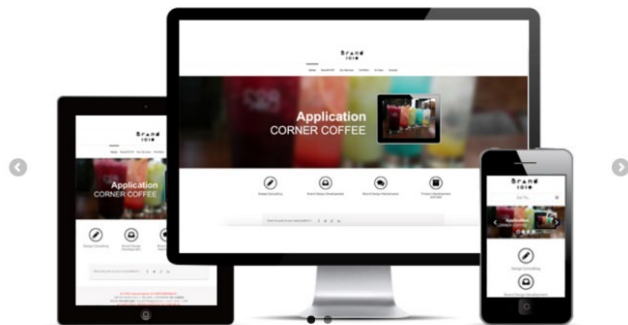


 **김쌤과 함께 쉽고 빠르게**

부트스트랩4를 활용한 반응형 웹 만들기



STEP 08

다양한 컴포넌트3_캐로셀부터 폼까지

INDEX

캐로셀

콜랩스

드랍다운

폼 양식

.carousel / .slide / .carousel-inner / .carousel-item

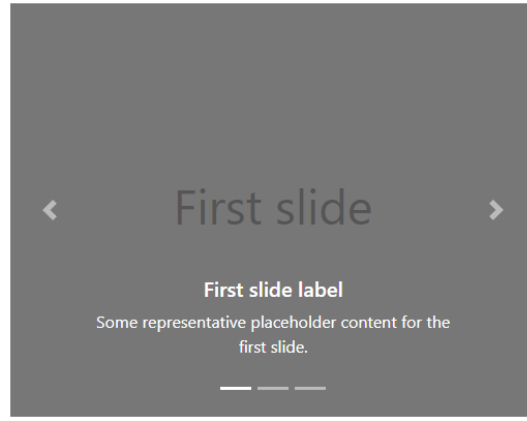
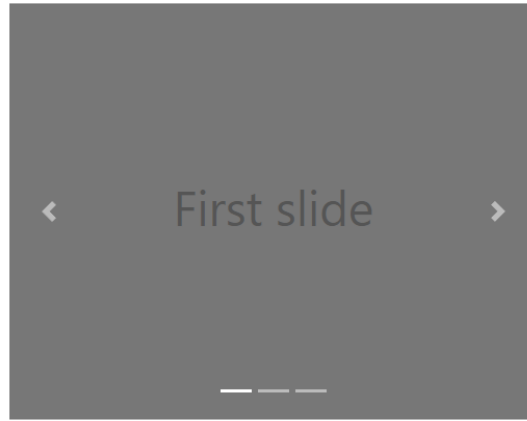


Second slide

```
<div id="carouselExampleSlidesOnly" class="carousel slide" data-ride="carousel">
  <div class="carousel-inner">
    <div class="carousel-item active">
      
    </div>
    <div class="carousel-item">
      
    </div>
    <div class="carousel-item">
      
    </div>
  </div>
</div>
```

캐러셀은 CSS 3D 변환과 약간의 JavaScript로 구축된 일련의 콘텐츠를 순환하는 슬라이드쇼이다. 일련의 이미지, 텍스트 또는 사용자 정의 마크업과 함께 작동한다

<button>.carousel-control-prev / . carousel-control-prev-icon
.carousel-indicators / <div>.carousel-caption



```
<div id="carouselExampleCaptions" class="carousel slide" data-ride="carousel">
  <ol class="carousel-indicators">
    <li data-target="#carouselExampleCaptions" data-slide-to="0" class="active"></li>
    <li data-target="#carouselExampleCaptions" data-slide-to="1"></li>
    <li data-target="#carouselExampleCaptions" data-slide-to="2"></li>
  </ol>
  <div class="carousel-inner">
    <div class="carousel-item active">
      
      <div class="carousel-caption d-none d-md-block">
        <h5>First slide label</h5>
        <p>Some representative placeholder content for the first slide.</p>
      </div>
    </div>
  </div>
  <button class="carousel-control-prev" type="button" data-target="#carouselExampleCaptions" data-slide="prev">
    <span class="carousel-control-prev-icon" aria-hidden="true"></span>
    <span class="sr-only">Previous</span>
  </button>
  <button class="carousel-control-next" type="button" data-target="#carouselExampleCaptions" data-slide="next">
    <span class="carousel-control-next-icon" aria-hidden="true"></span>
    <span class="sr-only">Next</span>
  </button>
</div>
```

슬라이드만 보이는 구조, 좌우 화살표 버튼을 포함한 구조, 하단에 인디케이터를 포함한 구조, 설명문구인 캡션을 포함한 구조 등의 기본 구조를 지원한다.

.carousel-fade / [data-interval="10000"]
/ [data-touch="false", data-interval="false"]

```
<div id="carouselExampleFade" class="carousel slide carousel-fade" data-ride="carousel">  
  <div class="carousel-inner">  
    <div class="carousel-item active">  
        
    </div>  
  </div>
```

```
<div id="carouselExampleInterval" class="carousel slide" data-ride="carousel">  
  <div class="carousel-inner">  
    <div class="carousel-item active" data-interval="10000">  
        
    </div>  
  </div>
```

```
<div id="carouselExampleControlsNoTouching" class="carousel slide" data-touch="false" data-interval="false">  
  <div class="carousel-inner">  
    <div class="carousel-item active">  
        
    </div>  
  </div>
```

슬라이딩시 fade효과를 주도록 하는 클래스명과 슬라이딩의 시간간격을 변경할 수 있는 속성, 터치스크린 화면에서 손가락으로 스위핑이 되는 기본 속성을 비활성화하는 속성들을 지원한다.

Via JavaScript

Call carousel manually with:

```
$('.carousel').carousel()
```

`.carousel('cycle')`

Cycles through the carousel items from left to right.

`.carousel('pause')`

Stops the carousel from cycling through items.

`.carousel(number) #`

Cycles the carousel to a particular frame (0 based, similar to an array). **Returns to the caller before the target item has been shown** (i.e. before the `slid.bs.carousel` event occurs).

`.carousel('prev')`

Cycles to the previous item. **Returns to the caller before the previous item has been shown** (i.e. before the `slid.bs.carousel` event occurs).

`.carousel('next')`

Options

Options can be passed via data attributes or JavaScript. For data attributes, append the option name to `data-`, as in `data-interval=""`.

