

5. 단위, 배경, 박스모델

1) 속성-정의와 구문

- 정의 : 해당 속성이 어떤 변화를 일으키고 어떻게 동작하는지 파악할 수 있습니다. 기본값, 상속 여부, 애니메이션 가능 여부, 사용 가능한 CSS 버전

- 문법 : 해당 속성값을 어떤 식으로 나열해 사용하는지를 파악할 수 있습니다.

- 속성값 : 해당 속성이 인식해 적용할 수 있는 값의 형태나, 키워드 등을 파악할 수 있습니다.

Initial : 초기값, 즉 속성의 기본값으로 정의 (ie에서 지원하지 않음)

Inherit : 부모 요소의 해당 속성값을 적용 (상속 가능할 요소일 경우) 즉, 상속이 불가능한 속성일 경우에는 적용 되지 않습니다.

- 지원 범위 : 해당 속성이 정의에 맞게 동작 가능한 CSS 버전, 브라우저별 버전을 확인할 수 있습니다. 어떤 브라우저의 어떤 버전이나에 따라 같은 값이어도 다르게 렌더링 될 수 있으므로, 현재 프로젝트의 사용자 제공 지원 범위를 잘 확인하고 적용해야 합니다.

- 예제 : 문법과 속성값을 바탕으로 실제로 속성을 동작하는 예제 코드를 확인할 수 있습니다.

- 참고사항 : 해당 속성에 대해 특이사항이나 버그에 대해서 확인할 수 있습니다.

<https://www.w3.org/>

<https://www.w3schools.com/>

<https://developer.mozilla.org/ko/>

2) 속성-단위

- 절대 길이 : 절대 길이는 고정된 크기 단위로, 다른 요소의 크기에 의해 영향을 받지 않습니다.

px (1px = 1/96th of 1 inch)

절대 길이이므로 다른 요소의 영향을 받지 않아 화면에서 고정된 크기를 가지지만, 장치의 해상도에 따라 상대적입니다. 여러 환경에서 디자인을 같게 표현하고 브라우저 호환성에 유리한 구조로 되어 있어서, 디자인 의도가 많이 반영된 웹 사이트의 경우 픽셀 단위를 사용하는 것을 권장하고 있습니다.

pt (1pt = 1/72 of 1 inch)

컴퓨터가 없던 시절부터 있던 단위입니다. 인쇄물이나 워드프로세서 프로그램에서 사용된 가장 작은 표준 인쇄 단위입니다. 웹 화면에 인쇄용 문서를 위한 스타일을 적용할 때 유용하게 사용할 수 있습니다.

그러나 사용하는 기기의 해상도에 따라 차이가 있어 W3C에서도 pt는 웹 개발 시 권장하는 단위가 아닙니다. 예를 들면 Windows에서는 9pt = 12px, Mac에서는 9pt = 9px로 보이게 됩니다.

- 상대 길이 : 상대 길이는 다른 요소의 크기나 폰트 크기, 브라우저(viewport) 등의 크기에 따라 상대적으로 값이 변합니다.

% : 부모의 값에 대해서 백분율로 환산한 크기를 갖게 됩니다.

em : font-size를 기준으로 값을 환산합니다. 소수점 3자리까지 표현 가능합니다.

rem : root의 font-size를 기준으로 값을 환산합니다.

vw : viewport의 width값을 기준으로 1%의 값으로 계산됩니다.

https://developer.mozilla.org/en-US/docs/Learn/CSS/Building_blocks/Values_and_units

https://www.w3schools.com/cssref/css_units.asp

3) 속성-색상

- Color 속성 : 폰트의 색상 값을 적용할 때 사용하는 속성입니다. h1{color: 색상 값;}

- 색상 값 지정 방식

컬러 키워드 : CSS 자체에서 사용 가능한 문자 식별자입니다. red, blue, black 등과 같이 미리 정의되어 있는 키워드를 이용해 색상을 표현할 수 있습니다. * 참고 : transparent는 투명성을 나타내는 키워드 *

16진법 ex. #RRGGBB : 6자리의 16진수(0-9, A-F)는 각각 두 자리씩 세 가지 색상을 나타냅니다. 첫 번

째 두 자리는 적색(RR), 가운데 두 자리는 녹색(GG), 마지막 두 자리는 청색(BB)을 의미합니다. 각 자리의 알파벳은 대소문자를 구분하지 않습니다.

16진법 ex. #RGB : 6자리의 16진수에서 각각의 두 자리가 같은 값을 가지면 3자리로 축약하여 사용할 수 있습니다. 예를 들어, #aa11cc 는 #a1c로 축약하여 사용할 수 있습니다.

RGB() : RGB 값은 rgb(R, G, B)의 형태로 각 변수 값(R 적색, G 녹색, B 청색)의 강도를 정의합니다. 0~255의 정수로 된 값을 지정할 수 있으며, 0 → 255는 검정 → 흰색으로 값의 변화를 나타냅니다.

RGBA() : RGBA 값은 기존 RGB에서 A값이 추가된 형태입니다. rgb(R, G, B, A)의 형태로 각 변수는(R 적색, G 녹색, B 청색, A 투명도)의 강도를 정의합니다. A 값은 0 ~ 1 사이의 값을 지정할 수 있으며, 0.5와 같이 소수점으로 표기합니다. 0 → 1은 투명 → 불투명으로 값의 변화를 나타냅니다. 예를 들어, rgba(0, 0, 0, 0)는 투명한 색상을 가지게 됩니다.

<코드실습>

<!DOCTYPE html>

<html lang="ko">

<head>

<meta charset="UTF-8">

<title>color</title>

</head>

<body>

<h1 style="color: red"> heading </h1>

<h1 style="color: #ff0000"> heading </h1>

<h1 style="color: #f00"> heading </h1>

<h1 style="color: rgb(255,0,0)"> heading </h1>

<h1 style="color: rgba(255,0,0, 0.5)"> heading </h1>

</body>

</html>

https://developer.mozilla.org/en-US/docs/Web/CSS/color_value

https://www.w3schools.com/css/css3_colors.asp

4) 속성-background

- background 관련 속성

background-color :

기본값 : transparent

배경의 색상을 지정하는 속성입니다. 앞선 색상 강의에서 배운 색상 값 적용 방식과 같습니다.

background-image :

기본값 : none

배경으로 사용할 이미지의 경로를 지정하는 속성입니다. url의 경로는 절대 경로, 상대 경로 모두 사용 가능합니다. 만약 background-color에 색상이 적용된 상태에서 background-image로 사용된 이미지에 투명한 부분이 있다면, 그 부분에 background-color 색상이 노출됩니다.

background-repeat :

기본값 : repeat

이미지의 반복 여부와 방향을 지정하는 속성입니다. 기본값이 repeat이기 때문에 따로 설정하지 않으면 x, y축으로 반복되어서 표시됩니다. background-repeat의 값으로 사용할 수 있는 것들은 다음과 같습니다.

<속성값>

repeat : x, y축으로 모두 반복합니다.

repeat-x : x축 방향으로만 반복합니다.
repeat-y : y축 방향으로만 반복합니다.
no-repeat : 이미지를 반복하지 않습니다.

background-position :

기본값 : 0% 0%

요소에서 배경 이미지의 위치를 지정하는 속성입니다. x축, y축으로부터의 위치를 지정할 수 있으며, 값의 선언 순서는 x축, y축으로부터의 간격입니다. 만일 한쪽만 지정된다면 나머지는 중앙값(center)으로 적용됩니다.

<속성값>

% : 기준으로부터 % 만큼 떨어진 지점과 이미지의 % 지점이 일치하는 곳에 위치시킵니다.

px : 기준으로부터 px 만큼 떨어진 지점과 이미지의 (0,0) 지점이 일치하는 곳에 위치시킵니다.

키워드 : top, left, right, bottom, center 키워드를 사용할 수 있습니다. 키워드는 선언 순서와 관계없이 top, bottom은 y축 기준으로 하며 left, right는 x축을 기준으로 합니다.

background-attachment :

기본값 : scroll

화면 스크롤에 따른 배경 이미지의 움직임 여부를 지정하는 속성입니다.

