

front-end rules (publisher style guide)

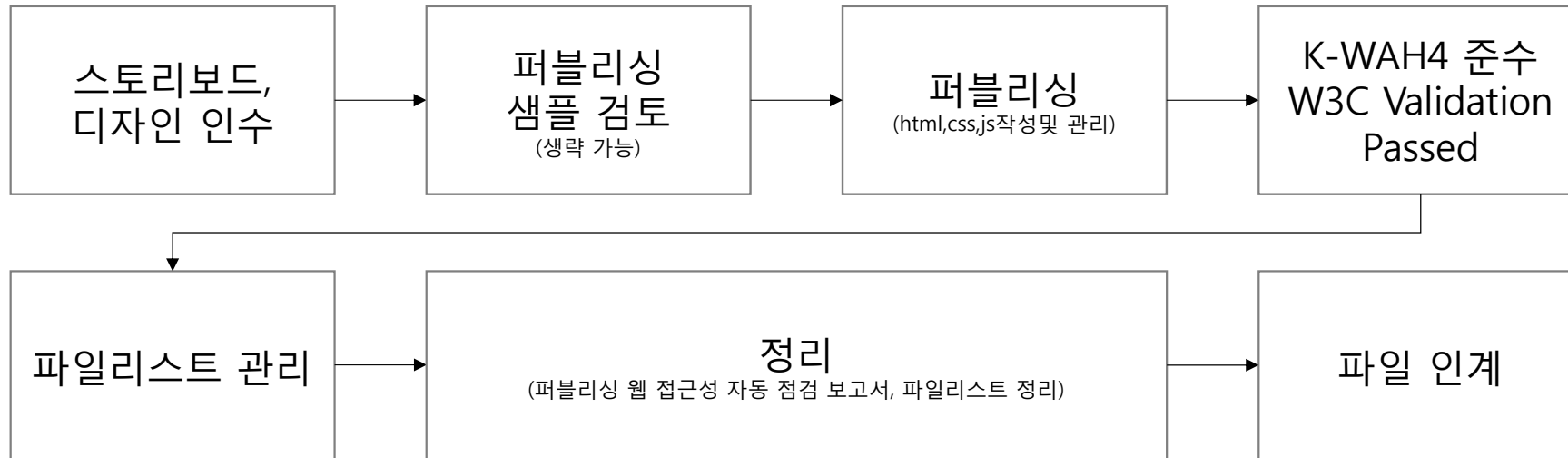
Written by naiyumie
ver 1.0.0

개 정 이 력

No	버전	작성자	작성일자	변경사유	변경내용	승인자
1	1.0	원상필	2015.01.05	최초작성		
1	1.1	원상필	2015.2.28	추가	assets관련 추가.	

0 워킹 프로세스

- 퍼블리싱 워킹 프로세스에 대해 설명 한다.



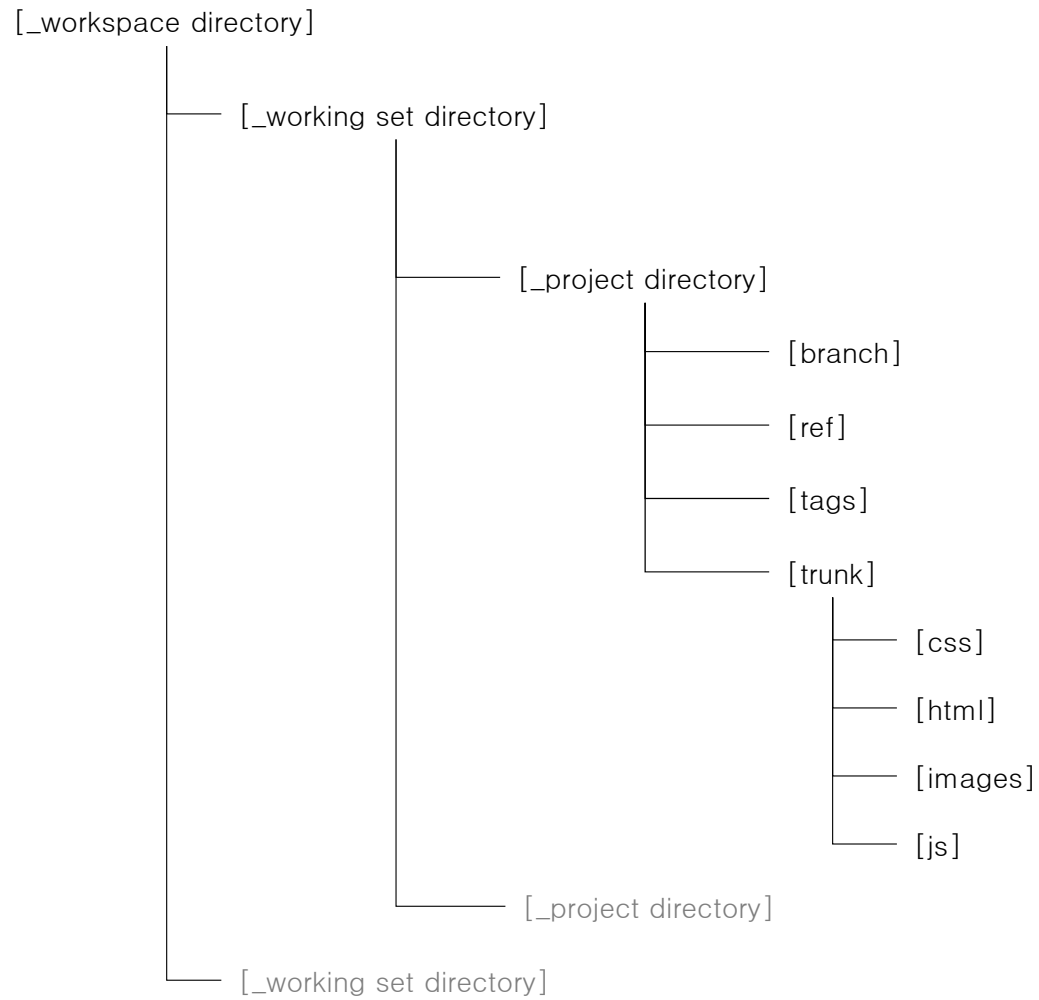
1. 스토리 보드와 디자인 파일을 인수 후 퍼블리싱 적합성을 검토 하고 퍼블리싱 작업을 시작한다.
2. 경우에 따라 본사 측에서 퍼블리싱 샘플의 품질을 검토 하여, 1~2page 정도를 수정 할 수 있다.
3. 퍼블리싱 하며 파일리스트를 관리 한다.
4. 완성된 결과물은 스토리보드 상의 기능과 디자인 적인 요소의 수치화를 만족 하여야 하며, K-WAH4와 W3C Validation에 Passed하여야 한다.
5. 위의 작업을 마쳤을 경우 동봉된 파일리스트에 체크 한뒤
6. 정리 한다.
 - 6.1 작성한 소스 외의 관련 파일을 ref에 넣는다. 웹접근성 자동점검 통과 문서를 동봉 필수.
 - 6.2 tags에 현 작성 완료본을 압축하여 넣는다.
 - 6.3 trunk를 유지한다.
 - 6.4 사용한 라이브러리의 원본을 branch에 넣는다.
7. 파일을 인계 한다.

1. 디렉터리

- 디렉터리의 구조에 대해 설명 한다.

1.1.1 SVN을 기초한 디렉터리 구조#1

- 워크스페이스를 기초하여 워킹셋으로 구분하고, 각각의 프로젝트 디렉터리가 있으며, 각 프로젝트 디렉터리는 SVN을 기초한 디렉터리구조를 형상화 한다.



1.1.2 SVN을 기초한 디렉터리 구조#2

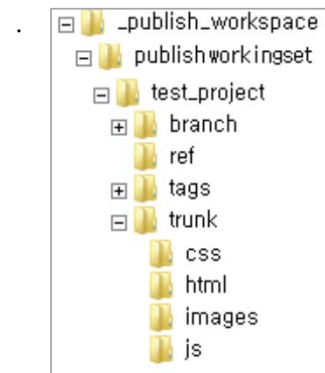


figure: 디렉터리 구조 예시

1.2 {projectDir}/ 디렉터리

- 프로젝트 루트 디렉터리는 기본적으로 **branch, ref, tags, trunk**의 4개의 디렉터리만 포함한다.

단, 개인 작업 스킬 역량에 따라 다양한 툴을 포함 할 수 있다.

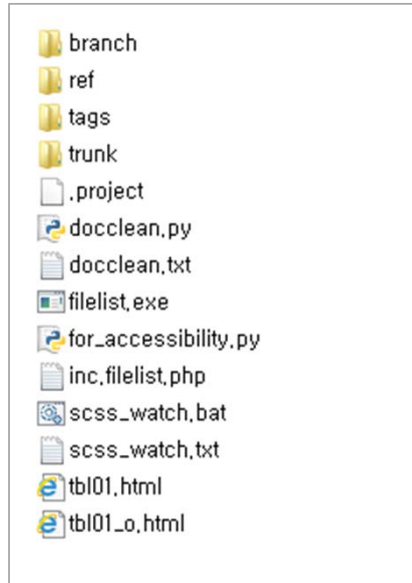


figure: 루트 디렉터리에 branch, ref, tags, trunk 디렉터리를 포함하고 이클립스 연동시 필요한 .project 파일 docclean.py filelist.exe 등 다양한 툴을 포함하여 위치한 모습.

1.3.1 {projectDir}/branch/ 디렉터리

- branch 디렉터리는 나무가지로서 나무줄기에서 뻗어나온 가지이다. 프로젝트 진행시 작은 프로젝트로 분류해서 개발하거나 따로 개발해야되는 경우에 사용한다. 프로젝트 안에 프로젝트 형태이다. 새로운 디렉터리와 소스코드가 들어간다.

이곳에 **별도의 테스트사항이나 사용한 원본 라이브러리를 넣는다.**

예를들어 "슬라이드 비주얼"을 구현하기 위하여 웹에서 jquery로 이루어진 적당한 라이브러리를 찾아서 사용 하였다면, 이곳에 별도의 라이브러리의 압축본 혹은 URL을 기록해두어야 한다.

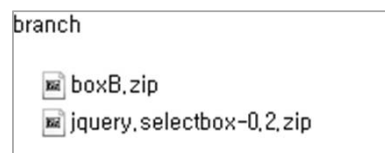


figure: branch디렉터리의 예시

1.3.2 {projectDir}/ref/ 디렉터리

- ref 디렉터리는 참조 문서가 위치한 곳으로써, 프로젝트 진행시 참조하여야 할 필수 문서 기타 부속 파일들을 위치한다.

참조 문서나 부속 파일을 넣는다.

GNB을 구현하기 위하여 이미지를 배열한 PSD를 남기거나, CI(로고)의 위치를 조절한 PSD를 갖고 있다면 이곳에 넣는다.



figure: ref의 예시 ci_guide.psd

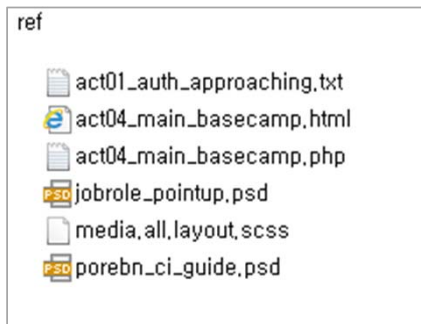


figure: ref디렉터리의 예시 사용한 파일들과 지금은 사용하지 않는 소스를 포함한 파일을 보관 하고 있다.

1.3.3 {projectDir}/tags/ 디렉터리

- 꼬리표라는 의미이다. 정기적으로 릴리즈되는 버전을 버전별로 디렉토리(꼬리표 형태)를 만들어 저장한다. 특히 tags 디렉토리 안에 버전명이 디렉토리 명으로 사용되어진다.

이곳에 타인에게 배포 하거나, 작업이 일시 중단 될 경우 압축하여 보관 한다.



figure: tags디렉터리의 예시 배포한 일자별로 압축하여 보관 하고 있다.

1.3.4 {projectDir}/trunk/ 디렉터리

- **주 작업 디렉터리이다.** 나무줄기, 몸통이라는 의미로 프로젝트에서 가장 중심이 되는 줄기이다. 모든 개발관련 작업은 이곳에서 수행된다. 이 줄기 아래에 프로젝트 디렉토리 구조 및 사용라이브러리, 코드가 들어간다.

