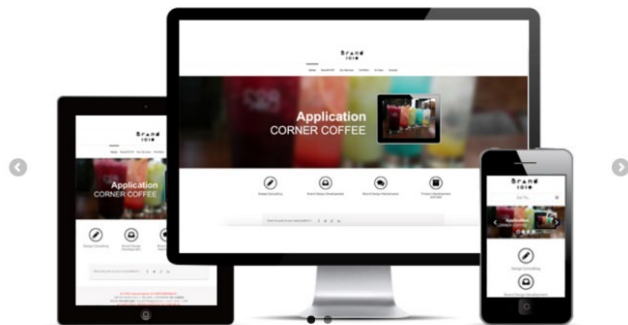


 **김쌤과 함께 쉽고 빠르게**

부트스트랩4를 활용한 반응형 웹 만들기



STEP 11

다양한 컴포넌트6_ 페이지네이션부터 프로그레스까지

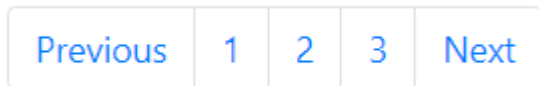
INDEX

페이지네이션

팝오버

프로그레스 바

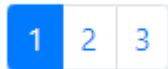
 .pagination / .page-item



```
<nav aria-label="Page navigation example">
  <ul class="pagination">
    <li class="page-item">
      <a class="page-link" href="#" aria-label="Previous">
        <span aria-hidden="true">&laquo;</span>
      </a>
    </li>
    <li class="page-item"><a class="page-link" href="#">1</a></li>
    <li class="page-item"><a class="page-link" href="#">2</a></li>
    <li class="page-item"><a class="page-link" href="#">3</a></li>
    <li class="page-item">
      <a class="page-link" href="#" aria-label="Next">
        <span aria-hidden="true">&raquo;</span>
      </a>
    </li>
  </ul>
</nav>
```

페이징은 목록 HTML 요소로 구축되므로 사용 가능한 링크 수를 알릴 수 있다. 태그에 .pagination을 주고 에 .page-item를 부여한다. 요소를 활성화하기 위해서는 .page-item에 .active를 비활성화하기 위해서는 .page-item에 .disabled를 부여한다.

.pagination .pagination-lg



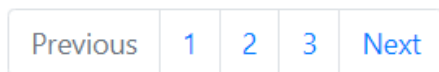
```
<nav aria-label="...">
  <ul class="pagination pagination-lg">
    <li class="page-item active" aria-current="page">
      <span class="page-link">1</span>
    </li>
    <li class="page-item"><a class="page-link" href="#">2</a></li>
    <li class="page-item"><a class="page-link" href="#">3</a></li>
  </ul>
</nav>
```

```
<nav aria-label="...">
  <ul class="pagination pagination-sm">
    <li class="page-item active" aria-current="page">
      <span class="page-link">1</span>
    </li>
    <li class="page-item"><a class="page-link" href="#">2</a></li>
    <li class="page-item"><a class="page-link" href="#">3</a></li>
  </ul>
</nav>
```

더 크거나 작은 페이징을 원하면 .pagination에 .pagination-lg 이나 .pagination-sm을 추가한다.

.pagination .pagination-lg

Alignment



```
<nav aria-label="Page navigation example">
  <ul class="pagination justify-content-center">
    <li class="page-item disabled">
      <a class="page-link">Previous</a>
    </li>
    <li class="page-item"><a class="page-link" href="#">1</a></li>
    <li class="page-item"><a class="page-link" href="#">2</a></li>
    <li class="page-item"><a class="page-link" href="#">3</a></li>
    <li class="page-item">
      <a class="page-link" href="#">Next</a>
    </li>
  </ul>
</nav>
```

