

2. HTML 기본 문서 구조

2.1 `<!DOCTYPE>` 선언

HTML 문서의 최상단에는 항상 아래와 같은 선언이 들어간다:

```
1 | <!DOCTYPE html>
```

이 선언은 브라우저에게 현재 문서가 어떤 HTML 표준을 따르고 있는지 알려주는 선언부다.

웹 페이지가 브라우저에서 정확하고 일관되게 렌더링되기 위한 필수 요소로, HTML에서 가장 먼저 이해해야 할 기본 구조 중 하나이다.

✓ `<!DOCTYPE>` 란?

- "Document Type Declaration" 의 줄임말
- 이 문서가 어떤 HTML 버전과 문법 규칙(DTD, 문서 유형 정의)을 따르는지 명시
- 브라우저가 문서를 표준 모드(Standards mode) 로 해석할지, 호환 모드(Quirks mode) 로 해석할지를 결정함

✓ HTML5에서의 `<!DOCTYPE>` 선언

```
1 | <!DOCTYPE html>
```

- HTML5에서는 위 한 줄로 충분함
- 과거보다 간단하고 짧아짐
- DTD(Document Type Definition) 와의 연결이 사라짐
- 모든 현대 브라우저는 이 선언을 표준 HTML5 모드로 인식함

🚩 선언이 없는 경우 어떻게 될까?

`<!DOCTYPE>` 가 생략되거나 잘못된 경우:

- 브라우저가 Quirks Mode (후진 호환 모드) 로 렌더링
- 레이아웃 해석 방식이 달라짐 → CSS가 비정상 동작할 수 있음
- 오래된 브라우저 호환을 위해 남아있는 비표준 처리 로직이 작동됨

예:

- `box-sizing`, `width`, `margin` 등의 계산 방식이 달라질 수 있음
- IE에서 두드러지게 발생

✓ 과거 버전의 DOCTYPE 예시 비교

버전	선언 방식
HTML 4.01 Strict	<code><!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd"></code>
HTML 4.01 Transitional	<code><!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd"></code>
XHTML 1.0 Strict	<code><!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"></code>
XHTML 1.0 Transitional	<code><!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"></code>
HTML5	<code><!DOCTYPE html></code> ← 단 하나로 통일

🔥 HTML5부터는 문서 유형 선언이 더 이상 복잡하지 않다.

오직 `<!DOCTYPE html>` 만 사용한다.

✓ 작동 원리 간략 요약

- 1 브라우저가 HTML 문서를 로딩할 때 :
- 2
- 3 1. `<!DOCTYPE>` 선언을 확인
- 4 2. 선언이 정확하면 → Standards Mode
- 5 3. 선언이 누락되었거나 틀리면 → Quirks Mode

→ 이것이 페이지가 브라우저에서 의도한 대로 보일지 여부를 결정하는 **출발점**이다.

✓ Quirks Mode vs Standards Mode 비교

항목	Quirks Mode	Standards Mode
CSS Box Model 계산	비표준 방식 (IE5 등 구버전 기준)	W3C 표준 방식
글꼴 렌더링	다르게 보일 수 있음	일관된 폰트, 간격 유지
렌더링 호환성	오래된 사이트를 위해 남겨둠	현대 웹사이트의 기본 동작 방식
의도한 디자인 반영	어려움	정확한 스타일 재현 가능

✓ 실제 예제

```
1 <!DOCTYPE html>
2 <html lang="ko">
3 <head>
4   <meta charset="UTF-8">
5   <title>DOCTYPE 테스트</title>
6   <style>
7     div {
8       width: 200px;
9       padding: 20px;
10      border: 10px solid black;
11    }
12  </style>
13 </head>
14 <body>
15   <div>표준 Box Model 예제</div>
16 </body>
17 </html>
```

✦ <!DOCTYPE html> 이 있으면

총 너비 = width + padding + border = 200 + 40 + 20 = 260px

없으면 일부 브라우저에서는 예전 방식(width가 전체 너비로 계산됨)으로 렌더링됨

✓ 한 줄 요약

<!DOCTYPE html> 선언은 브라우저가 HTML5로 올바르게 렌더링하도록 보장하는 핵심적인 첫 줄이다.

항상 문서 맨 위에 정확히 선언하자.

2.2 <html>, <head>, <body> 태그의 의미와 사용법

HTML 문서는 크게 세 가지 주요 블록으로 나뉜다:

바로 <html>, <head>, <body> 이다.

이 세 가지 태그는 HTML 문서의 전체 구조를 구성하는 핵심 골격이다.