파일의 루트에는 파일리스트만 위치하여야 하며, 파일리스트의 이름은 `filelist.html` 혹은 `listup.html` 혹은 `guide.html`과 같이 직관적이어야 하고, **파일 프로토콜을 사용하여 파일리스트를 사용하여 링크에 접근시 링크가 모두 유지되는 형태를 취해야 하며, 모든 클라이언트 기능이 동작 하여야 한다.**

예를들어 웹브라우저에 드래그해서 놓았을 경우 사용하였을 경우 브라우저(클라이언트)단에서 동작이 가능 하여야 한다.

`file:///D:/_workspace/_publish_workspace/publish5/porebnTest01/trunk/filelist.html`에 놓아서 링크를 클릭시

`file:///D:/_workspace/_publish_workspace/publish5/porebnTest01/trunk/html/act01_auth_approaching.html`와 연결 되는 형태의 상대경로를 취해야 한다.

클라이언트 (js 자바스크립트 언어와, css 스타일 시트, html마크업 언어)에 동작하는 모든 언어가 이에 해당된다.

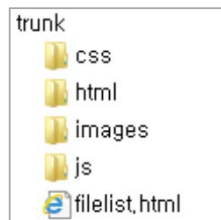



















figure: trunk디렉터리의 예시 이외에 다른 파일이 위치해선 안된다.

음악, 동영상, 웹 폰트, pdf 등의 웹에서 표현 할 수 있는 바이너리 파일은 assets에 넣는다. 또한 이곳에 파일 링크를 할 수 있다.

1.3.5 {projectDir}/trunk/html/ 디렉터리

- 모든 **html**파일을 넣는다.
별도로 정의한 접두어(prefix)를 통해 파일이 순차적으로 정렬 되어야 한다. natural ordering은 고려 하지않고 컴퓨터 기준으로 한다.
파일명은 영문 소문자와 숫자로 구성 되어야 하며 dash(-)와 under-score(_)를 포함 할 수 있다.
리눅스 시스템을 대응하여 camelCase를 허용 하지 않고, 영어를 제외한 한글/한자/일본어 등의 파일명은 허용될수 없다.



















html

 act01_auth_approaching.html	2014-12-31 오후 4:40	HTML 문서	4KB
 act01_auth_signin.html	2014-12-31 오후 4:52	HTML 문서	5KB
 act02_main_front0.html	2014-12-30 오후 2:12	HTML 문서	27KB
 act02_main_front1.html	2014-12-24 오전 1:...	HTML 문서	9KB
 act02_main_front2.html	2014-12-30 오후 2:12	HTML 문서	15KB
 act02_main_front3.html	2014-12-31 오후 1:29	HTML 문서	28KB
 act02_main_front4.html	2014-12-30 오후 5:24	HTML 문서	18KB
 act03_pop_01.html	2014-12-30 오후 5:24	HTML 문서	11KB
 act03_pop_02.html	2014-12-30 오후 2:43	HTML 문서	10KB
 act03_sub_01.html	2014-12-30 오후 5:36	HTML 문서	21KB
 act03_sub_02.html	2014-12-30 오후 2:43	HTML 문서	18KB
 act03_sub_03.html	2014-12-30 오후 2:12	HTML 문서	20KB
 act03_sub_04.html	2014-12-30 오후 5:24	HTML 문서	23KB
 act04_sub_01.html	2014-12-30 오후 5:09	HTML 문서	62KB
 act04_sub_02.html	2014-12-30 오후 5:09	HTML 문서	28KB
 act04_sub_03.html	2014-12-31 오후 1:56	HTML 문서	17KB
 act04_sub_04.html	2014-12-31 오후 1:32	HTML 문서	57KB

1.3.6 {projectDir}/trunk/css/ 디렉터리

- scss사용을 권장하나 본인의 스킴셋에 따라 자유롭게 css를 구성한다.






















CSS

 include.css	2014-12-30 오후 5:36	CSS 스타일시트 ...	185KB
 media.all.common.css	2014-12-04 오후 1:05	CSS 스타일시트 ...	4KB
 media.all.css	2014-11-24 오후 6:23	CSS 스타일시트 ...	1KB
 media.all.font.css	2014-11-24 오후 6:23	CSS 스타일시트 ...	2KB
 media.all.layout.css	2014-12-30 오후 5:36	CSS 스타일시트 ...	167KB
 media.all.multiclass.css	2014-11-24 오후 6:23	CSS 스타일시트 ...	72KB
 media.all.reset.css	2014-12-30 오후 4:42	CSS 스타일시트 ...	1KB
 media.all.table.css	2014-12-18 오후 8:21	CSS 스타일시트 ...	2KB
 media.print.layout.css	2014-12-17 오전 9:57	CSS 스타일시트 ...	1KB
 _library.scss	2014-11-28 오후 1:43	SCSS 파일	4KB
 include.scss	2014-12-17 오전 9:44	SCSS 파일	1KB
 media.all.common.scss	2014-12-04 오후 1:05	SCSS 파일	2KB
 media.all.font.scss	2014-11-24 오후 6:23	SCSS 파일	3KB
 media.all.layout.scss	2014-12-30 오후 5:36	SCSS 파일	58KB
 media.all.multiclass.scss	2014-11-24 오후 6:23	SCSS 파일	62KB
 media.all.reset.scss	2014-12-30 오후 4:42	SCSS 파일	2KB
 media.all.table.scss	2014-12-18 오후 8:21	SCSS 파일	2KB
 media.print.layout.scss	2014-12-08 오전 9:23	SCSS 파일	1KB

1.3.7 {projectDir}/trunk/js/ 디렉터리

- jQuery.1.10.2.js를 권장하나 구형 브라우저(ie8~ie9)에 한해서 jquery.1.8.3~4.js를 허용한다.
각 플러그인 들은 별도의 파일로 분리 하여야 하나 경우에따라 lib.anything.js에 포함하여 라이브러리를 배치 할 수 있다.



































js

 jquery.1.10.2.js	2014-11-24 오후 6:23	JScript 스크립트 ...	277KB
 jquery.1.10.2.min.js	2014-11-24 오후 6:23	JScript 스크립트 ...	91KB
 jquery.migrate.1.2.1.js	2014-11-24 오후 6:23	JScript 스크립트 ...	17KB
 jquery.migrate.1.2.1.min.js	2014-11-24 오후 6:23	JScript 스크립트 ...	8KB
 jquery.ui-1.10.3.custom.min.js	2014-11-24 오후 6:23	JScript 스크립트 ...	223KB
 lib.agentfix.js	2014-12-18 오후 1:33	JScript 스크립트 ...	5KB
 lib.anything.js	2014-12-24 오후 7:23	JScript 스크립트 ...	13KB
 lib.control.caller.js	2014-12-30 오후 5:30	JScript 스크립트 ...	9KB
 lib.control.expend.js	2014-12-30 오후 4:24	JScript 스크립트 ...	64KB
 lib.control.js	2014-12-08 오전 1...	JScript 스크립트 ...	57KB
 lib.global.js	2014-11-28 오후 1...	JScript 스크립트 ...	1KB
 lib.html5shiv.js	2014-11-24 오후 6:23	JScript 스크립트 ...	3KB
 lib.IE7.js	2014-11-24 오후 6:23	JScript 스크립트 ...	82KB
 lib.IE7.min.js	2014-11-24 오후 6:23	JScript 스크립트 ...	33KB
 lib.ie7-squish.js	2014-11-24 오후 6:23	JScript 스크립트 ...	2KB
 lib.ie7-squish.min.js	2014-11-24 오후 6:23	JScript 스크립트 ...	2KB
 lib.IE8.js	2014-11-24 오후 6:23	JScript 스크립트 ...	93KB
 lib.IE8.min.js	2014-11-24 오후 6:23	JScript 스크립트 ...	37KB
 lib.IE9.js	2014-11-24 오후 6:23	JScript 스크립트 ...	100KB
 lib.IE9.min.js	2014-11-24 오후 6:23	JScript 스크립트 ...	41KB
 lib.modernizr.js	2014-11-28 오전 1...	JScript 스크립트 ...	15KB

1.3.8 {projectDir}/trunk/images/ 디렉터리

- jpg, gif, png만 허용한다. png8과 png24를 구분하여 사용하는것을 권장한다.
임의로 넣은 이미지의 경우 tmp를 접두사로 붙인다.

images

 pagination_prev.png	2014-12-18 오후 5:49	PNG 파일	2KB
 quick_icon_wrapper.png	2014-12-23 오후 4:57	PNG 파일	2KB
 quick_icon_wrapper_bt.png	2014-12-23 오후 4:59	PNG 파일	1KB
 quick_icon01.png	2014-12-23 오후 4:49	PNG 파일	2KB
 quick_icon02.png	2014-12-23 오후 4:50	PNG 파일	2KB
 quickbarbg_btm.png	2014-12-17 오후 6:20	PNG 파일	1KB
 quickbarbg_top.png	2014-12-17 오후 5:30	PNG 파일	4KB
 quickbarcbg_btm.png	2014-12-17 오후 6:28	PNG 파일	1KB
 quickbarcbg_top.png	2014-12-17 오후 5:57	PNG 파일	2KB
 sidebarbg_btm.png	2014-12-17 오후 5:11	PNG 파일	2KB
 sidebarbg_top.png	2014-12-22 오후 6:11	PNG 파일	8KB
 sidebarcbg_btm.png	2014-12-17 오후 6:13	PNG 파일	2KB
 sidebarcbg_top.png	2014-12-17 오후 5:48	PNG 파일	2KB
 table01rbdr.png	2014-12-18 오후 5:22	PNG 파일	1KB
 theme_changer_icon_n.png	2014-12-17 오후 2:12	PNG 파일	2KB
 theme_changer_icon_o.png	2014-12-17 오후 2:12	PNG 파일	2KB
 tmp_1.png	2014-12-23 오전 9:32	PNG 파일	6KB
 tmp_approacing.png	2014-12-16 오후 2:00	PNG 파일	56KB
 tmp_btn.png	2014-11-26 오후 3:20	PNG 파일	3KB
 tmp_bullets01.png	2014-11-24 오후 1:34	PNG 파일	1KB
 tmp_inbi.png	2014-12-23 오전 1:...	PNG 파일	4KB
 tmp_inbi2.png	2014-12-23 오전 1:...	PNG 파일	3KB
 tmp_login.png	2014-12-16 오후 3:12	PNG 파일	23KB
 tmp_logo.png	2014-11-26 오후 2:21	PNG 파일	819KB
 tmp_mf_2_left1.png	2014-11-28 오후 3:45	PNG 파일	23KB
 tmp_mf_2_top.png	2014-11-28 오후 3:18	PNG 파일	21KB
 tmp_mf_2_top2.png	2014-11-28 오후 3:21	PNG 파일	5KB
 tmp_mig.png	2014-12-23 오후 5:35	PNG 파일	4KB
 tmp_qclose.png	2014-11-28 오후 5:02	PNG 파일	2KB
 tmp_qopen.png	2014-11-28 오후 4:17	PNG 파일	2KB
 tmp_solutionlogo.png	2014-11-27 오후 6:37	PNG 파일	2KB
 tmp_wr.png	2014-12-26 오후 5:03	PNG 파일	11KB
 user_info_bg.png	2014-12-23 오전 9:57	PNG 파일	2KB
 user_info_icon.png	2014-12-23 오전 9:34	PNG 파일	2KB

1.3.9 {projectDir}/trunk/assets/ 디렉터리

- 음악, 동영상, 웹 폰트, pdf 등의 웹에서 표현 할 수 있는 바이너리 파일은 assets에 넣는다.
또한 이곳에 파일 링크를 할 수 있다.

2. 컨벤션

- 파일명과 코딩 규칙에 대해 규약 한다.

w3c 웹표준에 준수 하여야 한다.

파일명은 영문~ 숫자로 구성 되어야 하며 dash(-)와 under-score(_)를 포함 할 수 있다.
영어를 제외한 한글/한자/일본어 등의 파일명은 인정하지 않는다.

tab&space는 4/4로 한다.

