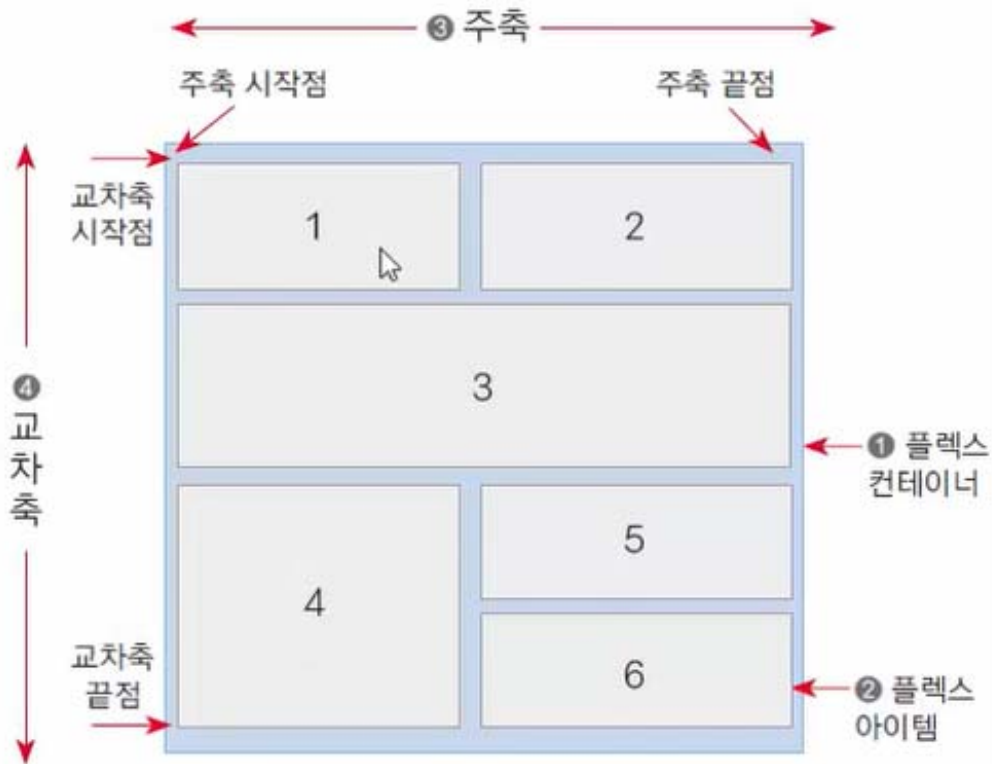


❖ 플렉스 박스 레이아웃(Flex Box Layout)

- 플렉스 박스 레이아웃이란 그리드 레이아웃을 기본으로 하여 플렉스 아이টে을 원하는 위치에 배치하는 레이아웃
- 플렉스 박스 레이아웃은 플렉스 아이টে의 크기뿐만 아니라 배치하는 방향과 순서, 정렬 방법, 간격 등을 제어할 수 있음
- 플렉스 박스 레이아웃을 사용하려면 콘텐츠를 플렉스 컨테이너로 묶고, 컨테이너 안에 플렉스 아이টে(이미지, 텍스트 등 웹 요소)을 배치
- 플렉스 박스는 주축(가로방향)을 따라 왼쪽에서 오른쪽으로 배치되고, 주축의 끝점까지 닿으면 교차축(세로방향)을 따라 아래로 이동한 후, 다시 왼쪽에서 오른쪽으로 배치되는 방식임



전체 -> 컨테이너라고 지칭

플렉스 컨테이너(부모 박스)

- 플렉스 박스 레이아웃을 적용할 대상을 묶는 요소

플렉스 아이টে(자식 박스)

- 플렉스 박스 레이아웃을 적용할 대상 요소

주축

- 플렉스 컨테이너 안에서 플렉스 아이টে을 배치하는 주 방향
- 기본으로 수평방향

교차 축

- 주축과 교차되는 방향(수직방향)

❖ 플렉스 박스 레이아웃(Flex Box Layout)

