
Coding Convention

for markup languages (HTML/CSS)

Contents

Naming Rules

1. 기본규칙
2. id 및 class 규칙
3. 파일 규칙
4. 폴더 규칙

HTML

1. DOCTYPE
2. Head
3. Body
4. 기본 규칙

CSS

1. 기본 규칙
2. Media Query

PHP

1. 파일 명명법
2. 변수 명명법
3. 작성 기본원칙

Naming Rules

기본규칙

- 이름은 영문 소문자, 숫자를 이용하여 작성한다.
- 첫 글자가 대문자, 숫자가 되지 않도록 한다. 첫 글자는 영문 소문자로 시작한다.
- 예약어는 이름으로 사용하지 않는다.

case	default	new	null	true	in
long	double	delete	name	area	text

id 및 class 규칙

- id는 하나의 파일 내 한번만 사용된다
- class는 여러개의 파일에서도 사용이 가능하다.
- id 명과 class 명과 겹치지 않도록 작성한다.
- 두 개의 단어가 합쳐질 때 첫 글자를 대문자로 작성한다.

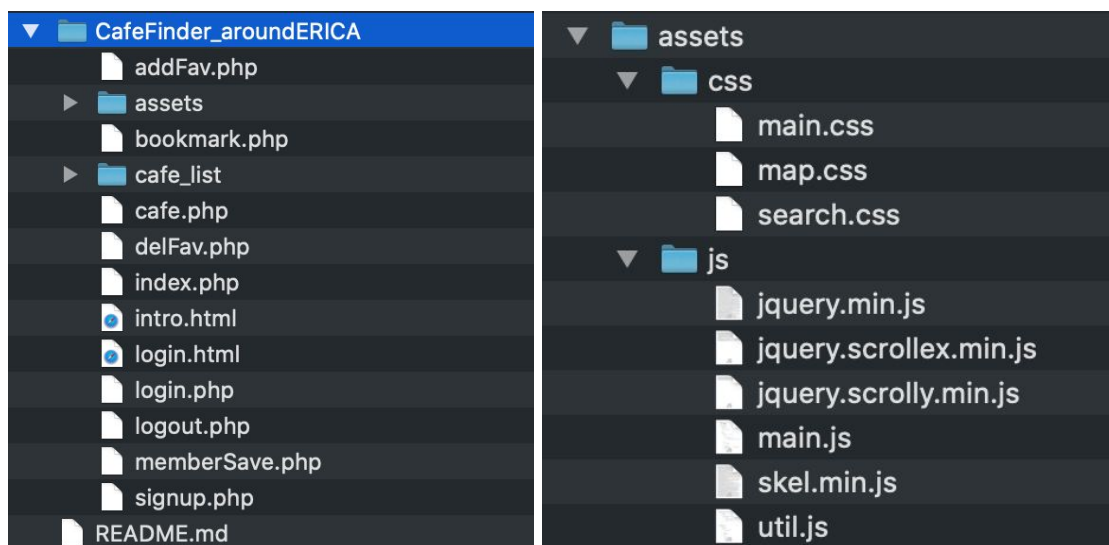
```
.searchBox {  
  width: 100%;  
  margin: 70px 0;  
  border: 3px solid #1e252c;  
}
```

파일 규칙

- 단어와 단어와의 연결은 언더 스코어 (_) 로 한다.
- HTML
 - 해당하는 파일의 주요 기능을 이름으로 작성한다.
- CSS
 - main.css
 - 메인페이지를 포함한 서브페이지들에 적용될 css 파일
 - map.css
 - 지도를 나타내는데에 적용될 css 파일
 - search.css
 - 검색창을 나타내는데에 적용될 css 파일
 - cafe_info.css
 - 여러개의 카페 정보를 담고 있는 페이지에 적용될 css 파일

폴더 규칙

- 단어와 단어와의 연결은 언더 스코어 (_) 로 한다.
- 메인페이지를 포함한 서브페이지는 메인 폴더에 위치한다.
- 여러 개의 css파일이나 js 파일들은 assets 파일에 따로 관리한다.
- 여러 정보들을 담고 있는 페이지 리스트들은 cafe_list 파일에 따로 관리한다.



- 이 외에 필요한 경우에 따라 폴더를 추가 하거나 사용하지 않는 폴더는 삭제한다.

HTML

DOCTYPE

HTML5 사용

```
<!DOCTYPE HTML>
```

Head

title, meta, link, script 순서로 엘리먼트를 선언한다.

1. title

문서의 제목을 정의한다.

문서 제목(<title> 태그) 규칙을 일원화함으로써, 서비스 통일성을 높인다.

