부트스트랩 200608

1. 프레임워크와 라이브러리

- 라이브러리
  - 단순 활용 가능한 도구 모음
  - 연장 같이, 툴, 양치
- 프레임워크
  - 원하는 기능 구현에 집중
  - 라이브러리 포함
    - 클래스+라이브러리
    - 그홀단위로 묶은 듯한
  - 딱 것 같은. 이미 완성된. 자, 버렸기
    - 재공
      - 요소
      - 틀
      - 규약 — 지켜야함
  - bootStrap, Spring, Django... — Flask도 프레임워크임 — 라즈베리 때 사용했던
- 자유도 차이

2. 부트스트랩이란?

- 프레임 워크
  - 재사용이 가능한 요소들의 집합
  - HTML, CSS, Js
  - 반응형
  - 모바일 우선
- document cf — 필요할 때 찾아쓰면 됨
- 부트스트랩 사이트에서 다운

3. 부트스트랩 적용

- CDN
  - 링크 걸기
    - CSS — 헤드
    - Js 등 — 바디
  - CDN? cf
    - 컨텐츠를 더 빠르게 제공
    - 물리적으로 떨어진 사용자에게
- 파일
  - 파일 받아서
    - 링크
    - cf — min.css — 공백 압축시킨 버전
  - jquery는 따로 받아야함
- ex. example 참고 — documentation

4. 텍스트 서식, 정렬, float...

- CTRL + K + F — 자동정렬 — vscode — cf
- 정렬
  - text-left
  - text-right
  - text-center
  - text-justify
  - text-nowrap
- float
  - float-left
  - float-right
  - float-none
- 색
  - primary success... — bg- — 배경
  - primary success... — text- — 글자색

5. 반응형그리드

- .container
  - 여백을 줌 줌
  - 안에 다시 .container 클래스 사용 불가
- 여백이 별로 없음 — .container-fluid
- 화면크기가 줄어들면 돌다 여백이 줌
- 그리드 row
  - 기본 100% — 한줄
  - 기본 col 기준이 12 칸분
    - .col-12
    - 한줄 꽉채움
  - 벗어나면 밑줄로 감 — .col-6
  - ...
  - ex.
    - 중복적용 가능
    - 576px 이상일 시 분할 적용
      - .-sm-
      - 더 작아지면 한 줄씩 차지
      - -md-
      - -lg-
      - -xl-
    - 화면사이즈
      - 4/12로 — 화면이 992px 이상일시
      - 6/12으로 한줄씩 — 그보다 작으면 — 화면이 576px 이상일 시
      - 요연하게 반응형으로 조절
      - .-.
    - ex. .col-sm-6 — .col-.*-12

```
<div class="container">
    <div class="row">
        <div class="col-sm-6 col-lg-4 bg-info">.col-6</div>
        <div class="col-sm-6 col-lg-4 bg-warning">.col-6</div>
        <!-- 벗어나면 밑으로 감 -->
        <div class="col-sm-6 col-lg-4 bg-primary">.col-6</div>
    </div>
```

```
<div class="row">
    <div class="bg-info text-white text-center col-lg-6 col-md-2 col-sm-12">Header</div>
    <div class="bg-success text-white text-center col-lg-3 col-md-4 col-sm-12">Content</div>
    <div class="bg-danger text-white text-center col-lg-2 col-md-6 col-sm-8">Content2</div>
    <div class="bg-warning text-center col-lg-1 col-md-12 col-sm-4">Menu</div>
</div>
```