<속성값>

scroll : 배경 이미지는 요소 자체를 기준으로 고정되어 있으며 내용과 함께 스크롤 되지 않습니다.

local : 배경 이미지는 요소의 내용을 기준으로 고정되어 있으며 내용과 함께 스크롤 됩니다.

fixed : 배경 이미지는 뷰포트를 기준으로 고정되어 있으며 스크롤에 영향을 받지 않습니다.

- 뷰포트 : 사용자가 시각적으로 볼 수 있는 웹페이지 영역을 의미합니다. 컴퓨터나 휴대폰과 같은 장치에 Display 요소가 표현되는 영역을 말합니다.

background 축약

background: [-color] [-image] [-repeat] [-attachment] [-position];

<코드실습>

<!DOCTYPE html>

<html lang="ko">

<head>

<meta charset="UTF-8">

<title>background</title>

<style>

div {

height: 500px;

background-color: yellow;

background-image: url(https://www.w3schools.com/CSSref/img_tree.gif);

background-repeat: no-repeat;

background-position: center top;

/* 축약형 */

background: yellow url(https://www.w3schools.com/CSSref/img_tree.gif) no-repeat center

top;

}

</style>

</head>

<body>

```
<div> css background 속성 실습 </div>
</body>
</html>
https://www.w3schools.com/css/css\_background.asp
```

5) 속성-boxmodel

문서를 배치할 때 브라우저의 렌더링 엔진은 표준 CSS 기본 박스 모델에 따라 각 요소를 사각형 상자로 나타냅니다. CSS를 이용해 이 상자의 크기, 위치 및 속성(색상, 배경, 테두리 크기 등)을 변경할 수 있습니다.

- boxmodel 구성



Content 영역 : 요소의 실제 내용을 포함하는 영역입니다. 따라서 크기는 내용의 너비 및 높이를 나타냅니다.

Border 영역 : content 영역을 감싸는 테두리 선을 border라고 합니다.

Padding 영역 : content 영역과 테두리 사이의 여백을 padding이라고 합니다. content 영역이 배경, 색 또는 이미지가 있을 때 패딩 영역까지 영향을 미칩니다. 이에 따라 padding을 content의 연장으로 볼 수도 있습니다.

Margin 영역 : border 바깥쪽의 영역을 margin이라고 합니다. border 영역을 다른 요소와 구별하기 위해 쓰이는 빈 영역입니다. 즉, 주변 요소와의 여백(간격)을 margin을 이용해 지정할 수 있습니다.

<코드실습>

```
<!DOCTYPE html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <title>box model</title>
  <style>
    div {
      margin: 50px;
      padding: 50px;
      border: 10px solid #000;
    }
  </style>
</head>
<body>
  <div> 박스 모델에 대하여 알아보시다 </div>
</body>
```

</html>

https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Box_Model/Introduction_to_the_CSS_box_model

6) 속성-border

- border 관련 속성

border-width

기본값 : medium

선의 굵기를 지정하는 속성입니다. border-top-width, border-bottom-width, border-right-width, border-left-width를 이용하여 상하좌우 선의 굵기를 다르게 표현할 수 있습니다.

border-width: [top] [right] [bottom] [left];

<속성값>

키워드 : thin, medium, thick

단위 : px, em, rem ... (% , 정수 단위 사용불가)

border-style

기본값 : none 선의 모양을 지정하는 속성입니다. border-top-style, border-bottom-style, border-right-style, border-left-style를 이용하여 상하좌우 선의 모양을 다르게 표현할 수 있습니다.

border-style: [top] [right] [bottom] [left];

또한, 위처럼 축약하여 공백을 이용해 각 방향에 대한 스타일을 지정할 수도 있습니다.

<속성값>

none : border를 표시하지 않습니다.

solid : border를 실선 모양으로 나타냅니다.

double : border를 이중 실선 모양으로 나타냅니다.

dotted : border를 점선 모양으로 나타냅니다.

그 밖에도 dashed, double, groove, ridge, inset, outset 등의 다양한 스타일이 있습니다.

border- color

기본값 : currentColor 선의 색상을 지정하는 속성입니다. border-top-color, border-bottom-color, border-right-color, border-left-color를 이용하여 상하좌우 선의 색상을 다르게 표현할 수 있습니다.

border-color: [top] [right] [bottom] [left];

또한, 위처럼 축약하여 공백을 이용해 각 방향의 색상을 지정할 수도 있습니다. 색상은 일반적인 CSS 색상 값 사용 방식과 같습니다.

border 축약

border: [-width] [-style] [-color];

위와 같이 공백으로 구분해 축약하여 사용할 수 있고, 정의되지 않은 속성값에 대해서는 기본값이 적용됩니다.

<코드실습>

<!DOCTYPE html>

<html lang="ko">

<head>

<meta charset="UTF-8">

<title>border</title>

<style>

div {

border-width: 10px;

```
border-style: solid;
border-color: #000;
```

```
/* 축약형 */
border: 10px solid #000;
}
</style>
</head>
<body>
    <div> css border 속성 실습 </div>
</body>
</html>
```

https://developer.mozilla.org/en-US/docs/Web/CSS/color_value#currentcolor_keyword
https://www.w3schools.com/css/css_border.asp

7) 속성-padding

- padding 속성

기본값 : 0

<속성값>

length : 고정값으로 지정합니다. (ex. px, em)

percent : 요소의 width에 상대적인 크기를 지정합니다.

padding-top : content 영역의 위쪽 여백을 지정합니다.

padding-right : content 영역의 오른쪽 여백을 지정합니다.

padding-bottom : content 영역의 아래쪽 여백을 지정합니다.

padding-left : content 영역의 왼쪽 여백을 지정합니다.

- syntax

padding: [-top] [-right] [-bottom] [-left];

0 10px 20px 30px /* 상, 우, 하, 좌 다름 */

0 10px 20px /* 좌, 우 같음 */

0 10px /* 상, 하 같음 & 좌, 우 같음 */

0 /* 상, 우, 하, 좌 모두 같음 */

기본적으로 padding의 4가지 속성을 위와 같이 축약하여 사용할 수 있습니다. 속성의 순서는 고정되어있으며, 위쪽을 기준으로 시계방향으로 돌아간다고 생각하면 쉽습니다. 축약형으로 사용할 때 반드시 4개의 속성에 대한 값을 모두 적어야 하는 것은 아닙니다. 속성값은 1~4개의 값을 사용할 수 있으며 border에서는 축약형 사용 시 정의되지 않은 속성값에 대해서 기본값이 적용되었지만, padding은 조금 다른 방식으로 동작합니다. 그 이유는 상하, 좌우 영역의 값이 같을 때 하나로 합쳐서 적용할 수도 있기 때문입니다.

padding : 20px 30px 40px 30px 일 때, 좌우의 패딩 값이 같을 때 padding : 20px 30px 40px 와 같이 함축하여 사용할 수 있습니다.

padding : 20px 30px 20px 일 때, 좌우 패딩과 마찬가지로 상하의 패딩 값이 같을 때 padding : 20px 30px 와 같이 함축하여 사용할 수 있습니다.

padding : 20px 20px (= 20px, 20px, 20px, 20px)일 때, 상하좌우 패딩 값이 모두 같을 때 padding : 20px 와 같이 하나의 값으로 함축하여 사용할 수 있습니다.

