



Seoul
Software
ACademy

웹 개발자 부트캠프 과정

SeSAC x CODINGOn

With. 팀 뽀빠드

HTML

HTML



CSS



JS





귀여운 — 형용사



미니언이 — 명사



춤춘다 — 동사



HTML 이란?



Hypertext Markup Language

웹 페이지를 구성하는 가장 기초적인 언어

링크

웹 페이지에서 다른 페이지로 이동할 수 있도록 하는 것

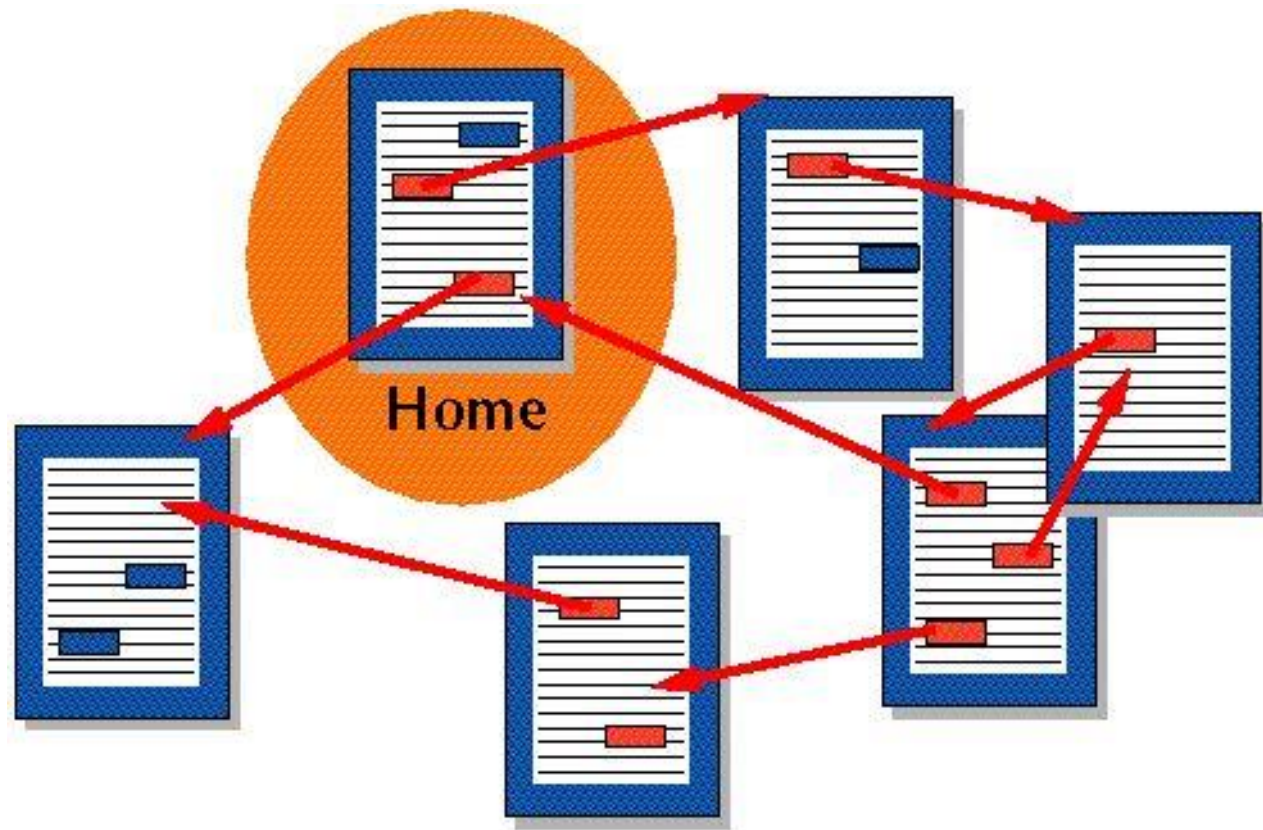
Hypertext Markup Language

마크업

문서나 데이터의 구조를 명기하는 언어

Hypertext

: 웹 페이지에서 다른 페이지로 이동할 수 있도록 하는 것



Markup

: 문서나 데이터의 구조를 명기하다

▼ 표시(Markup)가 없을 때

└ ...

SeSAC 1기 용산 웹 풀스택 과정. 화목 13:00 ~ 19:00, 토 12:00 ~ 18:00, 청년취업사관학교 용산:

▼ 표시(Markup)가 있을 때

└ ...

- 과정명 : SeSAC 1기 용산 웹 풀스택 과정
- 교육시간 : 화목 13:00 ~ 19:00, 토 12:00 ~ 18:00
- 교육 장소 : 청년취업사관학교 용산캠퍼스

HTML 구조 설명

```

1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8">
5      <title>Hello World</title>
6    </head>
7    <body>
8      <h1>Hello World</h1>
9      <p>안녕하세요! HTML5</p>
10   </body>
11 </html>

```

- HTML5 문서는 반드시 `<!DOCTYPE html>`으로 시작하여 문서 형식을 HTML5로 지정
- 실제적인 HTML 문서는 2번째 행부터 시작, `<html>`과 `</html>` 사이에 작성
- `<head>`와 `</head>` 사이에는 HTML 문서 정보를 정의하는 코드가 포함된다.(제목, 저장 방식, 브라우저의 크기 등등)
- 웹 브라우저에 출력되는 모든 요소는 `<body>`와 `</body>` 사이에 위치한다