Name	Type	Default	Description
interval	number	5000	The amount of time to delay between automatically cycling an item. If <code>false</code> , carousel will not automatically cycle.
keyboard	boolean	true	Whether the carousel should react to keyboard events.
pause	string boolean	'hover'	<p>If set to <code>'hover'</code>, pauses the cycling of the carousel on <code>mouseenter</code> and resumes the cycling of the carousel on <code>mouseleave</code>. If set to <code>false</code>, hovering over the carousel won't pause it.</p> <p>On touch-enabled devices, when set to <code>'hover'</code>, cycling will pause on <code>touchend</code> (once the user finished interacting with the carousel) for two intervals, before automatically resuming. Note that this is in addition to the above mouse behavior.</p>
ride	string	false	Autoplays the carousel after the user manually cycles the first item. If set to <code>'carousel'</code> , autoplays the carousel on load.
wrap	boolean	true	Whether the carousel should cycle continuously or have hard stops.
touch	boolean	true	Whether the carousel should support left/right swipe interactions on touchscreen devices.

Carousel() 메소드의 주어진 매개변수 값이나 속성값을 활용하면 캐로셀의 동작방식을 동적으로 다양하게 변경 가능하며 주어진 이벤트 타입을 활용하여 각 캐로셀 슬라이더가 동작하기 전과 동작한 바로 후에 원하는 기능을 추가 구현 가능하다.

.collapse

Link with href

Button with data-target



Link with href

Button with data-target

Some placeholder content for the collapse component. This panel is hidden by default but revealed when the user activates the relevant trigger.

```
<p>
  <a class="btn btn-primary" data-toggle="collapse" href="#collapseExample" role="button">
    Link with href
  </a>
  <button class="btn btn-primary" type="button" data-toggle="collapse" data-target="#collapseExample">
    Button with data-target
  </button>
</p>
<div class="collapse" id="collapseExample">
  <div class="card card-body">
    Some placeholder content for the collapse component. This panel is hidden by default.
  </div>
</div>
```

```
.collapse:not(.show) {
  display: none;
}

.collapsing {
  position: relative;
  height: 0;
  overflow: hidden;
  transition: height 0.35s ease;
}

.collapse:not(.show) {
  display: none;
}
```

버튼에 data-toggle="collapse" data-target="#아이디명" 속성이 있고 콘텐츠 요소에 .collapse id="아이디명" 이 있으면 버튼을 토글시 콘텐츠 요소의 클래스가 .collapse=>.collapsing =>.collapse.show 로 변화한다.

.collapse.width

Toggle width collapse



Toggle width collapse

This is some placeholder content for a horizontal collapse. It's hidden by default and shown when triggered.

This is some placeholder content for a horizontal collapse. It's hidden by default and shown when triggered.

```
<p>
  <button class="btn btn-primary" type="button" data-toggle="collapse" data-target="#collapseWidthExample" aria-expanded="false">
    Toggle width collapse
  </button>
</p>
<div style="min-height: 120px;">
  <div class="collapse width" id="collapseWidthExample">
    <div class="card card-body" style="width: 320px;">
      This is some placeholder content for a horizontal collapse. It's hidden by default and shown when triggered.
    </div>
  </div>
</div>
```

.collapse를 가진 요소에 .width를 주고 그 하위 요소의 스타일에 width값을 부여하면 수평으로 펼쳐지도록 구현한다.

. collapse.multi-collapse

[Toggle first element](#)[Toggle second element](#)[Toggle both elements](#)

Some placeholder content for the first collapse component of this multi-collapse example. This panel is hidden by default but revealed when the user activates the relevant trigger.

Some placeholder content for the second collapse component of this multi-collapse example. This panel is hidden by default but revealed when the user activates the relevant trigger.

```
<p>
  <a class="btn btn-primary" data-toggle="collapse" href="#multiCollapseExample1" role="button" aria-expanded="fa
  <button class="btn btn-primary" type="button" data-toggle="collapse" data-target="#multiCollapseExample2" aria-
  <button class="btn btn-primary" type="button" data-toggle="collapse" data-target=".multi-collapse" aria-expande
</p>
<div class="row">
  <div class="col">
    <div class="collapse multi-collapse" id="multiCollapseExample1">
      <div class="card card-body">
        Some placeholder content for the first collapse component of this multi-collapse example. This panel is h
      </div>
    </div>
  </div>
  <div class="col">
    <div class="collapse multi-collapse" id="multiCollapseExample2">
```

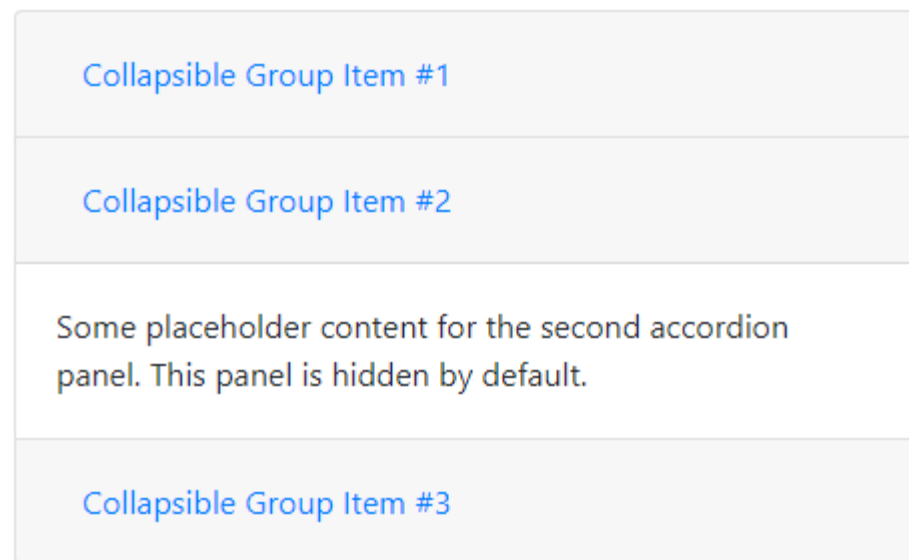
[Copy](#)

버튼요소의 #아이디명과 콘텐츠 요소의 id명을 일치시키면 여러 개의 콜랩스 요소를 만들 수 있다.

