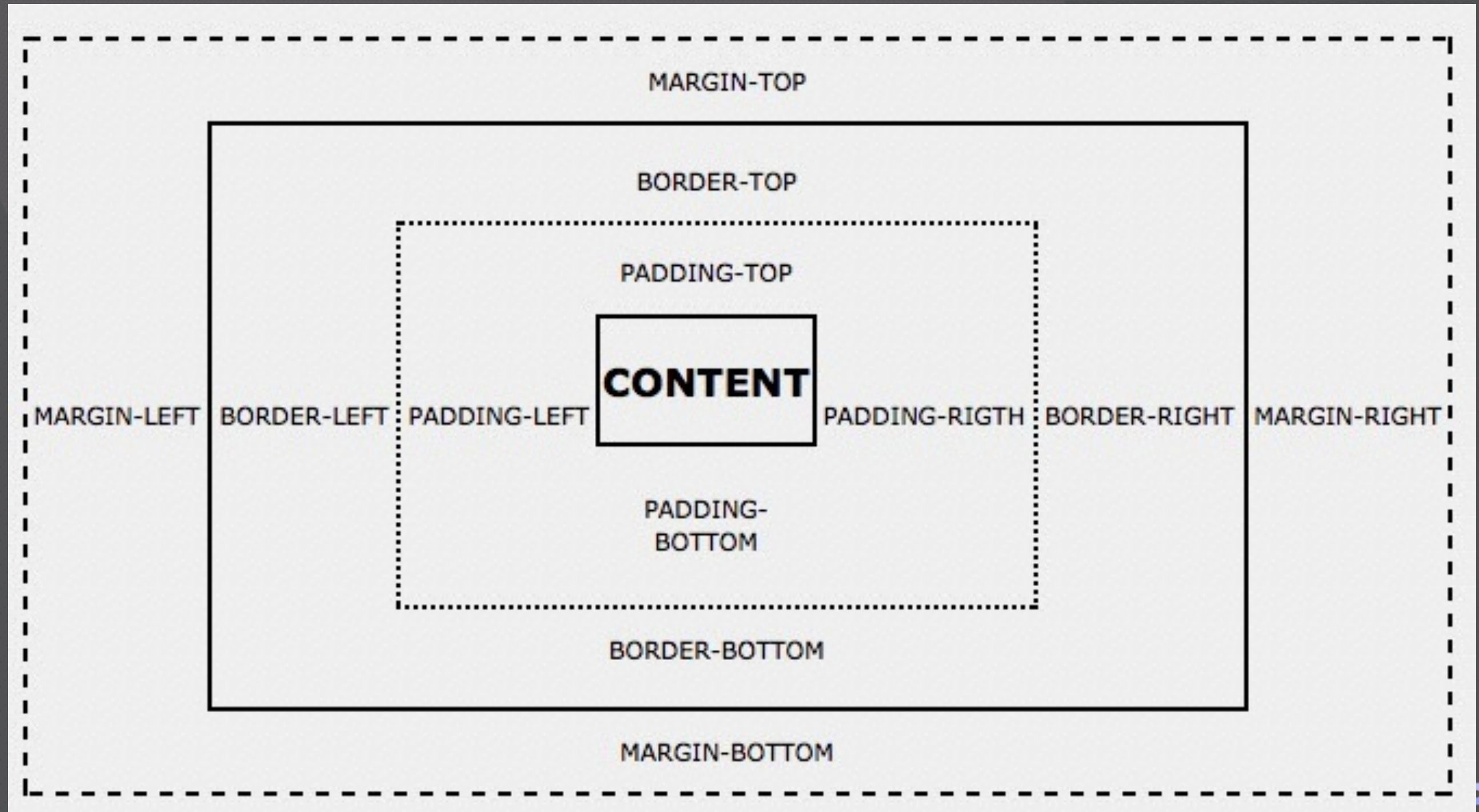
margin, padding

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



BOX MODEL

콘텐츠 영역과 콘텐츠를 감싸는 영역에 대한 정의



margin, padding

1. margin

- 1.margin-top
- 2.margin-right
- 3.margin-bottom
- 4.margin-left

2. padding

- 1.padding-top
- 2.padding-right
- 3.padding-bottom
- 4.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>
```

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

- 1.border-width
- 2.border-style
- 3.border-color

2. border-방향

- 1.border-방향-width
- 2.border-방향-style
- 3.border-방향-color

3. border-style 속성

- 1.solid (실선)
- 2.dashed (끊어진 실선)
- 3.dotted (점선)

border 축약형 #ex5.html

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

background

엘리먼트 영역에 색상, 이미지를 채우는 속성

- background-color : #fafafa ;
 - 색상이름/색상코드
 - transparent(투명)
- background-image : url(http://... .gif) ;
- background-repeat : no-repeat ;
 - repeat, no-repeat, repeat-x, repeat-y
- 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 ;
```


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

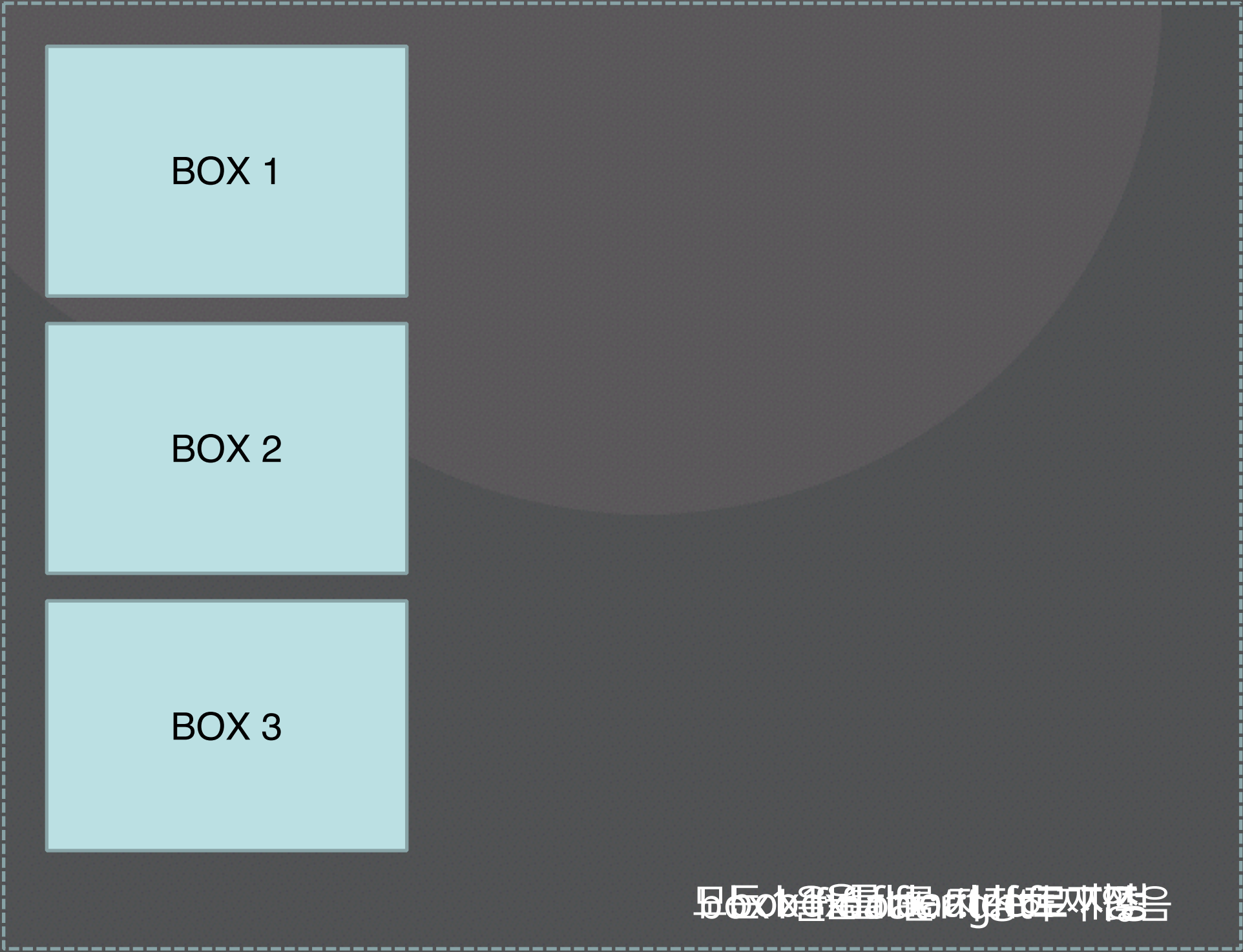
float

- 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;