- ex. .col-sm-6 — .col-.*-12

6. 마진, 패딩, 플렉스

- spacing
  - 마진클래스
    - m-
      - 위 — mt-
      - 왼쪽 오른쪽
        - 왼 — ml-
        - 오 — mr-
      - x축 — mx-
      - 위, 아래
        - 위 — mt-
        - 아래 — mb-
      - y축 — my-
      - ex. mx-auto — auto 가능 — 중앙
    - 1rem value default Spacer variable
  - 패딩클래스
    - p-
    - 마진과 같음 — ...
- 크기
  - w-
  - h-
  - %, 25, 50, 75, 100
  - div 기준, 높이를 지정하려면 기본적으로 높이값을 갖고 있어야함
- d-flex
  - 플렉스 적용
  - justify-content
    - 가로 정렬
    - .justify-content-start
    - .justify-content-end
    - .justify-content-center
    - .justify-content-between
    - .justify-content-around
  - align-items
    - 수직정렬
    - 높이가 있어야 함
    - .align-items-start
    - .align-items-end
    - .align-items-center
    - .align-items-baseline
    - .align-items-stretch
  - .d-sm-flex — ex. — 그리드처럼 화면사이즈 별 변화 가능
- flex
  - 방향
    - 수직
      - to set a vertical direction — .flex-column
      - .flex-column-reverse
    - 수평
      - to set a horizontal direction — .flex-row
      - 반대방향에서 시작 — .flex-row-reverse
  - ex.

```
<div class="container">
    <div class="row">
        <!-- d-flex 플렉스 적용 -->
        <div class="col-p-3 bg-secondary text-white d-flex justify-content-center align-items-center">
            <div class="p-2 bg-info">Flex Item1</div>
            <div class="p-2 bg-warning">Flex Item2</div>
            <div class="p-2 bg-primary">Flex Item3</div>
        </div>
    </div>
```

- 공간을 주고 싶은 목으로 각각의 아이템의 마진을 auto로 주면 됩니다
  - wrap
    - 아이템들이 화면끝으로 몰림
      - flex-wrap
      - 줄바꿈
      - 줄바꿈x — -nowrap
    - 모든 조절이 화면 사이즈별 가능함

7. 테이블과 리스트

- 기본 테이블 — 테이블 cf
  - ex.

```
<div class="container">
    <h2>기본 테이블</h2>
    <h3><code>.table</code></h3>

    <table class="table">

      <thead>
        <tr>
          <th>Firstname</th>
          <th>Lastname</th>
          <th>Email</th>
        </tr>
      </thead>

      <tbody>
        <tr>
          <td>존</td>
          <td>도우</td>
          <td>jhon@example.com</td>
        </tr>
        <tr>
          <td>제리</td>
          <td>모우</td>
          <td>mary@example.com</td>
        </tr>
        <tr>
          <td>줄리</td>
          <td>윌리</td>
          <td>july@example.com</td>
        </tr>
      </tbody>

    </table>

</div>
```

  - 열 — tr
  - 일반적 행 — td
  - 제목 행 — th
- .table
  - 한행 건너 색 — .table-striped 추가
  - 호버시 색 변경 — .table-hover 추가
  - thead-light — theadl에 색
  - ...
- pagenation cf
  - https://getbootstrap.com/docs/4.3/components/pagination/
  - ex.

```
<nav aria-label="Page navigation example">
  <ul class="pagination">
    <li class="page-item"><a class="page-link" href="#">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>
```

- 리스트 그룹
  - .active — 활성화되도 그 녀온 a 가 색이 입혀짐
  - ex. list-group-item-primary — list-group-item-색 — 리스트에 색 넣기
  - ul에 .list-group
  - ul에 list-group-flush
    - 양 옆 트임
    - 플러시 리스트 그룹
  - li에 .list-group-item

```
<div class="container">
    <h3>기본 리스트 그룹</h3>
    <ul class="list-group mb-5">
      <li class="list-group-item">My List Item1</li>
      <li class="list-group-item">My List Item2</li>
      <li class="list-group-item">My List Item3</li>
      <li class="list-group-item">My List Item4</li>
      <li class="list-group-item">My List Item5</li>
    </ul>
</div>
```

```
<div class="container">
    <h1>li 대신에 a</h1>
    <ul class="list-group mb-5">
      <a href="#" class= "list-group-item">My List Item1</a>
      <a href="#" class= "list-group-item">My List Item2</a>
      <a href="#" class= "list-group-item">My List Item3</a>
      <a href="#" class= "list-group-item">My List Item4</a>
      <a href="#" class= "list-group-item">My List Item5</a>
    </ul>
</div>
```

  - li 대신 a 태그로도 가능
  - li를 만들지 않고 바로 리스트를 만들 수 있겠음
  - badge — cf
    - 뱃지
    - 30
  - new 알림 같음
- ex — with js

```
var li = document.querySelectorAll('.mylist li')
for(var i =0; i<li.length; i++)
    {
        li[i].addEventListener('mouseenter', function()
        {
            this.classList.add('active')
        })

        li[i].addEventListener('mouseout', function()
        {
            this.classList.remove('active')
        })
    }
    // 이벤트가 발생했을 때, 해당 태그가 this.
```