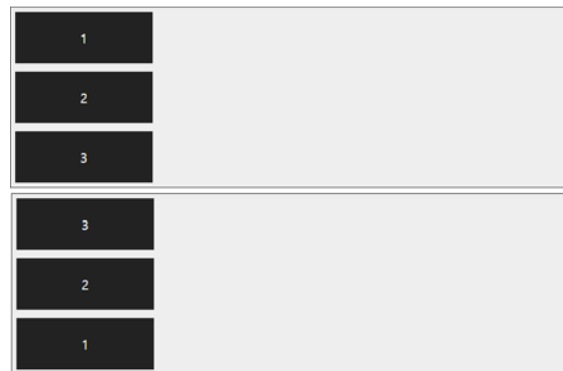
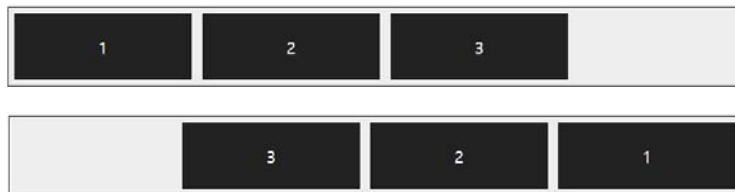
1. 플렉스 컨테이너 속성

✓ display 속성

- HTML 요소에 대한 플렉스 박스 타입을 선언(명시)
 - display: flex; 컨테이너 안의 아이টে을 블록 레벨 요소로 배치
 - display: inline-flex; 컨테이너 안의 아이টে을 인라인 레벨 요소로 배치

✓ flex-direction 속성

- 플렉스 컨테이너 안에서 플렉스 아이টে을 배치하는 방향(주축)을 지정
 - flex-direction: row; 주축을 가로로 지정(왼쪽에서 오른쪽으로 배치, 기본값)
 - flex-direction: row-reverse; 주축을 가로로 지정(오른쪽에서 왼쪽으로 배치)
 - flex-direction: column; 주축을 세로로 지정(위쪽에서 아래쪽으로 배치)
 - flex-direction: column-reverse; 주축을 세로로 지정(아래쪽에서 위쪽으로 배치)



```
<div class="container opt1">
  <div class="box"><p>1</p></div>
  <div class="box"><p>2</p></div>
  <div class="box"><p>3</p></div>
</div>
<div class="container opt2">
  <div class="box"><p>1</p></div>
  <div class="box"><p>2</p></div>
  <div class="box"><p>3</p></div>
</div>
<div class="container opt3">
  <div class="box"><p>1</p></div>
  <div class="box"><p>2</p></div>
  <div class="box"><p>3</p></div>
</div>
<div class="container opt4">
  <div class="box"><p>1</p></div>
  <div class="box"><p>2</p></div>
  <div class="box"><p>3</p></div>
</div>
```

flex-1.html

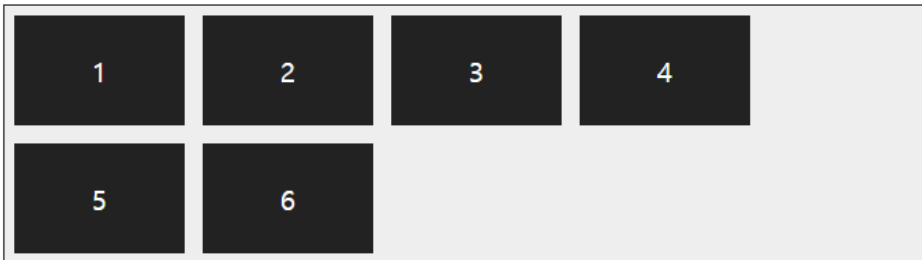
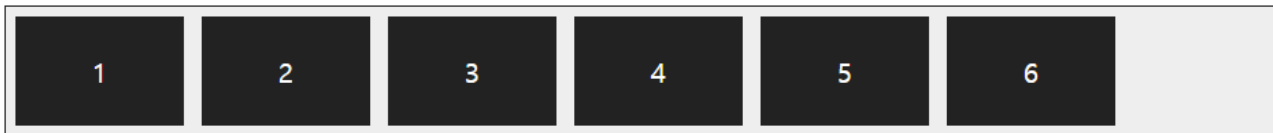
```
.container {
  display: flex;
  background-color: #eee;
  border: 1px solid #222;
  margin-bottom: 10px;
}
.opt1 {
  flex-direction: row;
}
.opt2 {
  flex-direction: row-reverse;
}
.opt3 {
  flex-direction: column;
}
.opt4 {
  flex-direction: column-reverse;
}
```

