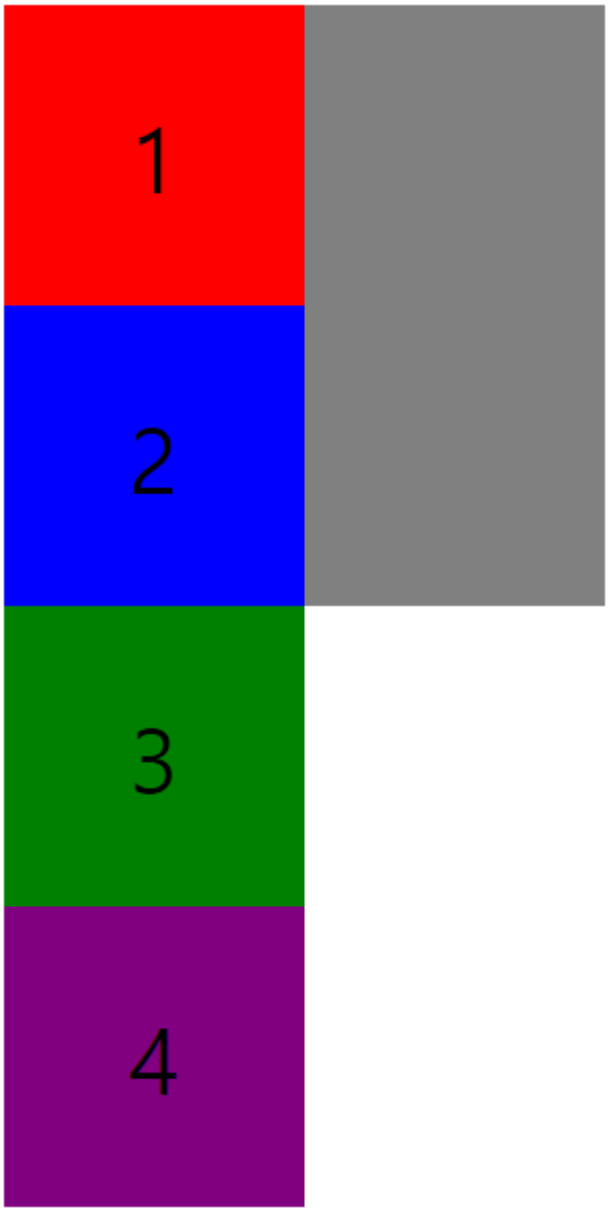
display 에는 inline, block, iline-block, content flex 등 다양한 속성들이 있다.
하지만 그중에서도 가장 활용도가 높은 display 는 flex 라고 생각한다.
레이아웃을 원하는대로 쉽고 효과적으로 배치하도록 해주는 고마운 속성이다.
일단 변화를 보기 위해 기본 코드와 결과를 확인하자.

기본 코드

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Document</title>
    <style>
        .base{
            width: 300px;
            height: 300px;
            background-color: gray;



        }
        .redBox{
            width: 150px;
            height: 150px;
            background-color: red;
            vertical-align: middle;
            line-height: 150px;
            text-align: center;
        }
        .blueBox{
            width: 150px;
            height: 150px;
            background-color: blue;
            line-height: 150px;
            text-align: center;
        }
        .greenBox{
            width: 150px;
            height: 150px;
            background-color: green;
```

```
            line-height: 150px;
            text-align: center;
        }
        .purpleBox{
            width: 150px;
            height: 150px;
            background-color: purple;
            line-height: 150px;
            text-align: center;
        }
        span{
            font-size: 45px;
            margin-bottom: auto;
            margin-top: auto;
        }
    </style>

</head>
<body>
    <div class="base">
        <div class="redBox"><span>1</span></div>
        <div class="blueBox"><span>2</span></div>
        <div class="greenBox"><span>3</span></div>
        <div class="purpleBox"><span>4</span></div>

