

SeSAC 도봉 1기

웹 풀스택 과정

HTML



왜 이렇게 나누어져 있을까요?

HTML



CSS



JS





HTML : 집의 구조

CSS : 인테리어

javascript : 집의 기능

HTML 구조

<html>



<head>

</head>

<body>

</body>

</html>

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 `<body>` 태그의 정보를 읽어 들이기 위한 정보가 존재(제목, 저장 방식, 브라우저의 크기 등등)
- 웹 브라우저에 출력되는 모든 요소는 `<body>`와 `</body>` 사이에 위치한다

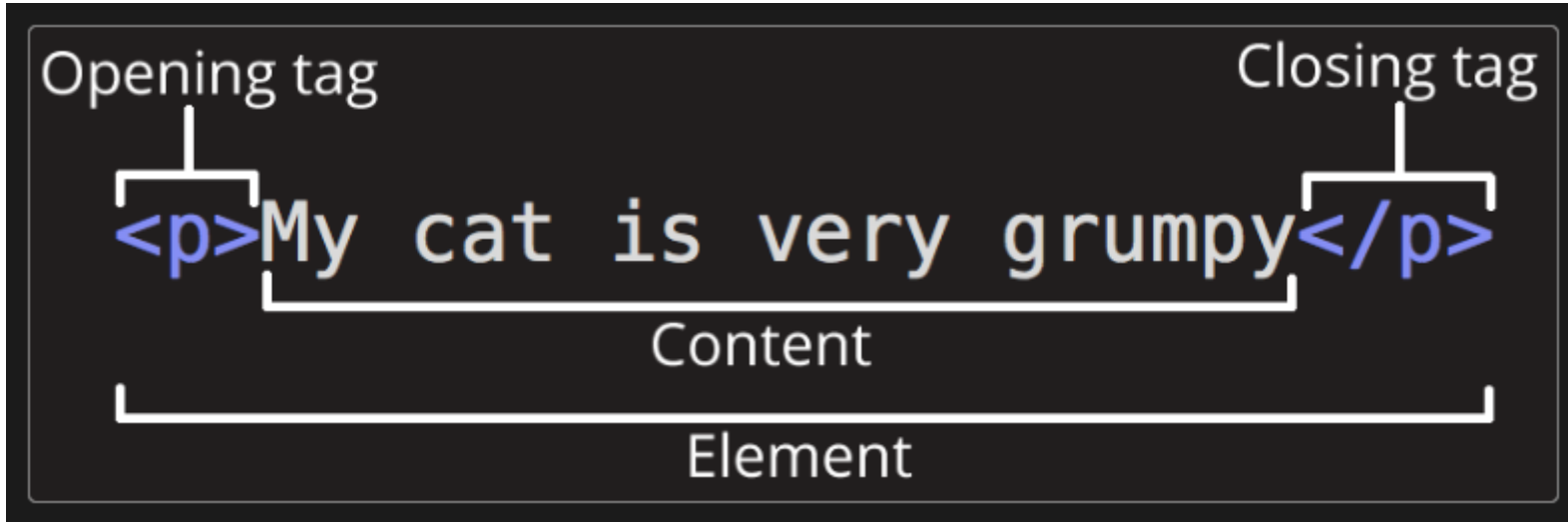
<head> 태그

- 메타 데이터(데이터를 위한 데이터)를 포함하기 위한 태그
- 웹 페이지에 유일하게 존재
- 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.

<body> 태그

- HTML 문서의 내용을 담는 태그
- 웹페이지를 구성하는 대부분의 요소가 body 태그 내에 기술

HTML 기본 문법, 요소(Element)

