|  |
| --- |
| **EiLAB** |
| **HTML 표준가이드** |

|  |
| --- |
| **EiLAB Publishing Team**  2020-06-15 |

목차

[**1.** **환경설정** 5](#_Toc20841752)

[**1.1.** **Responsive Web** 5](#_Toc20841753)

[**1.1.1.** **환경정의** 5](#_Toc20841754)

[**1.1.2.** **폴더구조** 5](#_Toc20841755)

[**2.** **기본규칙** 7](#_Toc20841759)

[**2.1.** **파일관리** 7](#_Toc20841761)

[**2.1.1.** **기본제공파일** 7](#_Toc20841762)

[**2.2.** **네이밍** 8](#_Toc20841763)

[**2.2.1.** **네이밍 기본규칙** 8](#_Toc20841764)

[**2.2.2.** **HTML파일 네이밍** 8](#_Toc20841765)

[**2.2.3.** **CSS파일 네이밍** 9](#_Toc20841766)

[**2.2.4.** **JAVASCRIPT파일 네이밍** 9](#_Toc20841767)

[**2.2.5.** **이미지파일 네이밍** 9](#_Toc20841768)

[**2.2.6.** **네이밍 예약어** 9](#_Toc20841769)

[**2.3.** **주석** 11](#_Toc20841770)

[**2.3.1.** **기본규칙** 11](#_Toc20841771)

[**2.3.2.** **수정이력 관리** 11](#_Toc20841772)

[**3.** **HTML(Hyper Text Markup Language)** 15](#_Toc20841773)

[**3.1.** **기본규칙** 15](#_Toc20841774)

[**3.1.1.** **표기법** 15](#_Toc20841777)

[**3.1.2.** **따옴표 사용 범위** 15](#_Toc20841778)

[**3.1.3.** **들여쓰기** 15](#_Toc20841779)

[**3.1.4.** **Entity 코드 사용** 15](#_Toc20841780)

[**3.2.** **DTD** 16](#_Toc20841781)

[**3.2.1.** **PC DTD 기본형식** 16](#_Toc20841787)

[**3.2.2.** **Mobile App/Web DTD 기본형식** 16](#_Toc20841788)

[**3.3.** **HEAD** 16](#_Toc20841789)

[**3.3.1.** **META** 16](#_Toc20841796)

[**3.3.2.** **META VIEWPORT** 17](#_Toc20841797)

[**3.3.3.** **TITLE** 17](#_Toc20841798)

[**3.3.4.** **LINK** 17](#_Toc20841799)

[**3.3.5.** **STYLE** 17](#_Toc20841800)

[**3.3.6.** **SCRIPT** 18](#_Toc20841801)

[**3.4.** **레이아웃** 19](#_Toc20841802)

[**3.4.1.** **PC WEB** 19](#_Toc20841804)

[**3.4.2.** **Mobile App/Web** 20](#_Toc20841805)

[**4.** **CSS(Cascading Style Sheet)** 21](#_Toc20841806)

[**4.1.** **기본규칙** 21](#_Toc20841807)

[**4.1.1.** **CSS 파일 불러오기** 21](#_Toc20841808)

[**4.1.2.** **표기법** 21](#_Toc20841809)

[**4.1.3.** **따옴표 사용 범위** 21](#_Toc20841810)

[**4.1.4.** **들여쓰기** 22](#_Toc20841811)

[**4.1.5.** **공백** 22](#_Toc20841812)

[**4.1.6.** **줄바꿈** 22](#_Toc20841813)

[**4.2.** **속성** 23](#_Toc20841814)

[**4.2.1.** **선언순서** 23](#_Toc20841815)

[**5.** **JAVASCRIPT** 24](#_Toc20841816)

[**5.1.** **기본규칙** 24](#_Toc20841817)

[**5.1.1.** **Script 파일 불러오기** 24](#_Toc20841818)

[**5.1.2.** **표기법** 24](#_Toc20841819)

[**5.1.3.** **들여쓰기** 24](#_Toc20841820)

[**5.1.4.** **변수** 25](#_Toc20841821)

[**5.1.5.** **함수** 25](#_Toc20841822)

INTRO

이 문서는 EiLAB에서 제공하는 HTML표준가이드로 마크업 개발자가 소스 코드 작성 시에 따라야 할 규칙 및 마크업 개발과 관련된 환경 정의를 기술한다.

