

# CSS 레이아웃 position



# 목차

- position 속성
- static
- absolute
- relative
- fixed
- z-index

# position

요소의 배치 기준을 지정하는 속성

```
div {  
    position: absolute;  
}
```

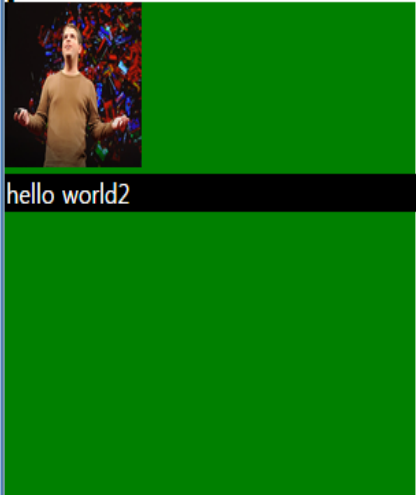
static (기본값)	기본 배치방향
absolute	기준점: static이 아닌 상위 요소
relative	기준점: 원래 배치 위치
fixed	기준점: 브라우저(viewport)

# position

```
<body>|
  <h1>position 이해하기</h1>
  <div>
    
    <p>hello world2</p>
  </div>
</body>
```

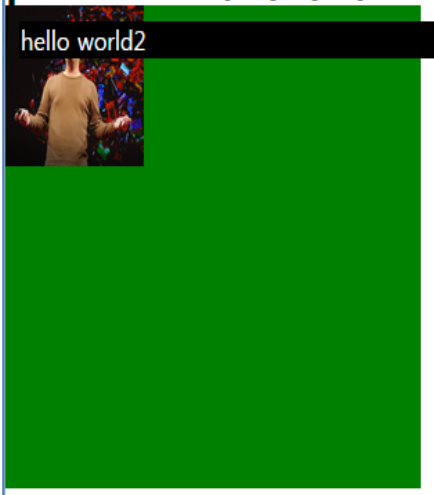
p요소의 position 속성을 변경했을 때 모습

position 이해하기



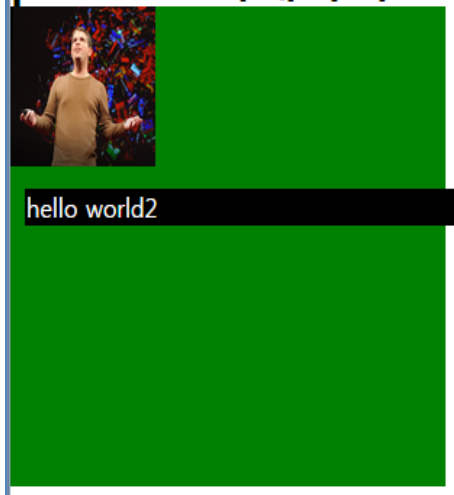
static

position 이해하기

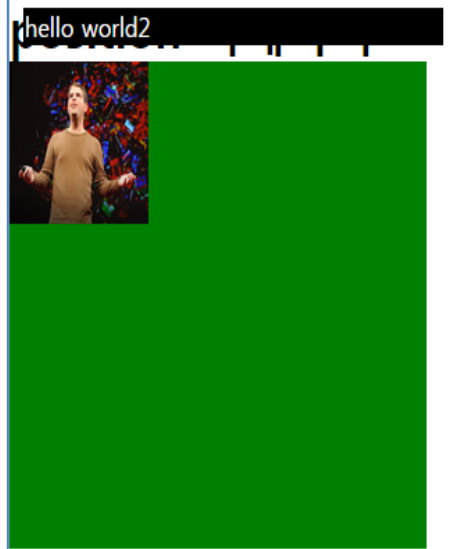


absolute

position 이해하기



relative



fixed

# 요소의 위치 설정

- 요소가 기준에서 **얼만큼 떨어질지** 지정하는 속성, 위치좌표

```
div {  
  top: 10px;  
  left: 10px;  
  right: 10px;  
  bottom: 10px;  
}
```

속성	의미
top: 10px	위쪽 기준점에서 10px만큼 떨어진 위치
left: 10px	왼쪽 기준점에서 10px만큼 떨어진 위치
right: 10px	오른쪽 기준점에서 10px만큼 떨어진 위치
bottom: 10px	아래쪽 기준점에서 10px만큼 떨어진 위치