  - 마우스 올라갔을 때 active 클래스 추가
  - 마우스 빠져나갈 때 active 클래스 제거
  - querySelectorAll() doesn't return a HTMLCollection but a NodeList — https://stackoverflow.com/questions/50643302/addeventlistener-on-a-queryselectorall-with-classlist
  - addEventListener를 한건 c
  - addEventListner를 해주는 것과 이벤트가 발생하는 것은 별개임 — li[i].classList.add('active')는 만듦

8. 점보트론, 버튼, progress bar, 툴팁, 팝오버

- 점보트론
  - jumbotron — 전광판 — 꾸며진 박스가 안에 넣음
- 버튼
  - 버튼그룹
    - btn-group
    - btn-group-vertical
    - 버튼을 감싸고 있는 갖에 클래스를 줌
  - 버튼스타일
    - btn btn-primary... — 텍스트 등과 비슷함
    - -outline- ~
- 프로그래스 바
  - 상위
    - .progress
    - .progress-bar
  - 하위
    - 스타일
      - progress-bar-striped
      - progress-bar-animated
    - 하나의 바에 여러개도 가능
  - ex.

```
<!-- Striped -->
    <div class="progress">
      <div class="progress-bar bg-info progress-bar-striped" style="width:30%">.</div>
    </div><br>
    <!-- Animated -->
    <div class="progress">
      <div class="progress-bar progress-bar-striped progress-bar-animated" style="width:
40%">.</div>
    </div><br>
    <!-- Multi -->
    <div class="progress">
      <div class="progress-bar bg-success" style="width:40%">
        Free Space
      </div>
      <div class="progress-bar bg-warning" style="width:10%">
        Warning
      </div>
      <div class="progress-bar bg-danger" style="width:20%">
        Danger
      </div>
```

- 툴팁
  - 속성을 추가함
    - 위치 — data-placement="top"
    - 툴팁에 나오는 내용 — title="툴팁!"
    - 툴팁 설정 — data-toggle="tooltip"
  - 클래스x
  - ex.

```
<button class="btn btn-primary" data-toggle="tooltip"
data-placement="top" title= "툴팁!">툴팁 위쪽</button>
```

  - 자바스크립트 필요

```
<script>
    $(data-toggle='tooltip').tooltip();
    // 데이터 토글이 tooltip이면 tooltip()
실행 </script>
```

- 팝오버
  - 속성 추가
    - data-toggle="popover"
    - data-placement=
    - 내용 — data-content=
  - 클릭시 나옴
  - ${'data-toggle="popover"'}.popover(); — 자바스크립트 필요

200609 부트스트랩

1. 네비게이션바

- '상단 메뉴바는 컨테이너에 넣지 않는다'
- nav
  - .navbar
  - .navbar-expand-(사이즈명)
  - .navbar-dark — 글자색 흰
  - .navbar-nav
    - ul
      - .nav-item
        - li
          - .nav-link
            - a
              - .active — 액티브 링크 상태
              - .disabled — 링크 안되는 상태
- https://getbootstrap.com/docs/4.5/components/navbar/

```
<nav class="navbar navbar-expand-lg bg-success navbar-dark">
  <!-- 리스트 -->
  <ul class="navbar-nav">
    <li class="navbar-item">
      <a href="#" class="nav-link">메뉴 1</a>
    </li>
    <li class="navbar-item">
      <a href="#" class="nav-link">메뉴 1</a>
    </li>
    <li class="navbar-item">
      <a href="#" class="nav-link">메뉴 1</a>
    </li>
  </ul>
</nav>
```

- 토글러 버튼
  - 화면 사이즈에 반응

```
<button class="navbar-toggler" data-toggle="collapse" data-target="#toggle">
  <span class="navbar-toggler-icon"></span>
</button>
```

  - ID가 toggle인 것을 안 보이게 할 것임
  - .navbar-toggler-icon — 기본 제공함
  - .navbar-toggler
    - 속성
      - data-toggle
      - data-target
        - ex. data-target="#toggle"
        - `<div class="collapse navbar-collapse" id="toggle">` … `</div>` — 토글에 반응할 부분을 지정해줌
- 버튼드롭
  - 버튼만들기
    - 버튼을 눌렀을 때 나올 메뉴 — 메뉴 아이템

```
<div class="container">
<code>버튼 드롭</code>
<div class="dropdown">
  <button
    type="button"
    class="btn btn-primary dropdown-toggle"
    data-toggle="dropdown".
  >
  드롭다운 버튼
  </button>
  <!-- 버튼을 눌렀을 때 나올 메뉴 -->
  <div class="dropdown-menu">
    <a href="https://www.naver.com" class="dropdown-item" target="_blank">네이버</a>
    <a href="https://www.google.com" class="dropdown-item">구글</a>
    <a href="https://www.daum.net" class="dropdown-item">다음</a>
  </div>
</div>
</div>
```

