HAPPY CSS CLASS

01

스타일의 상속과 적용 우선순위

- CSS3 Inheritance & Cascading -

1. 상속(Inheritance)

상속이란 상위(부모, 조상) 요소에 적용된 프로퍼티를 하위(자식, 자손) 요소가 물려 받는 것을 의미한다.

상속 기능이 없다면 각 요소의 Rule set에 프로퍼티를 매번 각각 지정해야 한다.

하지만 모든 프로퍼티가 상속되는 것은 아니다. 프로퍼티 중에는 상속이 되는 것과 되지 않는 것이 있다.

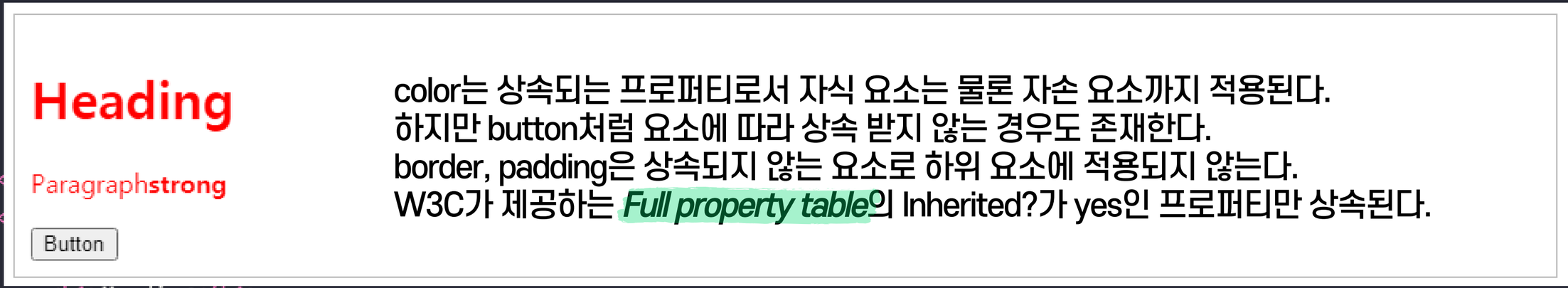
스타일의 상속과 적용 우선순위

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <style>
    .text-red {
      color: red;
      border: 1px solid #bcbcbc;
      padding: 10px;
    }
  </style>
</head>
<body>
  <div class="text-red">
    <h1>Heading</h1>
    <p>Paragraph<strong>strong</strong></p>
    <button>Button</button>
  </div>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <style>
    .text-red {
      color: red;
```


```
    <h1>Heading</h1>
    <p>Paragraph<strong>strong</strong></p>
    <button>Button</button>
  </div>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <style>
    .text-red {
      color: red;
```



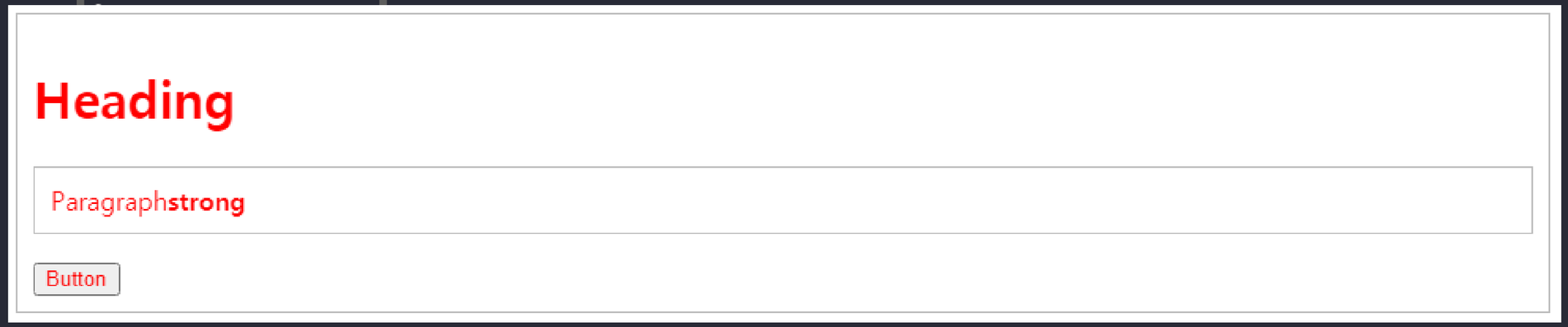

```
    <h1>Heading</h1>
    <p>Paragraph<strong>strong</strong></p>
    <button>Button</button>
  </div>
</body>
</html>
```

스타일의 상속과 적용 우선순위

```html
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <style>
    .text-red {
      color: red;
      border: 1px solid #bcbcbc;
      padding: 10px;
    }
    .text-red button {
      color: inherit;
    }
    .text-red p {
      border: inherit;
      padding: inherit;
    }
  </style>
</head>
<body>
  <div class="text-red">
    <h1>Heading</h1>
    <p>Paragraph<strong>strong</strong></p>
    <button>Button</button>
  </div>
</body>
</html>
```

스타일의 상속과 적용 우선순위

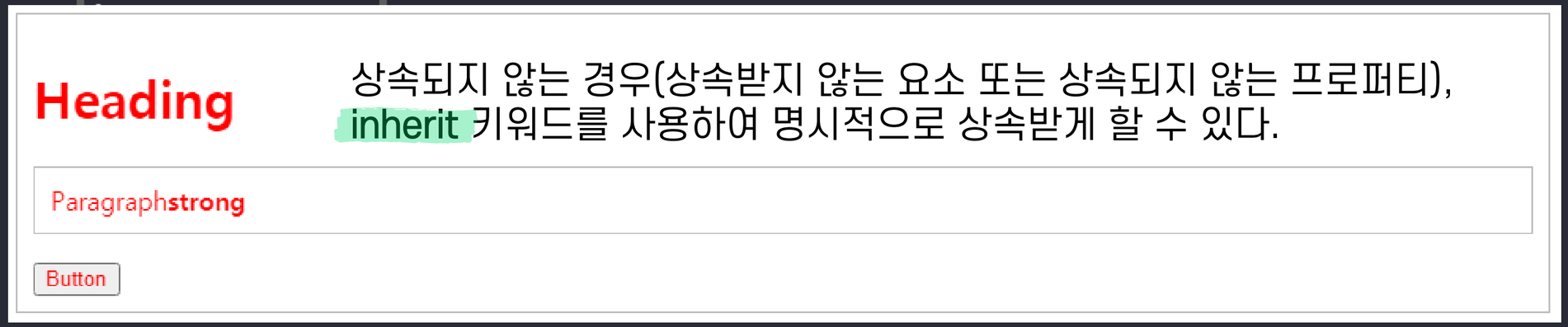
```html
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <style>
    .text-red {
      color: red;
      border: 1px solid #bcbcbc;
      padding: 10px;
```




```html
</head>
<body>
  <div class="text-red">
    <h1>Heading</h1>
    <p>Paragraph<strong>strong</strong></p>
    <button>Button</button>
  </div>
</body>
</html>
```

스타일의 상속과 적용 우선순위

```html
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <style>
    .text-red {
      color: red;
      border: 1px solid #bcbcbc;
      padding: 10px;
```



상속되지 않는 경우(상속받지 않는 요소 또는 상속되지 않는 프로퍼티),
inherit 키워드를 사용하여 명시적으로 상속받게 할 수 있다.

