

퍼블리싱 참고사항 — 단계 표시기 / 파일첨부 폼

2 차 · 2026-07-29 · 대상 : ppt-47.html, ppt-48.html / 관련 : js/utill.js, css/util.css

1. 단계(Step) 표시기 — ppt-47.html

```
<div class="wrap-step">
  <ul class="step-list">
    <li class="is-on">
      <span>1 단계</span>
      <strong>약관동의</strong>
    </li>
    <li><span>2 단계</span><strong>신청정보입력</strong></li>
    <li><span>3 단계</span><strong>신청자료 선택</strong></li>
    <li><span>4 단계</span><strong>전자서명 및 제출</strong></li>
  </ul>

  <!-- step : mode mobile -->
  <div class="mode-mobile"></div>
</div>
```

상태 클래스 : is-on / is-complete

클래스	상태	설명
is-complete	완료	현재 단계 이전의 단계. 원 안에 체크(✓), 원·연결선이 진회색으로 채워짐
is-on	현재	현재 보고 있는 단계. 파란 원으로 강조. 화면당 반드시 1 개
(없음)	미진행	이후 단계. 연회색 원 + 연회색 연결선

- **li** 1 개 = 단계 1 개이며, li 개수가 곧 전체 단계 수입입니다. 총 단계 수는 별도로 적지 않아도 자동 계산됩니다.
- **span**(단계 번호) / **strong**(단계명) 태그와 순서는 유지해야 합니다. 모바일 영역에서 이 값을 그대로 가져다 씁니다.
- **단계 이동 시 변경 대상은 li 의 class 뿐입니다.** 예) ppt-48.html(2 단계) = 1 단계 **is-complete** + 2 단계 **is-on**, 3·4 단계 클래스 없음.
- 스크립트가 `$('.step-list')` 를 전역으로 선택하므로 **한 화면에 step 표시기는 1 개만** 배치합니다.

2. .mode-mobile 필수 처리

div.mode-mobile 은 반드시 포함하되, 내부는 비워 둡니다. 모바일에서 노출되는 단계 정보는 `js/utill.js` 가 자동으로 생성해 주입합니다.

- 동작 : **li.is-on** 의 **span·strong** 텍스트와 전체 li 개수를 읽어 **.mode-mobile** 내부 HTML 을 통째로 새로 씁니다.
- 실행 시점 : 페이지 최초 로드 / window resize / step-list 의 li pointerdown
- 반응형 : 767px 이하에서 **.step-list** 의 텍스트가 숨겨지고 **.mode-mobile** 이 노출됩니다. (PC 에서는 반대)

[스크립트가 자동 생성한 결과 — ppt-48.html 기준]

```
<div class="mode-mobile">
  <span><b>2 단계</b> / 4 단계</span>
  <strong>신청정보입력</strong>
</div>
```

주의

누락 시 모바일에서 원형 인디케이터만 보이고 현재 단계 정보가 전혀 노출되지 않습니다.
내부에 직접 마크업을 작성해도 스크립트가 덮어쓰므로 빈 div 상태로 두시면 됩니다.

3. 파일첨부 폼 — ppt-48.html

"연구계획서 등 첨부" 영역입니다. 아래 마크업까지만 작성하며, 파일을 등록하면 등록 파일 목록과 등록 개수·전체삭제 영역이 스크립트로 자동 생성·갱신됩니다.

```
<div class="wrap-upload-box" data-upload-form>
  <p>첨부할 파일을 여기에 끌어다 놓거나, 파일 선택 버튼을 직접 선택해주세요.</p>
  <label for="fileInput" class="button-fill01">파일선택</label>
  <input type="file" id="fileInput" class="file-input" multiple>
</div>

<div class="file-list">
  <div class="file-list-info">
    <div class="list-count">등록파일 : <b></b>개</div>
    <div class="list-module">
      <button type="button" class="button-size01-frame01 ico-delete01-after">
        <i>전체삭제</i>
      </button>
    </div>
  </div>
</div>
```

작성 시 필수 조건

- **data-upload-form** 속성이 스크립트 바인딩 기준입니다. 없으면 동작하지 않습니다.
- **.file-list** 는 **.wrap-upload-box** 의 바로 다음 형제 요소 여야 합니다. (`$dropArea.next('.file-list')`)
- **input.file-input** 은 숨김 처리되어 있고 **label[for]** 이 버튼 역할을 합니다. id ↔ for 값 일치 필요.
- **.list-count** 의 **** 는 비워 둡니다. 초기 로드 시 0 으로 채워집니다.
- **ul.down-item** 은 작성하지 않습니다. 첫 파일 등록 시 자동 생성되고, 0 개가 되면 제거됩니다.

자동 처리 내용

대상	동작
.dragover	드래그가 업로드 박스 위에 올라오면 부여 → 배경색 변경
ul.down-item	파일 1 개당 li 1 개 생성. 파일명 뒤에 [확장자, 용량] 자동 표기 (B/KB/MB/GB 환산)
.has-file	등록 파일 1 개 이상일 때 .file-list 에 부여 → 숨겨져 있던 정보영역 노출
.list-count b	등록·삭제 시마다 현재 개수로 갱신
.ico-del / .list-module button	개별 삭제 / 전체 삭제 처리

[파일 등록 후 자동 생성된 결과]

```
<div class="file-list has-file">
  <div class="file-list-info"> ... </div>

  <ul class="down-item">
    <li>
      <p>연구계획서.hwp [hwp, 128.4KB]</p>
      <a href="javascript:void(0)" class="ico-down"><i>다운로드</i></a>
      <a href="javascript:void(0)" class="ico-view"><i>바로보기</i></a>
      <a href="javascript:void(0)" class="ico-del"><i>삭제</i></a>
    </li>
  </ul>
</div>
```

개발 연동 시 확인사항

확인 필요

- ① input 의 id 가 "fileInput" 으로 고정되어 있습니다. 한 화면에 업로드 품이 2 개 이상이면 id / for 값을 고유하게 변경해 주세요.
- ② 파일 선택 직후 스크립트가 input 의 value 를 초기화합니다(중복 선택 허용 목적). form submit 만으로는 파일이 전송되지 않으므로 실제 업로드는 FormData 등으로 별도 구현이 필요합니다. 드래그&드롭 파일도 동일합니다.
- ③ 중복 파일 검사, 확장자·용량 제한, 삭제 확인창은 포함되어 있지 않습니다.
- ④ 다운로드(.ico-down) / 바로보기(.ico-view) 의 href 는 javascript:void(0) 상태이므로 실제 경로 연결이 필요합니다.

기등록 파일을 서버에서 출력하는 경우 — `ul.down-item` 과 하위 `li` 만 출력하면 됩니다. 초기 로드 시 스크립트가 개수를 다시 세어 `has-file` 와 등록 개수를 자동으로 맞춥니다.