2.1.1 html 작성 규칙#1

- **html5를 기본 마크업 doctype으로 사용**한다. <http://www.w3.org/TR/html5/>

html파일은 서버 기술을 제외한 **클라이언트 고유 기능은 모두 동작** 하여야 하며,
K-WAH4.0의 웹 접근성 자동점검 보고서의 100%의 준수와
w3c Markup Validation Service(<http://validator.w3.org/>)에서 **Passed**되어야 한다. (예외로 warning은 인정함)
위 세가지를 준수 하지 않을 경우 작업의 결과물로 인정 하지 않는다.

웹표준과 접근성에 준수한 태그와 마크업 기술을 사용 하여야 한다.
이미지는 alt와 title을 포함하고 form요소는 fieldset을 포함 하여야 하며, input은 label요소를 포함 하여야 한다. table은 caption을 포함 하여야 한다.

2.1.2 html 작성 규칙#2

· **아래의 마크업 기본을 유지 한다.** 작업자의 스킬 역량에 따라 아래 사항에서 더 추가 되거나 변경 될 수 있다.

- A. 붉은색으로 표기된 내용은 기본적으로 꼭 포함하여야 하는 내용이다.
- B. 연녹색으로 표기된 내용은 파비콘과 애플 아이폰을 위한 숏컷 아이콘이다.
- C. 노란색으로 표기된 내용은 모바일 브라우저의 확대 배율 조절을 위한 메타 태그 이다.
- D. 보라색으로 표기된 내용은 css링크이다. / media.all.agentfix.css는 브라우저 픽스용이다.
- E. 연갈색으로 표기된 내용은 javascript 전역 변수이다.
- F. 연청색으로 표기된 내용은 크로스 브라우징을 위한 자바스크립트 패치 및 유저에이전트 구분을 위한 fix 내용이다.
- G. 진청색으로 표기된 내용에 구현시 필요한 자바스크립트를 넣는다.

```
A. <!doctype html>
<html xmlns="http://www.w3.org/1999/xhtml" lang="ko">
  <head>
    <title>페이지명 - 프로젝트명 by 회사명</title>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <!--[if IE]><meta http-equiv="X-UA-Compatible" content="IE=edge;chrome=1" /><![endif]-->
    <meta name="keywords" content="회사명" />
    <meta name="description" content="프로젝트명" />
  B. <link rel="shortcut icon" href="../images/shortcut_icon.ico" type="image/x-icon" />
    <link rel="icon" href="../images/favicon.ico" type="image/x-icon" />
  C. <meta name="viewport" id="viewport" content="user-scalable=no, initial-scale=1.0, minimum-scale=1.0, maximum-scale=1.0, width=device-width" />
  D. <link rel="stylesheet" type="text/css" href="../css/media.all.css" />
    <link rel="stylesheet" type="text/css" href="../css/media.all.agentfix.css" />
  E. <script type="text/javascript" src="../js/lib.global.js" charset="utf-8" ></script>
  F. <!--[if IE 7]><script type="text/javascript" src="../js/lib.IE7.min.js" charset="utf-8" ></script><![endif]-->
    <!--[if IE 8]><script type="text/javascript" src="../js/lib.IE8.min.js" charset="utf-8" ></script><![endif]-->
    <!--[if IE 9]><script type="text/javascript" src="../js/lib.IE9.min.js" charset="utf-8" ></script><![endif]-->
  G. <script type="text/javascript" src="../js/lib.anything.js" charset="utf-8" ></script>
    <script type="text/javascript" src="../js/lib.control.js" charset="utf-8" ></script>
    <script type="text/javascript" src="../js/lib.control.expend.js" charset="utf-8" ></script>
    <script type="text/javascript" src="../js/lib.control.caller.js" charset="utf-8" ></script>
  </head>
  <body>
  </body>
</html>
```

2.2.1 css 작성 규칙 #1

- **css level 2.1을 기본 마크업 규칙으로 사용**하나 javascript에 의한 패치(selectivizr, HTML5 shiv)시 구형 브라우저에서 사용 불가능한 일부 셀렉터를 사용 할 수 있다. ie8을 포함 하여야 한다.

공공기관 웹표준 적용시 이슈화 되므로 css핵을 사용하여선 절대 안된다. 별도의 scope형태로 분리 하거나, 별도의 css 파일에 기술하여야 한다.

css명명은 dash와 camelCase를 사용하지 않는다. under-score와 영문 숫자를 혼용 하여 사용한다.

예) 모두 맞춤 : table_01, my_table

모두 틀림 : table-01, tableMyTd, table01MyTd

js에서 접근하는 css값은 고유값으로 그 의미를 부여 가능하며 이경우 camelCase를 사용할 수 있다.

```
$('.myDiv').click(function(){});
```

<div class="my_div myDiv"></div> //my_div는 html에서 myDiv는 js에서 사용한다.

2.2.2 css 작성 규칙 #2

· 컨테이닝 기본으로 작성한다

```
<div class="example_container">  
  <div class="inner">  
    <span class="test"> </span>  
    <strong class="text"> </strong>  
  </div>  
</div>
```

css의 경우

```
.example_container {}  
.example_container .inner{}  
.example_container .inner .test{}  
.example_container .inner .text{}
```

scss의 경우

```
.example_container {  
  .inner{  
    .test{}  
    .text{}  
  }  
}
```

2.3.1 js 작성 규칙 #1

- **ecma-262 표준화 스크립트 프로그래밍을 기본으로 한다.** 객체 지향형, 함수형의 호출은 기본으로 하나 jQuery 플러그인 형태로 작성 될 수 있으며, 별도의 js파일로 분리 하여야 한다. html 페이지내에 함수의 호출외에 함수의 몸체가 포함 되서는 안된다.
잘 작성된 경우는 유효한 문법이라 표기 하였다.
대부분은 자유롭게 기능을 구현 하면되나, 유지보수를 위하여 "함수"와 "함수의 호출"을 분리 하고 "함수 덩어리"를 js로 분리 하는 형태만 유지된다면 유효하다.

// 이 "함수의 호출" 부분은 html내에 포함 될 수 있다. 유효하게 작성된 문법.

```
$(function(){  
    $('div#test').myTest();  
});
```

// 혹은 js파일 내에 아래와 같이 element의 존재 여부에 따라 "함수의 호출"을 별도로 분류 할 수 있다. 유효하게 작성된 문법.

```
$(function(){  
    if($('.pageHasElement').length > 0){  
        $('div#test').myTest();  
    }  
});
```

// 아래의 함수의 본체는 별도의 스크립트로 분류 되어야 한다.(lib.functions.js 혹은 작업자의 스킷셋의 역량에 따른 별도의 파일로 구분 되어야 한다.)

// 모두 유효한 문법이다.

```
$.fn.myTest = function(option) {  
    option = option || {};  
};
```

```
function myTest(option){  
    option = option || {};  
}
```

2.3.2 js 작성 규칙 #2

· // 객체 지향형으로 작성한 자바스크립트는 매우 권장하는 형태 이다. 유효한 문법

```
var myTest = {  
  option : null,  
  init:function(option){  
    this.option = option || {};  
  }  
};
```

// 그러나, 호출과 함수가 붙어있는 경우는 유효하지 않다.

```
function myTest(option){  
}();
```

// 아래와 같이 호출과 함수의 이벤트 바인딩 형태가 html페이지내에 올 수는 있다. 유효하다.

```
$(function(){  
  $('click').on('click',function(){  
    myClickFunction(); // "함수의 호출" 부분  
    return;  
  });  
});
```

// 아래와 같이 함수의 호출과 함수가 분리되지 않았다면 유효 하지 않다.

```
$(function(){  
  $('click').on('click',function(){  
    if(a = undefined && a = null){  
      //bla bla bla  
    }  
    return;  
  });  
});
```

2,3.3 images 네이밍 혹은 slice 규칙

- 슬라이스된 이미지의 네이밍에 경우는 자유로우나 임시로 사용한 이미지는 tmp_를 접두사로 붙인다.
png24의 경우 그림자를 포함하며, Transparency 하여야 하고, Matted 해서는 안된다.
파일 포맷의 선택에 관련하여서는 아래의 표를 참조한다.

요소	권장하는 파일 포맷
디자인 요소가 투명 영역을 포함 하는 경우	png24
디자인 요소가 투명 영역을 포함하나 도트 형태의 표현인 경우	png8
사진과 같이 픽셀의 조밀도가 압축이 필요한 경우	jpg(권장) or png24
디자인 요소가 플립북 애니메이션 효과를 필요로 하는 경우	gif를 사용하거나 jquery로 처리.

summary : 디자인요소의 이미지화 가이드 표.

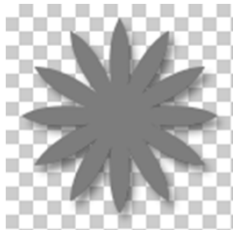


figure: Transparency



figure: Matte

3. 파일리스트

- 작업 내역을 한눈에 파악 할 수 있게 파일리스트를 관리 한다.

3.1 파일리스트의 관리 #1

- 파일리스트는 별첨된 파일리스트 양식을 사용한다.

파일리스트는 통상 html파일만 관리하며, 프로젝트의 성격에따라 css와 js가 관리 될 수 있다.
 통상 파일리스트에 기록되는 시기는 웹접근성과 웹표준이 완료 되었을 경우 이다.
 모든 작업이 완료 되었을 경우는 .dtr엘레먼트에 .compleated 클래스를 붙인다.

프로젝트의 제목 표시

프로젝트의 일자 표시

filelist of test project (2014-11-26 ~ 진행중)

published by Front-End Developer naiyumie (naiyumie@gmail.com)

작업을 진행하는 담당자의 이름과 이메일

프로젝트의 개요 표기

■ 프로젝트 개요

- 해당 웹퍼블리싱 페이지의 doctype은 html5로 결정함.
- html파일은 w3c validator와 K-WAH4.0 툴을 사용하여 점검 하며 준수를 100% 기본으로 한다.
- 이하 자세한 규칙은 {projectDir}/ref/ftrules의 내용을 따른다.

· file of contents [html]

html파일리스트의 관리

no	파일명 (파일이름과 경로)	구분 (파일의 용도)	일자 (작성 및 변경일자)	주석 (기타 변경 내용 및 특이사항)
0	./html/tmp.html	임시	· 2014-12-31 ~ 2015-01-01	· 테스트1 · 테스트2
1	./html/tmp2.html	임시 파일리스트	· 2014-12-31 ~ 2015-01-01	· 테스트1 · 테스트2

완료 되었을 경우 .compleated 클래스를 추가한다.

figure : 소규모의 그룹으로 html을 작성 하는 경우.

3.1 파일리스트의 관리 #2

filelist of test project (2014-11-26 ~ 진행중)

published by Front-End Developer naiyumie (naiyumie@gmail.com)

작업을 진행하는 선임자의 이름과 이메일

■ 프로젝트 개요

- 해당 웹퍼블리싱 페이지의 doctype은 html5로 결정함.
- html파일은 w3c validator와 K-WAH4.0 툴을 사용하여 점검 하며 준수를 100% 기본으로 한다.
- 이하 자세한 규칙은 {projectDir}/ref/fttrules의 내용을 따른다.

■ 특이사항

- js파일이 일괄 수정 되었습니다.
- 현재 리뉴얼 관련 작업 진행 중입니다.

특이사항의 표기

- file of contents [html]

no	파일명 (파일이름과 경로)	구분 (파일의 용도)	일자 (작성 및 변경일자)	주석 (기타 변경 내용 및 특이사항)
0	./html/tmp.html	임시	• 2014-12-31 ~ 2015-01-01	• 작성자 : 홍길동
1	./html/tmp2.html	임시 파일리스트	• 2014-12-31 ~ 2015-01-01	• 작성자 : 김길동

작성자/수정자의
주석 표기

figure : 중규모의 그룹으로 html을 작성 하는 경우.

3.2 파일리스트의 관리 #3

filelist of test project (2014-11-26 ~ 진행중)

published by Front-End Developer
naiyumie (naiyumie@gmail.com)

■ 프로젝트 개요

- 해당 웹퍼블리싱 페이지의 doctype은 html5로 결정함.
- html파일은 w3c validator와 K-WAH4.0 툴을 사용하여 점검 하며 준수율 100% 기본으로 한다.
- 이하 자세한 규칙은 {projectDir}/ref/ftrules의 내용을 따른다.