    </div>
</body>
</html>
```

결과:



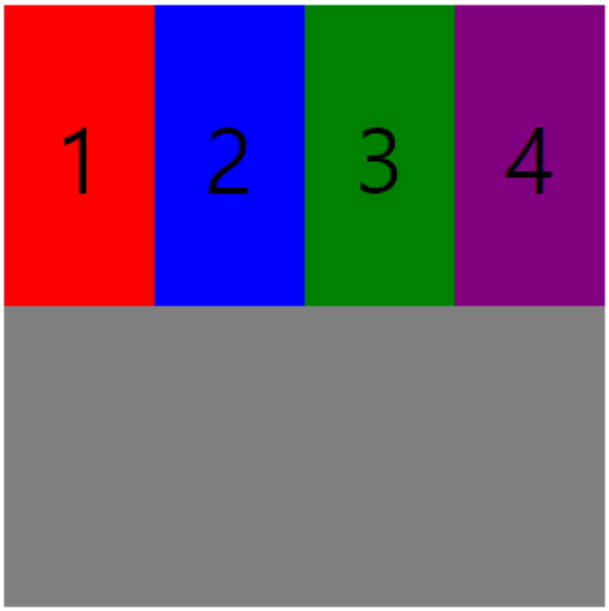
default

수평 정렬

```
.base{
            width: 300px;
            height: 300px;
            background-color: gray;
            display: flex;
        }
```

display: flex; 적용

결과:



display: flex;

무조건 1 행 수평으로 정렬한다.
만약 크기가 너무 커서 자리가 없다면 강제로 크기를 줄인다.
모든 크기를 균등하게 하는 것이 아니라 들어가는 태그들의 상호 크기에 비례해서 줄어든다.

만약 부모 div 인 회색의 크기가 자식의 div 크기보다 더 커서 크기를 딱 맞게 맞추고 싶은 경우 부모(회색) div 에 display: inline-flex 를 주면 딱 맞춰진다.
자바에서 swing 의 pack(); 과 유사하다고 보면 된다.

```
display: inline-flex;
```

display inline-block 과 다르게 일일히 크기를 맞춰줄 필요도 없고
float 보다 더 쉽게 원하는 모양을 만들 수 있다.

```
display: flex;
```

```
flex-direction: row;
```

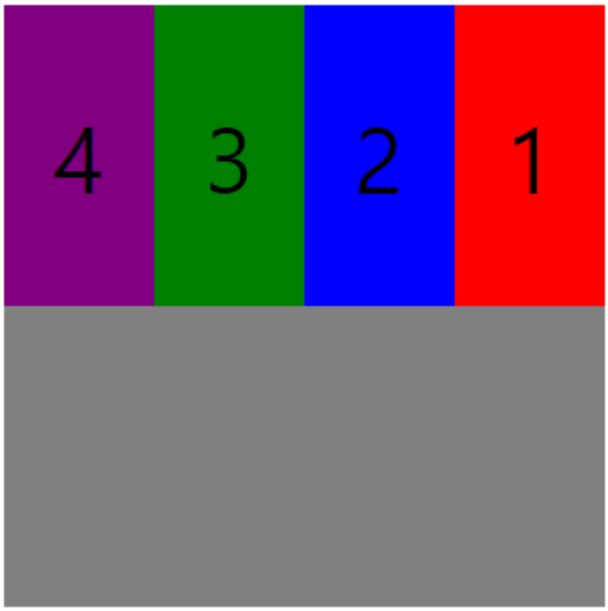
flex-direction: row; 를 추가해도 똑같은 결과지만