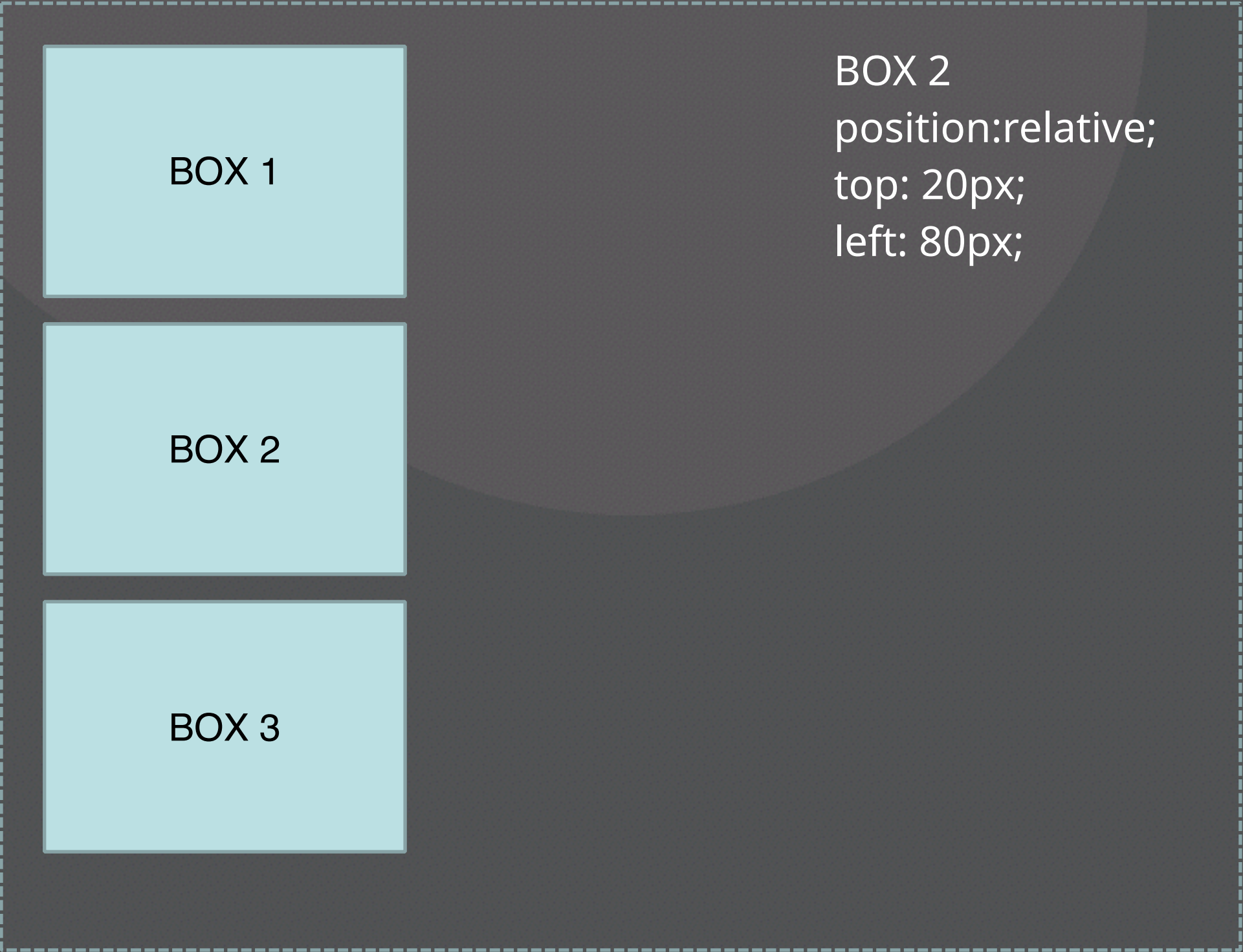
BOX 1

BOX 2

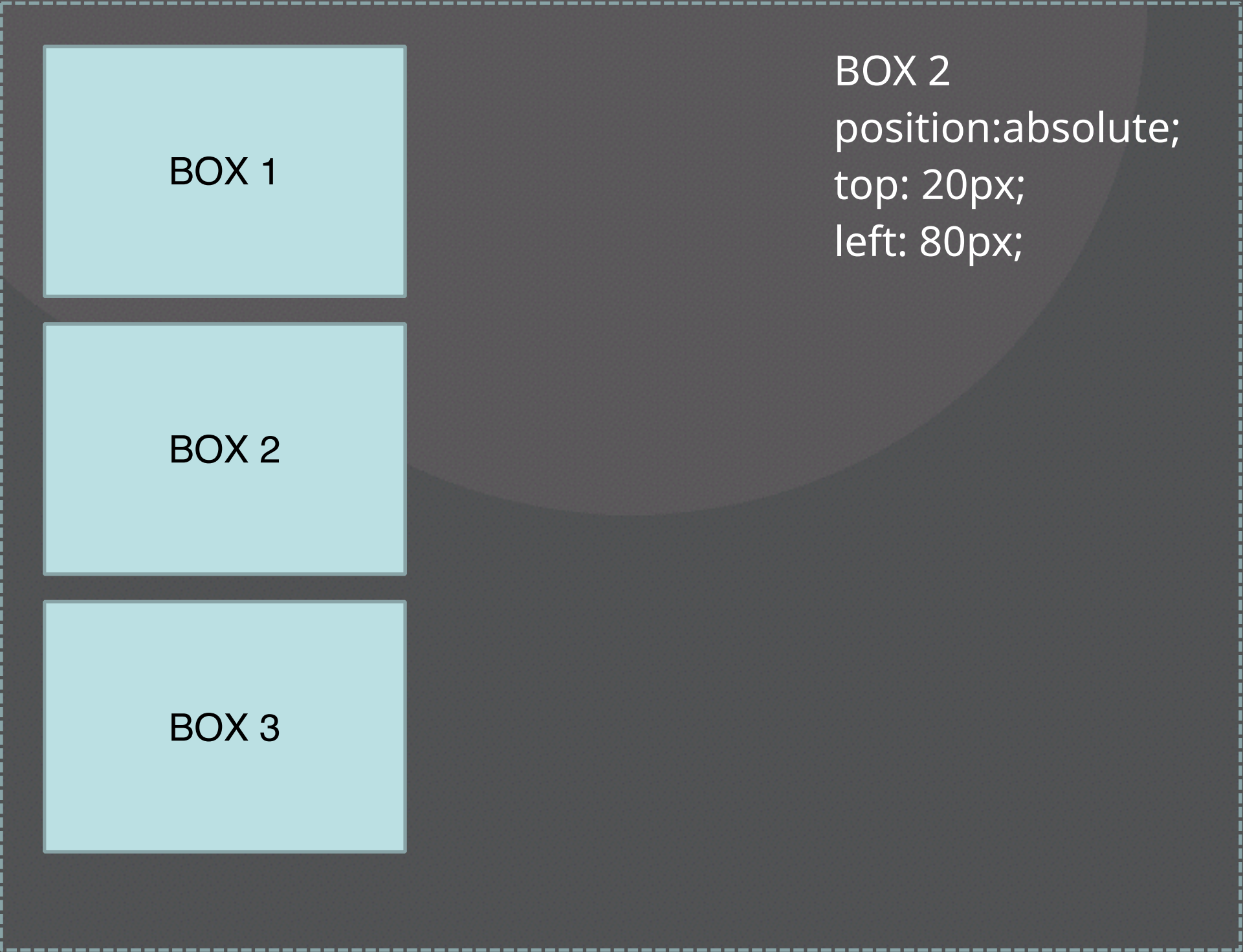
BOX 3



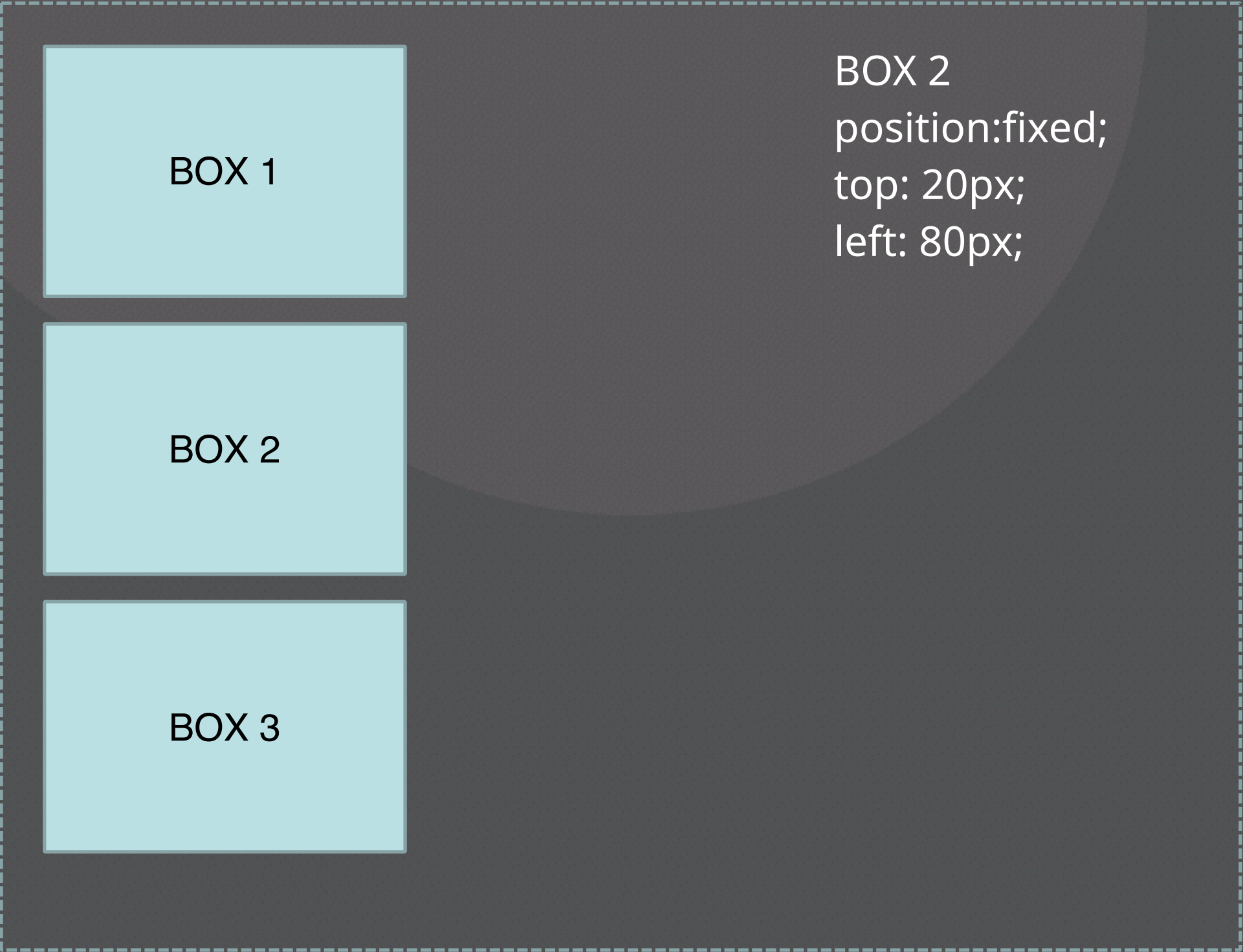