```html
</head>
<body>
  <div class="text-red">
    <h1>Heading</h1>
    <p>Paragraph<strong>strong</strong></p>
    <button>Button</button>
  </div>
</body>
</html>
```

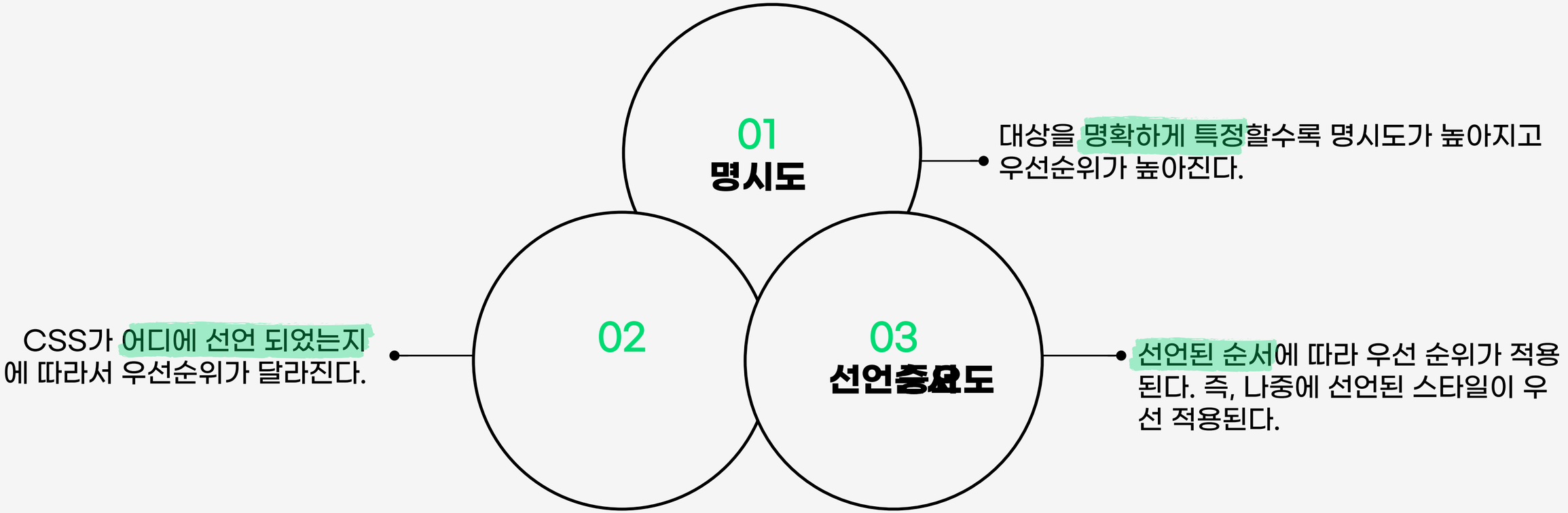
2. 캐스캐이딩(Cascading)

요소는 하나 이상의 CSS 선언에 영향을 받을 수 있다.

이때 충돌을 피하기 위해 CSS 적용 우선순위가 필요한데 이를 캐스캐이딩(Cascading Order)이라고 한다.

캐스캐이딩에는 다음과 같이 세가지 규칙이 있다.

스타일의 상속과 적용 우선순위



2.1 중요도

1. head 요소 내의 style 요소
2. head 요소 내의 style 요소 내의 @import 문
3. <link> 로 연결된 CSS 파일
4. <link> 로 연결된 CSS 파일 내의 @import 문
5. 브라우저 디폴트 스타일시트

캐스캐이딩(Cascading) - 중요도

```css
/* style.css */
body {
  background-color: red;
  color: white;
}
```

```html
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="style.css">
  <style>
    body {
      background-color: beige;
      color: navy;
    }
  </style>
</head>
<body>
  Lorem ipsum dolor sit amet, consectetur adipisicing elit,
  sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
</body>
</html>
```

캐스캐이딩(Cascading) - 중요도

```css
/* style.css */
body {
  background-color: red;
  color: white;
}
```

```html
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="style.css">
  <style>
    body {
      background-color: beige;
      color: navy;
    }
  </style>
</head>
<body>
  Lorem ipsum dolor sit amet, consectetur adipisicing elit,
  sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
</body>
</html>
```

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

2.2 명시도

대상을 명확하게 특정할수록 명시도가 높아지고 우선순위가 높아진다.

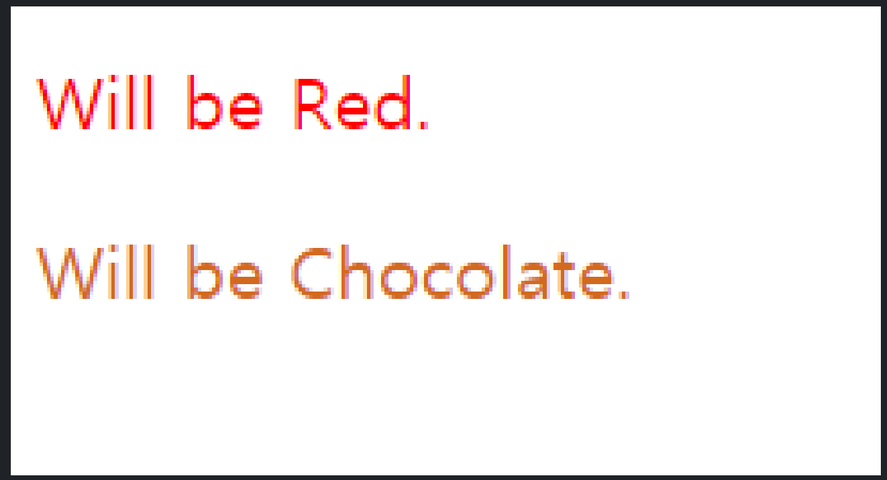
!important > 인라인 스타일 > 아이디 선택자 > 클래스/어트리뷰트/가상 선택자 > 태그 선택자 > 전체 선택자 > 상위 요소에 의해 상속된 속성

캐스캐이딩(Cascading) - 명시도

```
<!DOCTYPE html>
<html>
<head>
  <style>
    p         { color: red !important; }
    #thing    { color: blue; }

    div.food { color: chocolate; }
    .food     { color: green; }
    div       { color: orange; }
  </style>
</head>
<body>
  <p id="thing">Will be Red.</p>
  <div class="food">Will be Chocolate.</div>
</body>
</html>
```

캐스캐이딩(Cascading) - 명시도