```
display: flex;
```

```
flex-direction: row-reverse;
```

flex-direction: row-reverse; 뒤에 리버스를 추가하면 순서가 뒤바뀐다.

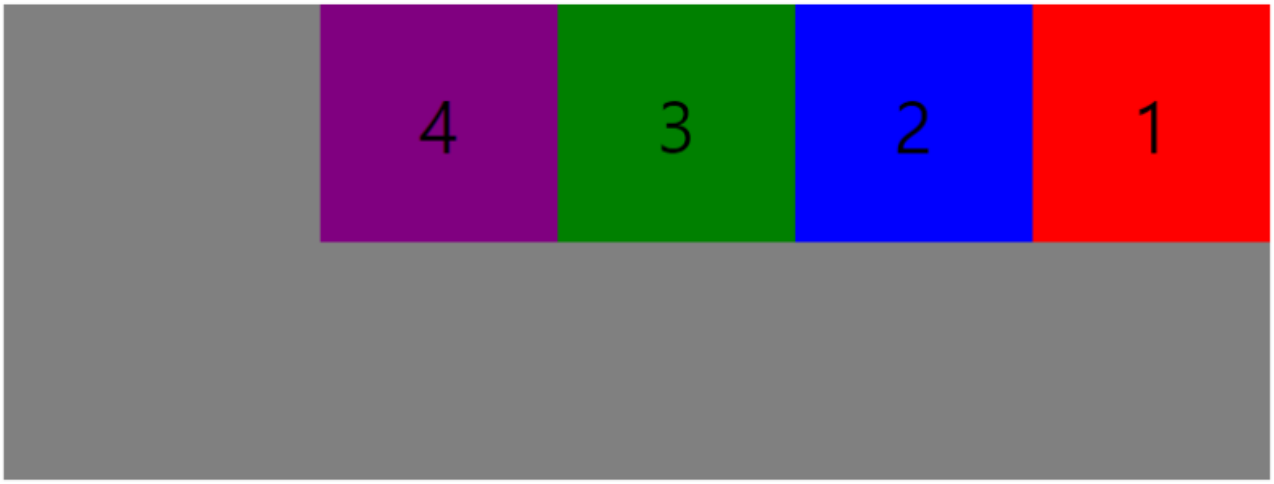
결과:



flex-direction: row-reverse;

순서가 뒤바뀌어 빨강이 아닌 보라색 부터 역순으로 진행되었다.
