

 **김쌤과 함께 쉽고 빠르게**

부트스트랩4를 활용한 반응형 웹 만들기



STEP 03

부트스트랩4에 추가된 레이아웃 관련 클래스

INDEX

패딩 마진 클래스

row columns 클래스

[1] `.m(l, r, x, t, b, y)-*` / `.p(l, r, x, t, b, y)-*` 클래스명을 사용하면
margin과 padding을 좌우상단 원하는 위치에 지정할 수 있다.

```
.m-0 {  
  margin: 0 !important;  
}
```

```
.m-5 {  
  margin: 3rem !important;  
}
```

```
.p-0 {  
  padding: 0 !important;  
}
```

```
.p-5 {  
  padding: 3rem !important;  
}
```

```
.mt-0,  
.my-0 {  
  margin-top: 0 !important;  
}
```

```
.mt-5,  
.my-5 {  
  margin-top: 3rem !important;  
}
```

```
.pt-0,  
.py-0 {  
  padding-top: 0 !important;  
}
```

```
.pt-5,  
.py-5 {  
  padding-top: 3rem !important;  
}
```

```
.mr-0,  
.mx-0 {  
  margin-right: 0 !important;  
}
```

```
.mr-5,  
.mx-5 {  
  margin-right: 3rem !important;  
}
```

```
.pr-0,  
.px-0 {  
  padding-right: 0 !important;  
}
```

```
.pr-5,  
.px-5 {  
  padding-right: 3rem !important;  
}
```

```
.mb-0,  
.my-0 {  
  margin-bottom: 0 !important;  
}
```

```
.mb-5,  
.my-5 {  
  margin-bottom: 3rem !important;  
}
```

```
.pb-0,  
.py-0 {  
  padding-bottom: 0 !important;  
}
```

```
.pb-5,  
.py-5 {  
  padding-bottom: 3rem !important;  
}
```

```
.ml-0,  
.mx-0 {  
  margin-left: 0 !important;  
}
```

```
.ml-5,  
.mx-5 {  
  margin-left: 3rem !important;  
}
```

```
.pl-0,  
.px-0 {  
  padding-left: 0 !important;  
}
```

```
.pl-5,  
.px-5 {  
  padding-left: 3rem !important;  
}
```

<https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/css/bootstrap.css>

[2] `.m(l, r, x, t, b, y)-n*` 클래스명을 사용하면
margin을 좌우상단 원하는 위치에 negative값으로 지정할 수 있으며
`.m(l, r, x, t, b, y)-auto` 클래스명을 사용하여 콘텐츠를 중간으로 배치할 수 있다.

```
.m-n1 {  
  margin: -0.25rem !important;  
}
```

```
.mt-n1,  
.my-n1 {  
  margin-top: -0.25rem !important;  
}
```

```
.mr-n1,  
.mx-n1 {  
  margin-right: -0.25rem !important;  
}
```

```
.mb-n1,  
.my-n1 {  
  margin-bottom: -0.25rem !important;  
}
```

```
.ml-n1,  
.mx-n1 {  
  margin-left: -0.25rem !important;  
}
```

```
.m-n5 {  
  margin: -3rem !important;  
}
```

```
.mt-n5,  
.my-n5 {  
  margin-top: -3rem !important;  
}
```

```
.mr-n5,  
.mx-n5 {  
  margin-right: -3rem !important;  
}
```

```
.mb-n5,  
.my-n5 {  
  margin-bottom: -3rem !important;  
}
```

```
.ml-n5,  
.mx-n5 {  
  margin-left: -3rem !important;  
}
```

```
.m-auto {  
  margin: auto !important;  
}
```

```
.mt-auto,  
.my-auto {  
  margin-top: auto !important;  
}
```

```
.mr-auto,  
.mx-auto {  
  margin-right: auto !important;  
}
```

```
.mb-auto,  
.my-auto {  
  margin-bottom: auto !important;  
}
```

```
.ml-auto,  
.mx-auto {  
  margin-left: auto !important;  
}
```