■ 특이사항

- js파일이 일괄 수정 되었습니다.
- 현재 리뉴얼 관련 작업 진행 중입니다.

file of contents [html]

no	파일명(파일이름과 경로)	구분(파일의 용도)	일자(작성 및 변경일자)	주석(기타 변경 내용 및 특이사항)
0	./html/tmp.html	임시	2014-12-31 ~ 2015-01-01	작성자 : 홍길동
1	./html/tmp2.html	임시 파일리스트	2014-12-31 ~ 2015-01-01	작성자 : 김길동

file of contents [images]

no	파일명(파일이름과 경로)	구분(파일의 용도)	일자(작성 및 변경일자)	주석(기타 변경 내용 및 특이사항)
1	./images/tmp1.jpg	임시	2014-12-31 ~ 2015-01-01	작성자 : 홍길동
2	./images/tmp2.jpg	임시	2014-12-31 ~ 2015-01-01	작성자 : 김길동

figure : 대규모의 그룹으로 html을 작성 하는 경우.

filelist of test project (2014-11-26 ~ 진행중)

published by Front-End Developer
naiyumie (naiyumie@gmail.com)

■ 프로젝트 개요

- 해당 웹퍼블리싱 페이지의 doctype은 html5로 결정함.
- html파일은 w3c validator와 K-WAH4.0 툴을 사용하여 점검 하며 준수율 100% 기본으로 한다.
- 이하 자세한 규칙은 {projectDir}/ref/ftrules의 내용을 따른다.

■ 특이사항

- js파일이 일괄 수정 되었습니다.
- 현재 리뉴얼 관련 작업 진행 중입니다.

· file of contents [html]

no	파일명(파일이름과 경로)	구분(파일의 용도)	일자(작성 및 변경일자)	주석(기타 변경 내용 및 특이사항)
0	./html/tmp.html	임시	2014-12-31 ~ 2015-01-01	작성자 : 홍길동
1	./html/tmp2.html	임시 파일리스트	2014-12-31 ~ 2015-01-01	작성자 : 김길동

모바일의 경우
한줄로 표기 된다.

figure : 모바일 프로젝트의 경우.

4. 크로스 브라우징 룰

- 기준 브라우저와 크로스 브라우징

4.1 크로스 브라우징과 기준 웹브라우저

- 일반적으로 **크로스 브라우징은 ie8~11을 기준으로 하며 ie8은 MS Winodws-xp, MS Windows-7두가지 버전이 해당한다.**
프로젝트 성격상 ie버전이 ie9이상으로 정해질 수 있다.
프로젝트 성격에 따라 브라우저 스펙에 따른 기능 구현 불가 요소에 대해서 대체 콘텐츠를 마련 할 수 있다.

브라우저의 우선 순위	해당 브라우저
최우선 순위의 기준 웹브라우저	ie8~ie11
2순위 웹브라우저	google chrome
3순위 웹브라우저	safari
4순위 웹브라우저	firefox
크로스 브라우징에 해당하지 않는 웹브라우저	코어 변형 렌더링 엔진을 사용하지 않는 국산과 외산 브라우저 (swing포함) 점유율 1%미만 웹브라우저(opera)

summary : 브라우저의 우선 순위 가이드 표.

4.2 크로스 브라우징과 기준 웹브라우저

- ie의 경우 호환성 모드가 아닌 가상머신 환경에서 테스트 하여야 한다.

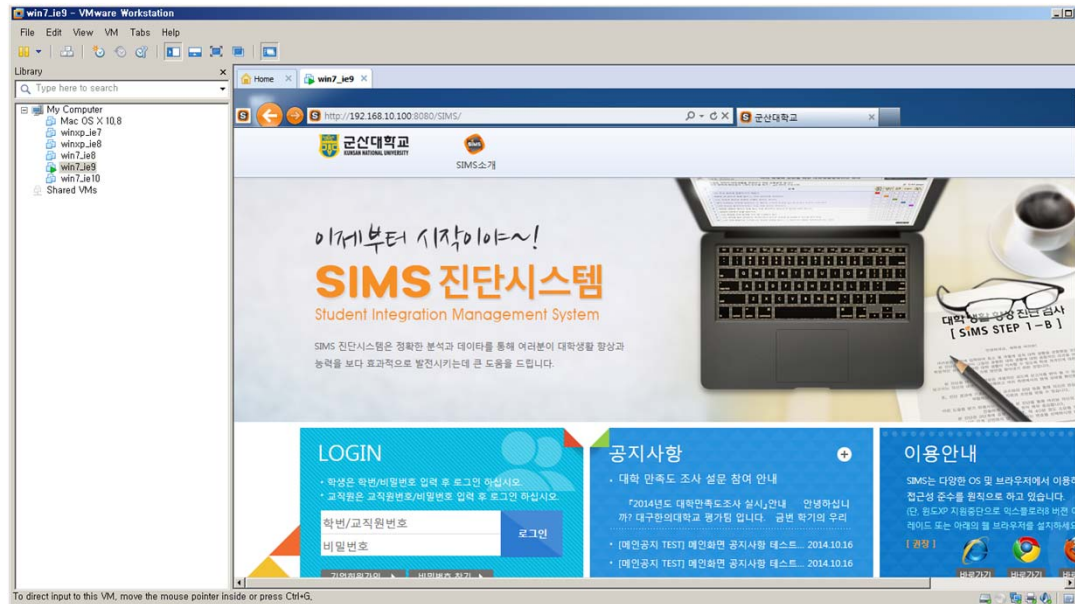


figure : VMWare를 사용하여 테스트 중인 모습.

5.1 K-WAH4 웹접근성 자동점검 통과.

· K-WAH4.0의 웹 접근성 자동점검 보고서의 100%의 준수.

위 Validation을 준수 하지 않을 경우 작업의 결과물로 인정 하지 않는다.

웹 접근성 자동점검은 수시로 진행하여야 하며, tags에 남기는 시점(타인에게 배포 하거나, 작업이 일시 중단 될 경우)에 필수한 사항이다.

웹 접근성 자동점검 보고서



웹사이트	D:\workspace\publish_workspace\publish3\porebnTest01\brunkhtml_sceability	수집 페이지	17 페이지
평가기준	한국형 웹 콘텐츠 접근성 지침 2.0	검사 일자	2014년 12월 30일

자 동 점 검 결 과

평가 항목	합계수	오류수	준수율
1. 대체 텍스트 제공	117	0	100.0
 대체 텍스트 없음	116	0	100.0
<area> 대체 텍스트 없음	-	-	-
<input type="image"> 대체 텍스트 없음	1	0	100.0
<applet> 대체 텍스트 없음	-	-	-
2. 제목 제공	17	0	100.0
페이지 제목 없음	17	0	100.0
<frame> 제목 없음	-	-	-
<iframe> 제목 없음	-	-	-
3. 기본언어 정의	17	0	100.0
기본언어 미정의	17	0	100.0
4. 새창열림 사전공지	56	0	100.0
<a> 알려지지 않은 새창열기	56	0	100.0
<area> 알려지지 않은 새창열기	-	-	-
5. 레이아웃 제공	42	0	100.0
<input> 레이아웃 미제공	42	0	100.0
<textarea> 레이아웃 미제공	-	-	-
<select> 레이아웃 미제공	-	-	-
6. 마크업 문법	17	0	100.0
id 중복 오류	17	0	100.0
속성 이름 중복 오류	17	0	100.0
태그 열고 닫음 오류	17	0	100.0
닫는 태그 누락 오류	17	0	100.0
여는 태그 누락 오류	17	0	100.0
태그 중첩 오류	17	0	100.0

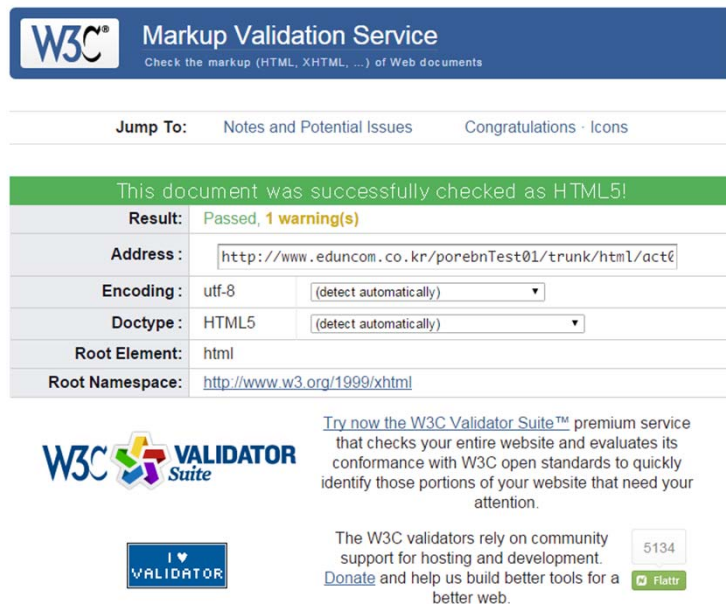
※ 주의사항

1. K-WAH 자동점검기는 웹 접근성 표준 중 기계적으로 평가 가능한 6개 항목에 대한 점검 결과만을 제공합니다.
2. 웹 접근성의 준수여부는 K-WAH 자동점검기만으로 판단될 수 없으며, 반드시 웹접근성표준의 전체 지표에 따른 수동평가를 통해 준수 여부를 확인하여야 합니다.
3. K-WAH가 수집하는 웹 페이지의 수집 순서는 네트워크의 속도에 따라 달라질 수 있습니다.
4. 평가항목 중 "기본언어정의", "마크업 문법"의 준수율은 전체 수집페이지 수 대비 문제가 없는 페이지 수의 비율로 계산됩니다.

figure : 웹 접근성 자동점검 보고서는 항상 html문서와 한 set를 이루어야 한다.

5.2 w3c validator 통과

- w3c Markup Validation Service(<http://validator.w3.org/>)에서 Passed되어야 한다. (예외로 warning은 인정함)
위 Validation을 준수 하지 않을 경우 작업의 결과물로 인정 하지 않는다.
Markup Validation은 퍼블리싱 진행시 해당 URL에 대해 진행하여야 한다.



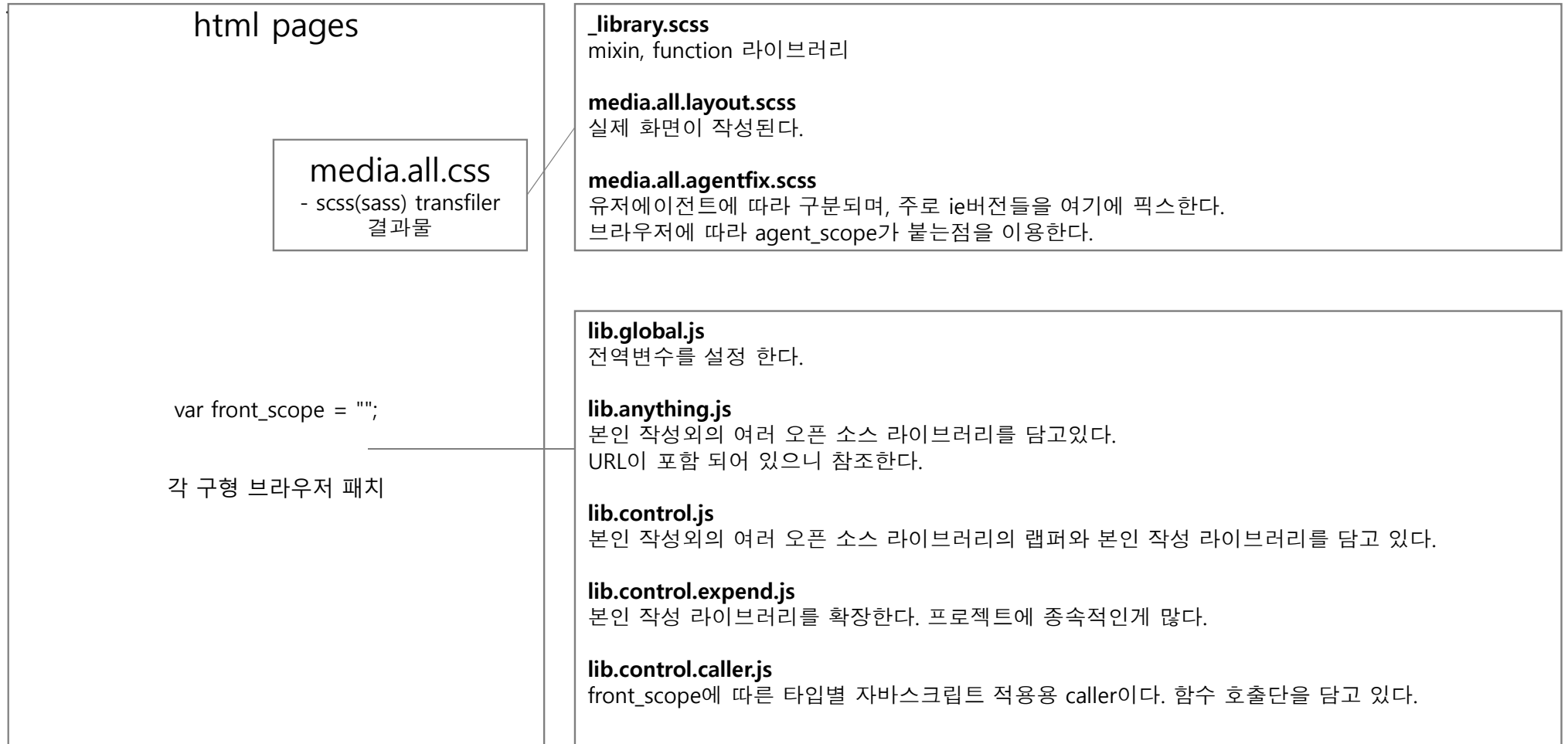
The screenshot displays the W3C Markup Validation Service interface. At the top, a blue header contains the W3C logo and the text "Markup Validation Service" with a subtitle "Check the markup (HTML, XHTML, ...) of Web documents". Below this, a navigation bar includes "Jump To:", "Notes and Potential Issues", and "Congratulations · Icons". A green banner states "This document was successfully checked as HTML5!". The main content area shows the validation results in a table-like format:

Result:	Passed, 1 warning(s)
Address :	<input type="text" value="http://www.eduncom.co.kr/porebnTest01/trunk/html/act&"/>
Encoding :	utf-8 (detect automatically) ▼
Doctype :	HTML5 (detect automatically) ▼
Root Element:	html
Root Namespace:	http://www.w3.org/1999/xhtml

Below the results, there is a section for the "W3C Validator Suite" with a logo and text: "Try now the W3C Validator Suite™ premium service that checks your entire website and evaluates its conformance with W3C open standards to quickly identify those portions of your website that need your attention." To the left of this text is a small "I ♥ VALIDATOR" button. To the right, it says "The W3C validators rely on community support for hosting and development. Donate and help us build better tools for a better web." with a "5134" counter and a "Flattr" button.

figure : w3c Markup Validation Service를 준수한다.

6.1 HTML 파일 헤더의 구조.



6.2 SCSS의 사용.

- *.scss 로 기술되어 있다. 이것은 SASS 트랜스파일러에 의해 *.css로 컨버팅 된다.
즉, css파일을 수정 하면 나중에 트랜스파일링 과 동시에 삭제 된다.
.scss는 _.scss를 인클루드 한다. (SASS 파일 명명 규칙상 앞에 _(언더스코어)가 붙었다.)

1. 공식사이트 : <http://sass-lang.com/guide>
2. 생활코딩 : <http://opentutorials.org/course/470/2488>
을 참조한다.

scss는 아주 쉬우며 CSS문법을 알고 있는 사람이라면 30분 이내로 적응 할 수 있고,
보통 트리 구조의 이해, mixin의 이해가 끝나면 바로 사용 할 수 있다.

6.3.1 front_scope 의 사용. #1

- 1 유지보수시 디자인 변경, 페이지 추가시 최소한의 코딩과 DOM수정을 하지 않기 위해서.

1.1 타입 구분용.

위처럼 총 8본에 프로젝트가 있습니다.

Atype : Page1,5,

Btype : Page2,4,6,7

Ctype : Page3

Dtype : Page8

위처럼 타입을 구분 할 수 있는데 이를 '타입 구분 요소'로 정의 할 수 있습니다.

1.2 공통요소(public element)

A~D타입 모두 공통적으로 쓸 수 있는 요소가 있습니다. 그러한 요소들을 하나의 CSS로 관리 하기 위함 입니다.

```
.front_scope_a, .front_scope_b, .front_scope_c, .front_scope_d {  
    .common_element{border:1px solid red;} // 이렇게 공통 요소를 정의합니다.  
}
```

6.3.2 #2

· 1.3 개별요소(individual element)

CSS특성상 해당 속성을 교체하면 모든 페이지가 일괄 교체됩니다.

페이지마다 CSS를 따로 두는 방식은 위의 8본의 웹페이지의 경우 8군데를 수정 하여야 하므로, 하나로 갑니다.

이때 atype의 특정요소만 디자인이 교체되거나, 변경사항이 일어 날 시 DOM을 교체 하지 않고 CSS와 jquery만으로 컨트롤이 가능 합니다.

```
<script type="text/javascript" language="javascript">
//<![CDATA[
    var front_scope = "front_scope_a";
//]]>
</script>
<div class="front_scope_atype">
위의 스코핑은 타입 구분 요소의 구분값이 됩니다.
```

.front_scope_a .common_element{border:1px solid blue;} 이런식으로 오버라이딩 하여 속성을 재정의 할수 있습니다.

즉 B,C,D타입은 테두리가 붉은색으로 나오나 A타입은 파란색으로 나오게 됩니다.

기존 방식이었다면

```
<div class="border_red">이러한 엘리먼트가 8페이지 모두 붙어있고, 이를 파란색으로 교체하기 위해서
<div class="border_blue">로 2페이지를 수정 해야 하는 상황이 오게 되는거지요.
```

또한 스크립트 역시 해당 스코핑으로 함수를 호출 할 수 있습니다.

```
if(front_scope == "front_scope_a"){
    // 타입a만 공통 js에서 호출.
}
```

6.3.3 #3

· 1.4 잘못된 퍼블리싱과 개발의 프렉탈

1.4.1 위의 스코핑을 사용하는 경우 경우 잘못된점은 페이지당 scope를 추가하여야 하고 어렵다 라는 점입니다.

1.4.2 인클루드는 사라져가고 있습니다.

python의 django / flask, ruby on rails, php-codeigniter / laravel등 신 프레임워크에서는

프레젠테이션 영역의 인클루드를 완전히 배제 하고 있습니다.

레이아웃을 정의하고 내부를 모조변수로 대치 하는 형태로 갑니다. jsp-tiles도 그러한 형태 입니다.

1.4.3 페이지 단위로 모두 CSS가 개별로 있는 경우 공통 부분을 추출하기도 어렵고,

페이지 추가시 공통요소도 다시 만들어야 하는 부담감이 있습니다. 보통 초보들이 하는 방식입니다.

1.4.4.1 스코핑이 적용되지 않는 경우 핵을 사용하게됩니다.

한번 사용하게되면 끝없이 꼬이고,

"css를 별도 파일로 분리해놓음 > 페이지 포함 css > 인라인 css > 스크립트로 css속성 수정 > 인터벌을 주어 지속적으로 속성 먹이기 형태로" 교체 하게 됩니다.

우측으로 갈 수록 유지보수가 어렵고 다시만들어야 합니다.

1.4.4.2 서버쪽에서 scriptlet (<% jsp 구문 %> <?php php구문 ?>) 붙이기도 복잡해 집니다. 과거 html4때의 개발을 떠올려 보세요.

1.4.4.3 또한 크로스 브라우징을 위하여 핵을 사용하게 되며, 많은 퍼블리셔들이 CSS핵을 당연하다는 듯이 사용하지만

보통 CSS핵을 사용하게 되면 단발성 프로젝트에만 적합한 기법입니다. (라이프 사이클이 짧은 웹 애플리케이션)

1.4.4.4 다양한 경우에 DOM을 수정하면 적게는 html 몇장 수정 부터, 서버스크립트의 수정, 쿼리 및 프로시저의 수정,

더 크게 c로 짜여진 cgi웹의 경우 unix gcc 버전 까지 영향을 받게 됩니다.

1.4.4.5 다양한 상황에서 DOM을 교체 할지 css를 교체 할지는 기능요구와 디자인에 따라 제가 판단합니다.

1.4.4.6 웹 프레젠테이션 영역과 로직영역의 분리는 서버쪽에서 html태그를 출력하는 형태를 최대한 자제 하여야 합니다.

2 CSS Validation & 보안 검사 문제

w3c html 웹표준 유효성 검사 외에도 css 문법 유효성 검사도 있습니다.

우리가 공산품을 살때 ISO, Ks마크가 있듯 웹에는 웹접근성마크가 있는 것이지요.

고객은 국가 표준인 웹 품질검사(웹표준, 웹접근성, owasp, fortify 소스보안검사 등)에 무사히 통과한 제품을 원하며

법률상으로 지정 되어 있으므로 RFP를 교체 할 수 가 없습니다.

보통 css 유효성 검사를 보진 않지만 까다로운 곳은 봅니다. scoping이 없다면 여러 핵들로 인해서 유효성 검사에 불합격을 받게 됩니다.

위와 같은 이슈로 인해서 핵을 사용하지 않고 user-agent의 구분과 scoping구분으로 품질 유지를 하고 있습니다.

6.4 agent_scope의 사용.

- 크로스 브라우징을 위하여 lib.agentfix.js 파일에서 agent_scope를 body에 addClass하며 agent_scope에 따라서 css 렌더링이 달라지게 된다. 특정 브라우저에서만 깨지는 요소들을 CSS HACK으로 단발성으로 처리 하지 않고 agent_scope를 이용하여 별도 기술 한다. 보통 AgentFix 라이브러리로 fix하나 server-side기술을 사용할 수 도 있다. jsp의 경우 아래의 코드를 참조한다.

```
browser_chrome.css
browser_chrome.scss
browser_firefox.css
browser_firefox.scss
browser_ie.css
browser_ie.scss
browser_safari.css
browser_safari.scss
```

위와 같은 파일을 준비하고

```
<%@ page contentType="text/html; charset=utf-8" pageEncoding="utf-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%
    String browser = request.getHeader("User-Agent");
    String browserCss = "";

    if ( browser != null && browser.indexOf("Chrome/") != -1 ) {
        browserCss = "chrome";
    } else if( browser != null && browser.indexOf("Firefox/") != -1 ){
        browserCss = "firefox";
    } else if( browser != null && browser.indexOf("Safari/") != -1 ){
        browserCss = "safari";
    } else if( browser != null && browser.indexOf("Opera/") != -1 ){
        browserCss = "opera";
    } else {
        browserCss = "ie";
    }
%>
<link rel="stylesheet" type="text/css" href="../css/browser_<%=browserCss%>.css" />
```

이런식으로 코딩 하면 된다.

6.5 html5의 사용.

· HTML5 8대 기술과 브라우저 연계성.

8대 기술 항목	설명	브라우저 지원	html5 shiv 패치 (구형 브라우저에서도 케어 가능)
Video & Audio	HTML5에 신규 추가된 video	ie9+/크롬/사파리류	O
3D, Graphics & Effects	SVG, Canvas, WebGL, CSS3 3D를 이용하여 3d효과를 준다. 많은 브라우저가 부분 지원하고 있다.	ie9+/크롬/사파리류	X (일부)
Offline & Storage	localStorage를 이용하여 사용자가 설정한 메인 페이지 배치설정 값을 기억해놓고 재접속시 이 상태를 유지한다.	ie9+/크롬/사파리류	X
Device & Access	장치의 방향, 가속도나 모션 센서 등에 대한 정보를 가지고 이를 이미지를 이용하여 사용자가 확인할 수 있게 한다.	모바일 ONLY	X
Web Socket	웹소켓을 이용한 다중 채팅	ie9+/크롬/사파리류	X
Geo-Location	GeoLocation을 이용한 현재 접속위치 위도,경도를 확인	ie9+/크롬/사파리류	?
Sementic	새로 추가된 <section>, <header>, <nav>, <article>, <footer> HTML5 tag를 사용하여 기존에 비해서 의미있게 마크업	ie9+/크롬/사파리류	O

6.6 RFP와 웹접근성 연계성

· 1 개발사 측면

최근 기술을 사용 하고 싶어 한다.

1 고객사 측면(RFP)

IE7+환경을 요구 하고 있으며, 웹접근성이 기본이다.

3 웹접근성 측면

xhtml 1.1 + k-wah 4.0 기반으로 웹접근성을 준수 하여야 RFP에 이상이 없다.