# 요소의 위치 설정

```
<body>
  <div>
    <h1>top, left, right, bottom 이해하기</h1>
    <div class="container">
      <p>안녕하세요</p>
    </div>
  </div>
</body>
```

top, left, right, bottom 이해하기

안녕하세요

top, left, right, bottom 이해하기

안녕하세요

top:100px

top, left, right, bottom 이해하기

안녕하세요

bottom:100px

top, left, right, bottom 이해하기

안녕하세요

left:100px

top, left, right, bottom 이해하기

안녕하세요

right:100px

# static position

```
div {  
    position: static;  
}
```

- 요소의 위치가 기본 배치흐름을 따르도록 설정하는 것
- 위치좌표 속성이 적용 안됨
- 거의 사용할 일이 없음

# absolute position

상위요소를 기준으로 잡고 위치좌표 속성 값에 따라 배치한다

- 기준이 되는 상위요소: position 속성이 **static**이 아닌 가장 근접한 상위요소

```
div.first {
  position: relative;
  background-color: green;
  width: 500px;
  height: 500px;
}
div.second {
  background-color: red;
  width: 500px;
  height: 200px;
}
div.third {
  background-color: blue;
  width: 500px;
  height: 250px;
}
</style>
</head>
<body>
  <h1>absoulte position 이해하기</h1>
  <div class="first">
    <div class="second"></div>
    <div class="third">
      
      <p>hello world2</p>
    </div>
  </div>
</body>
```





# absolute position

```
p {  
  width: 100px;  
  background-color: black;  
  color: white;  
  
  position: absolute;  
  top: 100px;  
}
```



```
div.third {  
  background-color: blue;  
  position: relative;  
  width: 500px;  
  height: 250px;  
}  
p {  
  width: 100px;  
  background-color: black;  
  color: white;  
  
  position: absolute;  
  top: 150px;  
}  
</style>  
</head>  
<body>  
  <h1>absoulte position 이해하기</h1>  
  <div class="first">  
    <div class="second"></div>  
    <div class="third">  
      <img src= "../30days.jpg" width="100px" height="100px">  
      <p>hello world2</p>  
    </div>  
  </div>  
</body>
```



# absolute position 주의사항

기준이 될 상위요소에

position: relative를 적용한 후

위치좌표 속성들을 사용할 것

# relative position

요소의 기본 위치를 기준으로 잡고 위치좌표를 설정해서 배치하는 기능

```
<body>
  <h1>position 이해하기</h1>
  <div>
    
    <p>hello world2</p>
  </div>
</body>
```



```
<style type="text/css">
  *{
    padding:0; margin:0;
  }
  p {
    border:1px solid black;
    width: 300px;
    background-color: black;
    color: white;
    position: relative;
    top: 10px;
    left: 10px;
  }
  div {
    background-color: green;
    width:300px;
    height: 300px;
  }
</style>
</head>
<body>
  <h1>position 이해하기</h1>
  <div>
    
    <p>hello world2</p>
  </div>
</body>
</html>
```



# relative position 유의사항

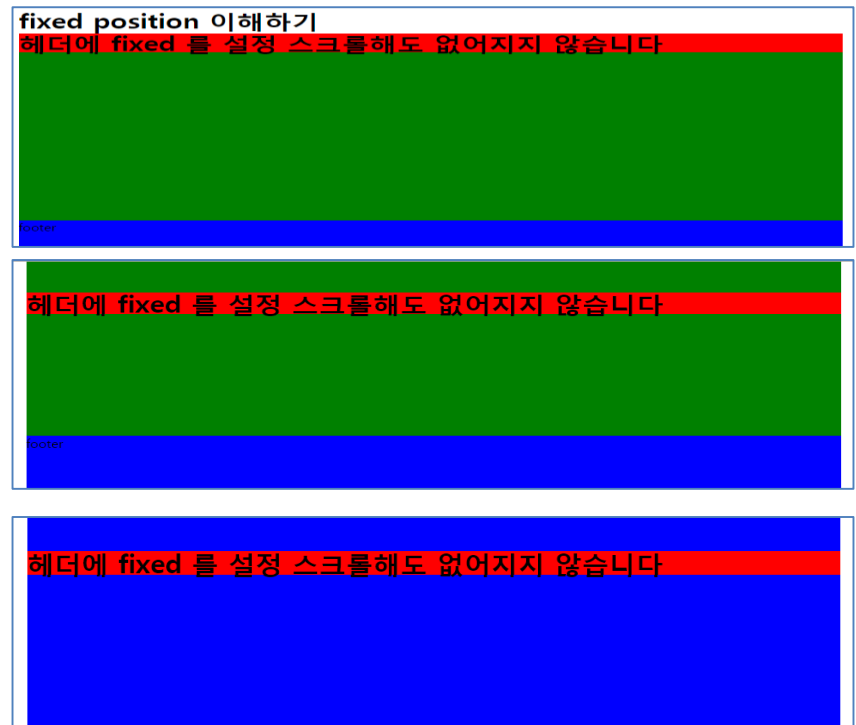
relative 속성은 요소 배치를 위해서는  
잘 사용되지 않습니다.