.accordion

Accordion example

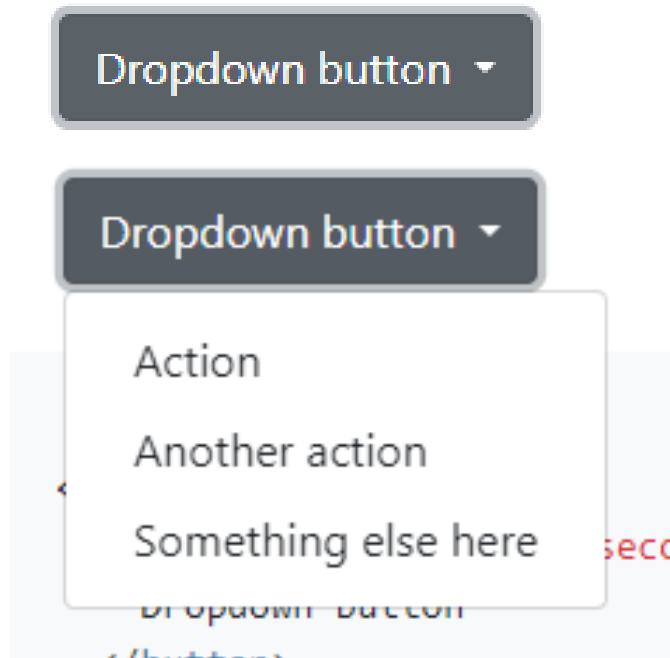
Using the `card` component, you can extend the default collapse behavior to create an accordion. To properly achieve the accordion style, be sure to use `.accordion` as a wrapper.



```
<div class="accordion" id="accordionExample">
  <div class="card">
    <div class="card-header" id="headingOne">
      <h2 class="mb-0">
        <button class="btn btn-link btn-block text-left" type="button" data-toggle="collapse" data-target="#collapseOne" aria-expanded="true" data-keybase="collapse">Collapsible Group Item #1
      </button>
    </h2>
    </div>
    <div id="collapseOne" class="collapse show" aria-labelledby="headingOne" data-parent="#accordionExample">
      <div class="card-body">
        Some placeholder content for the first accordion panel. This panel is shown
      </div>
    </div>
  </div>
  <div class="card">
    <div class="card-header" id="headingTwo">
      <h2 class="mb-0">
        <button class="btn btn-link btn-block text-left" type="button" data-toggle="collapse" data-target="#collapseTwo" aria-expanded="false" data-keybase="collapse">Collapsible Group Item #2
      </button>
    </h2>
    </div>
    <div id="collapseTwo" class="collapse" data-parent="#accordionExample">
      <div class="card-body">
        Some placeholder content for the second accordion panel. This panel is hidden by default.
      </div>
    </div>
  </div>
  <div class="card">
    <div class="card-header" id="headingThree">
      <h2 class="mb-0">
        <button class="btn btn-link btn-block text-left" type="button" data-toggle="collapse" data-target="#collapseThree" aria-expanded="false" data-keybase="collapse">Collapsible Group Item #3
      </button>
    </h2>
    </div>
    <div id="collapseThree" class="collapse" data-parent="#accordionExample">
      <div class="card-body">
        Some placeholder content for the third accordion panel. This panel is hidden by default.
      </div>
    </div>
  </div>
</div>
```

카드 구성요소를 사용하면 기본 축소 동작을 확장하여 아코디언을 만들 수 있다. 아코디언 스타일을 제대로 구현하려면 `.accordion`을 래퍼로 사용한다.

.dropdown-toggle / .dropdown-menu / .dropdown-item



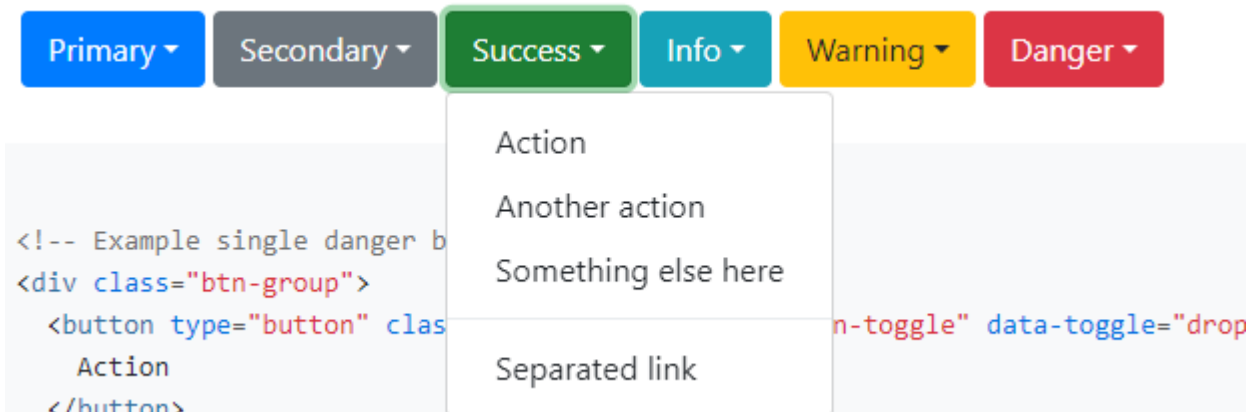
```
<div class="dropdown">
  <button class="btn btn-secondary dropdown-toggle" type="button" data-toggle="dropdown">
    Dropdown button
  </button>
  <div class="dropdown-menu">
    <a class="dropdown-item" href="#">Action</a>
    <a class="dropdown-item" href="#">Another action</a>
    <a class="dropdown-item" href="#">Something else here</a>
  </div>
</div>

<div class="dropdown">
  <a class="btn btn-secondary dropdown-toggle" href="#" role="button" data-toggle="dropdown">
    Dropdown link
  </a>

  <div class="dropdown-menu">
    <a class="dropdown-item" href="#">Action</a>
    <a class="dropdown-item" href="#">Another action</a>
    <a class="dropdown-item" href="#">Something else here</a>
  </div>
</div>
```

<a> 또는 <button> 태그에 .dropdown-toggle를 주고 콘텐츠 요소에 .dropdown-menu, 그 하위 메뉴 리스트에 .dropdown-item를 주어 링크 목록을 표시하거나 숨긴다. 바탕을 클릭하여 드롭다운 해제도 가능하다.