❖ 플렉스 박스 레이아웃(Flex Box Layout)

1. 플렉스 컨테이너 속성

✓ flex-wrap 속성

- 플렉스 라인에 더 이상의 여유 공간이 없을 때, 플렉스 아이템의 위치를 다음 줄로 넘길지를 설정
 - flex-wrap: nowrap; 플렉스 아이템을 한 줄에 표시(default)
 - flex-wrap: wrap; 플렉스 아이템을 여러 줄에 표시
 - flex-wrap: wrap-reverse; 플렉스 아이템을 여러 줄에 표시(아이템을 역으로 표시)



flex-2.html

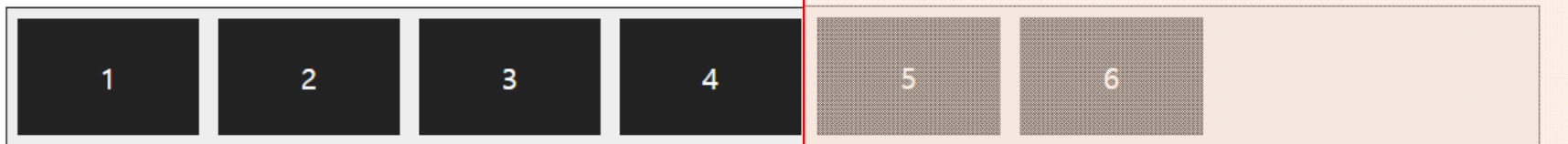
```
.container {  
  display: flex;  
  background-color: #eee;  
  border: 1px solid #222;  
  margin-bottom: 10px;  
}  
.opt1 {  
  flex-wrap: nowrap;  
}  
.opt2 {  
  flex-wrap: wrap;  
}  
.opt3 {  
  flex-wrap: wrap-reverse;  
}
```

❖ 플렉스 박스 레이아웃(Flex Box Layout)

1. 플렉스 컨테이너 속성

✓ flex-flow 속성

- flex-direction 속성과 flex-wrap 속성을 함께 지정
 - flex-flow: row wrap; 주축을 가로로 지정하고, 플렉스 아이টে을 여러 줄에 표시
 - flex-flow: row nowrap; 주축을 가로로 지정하고, 플렉스 아이টে을 한 줄에 표시(기본값)



flex-3.html

```
.opt1{  
  flex-flow: row wrap;  
}  
.opt2{  
  flex-flow: row nowrap;  
}
```

❖ 플렉스 박스 레이아웃(Flex Box Layout)

1. 플렉스 컨테이너 속성

✓ justify-content 속성

- 플렉스 아이템의 수평 방향 정렬 방식을 설정
 - justify-content: flex-start; 시작점에 맞춰 배치
 - justify-content: flex-end; 끝점에 맞춰 배치
 - justify-content: center; 중앙에 맞춰 배치
 - justify-content: space-between; 첫 아이템과 끝 아이템을 좌/우에 배치한 후,
나머지 아이템은 그 사이에 같은 간격으로 배치
 - justify-content: space-around; 모든 아이템을 같은 간격으로 배치



flex-4.html

```
.container {  
  display: flex;  
  background-color: #eee;  
  border: 1px solid #222;  
  margin-bottom: 20px;  
}  
.opt1 {  
  justify-content: flex-start;  
}  
.opt2 {  
  justify-content: flex-end;  
}  
.opt3 {  
  justify-content: center;  
}  
.opt4 {  
  justify-content: space-between;  
}  
.opt5 {  
  justify-content: space-around;  
}
```

❖ 플렉스 박스 레이아웃(Flex Box Layout)