* 참고 : CSS에서 0 값에 대해서는 단위를 따로 적지 않습니다. 0px = 0% = 0em = 0pt... => " 0 "

<코드실습>

```
<!DOCTYPE html>
```

```
<html lang="ko">
```

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <title>padding</title>
```

```
  <style>
```

```
    div {
```

```
      padding-top: 10px;
```

```
      padding-right: 20px;
```

```
      padding-bottom: 40px;
```

```
      padding-left: 20px;
```

```
      /* 축약형 */
```

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

```
    }
```

```
  </style>
```

```
</head>
```

```
<body>
```

```
  <div> css padding 속성 실습 </div>
```

```
</body>
```

```
</html>
```

<https://developer.mozilla.org/en-US/docs/Web/CSS/padding>

8) 속성-margin

- margin 속성

기본값 : 0

<속성값>

length : 고정값으로 지정합니다. (ex. px, em)

percent : 요소의 width에 상대적인 크기를 지정합니다.

auto : 브라우저에 의해 계산된 값이 적용됩니다.

margin-top : border 영역의 위쪽 여백을 지정합니다.

margin-right : border 영역의 오른쪽 여백을 지정합니다.

margin-bottom : border 영역의 아래쪽 여백을 지정합니다.

margin-left : border 영역의 왼쪽 여백을 지정합니다.

- syntax

margin: [-top] [-right] [-bottom] [-left];

0 10px 20px 30px /* 상, 우, 하, 좌 다름 */

0 10px 20px /* 좌, 우 같음 */

0 10px /* 상, 하 같음 & 좌, 우 같음 */

0 /* 상, 우, 하, 좌 모두 같음 */

padding과 마찬가지로 축약하여 사용할 수 있고, 상하, 좌우에 대해서 값이 같으면 하나로 합하여 사용

할 수 있습니다. margin에서는 수치 이외에 사용할 수 있는 'auto' 값이 있습니다.

margin auto : 기본적으로 브라우저에 의해 계산이 이루어지는데, 대부분의 경우 0(기본값) 또는 요소의 해당 측면에서 사용 가능한 공간과 같은 값을 가집니다. 이를 활용하여 수평 중앙 정렬을 할 수 있습니다. 아래 코드를 살펴봅시다.

margin-left: auto;

margin-right: auto;

좌우의 margin이 모두 auto로 적용되었다면, 브라우저는 요소가 가질 수 있는 가로 영역에서 자신의 width를 제외한 나머지 여백에 크기에 대해 균등 분할 하여 적용합니다. 이에 따라 요소는 수평 중앙 정렬이 됩니다. 상하의 경우 수직 중앙 정렬이 되지 않으며, 기본적인 플로우를 벗어나는 상황에 대해서 적용이 됩니다. 이는 좀 더 심화적인 개념이 필요하므로 기초에서는 수평 정렬에 대한 개념을 확실히 잡아두시길 바랍니다.

- margin collapse(마진 병합) : 마진 병합은 인접한 두 개 이상의 수직 방향 박스의 마진이 하나로 합쳐지는 것을 의미합니다.

1. 두 요소가 상하로 인접한 경우: 위 요소의 하단 마진과 아래 요소의 상단 마진의 병합이 일어납니다.



2. 부모 요소와 첫 번째 자식 요소 또는 마지막 자식 요소

부모 요소의 상단 마진과 첫 번째 자식 요소의 상단 마진 병합이 일어납니다.

부모 요소의 하단 마진과 마지막 자식 요소의 하단 마진 병합이 일어납니다.

3. 내용이 없는 빈 요소의 경우: 해당 요소의 상단 마진과 하단 마진의 병합이 일어납니다.

마진 병합은 수직 방향으로만 이루어지며, 좌우에 대해서는 일어나지 않습니다. 마진 병합은 마진이 반드시 맞닿아야 발생하기 때문에 2, 3번째의 경우 패딩 및 보더가 없어야 합니다.

마진 병합의 잘못된 예 (X)	마진 병합의 올바른 예 (O)
	

마진 병합을 활용하여 첫 번째와 두 번째 컴포넌트의 조합이 다양한 경우 여백을 다르게 사용 가능
<코드실습>

```
<!DOCTYPE html>
```

```
<html lang="ko">
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<title>margin</title>
```

```
<style>
```

```
div {
```

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

```
margin-right: 20px;
```

```
margin-bottom: 10px;
```

```
margin-left: 20px;
```



```

        /* 축약형 */
        margin: 10px 20px ;
    }
</style>
</head>
<body>
    <div> css margin 속성 실습 </div>
</body>
</html>

```

https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Box_Model/Mastering_margin_collapsing

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

9) 속성-margin&padding

- margin과 padding의 비교

	+	-	auto	단위
margin	o	o	o	px, %...
padding	o	x	x	px, %...

음수값 사용 가능 여부 : 왜 margin은 음수 값 적용이 가능하고, padding은 적용되지 않을까요? 예를 들어 생각해보자면, padding은 뼈와 우리 피부 사이의 지방이라고 생각하고, margin은 사람과 사람 사이의 간격이라고 생각하면 쉽습니다. 지방은 아무리 뺀다고 해서 피부가 뼈보다 밑으로 갈 수 없을 뿐만 아니라, 0 이하가 될 수 없으므로 양수만 된다고 생각하면 됩니다. 그러나 사람과 사람 사이는 멀리 떨어질 수도 있지만, 서로 겹쳐서 서 있을 수도 있으므로 음수 값이 가능하다고 생각하면 됩니다.

%값의 사용과 기준점 : css 속성을 사용하면서 어떤 값을 적용할 때 이 단위를 적용 할 수 있을까? 라는 생각을 가지고 코딩하는 자세는 매우 중요합니다. margin과 padding은 px과 같은 고정적인 단위 외에도 %라는 상대적인 단위를 사용 할 수 있습니다. %는 요소의 크기를 기준으로 상대적인 값을 결정짓게 됩니다. 얼핏 생각하면, 상하는 height 값에 대해 좌우는 width 값에 대해 크기가 계산될 거 같지만 그렇지 않습니다. %는 상하좌우의 방향에 관계 없이 모두 요소의 width 값을 기준으로 값이 결정됩니다.

```

div {
    width: 100px;
    height: 200px;
    margin: 10%;
    padding: 10%;
}

```

만약 위와 같은 코드의 경우에는 margin과 padding이 모두 20px 10px 20px 10px으로 적용되는 것이 아니라, 10px 10px 10px 10px 값으로 적용됩니다.

```

<코드실습>
<!DOCTYPE html>
<html lang="ko">
<head>
    <meta charset="UTF-8">
    <title>margin & padding</title>
    <style>
        div {

```

```

width: 300px;
height: 100px;
margin: -10px -10px 20px 30px;
padding: 10px 20px 30px 40px; /* padding 음수 사용 불가 */
}
</style>
</head>
<body>
  <div> css margin & padding 비교 실습 </div>
</body>
</html>

```

10) 속성 - width

- width 속성

기본값 : 0

<속성값>

auto : 브라우저에 의해 자동으로 계산하여 적용합니다. * 요소의 레벨 기본 특성에 따라 다르게 동작합니다.

length : 고정값으로 지정합니다. (ex. px, em)

percent : 부모 요소의 width에 상대적인 크기를 지정합니다.

width는 content 영역의 너비를 제어할 때 사용하는데 이때 auto가 아닌 특정한 값을 지정하여 사용하면, 그 크기는 box model 영역에서 margin 영역을 제외한 모든 영역에 대해 영향을 받습니다. (content, padding, border) 예를 들어,

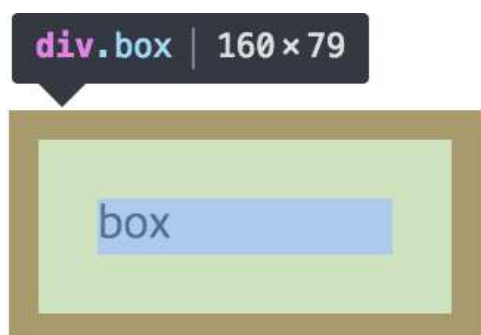
```
<div class="box"> box </div>
```

```

.box {
  width: 100px;
  padding: 20px;
  border: 10px solid black;
}

```

위와 같이 선언되어있다면 요소의 총 가로 크기는 160px가 됩니다. 분명 width: 100px를 적용했는데 width는 padding, border 영역에 대해서 영향을 받기 때문에 160px가 됩니다.



식으로 나타내면, 100px content + (20px * 2) padding + (10px * 2) border = 160px 가 된 것입니다. 또한, width는 %를 이용해서도 크기를 지정할 수 있습니다.