화면의 우측 부터 배치가 된다. 만약 회색이 크다면 우측부터 배치되므로 좌측에 빈공간이 생길 수 있다.

예시:

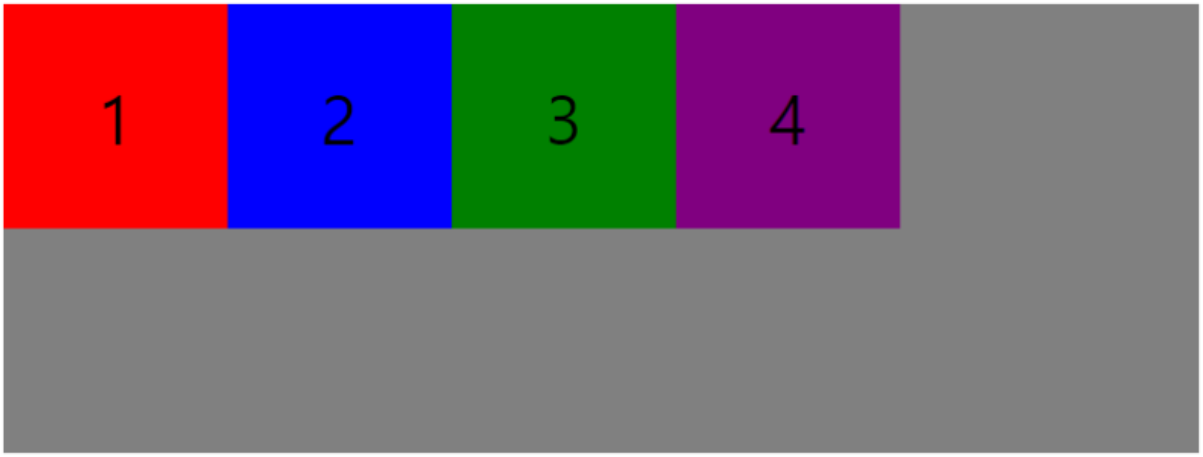


우측 부터 정렬

수평-여백

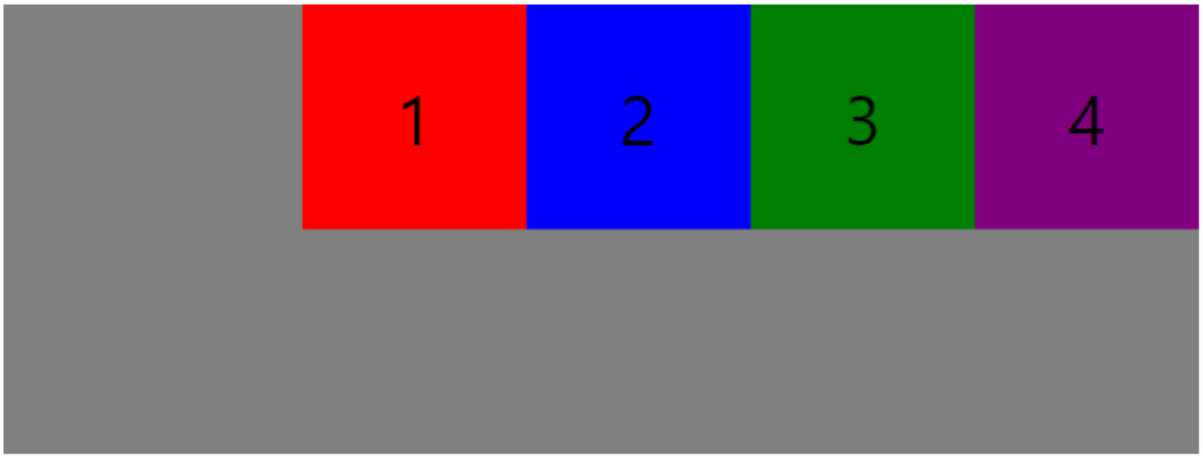
justify-content 속성을 사용하는데 특히 메뉴바에 자주 사용된다.

아래는 모두 .base 클래스(div)에 적용한 예시다.