문서 제목에 콘텐츠 제목을 포함시킴으로써, 외부 검색 서비스에서 해당 콘텐츠의 노출이 더 잘될 수 있도록 한다.

2. meta

문서의 기본 인코딩, 뷰포트, 스타일 형식 순서로 엘리먼트를 선언한다.

- charset(인코딩)

UTF-8 사용

```
<meta charset="utf-8" />
```

- name

<meta name = “value”>

viewport : 웹 페이지 전체 중 브라우저 창에 보이는 부분을 말며, 창 크기를 조절하여 크게 또는 작게 만들 수 있다.

- content

<meta content = “text”>

text : 메타 정보에 대한 내용

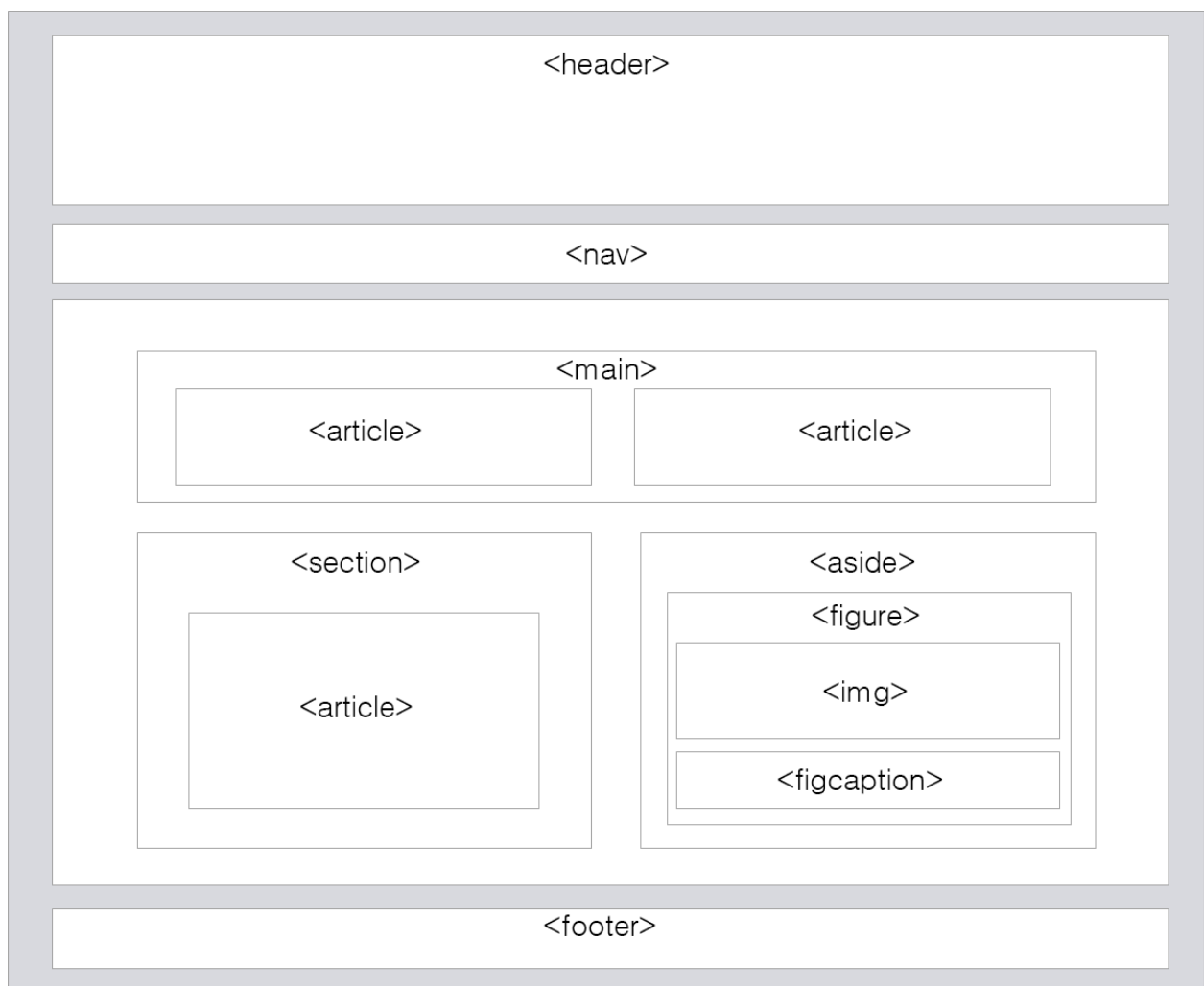
3. link

rel, href 순서로 애트리뷰트를 선언한다.

```
<link rel="stylesheet" href="assets/css/main.css" />
```