<head> 태그

- HTML 문서의 **메타 데이터**와 문서 정보를 정의하는 데 사용되는 태그
 - 메타 데이터: 데이터에 대한 정보를 제공하는 데이터
- HTML 문서의 **title, style, link, script, meta** 에 대한 데이터로 화면에 표시되지 않음
 - **title** : HTML 문서의 제목
 - **style** : HTML 문서의 Style 정보 정의
 - **link** : 외부 리소스와의 연결 정보를 정의(CSS 파일 연계에 사용)
 - **script** : JavaScript 를 정의
 - **meta** : 페이지 설명, 키워드, 저자, 화면 크기 등의 정보. 주로 브라우저 또는 검색 엔진에서 사용

〈body〉 태그

- HTML 문서의 **실제 콘텐츠**를 정의하는 데 사용되는 태그
- 다양한 태그를 사용하여 웹 페이지의 구조를 설계할 수 있다.

HTML 기본 문법, 중첩(Nested)

```

1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8">
5      <title>Hello World</title>
6    </head>
7    <body>
8      <h1>Hello World</h1>
9      <p>안녕하세요! HTML5</p>
10   </body>
11 </html>

```

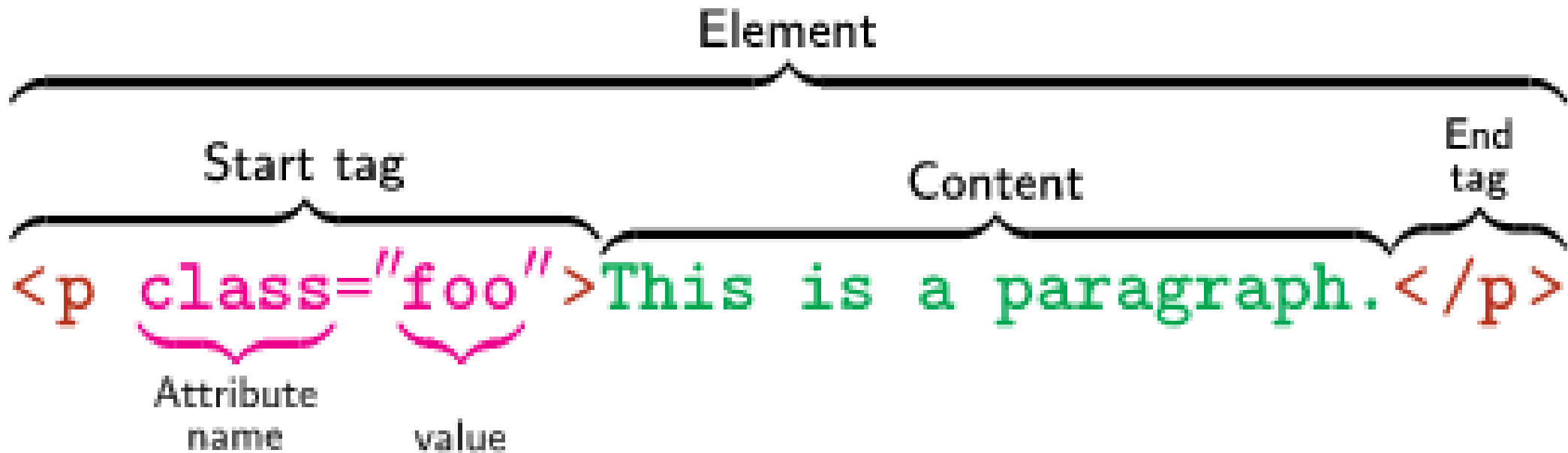
- `<html></html>`은 웹페이지를 구성하는 모든 요소들을 포함한다.
- 예를 보면 `<html>`은 `<head>` 와 `<body>`를 포함하며 `<head>` 는 `<meta>` 와 `<title>` 요소를 `<body>` 요소는 `<h1>`, `<p>`를 포함
- 중첩 관계(부모자식 관계)로 웹페이지의 구조(structure)를 표현한다

HTML 기본 문법, 주석(Comments)

```
1  <!--주석은 컴퓨터가 아닌 사람을 위한 것이며  
2  |   화면에 표시되지 않는다.-->  
3  <p>Hello, KDT World!</p>
```

- 주석(comment)는 개발자에게 코드를 설명하기 위해 사용
- 브라우저는 주석을 화면에 표시하지 않는다.

HTML 태그는 어떻게 생겼을까?