구현된 화면과 화면목록, 컴포넌트 가이드, 퍼블리싱 소스는 한 화면으로 확인 가능한 홈페이지 형식으로 제공되어있다.

# **환경설정**

## **Responsive Web**

### **환경정의**

|  |  |
| --- | --- |
| 범주 | 설명 |
| OS | Mobile – iOS11 이상, Android 7.0 이상  PC – Window 8.1 이상, Mac |
| Browser | Internet Explorer 11 이상, IE EDGE, 최신 모던 브라우저(Firefox, Chrome, Safari, Whale 등) |
| 해상도 | PC와 Tablet은 동일한 해상도로 구현(최저해상도 769px 기준)  Mobile은 구현(최고해상도 768px 기준) |
| 웹표준 | Web 표준 준수(W3C) \* 웹접근성, 장애인차별금지법을 통한 마크획득은 프로젝트 진행상황에 따라 시행 |
| Text Editor | 무관 |
| Encoding | UTF-8 |

### **폴더구조**

* 폴더 경로는 절대경로가 아닌 **상대경로**로 되어있다.

**\*경로는 프로젝트 환경 마다 달라질 수 있다.**

* HTML 파일**(\*.htm, \*.html)**은 html 폴더에 위치한다.
* 이미지 파일**(\*.jpg, \*.gif, \*.png)**은 resource/image 폴더 하위에 이미지 용도 별로 분류된 폴더에 위치한다.
* 공통 스크립트**(\*.js)**는 resource/js 폴더에 위치한다.
* 스타일시트**(\*.css)**는 resource/css 폴더에 위치한다.
* Web Font 파일은 resource/font 폴더에 위치한다.

|  |
| --- |
| **Responsive Web.**  r/  ├─── html/  │ └─── page/  └─── resource/  ├─── css/  ├─── font/  ├─── image/  └─── js/ |

# **기본규칙**



## **파일관리**

하기 사유에 의거 **html파일 및 ’** [**2.1.1. 기본제공파일**](#_기본제공파일) **(css, js)’ 의 파일은 EiLAB에서 관리 한다.**

1. 요청사항 및 퍼블리싱 수정이력을 html, css, js에 직접 기재하여 관리함.
2. 수정 파일을 경로에 맞춰 덮어씌우는 방식으로 파일을 전달함.
3. 이중관리 시 발생할 수 있는 소스의 충돌, 소실, 중복 등의 위험을 방지하기 위함.

상기 사유에 따라 소스 수정이 필요할 경우 추가 파일을 생성하여 관리하며 추가파일의 네이밍은   
[**2.2. 네이밍**의](#_네이밍) 규칙에 따른다.

### **기본제공파일**

**기본제공파일 CSS**

|  |  |
| --- | --- |
| 파일명 | 설명 |
| ui\_layout.css | layout 구성 css파일. |
| ui\_common.css | 공통요소 ( reset, component, web font 등 ) css파일. |
| ui\_contents.css | 페이지내용 구성 css파일. |
| ui\_plugin.css | framework, plugin 모음 css파일. |
| ui\_responsive.css | 반응형 구성 css파일 |
|  |  |

**기본제공파일 JS**

|  |  |
| --- | --- |
| 파일명 | 설명 |
| ui\_common\_head.js | 함수 선언 javascript파일. |
| ui\_common\_body.js | 함수 실행 javascript파일. |
| ui\_plugin.js | framework, plugin 모음 javascript파일. |
|  |  |