absolute position의 기준 요소로 설정  
하기 위해서 많이 사용됩니다.

# fixed position

브라우저의 viewport를 기준으로 위치좌표를 설정  
fixed 설정된 요소의 위치는 고정이므로 스크롤 해도 사라지지 않습니다.

```
footer {  
  background-color: blue;  
  width: 100%;  
  height: 600px;  
}  
header {  
  background-color: red;  
  width: 960px;  
  height: 30px;  
  line-height: 30px;  
  position: fixed;  
}  
</style>  
</head>  
<body>  
  <h1>fixed position 이해하기</h1>  
  <header>  
    <h1>헤더에 fixed를 설정하면 |스�크롤해도 없어지지 않습니다</h1>  
  </header>  
  <section>section  
  </section>  
  <footer>  
    footer  
  </footer>  
</body>  
</html>
```



# fixed position 유의사항

fixed가 적용된 요소 뒤에 따라오는 요소의 위치

```
header {  
  background-color: red;  
  width: 960px;  
  height: 30px;  
  line-height: 30px;  
  position: fixed;  
}  
</style>  
</head>  
<body>  
  <h1>fixed position 이해하기</h1>  
  <header>  
    <h1>헤더에 fixed를 설정하면 스크롤해도 없어지지 않습니다</h1>  
  </header>  
  <section>이 문구는 가려서 보여지지 않습니다.</section>  
  <footer>  
    footer  
  </footer>  
</body>
```

fixed position 이해하기

헤더에 fixed를 설정하면 스크롤해도 없어지지 않습니다

fixed position 이해하기

이 문구는 가려서 보여지지 않습니다.

헤더에 fixed를 설정하면 스크롤해도 없어지지 않습니다

# fixed position 유의사항

fixed가 적용된 요소 뒤에 오는 요소에는 padding 또는 margin을 설정해서 보이게 합니다.