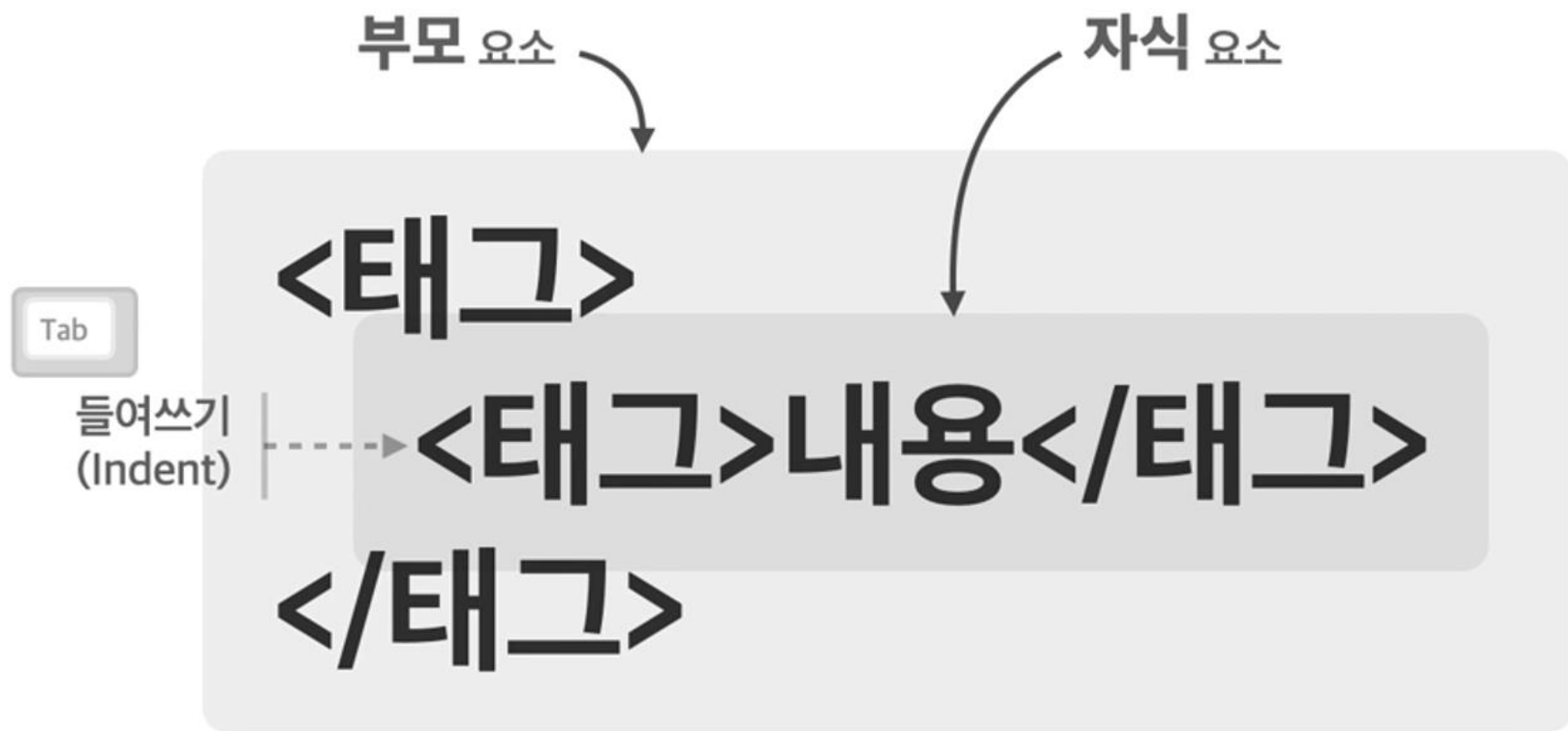


- HTML 요소(Element)는 시작 태그(Opening tag)와 종료 태그(Closing tag) 그리고 태그 사이에 위치한 내용(Content)로 구성

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.
- 예제를 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



<태그> 자식 요소

<태그>

<태그>내용</태그>

</태그>

</태그>

<태그> 부모 요소

<태그>

<태그>내용</태그>

</태그>

</태그>

<태그> 하위(후손) 요소

<태그>

<태그>내용</태그>

</태그>

</태그>

<태그> 상위(조상) 요소

<태그>

<태그>내용</태그>

</태그>

</태그>

HTML 기본 문법, 빈 요소(Empty)

```
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>
```

- 내용이 없는 요소 (내용을 가질 수 없거나, 필요 X)
- 빈 요소는 내용이 없으며, 속성만 소유
- 4번 라인의 `<meta>` 가 빈 요소!
- `
`, `<hr>`, ``, `<input>`, `<link>`, `<meta>` 등이 존재

편리함!

HTML 1/2/3/4/5

<태그>

VS

<태그 />

안전함!

XHTML / HTML5

HTML 기본 문법, 속성(Attribute)

Attribute Value

<태그 속성="값">내용</태그>

기능의 확장

- 속성(Attribute)이란 요소의 성질, 특징을 정의
- 속성은 요소에 추가적 정보(예를 들어 이미지 파일의 경로, 크기 등)를 제공한다.

HTML 기본 문법, 주석(Comments)

```
1  <!-- 주석입니다. -->
2  <!-- 웹페이지에 표시되지 않아요. -->
3  <p>Hello, World</p>
```

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

tag 종류

제목 태그, <h1~6></h1~6>

- 제목을 뜻하는 Heading 의 약자, h 사용!
- 제목이니까, 자동 줄 바꿈!
- 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, `
`

- 줄을 바꿔줍니다!
- 그런데.... VS Code에서 그냥 엔터 치면 안되나요?
- 그럼 공백은요?

목록, `` or ``

- `` 순서 없는 목록
- `` 순서 있는 목록
- `` 은 속성 사용 가능
 - `<ol type="?">` : 말머리 기호 변경(1, A, a, I, i)
 - 1 / A / a / I / i
 - `<ol start="?">` : 시작 값 변경
 - `<ol reversed>` : 역순으로 시작

수평 줄, `<hr>`

- 수평으로 된 줄을 그어 줍니다!

<summary>와 <detail>

▶ 내용을 숨길 수 있어요.

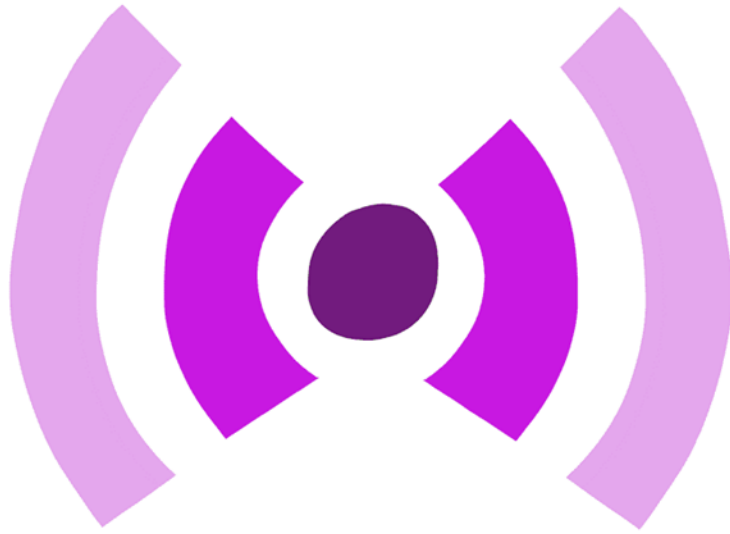
summary를 클릭해서 숨긴 내용을 보이게 하거나 감출 수 있습니다.

```
<details>  
  <summary>내용을 숨길 수 있어요.</summary>  
  숨긴 내용은 이 곳에 쓰면 됩니다.  
</details>
```