4 웹브라우저 측면

HTML5를 사용하되 HTML4의 하위 호완성을 유지 하여 코딩 하면 RFP상에 문제 없다.
CSS3를 사용하는 경우 jQuery로 대체 하여 사용 가능하다.
k-wah 4.0 체크시 html5모드로 놓고 체크하면 문제 없다.

6.7 개발과의 연계성 #1

1. SERVER-SIDE 개발과의 협업에 있어 무엇이 FRONT-END를 망쳐 깨진 애플리케이션을 만드는가?

- 웹표준 검사와 가상머신을 통한 크로스브라우징 테스트,
- 웹접근성 검사를 통한 검증된 HTML을 제공한다.

예시) 퍼블리셔 7년차 고민 사례

전에 하던 사람이 짜 놓은 코드를 개발없이 뷰 페이지에서 수정하는건 문제가 없는데,
개발코드가 들어가면서 슬라이드배너나 이런저런것들이 많이 깨졌는데 저더러 잡으라고 하네요..
노력은 하고 있는데, 왜 이거밖에 안됐냐고 타박이나 듣고 있고..

특히 이전 퍼블리셔가 해둔 라이브러리들..
뷰페이지에서 잘 나오는데, 개발 들어간 페이지에서 다 깨져서 나와요..
이거.. 제가 어떻게 잡아야하나요..

2. SERVER-SIDE 코드와 FRONT-END코드와의 연계성(커플링)을 최소화 한다.

- 스프링 MVC와 같은 MVC 모델을 사용하여(모델2)로 개발한다.
- MVC기본으로 최대한 HTML 본연의 모습을 유지한다.
- SERVER-SIDE 코드와 FRONT-END코드와의 연계성(커플링)을 최소화 한다.
 - javascript로 html을 조작하는 형태를 최대한 지향 한다.
 - view단에 서버 코드를 이용하여 출력을 제어하는 형태는 최대한 지양한다.
- controller단에 로직을 구현하고 view단에 로직을 하드코딩 하지 않는다.

틀린 예시) incorrect, jstl의 경우

```
<c:if test="${Conf.as != 10 || Conf.ab == 20}">  
  <div>The condition is true!</div>  
</c:if>
```

```
<c:if test="${Conf.as == 10 || Conf.ab != 20}">  
  <div>The condition is false </div>  
</c:if>
```

맞는 예시) correct, velocity의 경우

```
#if ($condition)  
  <div>The condition is true!</div>  
#else  
  <div>The condition is false</div>  
#end
```

6.8 테이블(표)의 웹접근성

· 3.3.2 (표의 구성) 표는 이해하기 쉽게 구성해야 한다.

표를 제공할 경우, 시각장애인 등도 이해할 수 있도록 표의 이해를 돕기 위한 내용 및 구조에 대한 정보를 제공해야 한다.

1) 표의 구성: 데이터를 표로 구성할 경우, 표의 내용, 구조 등을 이해할 수 있도록 구성해야 한다. 표에는 그 내용을 요약한 정보를 제목 또는 요약으로 제공하여 표의 내용을 예측할 수 있도록 한다. HTML의 경우, CAPTION 요소를 사용하여 표의 제목을 제공한다.

2) 셀의 구성: 표의 손쉬운 내비게이션을 위하여 표의 셀은 제목(<th>)과 내용(<td>)을 구분할 수 있는 태그를 이용해야 한다.

기대 효과

1) 논리적으로 구성된 콘텐츠는 인지, 언어, 학습 장애가 있는 사용자들이 콘텐츠를 이해하는 데 도움을 준다. 또한 화면 확대 프로그램을 사용할 때 맥락을 찾기 어려운 시각 장애인에게도 매우 유용하다.

2) 논리적으로 구성된 웹 콘텐츠는 스타일 시트(Style sheet)를 바꾸거나 기능을 제거 하더라도 그 내용을 순서대로 읽어 문서의 의미를 이해하기가 쉽다.

```
<table class="default" role="presentation" summary="Criteria Records of Past studies">
```

```
<colgroup>
```

```
<col width="30%" />
```

```
<col width="30%" />
```

```
<col width="30%" />
```

```
</colgroup>
```

```
<thead>
```

```
<tr>
```

```
<th scope="col" class="first">Records of Past studies</th>
```

```
<th scope="col">Personal interview</th>
```

```
<th scope="col">Total Score</th>
```

```
</tr>
```

```
</thead>
```

```
<tbody>
```

```
<tr>
```

```
<td>50%(150 pts)</td>
```

```
<td>50%(150 pts)</td>
```

```
<td>100%(300 pts)</td>
```

```
</tr>
```

```
</tbody>
```

```
</table>
```

이정도만 지켜도 아주 훌륭하다.

6.9 UI툴을 사용하여도 웹접근성은 필수

· 지침 4.2 (웹 애플리케이션 접근성) 웹 애플리케이션은 접근성이 있어야 한다.

용어 설명

- 1) 웹 애플리케이션: 웹 애플리케이션: 웹 콘텐츠에 포함되어 특정한 기능을 수행하도록 구성된 소프트웨어의 일종으로, 리치 인터넷 애플리케이션(RIA: Rich Internet Application)이라고도 한다. 본 표준이 적용되는 웹 애플리케이션은 웹 콘텐츠에 내장되어 복수의 웹 브라우저에서 공통적으로 사용할 수 있는 것으로 한정한다. 따라서 적용 대상은 플러그인(Plug-in) 콘텐츠와 자바 스크립트로 제작된 프로그램 등이다.
- 2) 플러그인(Plug-in): 웹 콘텐츠 내에 삽입되는 별도의 프로그램을 의미한다. 예를 들어, 플래시(Flash), 플렉스(Flex), 실버라이트(Microsoft Silverlight) 등이 이에 해당 한다.

4.2.1 (웹 애플리케이션 접근성 준수) 콘텐츠에 포함된 웹 애플리케이션은 접근성이 있어야 한다.

검사 항목

웹 콘텐츠에 포함된 부가 애플리케이션 또는 웹 페이지의 기능을 실행하는 데 필요한 웹 애플리케이션은 웹 페이지를 사용하거나 접근하는 것을 방해하지 않아야 한다. 웹 애플리케이션은 본 지침에서 설명한 모든 지침들을 적용하여 제작하여야 한다.

- 1) 접근성 API 사용: 웹 애플리케이션은 운영체제에서 제공하는 접근성 API 기능을 사용하여 제작되어야 한다. 그렇지 않으면 보조 기기가 웹 애플리케이션의 접근성 기능을 지원하지 못하는 경우가 발생할 수 있다.
- 2) 사용자 API 사용: 웹 애플리케이션이 운영체제와 호환되지 않는 접근성 API 기능을 사용하는 경우, 웹 애플리케이션이 제공하는 기능의 명칭, 역할, 상태에 관한 정보가 보조 기기에게 제공될 수 있어야 한다. 그렇지 않으면 보조 기기가 웹 애플리케이션의 접근성 기능을 지원할 수 없게 된다.
- 3) 국내의 보조 기기로 접근이 불가능한 웹 애플리케이션은 가능한 한 사용하지 않는것이 좋으며, 꼭 사용해야 하는 경우에는 대체 수단을 제공해야 한다.

기대 효과

- 1) 웹 애플리케이션이 접근성을 제공할 경우 보조 기기가 웹 애플리케이션과 상호작용이 가능하므로 보조 기기 사용자가 웹 애플리케이션의 활용할 수 있다.
- 2) 웹 애플리케이션의 자체적인 접근성을 평가하는 방법으로 본 지침을 적용할 수 있게 되어 접근성을 준수하는, 특색 있는 웹 애플리케이션의 개발이 가능할 것이다.
- 3) 새로운 기술의 경우에 자체적인 접근성 제공방법이 개발되지 않았다고 하더라도 대체 수단을 제공할 수 있다면 적용이 가능하다.

6.10 웹접근성 알기

· 웹접근성 / 웹표준 / **Front-End**에 대해 알 수 있는 유용한 읽을 거리 및 참조(reference) 사이트

- 나라디자인 >> <http://naradesign.net/wp/category/css/>
- W3C 접근성 문서 번역 >> http://www.wah.or.kr/w3c_doc/index.asp
- 웹 접근성을 고려한 게시판 제작 기법. >> <http://naradesign.net/wp/2008/11/14/257/>
- html5 국제 표준 뉴스 기사 >> http://news.inews24.com/php/news_view.php?g_menu=020200&g_serial=859769
- HTML5 권고안으로 확정 레퍼런스(28 October 2014) >> <http://www.w3.org/TR/html5/>

6.11 웹표준을 지켜주는 태그 사용.

· 웹접근성/ 웹표준 태그

- document.all은 IE만 동작함 document.getElementById("demo"); 사용. 혹은 \$('demo'), \$('#demo')로 접근 할 수 있음.
- id는 페이지에 유니크 값임. 즉, 하나만 있어야 함.
- 특정 폰트를 굵게가 에서 으로 바뀌었습니다.
- 의미없는 wrapping을 하고 싶다면? 태그를 사용한다.
- 영역 구분을 하고 싶다면? <div></div> 태그를 사용한다.
- 문장 <p></p>
- 순서 없는 목록
- 순서 있는 목록
- 나는 빨간색태그는 사라졌습니다.
나는 올바른 빨간색 혹은 <strong style="font-weight:normal;color:red">나는 올바른 빨간색
- <table>로 데이터셋을 출력 하고 싶다면 <table role="presentation">
- 단일 문서의 모든 태그는 소문자 혹은 대문자로 통일
- onClick, onKeyPress와 같이 camelcase로 작성 하지 않는다.

Front-End

- 페이지내에 javascript 함수는 호출만, 함수의 몸체는 validation.js와 같이 외부로 빼기
- iframe을 사용하여야 하는 상황이 나오면 비동기 처리에 경우만, 출력(UI)부분에 iframe을 넣을시 다량의 오류 발생 ajax로 방법을 바꾸는것을 고려 한다.
- 유저에이전트에 따른 예외처리는 문제가 많습니다.
- RIA 웹 프레임워크는 관리자 단만
- 기능 구현시 ie8+~ie11 / chrome / safari / firefox 테스트가 되어야합니다.

99. 부록 - VMWare 설치 가이드.

- 기본적으로 VMWare를 설치하여 퍼블리싱 화면을 검토 하며, VMWare 설치는 구글링 하여 설치 한다.

VMWare를 설치 링크 - <https://www.google.co.kr/search?q=vmware+windows+7+%EC%84%A4%EC%B9%98>

그러나 VMWare를 설치하고 WindowsXp ie8 Windows7 ie8, ie9, ie10을 설치 하기까지의 시간이 많이 걸리므로
WW192.168.10.101WEncFileW01.에듀가족W26.원상필W22. vmware tool and imagesW 에 **대상을 복사하여 open하여 사용** 할 수 있다.

사용하고 있는 PC의 OS버전이 windows7 이라면 가상화 <http://utilu.com/IECollection/>를 사용 할 수 있으나 본 가이드에서는 다루지 않는다.

ie가 제공하는 개발자 도구에서의 브라우저 에뮬레이션은 신뢰 할 수 없으며, ieTester 및 기타 에뮬레이션 툴을 허용 하지 않는다.

99.1 애플리케이션의 설치

- VMWare Workstation 10 애플리케이션을 설치 한다.

VMware Workstation 10.0.1 Build 1379776 의 디렉터리에 VMware-workstation-full-10.0.1-1379776.exe 를 설치 한다.
설치를 마치면 아래와 같은 화면이 보이게 된다.