position



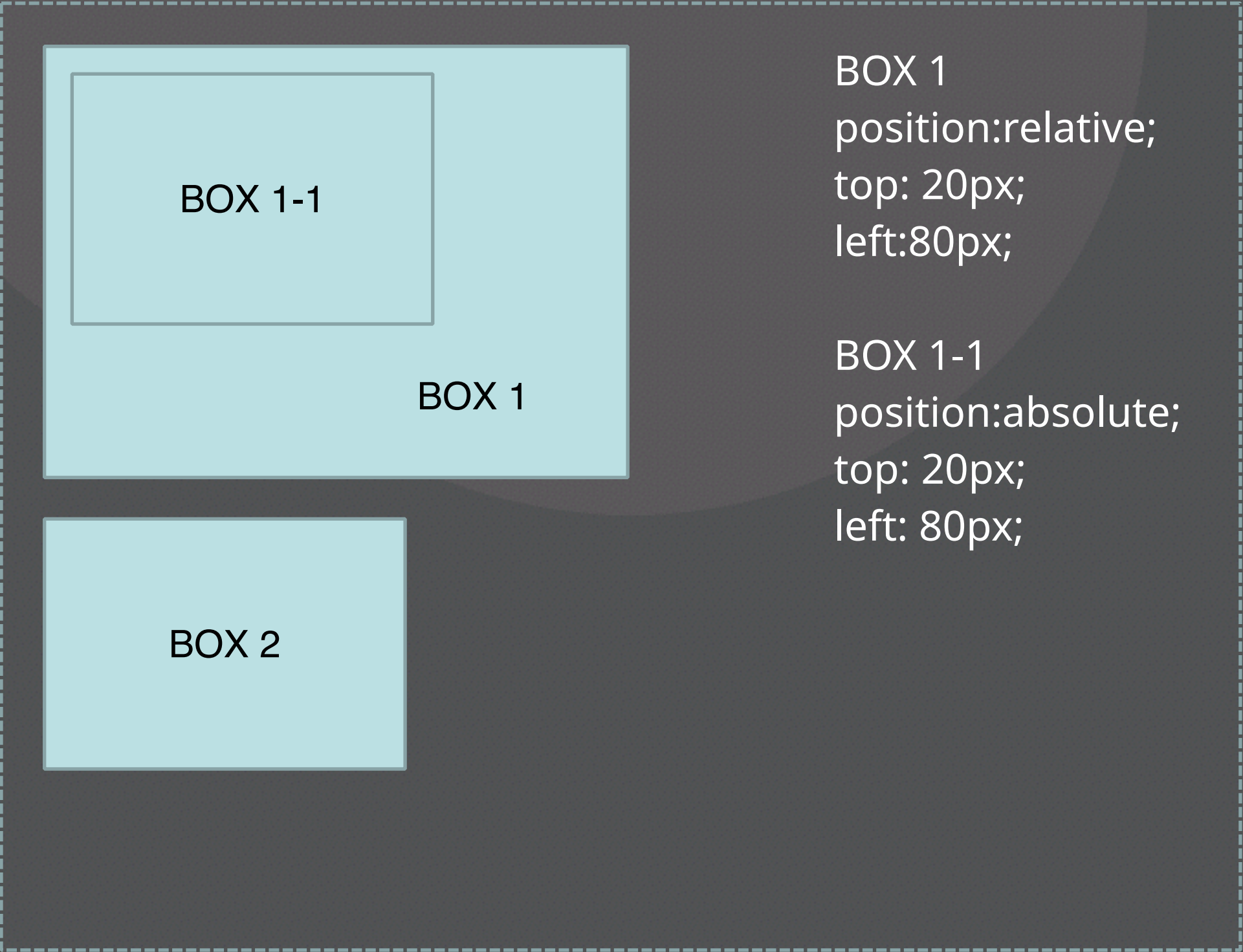
position



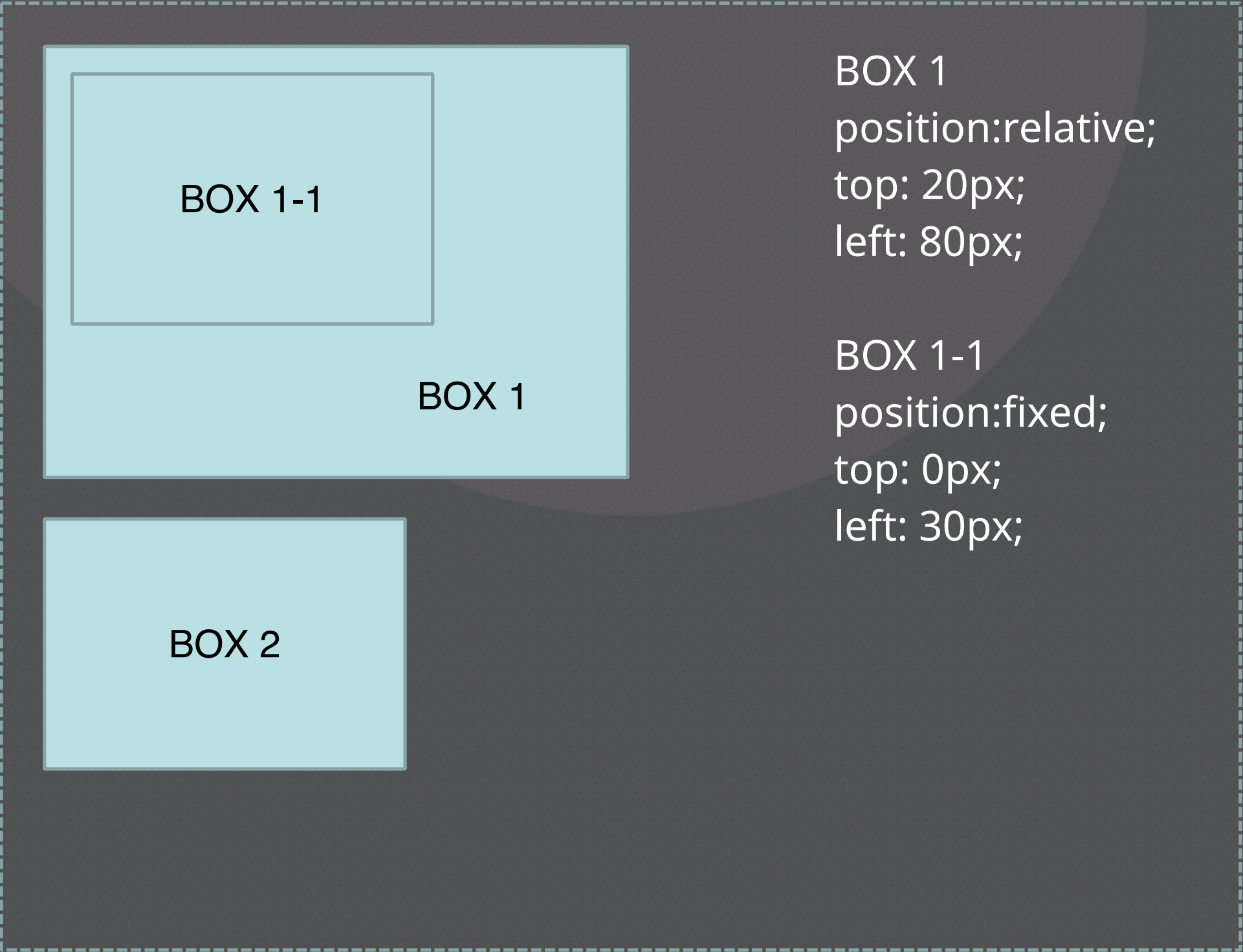
position