```

<div class="parent">
  <div class="child">

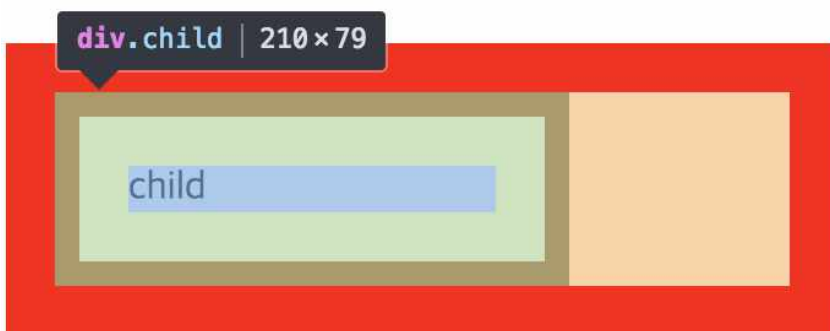
```

```

    child
  </div>
</div>

.parent {
  width: 300px;
  border: 20px solid red;
}
.child {
  width: 50%;
  padding: 20px;
  border: 10px solid black;
}

```



정답은 210px입니다. 우선, padding (20px * 2) + border (10px * 2) = 60px입니다. 210px - 60px = 150px 이며, 150px은 부모의 width가 300px이므로 300px의 50%인 150px이 width값으로 결정된 것입니다. 결국 %값일 때에는 부모의 width값에 대해서 계산되어지는데, 이때 부모의 width는 content 영역의 크기를 의미합니다. 부모의 padding과 border까지 더해진 요소의 전체 크기가 아닌, content 영역의 크기가 기준이라는 것을 기억하시면 됩니다.

<https://developer.mozilla.org/en-US/docs/Web/CSS/width>

11) 속성-height

- height 속성

기본값 : 0

<속성값>

auto : 브라우저 자동으로 계산하여 적용합니다. * 기본적으로 콘텐츠 영역의 내용만큼 높이를 가집니다.

length : 고정값으로 지정합니다. (ex. px, em)

percent : 부모 요소의 height에 상대적인 크기를 지정합니다. * 단, 부모 요소가 명시적으로 height 값이 있어야 합니다.

height는 content 영역의 높이를 제어할 때 사용하는데 이때 auto가 아닌 특정한 값을 지정하여 사용하면, width 속성과 마찬가지로 box model 영역에서 margin 영역을 제외한 모든 영역에 대해 영향을 받습니다. 예를 들어,

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

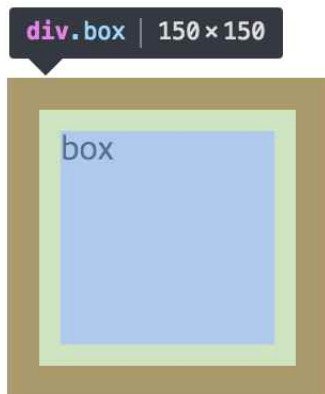
```

.box {
  width: 100px;
  height: 100px;
  padding: 10px;
}

```

```
border: 15px solid black;
}
```

위와 같이 선언되어있다면 요소의 총 세로 크기는 150px이 됩니다. 앞선 설명에서 언급한 바와 같이 height도 padding, border 영역에 대해서 추가로 영향을 받기 때문입니다.



식으로 나타내면, 100px content + (10px * 2) padding + (15px * 2) border = 150px 가 된 것입니다. %를 이용해서도 크기를 지정 할 수 있습니다.

```
<div class="parent">
  <div class="child">
    child
  </div>
</div>
```

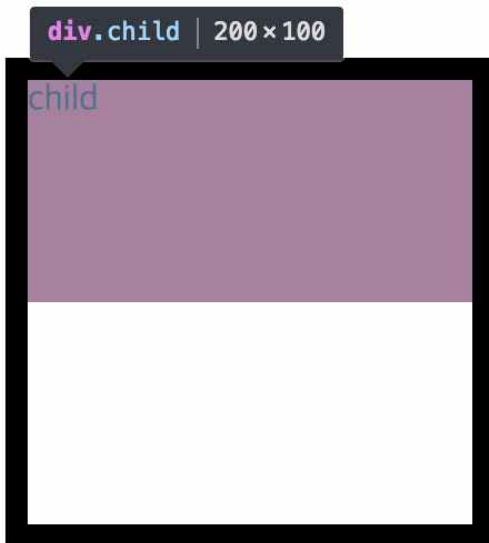
```
.parent {
  width: 200px;
  border: 10px solid black;
}
.child {
  height: 50%;
  background: red;
}
```



실제로 확인해보면, height: auto일 때와 height: 50%일 때 차이가 없는 것을 확인할 수 있습니다. 왜 두 값의 차이가 없는 걸까요? MDN에서 height 속성에 percent value에 대한 설명을 보면 "Containing Block의 높이에 대한 백분율로 높이를 정의합니다."고 나와 있습니다. 여기서 말하는 Containing Block은 부모를 의미한다고 생각하시면 됩니다. 즉, 현재 위의 코드에서는 부모가 명시적인 높이 값을 가지고 있지 않기 때문에 자식의 높이를 % 값으로 지정해줘도 적용되지 않았던 것입니다.

부모 코드에 height: 200px로 명시적으로 높이 값을 지정해주면, 위와 같이 %값이 적용되는걸 볼 수 있습니다.