HTML 구성요소 - 1) 태그

<html>

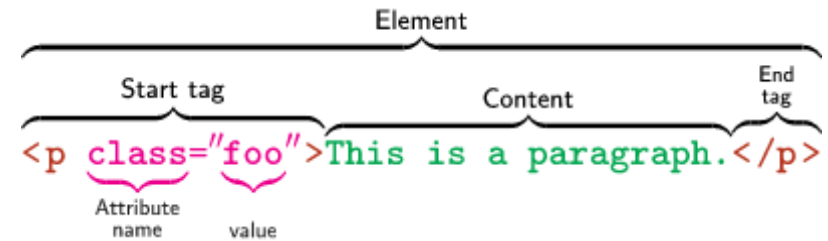
<head>

</head>

<body>

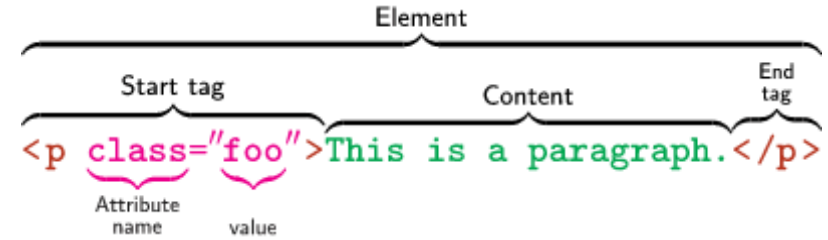
</body>

</html>



HTML 구성요소 - 2) 요소

- 시작태그와 종료태그, 그 사이의 내용



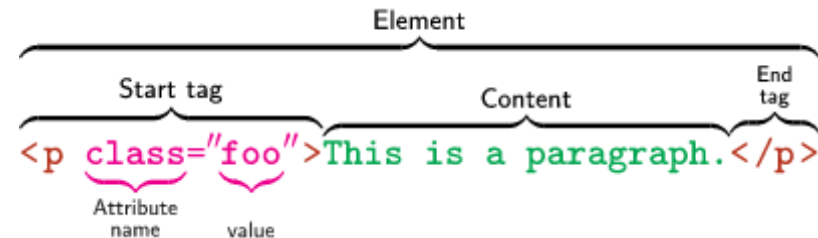
① 빈 요소 (Empty Element)

내용 없이 구조적인 기능만 하는 요소

(ex)
 <hr>

HTML 구성요소 - 2) 요소

- 시작태그와 종료태그, 그 사이의 내용



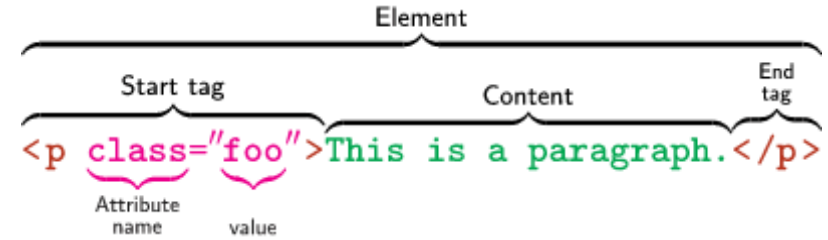
② 블록 요소 (Block Element)

- 기본적으로 부모 요소의 전체 너비(100%)를 차지한다.
- 태그가 시작되면 무조건 **개행(줄바꿈)**이 일어난다.
- 모든 **인라인 요소를 포함**하거나 **다른 블록 요소를 포함**한다.

(ex) <h1></h1>, <div></div>, <p></p>

HTML 구성요소 – 2) 요소

- 시작태그와 종료태그, 그 사이의 내용



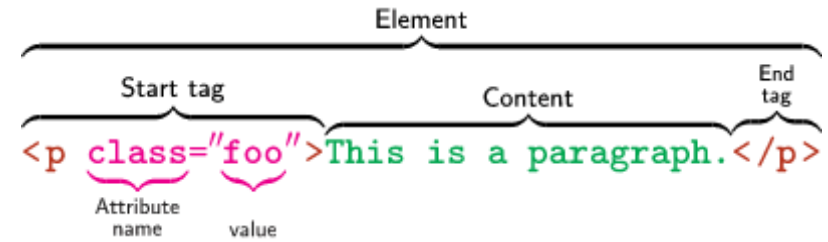
③ 인라인 요소 (Inline Element)

- 텍스트 혹은 이미지 크기에 맞는 **필요한 공간만을 차지**하는 요소이다.
- 너비와 높이를 지정할 수 없다.
- 블록 요소 안에 포함되어 있는 것으로, **인라인 요소만 가질 수 있다.**

(ex) <a>, ,

HTML 구성요소 – 3) 속성

- 태그를 보조하는 명령어로 태그 안에서 작동
- Ex) **id**, **class**, style, width, height 등등
- 태그마다 사용 가능한 속성이 정해져 있음
 - id, class, style은 모든 요소가 사용할 수 있는 속성
- Ex) ` `



tag 종류

HTML 기본태그

<h1> </h1>

<p> </p>

<a>

<div> </div>

 ,

제목 태그, <h1~6></h1~6>

- 제목을 뜻하는 **Heading** 의 약자, h 사용!
- **자동 줄 바꿈!** 왜? 제목이니까!
- 하나의 HTML 문서에는 **하나의 h1 태그를 권장**
- 웹 검색 엔진이 제일 먼저 검색하는 태그!

본문 태그, `<p></p>`

- 본문을 뜻하는 `paragraph` 의 약자, `<p>` 사용!
- 본문을 적기 위한 태그!

줄 바꿈 태그, `
`

- 과연 `
`은 무엇의 약자 일까요?
- 줄을 바꿔 준답니다!