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

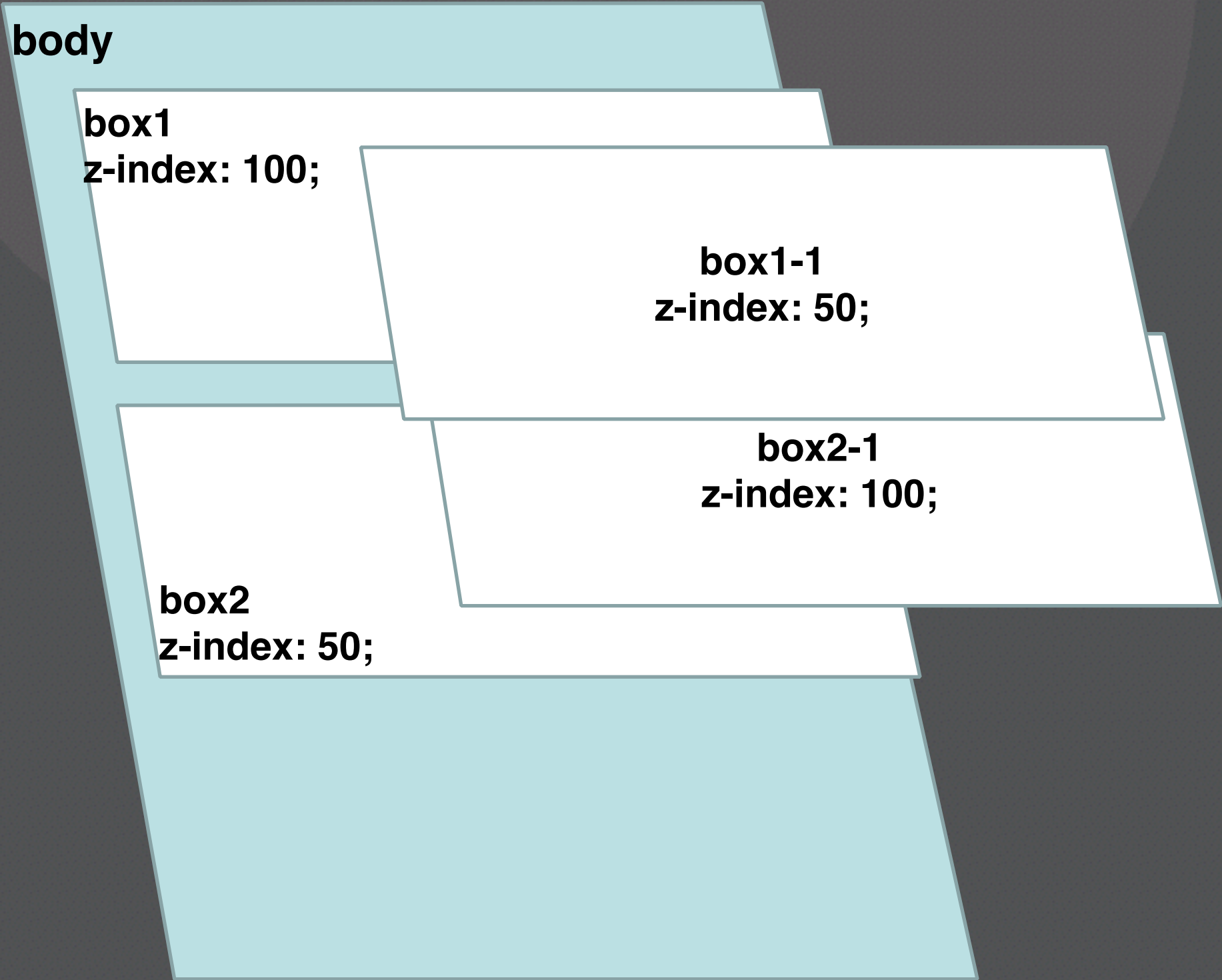
body

box1
z-index: 100;

box2
z-index: 50;