## **네이밍**

### **네이밍 기본규칙**

* 영문 소문자, 숫자, 언더스코어( \_ )로 작성한다.
* 영문대문자, 숫자로 시작하지 않는다. (html파일 예외)
* 예약어가 있는 경우 예약어를 사용한다.
* 명확하고 직관적인 단어를 선택한다.
* 언더스코어( \_ )는 2개 이상의 단어를 조합할 경우에 사용한다.
* 단어조합은 3단계 이하를 권장하나 경우에 따라 그 이상도 허용한다.  
  **Ex> btn\_accordion\_close\_on.png**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Example.**   |  |  |  | | --- | --- | --- | | 기본형 | 잘못된 예 | 올바른 예 | | section | sectionList, section\_1, section\_2 | section\_list, section1, section2, section3 | | no | - | no1, no2, no3, …, no10 | |

### **HTML파일 네이밍**

**스토리보드의 화면ID를 최우선 한다.**   
스토리보드의 화면ID를 알 수 없는 경우 아래의 우선순위에 따라 네이밍한다.

1. 프로젝트에서 요구하는 특정한 규칙.
2. IA문서의 category와 depth.
3. 폴더명과 화면 흐름 상의 순서
4. 파일의 특징, 분류, 성격.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Example.**   |  |  | | --- | --- | | 예제 | 설명 | | Smart\_Ready\_0100.html SmartReady0100.html | **스토리보드의 화면 ID** : Smart Ready 0100 | | CO\_0101.html CO010100.html | **IA문서** : 공통 > 1depth 첫 번째 > 2depth 첫 번째 화면 | | PAY\_process\_0102.html PAY\_process0102.html | **폴더명과 화면 흐름 상의 순서** : 결제페이지 세 번째 스텝 | | layer\_popup.html | **파일의 분류, 특징, 성격** | |

### **CSS파일 네이밍**

* **2.2.1.**의 네이밍 기본규칙에 의거하여 네이밍한다.
* 기능, 업무, 서비스 명을 따른다.   
  **Ex> payment.css service.css …**

### **JAVASCRIPT파일 네이밍**

* **2.2.1.**의 네이밍 기본규칙에 의거하여 네이밍한다.
* 기능, 업무, 서비스 명을 따른다.  
  **Ex> main.js, service.js …**

### **이미지파일 네이밍**

* ‘**분류**\_(형태 or 의미 or 기능 …)\_상태’ 의 흐름으로 조합한다.  
  분류는 필수이며 그 이 후 반복 및 생략이 가능하다. (예제 참고 必)
* **2.2.6.** 네이밍 예약어 규칙에 근거하여 작성한다.
* 같은 분류의 이미지가 두 개 이상일 시 파일명 마지막에 타입 및 숫자 정보를 추가하여 구분한다.
* 임시 이미지에 한해 @네이밍을 사용한다. **Ex> @temp\_header.gif**

|  |  |
| --- | --- |
| 예제 | 설명 |
| input\_checkbox.png input\_checkbox\_checked.png | 분류\_분류.png 분류\_분류\_상태.png |
| list\_arrow\_right.png | 분류\_형태\_형태.png |
| btn\_popup\_close.png | 분류\_분류\_기능.png |
| icon\_alert.png | 분류\_의미.png |
| btn\_search\_mail.png | 단어조합은 3단계 이하를 권장한다. |
| @temp\_header.gif | 임시 이미지에 한해 @네이밍을 사용한다. |

### **네이밍 예약어**

* 디자인과 기능(css, javascript)을 분리하여 사용한다.
* 기능을 구현할 때 **추가어트리뷰트**를 사용하여 디자인과 기능을 분리하는 것을 권장한다.   
  **Ex> <div class=”accordion active” data-accordion=”accordion\_list”></div>**