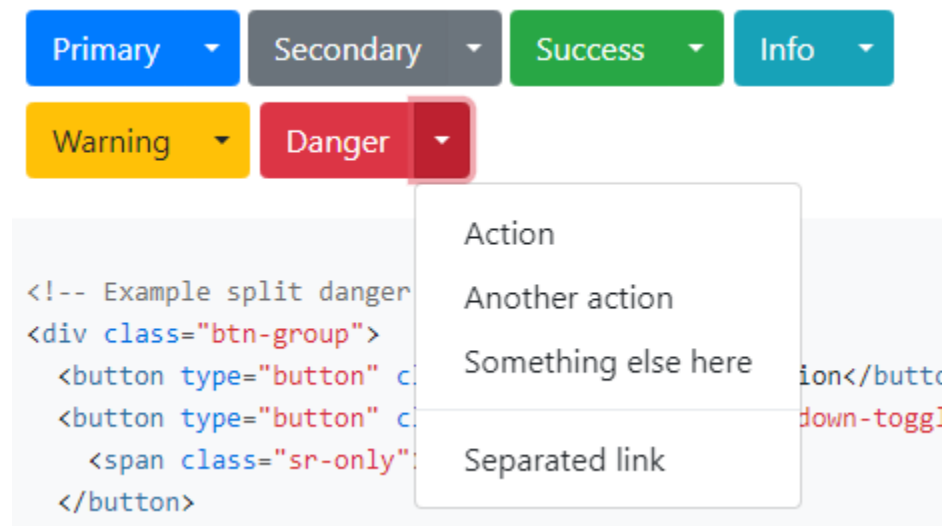
.btn-group / .dropdown-toggle / .dropdown-toggle-split / .btn-테마명



Primary Secondary Success Info Warning Danger

Action
Another action
Something else here
Separated link

```
<!-- Example single danger button -->
<div class="btn-group">
  <button type="button" class="btn btn-danger dropdown-toggle" data-toggle="dropdown">Action</button>
  <div class="dropdown-menu">
    <a class="dropdown-item" href="#">Action</a>
    <a class="dropdown-item" href="#">Another action</a>
    <a class="dropdown-item" href="#">Something else here</a>
    <div class="dropdown-divider"></div>
    <a class="dropdown-item" href="#">Separated link</a>
  </div>
</div>
```



Primary Secondary Success Info Warning Danger

Action
Another action
Something else here
Separated link

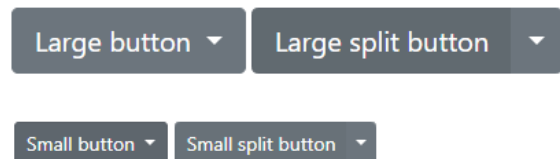
```
<!-- Example split danger button -->
<div class="btn-group">
  <button type="button" class="btn btn-danger">Warning</button>
  <button type="button" class="btn btn-danger dropdown-toggle-split" data-toggle="dropdown"></button>
  <div class="dropdown-menu">
    <a class="dropdown-item" href="#">Action</a>
    <a class="dropdown-item" href="#">Another action</a>
    <a class="dropdown-item" href="#">Something else here</a>
    <div class="dropdown-divider"></div>
    <a class="dropdown-item" href="#">Separated link</a>
  </div>
</div>
```

버튼을 변형하여 다양한 색을 주거나 분할된 버튼으로 응용 가능하다.

.btn. btn-lg / .btn. btn-sm / .btn-group.dropdown / .btn-group.dropright / .btn-group.dropleft

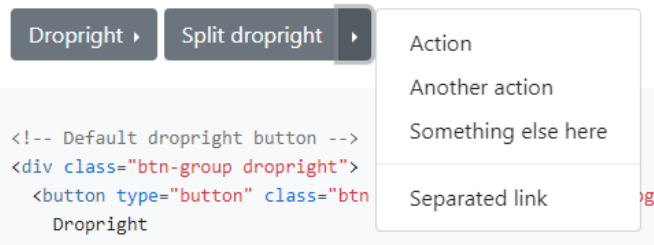
Sizing

Button dropdowns work with buttons of all sizes, including default and split dropdown buttons.



Dropright

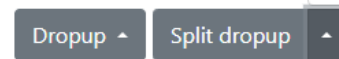
Trigger dropdown menus at the right of the elements by adding `.dropright` to the parent element.



Directions

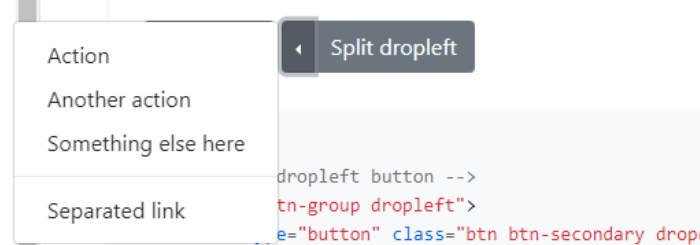
Dropup

Trigger dropdown menus above the parent element.



Dropleft

Trigger dropdown menus at the left of the elements by adding `.dropleft` to the parent element.

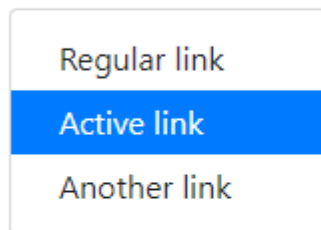


버튼 드롭다운은 기본 및 분할 드롭다운 버튼을 포함하여 모든 크기의 버튼에서 작동하며 드롭다운 메뉴를 상, 하, 좌, 우 어떤 방향으로든 동작시킬 수 있다.