z-index

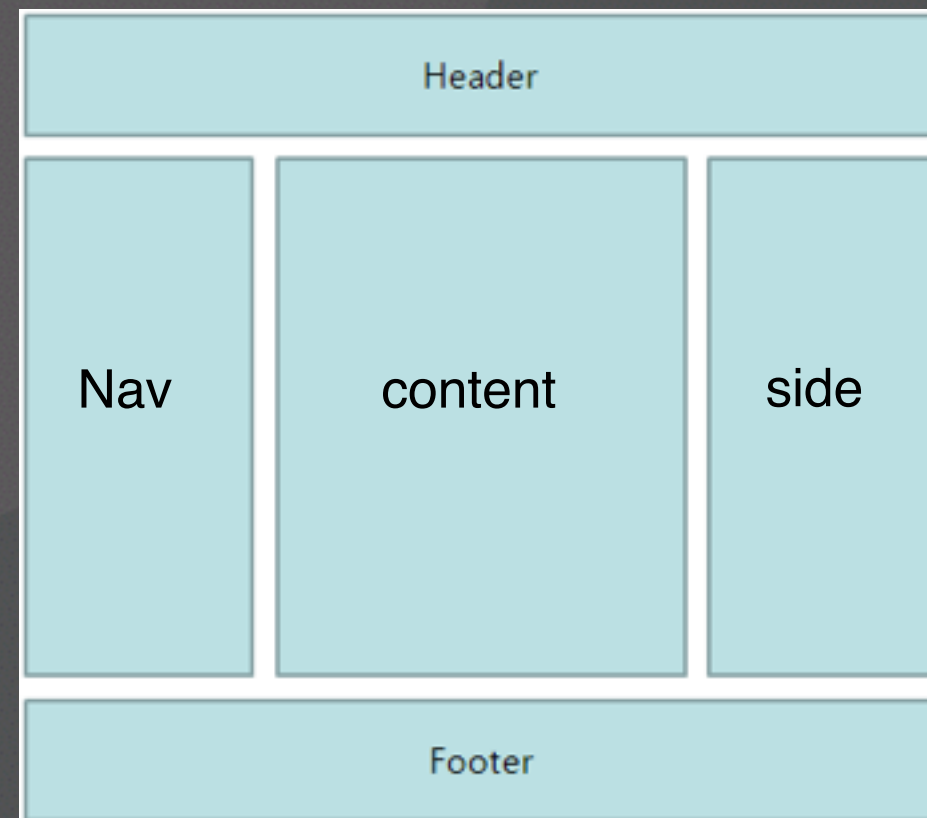


Layout

Layout ?

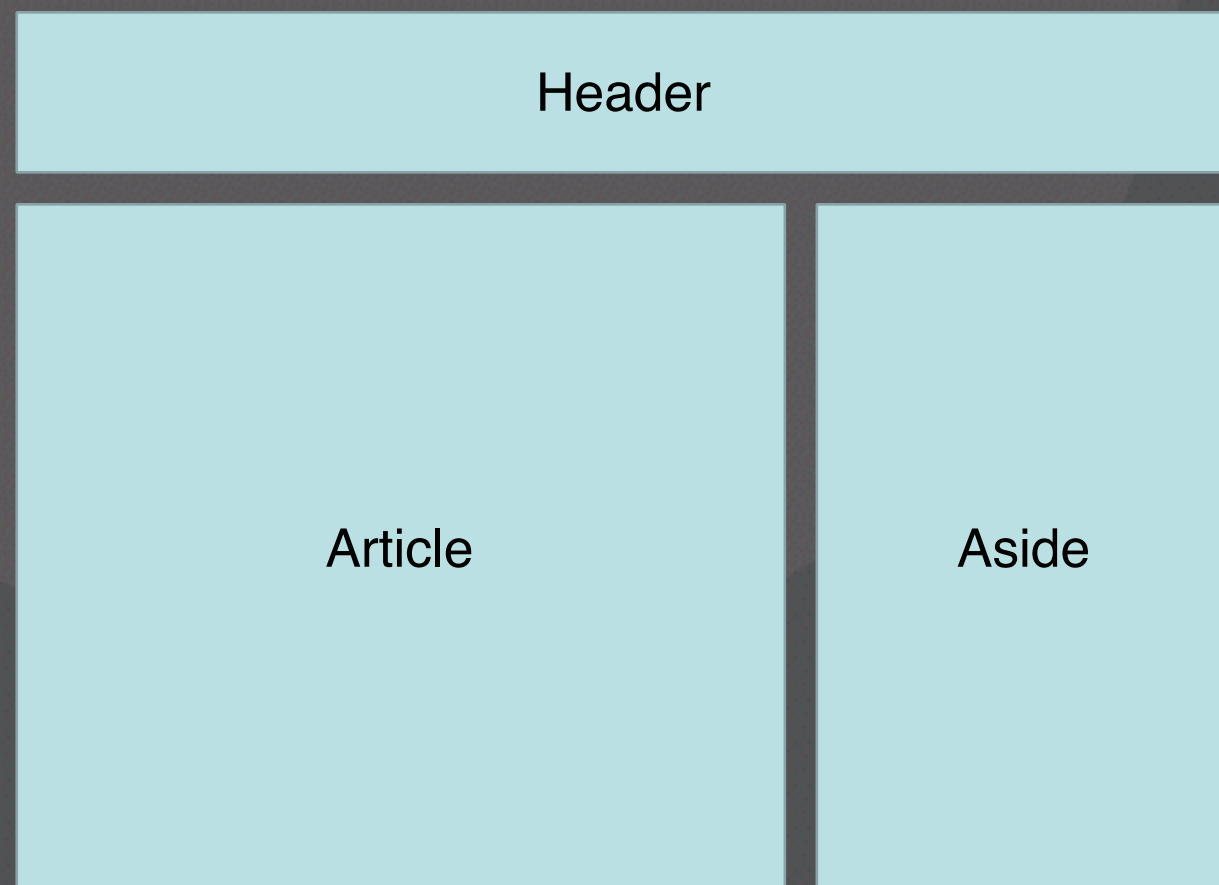
사전적의미 : 책이나 신문, 잡지에서 글이나 그림등을 효과적으로 정리하고 배치하는 일.

- Header : 사이트정보, 메인메뉴, 로고이미지
- Nav : 메뉴 또는 링크 정보
- content : 본문 내용
- side : 본문 이외에 주변에 배치할 내용
- Footer : 사이트 정보 및 부가 정보