문자 꾸미기 태그들

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

Live Server



도움이 되는 vscode 확장 프로그램 모음



Auto Rename Tag v0.1.10

Jun Han | 13,825,940 | ★★★★★ (185)

Auto rename paired HTML/XML tag

Disable | Uninstall | ⚙️

This extension is enabled globally.



Korean Language Pack for Visual Studio Code

Microsoft | microsoft.com | 2,406,825 | ★★★★★ (9)

Language pack extension for Korean

Uninstall | ⚙️

This extension is enabled globally.



Live Server v5.7.9

Ritwick Dey | 35,384,710 | ★★★★★ (444)

Launch a development local Server with live reload...

Disable | Uninstall | ⚙️

This extension is enabled globally.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>

</body>
</html>
```

<!DOCTYPE html>

- 웹문서가 어떤 형식으로 작성되었는지 선언하는 것 (document type)
- html5형식으로 작성되었다고 브라우저에 알려주는 것!
- html 문서 최상단에 위치
- <https://rypro.tistory.com/187>

<meta> 태그

- meta 정보?
 - 정보를 위한 정보! 실제 브라우저에 내용을 보여주기 위한 것이 아닌 브라우저 정보를 담고 있음
- 보여주기 위한 것이 아니므로 head 태그 안에 작성

meta charset="UTF-8"