#### 디자인 예약어

|  |  |  |  |
| --- | --- | --- | --- |
| 예약어 | 설명 | 예약어 | 설명 |
| nav | navigation | btn | 버튼 |
| gnb | global navigation bar | desc | 설명 |
| lnb | local navigation bar | msg | 메세지 |
| bg | 배경 | icon | 아이콘 |
| on | 켜짐 | arrow | 화살표 |
| off | 꺼짐 | up | 상 |
| empty | 없음 | down | 하 |
|  |  | left | 좌 |
|  |  | right | 우 |
|  |  | next | 다음 |
|  |  | prev | 이전 |

#### 기능 예약어

|  |  |  |  |
| --- | --- | --- | --- |
| 예약어 | 설명 | 예약어 | 설명 |
| active | 활성화 |  |  |
|  |  |  |  |

## **주석**

### **기본규칙**

* 주석 기호와 주석 내용 사이에는 공백 한 칸이 있어야 한다.
* 종료 주석에는 날짜만 기재하며 ‘E :’ 기호로 종료임을 표시한다.

### **수정이력 관리**

* ’날짜, 작업자, 요청자/상세내용’ 순으로 작성.
* HTML문서의 경우 수정이력을 문서에 직접 기재하여 관리한다.
* CSS, JAVASCRIPT의 수정이력은 필요한 경우에만 작성한다.

#### HTML

|  |
| --- |
| **Example.**  <!DOCTYPE html>  <!--  \* @ Copyright (c) EiLab  \* @ 페이지명 : 레이아웃  \* @ 파일명 : html/\_layout/layout/layout01.html  \* @ 수정이력  \* ===================================================  \* 날짜 작업자 요청자/상세내용  \* -------------------------------------------------------------------------------------  \* 20180730 서진석 최초작성  \* 20180815 서진석 ㅁㅁ은행 이경민대리 요청. 상세내용  \*===================================================  -->  <html lang="ko">  <!— S : 20180730 서진석 ㅁㅁ은행 이경민대리 요청/ 상세내용 -->  <div>  …  </div>  <!—E : 20180730 -->  <img /><!-- 20180730 서진석 ㅁㅁ은행 이경민대리 요청 / 상세내용 --> |

#### CSS, JAVASCRIPT 주석 규칙 공통사항

* CSS, JAVASCRIPT의 주석규칙은 아래 표를 기본으로 한다.

|  |  |
| --- | --- |
| 대 분류간 빈 줄 | 3줄 |
| 중분류간 빈 줄 | 2줄 |
| 소분류간 빈 줄 | 1줄 |

|  |
| --- |
| **Example.**  /\*=================================================  대 분류  =================================================\*/  √ 3줄: 대 분류간 빈 줄  √  √  /\*=================================================  대 분류  =================================================\*/  /\*--------------------------------------------------------------------  대 분류 - 중분류  --------------------------------------------------------------------\*/  √ 2줄: 중분류간 빈 줄  √  /\*--------------------------------------------------------------------  대 분류 - 중분류  --------------------------------------------------------------------\*/  /\* 대 분류 - 중분류 - 소분류 \*/  div.empty { display: none; }  √ 1줄: 소분류간 빈 줄  /\* 대 분류 – 중분류 - 소분류 \*/ |

#### CSS

* 대 분류(=), 중분류(-), 소분류(한줄주석) 순으로 분류한다.  
  소스가 길어질 경우를 대비하여 이전 분류를 항상 기재한다.
* 한 줄 이상일 경우 그룹화하여 표시, 한 줄일 경우 해당 선언 닫는 괄호( } ) 다음에 표시하며.  
  시작과 종료 주석 내용을 동일하게 한다. (하단 표 참고)

|  |
| --- |
| **Example.**  /\*=================================================  form(대 분류)  =================================================\*/  /\*--------------------------------------------------------------------  form(대 분류) – reset(중분류)  --------------------------------------------------------------------\*/  /\* form(대 분류) – reset(중분류)– input(소분류) \*/  /\* S : 2018-09-18 서진석 ㅁㅁ은행 이경민 대리 요청/ 삭제 (추후재사용 대비 임시주석처리) \*/  .temp .box { border: 1px solid #000; }  .temp .box { border: 1px solid #000; }  .temp .box { border: 1px solid #000; }  .temp .box { border: 1px solid #000; }  /\* E : 2018-09-18 서진석 ㅁㅁ은행 이경민 대리 요청/ 삭제 (추후재사용 대비 임시주석처리) \*/ |