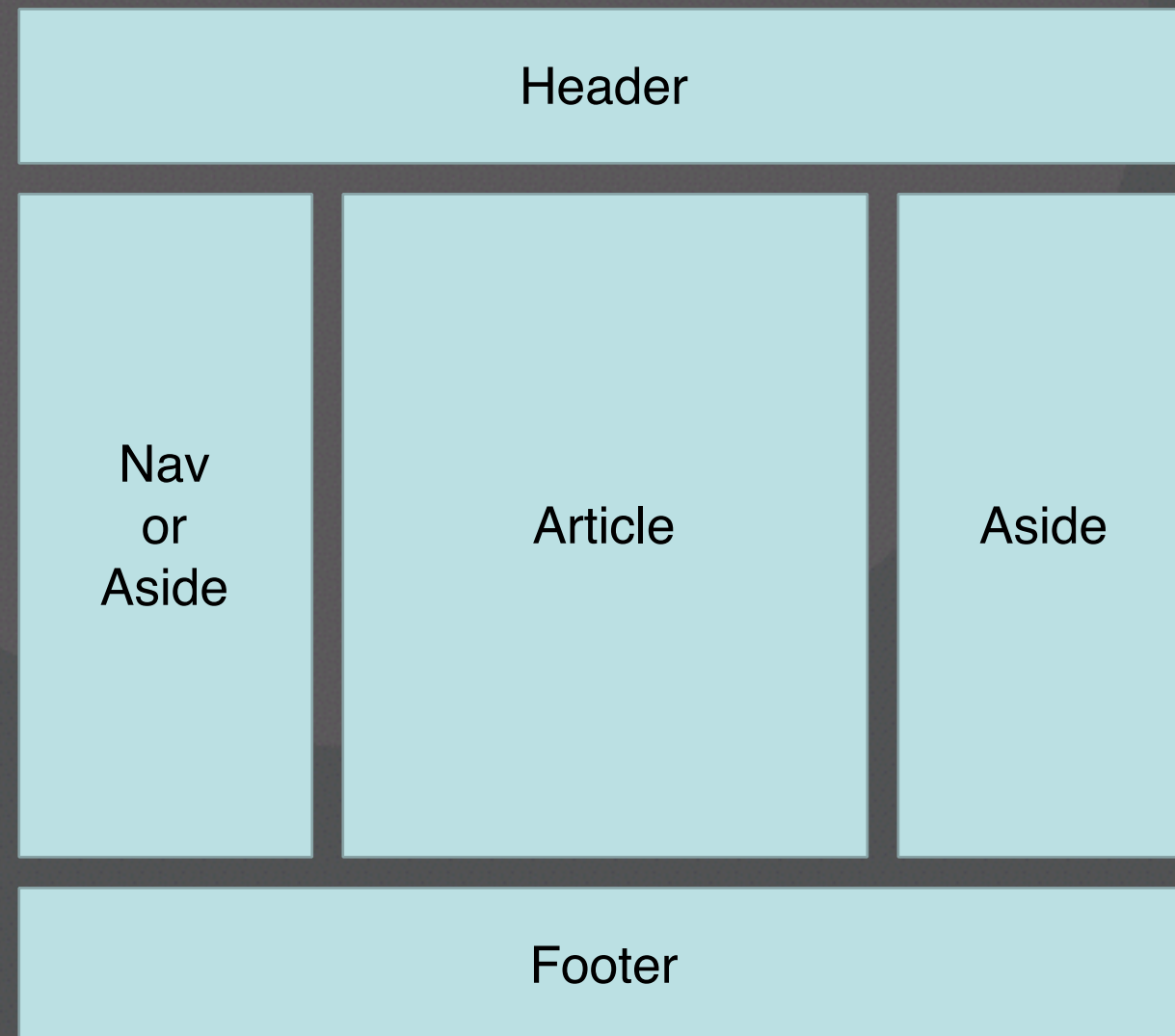
Float 활용

2단 레이아웃



Float 활용

3단 레이아웃

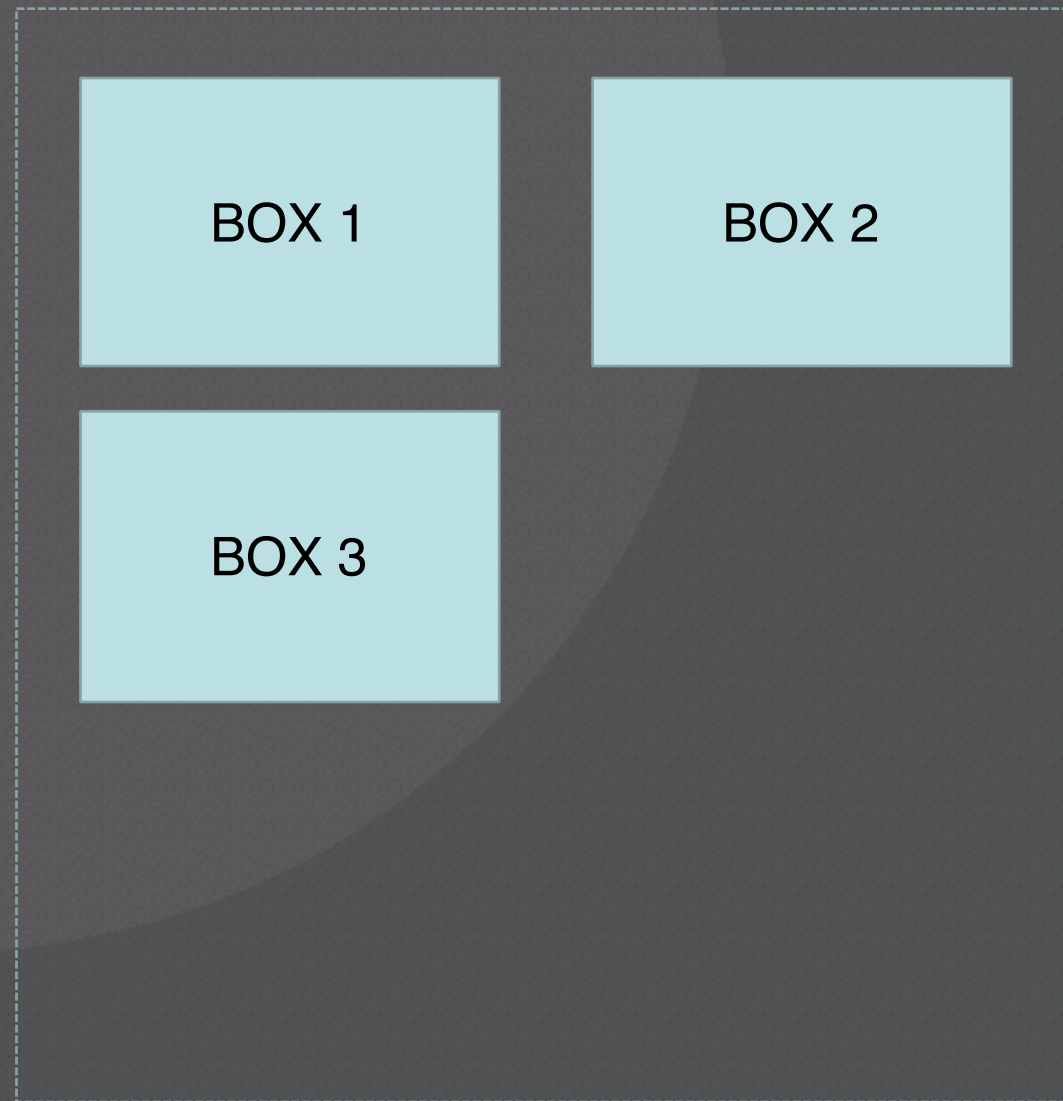


실습 - Float 활용 #ex1-1.html

BOX 1
float:left;

BOX 2
float:right;

BOX 3 은 잘보이는가?



CSS Layout



```
body { font:0.8em/1.5 verdana,AppleGothic,'굴림','돋움',gulim,dotum,sans-serif;}
#accessibility { float:right; }
#header { clear:both; }
#header h1 { float:left;display:block; }
#nav { float:right; }
#middle { clear:both; }
#nav ul li {display:inline;}
#content { float:left;width:70%; }
#sidebar { float:right;widht:25%; }
#footer { clear:both; }
#footertext { float:right; }
```