```
<코드실습>
<!DOCTYPE html>
<html lang="ko">
```



```
<head>
  <meta charset="UTF-8">
  <title>height</title>
  <style>
    div {
      height: 300px;
      margin: 10px 20px;
      padding: 30px;
      border: 20px solid black;
    }
  </style>
</head>
<body>
  <div> css height 속성 실습 </div>
</body>
</html>
```

<https://developer.mozilla.org/en-US/docs/Web/CSS/height>

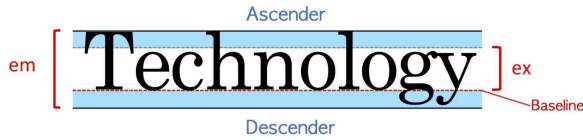
12) 속성-boxmodel정리

박스 모델은 content, padding, border, margin의 총 4가지 영역으로 나누어진다는 걸 배웠습니다. 추가적으로 content 영역의 너비는 width, 높이는 height를 통해서 제어 할 수 있으며, width와 height 그리고 padding과 border를 모두 더한 것이 요소의 전체 크기가 된다는 것에 대해 배웠습니다. 다른 요소와의 여백은 margin을 이용하여 줄 수 있으며 음수값을 사용할 수 있습니다. 또한, margin은 상하 요소 사이의 병합 현상이 일어나고 이때는 큰 값을 기준으로 병합된다는 걸 기억!!

6. 폰트, 텍스트

1) 속성-typography

- 타이포그래피의 구조 : 모든 폰트는 em박스를 가지고 있고 위 그림과 같은 구조로 이루어져 있습니다.



em : 폰트의 전체 높이를 의미

ex(= x-height) : 해당 폰트의 영문 소문자 x의 높이를 의미

Baseline : 소문자 x를 기준으로 하단의 라인을 의미

Descender : 소문자에서 baseline 아래로 처지는 영역을 의미, 서체에 따라 descender의 길이가 다릅니다. (g, j, p, q, y)

Ascender : 소문자 x의 상단 라인 위로 넘어가는 영역을 의미 (b, d, h, l)

<https://cssreference.io/typography/>

2) 속성-font-family

- font-family 속성 : 글꼴을 지정하는 속성

font-family : family-name | generic-family (| initial | inherit);

family-name: 사용할 폰트의 이름을 나타내며 ' , ' 로 구분하여 여러 개 선언할 수 있습니다. 먼저 선언된 순서대로 우선순위가 결정됩니다. 이름 중간에 공백이 있거나, 한글일 경우 홑따옴표로 묶어서 선언합니다.

generic-family : family-name으로 지정된 글꼴을 사용할 수 없을 경우를 대비해, 브라우저가 대체할 수 있는 폰트가 필요한 경우 선택할 수 있게 해줍니다. font-family 속성의 맨 마지막에 선언해야 하며, 키워드이기 때문에 따옴표 등의 인용부호로 묶지 않는 것이 원칙입니다. 예를 들면 아래와 같이 선언하여 사용할 수 있습니다.

font-family: Helvetica, Dotum, '돋움', Apple SD Gothic Neo, sans-serif;

가장 먼저 Helvetica를 사용하고, 이를 사용할 수 없을 때 Dotum을 사용하는 방식으로 우선순위에 따라 차례대로 적용 됩니다. 만약 "abc 가나다 123"이라는 글자가 있다면, "abc"와 "123"은 Helvetica로 표현이 되고, "가나다"는 Dotum으로 표현이 됩니다. "가나다"가 Dotum으로 표현된 이유는 Helvetica는 한글을 지원하는 폰트가 아니기 때문입니다. 그리고 예를 보면 돋움체를 영문으로 한번, 한글로 한번 선언 하였습니다. 한글을 지원하지 않는 디바이스일 경우 해당 한글 폰트를 불러올 수 없으므로 영문명으로도 선언해 주어야 합니다. 마지막에는 반드시 generic-family를 선언해 주어야 합니다. 그 이유는 선언된 모든 서체를 사용할 수 있다는 보장이 없기 때문입니다. 이때 generic-family를 선언해주면, 시스템 폰트 내에서 사용자가 의도한 스타일과 유사한 서체로 적용되기 때문입니다. 또한, 자식 요소에서 font-family를 재선언하면 부모에 generic-family가 선언되어있어도 다시 선언해주어야 합니다. Generic-Family에는 대표적인 서체로 serif, sans-serif가 있습니다. serif는 삐침이라는 뜻이고, sans는 프랑스어로 '~이 없이'라는 의미가 있습니다. serif는 글자 획에 삐침이 있는 폰트로 대표적으로 명조체가 있으며, sans-serif는 획에 삐침이 없는 폰트로 대표적으로 돋움체가 있습니다.

<https://developer.mozilla.org/ko/docs/Web/CSS/font-family>

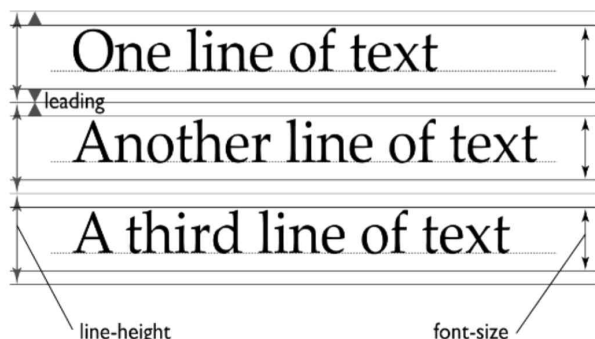
<https://qwerty.dev/>

3) 속성-line-height

- line-height 속성 : line-height는 텍스트 라인의 높이를 의미하는 것으로 주로 행간을 제어할 때 사용됩니다.

기본값 : normal

line-height: normal | number | length | initial | inherit ;



행간을 제어할 때 사용하는 속성이라 해서 줄 간격으로 생각해 오해하기 쉬울 수가 있습니다. 줄 바꿈이 되었을 때, 윗줄의 텍스트 하단과 아랫줄의 텍스트 상단까지의 간격이라고 생각할 수도 있지만, line-height로 제어되는 부분을 line-box라고도 하며 이는 타이포그래피 구조에서 배웠던 [em 박스] + [상하단의 여백]까지를 의미합니다.

<속성값>

normal : 기본값으로 브라우저의 기본 속성을 따릅니다. 폰트에 따라 브라우저에 따라 다르지만 보통 1.2 정도로 할당되어 있습니다.

number : font-size를 기준으로 설정한 숫자만큼 배율로 적용합니다.

length : px, em 등 고정 수치로 할당할 수 있습니다.

% : font-size를 기준으로 설정한 퍼센트만큼 배율로 적용합니다.

주의할 점은, line-height의 값으로 number를 선언할 때와 %로 선언할 때의 차이입니다. 두 값 모두 font-size를 기준으로 동작하기 때문에 1이나 100%를 같은 것이라고 오해할 수 있습니다. 하지만 두 값은 큰 차이가 있습니다. 바로 line-height의 값이 자식 요소로 상속되었을 때의 계산 방식입니다.

number : 부모 요소의 숫자 값이 그대로 상속됩니다. 즉, 자식 요소에서도 또 한 번 자식 요소의 font-size를 기준으로 계산된 값을 가집니다.

% : 부모 요소에서 %값이 그대로 상속되는 것이 아니고, %에 의해 이미 계산된 px값이 상속됩니다.

```
body { font-size: 20px; line-height: 2; } /* line-height = 40px; */
```

```
body { font-size: 20px; line-height: 200%; } /* line-height = 40px; */
```

두 경우 모두 <body>에 똑같이 line-height: 40px이 적용됩니다. 하지만 자식 요소로 <p>가 있다고 생각을 하면 얘기가 달라집니다.

```
body { font-size: 20px; line-height: 2; } /* line-height = 40px; */
```

```
p { font-size: 10px; } /* line-height = 20px; */
```

```
body { font-size: 20px; line-height: 200%; } /* line-height = 40px; */
```

```
p { font-size: 10px; } /* line-height = 40px; */
```

계산된 값이 아닌 숫자 값을 상속한다는 사실 때문에, 숫자 값을 사용하면 부모 엘리먼트에서 계산된 값 대신 비율을 그대로 상속받을 수 있으므로, 가능하면 단위가 없는 값을 사용하는 것이 좋습니다.

<코드실습>

```
<!DOCTYPE html>
```

```
<html lang="ko">
```

```
<head>
```

```

<meta charset="UTF-8">
<title>line-height</title>
<style>
    .parent {
        width: 200px;
        font-size: 10px;

        line-height: normal;
        line-height: 20px;
        line-height: 2;
        line-height: 200%;
    }

    .child {
        font-size: 20px;
    }
</style>
</head>

<body>
    <div class="parent">
        <div class="child">
            Lorem ipsum dolor sit amet consectetur adipisicing elit.
            Ipsam aspernatur vitae sapiente laudantium velit quo unde cupiditate autem,
            harum eaque natus perferendis ducimus saepe libero, voluptatibus voluptates possimus.
            Adipisci, delectus.
        </div>
    </div>
</body>
</html>

```

https://www.w3schools.com/cssref/pr_dim_line-height.asp

<https://wit.nts-corp.com/2017/09/25/4903>

<https://developer.mozilla.org/en-US/docs/Web/CSS/line-height>

4) 속성-font-size

- font-size 속성 : 글꼴의 크기를 지정하는 속성

기본값 : medium

font-size: keyword | length | initial | inherit ;

<속성값>

keyword : medium(기본 값), xx-small, x-small, small, large, x-large, xx-large, smaller, larger

length : px, em 등 고정 수치로 지정합니다.

% : 부모 요소의 font-size 기준의 퍼센트로 지정합니다.

absolute size (keyword) : 기본 값인 medium에 대한 상대적인 크기로, 브라우저마다 사이즈가 다르게 정의되어있습니다.

relative size (keyword) : 부모 요소의 font-size 크기에 대해 상대적입니다. smaller는 0.8배, larger는 1.2배입니다.

length : px, em, rem 등의 단위를 이용하여 고정된 크기를 지정할 수 있습니다. - em : 부모 요소의 font-size에 em 값을 곱한 크기 - rem : 루트의 font-size에 rem 값을 곱한 크기

percent (%) : 부모 요소의 font-size를 기준으로 백분을 계산된 값을 지정할 수 있습니다.

viewport units : vw, vh 단위로 뷰포트를 기준으로 하여, 유동적인 font-size를 지정할 수 있습니다. vw는 뷰포트 width의 1%, vh는 뷰포트 height의 1% 값을 가집니다.

<코드실습>

```
<!DOCTYPE html>
```

```
<html lang="ko">
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<title>font-size</title>
```

```
<style>
```

```
.parent {
```

```
    font-size: 20px;
```

```
}
```

```
.child {
```

```
    font-size: 1em;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<div class="parent">
```