- 유니코드(문자) 인코딩 방식: UTF-8
- 유니코드? 인코딩?
 - 유니코드 : 여러 나라의 문자들을 표시할 수 있는 표준 코드
'가' : 'U+AC00'
 - 인코딩 : 컴퓨터가 읽을 수 있도록 변환해주는 것
컴퓨터가 읽을 수 있는 1001 등의 이진수로 변환

meta name="viewport" content="width=device-width, initial-scale=1.0"

- width=device-width
 - 브라우저를 보는 장치의 화면 너비를 따르도록 페이지 너비를 설정
- initial-scale=1.0
 - 브라우저에서 페이지를 처음 로드 할 때 크기 설정

유용한 meta 태그 모음

- <https://webclub.tistory.com/354>

크로스 브라우징 (cross browsing)

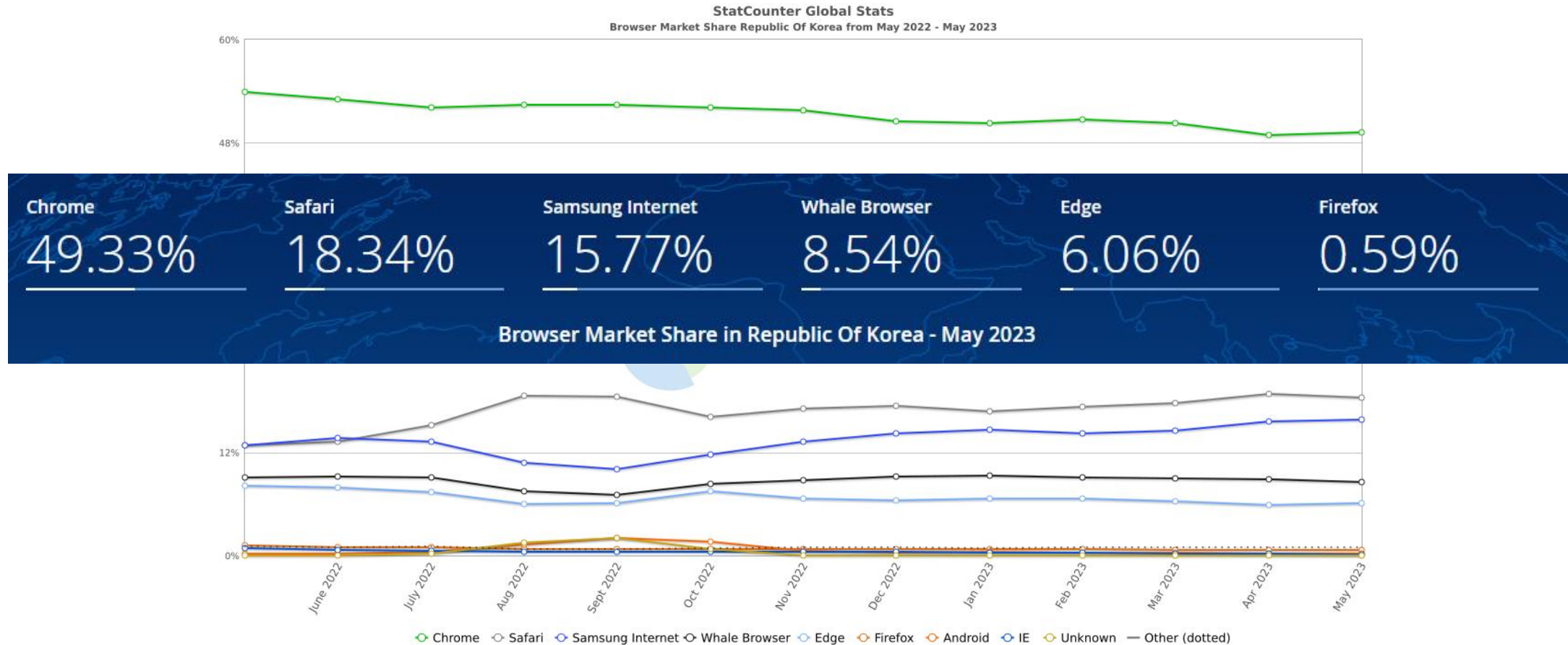
- 하나의 페이지가 많은 종류의 웹 브라우저에서 동작하도록 하는 것
 - 웹 브라우저: chrome, firefox, opera, naver whale, edge, ..
- 보통은 브라우저의 시장 점유율에 따라서 주가 되는 브라우저가 결정됨

브라우저 시장점유율 (World Wide)



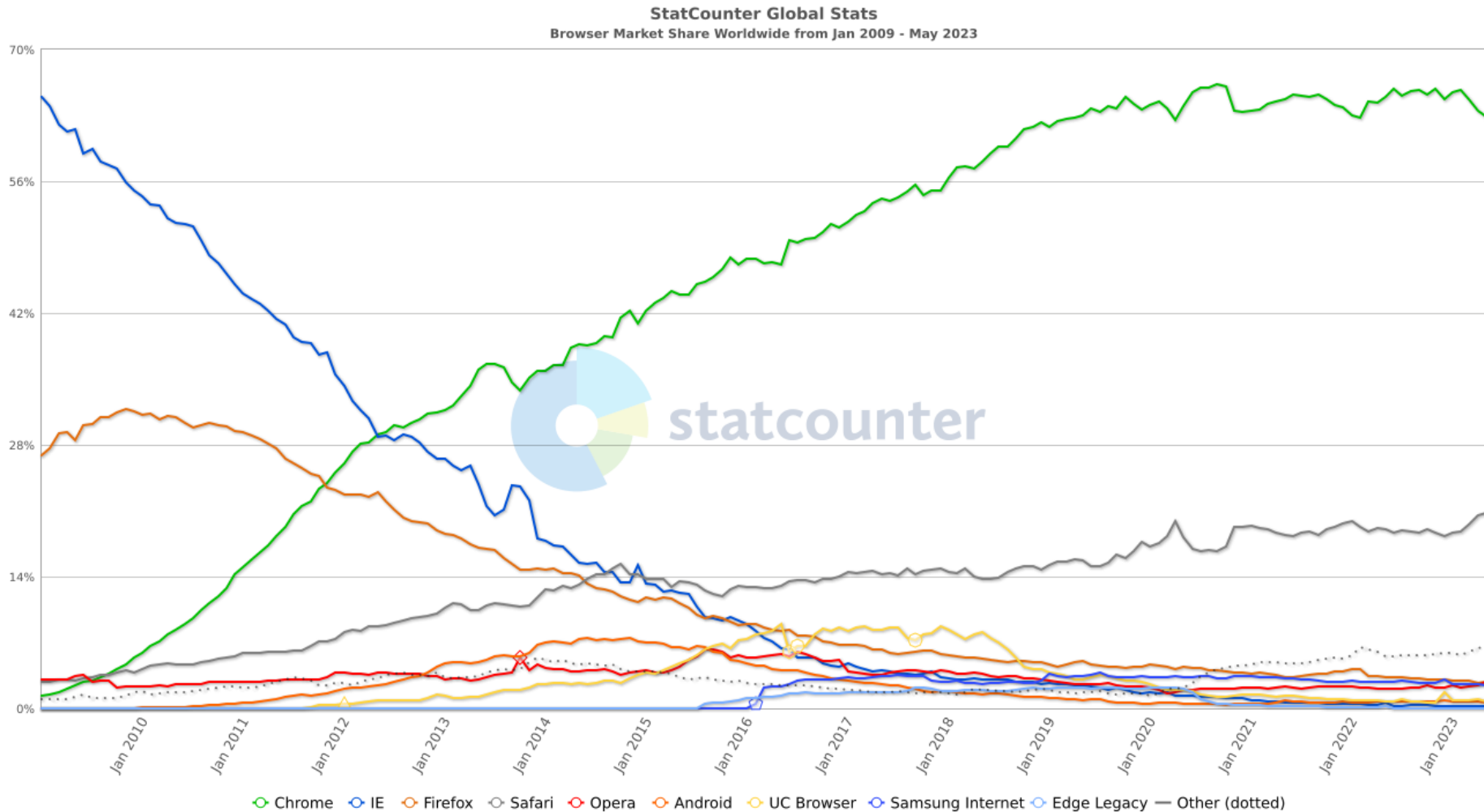
<https://gs.statcounter.com/>

브라우저 시장점유율 (한국)



<https://gs.statcounter.com/browser-market-share/all/south-korea#monthly-202205-202305>

브라우저 점유율 추이(World Wide)



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

하이퍼링크 <a>

- 기존 문서나 텍스트의 제약에서 벗어나 사용자가 원하는 정보를 취득할 수 있는 기능을 제공
- 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 / _self