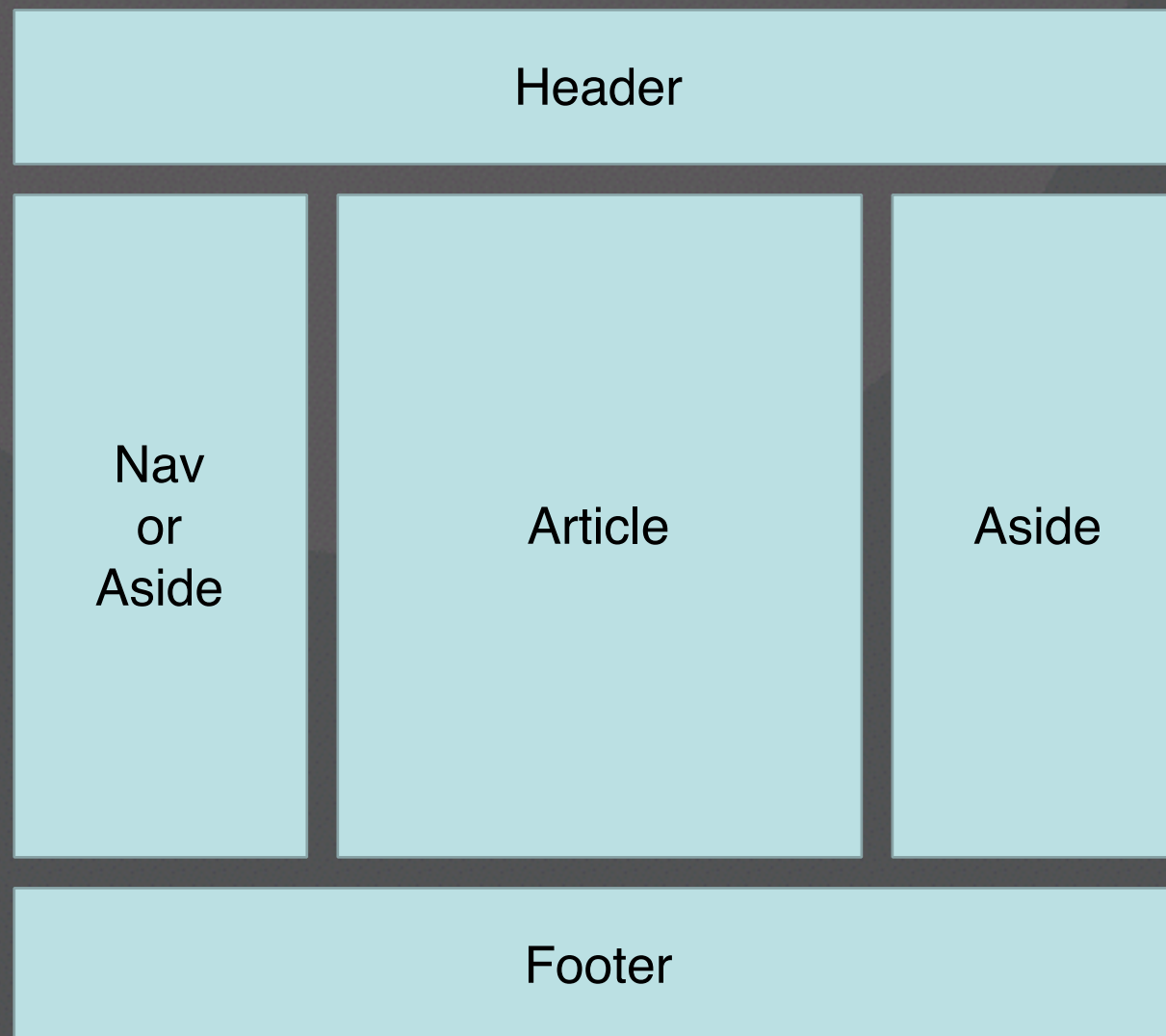
Float 해제 방법 #ex1-1.html 이어서

Float 된 요소 다음요소에 영향이 끼치지 않도록 하기위한 방법

- clear:both;
- overflow:hidden;
- display:inline-block;
- clear 속성을 갖고있는 빈 엘리먼트 추가
- 같이 float 를 하는 방법
- :after 가상선택자 (ie6, ie7 지원하지 않음, 핵사용 해야됨)

Position 활용 #position3.html

3단 레이아웃(높이가 유동적이지 않을경우)



실습 - Position 활용 #ex1-2.html

마크업 순서 파악 (제목 → 내용 → 더보기)

‘더보기’ 링크가 아래 이미지의 ‘더보기’ 위치와 비슷한 위치에 오게 하시오

전문가 칼럼[더보기 >](#)

- 뒷모습을 좌지우지하는 테일 램프디자..
- 시대에 맞는 유가 정책이 필요한 시점..
- 자동차의 인상을 좌우하는 헤드램프..
- 세계는 지금... ‘블루(Blue)’ 시대..
- 뒷모습을 좌지우지하는 테일 램프디자..
- 시대에 맞는 유가 정책이 필요한 시점..
- 자동차의 인상을 좌우하는 헤드램프..
- 세계는 지금... ‘블루(Blue)’ 시대..

Hint. 사용되는 속성

- position
- top
- right

실습

1. HTML 시멘틱 마크업 실습

찜 베스트

[+더보기](#)

1



일이삼사오육칠팔구십

123,456,789원

2

[디코랜드]헬로키티 자동차

3

[아발론]보행기 튜브

4

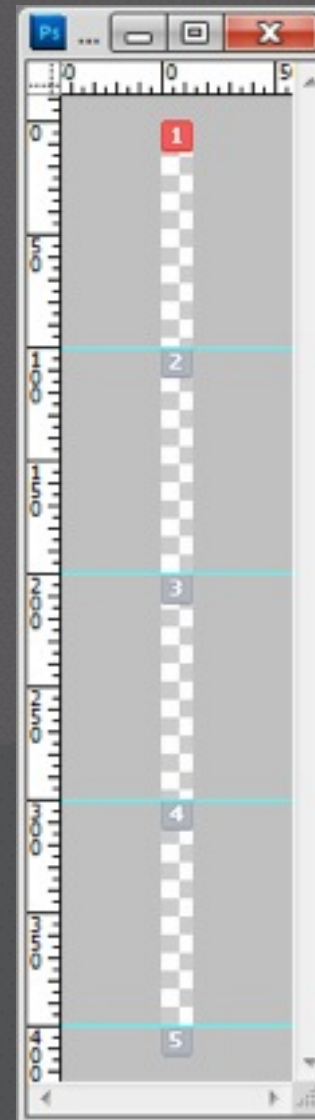
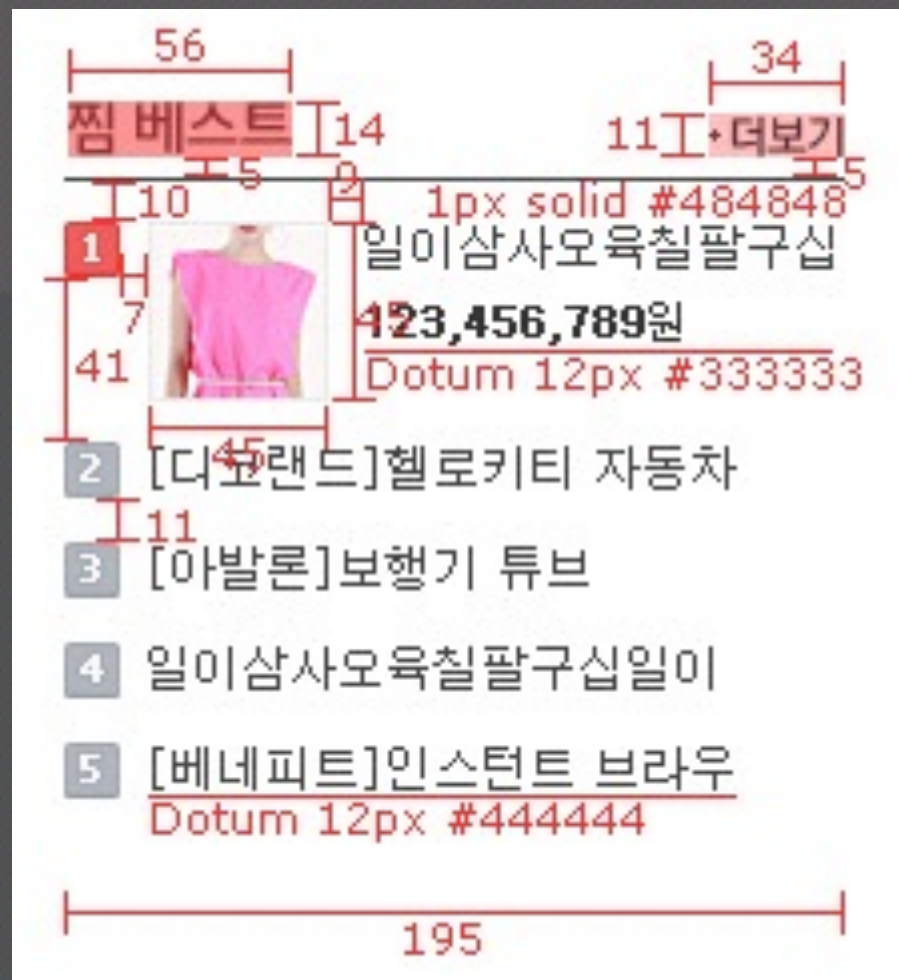
일이삼사오육칠팔구십일이

5

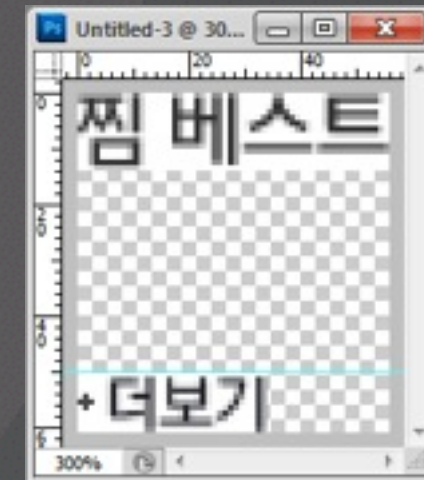
[베네피트]인스턴트 브라우

실습

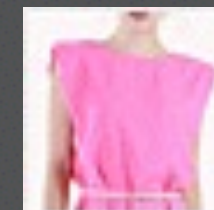
2. CSS적용 실습



background:url(num_sprites.gif)



background:url(txt_sprites.gif)



thumb.jpg

Cross Browsing

Cross browsing?

- 표준 웹기술을 이용하여 Vendor 호환성을 충족하는 기술
- 브라우저 제품별 특성과 버그를 해결하는 과정에서 예측 불가능한 문제를 해결하기 위해 시간이 소요됨
- 특히, CSS에 대해서는 브라우저에 따라 지원하는 정도가 달라 웹표준을 준수하더라도 일부 브라우저에서 적절하게 렌더링되지 않을 수 있음



Cross browsing?