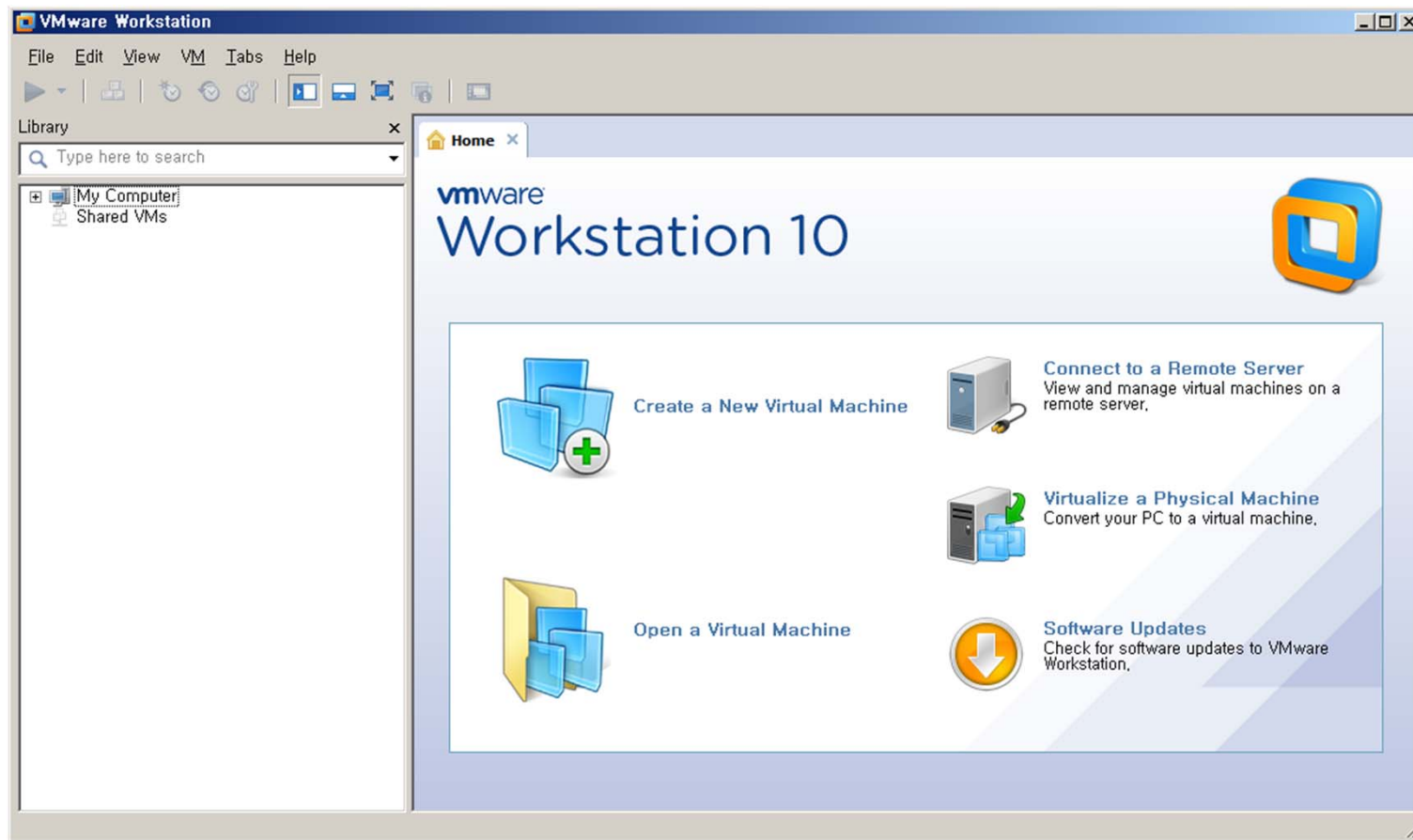


figure : vmware를 설치 한 모습.

99.2 VMWare 이미지 관련 파일의 복사

- **D:_applications\W 에 해당 디렉터리와 파일을 복사 한다.**

VM VMDK 디렉터리에는 가상화 이미지 포맷 표준인 *.vmdk 파일들이 위치하고
VMWare VMImages에는 VMWare 전용 Configuration파일인 *.vmx등이 위치 하고 있으므로,
VMWare VMImages 디렉터리의 내용을 import하여 사용 할 수 있고,
자신의 PC, 혹은 VMWare의 버전 환경에 따라 VM VMDK를 import하여 사용 할 수 있다.

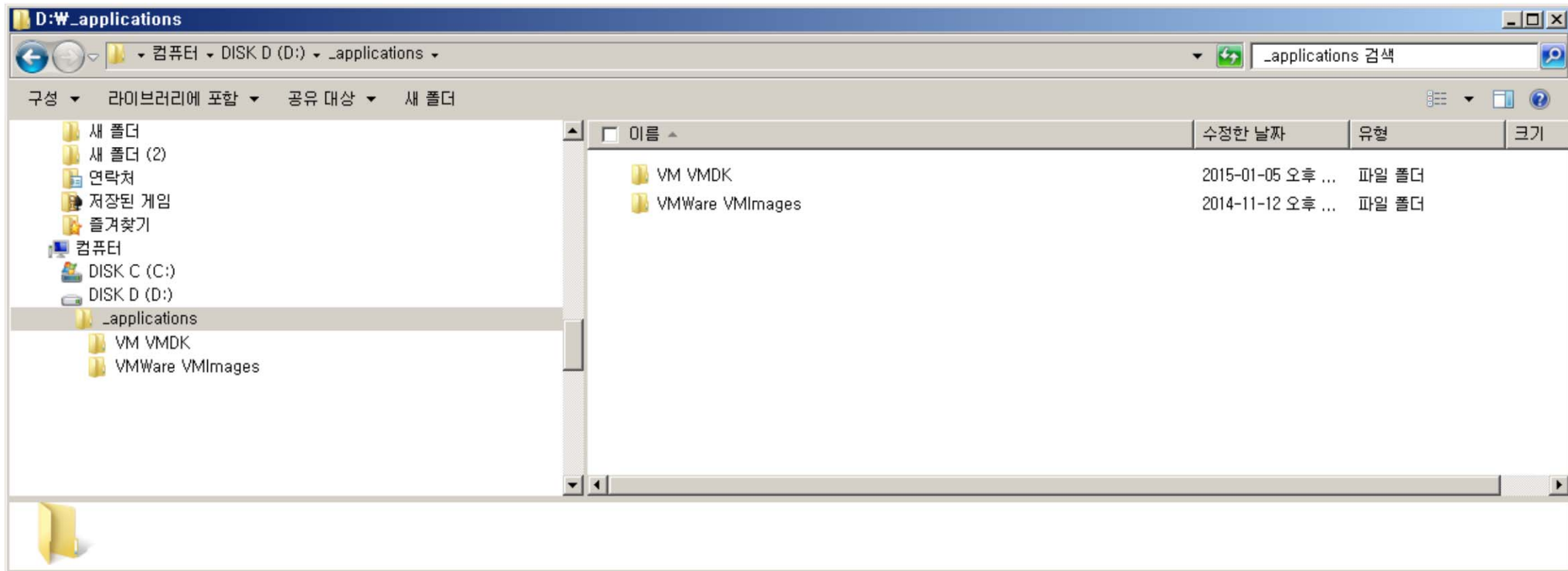


figure : d:_applications에 위치하여야 VMWare VMImages 디렉터리의 내용을 재 사용 할 수 있다.

99.3 Open a Virtual Machine

- VMWare에서 Open a Virtual Machine을 누른다.

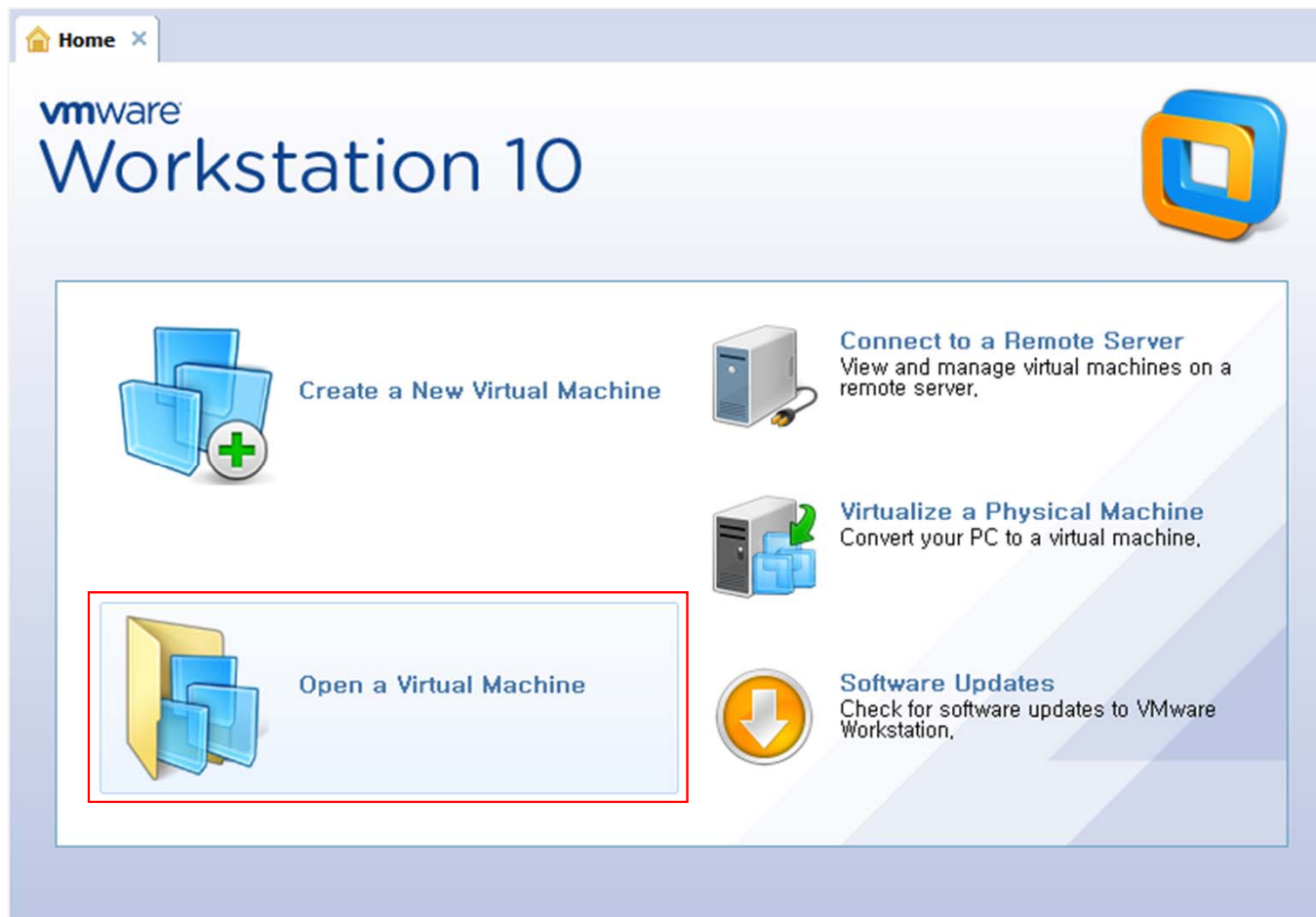


figure : 버튼을 누른다.

99.4 각각의 해당 하는 파일 열기

- d:\w_applications\VMWare VMImages\ 각 디렉터리에서 win7_ie10.vmx와 같은 파일을 선택한뒤 열기를 누른다.