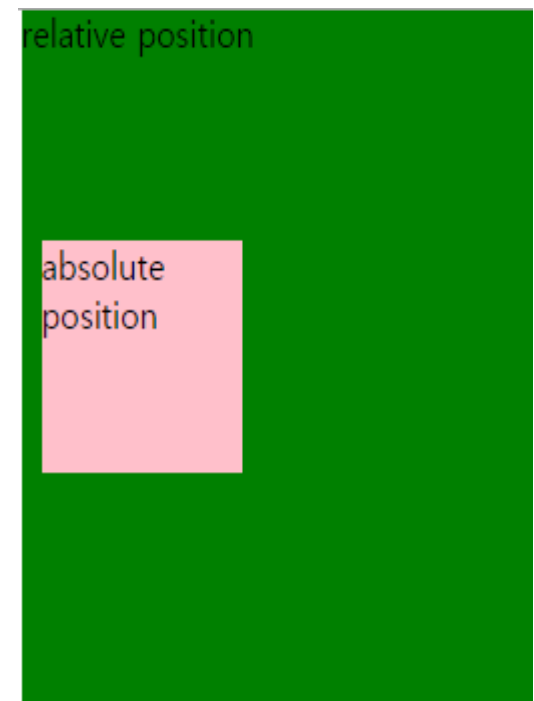
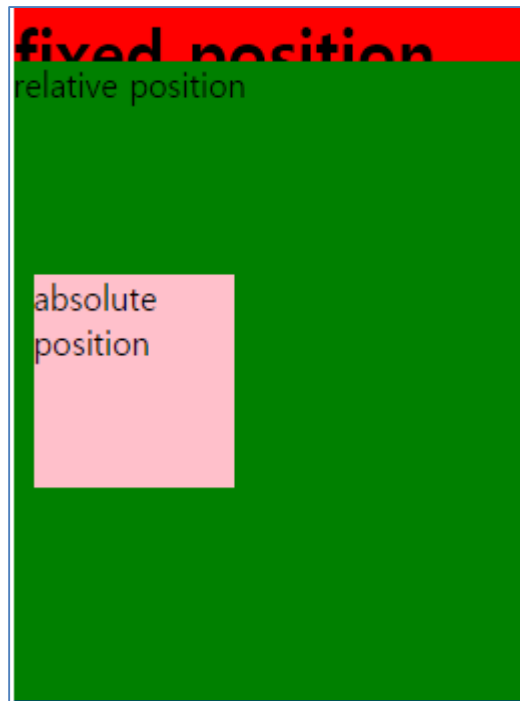
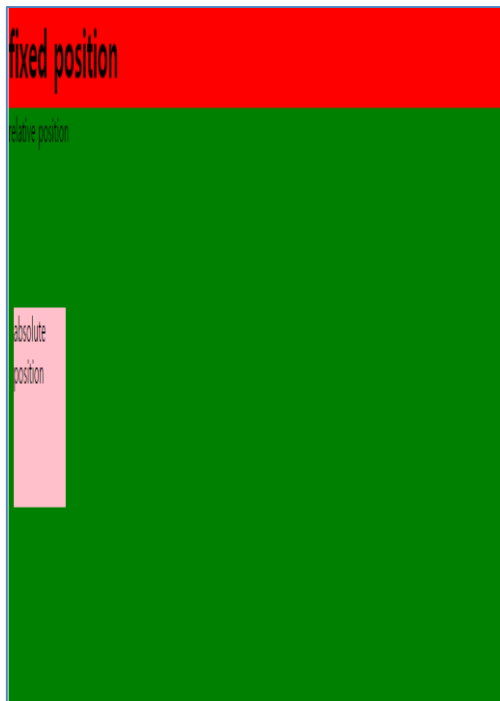
```
header {  
  background-color: red;  
  width: 960px;  
  height: 30px;  
  line-height: 30px;  
  position: fixed;  
}  
  
section {  
  padding-top: 30px;  
}  
/style>  
</head>  
<body>  
  <h1>fixed position 이해하기</h1>  
  <header>  
    <h1>헤더에 fixed를 설정하면 스크롤해도 없어지지 않습니다</h1>  
  </header>  
  <section>이 문구는 가려서 보여지지 않습니다.</section>  
  <footer>  
    footer  
  </footer>  
</body>
```

## fixed position 이해하기

헤더에 fixed를 설정하면 스크롤해도 없어지지 않습니다  
이 문구는 이제 보입니다.

# position 유의사항

- 정상에서 벗어난 것들의 충돌
  - absolute, relative, fixed
  - 세 개의 position 속성은 일반 배치흐름을 벗어나 상위 레이어에 배치되는 데 요소들의 충돌이 일어나게 되면 우선순위가 높은 것이 표현된다





# position 과 z-index

- 우선순위
  - html문서에서 가장 밑에 있는 요소가 우선순위가 높음
- 임의로 요소에 우선순위를 주는 방법
  - **z-index**: 속성의 값이 클수록 우선순위가 높음