.dropdown-item.active / .dropdown-item.disabled

Active

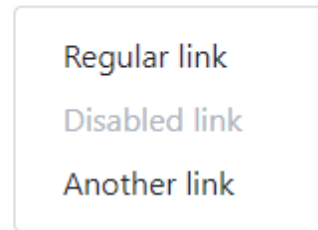
Add **.active** to items in the dropdown to **style them as active**.



```
<div class="dropdown-menu">
  <a class="dropdown-item" href="#">Regular link</a>
  <a class="dropdown-item active" href="#">Active link</a>
  <a class="dropdown-item" href="#">Another link</a>
</div>
```

Disabled

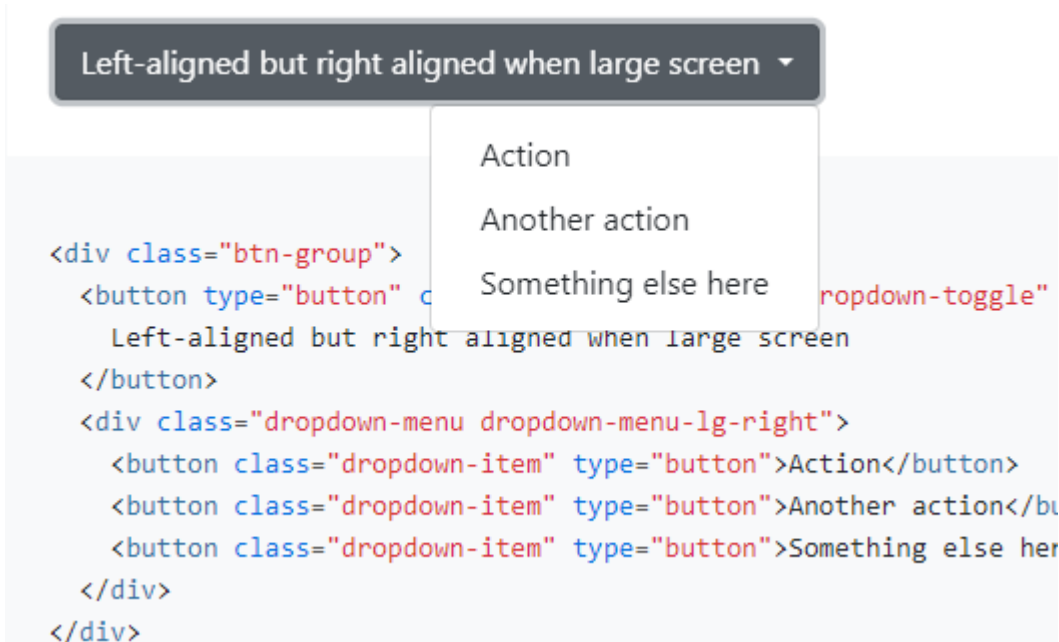
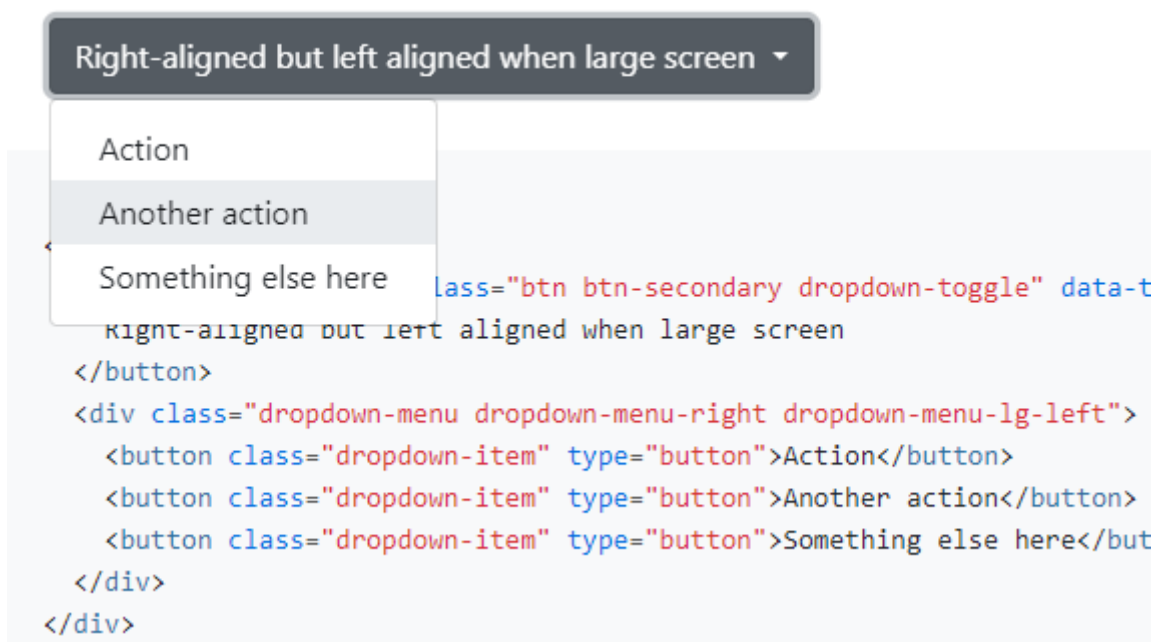
Add **.disabled** to items in the dropdown to **style them as disabled**.



```
<div class="dropdown-menu">
  <a class="dropdown-item" href="#">Regular link</a>
  <a class="dropdown-item disabled">Disabled link</a>
  <a class="dropdown-item" href="#">Another link</a>
</div>
```

.dropdown-item에 .active를 추가하여 활성 항목 스타일을 지정하거나 .disabled를 추가하여 비활성화 스타일을 지정한다.

.dropdown-menu.dropdown-menu-right
/ .dropdown-menu.dropdown-menu-left
/ .dropdown-menu.dropdown-menu-right.dropdown-menu-lg-left

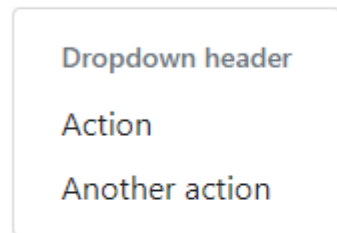


.dropdown-menu에 .dropdown-menu-right를 추가하여 버튼을 기준으로 메뉴가 우측정렬되도록 할 수 있으며 반응형으로 디바이스 사이즈에 따라서 정렬의 방향이 바뀌도록 지정 가능하다.

