|  |
| --- |
| **EiLAB** |
| **HTML개발표준가이드** |

|  |
| --- |
| **EiLAB Publishing Team**  **2023-11-27** |

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| 버전 | 일자 | 수정사항 | 비고 |
| 0.1 | 2017-07-28 | 초안작성 | 어준석, 김소현 |
| 0.2 | 2017-11-02 | 모바일관련 내용추가 및 정리 | 김소현, 윤선정 |
| 1.0 | 2017-11-07 | 검수완료 | 강민수 |
| 2.0 | 2019-01-07 | 전 항목 갱신 | 서진석 |
| 2.1 | 2022-10-14 | 전 항목 갱신 및 SCSS항목 추가 | 송서현, 이준성 |
| 2.2 | 2023-11-27 | - Mobile First 기반 목차순서 변경  - 환경설정 최신화  - SCSS항목 [별첨] 분리 | 서진석 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

목차

[**1.** **환경설정** 6](#_Toc152600143)

[**1.1.** **Mobile App/Web** 6](#_Toc152600144)

[**1.1.1.** **환경정의** 6](#_Toc152600145)

[**1.1.2.** **폴더구조** 6](#_Toc152600146)

[**1.2.** **PC Web** 7](#_Toc152600147)

[**1.2.1.** **환경정의** 7](#_Toc152600148)

[**1.2.2.** **폴더구조** 7](#_Toc152600149)

[**2.** **기본규칙** 8](#_Toc152600150)

[**2.1.** **파일관리** 8](#_Toc152600152)

[**2.1.1.** **기본제공파일** 8](#_Toc152600153)

[**2.2.** **파일 네이밍** 8](#_Toc152600154)

[**2.2.1.** **HTML파일 네이밍** 8](#_Toc152600155)

[**2.2.2.** **CSS파일 네이밍** 9](#_Toc152600156)

[**2.2.3.** **JAVASCRIPT파일 네이밍** 9](#_Toc152600157)

[**2.2.4.** **이미지파일 네이밍** 10](#_Toc152600158)

[**2.3.** **주석** 10](#_Toc152600159)

[**2.3.1.** **기본규칙** 10](#_Toc152600160)

[**3.** **HTML(Hyper Text Markup Language)** 14](#_Toc152600161)

[**3.1.** **기본규칙** 14](#_Toc152600162)

[**3.1.1.** **표기법** 14](#_Toc152600165)

[**3.1.2.** **따옴표 사용 범위** 14](#_Toc152600166)

[**3.1.3.** **들여쓰기** 14](#_Toc152600167)

[**3.1.4.** **Entity 코드 사용** 14](#_Toc152600168)

[**3.2.** **DTD** 15](#_Toc152600169)

[**3.2.1.** **Mobile App/Web DTD 기본형식** 15](#_Toc152600175)

[**3.2.2.** **PC DTD 기본형식** 15](#_Toc152600176)

[**3.3.** **HEAD** 15](#_Toc152600177)

[**3.3.1.** **META** 15](#_Toc152600184)

[**3.3.2.** **META VIEWPORT** 16](#_Toc152600185)

[**3.3.3.** **TITLE** 16](#_Toc152600186)

[**3.3.4.** **LINK** 16](#_Toc152600187)

[**3.3.5.** **STYLE** 16](#_Toc152600188)

[**3.3.6.** **SCRIPT** 17](#_Toc152600189)

[**3.4.** **레이아웃** 18](#_Toc152600190)

[**3.4.1.** **Mobile App/Web** 18](#_Toc152600192)

[**3.4.2.** **PC WEB** 19](#_Toc152600193)

[**4.** **CSS(Cascading Style Sheet)** 20](#_Toc152600194)

[**4.1.** **기본규칙** 20](#_Toc152600195)

[**4.1.1.** **CSS 파일 불러오기** 20](#_Toc152600196)

[**4.1.2.** **클래스명 네이밍 기본규칙** 20](#_Toc152600197)

[**4.1.3.** **표기법** 21](#_Toc152600198)

[**4.1.4.** **따옴표 사용 범위** 21](#_Toc152600199)

[**4.1.5.** **들여쓰기** 22](#_Toc152600200)

[**4.1.6.** **공백** 22](#_Toc152600201)

[**4.1.7.** **줄바꿈** 22](#_Toc152600202)