목록, `` or ``

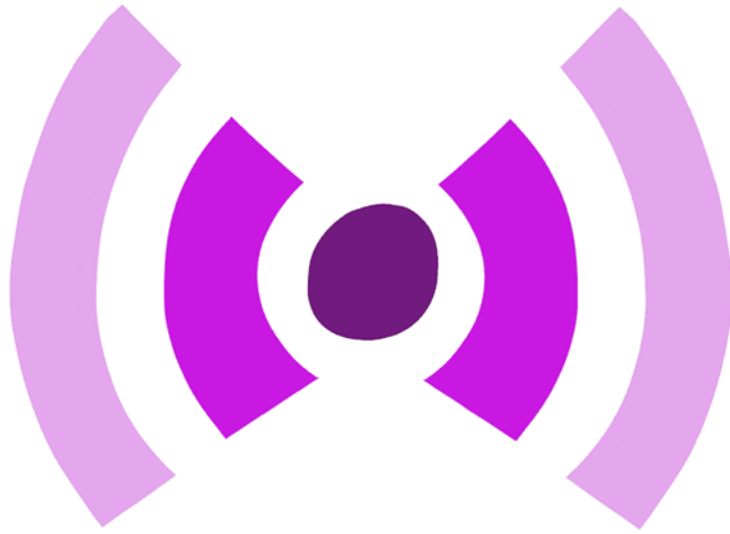
- ``: 순서 없는 목록 (unordered list)
- ``: 순서 있는 목록 (ordered list)

```
<!-- type 마커 모양 : disc, circle, square, none -->
<ul type="circle">
  <li>사과</li>
  <li>바나나</li>
  <li>파인애플</li>
</ul>
<!-- type : 1, A, a, I, i -->
<!-- start : 시작 번호 -->
<!-- reversed : 역순으로 -->
<ol type="i" start="3" reversed>
  <li>1</li>
  <li>2</li>
  <li>3</li>
</ol>
```

수평 줄, <hr>

- 수평으로 된 줄을 그어 줍니다!

Live Server



문자 꾸미기 태그들

- `` : 두껍게!
- `` : 두껍게! + Semantic 한 의미를 지님
- `<i></i>` : 이탤릭
- `` : Emphasized, 강조! 기울여서 표시됨
- `` : 중간 줄!
- `<u></u>` : 밑 줄!

이미지를 넣어주는

- 이미지를 넣을 때 사용!
- 속성 값 중 하나인, `src` 를 사용
- 이미지 로드가 안될 경우 `alt` 속성이 중요!
- 파일을 직접 가져오기, 인터넷 주소에서 가져오기 등등

HTML의 하이퍼링크, <a>

- Anchor 의 약자인 <a> 태그 사용
- 속성 값
 - href : Hypertext Reference 의 약자, 이동할 페이지의 링크
 - target : 링크 된 문서를 열었을 때 문서가 열릴 위치 표시
 - _blank : 새로운 탭에서
 - _self : 현재 탭에서 (기본 값)

사용자 입력 Form 태그 & Input 태그

입력 값 받기! <input>

- type
 - button
 - text
 - password
 - checkbox
 - radio
 - date
 - color
 - range
 - file

버튼, type="button"

- 버튼을 생성
- 주로 특정 기능을 수행 시킬 때 사용

```
<input type="button" value="버튼" />
```

텍스트, type="text"

- 텍스트 입력 값을 받는 폼을 생성
- 우리가 입력하는 ID 입력 부분?
- 텍스트 값을 입력 받아 전달하는 기능

패스워드, type="password"

- 비밀번호 값을 받는 폼을 생성
- 입력 값을 자동으로 안보이게 처리
- 중요 or 비밀 텍스트 값을 전달하는 기능

체크 박스, type="checkbox"

- 여러 선택지 중 여러 개를 선택 가능한 체크 박스 생성
- 속성
 - **name** : 체크 박스의 이름, 같은 분류의 체크 박스는 같은 이름으로 설정
 - **value** : 체크 박스가 실제로 전달하는 값을 지정
 - **checked** : 화면 최초 로딩 시에 선택 된 상태로 로딩

라디오 버튼, type="radio"

- 여러 선택지 중 **하나만 선택 가능한** 라디오 버튼 생성
- 속성
 - **name** : 라디오 버튼의 이름, 같은 name 을 가지는 라디오 버튼은 하나만 선택이 가능 → 하나를 선택하면 다른 선택 값이 취소 됨
 - **value** : 라디오 버튼이 실제로 전달하는 값을 지정
 - **checked** : 화면 최초 로딩 시에 선택 된 상태로 로딩

날짜 선택, type="date"

- 특정 날짜(년/월/일)를 선택
- 속성
 - name : 날짜 선택 폼 이름
- type="datetime-local"
 - 시간 까지 선택 가능!
 - 기존은 "datetime" 을 사용하였으나 시간은 시간대의 영향을 받기 때문에 정확한 데이터 값 전달이 불가능 하여 지금은 사용 X

선택 메뉴를 만드는, <select>

- 선택 메뉴(드롭 다운)를 만드는 태그!
- <select> : select 폼 생성
 - name : select 박스의 이름
- <option> : select 폼의 옵션 값 생성
 - value : 실제로 전달 되는 값
 - selected : 최초에 선택 된 값으로 설정
- <optgroup> : option 을 그룹화
 - label : optgroup 이름 설정
- disabled : 옵션은 보이지만 선택을 못하도록 설정

〈Table〉

테이블을 만들자! <table>

- 표를 만들 때 사용하는 태그
- 먼저 행을 쓰고 행의 자식 요소로 칸을 넣어주는 것이 기본!
 - <table> : 표를 감싸는 태그
 - <tr> : 표 내부의 행
 - <th> : 행 내부의 제목 칸
 - <td> : 행 내부의 일반 칸

HTML - Table

<table border=1>				
<tr>	<td> </td>	<td> </td>	<td> </td>	</tr>
<tr>	<td> </td>	<td> </td>	<td> </td>	</tr>
<tr>	<td> </td>	<td> </td>	<td> </td>	</tr>
</table>				