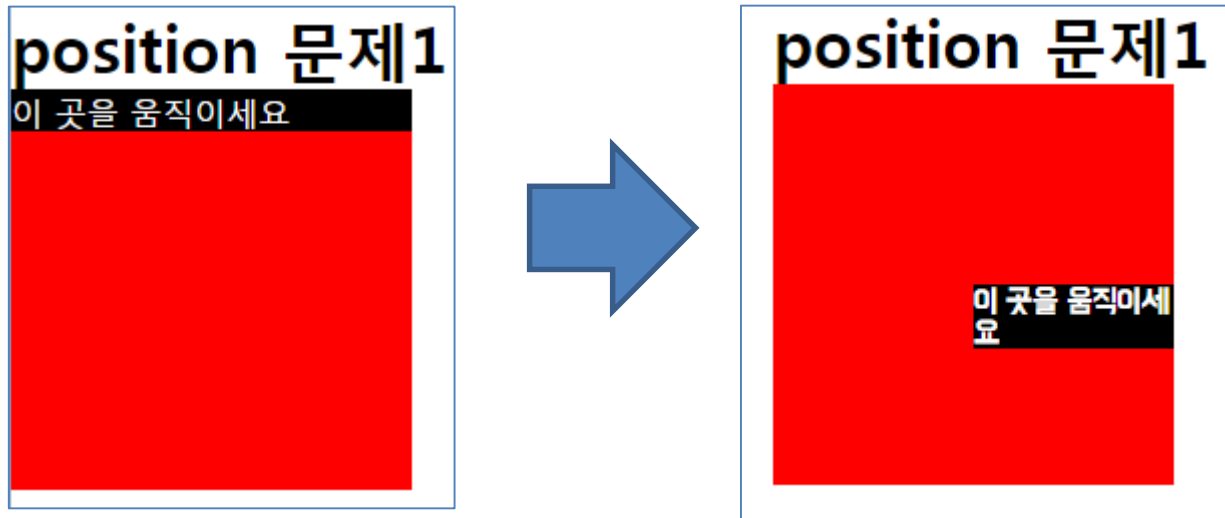
```
<body>
  <header>
    <h1>fixed position</h1>
  </header>
  <section>
    relative position
    <div class="container">
      absolute position
    </div>
  </section>
  <footer>
    footer
  </footer>
</body>
```

```
header h1 {
  position: fixed;
  background-color: red;
  width: 960px;
  height: 50px;
  z-index: 1000;
}
```



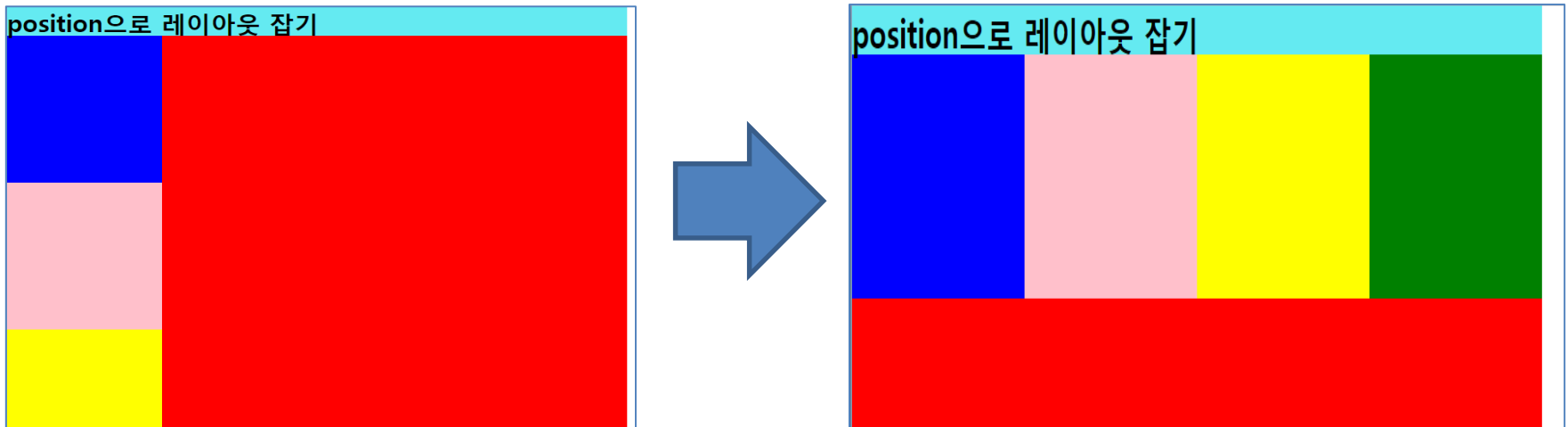
# 문제1

- css\_position\_exam1을 열어 아래 그림처럼 변경하세요
  - 빨간색 상자를 기준으로 위에서 100px 왼쪽에서 100px 이동



# 문제2

- css\_position\_exam2을 열어 아래 그림처럼 변경하세요
  - 클래스 'component' div를 일렬로 정렬하세요
  - float을 사용해도 좋고 absolute position 을 사용해도 됩니다



# 문제3

- css\_position\_exam2에서 작성한 문서를 가지고 header요소안의 h1을 fixed로 변경하세요. 스크롤 시에 position으로 레이아웃 잡기가 계속 보여야 합니다.
  - fixed position과 z-index를 사용