    - ex
  - `<div class="dropdown-header">`링크모음`</div>`
- 서치폼

```
<form action="" class="form-inline mr-auto">
  <input class="form-control mr-2" type="text" placeholder="Search" />
  <button class="btn btn-info" type="submit">Search</button>
</form>
```

  - input — form-control — input 텍스트창 보기좋게
  - form — .form-inline — inline 요소로 바꿈 — nav 안에 얼럴 배열

예제1

- 오류
  - row에 div들이 col을 차지하지 않고, 하나의 col에 모든 div가 들어가는 것을 확인
    - div나 인라인이 아니어서 그런가? 싶었지만 — bootStrap에서 row와 col은, 따로 div를 안라인 화할 필요 없이 자절로 한 행에 잘 들어갔었음
  - 원인은 단순했음
    - 첫번째 div를 안 닫아 준 것
    - 그러니 결과적으로 모든 것을 하나의 div로 여겨짐 — 하나의 열에 모든 div가 들어간 것
  - 태그의 시작과 끝을 신경쓸 것 — 기본임

```
<div class="container">
  <div class="row">
    <div class="col-md-4">
      <h2>Heading</h2>
      <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Modi, rem.</p>
      <p><a href="#" class="btn btn-primary btn-lg" role="button">
          View details» </a></p>
    </div>
    <div class="col-md-4">
      <h2>Heading</h2>
      <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Modi, rem.</p>
      <p><a href="#" class="btn btn-primary btn-lg" role="button">
          View details» </a></p>
    </div>
    <div class="col-md-4">
      <h2>Heading</h2>
      <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Modi, rem.</p>
      <p><a href="#" class="btn btn-primary btn-lg" role="button">
          View details» </a></p>
    </div>
  </div>
</div>
```

- 폰트오썸 사용할 때 — CDN 링크 걸어주기

```
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.13.0/css/all.min.css"
  integrity="sha256-h20CPZ0QyXlBuAw7A+KluUYx/3pK+c7lYEpqLTIxjYQ=" crossorigin="anonymous"/>
```

2. 카드

- https://getbootstrap.com/docs/4.5/components/card/
- 블록속성

```
<div class="card m-3">
  <div class="card-header">카드 헤더</div>
  <div class="card-body">카드 바디</div>
  <div class="card-footer">카드 푸터</div>
</div>
```

```
<div class="card m-3">
  <img class="card-img-top" src="https://source.unsplash.com/300x200/?nature,water" alt="물">
  <div class="card-body">
    <h4 class="card-title">Card Title</h4>
    <h6 class="card-subtitle text-muted">Card subtitle</h6>
    <p class="card-text">Lorem ipsum dolor sit amet, consectetur adipisicing elit. Illum, enim?</p>
    <a href="#" class="btn btn-outline-primary">Read more</a>
  </div>
</div>
```

  - .card-title
  - .card-subtitle
  - .card-text
  - 이미지 넣기
    - 카드바디보다 이미지가 위 — card-img-top
    - 이미지가 아래 — card-img-bottom
- card-img-overlay
  - 사진 위에 카드
  - 이미지는 따로 넣고
  - 카드 부분에 해당 클래스를 부여함
- 카드 레이아웃
  - 카드잡힘
    - 상위 div에 — .card-colums
      - 하위에 카드 넣음
    - 기본 3열 배치

```
Heads up! Your mileage with card columns may vary. To prevent cards breaking across columns, we must set them to display: inline-block as column-break-inside: avoid isn't a bulletproof solution yet.
```

  - Card groups
  - Card decks
    - Need a set of equal width and height cards
    - aren't attached to one another

3. Carousel

- https://getbootstrap.com/docs/4.5/components/carousel/

```
<!-- 슬라이드 only -->
<div id="carousel1" class="carousel slide" data-ride="carousel">
  <!-- 사진 슬라이드 -->
  <div class="carousel-inner">
    <div class="carousel-item active">
      <img src="https://source.unsplash.com/600x400/?animal,dog" alt="개" class="d-block w-100">
    </div>
    <div class="carousel-item">
      <img src="https://source.unsplash.com/600x400/?animal,cat" alt="고양" class="d-block w-100" alt="...">
    </div>
    <div class="carousel-item">
      <img src="https://source.unsplash.com/600x400/?animal,rabbit" alt="토끼" class="d-block w-100" alt="...">
    </div>
  </div>
</div>
```