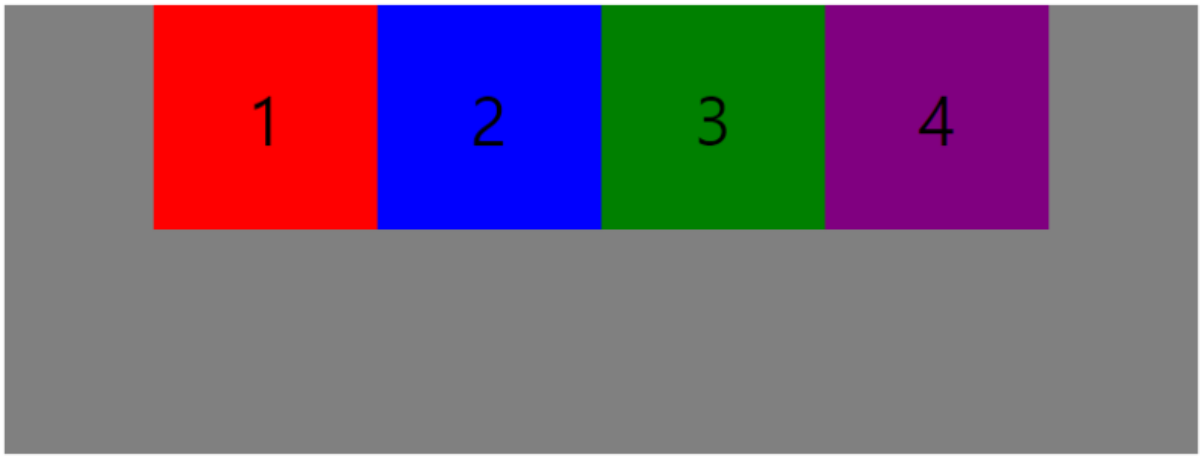
justify-content: flex-start;

기본 값과 같다.

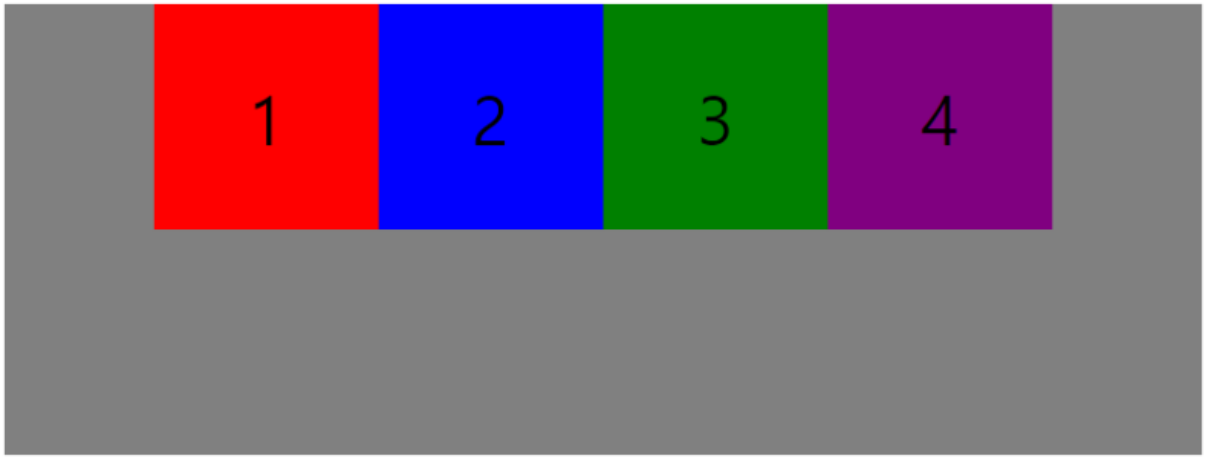


justify-content: flex-end;

우측에 붙는다.

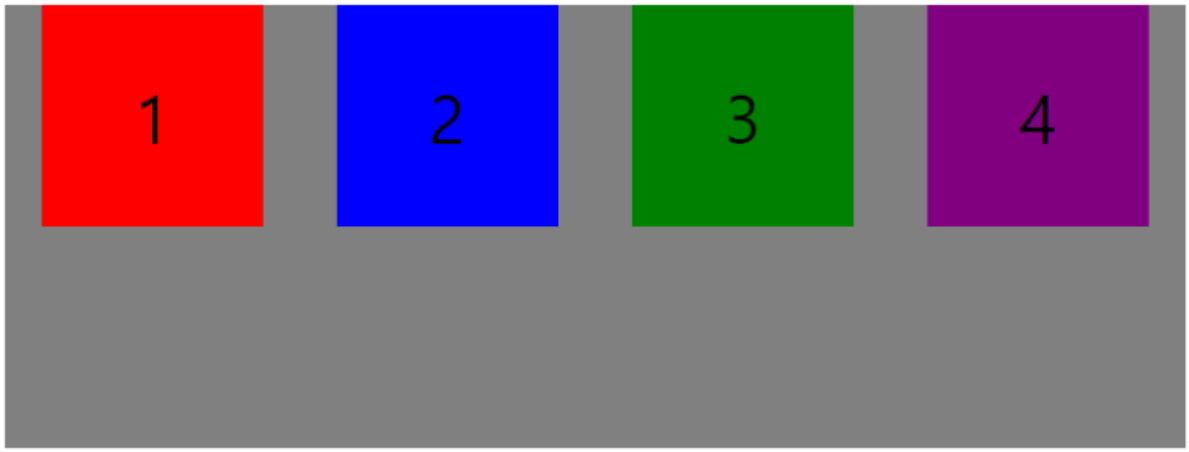


justify-content: center;



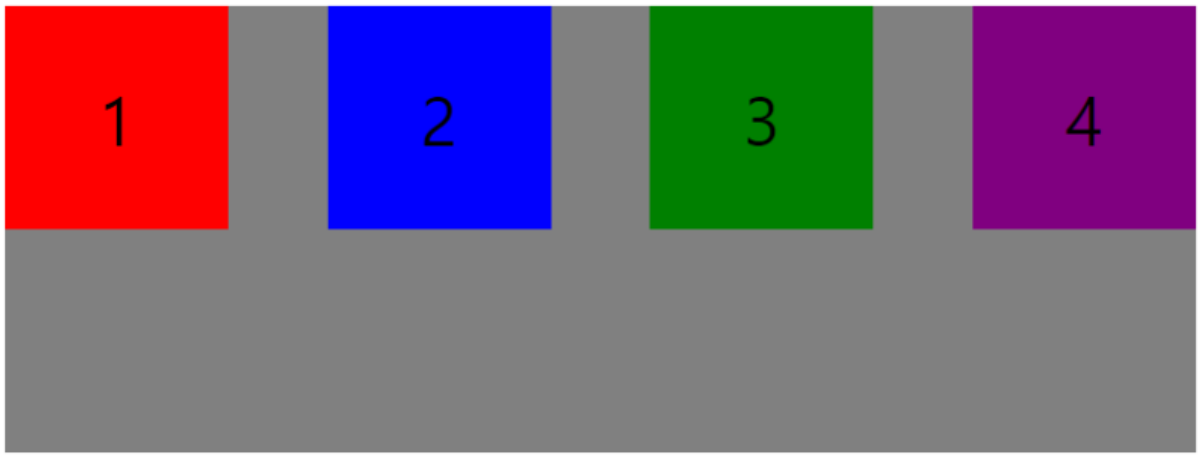