#### JAVASCRIPT

* 아래 Example. 의 기준으로 소스를 그룹화하며 그 외의 경우 javascript 기본주석( // )를 기본으로 사용한다.
* category (/) - 대 분류(\*) - 중분류(-), 소분류(한줄주석) 순으로 분류한다.

|  |
| --- |
| **Example.**  ////////////////////////////////////////////////////////////////////////////////////////////////////  ////// 함수정의 (category)  ////////////////////////////////////////////////////////////////////////////////////////////////////  /\*=================================================  대 분류 (grouping)  =================================================\*/  /\*--------------------------------------------------------------------  # 중분류 (function설명)  √  \* 함수설명 (필요 시에만 기재)  --------------------------------------------------------------------\*/  √ 2줄: 중분류간 빈 줄  √  /\*--------------------------------------------------------------------  # totalCheckbox(type) 체크박스 전체 선택  --------------------------------------------------------------------\*/  var totalCheckbox = function(type){  …중략…  // if 문 예시  if(true){ // 열기  …….  }else{ // 닫기  …….  }  √ 1줄: 소분류간 빈 줄  // # 문구삭제 버튼 추가 ( 함수 내 함수 정의 )  // +  // + 부가설명 ( 필요 시에만 기재)  // +  var btnTextErase = function(){  …중략…  /\*  \* 2018-07-30 임시 주석 처리  function callBackFunc(){  }  $(\_selector).click(function(){  callBackFunc();  });  \*/  **√ 3줄: 대 분류간 빈 줄**  **√**  **√**  /\*=================================================  대 분류  =================================================\*/ |

# **HTML(Hyper Text Markup Language)**

## **기본규칙**

* HTML은 해당 **DTD의 명세에 맞게 작성**하며, **W3C validation을 통과해야 한다.**



### **표기법**

* DTD를 제외한 모든 element와 attribute는 영문 소문자로 작성한다.

|  |  |
| --- | --- |
| 잘못된 예 | 올바른 예 |
| <SPAN Class=”msg”>간단한 설명</SPAN> | <span class=”msg”>간단한 설명</span> |

### **따옴표 사용 범위**

* Attribute값은 **큰따옴표(“ “)**로 묶어야 하며, 값은 항상 선언되어야 한다.

|  |  |
| --- | --- |
| 잘못된 예 | 올바른 예 |
| <img src=’text.gif’ width=’100’  alt=’테스트’ /> | <img src=”text.gif” width=”100” alt=”테스트” /> |
| <option value=”010” selected>010  </option> | <option value=”010” selected=”selected”>010  </option> |

### **들여쓰기**

* 들여쓰기 1탭 들여 쓰고, 1탭의 크기는 공백 4칸으로 설정한다. 문서 내에서 반드시 탭을 이용하여 들여쓰기를 하며, 탭을 대신하여 공백으로 들여 쓰지 않는다.

### **Entity 코드 사용**

* 특수기호는 entity name을 사용하여 entity 코드로 변환한다.
* <, >, “, ‘, &등의 특수기호를 entity 코드로 변환하지 않으면, 브라우저가 이를 시작/종료 element나 attribute로 잘못 해석할 수 있다.
* Entity코드는 ISO-8859-1을 기준으로 하며, 다음 경로에서 확인할 수 있다.  
  ( <http://en.wikipedia.org/wiki/List_of_XML_and_HTML_character_entity_references> )

|  |  |
| --- | --- |
| 잘못된 예 | 올바른 예 |
| <span>Bread & Butter</span> | <span>Bread &amp; Butter</span> |