[**4.2.** **속성** 23](#_Toc152600203)

[**4.2.1.** **선언순서** 23](#_Toc152600204)

[**5.** **JAVASCRIPT** 25](#_Toc152600205)

[**5.1.** **기본규칙** 25](#_Toc152600206)

[**5.1.1.** **Script 파일 불러오기** 25](#_Toc152600207)

[**5.1.2.** **표기법** 25](#_Toc152600208)

[**5.1.3.** **들여쓰기** 25](#_Toc152600209)

[**5.1.4.** **변수** 25](#_Toc152600210)

[**5.1.5.** **함수** 26](#_Toc152600211)

[**6.** **Sass(Syntactically Awesome Style Sheets, SCSS)** 27](#_Toc152600212)

[**6.1.** **환경설정** 27](#_Toc152600213)

[**6.1.1.** **개발환경설정** 27](#_Toc152600214)

[**6.1.2.** **폴더구조** 28](#_Toc152600215)

[**6.2.** **작동방법** 28](#_Toc152600216)

[**6.3.** **기본규칙** 28](#_Toc152600217)

[**6.3.1.** **SCSS 파일관리** 28](#_Toc152600218)

[**6.3.2.** **주요문법** 29](#_Toc152600219)

INTRO

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

구현된 화면과 화면 목록, 컴포넌트 가이드, 퍼블리싱 소스는 한 화면으로 확인 가능한 홈페이지 형식으로 제공되어 있다.

# **환경설정**

## **Mobile App/Web**

### **환경정의**

|  |  |
| --- | --- |
| 범주 | 설명 |
| OS | iOS 13 ~ 17 / Android 10 ~ 14 |
| 해상도 | - 기준화면크기 375x812 기반으로 가로크기 가변대응(최저: 360 ~ 최대: 428)  - 모든 모바일기기의 최적화를 위해 MobileWebView의 meta viewport 고정  \* viewport base 고정 ex> 안드로이드 360/ iPhone11 375  - Tablet의 경우 Mobile View를 기본으로 함 |
| 웹표준 | Web 표준 준수(W3C) |
| 접근성 | 장애인차별금지법을 통한 접근성 인증 마크획득은 프로젝트 진행상황에 따라 시행 |
| Text Editor | 무관 |
| Encoding | UTF-8 |
| 네이티브영역 | 메인, 전체메뉴, 헤더, 인증/보안, 촬영  \* 이 외의 공통영역은 개발 협의 하에 진행 |

### **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/images 폴더 하위, 이미지 용도 별로 분류된 폴더에 위치한다.
* UIUX관련 스크립트 파일**(\*.js)**은 resource/js/uiux 폴더에 위치한다.
* 스타일시트 파일**(\*.css)**은 resource/css/uiux 폴더에 위치한다.
* Web Font 파일은 resource/font 폴더에 위치한다.

|  |
| --- |
| **Mobile App/Web.**  mobile app/web  ├─── html/  │ └─── page/  └─── resource/  ├─── css/uiux  ├─── font/  ├─── image/콘텐츠별 폴더  └─── js/uiux |

## **PC Web**

### **환경정의**

|  |  |
| --- | --- |
| 범주 | 설명 |
| OS | 무관 |
| Browser | 최신 모던 브라우저 5종(Chrome, Edge, Safari, Firefox, Whale) |
| 해상도 | 기준화면크기 1920x1080 / 최저화면크기 1536x722 (노트북 1920x1080 125%화면비) \* 화면크기는 디자인시안에 따라 변경될 수 있음 |
| 웹표준 | Web 표준 준수(W3C) |
| 접근성 | 장애인차별금지법을 통한 접근성 인증 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.
* UIUX관련 스크립트 파일**(\*.js)**은 resource/js/uiux 폴더에 위치한다.
* 스타일시트 파일**(\*.css)**은 resource/css/uiux 폴더에 위치한다.
* Web Font 파일은 resource/font 폴더에 위치한다.

|  |
| --- |
| **PC Web.**  pcweb/  ├─── html/  │ └─── page/  └─── resource/  ├─── css/uiux  ├─── font/  ├─── image/콘텐츠별 폴더  └─── js/uiux |