```
<!DOCTYPE html>
<html>
<head>
  <style>
    p         { color: red !important; }
    #thing    { color: blue; }

    div.food { color: chocolate; }
    .food     { color: green; }
    div       { color: orange; }
  </style>
</head>
<body>
  <p id="thing">Will be Red.</p>
  <div class="food">Will be Chocolate.</div>
</body>
</html>
```



2.3 선언순서

선언된 순서에 따라 우선 순위가 적용된다.

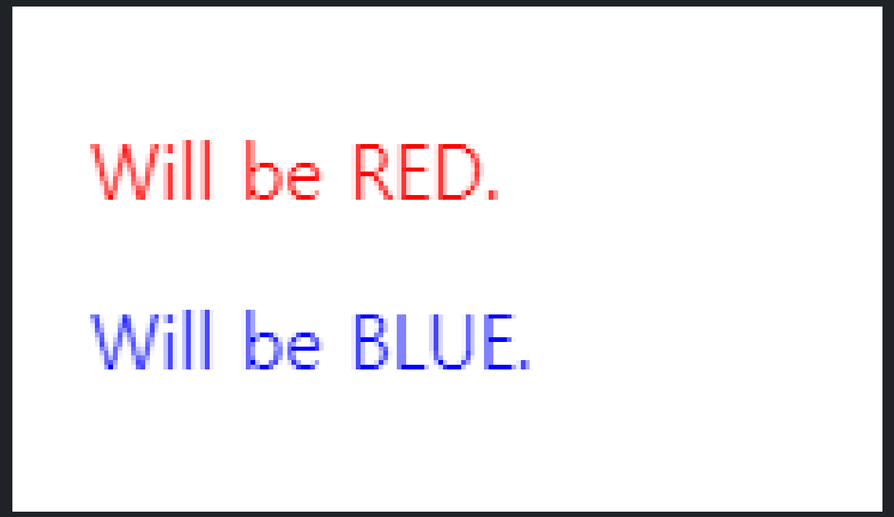
즉, 나중에 선언된 스타일이 우선 적용된다.

캐스캐이딩(Cascading) - 선언순서

```
<!DOCTYPE html>
<html>
<head>
  <style>
    p { color: blue; }
    p { color: red; }

    .red { color: red; }
    .blue { color: blue; }
  </style>
</head>
<body>
  <p>Will be RED.</p>
  <p class="blue red">Will be BLUE.</p>
</body>
</html>
```

캐스캐이딩(Cascading) - 선언순서

```
<!DOCTYPE html>
<html>
<head>
  <style>
    p { color: blue; }
    p { color: red; }

    .red { color: red; }
    .blue { color: blue; }
  </style>
</head>
<body>
  <p>Will be RED.</p>
  <p class="blue red">Will be BLUE.</p>
</body>
</html>
```



02

요소의 위치 정의

- Position -

1. position 프로퍼티

position 프로퍼티는 요소의 위치를 정의한다. top, bottom, left, right 프로퍼티와 함께 사용하여 위치를 지정한다.


01
static
(기본위치)
부모 요소의 위치를 기준으로 배치된다.
02
relative
(상대위치)
static을 선언한 요소와 relative를 선언한 요소의 차이점은 좌표 프로퍼티의 동작 여부뿐이며 그외는 동일하게 동작한다.
03
absolute
(절대위치)
부모 요소 또는 가장 가까이 있는 조상 요소(static 제외)를 기준으로 좌표 프로퍼티만큼 이동한다.
04
fixed
(고정위치)
부모 요소와 관계없이 브라우저의 view port를 기준으로 좌표프로퍼티를 사용하여 위치를 이동시킨다.

1.1 static (기본위치)

- static은 position 프로퍼티의 기본값으로 position 프로퍼티를 지정하지 않았을 때와 같다.

- 기본적인 요소의 배치 순서에 따라 위에서 아래로, 왼쪽에서 오른쪽으로 순서에 따라 배치되며 부모 요소 내에 자식 요소로서 존재할 때는 부모 요소의 위치를 기준으로 배치된다.

- 기본적으로 이 값을 지정할 일은 없지만 이미 설정된 position을 무력화하기 위해 사용될 수 있다.

- 좌표 프로퍼티(top, bottom, left, right)를 같이 사용할 수 없으며 사용할 경우에는 무시된다.

position 프로퍼티 - static

```html
<!DOCTYPE html>
<html>
<head>
  <link href="style.css"/>
</head>
<body>
  <div class="parent">
    <div class="static-box">static box</div>
  </div>
</body>
</html>
```

```css
body {
  margin: 0;
}
.parent {
  width: 150px;
  height: 150px;
  background: #bcbcbc;
  border: 1px solid #bcbcbc;
}
.static-box {
  position: static;
  background: #2E303D;
  color: #e55c3c;
  font-weight: bold;
  text-align: center;
  line-height: 150px;
}
```

position 프로퍼티 - static

```
<!DOCTYPE html>
<html>
<head>
  <link href="styl
</head>
<body>
  <div class="pare
    <div class="st
  </div>
</body>
</html>
```

static box

```
body {
  margin: 0;
```

```
bc;
```

1.2 relative (상대위치)

기본 위치(static으로 지정되었을 때의 위치)를 기준으로 좌표 프로퍼티(top, bottom, left, right)를 사용하여 위치를 이동시킨다.

static을 선언한 요소와 relative를 선언한 요소의 차이점은 좌표 프로퍼티의 동작 여부 뿐이며 그외는 동일하게 동작한다.

position 프로퍼티 - relative

```html
<!DOCTYPE html>
<html>
<body>
  <div class="parent">
    <div class="relative-box">relative box</div>
  </div>
</body>
</html>
```

```css
body {
  margin: 0;
}
.parent {
  width: 150px;
  height: 150px;
  background: #bcbcbc;
  border: 1px solid #bcbcbc;
  margin: 50px;
}
.relative-box {
  position: relative;
  top: 50px;
  left: 50px;
  background: #2E3D3D;
  color: #e55c3c;
  font-weight: bold;
  text-align: center;
  line-height: 150px;
}
```

position 프로퍼티 - relative

```html
<!DOCTYPE html>
<html>
<body>
  <div class="parent
    <div class="rela
  </div>
</body>
</html>
```

```css
body {
  margin: 0;
}

bc;