```
<div class="child">
```

```
    Lorem ipsum dolor sit amet consectetur adipisicing elit.
```

```
    Ipsam aspernatur vitae sapiente laudantium velit quo unde cupiditate autem,
```

```
    harum eaque natus perferendis ducimus saepe libero, voluptatibus voluptates possimus.
```

```
    Adipisci, delectus.
```

```
</div>
```

```
</div>
```

```
</body>
```

```
</html>
```

https://www.w3schools.com/cssref/pr_font_font-size.asp

<https://developer.mozilla.org/ko/docs/Web/CSS/font-size>

5) 속성-font-weight

- font-weight 속성 : 글꼴의 굵기를 지정하는 속성

기본값 : normal

font-weight: normal | bold | bolder | lighter | number | initial | inherit ;

<속성값>

normal : 기본값 (400)

bold : 굵게 표현(700)

bolder : 부모 요소 보다 두껍게 표현

lighter : 부모 요소 보다 얇게 표현

number : 100, 200, 300, 400, 500, 600, 700, 800, 900 (클수록 더 두껍게 표현)

* 실무에서는 normal과 bold를 많이 사용하고, 부모 요소에 영향이 있는 bolder와 lighter는 사용될 수 있으면 지양하는 편입니다. 물론 상속 관계에서 바뀌어야 하는 스펙이라면 당연히 유용하게 사용될 수 있지만, 그 외의 경우에는 사용에 있어 신중해야 합니다. font-weight는 normal, bold와 같은 키워드 외에 숫자로도 그 굵기를 표현할 수 있습니다. 100~900까지 100단위로 값을 지정하여 사용할 수 있고 숫자가 커질수록 더욱 굵게 표현됩니다. 기본적으로 400은 normal과 같고, 700은 bold와 같습니다. 그러나 수치를 이용한 font-weight는 폰트 자체에서 지원을 해야 표현할 수 있습니다. 폰트에 따라 font-weight를 적용해도 굵기에 변화가 없을 수도 있으며, normal과 bold만 지원하는 폰트일 경우에는 100~500까지는 normal로, 600~900까지는 bold로 표현됩니다. 폰트가 점차 다양해지면서 굵기 자체를 폰트 family-name으로 가지고 있는 경우도 있습니다. 또한, 디바이스별로 조금 다르게 표현될 수도 있습니다. font-weight와 font-family, font-size는 서로 밀접하게 연관되어있습니다.

<코드실습>

```
<!DOCTYPE html>
```

```
<html lang="ko">
```

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <title>font-size</title>
```

```
  <link href="//fonts.googleapis.com/css?family=Open+Sans:400,600,700,800,300" rel="stylesheet" type="text/css">
```

```
  <style>
```

```
    body {
```

```
      padding: 0 20px;
```

```
      font-family: 'Open Sans';
```

```
    }
```

```
    .w100 { font-weight: 100; }
```

```
    .w200 { font-weight: 200; }
```

```
    .w300 { font-weight: 300; }
```

```
    .w400 { font-weight: 400; }
```

```
    .w500 { font-weight: 500; }
```

```
    .w600 { font-weight: 600; }
```

```
    .w700 { font-weight: 700; }
```

```
    .w800 { font-weight: 800; }
```

```
    .w900 { font-weight: 900; }
```

```
  </style>
```

```
</head>
```

```
<body>
```

```
  <p class="w100">This is 100 weight</p>
```

```
  <p class="w200">This is 200 weight</p>
```

```
  <p class="w300">This is 300 weight (available)</p>
```

```
  <p class="w400">This is 400 weight (available)</p>
```

```
  <p class="w500">This is 500 weight</p>
```

```
  <p class="w600">This is 600 weight (available)</p>
```

```
  <p class="w700">This is 700 weight (available)</p>
```

```
<p class="w800">This is 800 weight (available)</p>
<p class="w900">This is 900 weight</p>
</body>
</html>
https://www.w3schools.com/cssref/pr\_font\_weight.asp
https://codepen.io/impressivewebs/pen/EIncg
https://developer.mozilla.org/ko/docs/Web/CSS/font-weight
```

6) 속성-font-style

- font-style 속성 : 글꼴의 스타일을 지정하는 속성
기본값 : normal
font-style: normal | italic | oblique | initial | inherit;

This should be the Regular face

This should be the Italic face

This should be the Oblique face

<속성값>

normal : font-family 내에 분류된 기본값

italic : italic 스타일로 표현합니다.

oblique : oblique 스타일로 표현합니다.

oblique 텍스트의 기울기에 대한 각도를 추가로 지정할 수 있습니다.

font-weight oblique <각도>;

유효한 값은 -90 ~ 90도이며, 따로 각도를 지정하지 않으면 14도가 사용됩니다. 양수 값은 글의 끝부분 쪽으로 기울어지며, 음수값은 시작 부분 쪽으로 기울어집니다. 그러나 아직 초안 단계로 CSS Fonts Module Level 4를 지원하는 브라우저에서만 사용 가능합니다.

<https://developer.mozilla.org/ko/docs/Web/CSS/font-style>

https://www.w3schools.com/cssref/pr_font_font-style.asp

7) 속성-font-variant

- font-variant 속성 : 글꼴의 형태를 변형하는 속성으로 소문자를 작은 대문자로 변환할 수 있습니다.

기본값 : normal

font-variant: normal | small-caps | initial | inherit ;

<속성값>

normal : 기본값

small-caps : 소문자를 작은 대문자로 변형합니다.

<코드실습>

<!DOCTYPE html>

<html lang="ko">

<head>

<meta charset="UTF-8">

<title>font-variant</title>

<style>

```

    p {
        font-variant: small-caps;
    }
</style>
</head>

```

```

<body>
    <p>Font-Variant: Small-Caps</p>
</body>
</html>

```

https://www.w3schools.com/cssref/pr_font_font-variant.asp

<https://developer.mozilla.org/en-US/docs/Web/CSS/font-variant>

8) 속성-font

- font 관련 속성 : font-style, font-variant, font-weight, font-size/line-height, font-family 속성들을 한 번에 선언할 수 있는 축약형 속성입니다.

기본값 : 각 속성들의 기본값

font: font-style font-variant font-weight font-size/line-height font-family | initial | inherit;

<속성값>

font-style : font-style 지정, 기본 값은 normal

font-variant : font-variant 지정, 기본 값은 normal

font-weight : font-weight 지정, 기본 값은 normal

font-size/line-height : font-size/line-height 지정, 기본 값은 normal

font-family : font-family 지정

/* style | size | family */

font: oblique 2em "돋움", dotum, sans-serif;

/* style | variant | weight | size/line-height | family */

font: oblique small-caps bold 16px/1.5 '돋움';

/* The font used in system dialogs */

font: message-box;

font: icon;

축약형을 선언할 때 유의사항

font-size와 font-family는 반드시 선언해야 하는 필수 속성입니다.

빠진 속성이 있다면 기본값으로 지정됩니다.

각 속성의 선언 순서를 지켜야 합니다.

https://www.w3schools.com/cssref/pr_font_font.asp

<https://developer.mozilla.org/en-US/docs/Web/CSS/font>

9) 속성-webfont

- @font-face : 웹에 있는 글꼴을 사용자의 로컬 환경(컴퓨터)으로 다운로드하여 적용하는 속성입니다.

기본값 : 없음

@font-face {

```

font-properties
}
<속성값>
font-family(필수) : 글꼴의 이름을 지정
src(필수) : 다운로드 받을 글꼴의 경로(URL)

font-style(옵션) : 글꼴의 스타일 지정, 기본값은 normal
font-weight(옵션) : 글꼴의 굵기 지정, 기본값은 normal
@font-face {
    font-family: webNanumGothic; /* 사용자 지정 웹 폰트명 */
    src: url(NanumGothic.eot); /* 적용 될 웹 폰트의 경로 */
    font-weight: bold; /* 필요에 따라 지정 */
    font-style: italic; /* 필요에 따라 지정 */
}

```

```

body {
    font-family: webNanumGothic;
}

```

웹 폰트는 선언 시 필요에 따라 굵기나 스타일 등을 같이 지정해서 사용할 수 있습니다.