```
<a href="https://www.naver.com" target="_blank"> Click </a>
```

사용자 입력

Form 태그

입력 값 받기! `<input>`

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

버튼, type="button"

- 버튼을 생성
- 주로 특정 기능을 수행 시킬 때 사용

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
 - Checked : 화면 최초 로딩 시에 선택 된 상태로 로딩

라디오 버튼, type="radio"

- 여러 선택지 중 **하나만 선택 가능**한 라디오 버튼 생성
- 속성
 - **Name** : 라디오 버튼의 이름, 같은 name 을 가지는 라디오 버튼은 하나만 선택이 가능 → 하나를 선택하면 다른 선택 값이 취소 됨
 - **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!

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
- 과거에는 테이블을 사용하여 레이아웃을 구성 하였으나 기능 상의 한계로 인하여 최근에는 공간 분할 태그인 <div> 태그를 사용
- 먼저 행을 쓰고 행의 자식 요소로 칸을 넣어주는 것이 기본!
- <table> : 표를 감싸는 태그
- <tr> : 표 내부의 행
- <th> : 표 내부의 제목 칸
- <td> : 행 내부의 일반 칸

`<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>`

<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
Semantic하게 작성되어야 한다.

“웹 페이지를 이루는 요소에 **의미와 맞는 태그**를 사용하자!”