✓ 전체 구조 예시

```
1 <!DOCTYPE html>
2 <html lang="ko">
3   <head>
4     <meta charset="UTF-8">
5     <title>문서 제목</title>
6   </head>
7   <body>
8     <h1>페이지 본문</h1>
9     <p>이곳은 콘텐츠가 보이는 영역입니다.</p>
10  </body>
11 </html>
```

◆ 1. <html> 태그

✓ 의미

- HTML 문서 전체를 감싸는 **루트 요소 (Root Element)**
- 반드시 <head> 와 <body> 를 포함해야 함
- HTML5에서는 lang 속성을 통해 문서의 기본 언어를 지정함

✓ 문법

```
1 <html lang="ko">
2   ...
3 </html>
```

✓ 주요 속성

속성	설명
lang	문서의 기본 언어 설정 (예: ko, en, ja)
xmlns	XML 네임스페이스 (XHTML 사용 시 필요) — HTML5에선 사용 X

✓ 활용 예

```
1 <html lang="en">
2   ...
3 </html>
```

◆ 2. <head> 태그

✓ 의미

- 브라우저에 의해 **표시되지 않는 설정 정보**(메타데이터)를 담는 영역
- 문서의 제목, 문자 인코딩, 외부 리소스, SEO 관련 정보 등을 선언

✓ 구성 가능한 대표 요소들

태그	설명
<meta>	문자셋, 문서 설명, 뷰포트 등 메타 정보
<title>	브라우저 탭 제목, 즐겨찾기 제목
<link>	외부 CSS, 파비콘, 웹폰트 등 연결
<style>	문서 내부 스타일 정의
<script>	JavaScript 파일 또는 코드 연결

✓ 예시

```
1 <head>
2   <meta charset="UTF-8" />
3   <title>나의 블로그</title>
4   <link rel="stylesheet" href="main.css" />
5   <script src="main.js" defer></script>
6 </head>
```

◆ 3. <body> 태그

✓ 의미

- 브라우저 화면에 표시되는 실제 콘텐츠를 담는 영역
- 사용자가 보는 모든 것: 텍스트, 이미지, 버튼, 링크 등 포함
- JavaScript에 의해 조작되는 DOM 객체 대부분이 이 안에 위치함

✓ 포함할 수 있는 대표 요소들

콘텐츠 종류	대표 태그
제목	<h1> ~ <h6>
문단	<p>
이미지	
링크	<a>
리스트	, ,
입력 양식	<form>, <input>, <textarea>, <select>
구조 구획	<div>, <section>, <article>
스크립트 실행	<script>

✓ 예시

```
1 <body>
2   <h1>환영합니다!</h1>
3   <p>이 페이지는 HTML 구조 학습을 위한 예제입니다.</p>
4   
5 </body>
```

✓ DOM 관점에서의 구조 요약

```
1 <html>
2   └─ <head>   → 메타 정보 (보이지 않음)
3   └─ <body>   → 사용자에게 보이는 콘텐츠
```

✓ 필수 유효성 체크

요소	필수 여부	설명
<html>	✓	모든 문서는 루트 태그로 감싸야 함
<head>	✓	문서 정보 필수 (없으면 비정상 렌더링 위험)
<body>	✓	콘텐츠 표시 영역, 생략하면 빈 페이지 출력 가능

✦ 일부 브라우저는 <head> 또는 <body> 생략을 자동 보정하지만, **표준 방식은 명시하는 것이다.**

✓ 전체 예제 (정석 구조)

```
1 <!DOCTYPE html>
2 <html lang="ko">
3   <head>
4     <meta charset="UTF-8" />
5     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6     <title>HTML 문서 구조 예제</title>
7     <link rel="stylesheet" href="style.css" />
8   </head>
9   <body>
10    <h1>HTML 기본 구조</h1>
11    <p>이 예제는 HTML의 뼈대를 설명하는 구조입니다.</p>
12  </body>
13 </html>
```

✓ 한 줄 요약

HTML 문서는 <html> 루트 아래
문서 설정용 <head>, 콘텐츠 표시용 <body>로 구성된다.
이 구조는 웹 페이지의 생명선이며, 모든 문서의 시작이다.

2.3 메타 정보 정의 (<meta>)

<meta> 태그는 HTML 문서의 <head> 영역에서 사용되며, 브라우저나 검색엔진, 소셜 미디어, 기타 외부 시스템이 문서를 올바르게 이해하고 처리할 수 있도록 정보를 제공하는 태그이다. 즉, <meta>는 사람이 보는 콘텐츠가 아니라, "문서에 대한 설명"을 기계에게 제공하는 역할을 한다.