1. 플렉스 컨테이너 속성

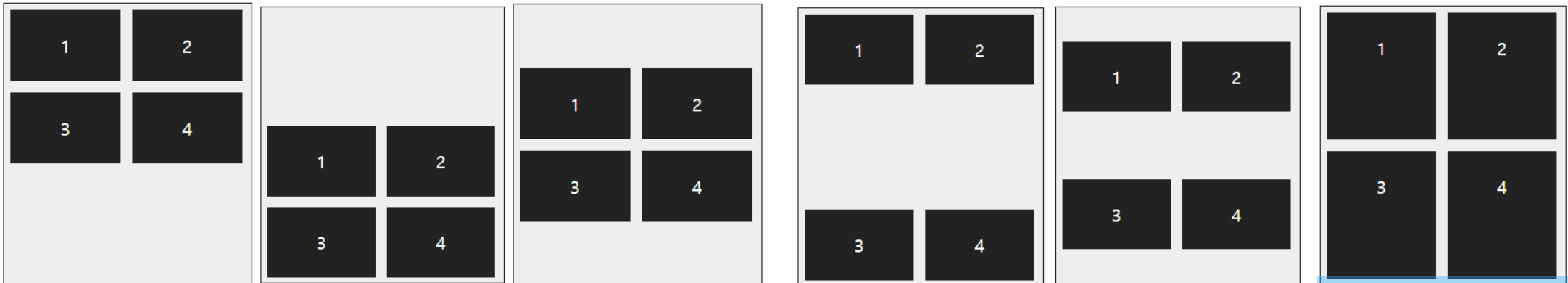
✓ align-content 속성

- 교차 축에서 플렉스 아이템 간격을 지정
 - align-content: flex-start; 시작점에 맞춰 배치
 - align-content: flex-end; 끝점에 맞춰 배치
 - align-content: center; 중앙에 맞춰 배치
 - align-content: space-between; 첫 아이템과 끝 아이템을 상/하에 배치한 후, 나머지 아이템은 그 사이에 같은 간격으로 배치
 - align-content: space-around; 모든 아이템을 같은 간격으로 배치
 - align-content: stretch; 아이템을 늘려서 간격 없이 배치

```
.container {  
  width:220px;  
  height:250px;  
  display:flex;  
  flex-flow: row wrap;  
  background-color:#eee;  
  border:1px solid #222;  
  margin-bottom:20px;  
}
```

flex-5.html

```
.opt1{  
  align-content: flex-start;  
}  
.opt2{  
  align-content: flex-end;  
}  
.opt3{  
  align-content: center;  
}  
.opt4{  
  align-content: space-between;  
}  
.opt5{  
  align-content: space-around;  
}  
.opt6{  
  align-content: stretch;  
}
```



❖ 플렉스 박스 레이아웃(Flex Box Layout)