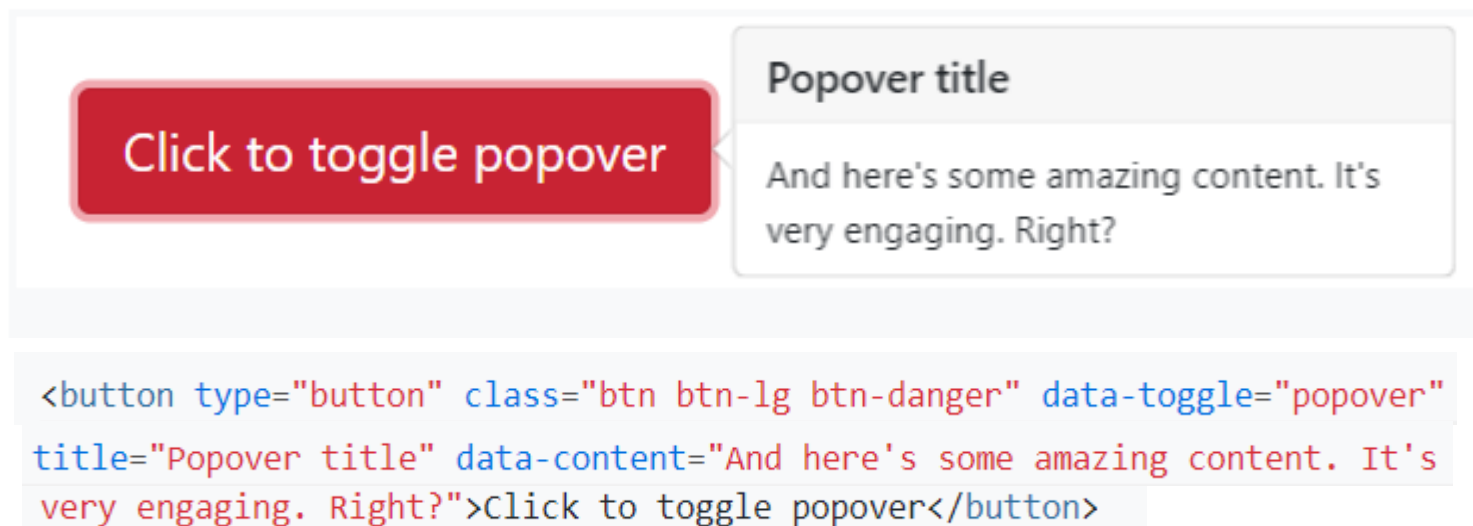


```
<nav aria-label="Page navigation example">
  <ul class="pagination justify-content-end">
    <li class="page-item disabled">
      <a class="page-link">Previous</a>
    </li>
    <li class="page-item"><a class="page-link" href="#">1</a></li>
    <li class="page-item"><a class="page-link" href="#">2</a></li>
    <li class="page-item"><a class="page-link" href="#">3</a></li>
    <li class="page-item">
      <a class="page-link" href="#">Next</a>
    </li>
  </ul>
</nav>
```

페이징 요소의 정렬을 위해 .pagination에 .justify-content-center 나 .justify-content-end를 추가한다.

[data-toggle="popover"], [title="팝오버의 제목"], [data-content="팝오버의 내용"]

Example



data-toggle="popover", title="팝오버의 제목", data-content="팝오버의 내용" 을 준 후 자바스크립트로 호출한다.
버튼을 토글시에만 팝오버가 사라진다.

```
popover({trigger:'focus'})
```

Dismiss on next click

Dismissible popover

Dismissible popover

And here's some amazing content. It's very engaging. Right?

```
<a tabindex="0" class="btn btn-lg btn-danger" role="button" data-toggle="popover" data-trigger="focus" title="Dismissible popover" data-content="And here's It's very engaging. Right?">Dismissible popover</a>
```

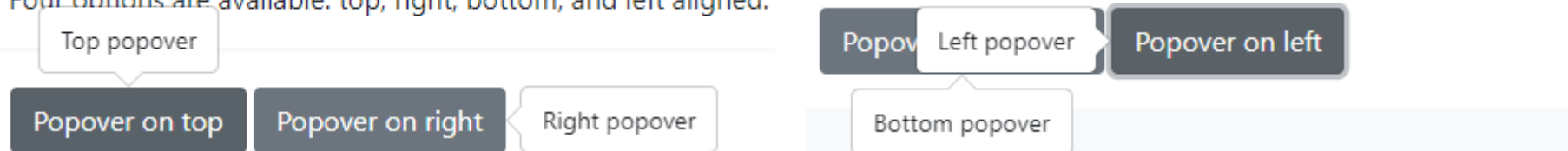
```
$('.popover-dismiss').popover({  
  trigger: 'focus'  
})
```

사용자가 토글 요소가 아닌 다른 요소를 클릭할 때 창을 닫으려면 트리거를 사용하여 팝오버를 닫는다.

[data-placement="top, bottom, left, right"]

Four directions

Four options are available: top, right, bottom, and left aligned.



```
<button type="button" class="btn btn-secondary" data-container="body" data-toggle="popover" data-placement="top" data-content="Top popover">
  Popover on top
</button>
<button type="button" class="btn btn-secondary" data-container="body" data-toggle="popover" data-placement="right" data-content="Right popover">
  Popover on right
</button>
<button type="button" class="btn btn-secondary" data-container="body" data-toggle="popover" data-placement="bottom" data-content="Bottom popover">
  Popover on bottom
</button>
<button type="button" class="btn btn-secondary" data-container="body" data-toggle="popover" data-placement="left" data-content="Left popover">
  Popover on left
</button>
```