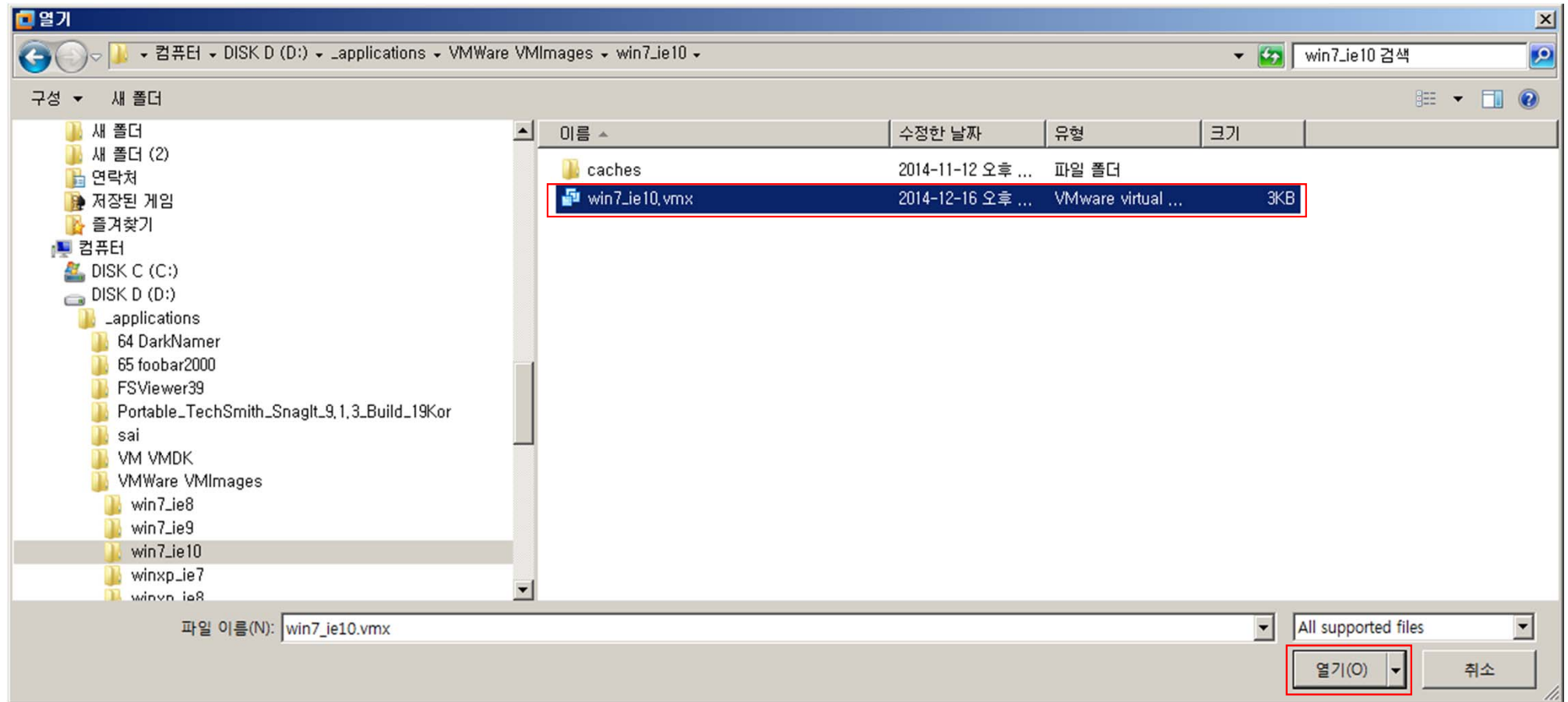


figure : vmx파일을 임포트 하자.

99.5 import를 마친 모습

- import를 마치고 Power on this virtual machine 버튼을 눌러 가상머신을 구동 할 수 있다.

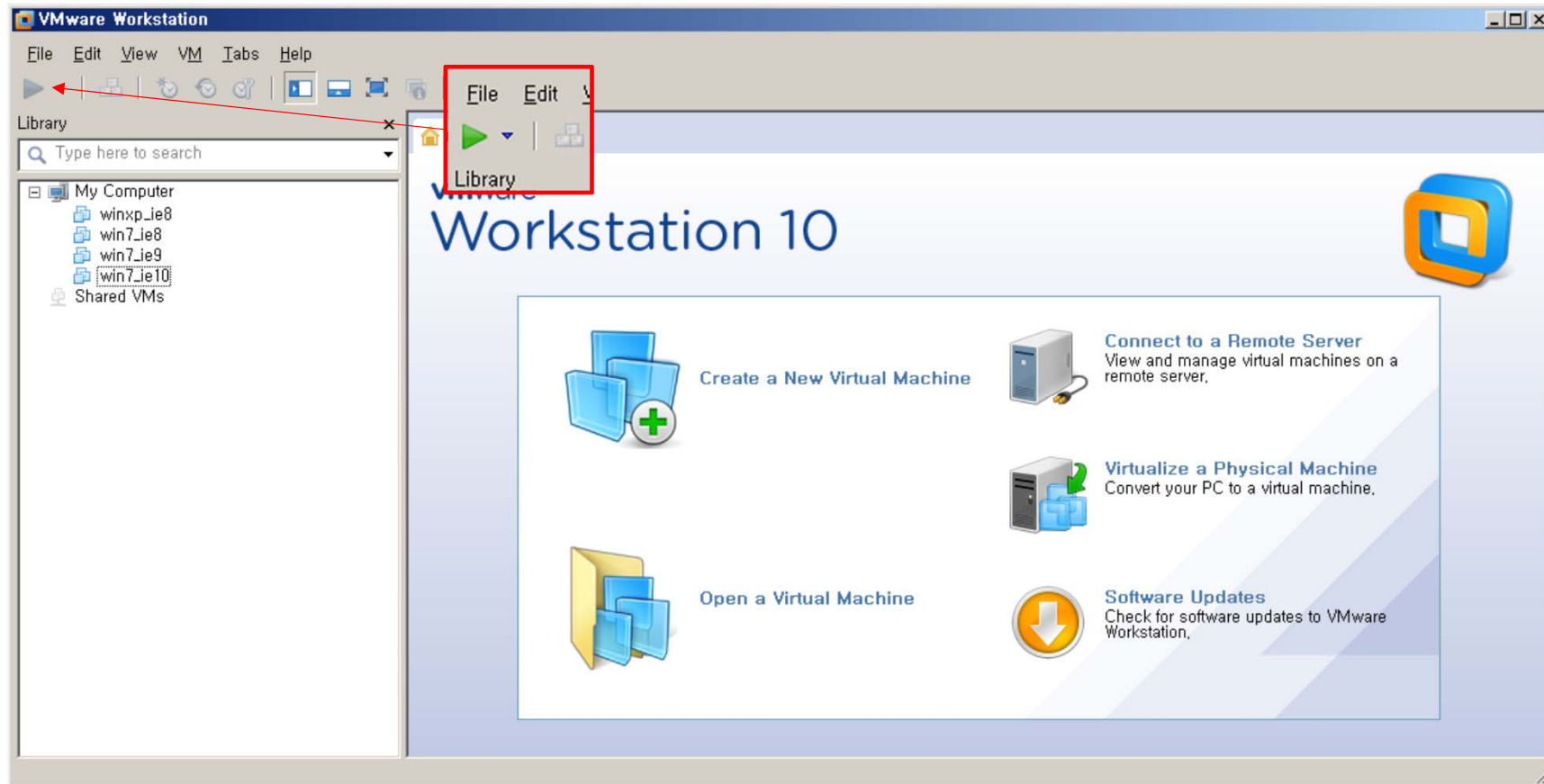


figure : 각각의 vmx 파일을 open하면 좌측에 목록이 나타나게 된다. 각각의 아이콘을 누르면 탭이 활성화 되고 시작 버튼을 눌러 가상머신을 구동 할 수 있다.

99.6.1 프로젝트 파일의 검사 방법 #1

- 해당 프로젝트 파일을 드래그 앤 드랍 하여 VMWare에 놓아 웹브라우저로 검사 하거나

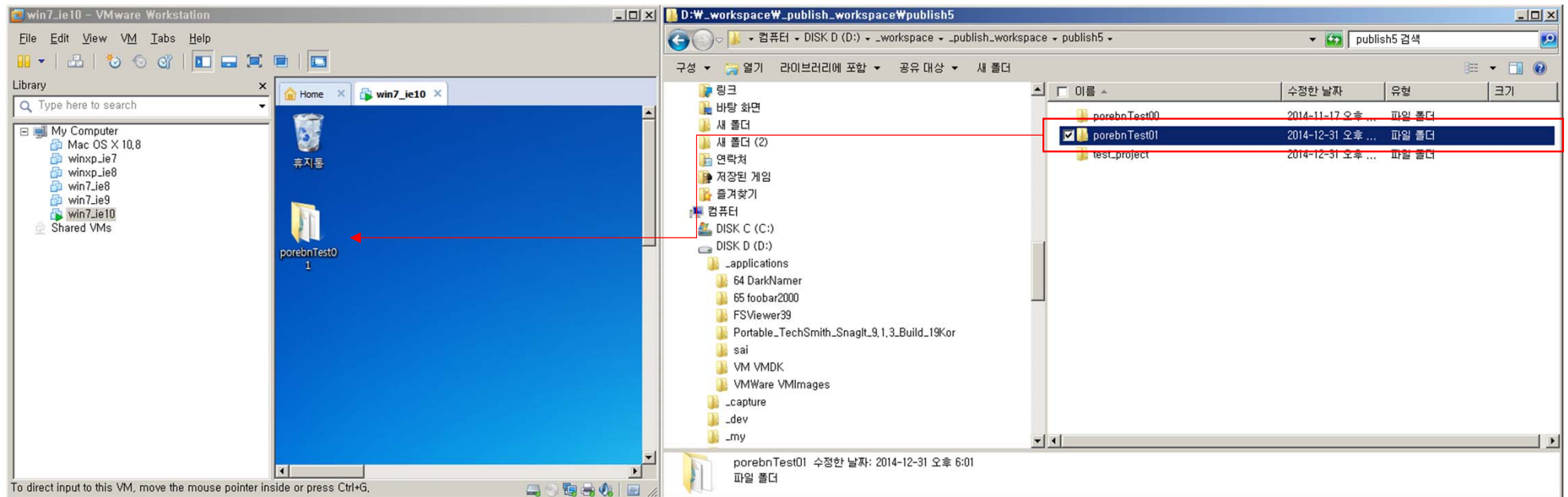


figure : 드래그 앤 드랍 하여 파일을 놓아 검사 할 수 있다.

99.6.2 프로젝트 파일의 검사 방법 #2

- 독자적인 로컬 호스트 서버를 구축하여 볼수도 있다.

filelist of porebn (2014-11-26~진행중)

published by Front-End Developer naiyumie (naiyumie@gmail.com)

■ 프로젝트 개요

- 해당 웹페이지의 페이지의 doctype은 html5로 결정함.
- html파일은 w3c validator와 K-WA4.0 등을 사용하여 정검 하며 주수를 100% 기본으로 한다.
- 이하 자세한 규칙은 (projectDir)/ref/rules의 내용을 따른다.

■ 특이사항

- html이 전체적으로 바뀌었습니다.

· file of contents [html]

no	파일명 (파일이름과 경로)	구분 (파일의 용도)	일자 (작성 및 변경일자)	주석 (기타 변경 내용 및 특이사항)
0	./html/act01_auth_approaching.html	인트로 페이지	• 2014-12-16	
1	./html/act01_auth_signin.html	로그인 페이지	• 2014-12-16	
2	./html/act02_main_front0.html			
3	./html/act02_main_front1.html			
4	./html/act02_main_front2.html			
5	./html/act02_main_front3.html			
6	./html/act02_main_front4.html	메인 레이아웃 페이지 (샘플)	• 2014-12-16 ~ 2014-12-18	
7	./html/act03_pop_01.html	03 위원회 - 팝업 - 교원검사 [열기]	• 2014-12-16 ~ 2014-12-18	
8	./html/act03_pop_02.html	03 위원회 - 팝업 - 전문가 Pool [열기]	• 2014-12-16 ~ 2014-12-18	
9	./html/act03_sub_01.html	03 위원회 - 서브 - 교육과정개편 위원회등록 #1	• 2014-12-16 ~ 2014-12-18	
10	./html/act03_sub_02.html	03 위원회 - 서브 - 교육과정개편 위원회등록 #2	• 2014-12-16 ~ 2014-12-18	
11	./html/act03_sub_03.html	03 위원회 - 서브 - 교육과정개편 위원회등록 #1	• 2014-12-16 ~ 2014-12-18	
12	./html/act03_sub_04.html	03 위원회 - 서브 - 교육과정개편 위원회등록 #2	• 2014-12-16 ~ 2014-12-18	
13	./html/act04_sub_01.html	04 직무분석 - 서브 - 직무분석	• 2014-12-23 ~ 2014-12-30	
14	./html/act04_sub_02.html	04 직무분석 - 서브 - 직무기술서 작성	• 2014-12-23 ~ 2014-12-30	
15	./html/act04_sub_03.html	04 직무분석 - 서브 - 직무분석 확장 #1	• 2014-12-23 ~ 2014-12-30	
16	./html/act04_sub_04.html	04 직무분석 - 서브 - 직무분석 확장 #2	• 2014-12-23 ~ 2014-12-30	

· file of contents [css]

no	파일명 (파일이름과 경로)	구분 (파일의 용도)	일자 (작성 및 변경일자)	주석 (기타 변경 내용 및 특이사항)
----	-------------------	----------------	-------------------	-------------------------

· file of contents [js]

figure : 가상머신상의 ie10에서 화면을 테스트 하는 모습.

figure : 로컬 컴퓨터의 ie11에서 화면을 테스트 하는 모습.