}
```

1.3 absolute (절대위치)

부모 요소 또는 가장 가까이 있는 조상 요소(static 제외)를 기준으로 좌표 프로퍼티(top, bottom, left, right)만큼 이동한다. 즉, relative, absolute, fixed 프로퍼티가 선언되어 있는 부모 또는 조상 요소를 기준으로 위치가 결정된다.

만일 부모 또는 조상 요소가 static인 경우, document body를 기준으로 하여 좌표 프로퍼티대로 위치하게 된다.

따라서 부모 요소를 배치의 기준으로 삼기 위해서는 부모 요소에 relative를 정의하여야 한다.

이때 다른 요소가 먼저 위치를 점유하고 있어도 뒤로 밀리지 않고 덮어쓰게 된다. (이런 특성을 부유 또는 부유 객체라 한다)

absolute 선언 시, block 레벨 요소의 width는 inline 요소와 같이 content에 맞게 변화되므로 적절한 width를 지정하여야 한다.

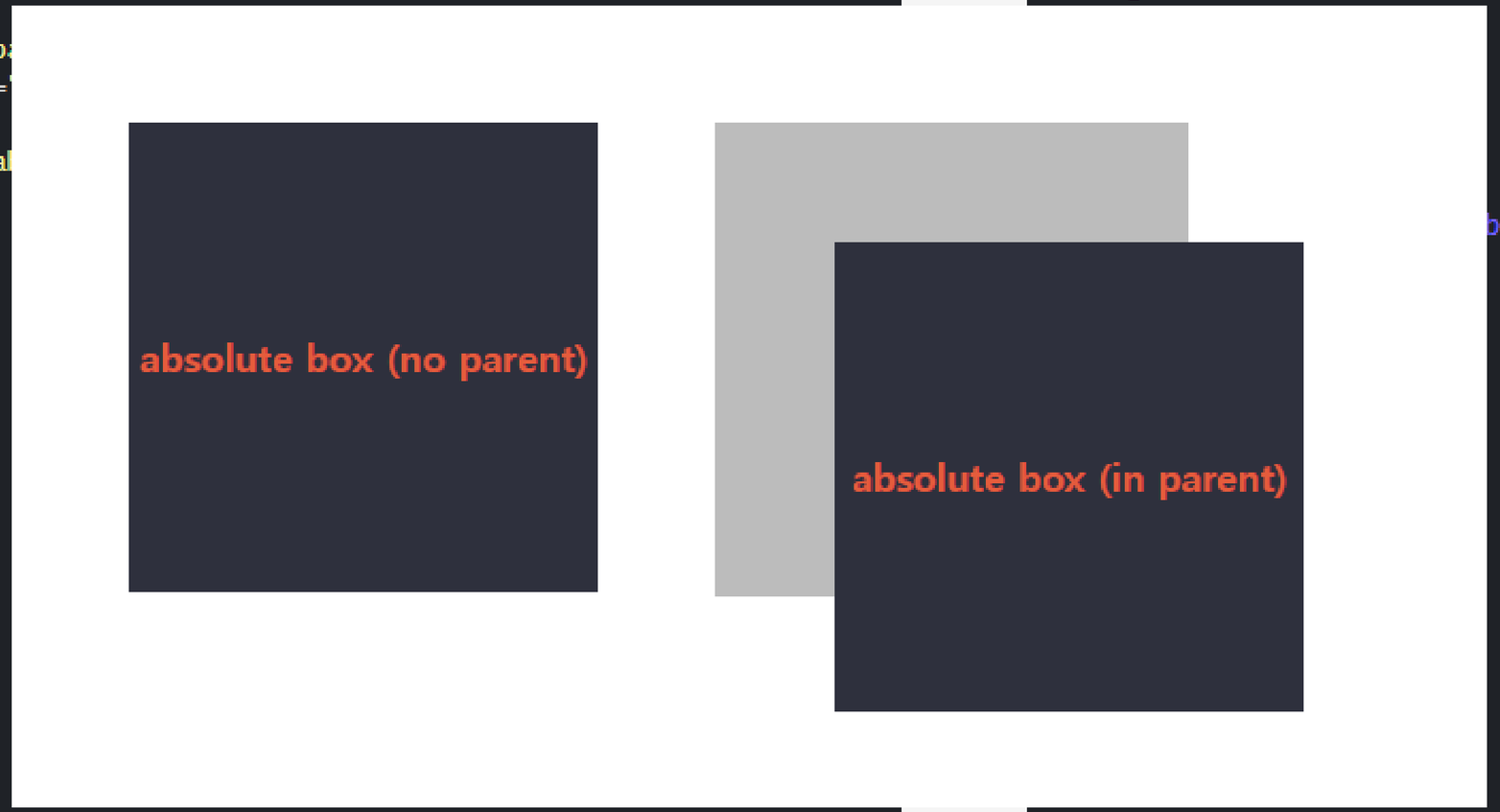
position 프로퍼티 - absolute

```html
<!DOCTYPE html>
<html>
<body>
  <div class="parent">
    <div class="absolute-box">absolute box (in parent)</div>
  </div>
  <div class="absolute-box">absolute box (no parent)</div></body>
</html>
```

```css
body {
  margin: 0;
}
.parent {
  width: 200px;
  height: 200px;
  background: #bcbcbc;
  border: 1px solid #bcbcbc;
  margin: 50px 0 0 300px;
  position: relative;
}
.absolute-box {
  position: absolute;
  height: 200px;
  width: 200px;
  top: 50px;
  left: 50px;
  color: #e55c3c;
  font-weight: bold;
  text-align: center;
  background: #2E303D;
  line-height: 200px;
}
```

position 프로퍼티 - absolute

```
<!DOCTYPE html>
<html>
<body>
  <div class="p
    <div class="
  </div>
  <div class="a
</html>
```




```
body {
  margin: 0;
}
```

```
bc;
```

1.4 fixed (고정위치)

부모 요소와 관계없이 브라우저의 viewport를 기준으로 좌표프로퍼티(top, bottom, left, right)을 사용하여 위치를 이동시킨다.

스크롤이 되더라도 화면에서 사라지지 않고 항상 같은 곳에 위치한다.

fixed 프로퍼티 선언 시, block 요소의 width는 inline 요소와 같이 content에 맞게 변화되므로 적절한 width를 지정하여야 한다.

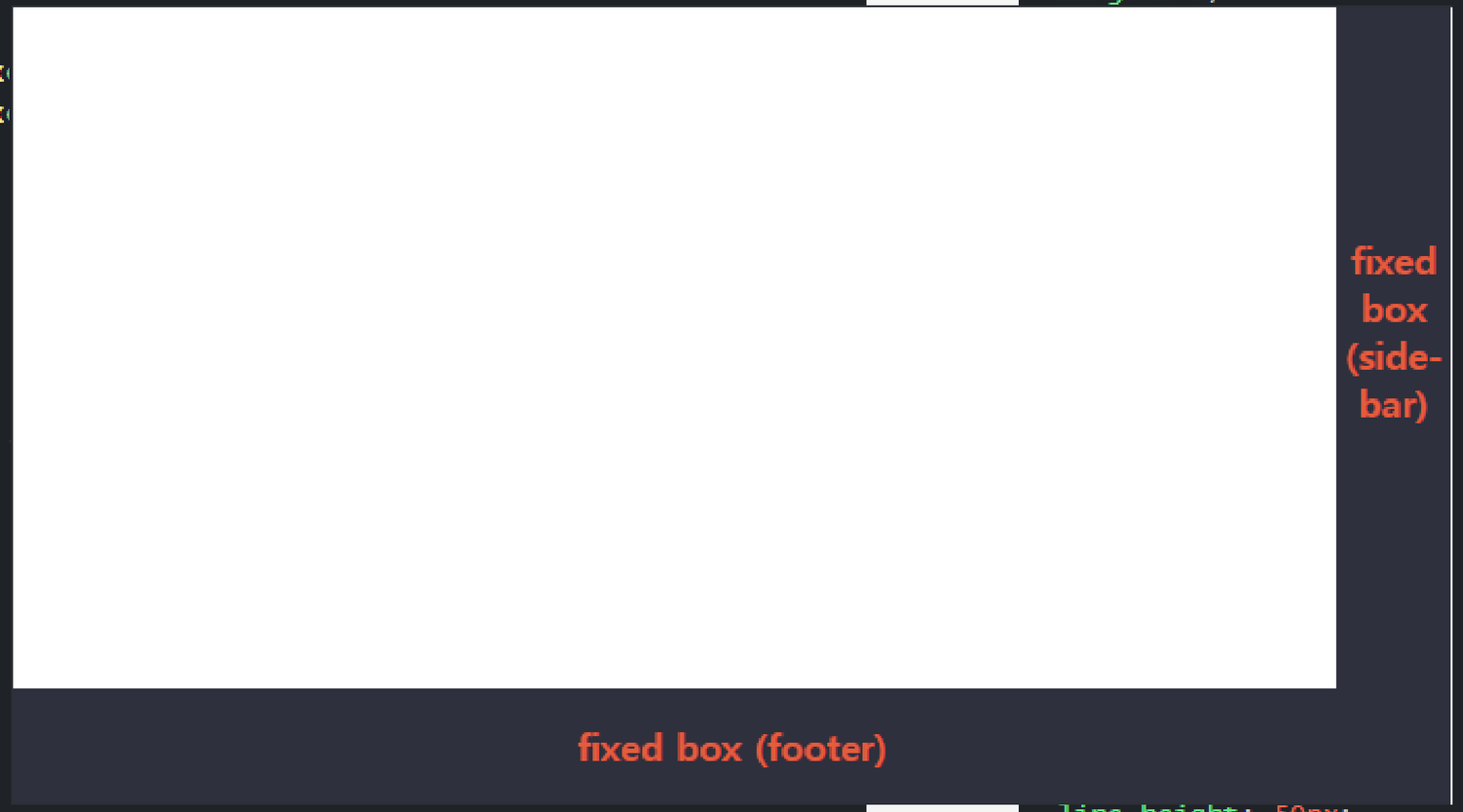
position 프로퍼티 - fixed