✓ 기본 문법

```
1 | <meta name="..." content="..." />
```

또는

```
1 | <meta charset="UTF-8" />
```

✓ 대표적인 <meta> 종류 및 속성

속성	설명
charset	문자 인코딩 (ex: UTF-8) 지정
name	메타 정보의 이름 (description, author 등)
content	해당 정보의 실제 값
http-equiv	HTTP 헤더 역할 (캐시 제어 등)

◆ 1. 문자 인코딩 지정: charset

```
1 | <meta charset="UTF-8" />
```

- 문서의 문자 인코딩을 지정
- **UTF-8**은 전 세계 언어를 포괄하는 가장 보편적 인코딩
- 반드시 <head> 내 **최상단에 배치**해야 한다 (브라우저 파서가 먼저 읽어야 함)

◆ 2. 문서 설명: description

```
1 | <meta name="description" content="이 문서는 HTML의 meta 태그를 설명합니다.">
```

- 검색엔진에 표시되는 **검색결과 요약 문구**로 사용됨
- SEO(검색엔진 최적화)에 매우 중요
- 길이는 150자 내외 추천

◆ 3. 문서 키워드: keywords (과거 SEO)

```
1 | <meta name="keywords" content="HTML, meta, SEO, 웹 표준">
```

- 한때 검색엔진 최적화를 위해 사용되었지만,
- **현재는 대부분의 검색엔진이 무시함** (Google 등)

◆ 4. 문서 작성자: `author`

```
1 <meta name="author" content="오정석">
```

- 문서의 제작자 정보 명시

◆ 5. 뷰포트 설정 (모바일 대응)

```
1 <meta name="viewport" content="width=device-width, initial-scale=1.0" />
```

- 모바일/반응형 웹 필수 태그
- 뷰포트는 웹페이지를 어떤 크기로 보여줄지 설정하는 장치
- `width=device-width`: 기기 화면 너비 기준
- `initial-scale=1.0`: 확대/축소 배율

◆ 6. 캐시 제어: `http-equiv`

```
1 <meta http-equiv="Cache-Control" content="no-cache" />
```

- HTTP 응답 헤더처럼 동작
- 주로 자동 새로고침, 캐시 무효화 등에 사용

추가 예시:

```
1 <meta http-equiv="refresh" content="5; url=https://example.com" />
```

→ 5초 후 해당 URL로 자동 리디렉션

◆ 7. Open Graph (OG) 메타 태그 (SNS 공유)

```
1 <meta property="og:title" content="HTML 구조 강의" />
2 <meta property="og:description" content="HTML의 meta 태그에 대해 배워보자!" />
3 <meta property="og:image" content="https://example.com/image.jpg" />
4 <meta property="og:url" content="https://example.com/meta.html" />
```

- Facebook, Twitter, Kakao 등 SNS에서 페이지를 공유할 때
 - 제목, 설명, 썸네일을 보여주기 위해 필요
 - `property="og:..."` 는 표준 Open Graph 프로토콜
-

◆ 8. 기타 유용한 <meta> 예시

목적	코드 예시
IE 호환성 모드 해제	<code><meta http-equiv="X-UA-Compatible" content="IE=edge" /></code>
테마 색상 지정 (모바일 브라우저 탭 색)	<code><meta name="theme-color" content="#4285f4" /></code>
페이지 로딩 시간 측정 차단	<code><meta http-equiv="Pragma" content="no-cache" /></code>

✓ 전체 예시

```
1 <head>
2   <meta charset="UTF-8" />
3   <meta name="viewport" content="width=device-width, initial-scale=1.0" />
4   <meta name="description" content="HTML meta 태그의 모든 것을 배운다" />
5   <meta name="author" content="오정석" />
6   <meta property="og:title" content="HTML 문서 구조 강의" />
7   <meta property="og:image" content="https://example.com/thumb.jpg" />
8   <title>meta 태그 예제</title>
9 </head>
```

✓ 메타 태그 체크리스트

항목	권장 여부	비고
<code><meta charset="UTF-8"></code>	✓ 필수	반드시 맨 위에 위치
<code><meta name="viewport"></code>	✓ 필수	모바일 대응
<code><meta name="description"></code>	✓ 권장	SEO 최적화
<code><meta name="keywords"></code>	✗	현재는 효과 없음
<code><meta property="og:..."></code>	✓ 선택	SNS 공유 시 필수
<code>http-equiv="refresh"</code>	⚠ 신중	UX 측면에서 자동 리디렉션은 주의

✓ 한 줄 요약

<meta> 태그는 HTML 문서의 "머릿말"을 담당하며, 검색엔진, 브라우저, SNS 등 외부 시스템과의 상호작용에 있어 필수적인 정보의 전달자이다.