## **DTD**

* 웹 페이지의 문서 종류를 알려주는 선언문으로, HTML 문서의 최 상단에 선언한다.
* **DTD는 프로젝트 환경마다 달라질 수 있다.**



### **Responsive Web DTD 기본형식**

* HTML5 DTD를 기본으로 사용한다.
* 구형 브라우저(Ie8, IE9, IE10) 환경에 최적화를 해야 할 경우 XHTML 1.0 Transitional DTD를 기본으로 사용한다.   
  단, 이와 같은 경우 html5 신규 테그와 , animation과 같은 css3 기술을 사용할 수 없게 된다.

|  |
| --- |
| **Example.**  <!DOCTYPE html> |

## **HEAD**

<meta>, <title>, <link>, <style>, <script>순으로 나열한다.



### **META**

* 문서의 기본 encoding 및 viewport등을 선언한다.
* Encoding 설정은 DB의 인코딩 방식과도 관련이 있으므로 반드시 담당 개발자와 협의하여 정해야 한다. **특이 사항 없는 경우 utf-8을 선언**한다.
* DTD에 알맞은 인코딩 설정을 선언한다.

|  |
| --- |
| **Example.**  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  <meta encoding="utf-8" /> |

### **META VIEWPORT**

* 개발과 관련하여 특별한 사항이 없는 경우 device-width와 사용자 확대 불가로 선언한다.

|  |
| --- |
| **Example.**  <meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0, user-scalable=no" /> |

### **TITLE**

* 현재 페이지의 제목을 입력한다.
* 특별한 사항이 없는 경우, 서비스명 혹은 브랜드 명은 가장 마지막에 작성한다.

|  |
| --- |
| **Example.**  <title>회사소개 - 이아이랩</title> |

### **LINK**

* rel, type, href 순서로 attribute를 선언한다.
* HTML5 DTD일 경우, type은 생략 가능하다.

|  |
| --- |
| **Example.**  <link rel="stylesheet" type="text/css" href="css/default.css" /> |

### **STYLE**

* type attribute를 선언한다.
* HTML5 DTD일 경우, type은 생략 가능하다.
* 하위에 선언되는 style은 1depth 들여 쓴다.

|  |
| --- |
| **Example.**  <style type="text/css">  body { ... }  </style> |

### **SCRIPT**

* type, src 순서로 attribute를 선언한다.
* HTML5 DTD일 경우, type은 생략 가능하다.
* 하위에 선언되는 script는 1depth 들여 쓴다.
* Script 선언 위치는 <head> 내에 선언하는 것을 원칙으로 하되, 성능 상의 이슈 및 특별한 사항이 있는 경우 선언 위치를 변경할 수 있다.

|  |
| --- |
| **Example.**  <script type="text/javascript" src="js/base.js"></script>  <script type="text/javascript">  var d = document;  </script> |

## **레이아웃**

다음은 HTML의 기본 레이아웃 구조를 나타낸 것이다.  
**\* 레이아웃 구조는 프로젝트마다 달라질 수 있다.**

\* 레이아웃에 style은 class에만 선언한다. (Id에 style을 선언하지 않는다.)



### **Responsive WEB**

|  |
| --- |
| **Example.**  id=“layout” class=”layout”  id=“header” class=”header”  class=“header\_wrap”  id=“footer” class=”footer”  class=“footer\_wrap”  id=“container” class=”container”  class=“container\_wrap”  id=”lnb” class=“lnb”  class=“contents\_wrap”  class=“contents” |

#### 레이아웃 예약어

Responsive Web 레이아웃에는 다음 표에 예약된 id 와 class를 기초로 한다.

|  |  |
| --- | --- |
| 예약어 | 범위 |
| #layout.layout | 페이지 전체 영역 |
| #header.heaer | 머리글 영역 |
| .header\_wrap | 머리글 영역(안쪽) |
| #container.container | 본문 영역 |
| .container\_wrap | 본문 영역(안쪽) |
| .contents | 주요 콘텐츠 영역 |
| #footer.footer | 바닥글 영역 |
| .footer\_wrap | 바닥글 영역(안쪽) |