justify-content: center;

가운데에 붙는다.



justify-content: space-around;

양 끝에 약간의 간격을 두고
각 태그 간에 일정한 비율의 간격을 가진다.



justify-content: space-between;

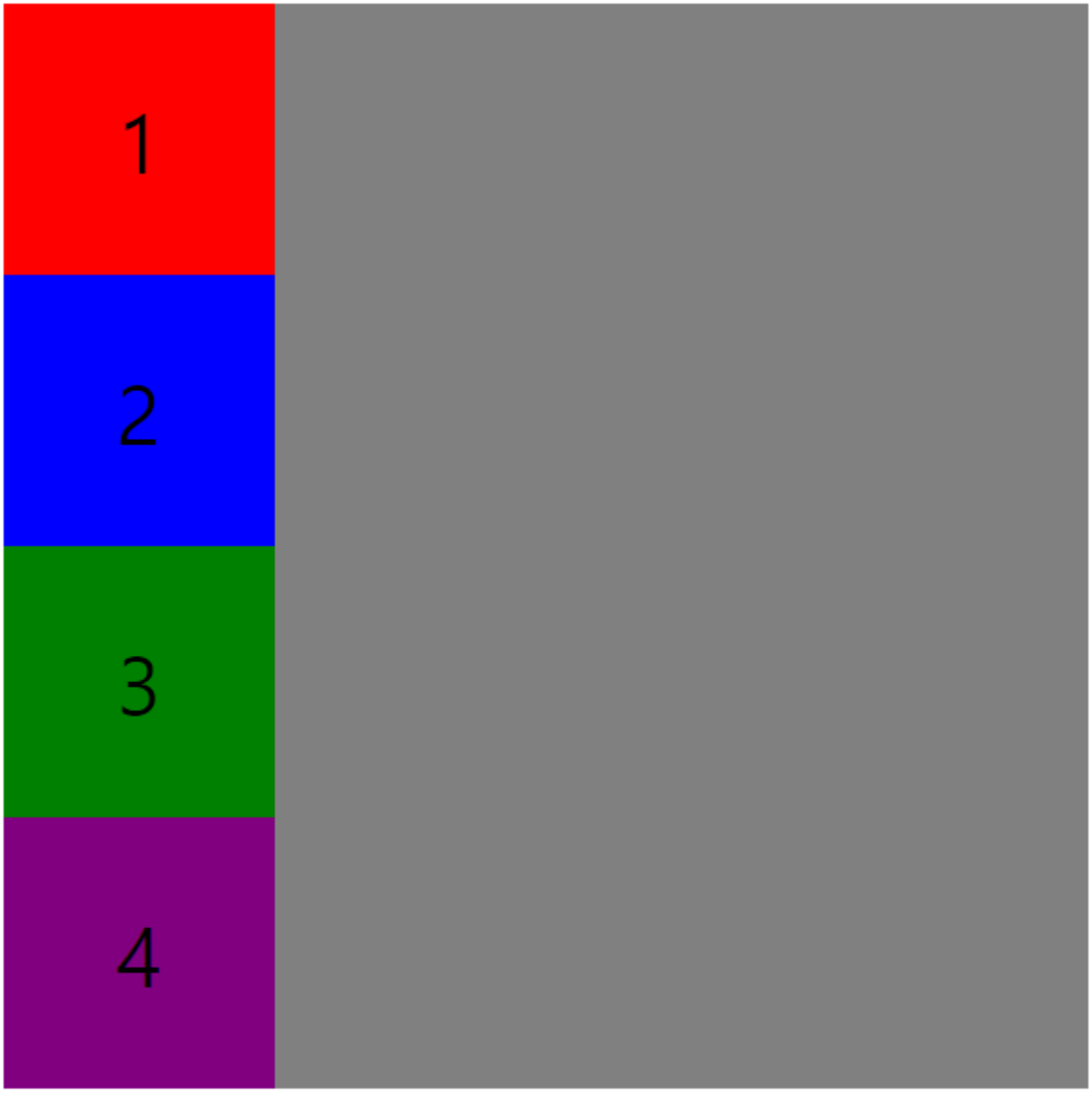
양 끝에 간격 없이
각 태그 간에 일정한 비율의 간격을 가진다.

수직 정렬