```html
<!DOCTYPE html>
<html>
<body>
  <div class="fixed-box sidebar">fixed box (side-bar)</div>
  <div class="fixed-box footer">fixed box (footer)</div>
</body>
</html>
```

```css
body {
  margin: 0;
}
.fixed-box {
  position: fixed;
  color: #e55c3c;
  font-weight: bold;
  text-align: center;
  background: #2E303D;
}
.sidebar {
  width: 50px;
  height: 100%;
  top: 0;
  right: 0;
  padding-top: 100px;
}
.footer {
  width: 200px;
  width: 100%;
  height: 50px;
  bottom: 0;
  left: 0;
  line-height: 50px;
}
```

position 프로퍼티 - fixed

```html
<!DOCTYPE html>
<html>
<body>
  <div class="fix
  <div class="fix
</body>
</html>
```




```css
body {
  margin: 0;
  line-height: 50px;
}
```

2. z-index 프로퍼티

z-index 프로퍼티에 큰 숫자값을 지정할수록 화면 전면에 출력된다.

positon 프로퍼티가 static 이외인 요소에만 적용된다.

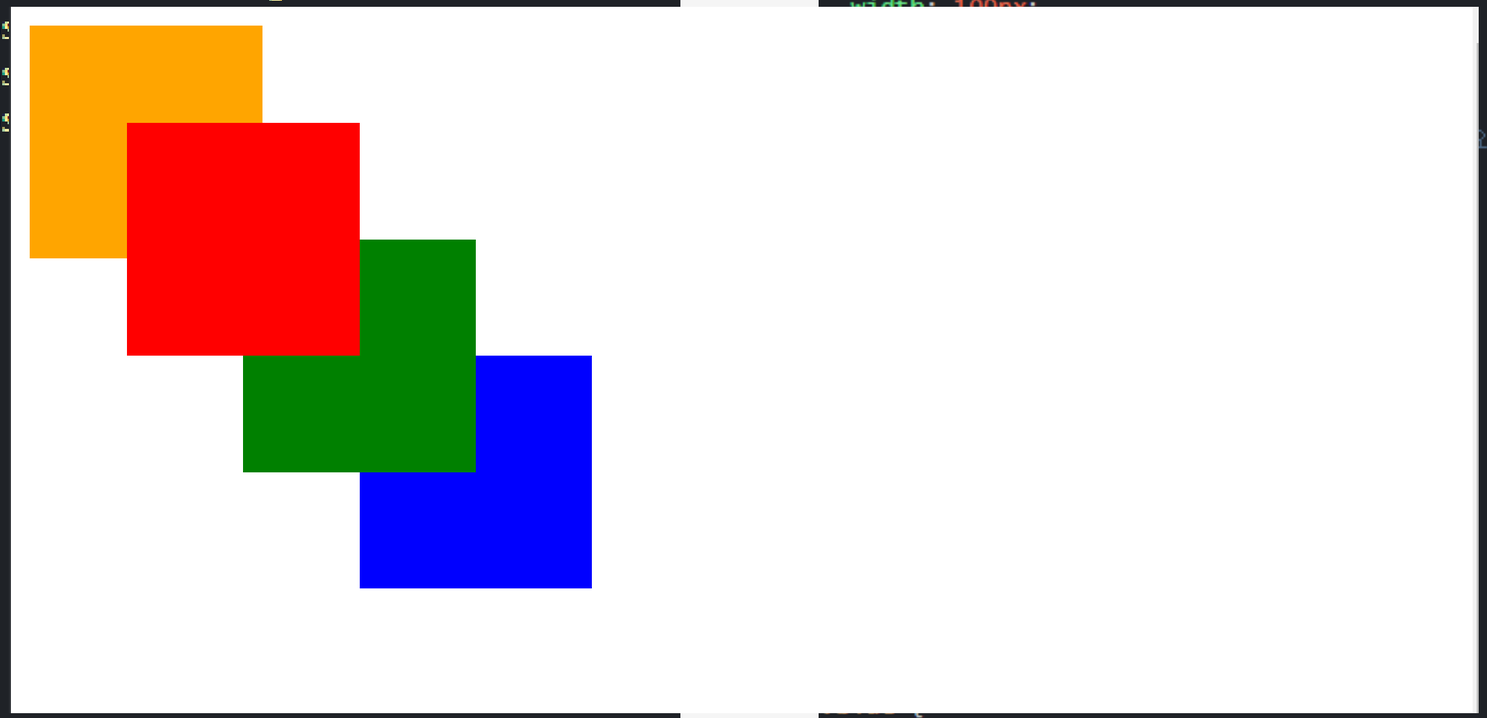
z-index 프로퍼티

```html
<!DOCTYPE html>
<html>
<body>
  <div class="normal-box orange"></div>
  <div class="absolute-box red"></div>
  <div class="absolute-box green"></div>
  <div class="absolute-box blue"></div>
</body>
</html>
```

```css
.normal-box {
  width: 100px;
  height: 100px;
}
.absolute-box {
  width: 100px;
  height: 100px;
  position: absolute;
}
/* z-index는 positon 프로퍼티가 static 이외인 요소에만 적용된다. */
.orange {
  background-color: orange;
  z-index: 1000;
}
.red {
  background-color: red;
  left: 50px;
  top: 50px;
  z-index: 100;
}
.green {
  background-color: green;
  left: 100px;
  top: 100px;
  z-index: 10;
}
.blue {
  background-color: blue;
  left: 150px;
  top: 150px;
  z-index: 1;
}
```

z-index 프로퍼티

```html
<!DOCTYPE html>
<html>
<body>
  <div class="normal-box orange"></div>
  <div class="abs
  <div class="abs
  <div class="abs
</body>
</html>
```



```css
.normal-box {
  width: 100px;
  height: 100px;
}
.absolute-box {
  width: 100px;
```

```
요소에만 적용된다. */
```

```css
  background-color: blue;
  left: 150px;
  top: 150px;
  z-index: 1;
}
```

3. overflow 프로퍼티

overflow 프로퍼티는 자식 요소가 부모 요소의 영역를 벗어났을 때 처리 방법을 정의한다.

프로퍼티값	Description
visible	영역을 벗어난 부분을 표시한다. (기본값)
hidden	영역을 벗어난 부분을 잘라내어 보이지 않게 한다.
scroll	영역을 벗어난 부분이 없어도 스크롤 표시한다.(현재 대부분 브라우저는 auto과 동일하게 작동한다)
auto	영역을 벗어난 부분이 있을때만 스크롤 표시한다.

overflow 프로퍼티

```html
<!DOCTYPE html>
<html>
<body>
  <h1>overflow</h1>
  <div class="visible"><h3>visible</h3>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.</div>
  <div class="hidden"><h3>hidden</h3>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.</div>
  <div class="scroll"><h3>scroll</h3>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.</div>
  <div class="auto"><h3>auto</h3>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.</div>
</body>
</html>
```