<https://wit.nts-corp.com/2017/02/13/4258>

<https://web.dev/fast/#optimize-webfonts>

10) 속성-vertical-align

- vertical-align 속성 : 요소의 수직 정렬을 지정하는 속성

기본값 : baseline

vertical-align: keyword | length | percent | initial | inherit ;

<속성값>

length : 요소를 지정한 길이만큼 올리거나 내림. 음수 허용

% : 요소를 line-height를 기준으로 올리거나 내림. 음수 허용

keyword : baseline(기본값), sub, super, top, text-top, middle, bottom, text-bottom

vertical-align은 기본값이 baseline입니다. 이전에 타이포그래피 구조 강의에서 설명했듯이 baseline은 소문자 x를 기준으로 하단 라인을 의미합니다.

keyword sub : 부모 아래 첨자 기준으로 정렬 super : 부모 위 첨자 기준으로 정렬 text-top : 부모 텍스트의 맨 위(Ascender 제외) text-bottom : 부모의 텍스트의 맨 아래(Descender 제외) middle : 부모의 중앙에 위치 top : 부모의 맨 위 위치 bottom : 부모의 맨 아래 위치

length : px값 사용 시 baseline을 기준으로 이동하며, 음수 값도 사용 가능합니다.

percent(%) : line-height를 기준으로 내에서 이동하며 음수 값 사용 가능합니다.

<코드실습>

```
<!DOCTYPE html>
```

```
<html lang="ko">
```

```
<head>
```

```
<meta charset="UTF-8">
```

```
<title>vertical-align</title>
```

```
<style>
```

```

p {
  padding: 10px;
  border: 1px dashed #aaa;
  line-height: 1;
  font-size: 16px;
}

p span {
  background-color: rgba(0, 255, 255, 0.5);
  border: 1px solid #aaa;
}

p span:nth-child(1) {
  background-color: rgba(255, 255, 0, 0.5);
}

p span:nth-child(2),
p span:nth-child(4) {
  font-weight: bold;
  font-size: 20px;
}

p span:nth-child(3) {
  background-color: rgba(0, 0, 0, 0.2);
}

/* table */

table {
  width: 100%;
  border-collapse: collapse;
}

table td,
table th {
  border: 1px solid #aaa;
  height: 50px;
}
</style>
</head>
<body>
  <h1>vertical-align</h1>
  <p><span>vertical-align:</span>
    <span style="vertical-align: baseline;">baseline;</span>

```

```

<span>---</span>
<span style="vertical-align: baseline;">수직정렬</span></p>
<p><span>vertical-align:</span>
  <span style="vertical-align: sub;">sub;</span>
  <span>---</span>
  <span style="vertical-align: sub;">수직정렬</span></p>
<p><span>vertical-align:</span>
  <span style="vertical-align: super;">super;</span>
  <span>---</span>
  <span style="vertical-align: super;">수직정렬</span></p>
<p>
  <span>vertical-align:</span>
  <span style="vertical-align: text-top;">text-top;</span>
  <span>---</span>
  <span style="vertical-align: text-top;">수직정렬</span></p>
<p><span>vertical-align:</span>
  <span style="vertical-align: text-bottom;">text-bottom;</span>
  <span>---</span>
  <span style="vertical-align: text-bottom;">수직정렬</span></p>
<p><span>vertical-align:</span>
  <span style="vertical-align: middle;">middle;</span>
  <span>---</span>
  <span style="vertical-align: middle;">수직정렬</span></p>
<p><span>vertical-align:</span>
  <span style="vertical-align: top;">top;</span>
  <span>---</span>
  <span style="vertical-align: top;">수직정렬</span></p>
<p><span>vertical-align:</span>
  <span style="vertical-align: bottom;">bottom;</span>
  <span>---</span>
  <span style="vertical-align: bottom;">수직정렬</span></p>
<p><span>vertical-align:</span>
  <span style="vertical-align: 4em;">4em;</span>
  <span>---</span></p>
<p><span>vertical-align:</span>
  <span style="vertical-align: 4px;">4px;</span>
  <span>---</span></p>
<p><span>vertical-align:</span>
  <span style="vertical-align: 20%;">20%;</span>
  <span>---</span></p>
<p><span>vertical-align:</span>
  <span style="vertical-align: -10px;">-10px;</span>
  <span>---</span></p>

```

```

<table>
  <caption>table cell vertical-align</caption>
  <thead>
    <tr>
      <th>속성값</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td style="vertical-align:baseline;">vertical-align:baseline</td>
    </tr>
    <tr>
      <td style="vertical-align:top;">vertical-align:top</td>
    </tr>
    <tr>
      <td style="vertical-align:middle;">vertical-align:middle(cell 기본값)</td>
    </tr>
    <tr>
      <td style="vertical-align:bottom;">vertical-align:bottom</td>
    </tr>
    <tr>
      <td>null</td>
    </tr>
  </tbody>
</table>
</body>
</html>

```

https://www.w3schools.com/cssref/pr_pos_vertical-align.asp
<https://developer.mozilla.org/ko/docs/Web/CSS/vertical-align>

11) 속성-text-align

- text-align 속성 : 텍스트의 정렬을 지정하는 속성입니다.

기본값 : left (Right to Left 언어일 경우는 right)

text-align: left | right | center | justify | initial | inherit :

기본 값은 left이지만 경우에 따라 다릅니다. 문서의 방향이 LTR(Left To Right) 왼쪽에서 오른쪽 방향인 언어일 경우 left가 기본값이고, RTL(Right To Left) 로 오른쪽에서 왼쪽으로 읽힐 경우 right가 기본값이 됩니다.

<속성값>

left : 텍스트를 왼쪽으로 정렬

right : 텍스트를 오른쪽으로 정렬

center : 텍스트를 중앙으로 정렬

justify : 텍스트를 라인 양쪽 끝으로 붙여서 정렬. (마지막 라인은 정렬 하지 않음)

- text-align과 display의 관계 : text-align은 inline-level에 적용, text-align은 block-level에 적용할 수 없음

block 요소를 가운데 정렬 : 박스모델 챕터에서 다룬 margin의 auto
가운데 정렬 인라인 요소 : text-align (center) 블록 요소 : margin (auto)

<코드실습>

```
<!DOCTYPE html>
```

```
<html lang="ko">
```

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <title>text-align</title>
```

```
  <style>
```

```
    p {
```

```
      max-width: 630px;
```

```
      border: 1px solid #888;
```

```
      padding: 10px;
```

```
    }
```

```
  </style>
```

```
</head>
```

```
<body>
```

```
  <h1>text-align</h1>
```

```
  <h2>left</h2>
```

<p style="text-align: left;">이 안내서를 시작하기 전에, 사용법에 익숙한 텍스트 파일을 편집할 수 있는 편집 툴이 필요하며 최신 브라우저도 필요하다. 파일을 편집하고 싶지 않다면 그냥 안내서를 읽고 해당 예시 그림을 보라. 그러나 배우는데는 더딜 것이다.</p>

```
  <h2>right</h2>
```

<p style="text-align: right;">이 안내서를 시작하기 전에, 사용법에 익숙한 텍스트 파일을 편집할 수 있는 편집 툴이 필요하며 최신 브라우저도 필요하다. 파일을 편집하고 싶지 않다면 그냥 안내서를 읽고 해당 예시 그림을 보라. 그러나 배우는데는 더딜 것이다.</p>

```
  <h2>center</h2>
```

<p style="text-align: center;">이 안내서를 시작하기 전에, 사용법에 익숙한 텍스트 파일을 편집할 수 있는 편집 툴이 필요하며 최신 브라우저도 필요하다. 파일을 편집하고 싶지 않다면 그냥 안내서를 읽고 해당 예시 그림을 보라. 그러나 배우는데는 더딜 것이다.</p>