```
.base{
    width: 600px;
    height: 600px;
    background-color: gray;
    display: flex;
    flex-direction: column;
}
```

flex-direction: column; 적용

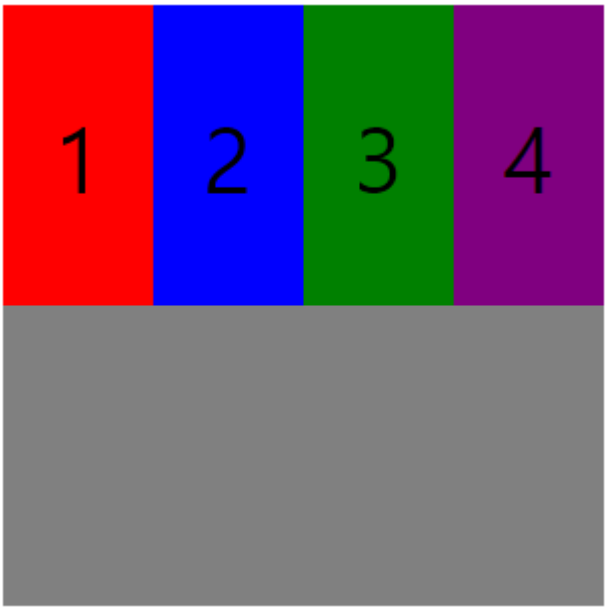
결과:



row 와 마찬가지로 뒤에 -reverse 를 붙이면 순서가 뒤바뀐다.

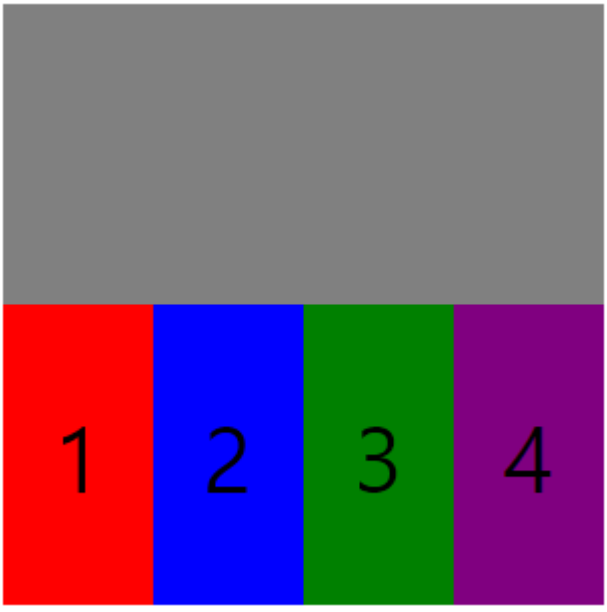
수직 여백

align-items 속성을 사용한다.



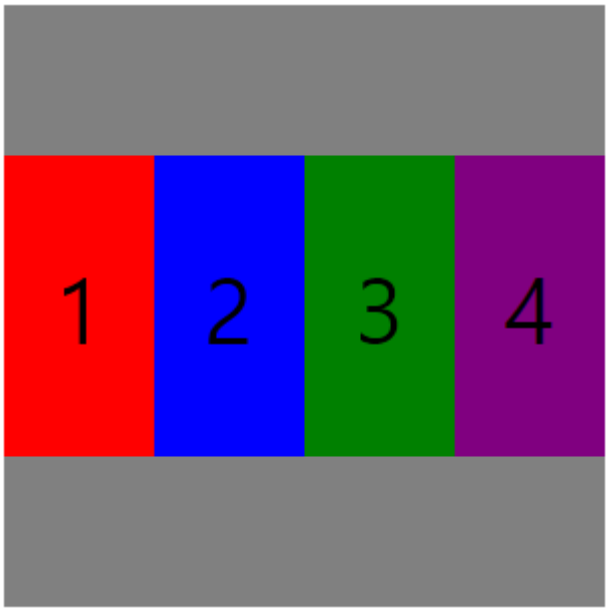
align-items: flex-start;

기본값이다.
flex-direction: column; 를 적용할 경우 수직이 아닌, 수평 정렬이 되는 현상이 일어나므로 이 예시에서는 적용하지 않았다.
만약 flex-direction: column; 를 적용하면서 수직 정렬을 하고 싶다면 div 로 내용들을 감싸고 그 div 에 flex-direction: column; 를 적용하면 된다.



align-items: flex-end;

밑에 붙는다.



align-items: center;

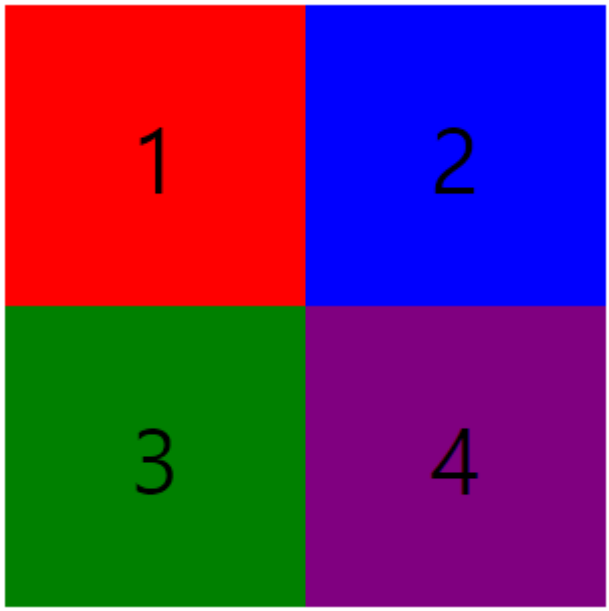
가운데 붙는다.

--

자동 줄바꿈