HTML – Table

```
<table>  
  <tr>  
    <td></td>  
    <td></td>  
  </tr>  
</table>
```

〈table〉 속성

- 〈table〉
 - **border** : 테두리 두께
 - **cellspacing** : 테두리 간격 사이의 너비
 - **cellpadding** : 셀 내부의 간격
 - **align** : 테이블 정렬 속성
 - **width** 와 **height** : 테이블의 너비와 높이
 - **bgcolor** 와 **bordercolor** : 테이블 배경색과, 테두리 색

<td> 속성

<td>

- **colspan** : 해당 칸이 점유하는 열의 수 지정
- **rowspan** : 해당 칸이 점유하는 행의 수 지정

Semantic

HTML 요소는
Semantic 하게 작성되어야 한다.

HTML 요소는 *Semantic* 하게 작성되어야 한다.

“웹 페이지를 이루는 요소에 **의미에 맞는 태그**를 사용하자!”

Semantic tag

〈header〉

〈nav〉

〈footer〉

〈aside〉

〈main〉

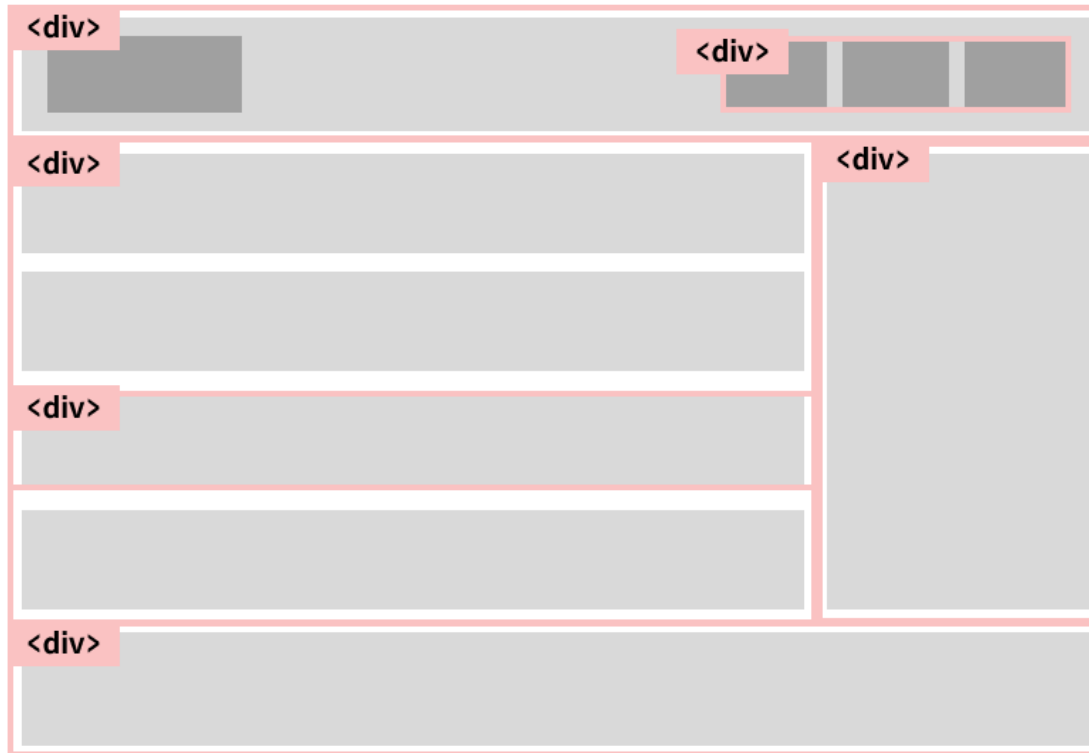
〈article〉

〈h1〉 ~ 〈h6〉