2.4 문서 제목 및 외부 리소스 (<title>, <link>, <script>)

HTML 문서의 <head> 안에는 브라우저나 외부 시스템에게 문서의 정보 및 외부 리소스를 연결하는 중요한 태그들이 위치한다.

이 항목에서는 문서의 제목을 정의하는 <title>, CSS와 파비콘 등을 연결하는 <link>, JavaScript를 연결하는 <script> 태그의 의미와 사용법을 자세히 설명한다.

✓ 1. <title> 태그 — 문서 제목

✓ 역할

- 브라우저의 탭 제목, 북마크 이름, 검색결과 제목 등으로 사용됨
- 반드시 <head> 안에 있어야 하며, 문서당 한 번만 등장

✓ 예시

```
1 <head>
2   <title>오정석의 HTML 강의</title>
3 </head>
```

🔥 브라우저 탭에는 "오정석의 HTML 강의" 라는 문구가 표시됨

✓ SEO 관점에서 중요성

항목	설명
검색결과 제목	대부분 <title> 태그 내용을 기반으로 생성됨
길이 제한	약 50~60자 이하가 권장됨
키워드 포함 권장	검색 최적화를 위해 핵심 키워드 포함하는 것이 좋음

✓ 2. <link> 태그 — 외부 리소스 연결

✓ 역할

- HTML 문서와 외부 리소스를 연결 (주로 CSS, 파비콘, 웹폰트 등)
- 빈 태그(self-closing) 형태로 사용 (<link />)

✓ 기본 문법

```
1 <link rel="stylesheet" href="style.css" />
```

속성	설명
<code>rel</code>	관계 (relationship) → <code>stylesheet</code> , <code>icon</code> , <code>preload</code> 등
<code>href</code>	연결할 파일의 경로
<code>type</code>	MIME 타입 (대개 생략 가능)
<code>media</code>	적용 조건 (예: <code>screen</code> , <code>print</code> , <code>max-width:600px</code>)

✓ 사용 예시들

👉 외부 CSS 연결

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

👉 파비콘(favicon) 설정

```
1 | <link rel="icon" href="/favicon.ico" type="image/x-icon" />
```

👉 구글 웹폰트 연결

```
1 | <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Roboto" />
```

👉 프린터 전용 CSS

```
1 | <link rel="stylesheet" href="print.css" media="print" />
```

👉 preload (성능 최적화)

```
1 | <link rel="preload" href="main.css" as="style" />
```

✓ 3. `<script>` 태그 — JavaScript 삽입/연결

✓ 역할

- 문서에 **JavaScript**를 삽입하거나 외부 파일을 불러옴
- `<body>` 또는 `<head>` 안에서 사용 가능
- 기본적으로 HTML 파싱을 차단함

✓ 사용 방식

👉 외부 스크립트 연결

```
1 <script src="main.js"></script>
```

👉 문서 내부에서 직접 작성

```
1 <script>
2   console.log('페이지가 로드됨');
3 </script>
```

✓ 주요 속성

속성	설명
<code>src</code>	외부 JS 파일 경로
<code>defer</code>	HTML 파싱 완료 후 실행 (순서 보장됨)
<code>async</code>	다운로드 즉시 실행 (순서 보장 안됨)
<code>type</code>	MIME 타입 (<code>text/javascript</code>) — 생략 가능

✓ `defer` vs `async` 차이

속성	설명
<code>defer</code>	HTML 파싱 끝난 후, 스크립트 실행. DOM 완성 후 실행됨. 순서 보장
<code>async</code>	다운로드 후 바로 실행. DOM보다 먼저 실행될 수 있음. 순서 보장 안됨

예시:

```
1 <script src="script1.js" defer></script>
2 <script src="script2.js" defer></script>
```

→ script1 → script2 순서 보장됨 (HTML 파싱 후 실행)

✓ 전체 예제

```
1 <head>
2   <meta charset="UTF-8" />
3   <title>HTML 리소스 연결 예제</title>
4   <link rel="stylesheet" href="styles.css" />
5   <link rel="icon" href="/favicon.ico" />
6   <script src="main.js" defer></script>
7 </head>
```

✓ 한눈에 정리

태그	용도	위치
<code><title></code>	브라우저 탭/검색 제목 지정	<code><head></code> 내부
<code><link></code>	CSS/폰트/파비콘 연결	<code><head></code> 내부
<code><script></code>	JavaScript 연결 또는 삽입	<code><head></code> 또는 <code><body></code>

✓ 한 줄 요약

`<title>`, `<link>`, `<script>`는 문서 정보와 외부 리소스를 정의하는 핵심 태그들로, 사용자 경험, 성능, 검색 최적화에 모두 큰 영향을 미친다.