Body

1. 기본 레이아웃



<header>	HTML 문서나 섹션(section) 부분에 대한 헤더(header)를 정의함.
<nav>	HTML 문서의 탐색 링크를 정의함.
<section>	HTML 문서에서 섹션(section) 부분을 정의함.
<article>	HTML 문서에서 독립적인 하나의 글(article) 부분을 정의함.
<aside>	HTML 문서에서 페이지 부분 이외의 콘텐츠(content)를 정의함.
<footer>	HTML 문서나 섹션(section) 부분에 대한 푸터(footer)를 정의함.

큰 레이아웃은 전체를 Wrapper로 싸고 header, nav로 쪼갬다.

2. 색인

일반 문서를 HTML로 다음과 같은 규칙에 맞게 색인화 한다.

- h1 : 사이트명
- h2 : 페이지명, 메인메뉴
- h3 : 세부 콘텐츠
- h4 : section의 하위 그룹 제목 혹은 리스트 제목

※ 문서의 구성요소에 따라 변경하여 의미 있게 사용한다.

※ h5 ~ h6은 가급적 사용하지 않는다.

기본 규칙

1. 줄 바꿈

중첩되어 깊이가 깊어질 때 마다 줄을 바꾼다.

2. 들여쓰기

- 들여쓰기를 하면 코드의 가독성이 높아지고 전체 HTML 구조를 쉽게 파악할 수 있다.
- 마크업의 중첩이 깊어질 때마다 자식 엘리먼트는 1탭 들여 쓰고, 1탭의 크기는 공백 4칸으로 설정한다.

3. 주석표시

1. 기본형식

시작 주석 `<!-- 주석 내용 -->`

종료 주석 `<!-- //주석 내용 -->`

- 주석 기호와 주석 내용 사이에는 반드시 공백 한칸이 있어야 한다.

2. 레이아웃 및 콘텐츠 영역의 주석 표기

- 레이아웃과 독립된 콘텐츠 영역의 시작에 주석을 표기하고, 레이아웃은 id 이름과 동일하게 주석을 넣는다.

```
<!-- Header -->
```

4. 애트리뷰트값 표기

애트리뷰트값은 큰 따옴표(“ ”)로 묶는다.

CSS

기본규칙

Charset

문서의 언어셋은 UTF-8으로 최상위에 선언한다. 언어셋이 정해진 번들링 파일이라면 선언하지 않는다.

```
@charset "UTF-8";
```

CSS 데이터

CSS 데이터는 섹션의 경우 head에 메인(main.css)으로, 간단한 콘텐츠의 경우 하나의 css 파일로 로딩될 수 있도록 권장한다.

```
<link rel="stylesheet" href="assets/css/main.css" />
```

줄바꿈

CSS 스타일 속성 간 개행하며, 클래스명을 선언한 뒤 한 칸의 공백을 두고 세부 속성간에도 공백을 준다. CSS 선언의 마지막 속성에 세미콜론(;)을 붙인다.

잘못된 예

```
table{border-collapse: collapse; border-spacing: 0;} (x)
```

```
table {border-collapse: collapse; border-spacing: 0;} (x)
```

```
table{border-collapse:collapse;border-spacing: 0;} (x)
```

올바른 예

```
table {  
    border-collapse: collapse;  
    border-spacing: 0;  
}
```

따옴표 사용범위

공백이 포함된 폰트명, 한글폰트명, 문자열 데이터 타입, filter 속성의 파라미터 값, url 데이터 타입 등 모두 큰따옴표(“ ”)를 사용하여 작성한다. 작은 따옴표(‘ ’)는 사용하지 않는다.

빈 줄

객체를 구분하기 위하여 코드 그룹 간 1줄씩 빈 줄을 넣는다.

주석

비슷한 속성은 그룹으로 묶고 다른 속성은 주석(/**/)을 사용하여 나눈다. 주석 다음에 오는 속성들은 들여쓰기를 하여 작성한다.

```
/* Box */

.box {
  border: solid 2px;
  margin-bottom: 2rem;
  padding: 1.5rem;
}

.box > :last-child,
.box > :last-child > :last-child,
.box > :last-child > :last-child > :last-child {
  margin-bottom: 0;
}

.box.alt {
  border: 0;
  border-radius: 0;
  padding: 0;
}

.box {
  border-color: #eeeeee;
}

/* Button */

input[type="submit"],
input[type="reset"],
input[type="button"],
button,
.button {
  -moz-appearance: none;
}
```

Media Query

- Media Query는 컴포넌트 단위로 분류하여 관련 규칙 바로 뒤에 작성한다.
- 파편화된 스타일이 한 곳으로 모아져 가독성이 좋아진다.
- 불가능하다면 문서의 마지막에 모아서 작성한다.