Semantic tag

<header>

<footer>

<main>

<h1> ~ <h6>

<nav>

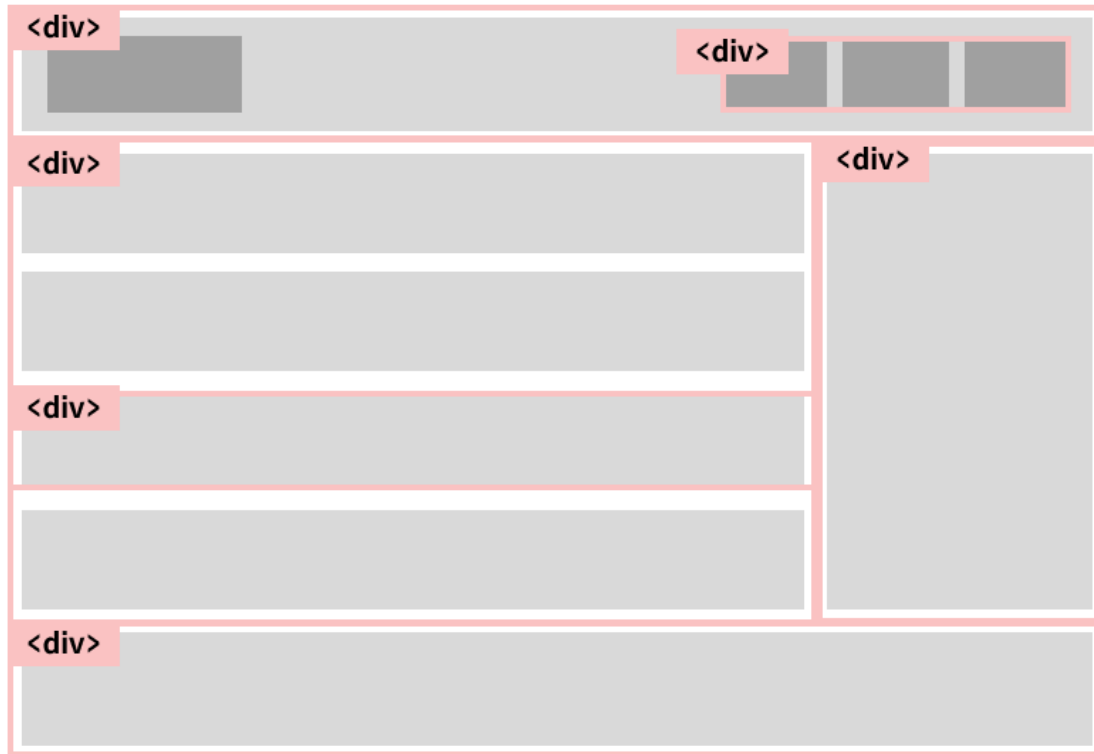
<aside>

<article>

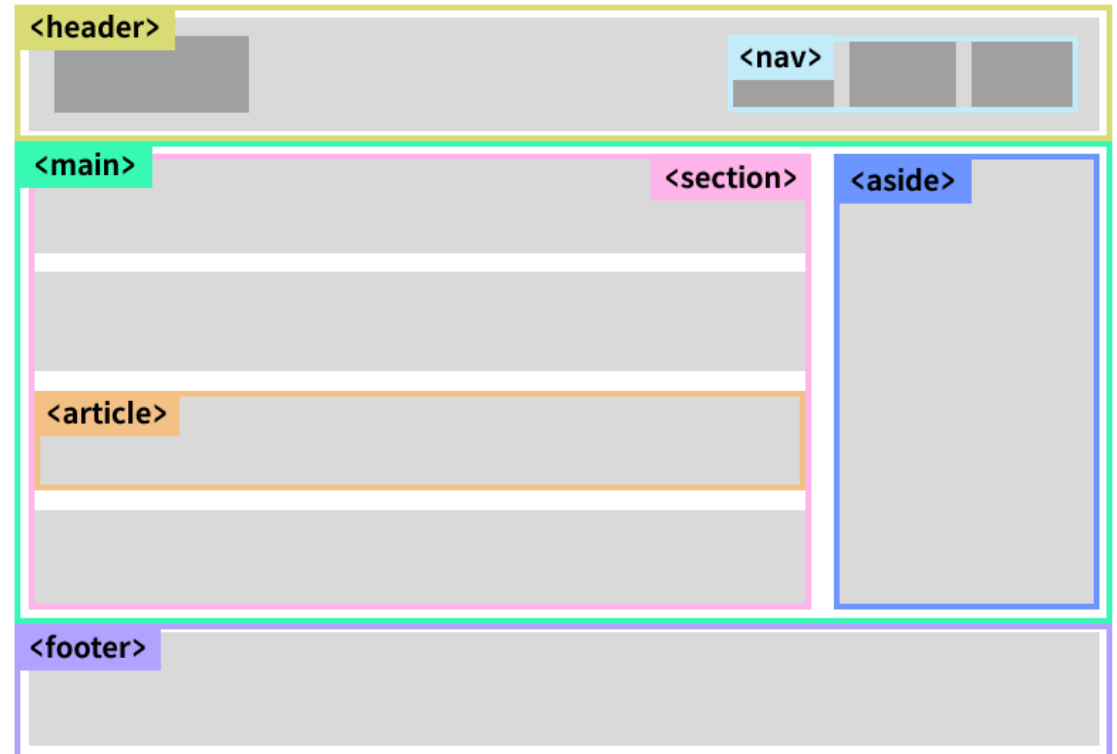
<section>

Semantic tag

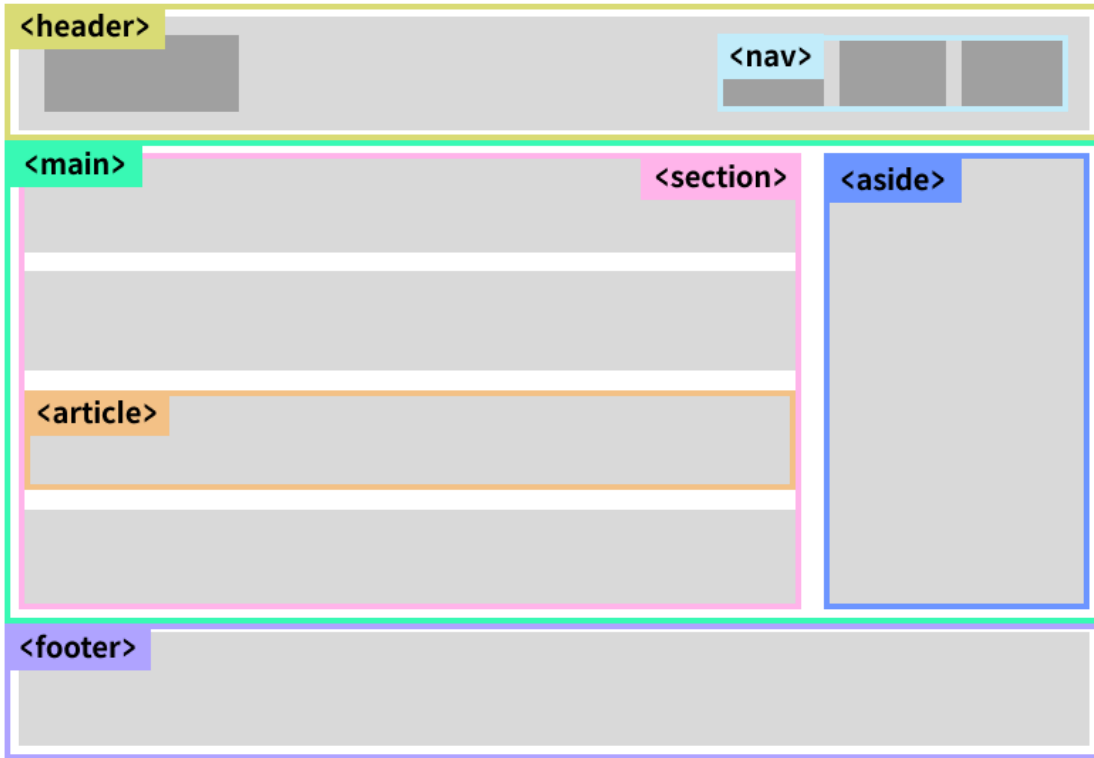
before



after



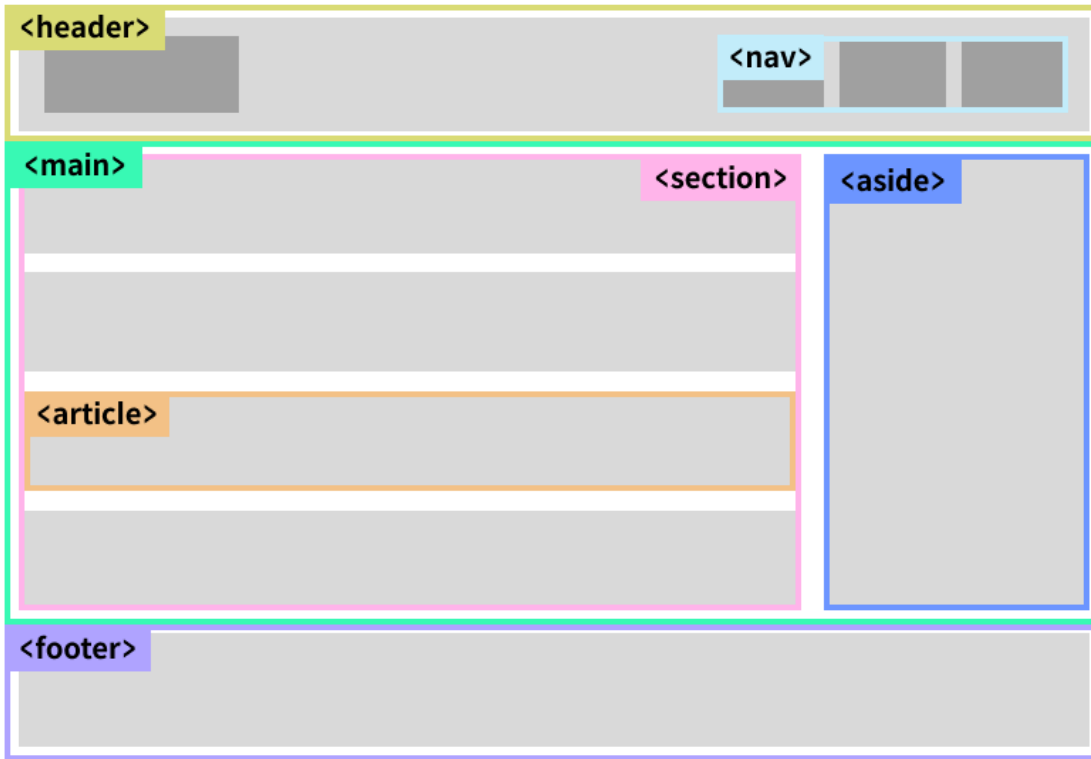
Semactic tag



<header> : 헤더 (로고, 메뉴 아이템 등)

<footer> : 푸터 (하단 영역)

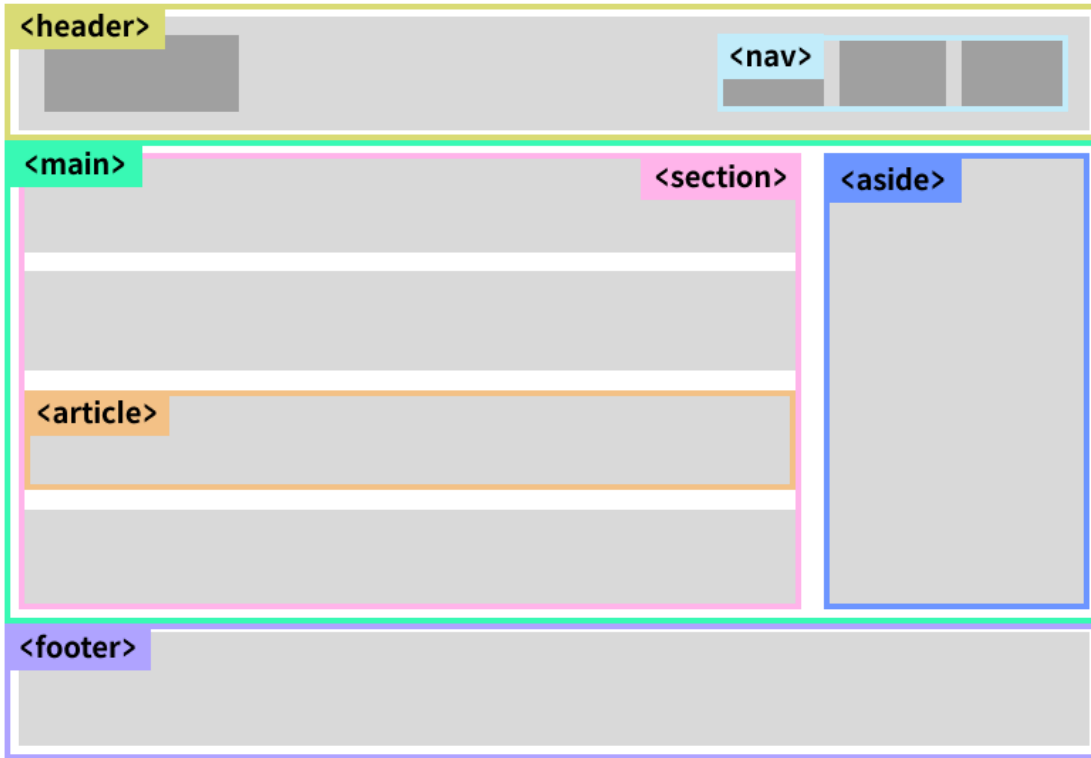
Semantic tag



<nav>

: 네비게이션 (다른 곳으로 이동 가능한 링크들)