[3] `.m(l, r, x, t, b, y)-(media size)-*` / `.p(l, r, x, t, b, y)-(media size)-*`
 클래스명을 사용하면 margin과 padding을 디바이스 사이즈 별로 조정가능하다.

```
@media (min-width: 576px) {
  .m-sm-0 {
    margin: 0 !important;
  }
  .mt-sm-0,
  .my-sm-0 {
    margin-top: 0 !important;
  }
  .mr-sm-0,
  .mx-sm-0 {
    margin-right: 0 !important;
  }
  .mb-sm-0,
  .my-sm-0 {
    margin-bottom: 0 !important;
  }
  .ml-sm-0,
  .mx-sm-0 {
    margin-left: 0 !important;
  }
  .m-sm-1 {
    margin: 0.25rem !important;
  }
  .p-sm-0 {
    padding: 0 !important;
  }
  .pt-sm-0,
  .py-sm-0 {
    padding-top: 0 !important;
  }
  .pr-sm-0,
  .px-sm-0 {
    padding-right: 0 !important;
  }
  .pb-sm-0,
  .py-sm-0 {
    padding-bottom: 0 !important;
  }
  .pl-sm-0,
  .px-sm-0 {
    padding-left: 0 !important;
  }
  .m-xl-n5 {
    margin: -3rem !important;
  }
  .mt-xl-n5,
  .my-xl-n5 {
    margin-top: -3rem !important;
  }
  .mr-xl-n5,
  .mx-xl-n5 {
    margin-right: -3rem !important;
  }
  .mb-xl-n5,
  .my-xl-n5 {
    margin-bottom: -3rem !important;
  }
  .ml-xl-n5,
  .mx-xl-n5 {
    margin-left: -3rem !important;
  }
  .m-xl-auto {
    margin: auto !important;
  }
  .mt-xl-auto,
  .my-xl-auto {
    margin-top: auto !important;
  }
  .mr-xl-auto,
  .mx-xl-auto {
    margin-right: auto !important;
  }
  .mb-xl-auto,
  .my-xl-auto {
    margin-bottom: auto !important;
  }
  .ml-xl-auto,
  .mx-xl-auto {
    margin-left: auto !important;
  }
}
```

<https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/css/bootstrap.css>

[1] `.row`에 `.row-cols-*`를 추가하여
클래스명을 사용하면 세로로 쌓이는 카드 형식의 레이아웃을 쉽게 구현할 수 있다.

Column	Column
Column	Column

```
<div class="container">
  <div class="row row-cols-2">
    <div class="col">Column</div>
    <div class="col">Column</div>
    <div class="col">Column</div>
    <div class="col">Column</div>
  </div>
</div>
```

[Copy](#)

Column	Column	Column
Column		

```
<div class="container">
  <div class="row row-cols-3">
    <div class="col">Column</div>
    <div class="col">Column</div>
    <div class="col">Column</div>
    <div class="col">Column</div>
  </div>
</div>
```

[Copy](#)

<https://getbootstrap.com/docs/4.6/layout/grid/#row-columns>

[2] `.row`에 `.row-cols-*`를 추가할 때 자식요소로 `.col(-*)`클래스를 주어 컨텐츠 영역의 비율도 조절 가능하다.

Column	Column	Column
Column		

```
<div class="container">
  <div class="row row-cols-4">
    <div class="col">Column</div>
    <div class="col">Column</div>
    <div class="col-6">Column</div>
    <div class="col">Column</div>
  </div>
</div>
```

[Copy](#)

Column	Column	Column
Column		

```
<div class="container">
  <div class="row row-cols-4">
    <div class="col">Column</div>
    <div class="col">Column</div>
    <div class="col-6">Column</div>
    <div class="col">Column</div>
  </div>
</div>
```

<https://getbootstrap.com/docs/4.6/layout/grid/#row-columns>

[3] `.row`에 `.row-cols-*`를 디바이스 사이즈별로 추가하여
반응형 레이아웃을 쉽게 구현할 수 있다.

Column	Column	Column	Column
--------	--------	--------	--------

```
<div class="container">
  <div class="row row-cols-1 row-cols-sm-2 row-cols-md-4">
    <div class="col">Column</div>
    <div class="col">Column</div>
    <div class="col">Column</div>
    <div class="col">Column</div>
  </div>
</div>
```

[Copy](#)

Column	Column
Column	Column





```
<div class="container">
  <div class="row row-cols-1 row-cols-sm-2 row-cols-md-4">
    <div class="col">Column</div>
    <div class="col">Column</div>
    <div class="col">Column</div>
    <div class="col">Column</div>
  </div>
</div>
```

Column
Column
Column
Column

```
<div class="container">
  <div class="row row-cols-1 row-cols-sm-2 row-cols-md-4">
    <div class="col">Column</div>
    <div class="col">Column</div>
    <div class="col">Column</div>
    <div class="col">Column</div>
  </div>
</div>
```