.dropdown-header / .dropdown-divider

Headers

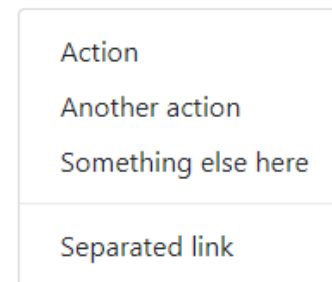
Add a header to label sections of actions in any dropdown menu.



```
<div class="dropdown-menu">
  <h6 class="dropdown-header">Dropdown header</h6>
  <a class="dropdown-item" href="#">Action</a>
  <a class="dropdown-item" href="#">Another action</a>
</div>
```

Dividers

Separate groups of related menu items with a divider.



```
<div class="dropdown-menu">
  <a class="dropdown-item" href="#">Action</a>
  <a class="dropdown-item" href="#">Another action</a>
  <a class="dropdown-item" href="#">Something else here</a>
  <div class="dropdown-divider"></div>
  <a class="dropdown-item" href="#">Separated link</a>
</div>
```

.dropdown-header를 주어 목록 메뉴 콘텐츠의 heading을 처리하거나 .dropdown-divider 를 주어 목록 메뉴를 구분할 수 있다.

. form-group / <input>. form-control

Email address

We'll never share your email with anyone else.

Password

☐ Check me out

Submit

```
<form>
  <div class="form-group">
    <label for="exampleInputEmail1">Email address</label>
    <input type="email" class="form-control" id="exampleInputEmail1" >
    <small id="emailHelp" class="form-text text-muted">We'll never sha
  </div>
  <div class="form-group">
    <label for="exampleInputPassword1">Password</label>
    <input type="password" class="form-control" id="exampleInputPassw
  </div>
  <div class="form-group form-check">
    <input type="checkbox" class="form-check-input" id="exampleCheck1"
    <label class="form-check-label" for="exampleCheck1">Check me out</
  </div>
  <button type="submit" class="btn btn-primary">Submit</button>
</form>
```

Bootstrap의 양식 스타일은 입력 영역 클릭시 파란색 보더로 활성화된다.

[type=file]. form-control-file / .form-control.form-control-lg

Example file input

파일 선택 선택된 파일 없음

```
<form>
  <div class="form-group">
    <label for="exampleFormControlFile1">Example file input</label>
    <input type="file" class="form-control-file" id="exampleFormCo
  </div>
</form>
```

Sizing

Set heights using classes like `.form-control-lg` and `.form-control-sm`.

`.form-control-lg`

Default input

`.form-control-sm`

```
<input class="form-control form-control-lg" type="text" placeholder="."
<input class="form-control" type="text" placeholder="Default input">
<input class="form-control form-control-sm" type="text" placeholder=".
```

Copy

파일을 선택하는 [type=file]의 스타일은 `.form-control-file` 을 주고 input영역의 크기를 조절하기 위해 `.form-control-
사이즈 클래스명을 준다.`

<input>[readonly]

Readonly plain text

If you want to have `<input readonly>` elements in your form styled as plain text, use the `.form-control-plaintext` class to remove the default form field styling and preserve the correct margin and padding.

Email	<input type="text" value="email@example.com"/>
Password	<input type="password"/>

```
<form>
  <div class="form-group row">
    <label for="staticEmail" class="col-sm-2 col-form-label">Email</label>
    <div class="col-sm-10">
      <input type="text" readonly class="form-control-plaintext" id="staticEmail">email@example.com
    </div>
  </div>
  <div class="form-group row">
    <label for="inputPassword" class="col-sm-2 col-form-label">Password</label>
    <div class="col-sm-10">
      <input type="password" class="form-control" id="inputPassword">
    </div>
  </div>
</form>
```

사용자가 텍스트 변경 불가능하고 스크린 리더기만 읽도록 하려면 `<input readonly>` 속성을 부여한다.

<input>.form-control-range

Range Inputs

Set horizontally scrollable range inputs using `.form-control-range`.

Example Range input



Copy

```
<form>
  <div class="form-group">
    <label for="formControlRange">Example Range input</label>
    <input type="range" class="form-control-range" id="formCor
  </div>
</form>
```

.form-control-range을 사용하여 가로로 스크롤 가능한 범위 입력을 설정한다.

.form-inline / .

Inline

email@example.com

Password

Confirm identity

```
<form class="form-inline">
  <div class="form-group mb-2">
    <label for="staticEmail2" class="sr-only">Email</label>
    <input type="text" readonly class="form-control-plaintext" id="staticEmail2">email@example.com
  </div>
  <div class="form-group mx-sm-3 mb-2">
    <label for="inputPassword2" class="sr-only">Password</label>
    <input type="password" class="form-control" id="inputPassword2" placeholder="Password">
  </div>
  <button type="submit" class="btn btn-primary mb-2">Confirm identity</button>
</form>
```

Form grid

More complex forms can be built using our grid classes. Use these for form layouts that require multiple columns, varied widths, and additional alignment options.

First name

Last name

```
<form>
  <div class="row">
    <div class="col">
      <input type="text" class="form-control" placeholder="First name">
    </div>
    <div class="col">
      <input type="text" class="form-control" placeholder="Last name">
    </div>
  </div>
</form>
```

Copy

.form-inline 클래스를 사용하면 일련의 라벨, 양식 컨트롤 및 버튼을 단일 수평 행에 표시할 수 있으며 컨트롤폭이 576px 이상인 뷰포트에 인라인으로 표시된다. .row에 .col을 그리드 형식으로 활용하여 구성할 수도 있다.

.form-group / <input>.form-control

Form row

You may also swap `.row` for `.form-row`, a variation of our standard grid row that overrides the default column gutters for tighter and more compact layouts.

[Copy](#)

```
<form>
  <div class="form-row">
    <div class="col">
      <input type="text" class="form-control" placeholder="First name">
    </div>
    <div class="col">
      <input type="text" class="form-control" placeholder="Last name">
    </div>
  </div>
</form>
```

Horizontal form

Email

Password

```
<form>
  <div class="form-group row">
    <label for="inputEmail3" class="col-sm-2 col-form-label">Email</label>
    <div class="col-sm-10">
      <input type="email" class="form-control" id="inputEmail3">
    </div>
  </div>
  <div class="form-group row">
    <label for="inputPassword3" class="col-sm-2 col-form-label">Password</label>
    <div class="col-sm-10">
      <input type="password" class="form-control" id="inputPassword3">
    </div>
  </div>
</form>
```

.row을 .form-row로 교체할 수도 있다. 이는 더 촘촘하고 컴팩트한 레이아웃을 위해 기본 열 여백을 재정의한다. 그룹을 형성하기 위해 .row 클래스를 추가하고 .col-사이즈-숫자 클래스를 사용하여 레이블과 컨트롤의 너비를 지정하여 그리드가 있는 수평 양식을 만든다.