# **CSS(Cascading Style Sheet)**

## **기본규칙**

* CSS는 사용 가능한 Hack과 CSS3를 제외하고 W3C validation을 통과해야 한다
* CSS 파일에서 한글이 제대로 표기 되지 않을 경우를 대비하여 css파일 첫 줄에 html문서와 동일한 인코딩을 선언한다.   
  **Ex> @charset "utf-8";**

### **CSS 파일 불러오기**

* CSS 데이터를 선언한 순서대로 불러오기 위하여 link 방식을 사용한다.  
  ( @import 방식은 사용하지 않는다. )
* CSS 데이터는 ui\_layout - ui\_common - ui\_content – ui\_responsive - etc 순으로 선언한다.

|  |
| --- |
| **Example.**  <!-- Publish CSS -->  <link rel="stylesheet" type="text/css" href="경로/ui\_layout.css" />  <link rel="stylesheet" type="text/css" href="경로/ui\_common.css" />  <link rel="stylesheet" type="text/css" href="경로/ui\_content.css" />  <link rel="stylesheet" type="text/css" href="경로/ui\_responsive.css" /> |

### **표기법**

* 모든 속성은 영문 소문자로 작성한다.

|  |  |
| --- | --- |
| 잘못된 예 | 올바른 예 |
| .class { Font-Family: Arial; } | .class { font-family: Arial; } |

### **따옴표 사용 범위**

* @charset 선언 시에는 속성값을 큰따옴표(" ")로 감싼다.
* url, 공백이 포함된 폰트명, 문자열(string)데이터 타입은 작은따옴표(' ')를 사용한다.
* url 입력 시 작은따옴표(‘ ‘) 사용을 권장한다.  
  **Ex> background: url(‘../../image/content/@temp\_header.png’);**
* 그 외의 경우에는 따옴표를 사용하지 않는다.

|  |
| --- |
| **Example.**  @charset "utf-8";  span.property { font-family: '맑은 고딕', 'consolas’; background: url(‘../image/content/aa.png’)…; }  span.property:after { content: ‘’; display: block; clear: left; } |

### **들여쓰기**

* CSS 코드를 작성할 때는 들여쓰기를 하지 않는다.   
  **단, 미디어쿼리나 애니메이션과 같이 중괄호{}가 중첩되는 경우는 예외로 한다.**

|  |
| --- |
| **Example.**  @media all and (min-width: 480px){  #contents { width:100% }  } |

### **공백**

* 속성(Property)과 콜론(colon / : ) 사이의 공백, 값(value)과 세미콜론(semicolon / ; ) 사이의 공백, 값(value)과 !important 사이의 공백은 **한 칸만 띄고, 그 이상은 하지 않는다.**

|  |
| --- |
| **Example.** span.property**√**{**√**display:**√**block;**√**rgba(255,**√**255,**√**255,**√**0.1);**√**} span.property { display: block; rgba(255, 255, 255, 0.1); } |

### **줄바꿈**

* **소스 가독성을 위해 대괄호{ } 안 코드는 줄 바꿈을 하지 않는다.**단, @keyframe과 같이 **한 줄로 썼을 시** **가독성이 떨어지는 경우 줄 바꿈을 허용한다.**

|  |
| --- |
| **Example.**  span.property { display: block; rgba(255, 255, 255, 0.1); }  span.property { animation: movent 1000ms forwards ease-in-out; }  @keyframes movent {  0% { transform: translateX(20px); opacity: 0; }  100% { transform: translateX(0); opacity: 1; }  } |