# **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. 파일 네이밍**의](#_네이밍) 규칙에 따른다.

### **기본제공파일**

**기본제공파일 CSS**

|  |  |
| --- | --- |
| 파일명 | 설명 |
| ui\_layout.css | layout 구성 CSS파일 |
| ui\_common.css | 공통요소 (normalize, reset, web font, helper, 콘텐츠공통) 및 컴포넌트 구성 CSS파일 |
| ui\_contents.css | 콘텐츠화면 구성 CSS파일 |
| ui\_plugin.css | framework, plugin 모음 CSS파일 |
| ui\_style.css | @Import CSS파일(모든 UIUX CSS파일들은 이 파일에서 관리) |
|  |  |

**기본제공파일 JS**

|  |  |
| --- | --- |
| 파일명 | 설명 |
| ui\_common.js | 레이아웃, 페이지내용, 컴포넌트 함수 선언 및 실행 JS파일 |
| ui\_plugin.js | framework, plugin 모음 JS파일 |
|  |  |

## **파일 네이밍**

### **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.

* 영문, 숫자, 언더스코어( \_ )를 사용하여 작성한다.
* 2개 이상의 단어를 조합할 경우 언더스코어( \_ ) 혹은 카멜 표기법을 사용한다.   
  \* 파일명에 공백을 사용하지 않는다.  
  \* 프로젝트에서 요구하는 특정한 규칙이 있다면 프로젝트 규칙을 우선한다.  
  \* 카멜 표기법: 영문 소문자를 기본으로 하여 단어와 단어 사이를 대문자로 구분하는 방식

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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**

* 영문 소문자, 숫자, 언더스코어( \_ )를 사용하여 작성한다.
* 영문 대문자, 숫자로 시작하지 않는다.
* 기능, 업무, 서비스 명을 따른다.   
  **Ex> payment.css service.css …**
* 2개 이상의 단어를 조합할 경우 언더스코어( \_ )를 사용한다.  
  \* 단어조합은 3단계 이하를 권장하나 경우에 따라 그 이상도 허용한다.

\* 파일명에 공백을 사용하지 않는다.

### **JAVASCRIPT파일 네이밍**

* 영문 소문자, 숫자, 언더스코어( \_ )를 사용하여 작성한다.
* 영문 대문자, 숫자로 시작하지 않는다.
* 기능, 업무, 서비스 명을 따른다.  
  **Ex> main.js, service.js …**
* 2개 이상의 단어를 조합할 경우 언더스코어( \_ )를 사용한다.\* 단어조합은 3단계 이하를 권장하나 경우에 따라 그 이상도 허용한다.

\* 파일명에 공백을 사용하지 않는다.

### **이미지파일 네이밍**

* 영문 소문자, 숫자, 언더스코어( \_ )를 사용하여 작성한다.
* 영문 대문자, 숫자로 시작하지 않는다.
* 예약어가 있는 경우 예약어를 사용한다. (**4.1.2.1. 디자인 예약어~4.1.2.2. 기능 예약어** 규칙 참고)
* 2개 이상의 단어를 조합할 경우 언더스코어( \_ )를 사용한다.  
  \* 단어조합은 3단계 이하를 권장하나 경우에 따라 그 이상도 허용한다.  
  \* 파일명에 공백을 사용하지 않는다.
* ‘**분류\_(형태 or 의미 or 기능)\_상태**’의 순으로 조합한다.  
  \* 분류는 필수이며 그 이 후 반복 및 생략이 가능하다. (아래 예제 참고 必)  
  \* 같은 분류의 이미지가 두 개 이상일 시 파일명 마지막에 타입 및 숫자정보를 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. |

## **주석**

### **기본규칙**

* 주석 기호와 주석 내용 사이에는 공백 한 칸이 있어야 한다.
* 종료 주석에는 ‘//’ 기호로 종료임을 표시한다.
* 수정 이력이 한 줄 이상일 경우 그룹화하여 표시하며  
  시작과 종료 주석 내용을 동일하게 한다.