- 디스플레이를 block으로 한 것 — d-block
- width는 보이는 정도로 크기 조절할 때 vw임 — cf

```
<!-- with indicator -->
<div id="carousel3" class="carousel slide" data-ride="carousel">
  <!-- 사진 슬라이드 -->
  <ol class="carousel-indicators">
    <li data-target="#carousel3" data-slide-to="0"></li>
    <li data-target="#carousel3" data-slide-to="1" class="active"></li>
    <li data-target="#carousel3" data-slide-to="2"></li>
  </ol>
  <div class="carousel-inner">
    <div class="carousel-item">
      <img src="https://source.unsplash.com/1600x900/?animal,dog" alt="개" class="caro">
    </div>
    <div class="carousel-item active">
      <img src="https://source.unsplash.com/1600x900/?animal,cat" alt="고양" class="caro" alt="...">
    </div>
    <div class="carousel-item">
      <img src="https://source.unsplash.com/1600x900/?animal,rabbit" alt="토끼" class="caro" alt="...">
    </div>
  </div>
  <!--컨트롤 화살표-->
  <a class="carousel-control-prev" href="#carousel3" role="button" data-slide="prev">
    <span class="carousel-control-prev-icon" aria-hidden="true"></span>
    <span class="sr-only">Previous</span>
  </a>
  <a class="carousel-control-next" href="#carousel3" role="button" data-slide="next">
    <span class="carousel-control-next-icon" aria-hidden="true"></span>
    <span class="sr-only">Next</span>
  </a>
</div>
```

- caption
  - 슬라이드 설명
  - carousel-item 당 하나씩 하위에 배치

```
<div class="carousel-caption d-none d-md-block">
  <h5>고양고양 label</h5>
  <p>Nulla vitae elit libero, a pharetra augue mollis interdum.</p>
</div>
```

  - 그 미만에선 안 보임 — d-none
  - md사이즈 이상에선 보이고 — d-md-block
- fade 전환 — .carousel-fade 추가시

```
<script>
  $(' .carousel').carousel({
    interval: 1000,
    pause: 'hover',
    wrap: true
  })
</script>
```

- 슬라이드 시간 조절

4. Modal

- https://getbootstrap.com/docs/4.5/components/modal/
- 열기
  - 버튼

```
<button class="btn btn-primary" data-toggle="modal" data-target="#myModal">
  Modal버튼
</button>
```

    - data-target이 열려게 될 대상
  - 링크

```
<a href="#myModal" class="btn btn-primary" data-toggle="modal">Modal링크</a>
```

  - JS

```
<button id="myBtn1" class="btn btn-danger">JS버튼</button>
<script>
  $("#myBtn1").click(function(){
    $("#myModal").modal("show");
  });
</script>
```

- 예제

```
<div class="modal" id="myModal">
  <div class="modal-dialog">
    <div class="modal-content">

      <!-- modal header -->
      <div class="modal-header">
        <h4 class="modal-title">Modal 제목</h4>
        <button class="close" data-dismiss="modal">&times;</button>
      </div>

      <!-- modal body -->
      <div class="modal-body">
        내용내용내용
      </div>
      <div class="modal-footer">
        <button class="btn btn-danger" data-dismiss="modal">Close</button>
      </div>
    </div>
  </div>
</div>
```