<label>.col-form-label-sm / <input>. form-control-sm
/ <label>.col-form-label-lg / <input>. form-control-lg

Horizontal form label sizing

Be sure to use `.col-form-label-sm` or `.col-form-label-lg` to your `<label>`s or `<legend>`s to correctly follow the size of `.form-control-lg` and `.form-control-sm`.

Email

col-form-label-sm

Email

col-form-label

Email

col-form-label-lg

```
<form>
  <div class="form-group row">
    <label for="colFormLabelSm" class="col-sm-2 col-form-label col-form-label-sm">E
    <div class="col-sm-10">
      <input type="email" class="form-control form-control-sm" id="colFormLabelSm"
    </div>
  </div>
  <div class="form-group row">
    <label for="colFormLabel" class="col-sm-2 col-form-label">Email</label>
    <div class="col-sm-10">
      <input type="email" class="form-control" id="colFormLabel" placeholder="col-f
    </div>
  </div>
  <div class="form-group row">
    <label for="colFormLabelLg" class="col-sm-2 col-form-label col-form-label-lg">E
    <div class="col-sm-10">
      <input type="email" class="form-control form-control-lg" id="colFormLabelLg"
    </div>
  </div>
</form>
```

레이블의 크기와 입력영역의 크기를 조절하기 위한 클래스들을 활용한다.

.form-row / .col-auto

Column sizing

As shown in the previous examples, our grid system allows you to place any number of `.cols` within a `.row` or `.form-row`. They'll split the available width equally between them. You may also pick a subset of your columns to take up more or less space, while the remaining `.cols` equally split the rest, with specific column classes like `.col-7`.

City State Zip

```
<form>
  <div class="form-row">
    <div class="col-7">
      <input type="text" class="form-control" placeholder="City">
    </div>
    <div class="col">
      <input type="text" class="form-control" placeholder="State">
    </div>
    <div class="col">
      <input type="text" class="form-control" placeholder="Zip">
    </div>
  </div>
</form>
```

Auto-sizing

The example below uses a flexbox utility to vertically center the contents and changes `.col` to `.col-auto` so that your columns only take up as much space as needed. Put another way, the column sizes itself based on the contents.

Jane Doe @ Username
☐ Remember me

```
<form>
  <div class="form-row align-items-center">
    <div class="col-auto">
      <label class="sr-only" for="inlineFormInput">Name</label>
      <input type="text" class="form-control mb-2" id="inlineFormInput" placeh
    </div>
    <div class="col-auto">
      <label class="sr-only" for="inlineFormInputGroup">Username</label>
      <div class="input-group mb-2">
        <div class="input-group-prepend">
          <div class="input-group-text">@</div>
        </div>
      </div>
    </div>
  </div>
</form>
```

```
.col-auto {
  -ms-flex: 0 0 auto;
  flex: 0 0 auto;
  width: auto;
  max-width: 100%;
}
```

.col을 .col-auto로 변경하여 열이 필요한 만큼만 공간을 차지하도록 하면 내용에 따라 열 크기가 자체적으로 조정된다.

.form-group / <input>.form-control

Help text

Password

Password

Your password must be 8-20 characters long, contain letters and numbers, and must not contain spaces, special characters, or emoji.

```
<label for="inputPassword5">Password</label>
<input type="password" id="inputPassword5" class="form-control" aria-describedby="passwordHelpBlock">
<small id="passwordHelpBlock" class="form-text text-muted">
  Your password must be 8-20 characters long, contain letters and numbers,
</small>
```

Password

Must be 8-20 characters long.

```
<form class="form-inline">
  <div class="form-group">
    <label for="inputPassword6">Password</label>
    <input type="password" id="inputPassword6" class="form-control">
    <small id="passwordHelpInline" class="text-muted">
      Must be 8-20 characters long.
    </small>
  </div>
</form>
```

```
.form-text {
  display: block;
  margin-top: 0.25rem;
}
```

input태그 아래 도움말 텍스트는 .form-text을 사용하여 스타일을 지정할 수 있다. 이 클래스에는 display: block가 포함되어 있으며 위 입력에서 쉽게 간격을 둘 수 있도록 상단 여백을 추가하며 되며 .text-muted 를 이용하여 gray톤으로 도움말 텍스트를 작성할 수 있다.

<form>.needs-validation[novalidate] / <input>[required] / .valid-feedback
.form-control.is-valid / .form-control.is-invalid

Validation

First name

Looks good!

City

☐ Agree to terms and conditions
You must agree before submitting.

Submit form

Server side

Last name

Looks good!

State

Please select a valid state.

Zip

Please provide a valid zip.

```
<form class="needs-validation" novalidate>
  <div class="form-row">
    <div class="col-md-6 mb-3">
      <label for="validationCustom01">First name</label>
      <input type="text" class="form-control" id="validationCustom01" value="Mark" required>
      <div class="valid-feedback">
        Looks good!
      </div>
    </div>
```

폼 양식 미입력시 유효성 검증이 되도록 구현하는 client-side 방식과 처음부터 해당 항목에 유효성 검증 스타일이 뜨도록 하는 server side 방식 중 선택하여 구현한다.

<form>.needs-validation / <input>[required] / .valid-feedback

Tooltips

First name

 ✓

Looks good!

City

 ✓

Last name

 ✓

Looks good!

State

 ⚠

Please select a valid state.

Zip

 ✓

Submit form

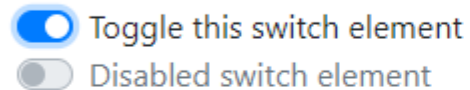
```
<form class="needs-validation" novalidate>
  <div class="form-row">
    <div class="col-md-6 mb-3">
      <label for="validationTooltip01">First name</label>
      <input type="text" class="form-control" id="validationTooltip01" value="Mark" required>
      <div class="valid-tooltip">
        Looks good!
      </div>
    </div>
  </div>
```

양식 제출 버튼 클릭시 유효성 검사 피드백을 표시를 툴팁으로 가능하다.