#### HTML

* 수정이력이 있을 경우, **날짜**(yymmdd) **수정 : 상세 내용** 순으로 작성한다. (아래의 예시 참고)

|  |
| --- |
| **Example.**  <!DOCTYPE html>  <html lang="ko">  <!-- 220911 수정: 상세내용 -->  <div>  …  </div>  <!-- //220911 수정: 상세내용 -->  <img><!-- 220911 수정: 상세내용 --> |

#### CSS, JAVASCRIPT 주석 규칙 공통사항

* CSS, JAVASCRIPT의 주석규칙은 아래 표를 기본으로 한다.
* 중분류와 소분류의 구분은 빈 줄의 개수로 한다.

|  |  |
| --- | --- |
| 대분류간 빈 줄 | 3줄 |
| 중분류간 빈 줄 | 2줄 |
| 소분류간 빈 줄 | 1줄 |

|  |
| --- |
| **Example.**  /\* ##############################################################  Large category #대분류  ############################################################## \*/  √ 3줄: 대분류간 빈 줄  √  √  /\* ##############################################################  Large category #대분류  ############################################################## \*/  /\* 중분류 \*/  √ 2줄: 중분류간 빈 줄  √  /\* 소분류 \*/  √ 1줄: 소분류간 빈 줄  /\* 소분류 \*/ |

#### CSS

* 대분류 – 중분류 - 소분류 순으로 분류한다.
* 수정이력이 있을 경우, **날짜**(yymmdd) **수정 : 수정내용** 순으로 작성한다. (아래의 예시 참고)

|  |
| --- |
| **Example.**  /\* ##############################################################  Form(대분류) #폼  ############################################################## \*/  /\* reset(중분류) \*/  √ 2줄: 중분류간 빈 줄  √  /\* default(중분류) \*/  √ 2줄: 중분류간 빈 줄  √  /\* ty-01(소분류) \*/  √ 1줄: 소분류간 빈 줄  /\* ty-02(소분류) \*/  /\* 220911 수정 : 수정내용 \*/  .temp .box { border: 1px solid #000; }  .temp .box { border: 1px solid #000; }  .temp .box { border: 1px solid #000; }  .temp .box { border: 1px solid #000; }  /\* //220911 수정 : 수정내용 \*/ |

#### JAVASCRIPT

* 아래 예시를 기준으로 소스를 모듈화하며 그 외의 경우 javascript 한 줄 주석( // )를 기본으로 사용한다.
* 대분류 – 소분류 순으로 분류한다.
* 자세한 함수 설명이 필요할 시 함수 용례와 매개변수 정보를 담고 있는 주석을 사용한다.
* 수정이력이 있을 경우, **날짜**(yymmdd) **수정 : 수정내용** 순으로 작성한다. (아래의 예시 참고)

|  |
| --- |
| **Example.**  /\*\* -------------------------------------------  select(대분류) #셀렉트  ---------------------------------------------\*/  /\* 함수(소분류) \*/  var fnName = (function() {  return {  // 한 줄주석  ….중략….  }  })();  √ 1줄: 소분류간 빈 줄  /\* 이벤트(소분류) \*/  $(document).on(‘click’, ‘.selector’, function() {  /\* 220911 수정 : 수정내용 \*/  ….  /\* //220911 수정 : 수정내용 \*/  });  /\*\* -------------------------------------------  \* select(대분류) #셀렉트’  \*  \* x를 n번 곱한 수 반환(함수용례)  \* @param {number} x 거듭제곱할 숫자  \* @param {number} n 곱할 횟수  \* @return {number} x의 n거듭제곱을 반환함  ---------------------------------------------\*/  function pow(x,y) {  …. } |

# **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”> |

### **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.**



### **Mobile App/Web DTD 기본형식**

* HTML5 DTD를 기본으로 사용한다.

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

### **PC DTD 기본형식**

* HTML5 DTD를 기본으로 사용한다.
* HTML5를 지원하지 않는 DTD(XHTML 1.0 Transitional DTD 등)를 사용할 경우 HTML5의 신규 태그와 animation과 같은 CSS3 신기술을 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 charset=utf-8"> |

### **META VIEWPORT**

* Viewport는 모바일 및 반응형 서비스에서만 선언한다.
* 개발과 관련하여 특별한 사항이 없는 경우 device-width로 선언한다.

|  |
| --- |
| **Example.**  <meta name="viewport" content="width=device-width, initial-scale=1.0"> |

### **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 1탭(공백 4칸)을 들여 쓴다.

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

### **SCRIPT**

* type, src 순서로 attribute를 선언한다.
* HTML5 DTD일 경우, type은 생략 가능하다.
* 하위에 선언되는 script는 1탭(공백 4칸)을 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/uiux/ui\_common.js"></script>  <script type="text/javascript">  var $layout = $(‘.layout’);  </script> |