팝오버의 위치를 top, right, bottom, left 정렬 4가지 방향으로 설정 가능하다. data-placement의 속성값을 원하는 방향으로 설정한다.

```
popover({delay:3000, trigger:'hover'})
```

trigger	string	'click'	How popover is triggered - click hover focus manual. You may pass multiple triggers; separate them with a space. manual cannot be combined with any other trigger.
content	string element function	"	Default content value if data-content attribute isn't present. If a function is given, it will be called with its this reference set to the element that the popover is attached to.
delay	number object	0	Delay showing and hiding the popover (ms) - does not apply to manual trigger type If a number is supplied, delay is applied to both hide/show Object structure is: delay : { "show": 500, "hide": 100 }

팝오버의 동작 방식을 데이터 속성이나 자바스크립트를 이용하여 다양하게 변경 가능하다.

.progress > .progress-bar.w-25배수



```
<div class="progress">
  <div class="progress-bar" role="progressbar" aria-valuenow="0" aria-valuemin=
</div>
<div class="progress">
  <div class="progress-bar" role="progressbar" style="width: 25%" aria-valuenow
</div>
<div class="progress">
  <div class="progress-bar" role="progressbar" style="width: 50%" aria-valuenow
</div>
<div class="progress">
  <div class="progress-bar" role="progressbar" style="width: 75%" aria-valuenow
</div>
<div class="progress">
  <div class="progress-bar" role="progressbar" style="width: 100%" aria-valuenow
</div>
```

.progress > .progress-bar 를 사용하고 진행 상황을 나타내기 위해 너비를 설정한다.

`.progress[height] > .progress-bar[aria-valuenow]`

Labels

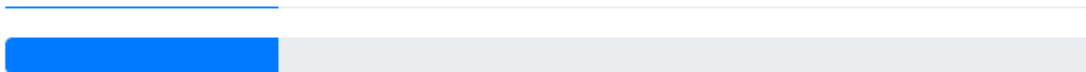
Add labels to your progress bars by placing text within the `.progress-bar`.

A horizontal progress bar with a blue filled section on the left representing 25% completion. The text "25%" is centered within the blue section.

```
<div class="progress">
  <div class="progress-bar" role="progressbar" style="width: 25%;" aria-valuenow="25">25%</div>
</div>
```

Height

We only set a `height` value on the `.progress`, so if you change that value the inner `.progress-bar` will automatically resize accordingly.



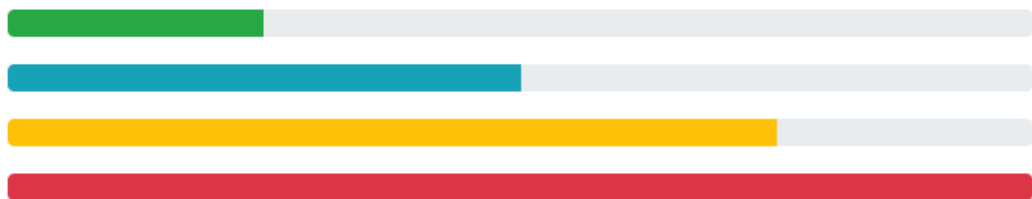
```
<div class="progress" style="height: 1px;">
  <div class="progress-bar" role="progressbar" style="width: 25%;" aria-valuenow="25">25%</div>
</div>
<div class="progress" style="height: 20px;">
  <div class="progress-bar" role="progressbar" style="width: 25%;" aria-valuenow="25">25%</div>
</div>
```

`.progress[height]` 를 설정하여 높이를 변경하고 `.progress-bar[aria-valuenow]` 를 설정하여 진행율을 변경한다.

.progress-bar.bg-테마명 / .progress-bar를 형제관계로 구성

Backgrounds

Use background utility classes to change the appearance of individual progress bars.



```
<div class="progress">
  <div class="progress-bar bg-success" role="progressbar" style="width: 25%" ar
</div>
<div class="progress">
  <div class="progress-bar bg-info" role="progressbar" style="width: 50%" aria-
</div>
<div class="progress">
  <div class="progress-bar bg-warning" role="progressbar" style="width: 75%" ar
</div>
<div class="progress">
  <div class="progress-bar bg-danger" role="progressbar" style="width: 100%" ar
</div>
```

Multiple bars

Include multiple progress bars in a progress component if you need.



```
<div class="progress">
  <div class="progress-bar" role="progressbar" style="width: 15%" aria-valu
  <div class="progress-bar bg-success" role="progressbar" style="width: 30%
  <div class="progress-bar bg-info" role="progressbar" style="width: 20%" a
</div>
```

.progress-bar.bg-테마명을 주면 진행율 스타일을 다양하게 주고 .progress-bar를 형제관계로 구성하면 다중막대로 구성된다.