```css
div {
  width: 150px;
  height: 150px;
  padding: 10px;
  margin: 30px;
  font-size: 1.2em;
  border-radius: 6px;
  border-color: gray;
  border-style: dotted;
  float: left;
}
.visible { overflow: visible; }
.hidden  { overflow: hidden; }
.scroll  { overflow: scroll; }
.auto    { overflow: auto; }
```

overflow 프로퍼티

```
<!DOCTYPE html>
<html>
<body>
  <h1>overfl
  <div class
sit amet, co
empor incidi
  <div class
it amet, con
mpor incidid
  <div class
it amet, con
mpor incidid
  <div class
met, consect
incididunt u
</body>
</html>
```

```
div {
  width: 150px;
```

overflow

visible

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

hidden

Lorem ipsum dolor sit amet, consectetur

scroll

Lorem ipsum dolor sit amet, consectetur

auto

Lorem ipsum dolor sit amet, consectetur

03

수평/수직 중앙 정렬

- Horizontal & Vertical Centering -

1. 수평 정렬(Horizontal Align)

1.1 inline/inline-block 요소

정렬 대상 요소(텍스트 또는 링크 등의 inline 레벨 요소 또는 inline-block 레벨 요소)의 부모 요소에 text-align: center;를 지정한다.

수평 정렬(Horizontal Align)

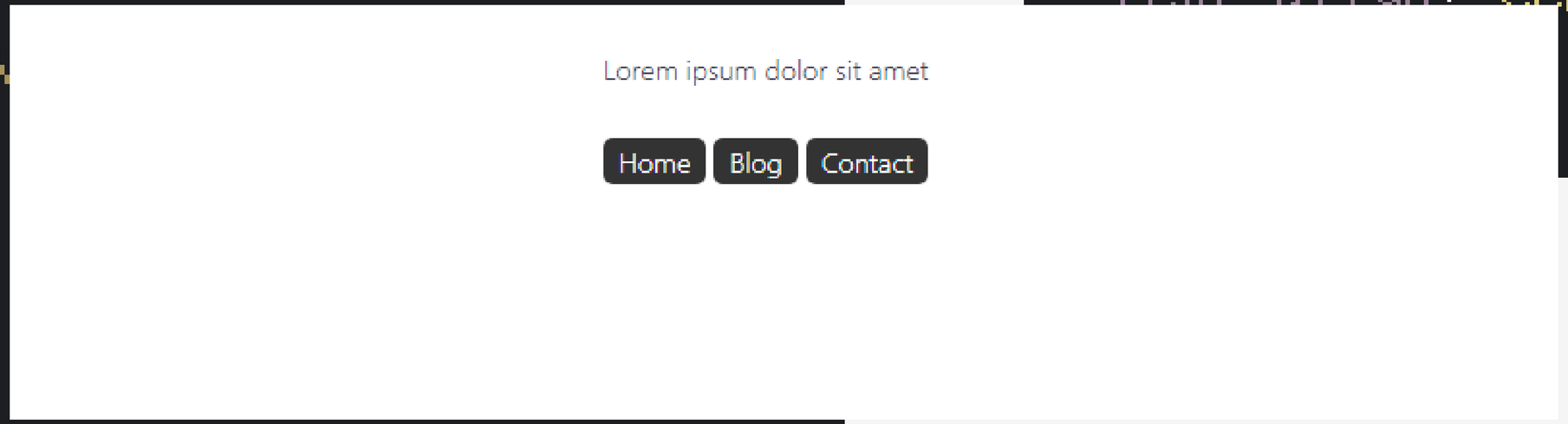
```
<div class="center inline">
  Lorem ipsum dolor sit amet
</div>
<div class="center inline">
  <a href="#">Home</a>
  <a href="#">Blog</a>
  <a href="#">Contact</a>
</div>
```

```
.center.inline {
  text-align: center;
}
```

수평 정렬(Horizontal Align)

```
<div class="center inline">
  Lorem ipsum dolor sit amet
</div>
<div
  <a
  <a
  <a href="#">Contact</a>
</div>
```

```
.center.inline {
  text-align: center;
```



1.2 block 요소

정렬 대상 요소에 너비를 명시적으로 지정하고 margin-right와 margin-left 프로퍼티에 auto를 지정한다.

정렬 대상 요소에 너비를 명시적으로 지정하지 않으면 너비는 full width가 되므로 중앙 정렬이 필요없다.

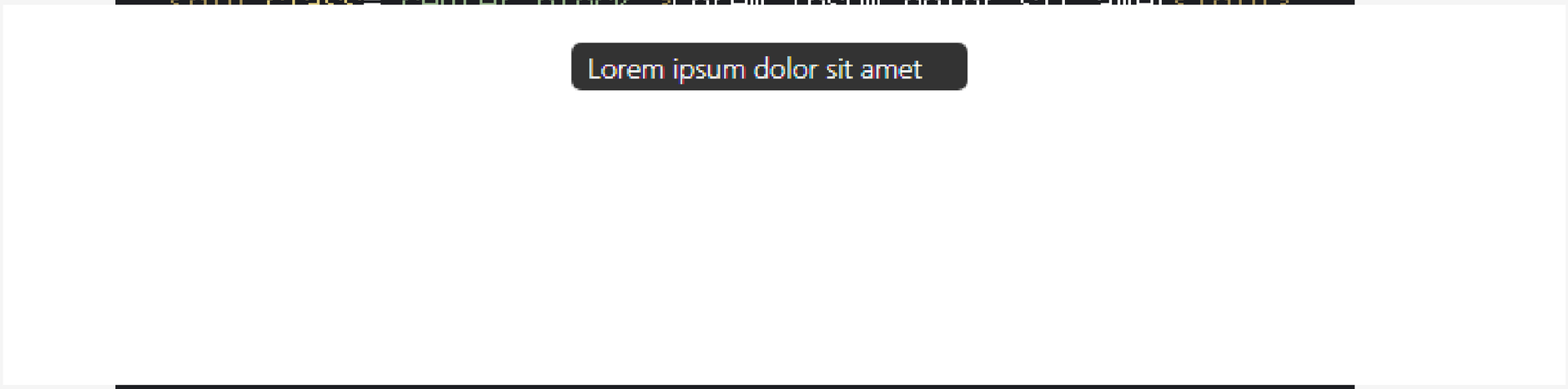
수평 정렬(Horizontal Align)

```
<div class="wrapper">
  <div class="center block">Lorem ipsum dolor sit amet</div>
</div>
```

```
.center.inline {
  text-align: center;
}
```

수평 정렬(Horizontal Align)

```
<div class="wrapper">
  <div class="center block">Lorem ipsum dolor sit amet</div>
```