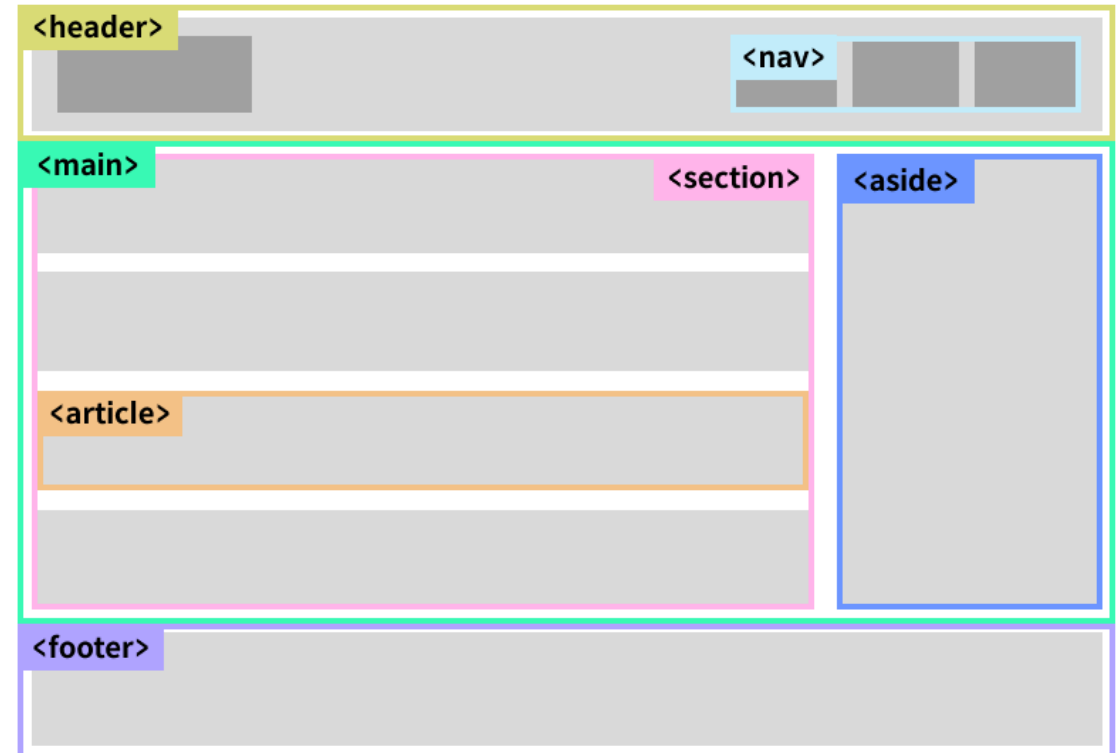
〈section〉

Semantic tag

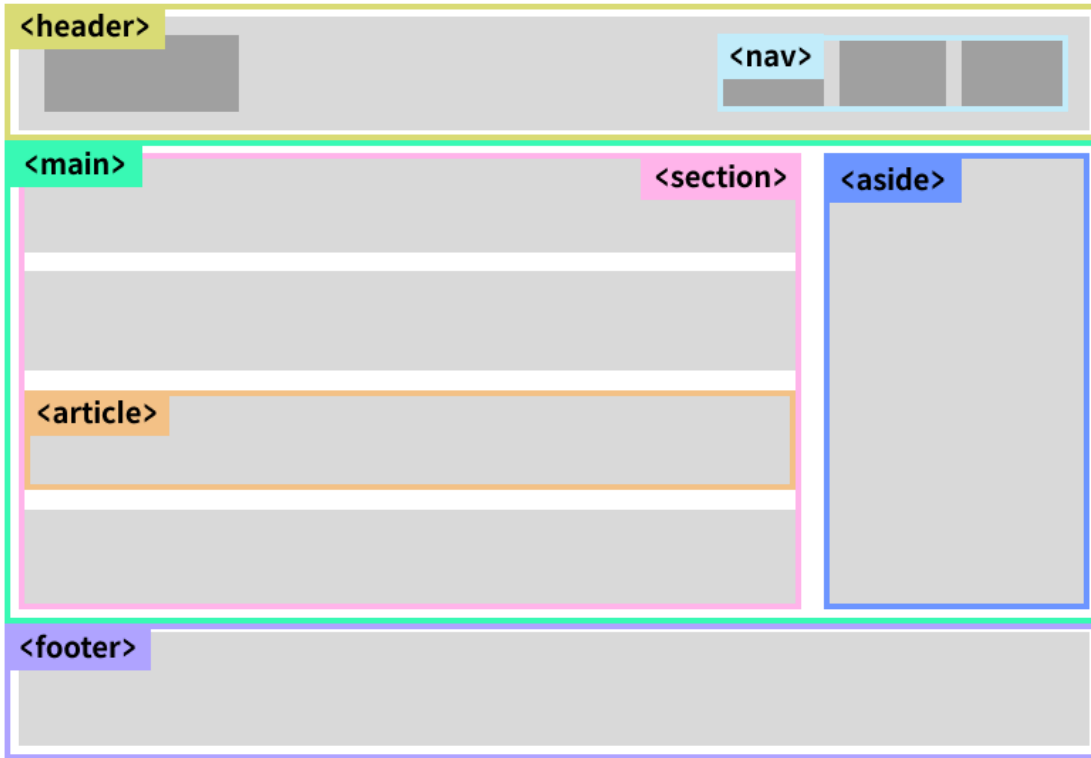
before



after



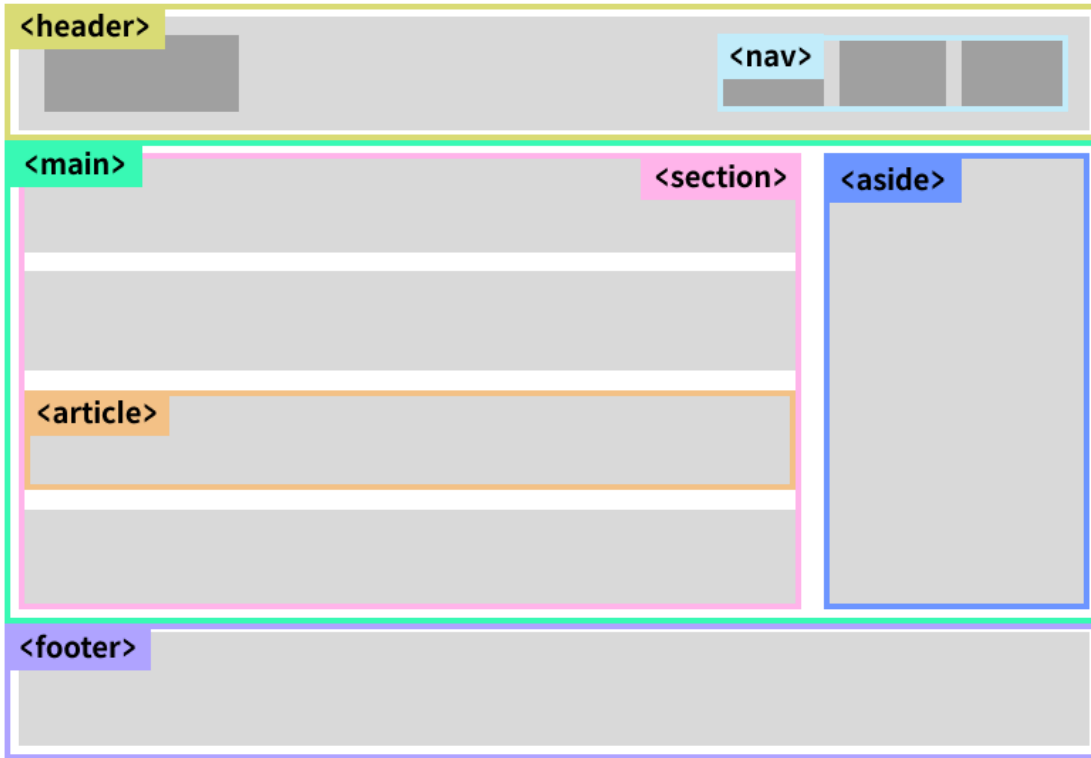
Semantic tag



<header> : 헤더 (로고, 메뉴 아이템 등)

<footer> : 푸터 (하단 영역)

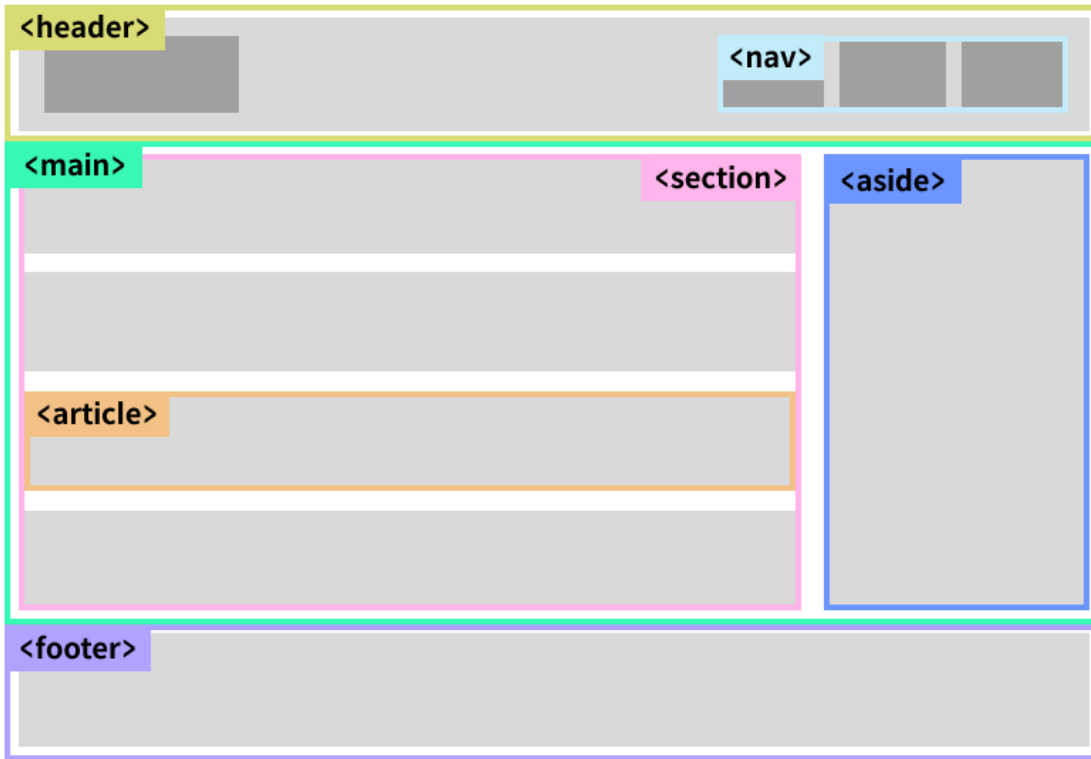
Semantic tag



<nav>

: 네비게이션 (다른 곳으로 이동 가능한 링크들)