.custom-switch

Switches

A switch has the markup of a custom checkbox but uses the `.custom-switch` class to render a toggle switch. Switches also support the `disabled` attribute.



```
<div class="custom-control custom-switch">
  <input type="checkbox" class="custom-control-input" id="customSwitch1">
  <label class="custom-control-label" for="customSwitch1">Toggle this switch element</label>
</div>
<div class="custom-control custom-switch">
  <input type="checkbox" class="custom-control-input" disabled id="customSwitch2">
  <label class="custom-control-label" for="customSwitch2">Disabled switch element</label>
</div>
```

[Copy](#)

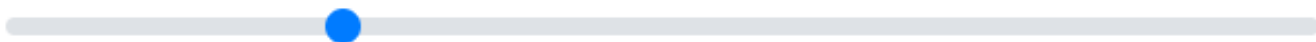
스위치에는 `.custom-switch` 클래스를 사용하여 토글 스위치를 렌더링한다. 스위치는 `disabled` 속성도 지원한다.

. custom-range

Range

Create custom `<input type="range">` controls with `.custom-range`. The track (the background) and thumb (the value) are both styled to appear the same across browsers. As only IE and Firefox support “filling” their track from the left or right of the thumb as a means to visually indicate progress, we do not currently support it.

Example range



```
<label for="customRange1">Example range</label>
<input type="range" class="custom-range" id="customRange1">
```

Copy

```
.custom-range {
  width: 100%;
  height: 1.4rem;
  padding: 0;
  background-color: transparent;
  -webkit-appearance: none;
  -moz-appearance: none;
  appearance: none;
}
```

`<input type="range">` 컨트롤에서는 각각 `min` 및 `max` 속성을 사용하여 새 값을 지정할 수 있다.

꼭 기억해야 할 포인트 정리

1. 캐로셀에서 슬라이딩의 시간간격을 2초로 변경하기 위해 .carousel-item에 속성을 `data-bs-interval="2000"`로 추가한다.
2. 캐로셀이 무한반복하여 슬라이딩하지 않도록 하기 위해서 .carousel() 메소드의 매개변수로 `infinite: false` 속성값을 true로 설정한다.
3. .dropdown-menu에 `data-bs-toggle="dropdown"`를 추가하여 버튼을 기준으로 메뉴가 우측정렬되도록 할 수 있으며 반응형으로 데스크탑에서는 좌측정렬하도록 하려면 `data-bs-toggle="dropdown" data-bs-direction="downward"`를 부여한다.
4. 폼 양식 미입력시 유효성 검증이 되도록 구현하는 client-side 방식을 사용하려면 <form> 태그에 `novalidate` 클래스명을 주고 속성으로는 `novalidate`을 부여한다.

꼭 기억해야 할 포인트 정리

1. 캐로셀에서 슬라이딩의 시간간격을 2초로 변경하기 위해 .carousel-item에 속성을 **data-interval="2000"** 로 추가한다.
2. 캐로셀이 무한반복하여 슬라이딩하지 않도록 하기 위해서 .carousel() 메소드의 매개변수로 **속성값을 true**로 설정한다.
3. .dropdown-menu에 **를 추가하여 버튼을 기준으로 메뉴가 우측정렬되도록 할 수 있으며 반응형으로 데스크탑에서는 좌측정렬하도록 하려면** **를 부여한다.**
4. 폼 양식 미입력시 유효성 검증이 되도록 구현하는 client-side 방식을 사용하려면 <form> 태그에 **클래스명을 주고 속성으로는** **을 부여한다.**

꼭 기억해야 할 포인트 정리

1. 캐로셀에서 슬라이딩의 시간간격을 2초로 변경하기 위해 .carousel-item에 속성을 **data-interval="2000"** 로 추가한다.
2. 캐로셀이 무한반복하여 슬라이딩하지 않도록 하기 위해서 .carousel() 메소드의 매개변수로 **wrap** 속성값을 true로 설정한다.
3. .dropdown-menu에 **dropdown-menu-right**를 추가하여 버튼을 기준으로 메뉴가 우측정렬되도록 할 수 있으며 반응형으로 데스크탑에서는 좌측정렬하도록 하려면 **dropdown-menu-left**를 부여한다.
4. 폼 양식 미입력시 유효성 검증이 되도록 구현하는 client-side 방식을 사용하려면 <form> 태그에 **novalidate** 클래스명을 주고 속성으로는 **novalidate**을 부여한다.

꼭 기억해야 할 포인트 정리

1. 캐로셀에서 슬라이딩의 시간간격을 2초로 변경하기 위해 `.carousel-item`에 속성을 `data-interval="2000"`로 추가한다.
2. 캐로셀이 무한반복하여 슬라이딩하지 않도록 하기 위해서 `.carousel()` 메소드의 매개변수로 `wrap` 속성값을 `true`로 설정한다.
3. `.dropdown-menu`에 `.dropdown-menu-right`를 추가하여 버튼을 기준으로 메뉴가 우측정렬되도록 할 수 있으며 반응형으로 데스크탑에서는 좌측정렬하도록 하려면 `.dropdown-menu-lg-left`를 부여한다.
4. 폼 양식 미입력시 유효성 검증이 되도록 구현하는 client-side 방식을 사용하려면 `<form>` 태그에 클래스명을 주고 속성으로는 을 부여한다.

꼭 기억해야 할 포인트 정리

1. 캐로셀에서 슬라이딩의 시간간격을 2초로 변경하기 위해 `.carousel-item`에 속성을 `data-interval="2000"` 로 추가한다.
2. 캐로셀이 무한반복하여 슬라이딩하지 않도록 하기 위해서 `.carousel()` 메소드의 매개변수로 `wrap` 속성값을 `true`로 설정한다.
3. `.dropdown-menu`에 `.dropdown-menu-right` 를 추가하여 버튼을 기준으로 메뉴가 우측정렬되도록 할 수 있으며 반응형으로 데스크탑에서는 좌측정렬하도록 하려면 `.dropdown-menu-lg-left` 를 부여한다.
4. 폼 양식 미입력시 유효성 검증이 되도록 구현하는 client-side 방식을 사용하려면 `<form>` 태그에 `.needs-validation` 클래스명을 주고 속성으로는 `[novalidate]` 을 부여한다.

영상 코딩 디자인은 영코디 김쌤



영코디 김쌤