PHP

파일 명명법

1. 각 단어의 첫 문자를 대문자로 표기하되, 첫 문자는 소문자로 하는 camelCase를 사용한다.

<input type="checkbox"/> 이름	수정한 날짜	유형	크기
assets	2020-06-22 오후 8:33	파일 폴더	
cafe_list	2020-06-22 오후 10:23	파일 폴더	
.DS_Store	2020-06-22 오후 10:23	DS_STORE 파일	9KB
addFav.php	2020-06-24 오전 10:06	PHP 파일	1KB
bookmark.php	2020-06-24 오전 10:09	PHP 파일	9KB
cafe.php	2020-06-24 오전 10:12	PHP 파일	17KB
delFav.php	2020-06-24 오전 10:06	PHP 파일	1KB
index.php	2020-06-24 오전 10:12	PHP 파일	3KB

2. 단 cafe_list 폴더 안 파일은 카페의 추가 및 제거 소요를 고려해 '_' 를 허용한다.

<input type="checkbox"/> 이름	수정한 날짜	유형	크기
icon	2020-06-22 오후 8:33	파일 폴더	
image	2020-06-22 오후 10:23	파일 폴더	
.DS_Store	2020-06-22 오후 10:23	DS_STORE 파일	9KB
cafe_0.php	2020-06-22 오후 10:23	PHP 파일	11KB
cafe_1.php	2020-06-22 오후 10:23	PHP 파일	7KB
cafe_2.php	2020-06-22 오후 10:23	PHP 파일	7KB
cafe_3.php	2020-06-22 오후 10:23	PHP 파일	7KB
cafe_4.php	2020-06-22 오후 10:23	PHP 파일	7KB
cafe_5.php	2020-06-22 오후 10:23	PHP 파일	8KB
cafe_6.php	2020-06-22 오후 10:23	PHP 파일	4KB

변수 명명법

1. mysql을 시행하는 변수는 \$db로 통일한다.

```
$db = new PDO();
```

2. <form>의 method는 “GET”이 아닌 “POST”로 한다.
3. 변수의 “”(quote)를 추가한 변수를 따로 ‘q_변수’로 선언하여 사용한다.

```
$userId = $_SESSION['user_id'];  
  
$q_userId = $db -> quote($userId);
```

4. 사용자 정보를 가진 SESSION의 변수로는 user_name(사용자 이름), user_id(사용자 ID) 등 각 단어 간 _ 를 사용한다.

```
$user_name = $_SESSION['user_name'];  
  
$user_id = $_SESSION['user_id'];
```

5. Error에 관한 SESSION의 변수로는 idError(ID 에러 발생), pwError(패스워드 에러 발생), signUpOk(회원가입 완료) 등 camelCase를 따른다.

```
$_SESSION['signUpOk'] = "no";  
  
$_SESSION['idError'] = "noError";  
  
$_SESSION['pwError'] = "noError";
```

6. 사용자 데이터베이스의 변수명은 user_info(사용자 정보), user_favorite(사용자 북마크) 등 각 단어 간 _ 를 사용한다.

```
+-----+  
| Tables_in_ericacafe |  
+-----+  
| user_favorite       |  
| user_info           |  
+-----+
```

7. for, switch문 등 내부의 변수는 \$i, \$j, \$k ... 의 변수를 사용한다.

작성 기본원칙

1. Session의 시작(session_start())은 php의 시작인 <?php 바로 뒤에 사용한다.

```
<?php session_start(); ?>
```

2. 페이지 이동을 할 때는 기존 페이지를 나가는 exit;을 사용한다.
3. MySQL의 dbname은 ericacafe로 하되, host, port, 비밀번호(초기 비밀번호 : a12345)는 개발자가 지정한다.
4. 카페의 추가를 위해 bookmark.php에 case문을 추가하며 다음을 따른다. 인덱스는 오름차순으로 지정한다.

```
case "cafe_카페인덱스" :  
  
echo '<center> <a href="cafe_list/cafe_카페인덱스.php"class="button special  
fit">카페명</a> </center>';  
  
break;
```

5. 카페의 제거를 위해 bookmark.php에 case문에 해당되는 카페를 제거한다. 단 제거 후에 나머지 카페에 대한 인덱스는 수정하지 않으며, 인덱스를 SHIFT하는 작업은 하지 않는다.