## **속성**

### **선언순서**

* class 충돌을 방지하기 위하여 tag와 선택자를 함께 기재하여 명확히 구분한다.  
  **단, 버튼 과 같이 여러 tag에 범용적으로 사용될 가능성이 있는 class는 제외.  
  Ex> div.property, .eb\_btn**
* 속성 선언 순서는 **표시-위치-사이즈-컨텐츠-부가속성**으로 작성 한다. (아래 표 참고)

|  |  |
| --- | --- |
| 범주 | 속성 |
| (우선순위) | z-index |
| 표시 | display |
|  | opacity, overflow, float |
| 위치 | position |
|  | top, right, bottom, left, margin(위치로 이용할 경우) |
| 사이즈 | width, height, line-height (크기로 이용할 경우) |
|  | margin, padding |
|  | border |
|  | box-sizing, border-radius, box-shadow |
| (정렬) | vertical-align, text-align |
| 컨텐츠 | font-size, line-height, color … |
| 부가속성 | background, cursor, appearance … |

# **JAVASCRIPT**

## **기본규칙**

### **Script 파일 불러오기**

* core.js(jquery, jquery ui … ) - ui\_common\_head.js - ui\_plugin.js - ui\_etc.js 순으로 선언한다.
* ui\_common\_head.js는 함수를 정의하는 js이며 <head>에 위치한다.
* ui\_common\_body.js는 함수를 실행하는 js이며 <body> 최 하단에 위치한다.

|  |
| --- |
| **Example.**  ……중략……  <!-- Publish JS -->  <script type="text/javascript" src="경로/jquery-2.1.1.min.js"></script>  <script type="text/javascript" src="경로/jquery-ui-1.11.4.js"></script>  <script type="text/javascript" src="경로/ui\_common\_head.js"></script>  <script type="text/javascript" src="경로/ui\_plugin.js"></script>  ……중략……  <div id=”layout” class=”layout”>  ……중략……  </div>  <!-- Publish JS -->  <script type="text/javascript" src="../../../resource/js/ui\_common\_body.js"></script>  </body>  </html> |

### **표기법**

* 영문 소문자를 기본으로 사용한다.
* 함수명 및 변수명 은 카멜표기법을 사용한다.\*카멜표기법: 영문 소문자를 기본으로 하여 단어와 단어 사이를 대문자로 구분 하는 방식  
  **Ex> popupOpen, windowHeight, boxControllerOpen …**

### **들여쓰기**

* 들여쓰기 1탭 들여 쓰고, 1탭의 크기는 공백 4칸으로 설정한다. 문서 내에서 반드시 탭을 이용하여 들여쓰기를 하며, 탭을 대신하여 공백으로 띄어 들여 쓰지 않는다.

### **변수**

* 변수명의 처음 시작부분에 변수타입에 해당하는 문자열을 기입한다. **(하단 표 참조)**
* 변수타입 문자열 이후 변수명 표기는 카멜표기법을 기준으로 한다
* 변수명에는 특수문자를 사용하지 않는다.
* 범주가 같은 것끼리는 그룹화하는 것을 권장한다.

|  |  |
| --- | --- |
| 문자열 | 설명 |
| $ | Jquery 셀랙터값 변수명 표기시 |
| Num | 데이터 타입이 ‘number’인 변수명 표기시 |
| Str | 데이터 타입이 ‘string’인 변수명 표기시 |
| Boo | 데이터 타입이 ‘boolean’인 변수명 표기시 |
| Obj | 데이터 타입이 ‘object’인 변수명 표기시 |
| Arr | 데이터 타입이 ‘array’인 변수명 표기시 |

|  |
| --- |
| **Example.**  // variable define  var $this = $(this);  var NumThisH = $this.outerHeight();  var StrThisTxt = $this.text();  var BooThisState = $this.hasClass(‘active’);  var ObjThisProperty = { type: ‘string’, length: 1 };  var ArrThisGroup = [ $this, $thisParent, $thisChildren];  // height  var NumLayoutWidth = $layout.width(),  NumLayoutHeight = $layout.height(); |

### **함수**

* 업무적인 특성 혹은 적용해야 하는 함수의 의미를 고려하여 적절한 함수명을 정의한다. ( ex. calendarDate )