- 형태
  - .modal — 상위
  - .modal-dialog
    - modal 사이즈 (.modal-dialog 클래스 옆에 추가)
    - modal-dialog-centered
  - .modal-content
  - .modal-header
    - 제목 — .modal.title
    - × 는 x

```
<div class="modal" id="myModal">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <h4 class="modal-title">Modal 그리드</h4>
        <button class="close" data-dismiss="modal">&times;</button>
      </div>

      <div class="modal-body">
        <!-- form을 row 위에 위치시키봄 -->
        <form>
          <div class="row">

            <!-- 이런 식으로 form을 row 아래에 배치하면, row를 하위 div가 아닌 form 양어리가 먹음 -->
            <!-- <form>  -->
            <div class="col-6">
              <div class="form-group">
                <label for="">한글이름</label>
                <input type="text" class="form-control">
              </div>
            </div>

            <div class="col-6">
              <div class="form-group">
                <label for="">영문이름</label>
                <input type="text" class="form-control">
              </div>
            </div>

          <!-- </div> 유무임 div를 잘못줘서, row가 footer까지 들어가고 있음 -->

            <div class="col-12">
              <div class="form-group">
                <label for="">Email</label>
                <input type="text" class="form-control">
              </div>
            </div>

            <div class="col-12">
              <div class="form-group">
                <label for="">Comments</label>
                <input type="textarea" class="form-control">
              </div>
            </div>
          </form>

        </div>
        <!-- body div를 footer까지 감싸는 형태로 만들 -->


        <div class="modal-footer">
          <button class="btn btn-first" data-dismiss="modal">Close</button>
          <button class="btn btn-primary" data-dismiss="modal">Send Email</button>
        </div>

      </div>
    </div>
  </div>
```

- 요는 태그의 범위를 주의할 것.
  - html에서 자주 관련 실수를 하고 있음
  - 왜?
    - 부주의한 복사, 붙여넣기
    - 무계획한 요소의 배치
  - 어떻게 조치를 취할 수 있을까?
    - 태그범위 확인하기
      - 그.외에도 중간중간
    - 복사 붙여넣기 시 꼭 체크할것
    - 무작정 코드부터 짜기보다, 상하위관계를 생각해봐못하기
- 이 경우, 크게 두가지 div 실수가 있었음
  - 1. div를 잘못된 위치에서 닫아서, row를 footer까지 다 뒤집어씌우게 만듦
    - d-flex도 워고 안 먹음
    - 현재 footer역시도 row의 영향-> 버튼 위치가 왼쪽에 고정됨
      - 활용해보자면 — modal-footer에 w-100을 주면 오른쪽으로 가긴 함
        - 다소 조악하나 — col-x 값을 안 주고, 크기와 마진을 준다면 위치를 다소 다른게 지정할 수도 있음
        - 굳이 사용할 필요 없을 수도 있고
    - modal-footer와 modal-header에선 정렬값이 modal-body와 다름
    - row의 영향을 받는 col 내부에 다시 row 사용 가능함 — cf
      - 이경우 modal-footer클래스는 row 영향 아래에 있음
      - 내부에서 다시 row를 주고, w-100(width 100%)를 줘봤
      - 안의 col-4 내용물들이 3등분을 해서 공간을 채우게 됨
    - 현재 굳이 그렇게 할 것 없이, 이 경우 modal-footer)에 w-100을 주면 독립적 적용여건 됨
      - *justify-content:flex-end 니까*
  - 2. div를 잘못된 위치에서 닫아서, modal-body div가 footer까지 다 감싸고 있었음
    - 어쩌든 의도하지 않은 실수임
    - css를 확인하면
      - justify-content: space-between; — modal-header의 경우
      - 임 — justify-content: flex-end; — modal-footer의 경우
      - 지금의 경우, 결과상으론 문제가 없었을 것임. 내부의 버튼들은 modal-footer의 영향아래에 있기 때문
- 그리고 bootStrap에서 <form>의 경우
  - 각각의 label과 input들을 <form>내부에 한덩어리씩 묶어서 관리를 하는 것 같았음

```
<form>
  (class form-gorup)(/)
  (class form-gorup)(/)
  ...
</form>
```

  - 안 해도 적용은 되는데, 관리의 용이함을 위해서인가?
  - 해당 예제의 경우
    - 각 form-group 덩어리들을 묶은 form을 row 아래에 두면, 그 form이 해당 영향을 받게됨
    - 아래의 내용들이 직접x
      - col- 값을 준 div보다 하위에 위치해야함
      - row의 바로 하위에 오면x
      - form-group class의 경우도.

```
<div class="col-6 d-flex">
  <div class="form-group">
    <label for="">한글이름</label>
    <input type="text" class="form-control">
  </div>
  <div class="form-group">
    <label for="">한글이름</label>
    <input type="text" class="form-control">
  </div>
  <div class="form-group">
    <label for="">한글이름</label>
    <input type="text" class="form-control">
  </div>
</div>
```

  - row 값을 받는 하위 항목에 d-flex를 따로 줄 수도 있음 — cf

넘어감

- display와 visibility
  - collapse

XMind 평가판 모드