.progress-bar .progress-bar-striped .progress-bar-animated

Striped

Add `.progress-bar-striped` to any `.progress-bar` to apply a stripe via CSS gradient over the progress bar's background color.



```
<div class="progress">
  <div class="progress-bar progress-bar-striped" role="progressbar" style="width: 25%;">
</div>
<div class="progress">
  <div class="progress-bar progress-bar-striped bg-success" role="progressbar" style="width: 50%;">
</div>
<div class="progress">
  <div class="progress-bar progress-bar-striped bg-info" role="progressbar" style="width: 75%;">
</div>
```

Animated stripes

The striped gradient can also be animated. Add `.progress-bar-animated` to `.progress-bar` to animate the stripes right to left via CSS3 animations.










Toggle animation

```
<div class="progress">
  <div class="progress-bar progress-bar-striped progress-bar-animated" role="progressbar" style="width: 100%;">
</div>
```

.progress-bar. 에 .progress-bar-striped를 주면 줄무늬 스트라이프 스타일을 주고 .progress-bar-animated 를 주면 애니메이션을 동작시킨다.

꼭 기억해야 할 포인트 정리

1. 팝오버는 `data-toggle="popover"`, `title="팝오버의 제목"`, `data-content="팝오버의 내용"`을 준 후 자바스크립트로  메소드를 호출한다
2. 더 크거나 작은 페이지를 원하면 `.pagination`에  이나  을 추가한다.
3. 사용자 정의 진행률 표시줄을 나타내기 위해  >  를 사용하고 진행 상황을 나타내기 위해 너비를 설정한다.
4. `.progress-bar` 에  를 주면 줄무늬 스트라이프 스타일을 주고  를 주면 애니메이션을 동작시킨다.



꼭 기억해야 할 포인트 정리

1. 팝오버는 `data-toggle="popover"`, `title="팝오버의 제목"`, `data-content="팝오버의 내용"`을 준 후 자바스크립트로 **Popover()** 메소드를 호출한다
2. 더 크거나 작은 페이지를 원하면 `.pagination`에 [] 이나 [] 을 추가한다.
3. 사용자 정의 진행률 표시줄을 나타내기 위해 [] > [] 를 사용하고 진행 상황을 나타내기 위해 너비를 설정한다.
4. `.progress-bar` 에 [] 를 주면 줄무늬 스트라이프 스타일을 주고 [] 를 주면 애니메이션을 동작시킨다.

꼭 기억해야 할 포인트 정리

1. 팝오버는 `data-toggle="popover", title="팝오버의 제목", data-content="팝오버의 내용"`을 준 후 자바스크립트로 **Popover()** 메소드를 호출한다
2. 더 크거나 작은 페이지징을 원하면 `.pagination`에 **.pagination-lg** 이나 **.pagination-sm** 을 추가한다.
3. 사용자 정의 진행률 표시줄을 나타내기 위해 **[블록]** > **[블록]** 를 사용하고 진행 상황을 나타내기 위해 너비를 설정한다.
4. `.progress-bar.` 에 **[블록]** 를 주면 줄무늬 스트라이프 스타일을 주고 **[블록]** 를 주면 애니메이션을 동작시킨다.

꼭 기억해야 할 포인트 정리

1. 팝오버는 `data-toggle="popover"`, `title="팝오버의 제목"`, `data-content="팝오버의 내용"`을 준 후 자바스크립트로 `Popover()` 메소드를 호출한다
2. 더 크거나 작은 페이지징을 원하면 `.pagination`에 `.pagination-lg` 이나 `.pagination-sm` 을 추가한다.
3. 사용자 정의 진행률 표시줄을 나타내기 위해 `.progress` > `.progress-bar` 를 사용하고 진행 상황을 나타내기 위해 너비를 설정한다.
4. `.progress-bar` 에 `.progress-bar-striped` 를 주면 줄무늬 스트라이프 스타일을 주고 `.progress-bar-animated` 를 주면 애니메이션을 동작시킨다.

꼭 기억해야 할 포인트 정리

1. 팝오버는 `data-toggle="popover"`, `title="팝오버의 제목"`, `data-content="팝오버의 내용"`을 준 후 자바스크립트로 `Popover()` 메소드를 호출한다
2. 더 크거나 작은 페이지징을 원하면 `.pagination`에 `.pagination-lg` 이나 `.pagination-sm` 을 추가한다.
3. 사용자 정의 진행률 표시줄을 나타내기 위해 `.progress` > `.progress-bar` 를 사용하고 진행 상황을 나타내기 위해 너비를 설정한다.
4. `.progress-bar` 에 `.progress-bar-striped` 를 주면 줄무늬 스트라이프 스타일을 주고 `.progress-bar-animated` 를 주면 애니메이션을 동작시킨다.

영상 코딩 디자인은 영코디 김쌤



영코디 김쌤