## **레이아웃**

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

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



### **Mobile App/Web**

|  |
| --- |
| **Example.**  class=”layout”  class=”container”  class=“contents”  class=“scroll-area”  class=”header”  class=“header-wrap” |

#### 레이아웃 예약어

Mobile App/Web 레이아웃에는 다음 표에 예약된 class를 기초로 한다.

|  |  |
| --- | --- |
| .예약어 | 범위 |
| .layout | 페이지 전체 영역(바깥쪽) |
| .header | 머리글 영역 |
| .header-wrap | 머리글 영역(안쪽) |
| .container | 페이지 전체 영역(안쪽) |
| .contents | 본문 영역 |
| .scroll-area | 스크롤 되는 영역(안쪽) |

### **PC WEB**

|  |
| --- |
| **Example.**  class=”layout”  class=”header”  class=“header-wrap”  class=”footer”  class=“footer-wrap”  class=”container”  class=“container-wrap”  class=“lnb”  class=“contents-wrap”  class=“contents” |

#### 레이아웃 예약어

PC Web 레이아웃에는 다음 표에 예약된 class를 기초로 한다.

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

# **CSS(Cascading Style Sheet)**

## **기본규칙**

* CSS는 사용 가능한 Hack을 제외하고 W3C validation을 통과해야 한다
* CSS파일에서 한글이 제대로 표기되지 않을 경우를 대비하여 CSS파일 첫 줄에 HTML문서와 동일한 인코딩을 선언한다.   
  **Ex> @charset "utf-8";**

### **CSS 파일 불러오기**

* CSS파일을 불러오기 위하여 <link> 방식을 사용한다.
* 유동적인 CSS파일 관리를 위해 UIUX관련 CSS는 ui\_style.css 파일 내에서 @import 방식으로 관리한다.  
  (ex> 솔루션(보안키패드,텍스트에디터) 등은 ui\_style.css에 넣지 않는다.)
* <head>에 ui\_plugin.css - ui\_style.css - etc 순으로 선언한다.

|  |
| --- |
| **Example.**  <head>  <!-- Publish CSS -->  <link rel=”stylesheet type=”text/css” href=”경로/ui\_plugin.css”>  <link rel="stylesheet" type="text/css" href="경로/ui\_style.css">  </head> |

### **클래스명 네이밍 기본규칙**

* 영문 소문자, 숫자, 하이픈( - )을 사용하여 작성한다.
* 영문 대문자, 숫자로 시작하지 않는다.
* 명확하고 직관적인 단어를 선택한다.
* 예약어가 있는 경우 예약어를 사용한다. (아래 표 참고)
* 2개 이상의 단어를 조합할 경우 하이픈( - )을 사용한다.  
  \* 단어조합은 3단계 이하를 권장하나 경우에 따라 그 이상도 허용한다.
* 디자인(CSS)과 기능(JS)을 분리하여 사용한다.

\* 기능을 구현할 때 **추가 데이터 어트리뷰트**를 사용하여 디자인과 기능을 분리하는 것을 권장한다.   
**Ex> <div class=”accordion active” data-accordion=”accordion\_list”></div>**

#### 디자인 예약어

|  |  |  |  |
| --- | --- | --- | --- |
| 예약어 | 설명 | 예약어 | 설명 |
| se | stove element | sp | stove package |
| sl | stove layout |  |  |
| nav | navigation | btn | 버튼 |
| gnb | global navigation bar | desc | 설명 |
| lnb | local navigation bar | msg | 메세지 |
| bg | 배경 | icon | 아이콘 |
| arrow | 화살표 | empty | 없음 |
| up | 상 | down | 하 |
| left | 좌 | right | 우 |
| prev | 이전 | next | 다음 |
|  |  |  |  |

#### 기능 예약어

|  |  |  |  |
| --- | --- | --- | --- |
| 예약어 | 설명 | 예약어 | 설명 |
| active | 기능활성화 | on | 켜짐 |
|  |  |  |  |

### **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";  .property { font-family: '맑은 고딕', 'consolas’; background: url(‘../image/content/aa.png’); }  .property:after { content: ‘’; display: block; clear: both; } |

### **들여쓰기**

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

|  |
| --- |
| **Example.**  @keyframes slidein {  from {  transform: translateX(0%);  }  to {  transform: translateX(100%);  }  } |