```
.base{
    width: 300px;
    height: 300px;
    background-color: gray;
    display: flex;
    flex-wrap: wrap;
}
```

결과:



flex-wrap: wrap;

앞에서 본 바 display: flex; 를 적용할 경우 내용의 크기에 상관 없기 무조건 1 행으로 강제 되었다.
하지만 flex-wrap: 를 적용할 경우 내용의 크기가 넘친다면 자동으로 줄바꿈을 해준다.
마찬가지로 -reverse 를 붙일 경우 순서가 반대로 적용된다.

참고로 flex-wrap 을 따로 설정하지 않았다면 기본 속성은 nowrap 이다.

앞에 설명한 flex-direction 와 flex-wrap 를 한줄로 축약할 수도 있다.
바로 flex-flow 속성이다.

```
.base{
    width: 300px;
    height: 300px;
    background-color: gray;
    display: flex;
    flex-flow: column wrap;
}
```

flex-flow: column wrap; 는
flex-direction: column 와
flex-wrap: wrap; 을 합친 속성이다.

적용 방법
flex-flow: (direction) (wrap)

이렇게 어떤 형태로든 합칠 수 있으니 direction 와 wrap 을 동시에 사용해야 하는 경우라면 flex-flow 를 사용하는 것이 편하다.

flex 의 편리한 속성 덕분에 웹, 탭, 모바일 기기의 환경에 따라 컨텐츠들의 크기가 바뀌는 반응형 웹에도 많이 쓰인다.