IE 8 beta



Debug

CSS 문법오류

→ W3C CSS Validator : <http://jigsaw.w3.org/css-validator/>

더블마진 플로트 버그 (IE 6)

→ `display:inline;` 지정하여 해결

float를 지정한 엘리먼트 주변 텍스트 3픽셀 공간 버그 (IE 6)

→ 텍스트관련 엘리먼트를 함께 float 시킨다.

float된 엘리먼트가 여러 개 있을 때 IE 6의 문자 중복 표시 버그 (IE 6)

→ 주석 제거



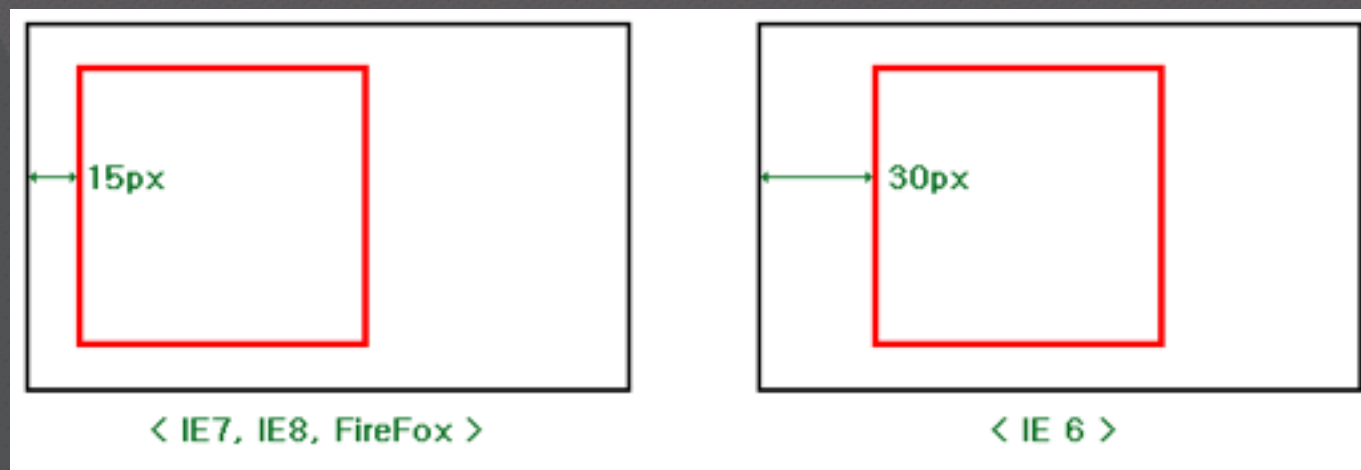
Debug

CSS 문법오류

→ W3C CSS Validator : <http://jigsaw.w3.org/css-validator/>

더블마진 플로트 버그 (IE 6)

→ `display:inline;` 지정하여 해결



float를 지정한 엘리먼트 주변 텍스트 3픽셀 공간 버그 (IE 6)

→ 텍스트관련 엘리먼트를 함께 float 시킨다.

float된 엘리먼트가 여러 개 있을 때 IE 6의 문자 중복 표시 버그 (IE 6)

→ 주석 제거

Debug

CSS 문법오류

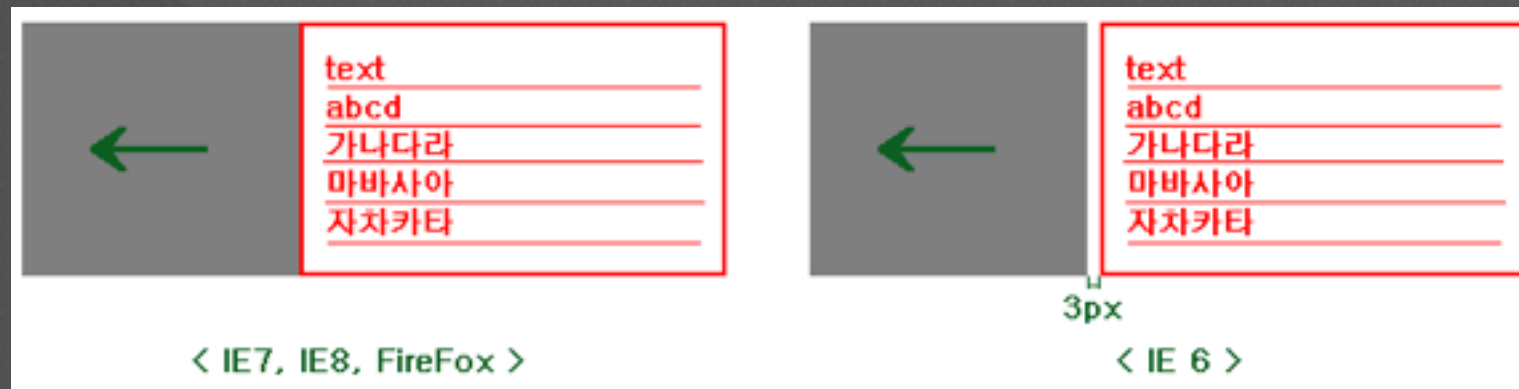
→ W3C CSS Validator : <http://jigsaw.w3.org/css-validator/>

더블마진 플로트 버그 (IE 6)

→ `display:inline;` 지정하여 해결

float를 지정한 엘리먼트 주변 텍스트 3픽셀 공간 버그 (IE 6)

→ 텍스트관련 엘리먼트를 함께 float 시킨다.



float된 엘리먼트가 여러 개 있을 때 IE 6의 문자 중복 표시 버그 (IE 6)

→ 주석 제거

Debug

CSS 문법오류

→ W3C CSS Validator : <http://jigsaw.w3.org/css-validator/>

더블마진 플로트 버그 (IE 6)

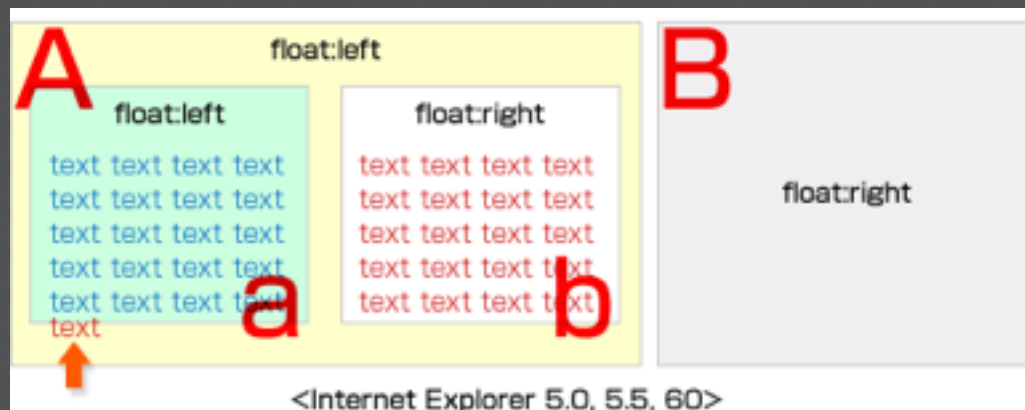
→ `display:inline;` 지정하여 해결

float를 지정한 엘리먼트 주변 텍스트 3픽셀 공간 버그 (IE 6)

→ 텍스트관련 엘리먼트를 함께 float 시킨다.

float된 엘리먼트가 여러 개 있을 때 IE 6의 문자 중복 표시 버그 (IE 6)

→ 주석 제거



누구도 생각하지 않았던 아이디어로 세상을 더 즐겁게
변화시키고, 누구에게나 편리함과 새로움을 제공하여
모든 사람들이 희망하는 일상생활의 바람을 충족시켜
서 또 다른 희망을 꿈꿀 수 있게 하여 그 꿈과 이상이 실현
될 수 있다는 믿음을 갖게 하는 것이
가치이며, 만들어 나아 갈 세상입니다.

감사합니다 

<http://mydeute.com> , mydeute@gmail.com