### **공백**

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

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

### **줄바꿈**

* **한줄 코딩을 원칙으로 한다.** (고객사 요청 혹은 SCSS사용 등 여러줄 코딩 필요시 협의)

|  |
| --- |
| **한줄코딩 Example.**  .list-wrap { display: block; background: rgba(255, 255, 255, 0.1); }  .list-wrap ul { display: flex; }  .list-wrap ul li { flex: 1; }  **여러줄코딩 Example.**  .list-wrap {  display: block;  background: rgba(255, 255, 255, 0.1);  }  .list-warp ul {  display: flex;  }  .list-wrap ul li {  flex: 1;  } |

## **속성**

### **선언순서**

* 유동적인 tag사용을 위해 tag선택자와 class 동시 사용을 지양한다.
* 속성 선언 순서는 **우선순위 - 표시 - 위치 - 레이아웃 – 박스모델 – 폰트 – 배경 - 부가속성**으로 작성한다. (아래 표 참고)

|  |  |
| --- | --- |
| 범주 | 속성 |
| 콘텐츠생성(:before, :after사용시) | content |
| 우선순위 | z-index |
| 표시 | overflow, opacity, visibility |
| 위치 | position, top, right, bottom, left / margin, transform (위치로사용할경우) |
| 레이아웃 | display, float |
|  | vertical-align, justify-contents, align-items |
| 박스모델 | width, height / line-height(크기로 사용할 경우) |
|  | margin, padding, border |
|  | box-sizing, border-radius |
| 폰트 | text-align, color, font-size, font-weight, line-height … |
| 배경 | background, box-shadow |
| 부가속성 | cursor, transition, animation… |

# **JAVASCRIPT**

## **기본규칙**

### **Script 파일 불러오기**

* <head>에 **플러그인 및 프레임워크 리소스(jquery, jquery ui … ) - ui\_plugin.js - ui\_common.js** 순으로 선언한다.

|  |
| --- |
| **Example.**  <head>  <!-- 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\_plugin.js"></script>  <script type="text/javascript" src="경로/ui\_common.js"></script>  </head> |

### **표기법**

* 영문 소문자를 기본으로 사용한다.
* 함수명 및 변수명은 카멜 표기법을 사용한다.\* 카멜 표기법: 영문 소문자를 기본으로 하여 단어와 단어 사이를 대문자로 구분하는 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**

* 명확한 네이밍을 사용하며 단축어 사용시 의미를 유추 가능하게 사용한다. Ex> a, b, c 등 지양
* 선택자 변수명 앞에 ‘$(달러기호)’를 사용하여 값 형태의 변수와 구분한다.
* 범주가 같은 것 끼리는 그룹화하는 것을 권장한다.

|  |
| --- |
| **Example.**  // selector  var $layout = $(‘.layout’),  $header= $(‘.header’);  // width, height  var headerW = $header.outerWidth(),  headerH = $header.outerHeight(); |

### **함수**

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

[별첨1] ※ 본 섹션은 CSS전처리기 사용시 참고해야 할 셋팅/관리/코딩 규칙이 기재되어 있음.

# **Sass(Syntactically Awesome Style Sheets, SCSS)**

## **환경설정**

### **개발환경설정**

스타일시트는 SCSS파일을 CSS로 컴파일하여 관리한다.   
※ Microsoft Visual Stude Code 기준의 환경 셋팅   
(컴파일환경 셋팅은 본 가이드 외의 방법으로도 가능하므로 개발환경에 맞게 선택하여 사용하여도 무방함)

1. VS Code (Visual Studio Code) 텍스트 에디터를 설치한다.
2. VS Code의 확장프로그램(extension)을 설치한다.  
   - Sass : VS Code 내에서 SCSS파일 작성을 가능하게 하는 확장프로그램  
   - Live Sass Compiler : SCSS를 CSS로 자동으로 컴파일 해주는 확장프로그램
3. settings.json을 수정한다.