```
  text-align: center;
}
```

1.3 복수의 block 요소

복수의 block 요소는 기본적으로 수직 정렬된다. 이것을 수평정렬하기 위해서는 정렬 대상 block 요소를 inline-block 요소로 변경한 후 부모 요소에 text-align: center;를 지정한다.

정렬 대상 요소에 width를 지정하지 않으면 콘텐츠에 너비에 맞추어 너비가 결정되므로 명시적으로 너비를 지정한다.

1.3 복수의 block 요소

복수의 block 요소는 기본적으로 수직 정렬된다. 이것을 수평정렬하기 위해서는 정렬 대상 block 요소를 inline-block 요소로 변경한 후 부모 요소에 text-align: center;를 지정한다.

정렬 대상 요소에 width를 지정하지 않으면 콘텐츠에 너비에 맞추어 너비가 결정되므로 명시적으로 너비를 지정한다.

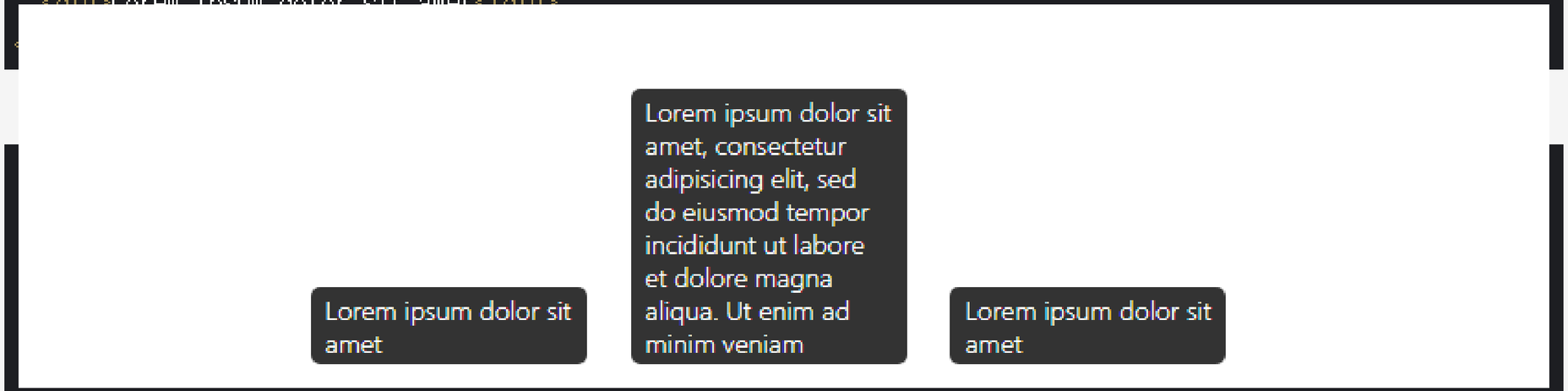
수평 정렬(Horizontal Align)

```
<div class="center inline">
  <div>Lorem ipsum dolor sit amet</div>
    <div>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam</div>
  <div>Lorem ipsum dolor sit amet</div>
</div>
```

```
.center.block {
  width: 200px;
  margin: 20px auto;
  background: #333;
  border-radius: 5px;
  color: white;
  padding: 3px 8px;
}
```

수평 정렬(Horizontal Align)

```
<div class="center inline">
  <div>Lorem ipsum dolor sit amet</div>
    <div>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam</div>
  <div>Lorem ipsum dolor sit amet</div>
</div>
```




```
  border-radius: 5px;
  color: white;
  padding: 3px 8px;
}
```

1.4 Flexbox

flexbox를 사용할 수도 있다. 정렬 대상의 부모 요소에 아래의 룰셋을 선언한다.

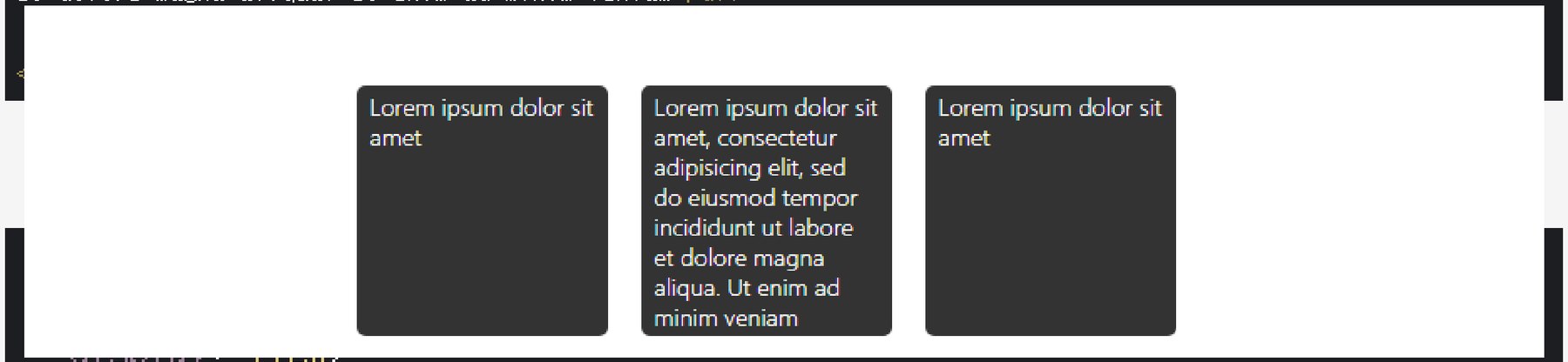
수평 정렬(Horizontal Align)

```
<div class="center flex-center">
  <div>Lorem ipsum dolor sit amet</div>
    <div>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam</div>
  <div>Lorem ipsum dolor sit amet</div>
</div>
```

```
.center.flex-center {
  display: flex;
  justify-content: center;
}
```

수평 정렬(Horizontal Align)

```
<div class="center flex-center">
  <div>Lorem ipsum dolor sit amet</div>
    <div>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam</div>
```




```
  justify-content: center;
}
```

2. 수직 정렬(Vertical Align)

2.1 inline/inline-block 요소

2.1.1 Single line
- 정렬 대상의 부모 요소에 padding-top과 padding-bottom 프로퍼티값을 동일하게 적용한다.
- padding을 사용할 수 없는 경우, 요소의 height와 line-height 프로퍼티값을 동일하게 적용한다.
단 이 방법은 행간이 지나치게 넓어지거나 Click Dead Zone 이슈가 발생하는 등 여러 줄의 텍스트에는 사용할 수 없다.

2.1.2 Multiple lines
- 여러 줄의 텍스트의 경우, padding-top과 padding-bottom 프로퍼티값을 동일하게 적용하는 방법도 가능하다.
- 또 다른 방법으로 vertical-align 프로퍼티를 사용한 방법도 가능하다. 이 방법은 table 속성을 사용하여야 한다.

2.1.3 Flexbox
- vertical-align 프로퍼티를 사용하는 방법은 table 프로퍼티를 사용하여야 하므로 번거로울 수 있다. 좀 더 간단한 방법은 flexbox를 사용하는 것이다.

수직 정렬(Vertical Align)

```
  <div class="wrapper">
    <p>padding-top과 padding-bottom 속성값을 동일하게 적용</p>
    <div class="center inline">
      Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
    </div>
    <p>height와 line-height 속성값을 동일하게 적용</p>
    <div class="center line-height">
      <a href="#">Home</a>
      <a href="#">Blog</a>
      <a href="#">Contact</a>
    </div>
    <p>vertical-align 속성과 table을 적용</p>
    <div class="center vertical-align">
      <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.</p>
    </div>
    <p>Flexbox</p>
    <div class="center flex">
      <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.</p>
    </div>
  </div>
  </div>
```

