

HTML

🕒 작성일시	@2022년 8월 1일 오전 11:05
☰ 내용	HTML 웹
▼ 주차	3주차
📎 자료	

HTML이란

- **Hyper Text Markup Language**

- **Hyper Text**

- 참조(하이퍼링크)를 통해 사용자가 한 문서에서 다른 문서로 즉시 접근할 수 있는 텍스트

- **Markup Language**

- 태그 등을 이용하여 문서나 데이터의 구조를 명시하는 언어
 - HTML, Markdown

- 웹페이지를 작성, 구조화 하기 위한 언어

HTML 스타일 가이드

```
<body>
  <h1> 웹문서 </h1>
  <ul>
    <li>HTML</li>
    <li>CSS</li>
  </ul>
</body>
```

마크업 스타일 가이드(2 space)

HTML 기본구조

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
</head>
<body>

</body>
</html>
```

- html: 문서의 최상위(root)요소
- head: 문서 메타데이터 요소
 - 문서 제목, 인코딩, 스타일, 외부파일 로딩 등
 - 일반적으로 브라우저에 나타나지 않는 내용
- body:
 - 실제 화면 구성과 관련된 내용

Head

```
<head>
  <title>HTML 수업</title>
  <meta charset="UTF-8">
  <link href="style.css" rel="stylesheet">
  <script src="javascript.js"></script>
  <style>
    p {
      color: black;
    }
  </style>
</head>
```

- <title>: 브라우저 상단 타이틀

- <meta>: 문서 레벨 메타데이터 요소
- <link>: 외부 리소스 연결 요소 (CSS파일, favicon)
- <script>: 스크립트 요소(Javascript 파일/코드)
- <style>: CSS 직접 작성

요소(element)

- HTML의 요소는 태그와 내용(contents)로 구성되어 있다.

```
<h1>contents</h1>
```

- HTML 요소는 시작태그와 종료태그 그리고 태그 사이에 위치한 내용으로 구성
 - 요소는 태그로 콘텐츠(내용)을 감싸는 것으로 그 정보의 성격과 의미를 정의
 - 내용이 없는 태그들도 존재(닫는 태그가 없음)
 - br, ht, img, input, link, meta
- 요소는 중첩(nested)될 수 있음
 - 요소의 중첩을 통해 하나의 문서를 구조화
 - 여는 태그와 닫는 태그의 쌍을 잘 확인해야 함
 - 오류를 반환하는 것이 아닌 그냥 레이아웃이 깨진 상태로 출력되기 때문에, 디버깅이 힘들 수 있음

속성(attribute)

- 태그별로 사용할 수 있는 속성은 다르다.

```
<a href="https://google.com"></a>
  공백 x          "(쌍따옴표)사용!
```

- 속성을 통해 태그의 부가적인 정보를 설정할 수 있음
- 요소는 속성을 가질 수 있으며, 경로나 크기와 같은 추가적인 정보를 제공
- 요소의 시작 태그에 작성하며 보통 이름과 값이 하나의 쌍으로 존재
- 태그와 상관없이 사용 가능한 속성(HTML Global Attribute)도 있음
 - **HTML Global Attribute**

- **id**: 문서 전체에서 유일한 고유 식별자 지정(CSS)
- **class**: 공백으로 구분된 해당 요소의 클래스 목록
- **data-***: 페이지에 개인 사용자 정의 데이터를 저장하기 위해 사용
- **style**: inline 스타일
- **title**: 요소에 대한 추가 정보 지정
- **tabindex**: tab 순서 지정

HTML 코드 예시

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
</head>
<body>
  <!-- 이것은 주석입니다. -->
  <h1>나의 첫번째 HTML</h1>
  <p>이것은 본문입니다.</p>
  <span>이것은 인라인요소</span>
  <a href="https://www.naver.com">네이버로 이동!!</a>
</body>
</html>
```

시맨틱 태그

- HTML 태그가 특정 목적, 역할 및 의미적 가치(semantic value)를 가지는 것
 - 예를 들어 h1 태그는 “이 페이지에서 최상위 제목”인 텍스트를 감싸는 역할(또는 의미)를 나타냄
- Non semantic 요소로는 div, span 등이 있으며 a, form, table 태그들도 시맨틱 태그로 볼 수 있음
- HTML5에서는 기존에 단순히 콘텐츠의 구획을 나타내기 위해 사용한 div 태그를 대체하여 사용하기 위해 의미론적 요소를 담은 태그들이 추가됨
- 대표적인 시맨틱 태그 목록
 - header: 문서 전체나 섹션의 헤더(머리말 부분)

- nav: 내비게이션
- aside: 사이트에 위치한 공간, 메인 콘텐츠와 관련성이 적은 콘텐츠
- section: 문서의 일반적인 구분, 콘텐츠의 그룹을 표현
- article: 문서, 페이지, 사이트 안에서 독립적으로 구분되는 영역
- footer: 문서 전체나 섹션의 푸터(마지막 부분)

```
<div>
  <div></div>
</div>
<div>
  <div></div>
  <div></div>
</div>
<div></div>
```

```
<header>
  <nav></nav>
</header>
<section>
  <article></article>
  <article></article>
</section>
<footer></footer>
```

시맨틱 태그를 사용해야 하는 이유

- 의미론적 마크업
 - 개발자 및 사용자 뿐만 아니라 검색엔진 등에 의미있는 정보의 그룹을 태그로 표현
 - 단순히 구역을 나누는 것 뿐만 아니라 ‘의미’를 가지는 태그들을 활용하기 위한 노력
 - 요소의 의미가 명확해지기 때문에 코드의 가독성을 높이고 유지보수를 쉽게 함
 - 검색 엔진 최적화(SEO)를 위해서 메타태그, 시맨틱 태그 등을 통한 마크업을 효과적으로 활용해야함

텍스트로 작성된 코드가 어떻게 웹 사이트가 되는 걸까?

- 렌더링(Rendering)
 - 웹사이트 코드를 사용자가 보게 되는 웹사이트로 바꾸는 과정
- DOM(Document Object Model)트리
 - 텍스트 파일인 HTML 문서를 브라우저에서 렌더링 하기 위한 구조
 - HTML 문서에 대한 모델을 구성함
 - HTML 문서 내의 각 요소에 접근 / 수정에 필요한 프로퍼티와 메서드를 활용함

```
<body>  
  <h1> 웹문서 </h1>  
  <ul>  
    <li>HTML</li>  
    <li>CSS</li>  
  </ul>  
</body>
```

```
<body>  
  <h1> 웹문서 </h1>  
  <ul>  
    <li>HTML</li>  
    <li>CSS</li>  
  </ul>  
</body>
```