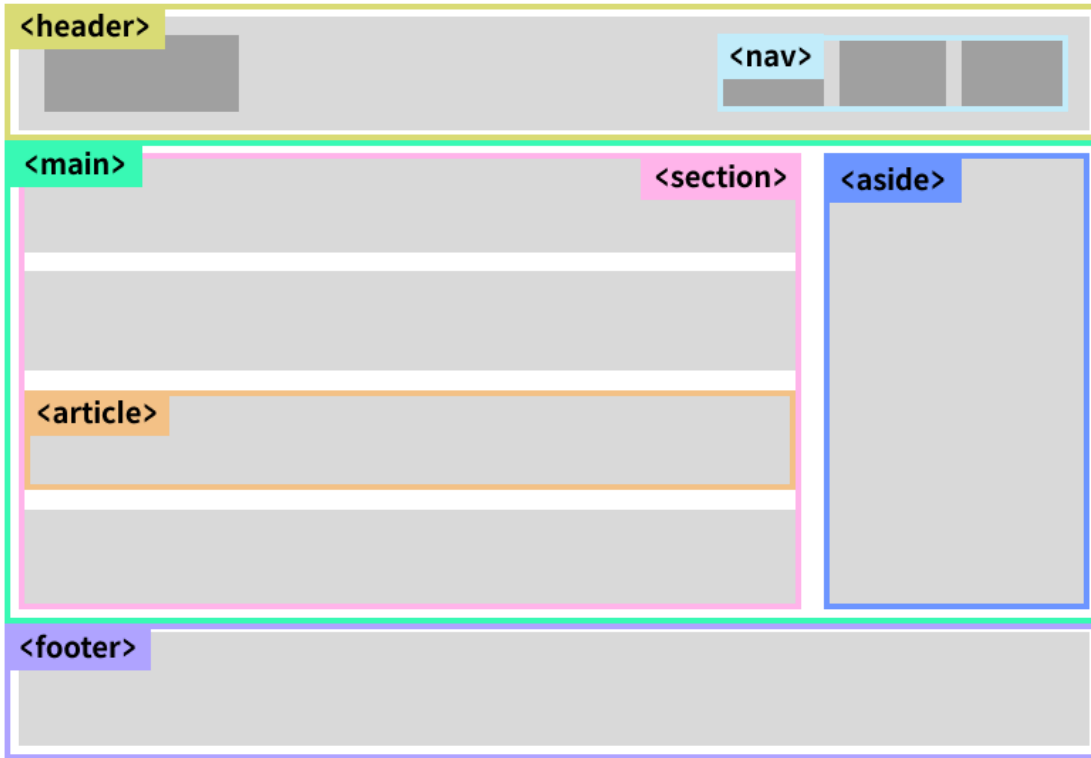
Semantic tag



<main>

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

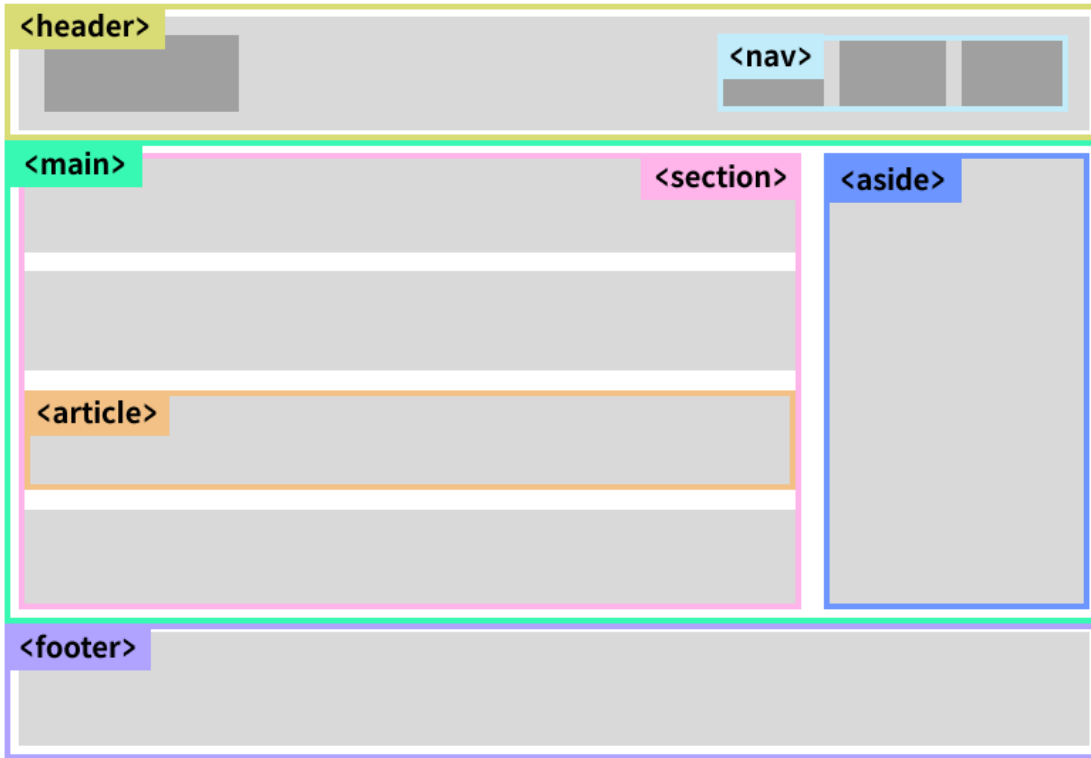
Semantic tag



<aside>

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

Semactic tag



<article> : 하나의 의미있는 요

소 (ex. 기사 리스트 중에서 기사 하나)

<section> : article을 포함 (그룹화 목적)

왜 Semactic 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 힘듭니다!
 - Semantic tag를 활용해 콘텐츠를 적절히 구분시키자!