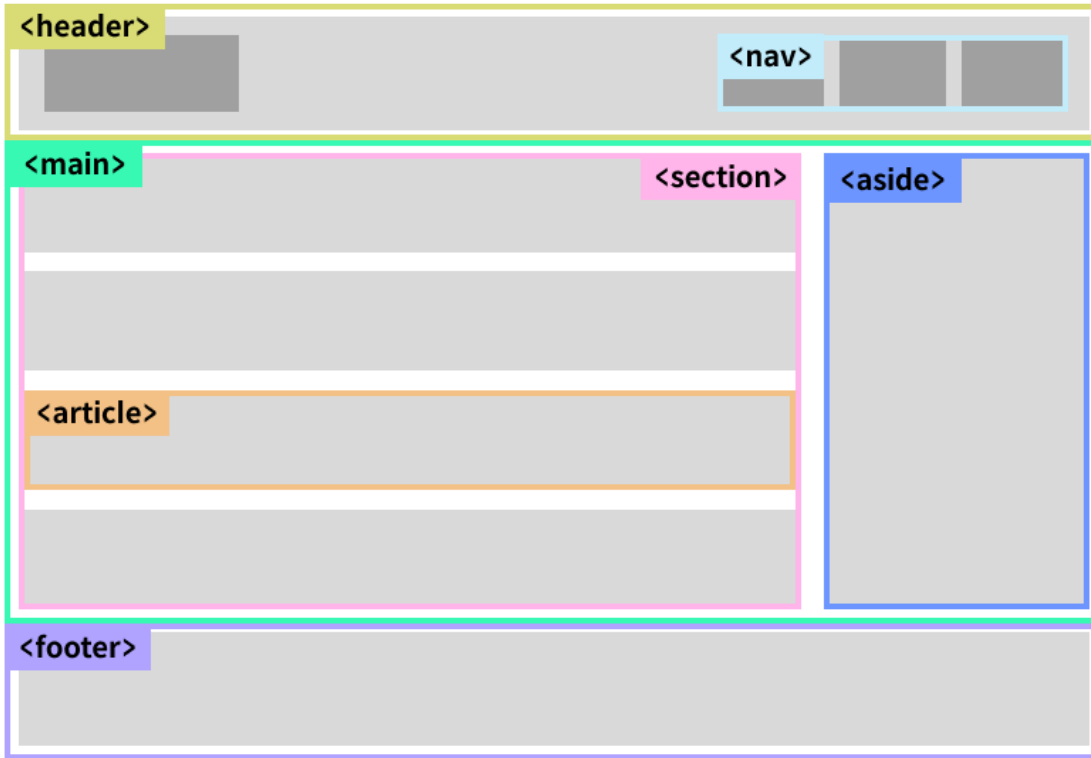
Semantic tag



〈main〉

: 콘텐츠 영역임을 표시. 한 html 페이지당 한번 사용하는 것이 바람직함 (중첩 x)

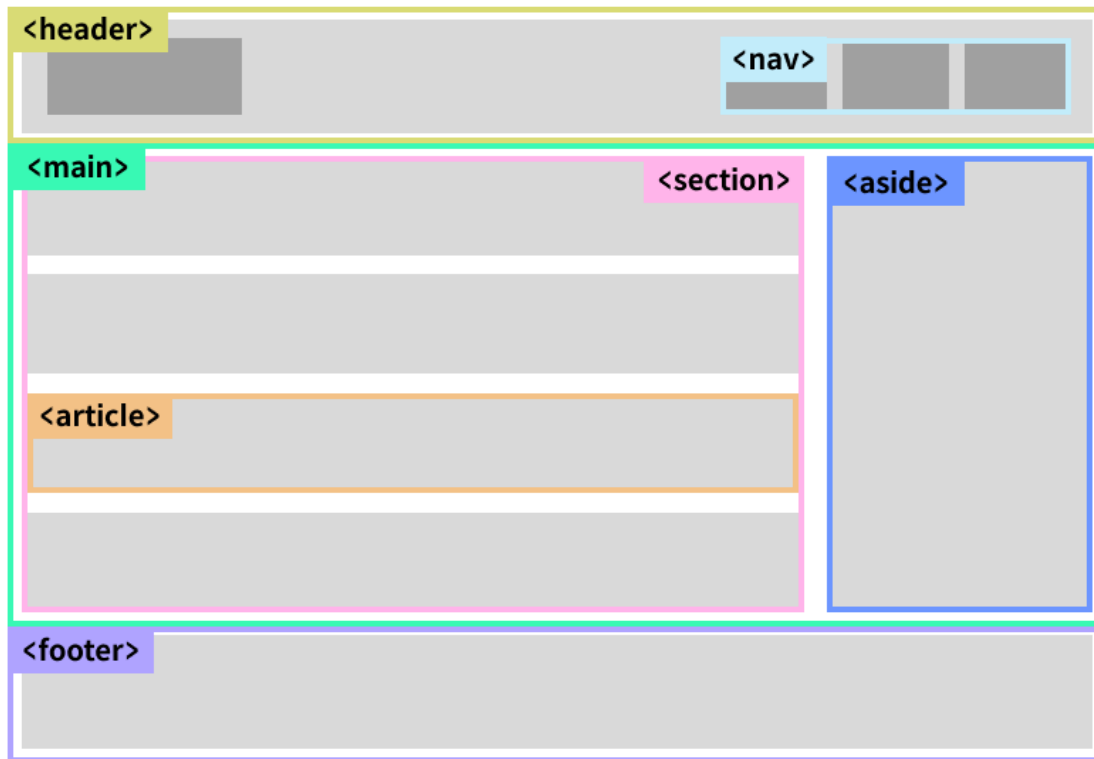
Semantic tag



〈aside〉

: 콘텐츠와 직접적인 연관이 없는 옆에 있는 부분 (ex. 광고 등)

Semantic tag



<article> : 하나의 의미있는 요소
(ex. 기사 리스트 중에서 기사 하나)

<section> : article을 포함 (그룹화 목적)

왜 Semantic 해야할까?

- **SEO** (Search Engine Optimization, 검색최적화)

- 검색엔진은 HTML 코드를 분석해 사이트를 노출시킨다
- 크롤러가 페이지의 구성 요소들이 무엇을 의미하는지 이해하여 사이트 노출에 영향을 준다.

- **유지보수성**

- 페이지를 유지보수 할 때, <div> 태그만 가득하다면 요소를 찾기 정말 힘들다..!!!
- 시멘틱 태그를 활용해 콘텐츠를 적절히 구분시키자!