<https://getbootstrap.com/docs/4.6/layout/grid/#row-columns>

꼭 기억해야 할 포인트 정리

1.  클래스명을 사용하면 `margin-left`, `margin-right`를 `3rem`으로 가장 크게 주어 좌우 콘텐츠와의 간격을 조절할 수 있다.
2.  클래스명을 사용하면 `margin-top`, `margin-bottom`를 `-3rem` 으로 가장 크게 마이너스로 주어 상하 콘텐츠와의 간격을 좁힐 수 있다.
3. 클래스명 `.row` 에 클래스명  을 추가하여 클래스명을 사용하면 자식 요소인 `.col`이 세 개를 초과하면 나머지가 세로로 쌓이는 카드 형식의 레이아웃을 쉽게 구현할 수 있다.
4. 클래스명 `.row`에 `.row-cols-*`를 추가할 때 자식 요소로  클래스를 주면 콘텐츠 영역의 비율을 `1/2`로 조절 가능하다.




꼭 기억해야 할 포인트 정리

1. **.mx-5** 클래스명을 사용하면 margin-left, margin-right를 3rem으로 가장 크게 주어 좌우 콘텐츠와의 간격을 조절할 수 있다.
2. 클래스명을 사용하면 margin-top, margin-bottom를 -3rem 으로 가장 크게 마이너스로 주어 상하 콘텐츠와의 간격을 좁힐 수 있다.
3. 클래스명 .row 에 클래스명 을 추가하여 클래스명을 사용하면 자식 요소인 .col이 세 개를 초과하면 나머지가 세로로 쌓이는 카드 형식의 레이아웃을 쉽게 구현할 수 있다.
4. 클래스명 .row에 .row-cols-*를 추가할 때 자식 요소로 클래스를 주면 콘텐츠 영역의 비율을 1/2로 조절 가능하다.

꼭 기억해야 할 포인트 정리

1. **.mx-5** 클래스명을 사용하면 margin-left, margin-right를 3rem으로 가장 크게 주어 좌우 콘텐츠와의 간격을 조절할 수 있다.
2. **.my-n5** 클래스명을 사용하면 margin-top, margin-bottom를 -3rem 으로 가장 크게 마이너스로 주어 상하 콘텐츠와의 간격을 좁힐 수 있다.
3. 클래스명 .row 에 클래스명 을 추가하여 클래스명을 사용하면 자식 요소인 .col이 세 개를 초과하면 나머지가 세로로 쌓이는 카드 형식의 레이아웃을 쉽게 구현할 수 있다.
4. 클래스명 .row에 .row-cols-*를 추가할 때 자식 요소로 클래스를 주면 콘텐츠 영역의 비율을 1/2로 조절 가능하다.

꼭 기억해야 할 포인트 정리

1. **.mx-5** 클래스명을 사용하면 margin-left, margin-right를 3rem으로 가장 크게 주어 좌우 콘텐츠와의 간격을 조절할 수 있다.
2. **.my-n5** 클래스명을 사용하면 margin-top, margin-bottom를 -3rem 으로 가장 크게 마이너스로 주어 상하 콘텐츠와의 간격을 좁힐 수 있다.
3. 클래스명 .row 에 클래스명 **.row-cols-3** 을 추가하여 클래스명을 사용하면 자식 요소인 .col이 세 개를 초과하면 나머지가 세로로 쌓이는 카드 형식의 레이아웃을 쉽게 구현할 수 있다.
4. 클래스명 .row에 .row-cols-*를 추가할 때 자식 요소로  클래스를 주면 콘텐츠 영역의 비율을 1/2로 조절 가능하다.

꼭 기억해야 할 포인트 정리

1. **.mx-5** 클래스명을 사용하면 margin-left, margin-right를 3rem으로 가장 크게 주어 좌우 콘텐츠와의 간격을 조절할 수 있다.
2. **.my-n5** 클래스명을 사용하면 margin-top, margin-bottom를 -3rem 으로 가장 크게 마이너스로 주어 상하 콘텐츠와의 간격을 좁힐 수 있다.
3. 클래스명 .row 에 클래스명 **.row-cols-3** 을 추가하여 클래스명을 사용하면 자식 요소인 .col이 세 개를 초과하면 나머지가 세로로 쌓이는 카드 형식의 레이아웃을 쉽게 구현할 수 있다.
4. 클래스명 .row에 .row-cols-*를 추가할 때 자식 요소로 **.col-6** 클래스를 주면 콘텐츠 영역의 비율을 1/2로 조절 가능하다.

영상 코딩 디자인은 영코디 김쌤



영코디 김쌤