```
.center.inline {
  padding: 30px;
}

.center.line-height {
  height: 100px;
  line-height: 100px;
}

.center.vertical-align {
  display: table;
  height: 100px;
}

.center.vertical-align p {
  display: table-cell;
  vertical-align: middle;
}

.center.flex {
  display: flex;
  /*위에서 아래로 수직 배치*/
  flex-direction: column;
  /*중앙정렬*/
  justify-content: center;
  height: 100px;
}
```

수직 정렬(Vertical Align)

padding-top과 padding-bottom 속성값을 동일하게 적용

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

height와 line-height 속성값을 동일하게 적용

Home Blog Contact

vertical-align 속성과 table을 적용

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Flexbox

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

2.2 block 요소

2.2.1 요소의 높이가 고정되어 있는 경우
- 부모 요소를 기준으로 절대 위치를 지정한다.

2.2.2 요소의 높이가 불확정 상태의 경우
- 부모 요소를 기준으로 절대 위치를 지정한다.

2.2.3 Flexbox
- 부모 요소에 Flexbox layout을 지정한다.

수직 정렬(Vertical Align)

```
  <div class="wrapper-block">
    <p>요소의 높이가 고정되어 있는 경우</p>
    <div class="center block">
      Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
    </div>
  </div>

  <div class="wrapper-block">
    <p>요소의 높이가 불확정 상태의 경우</p>
    <div class="center block-unknown-height">
      Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
    </div>
  </div>

  <div class="wrapper-block-flex">
    <p>Flexbox</p>
    <div class="center flex">
      Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
    </div>
  </div>
```

```
.center.block {
  /*요소의 높이가 고정되어 있는 경우*/
  height: 100px;
  padding: 30px;
  position: absolute;
  top: 50%;
  /*요소의 높이(100px)의 반 만큼 위로 이동*/
  margin-top: -50px;
}

.center.block-unknown-height {
  padding: 30px;
  position: absolute;
  top: 50%;
  /*요소의 높이의 반 만큼 위로 이동*/
  transform: translateY(-50%);
}

.wrapper-block-flex {
  max-width: 700px;
  height: 200px;
  margin: 10px auto;
  padding: 10px;
  background-color: #fff;
  display: flex;
  flex-direction: column;
  justify-content: center;
}

.wrapper-block-flex .center.flex {
  padding: 30px;
}
```

수직 정렬(Vertical Align)

요소의 높이가 고정되어 있는 경우

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

요소의 높이가 불확정 상태의 경우

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Flexbox

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

3. 수평/수직 정렬(Horizontal & Vertical Align)

요소의 너비와 높이가 고정되어 있는 경우, 요소의 너비와 높이가 불확정 상태의 경우 모두 사용 가능한 방법이다

Flexbox를 사용한 방법도 있다.

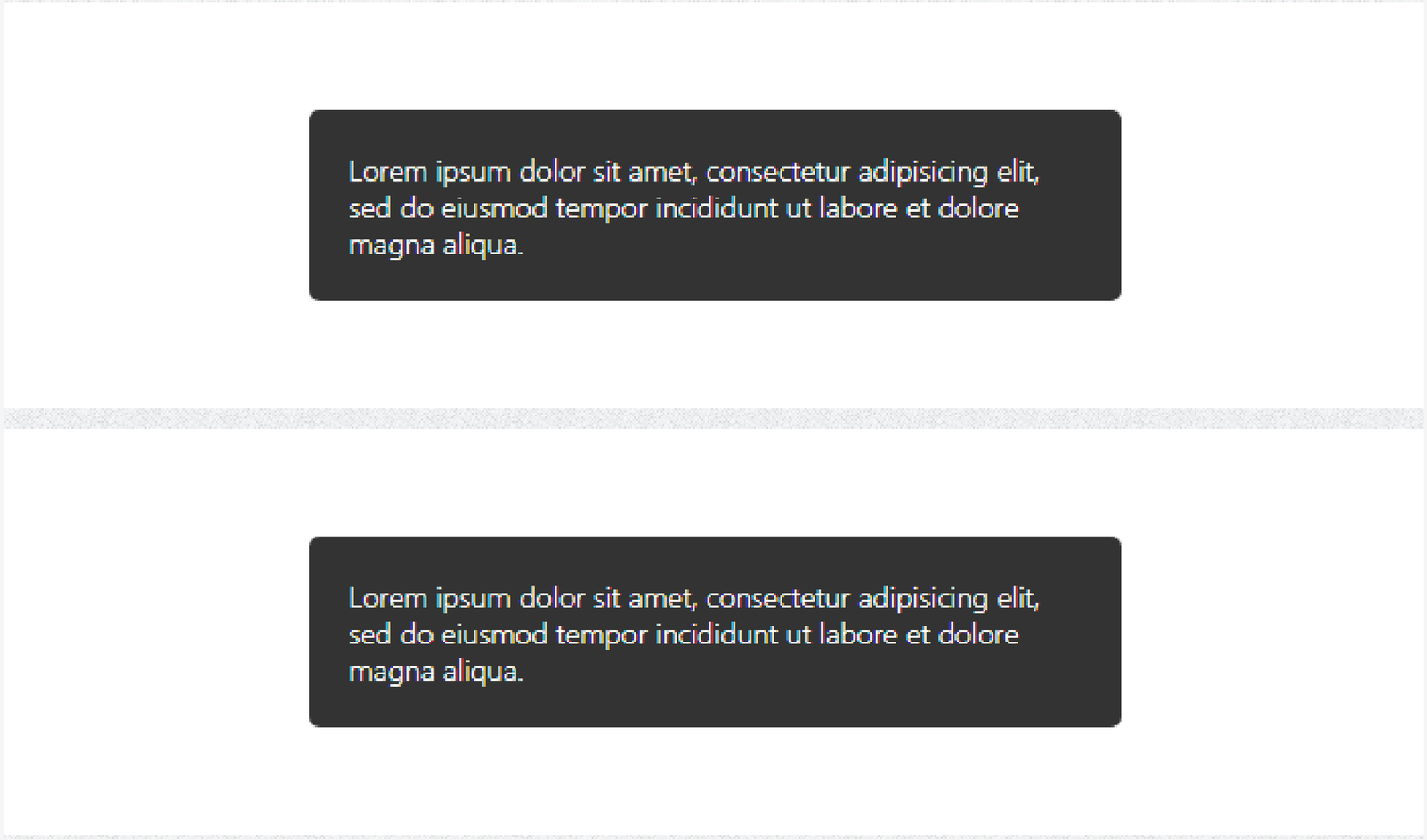
수평/수직 정렬(Horizontal & Vertical Align)

```html
<div class="wrapper">
  <div class="center">
    Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
  </div>
</div>

<div class="wrapper flex">
  <p>Flexbox</p>
  <div class="center">Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.</div>
</div>
```

```css
.wrapper {
  max-width: 700px;
  height: 200px;
  margin: 10px auto;
  padding: 10px;
  background-color: #fff;
  position: relative;
}

.center {
  width: 400px;
  background: #333;
  border-radius: 5px;
  color: white;
  padding: 20px;
}
```

```css
.wrapper.flex {
  display: flex;
  justify-content: center;
  align-items: center;
}

.wrapper .center {
  position: absolute;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
}

.wrapper.flex .center {
  background: #333;
  border-radius: 5px;
  color: white;
  padding: 20px;
}
```

수평/수직 정렬(Horizontal & Vertical Align)



<참조>

- 스타일의 상속과 적용 우선 순위
  https://poiemaweb.com/css3-inheritance-cascading
- 요소의 위치 정의
  https://poiemaweb.com/css3-position
- 수평/수직 중앙 정렬
  https://poiemaweb.com/css3-centering
- Full property table
  https://www.w3.org/TR/CSS22/propidx.html

See you Again!

다음 이 시간에 만나요!