2.5 문자 인코딩 설정 (charset="UTF-8" 등)

HTML 문서에서 문자 인코딩(charset)은 문자들이 어떻게 컴퓨터에서 저장되고 해석될지 결정하는 규칙이다. 잘못된 인코딩 설정은 한글 깨짐, 특수문자 오류, 브라우저 오작동 등의 원인이 되므로 반드시 올바르게 설정해야 한다.

✓ 문자 인코딩이란?

- 문자를 0과 1의 이진 코드로 변환하는 방식
- 예를 들어 "가" → 0xAC00 → UTF-8에서는 0xEAB080
- 각 인코딩은 문자마다 고유한 바이트값을 부여함

✓ 대표 문자 인코딩 종류

인코딩 이름	설명
UTF-8	전 세계 대부분의 언어를 표현 가능. 가변 길이(1~4바이트). ✓ 국제 표준
EUC-KR	한글 전용. 과거 한국 사이트에서 많이 사용
ISO-8859-1	서유럽 언어 중심. HTML 기본 인코딩이었음

인코딩 이름	설명
UTF-16	UCS 기반 유니코드. BOM(Byte Order Mark) 필요함

🔴 HTML5에서는 **UTF-8**을 강력히 권장하며, 사실상 표준이다.

✅ HTML에서 인코딩 설정 방법

```
1 <meta charset="UTF-8">
```

- **HTML5에서 가장 간결하고 표준적인 방식**
- `<head>` 내에서 **가장 상단에** 위치해야 브라우저가 올바르게 파싱 가능
- 과거 방식과의 비교:

```
1 <!-- 과거 방식 (HTML4) -->
2 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
```

→ HTML5에서는 더 이상 필요 없음

✅ 왜 UTF-8이 표준인가?

장점	설명
✅ 유니코드 기반	전 세계 언어 표현 가능
✅ ASCII 호환	기본 영문자와 호환성 유지
✅ 가변 길이 인코딩	저장 공간 효율적
✅ BOM(Byte Order Mark) 필요 없음	웹에서는 BOM 없이도 해석 가능
✅ 브라우저, 데이터베이스, JSON 등 거의 모든 기술에서 기본	범용성 탁월

✅ 잘못된 인코딩 설정의 예

❌ HTML은 UTF-8인데 파일 자체가 EUC-KR로 저장됨

결과:

```
1 <p>안녕하세요</p>
```

→  등으로 깨져 보임

✓ 해결 방법

- HTML 상단에 `<meta charset="UTF-8">` 명시
- 파일 자체도 UTF-8로 저장 (에디터에서 인코딩 지정 필요)

✓ 에디터에서 인코딩 설정 방법

도구/IDE	설정 위치
Visual Studio Code	파일 > 인코딩 > UTF-8로 저장
IntelliJ / WebStorm	Settings > Editor > File Encodings > UTF-8
Notepad++	인코딩 > UTF-8로 인코딩 선택
Sublime Text	File > Save with Encoding > UTF-8

✓ 실제 사용 예시 (전체 구조)

```
1 <!DOCTYPE html>
2 <html lang="ko">
3 <head>
4   <meta charset="UTF-8">
5   <title>문자 인코딩 예제</title>
6 </head>
7 <body>
8   <p>안녕하세요! 한글이 잘 보이시나요?</p>
9 </body>
10 </html>
```

✓ 브라우저 인코딩 오동작 원인

원인	설명
HTML에 charset 누락	브라우저가 인코딩을 추측함 (문제 발생 가능)
파일 저장 인코딩과 <code><meta></code> 불일치	HTML은 UTF-8인데, 실제 저장은 EUC-KR
서버 응답 헤더의 <code>Content-Type</code> 우선 적용	<code><meta></code> 보다 서버 헤더가 우선 적용되기도 함

✓ 서버에서도 인코딩 지정 필요

예: Apache `.htaccess`

```
1 | AddDefaultCharset UTF-8
```

예: Node.js 응답 헤더

```
1 | res.setHeader('Content-Type', 'text/html; charset=UTF-8');
```

✓ 한 줄 요약

`<meta charset="UTF-8">`은 모든 HTML 문서에서 가장 먼저 선언해야 하는 기본 중의 기본이다.
브라우저가 문서를 깨지지 않게 해석하기 위한 전제 조건이다.