```
  <h2>justify</h2>
```

<p style="text-align: justify;">이 안내서를 시작하기 전에, 사용법에 익숙한 텍스트 파일을 편집할 수 있는 편집 툴이 필요하며 최신 브라우저도 필요하다. 파일을 편집하고 싶지 않다면 그냥 안내서를 읽고 해당 예시 그림을 보라. 그러나 배우는데는 더딜 것이다.</p>

```
</body>
```

```
</html>
```

https://www.w3schools.com/cssref/pr_text_text-align.asp

<https://developer.mozilla.org/en-US/docs/Web/CSS/text-align>

12) 속성-text-indent

- text-indent 속성 : 텍스트의 들여쓰기를 지정하는 속성입니다.

기본값 : 0

text-indent: length | initial | inherit;

<속성값>

length : px, em 등 고정 수치로 지정. 음수 허용

% : 부모 요소의 width를 기준으로 퍼센트로 지정

length : 문단의 첫 줄에 대한 들여쓰기를 수행합니다. 음수 값을 사용할 수 있으며, 음수 값 사용 시 왼쪽으로 이동합니다.

percent(%) : 텍스트를 포함하는 컨테이너 블록의 width(부모의 width)를 기준으로 변환된 백분율 값으로 들여쓰기합니다.

<코드실습>

```
<!DOCTYPE html>
```

```
<html lang="ko">
```

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <title>text-indent</title>
```

```
</head>
```

```
<body>
```

```
  <h1>text-indent</h1>
```

```
  <h2>length 1em</h2>
```

<p style="text-indent: 1em;">이 안내서를 시작하기 전에, 사용법에 익숙한 텍스트 파일을 편집할 수 있는 편집 툴이 필요하며 최신 브라우저도 필요하다. 파일을 편집하고 싶지 않다면 그냥 안내서를 읽고 해당 예시 그림을 보라. 그러나 배우는 데는 더딜 것이다.</p>

```
  <h2>length 40px</h2>
```

<p style="text-indent: 40px;">이 안내서를 시작하기 전에, 사용법에 익숙한 텍스트 파일을 편집할 수 있는 편집 툴이 필요하며 최신 브라우저도 필요하다. 파일을 편집하고 싶지 않다면 그냥 안내서를 읽고 해당 예시 그림을 보라. 그러나 배우는 데는 더딜 것이다.</p>

```
  <h2>percentage 15%</h2>
```

<p style="text-indent: 15%;">이 안내서를 시작하기 전에, 사용법에 익숙한 텍스트 파일을 편집할 수 있는 편집 툴이 필요하며 최신 브라우저도 필요하다. 파일을 편집하고 싶지 않다면 그냥 안내서를 읽고 해당 예시 그림을 보라. 그러나 배우는 데는 더딜 것이다.</p>

```
  <h2>length -40px (비추천)</h2>
```

<p style="text-indent:-40px;">이 안내서를 시작하기 전에, 사용법에 익숙한 텍스트 파일을 편집할 수 있는 편집 툴이 필요하며 최신 브라우저도 필요하다. 파일을 편집하고 싶지 않다면 그냥 안내서를 읽고 해당 예시 그림을 보라. 그러나 배우는 데는 더딜 것이다.
들여쓰기에 음수사용은 올바르지 않으며 본래의 기능에서 벗어남</p>

```
</body>
```

```
</html>
```

https://www.w3schools.com/cssref/pr_text_text-indent.asp

<https://developer.mozilla.org/en-US/docs/Web/CSS/text-indent>

<http://www.w3big.com/ko/cssref/pr-text-text-indent.html>

13) 속성-text-decoration

- text-decoration 속성 : 텍스트의 장식을 지정하는 속성입니다.

기본값 : none currentColor solid

text-decoration: text-decoration-line text-decoration-color text-decoration-style | initial | inherit;

text-decoration-line 텍스트 꾸밈의 종류를 지정하는 속성입니다.

기본값 : none

<속성값>

none : 텍스트 꾸밈을 생성하지 않음 (기본값)

underline : 밑줄로 꾸밈을 설정

overline : 윗줄로 꾸밈을 설정

line-through : 중간을 지나는 줄로 꾸밈을 설정

text-decoration-color

텍스트 꾸밈의 색상을 지정하는 속성입니다.

기본값 : currentColor

색상 값을 사용하여 원하는 색상을 지정할 수 있습니다.

text-decoration-style 꾸밈에 사용되는 선의 스타일을 지정하는 속성입니다.

기본값 : solid

<속성값>

solid : 한줄 스타일 (기본 값)

double : 이중선 스타일

dotted : 점선 스타일

dashed : 파선 스타일

wavy : 물결 스타일

<코드실습>

<!DOCTYPE html>

<html lang="ko">

<head>

<meta charset="UTF-8">

<title>text-decoration</title>

</head>

<body>

<h2>일반 경우</h2>

<p style="text-decoration: overline;">

text-decoration: overline;

</p>

<p style="text-decoration: underline;">

text-decoration: underline;

</p>

<p style="text-decoration: line-through;">

text-decoration: line-through;

</p>

<h2>부모 내 자식요소가 float될 경우 상속이 해제됨</h2>

text-decoration: overline;

<h2>부모 내 자식요소가 absolute 경우 상속이 해제됨</h2>


```

text-decoration: overline;</span>
</a>
</body>
</html>
https://www.w3schools.com/cssref/pr\_text\_text-decoration.asp
https://www.w3schools.com/cssref/css3\_pr\_text-decoration-line.asp
https://www.w3schools.com/cssref/css3\_pr\_text-decoration-color.asp
https://www.w3schools.com/cssref/css3\_pr\_text-decoration-style.asp
https://developer.mozilla.org/en-US/docs/Web/CSS/text-decoration

```

14) 속성-단어 관련 속성

- white-space 속성 : 요소 안에 공백을 어떻게 처리할지 지정하는 속성입니다.

기본값 : normal

white-space: normal | nowrap | pre | pre-line | pre-wrap | initial | inherit;

<속성값>

normal : 공백과 개행을 무시하고, 필요한 경우에 자동 줄 바꿈 발생. 기본값

nowrap : 공백과 개행을 무시하고, 자동 줄 바꿈이 일어나지 않음.

pre : 공백과 개행을 표현하고, 자동 줄 바꿈이 일어나지 않음.

pre-line : 공백은 무시하고, 개행만 표현. 필요한 경우에 자동 줄 바꿈 발생.

pre-wrap : 개행은 무시하고, 공백만 표현. 필요한 경우 자동 줄 바꿈 발생.

- letter-spacing 속성 : 자간을 지정하는 속성입니다.

기본값 : normal

letter-spacing: normal | length | initial | inherit;

<속성값>

normal : 기본값

length : 길이만큼 자간을 지정. 음수 허용

- word-spacing 속성 : 단어 사이의 간격을 지정하는 속성입니다.

기본값 : normal

word-spacing: normal|length|initial|inherit;

<속성값>

normal : 기본값

length : 길이만큼 단어 사이의 간격을 지정. 음수 허용

- word-break 속성 : 단어가 라인 끝에 나올 경우 어떻게 처리할지(중단점) 지정하는 속성입니다.

기본값 : normal

word-break: normal | break-all | keep-all | initial | inherit;

<속성값>

normal : 기본 값. 중단점은 공백이나 하이픈(-)(CJK는 음절)

break-all : 중단점은 음절. 모든 글자가 요소를 벗어나지 않고 개행

keep-all : 중단점은 공백이나 하이픈(-)(CJK는 그 외 기호도 포함)

- word-wrap 속성 : 요소를 벗어난 단어의 줄 바꿈을 지정하는 속성입니다.

기본값 : normal

word-wrap: normal|break-word|initial|inherit;

<속성값>

normal : 기본값. 중단점에서 개행

break-word : 모든 글자가 요소를 벗어나지 않고 강제로 개행

https://www.w3schools.com/cssref/pr_text_white-space.asp

https://www.w3schools.com/cssref/pr_text_letter-spacing.asp

https://www.w3schools.com/cssref/pr_text_word-spacing.asp

https://www.w3schools.com/cssref/css3_pr_word-break.asp

https://www.w3schools.com/cssref/css3_pr_word-wrap.asp

https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Text