* File(파일) – Preferences(기본설정) – Settings(셋팅)  
  \* 단축키 ctrl + ,(쉼표)
* 화면 왼쪽 메뉴 – Extensions – Live Sass Compiler – Edit in settings.json 클릭한다.
* settings.json파일은 아래의 예시와 같이 설정한다.  
  \* settings.json 파일의 설정 값은 프로젝트마다 달라질 수 있다.

|  |
| --- |
| **Example.**  "liveSassCompile.settings.formats": [  {  "format": "expanded,  "extensionName": ".css",  "savePath": "~/..",  "indentWidth": 4,  },  ],  "liveSassCompile.settings.generateMap": false,  "liveSassCompile.settings.compileOnWatch": false,  "liveSassCompile.settings.includeItems": [  "/\*\*/css/uiux/scss/\*.scss"  ], |

### **폴더구조**

SCSS폴더 구조는 아래의 예시를 따른다.

* **\*.scss** 파일은 uiux/scss 폴더에 위치한다.

|  |
| --- |
| **Example .**  css/  └─── uiux/  ├─── scss/  │ ├─── \_library.scss  │ ├─── ui\_layout.scss  │ ├─── ui\_common.scss  │ └─── ui\_contents.scss  ├ ui\_layout.css  ├ ui\_common.css  ├ ui\_contents.css  ├ ui\_plugin.css  └ ui\_style.css |

## **작동방법**

* VS Code 화면 오른쪽 하단의 Watch Sass를 누르면 Watching으로 바뀌는 것을 확인한다  
  \* Watching 상태여야 SCSS가 실시간으로 반영된다.
* SCSS파일을 성공적으로 저장했을 경우 Success 표시된다.  
  \* error 표시확인시 저장이 안된다. (error라인을 수정하여 저장해야 한다.)

## **기본규칙**

* **SCSS의 작성 기본규칙은 CSS 기본규칙을 기반으로 한다.** (4.1.7.줄바꿈 제외)
* SCSS파일에서 한글이 제대로 표기되지 않을 경우를 대비하여 SCSS파일 첫 줄에 HTML문서와 동일한 인코딩을 선언한다.   
  **Ex> @charset "utf-8";**

### **SCSS 파일관리**

* CSS파일로 컴파일 되지 않는 파일은 파일명 앞에 언더스코어( \_ )를 추가하여 작성하며, SCSS파일 상단에 @import하여 사용한다.  
  **Ex> @import "\_ library.scss";**

|  |  |
| --- | --- |
| 파일명 | 설명 |
| \_library.scss | (전역)변수선언, mixin(공통) 내용작성 SCSS파일 |
| ui\_layout.scss | layout 구성 SCSS파일 |
| ui\_common.scss | 공통요소 (normalize, reset, web font, helper, 콘텐츠공통) 및 컴포넌트 구성 SCSS파일 |
| ui\_contents.scss | 콘텐츠화면 구성 SCSS파일 |

### **주요문법**

#### 중첩작성(Nesting)

* 아래 예시와 같이 중첩으로 작성한다.

|  |
| --- |
| **Example.**  .nav {      ul {          margin: 0;          padding: 0;          list-style: none;          li {              display: inline-block;              a {                  display: block;                  padding: 6px 12px;                  text-decoration: none;              }          }      }  } |

#### 부모선택자(&)

* 부모선택자(&)를 사용하여 자신이 포함된 상위 선택자를 참조할 수 있다.

|  |
| --- |
| **Example.**  .nav {      ul {          margin: 0;          padding: 0;          list-style: none;          li {              display: inline-block;              a {                  display: block;                  padding: 6px 12px;                  text-decoration: none;  &:hover {  background-color: #ccc;              }              }          }      }  } |

#### 변수

* 반복적으로 사용되는 속성값을 변수에 저장하여 재사용 할 수 있다.
* 변수명 앞에 항상 ‘$(달러기호)’를 붙여준다.

|  |
| --- |
| **Example.**  $color-01 : #F6F7FA;  $color-02 : #EBEBED;  $color-03 : #DCDEE2;  span {  color: $color-01;  } |

#### 믹스인(Mixin)

* 믹스인을 사용하여 스타일 시트 전체에서 재사용할 수 있는 CSS속성을 그룹화하여 정의한다.
* @mixin을 사용하여 선언하며, 이를 사용할 때는 @include를 사용한다.

|  |
| --- |
| **Example.**  @mixin hiddentxt {  overflow: hidden;  position: absolute;  width: 1px;  height: 1px;  font-size: 1px;  margin: 0;  padding: 0;  border: 0;  white-space: nowrap;  clip: rect(1px,1px,1px,1px);  clip-path: inset(50%);  }  .hidden-txt {  @include hiddentxt;  } |