1. 플렉스 컨테이너 속성

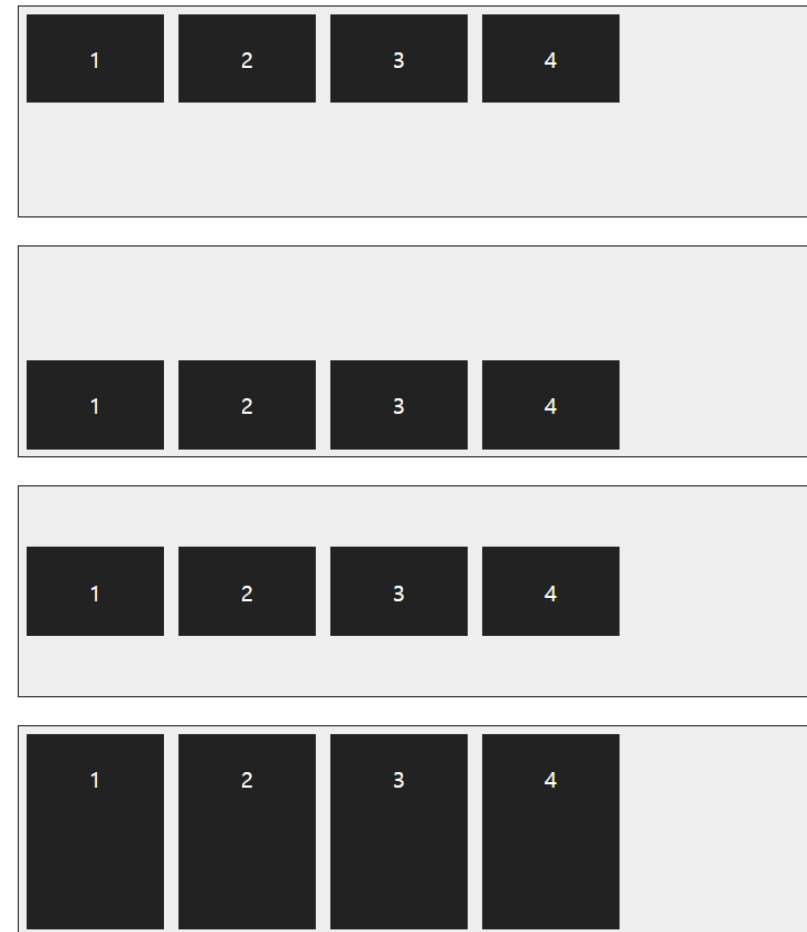
✓ align-items 속성

- 교차 축에서 플렉스 아이템의 상대 위치와 크기를 지정 (플렉스 요소의 수직 방향 정렬 방식을 설정)
 - align-items: flex-start; 세로로 맨 위에 배치
 - align-items: flex-end; 세로로 맨 아래 배치
 - align-items: center; 세로로 중앙에 맞춰 배치
 - align-items: baseline; 세로로 기준선에 배치
 - align-items: stretch; 아이템을 늘려서 간격 없이 배치

flex-6.html

```
.container {  
  width:100%;  
  height:150px;  
  display:flex;  
  background-color:#eee;  
  border:1px solid #222;  
  margin-bottom:20px;  
}
```

```
.opt1{  
  align-items: flex-start;  
}  
.opt2{  
  align-items: flex-end;  
}  
.opt3{  
  align-items: center;  
}  
.opt4{  
  align-items: stretch;  
}
```

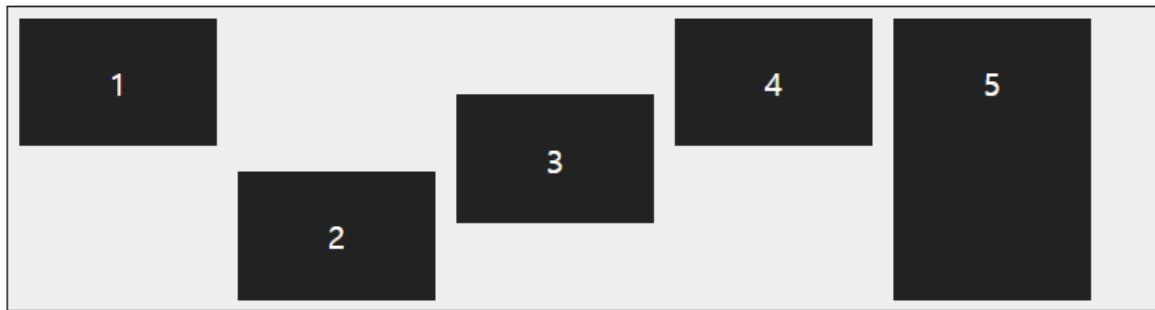


❖ 플렉스 박스 레이아웃(Flex Box Layout)

2. 플렉스 아이템 속성

✓ align-self

- 플렉스 아이템마다 서로 다른 align 속성값을 설정함
- align-items 속성은 플렉스 컨테이너에서 아이템의 배치 방법을 한꺼번에 조절하는 반면, align-self는 플렉스 아이템을 따로 원하는 형태로 배치할 수 있음
 - align-self: flex-start; 세로로 맨 위에 배치
 - align-self: flex-end; 세로로 맨 아래 배치
 - align-self: center; 세로로 중앙에 맞춰 배치
 - align-self: baseline; 세로로 기준선에 배치
 - align-self: stretch; 아이템을 늘려서 간격 없이 배치



```
<div class="container">|
  <div class="box box1"><p>1</p></div>
  <div class="box box2"><p>2</p></div>
  <div class="box box3"><p>3</p></div>
  <div class="box box4"><p>4</p></div>
  <div class="box box5"><p>5</p></div>
</div>
```

flex-7.html

```
.box1{
  align-self: flex-start;
}
.box2{
  align-self: flex-end;
}
.box3{
  align-self: center;
}
.box4{
  align-self: baseline;
}
.box5{
  align-self: stretch;
}
```


❖ 플렉스 박스 레이아웃(Flex Box Layout)

2. 플렉스 아이템 속성

✓ flex 속성

- 같은 플렉스 컨테이너 안에 있는 플렉스 요소의 너비를 상대적으로 설정함.
 - flex: n; 늘릴 비율

flex-8.html

```
.box1{  
  flex:3;  
}  
.box2{  
  flex:1;  
}  
.box3{  
  flex:2;  
}
```

```
<div class="container">  
  <div class="box box1"><p>1</p></div>  
  <div class="box box2"><p>2</p></div>  
  <div class="box box3"><p>3</p></div>  
</div>
```

