

5. 링크와 네비게이션

5.1 하이퍼링크 ()

<a> 태그는 HTML의 핵심 기능인 '하이퍼링크'를 구현하는 태그이다. a는 "anchor(닻)"의 약자로, 다른 문서나 위치로 이동하는 고리를 만든다. 웹이 '웹'이라 불리는 이유는 바로 이 하이퍼링크로 문서가 연결되어 있기 때문이다.

✓ 기본 문법

```
1 | <a href="https://www.example.com">Example 사이트로 이동</a>
```

🔴 href 속성은 "hypertext reference"의 약자이며, 사용자가 클릭 시 이동할 **링크 주소(URL)**를 지정한다.

✓ 출력 예시

🔗 [Example 사이트로 이동](#)

(위 코드는 <a href>로 만든 링크를 브라우저에서 렌더링한 모습)

✓ 주요 속성 정리

속성	설명
href	연결할 주소 (필수)
target	링크를 어디에 열지 지정 (_blank, _self 등)
rel	링크의 관계 정의 (보안, SEO)
title	마우스를 올렸을 때 보이는 툴팁
download	클릭 시 파일 다운로드 (HTML5)

◆ href 값 종류

유형	예시	설명
절대 URL	https://example.com/page	외부 사이트 전체 주소
상대 URL	page.html / ../img/logo.png	현재 경로 기준의 파일
내부 북마크	#section1	같은 페이지 내 특정 위치로 이동
이메일 링크	mailto:me@example.com	이메일 클라이언트 열기

유형	예시	설명
전화 링크	<code>tel:010-1234-5678</code>	스마트폰에서 전화 앱 실행

◆ `target` 속성

값	설명
<code>_self</code>	현재 창에서 열기 (기본값)
<code>_blank</code>	새 탭(또는 창)에서 열기
<code>_parent</code>	부모 프레임에서 열기
<code>_top</code>	최상위 프레임에서 열기

예:

```
1 | <a href="https://example.com" target="_blank">새 탭에서 열기</a>
```

◆ `rel` 속성 (보안 및 의미 부여)

값	의미
<code>noopener</code>	새 탭 열 때 보안 취약점 방지
<code>noreferrer</code>	Referer 정보 전송 차단
<code>nofollow</code>	검색엔진이 링크를 따라가지 않도록 함
<code>external</code>	외부 링크임을 명시 (문서 의미 목적)

예:

```
1 | <a href="https://example.com" target="_blank" rel="noopener noreferrer">외부 링크</a>
```

◆ 다운로드 링크로 사용

```
1 | <a href="files/sample.pdf" download>파일 다운로드</a>
```

🔥 브라우저가 링크된 파일을 새 창이 아닌 다운로드하도록 동작

✓ 내비게이션 예시

```
1 <nav>
2   <ul>
3     <li><a href="/index.html">홈</a></li>
4     <li><a href="/about.html">소개</a></li>
5     <li><a href="/contact.html">문의</a></li>
6   </ul>
7 </nav>
```

🔥 `<a>` 는 보통 메뉴, 사이드바, 버튼 등 웹 인터페이스 전반에서 중심적인 역할

✓ 내부 북마크(스크롤 이동)

```
1 <a href="#section1">섹션1로 이동</a>
2
3 <h2 id="section1">섹션 1</h2>
```

🔥 `` 형식으로 같은 페이지 안에서 스크롤 이동 가능

✓ 스타일 커스터마이징 (기본 스타일 제거)

```
1 a {
2   text-decoration: none;
3   color: inherit;
4 }
```

🔥 기본 파란 밑줄 링크를 제거하고, 주변 텍스트와 스타일 통일 가능

✓ 접근성 및 시맨틱 측면

항목	내용
키보드 탐색	<code><a></code> 는 자동으로 Tab 탐색 가능
스크린 리더 대응	링크 텍스트를 명확하게 써야 함 (여기, 클릭 대신 "지원 페이지 보기" 등)
버튼과의 차이	<code><a></code> 는 이동, <code><button></code> 은 동작 수행에 사용

✓ 잘못된 예시

```
1 <a>여기 클릭</a>           <!-- href 없음 → 클릭 안 됨 -->
2 <a href="#">클릭</a>       <!-- 목적 없는 링크 → 접근성 나쁨 -->
3 <a href="page.html">       <!-- 닫는 태그 없음 → 오류 가능성 -->
```

✓ 한 줄 요약

``는 HTML 문서 간 연결을 위한 하이퍼링크의 기본 단위로, 웹의 본질인 연결성과 탐색을 실현하는 핵심 태그이다.

5.2 절대경로 vs 상대경로

✓ 정의

- 전체 URL 또는 루트(/)부터 시작하는 경로
- 항상 같은 위치를 가리킴
- 어디서든 접근 가능

✓ 예시

```
1 <a href="https://example.com/about.html">회사 소개</a>
2 
```

✓ 특징

장점	단점
어디서나 접근 가능	도메인 변경 시 유지보수 어려움
외부 리소스 참조 가능	로컬 테스트 불편

✓ 상대경로 (Relative Path)

✓ 정의

- 현재 문서(기준 위치) 기준으로 계산된 경로
- `.`과 `..`을 활용해 상대적인 파일 위치 지정

✓ 예시

표현	의미
<code>./</code>	현재 폴더
<code>../</code>	상위 폴더
<code>images/logo.png</code>	하위 폴더의 파일

```
1 <a href="about.html">소개</a>           <!-- 같은 폴더 내 -->
2 <a href="../index.html">홈</a>        <!-- 상위 폴더 -->
3 <a href="./images/photo.jpg">이미지</a> <!-- 현재 폴더 내 images -->
```

✓ **비교 예제 (구조 가정)**

```
1 | /project/
2 |   ├── index.html
3 |   ├── about.html
4 |   └── /images/
5 |       └── logo.png
```

절대경로 방식

```
1 | 
```

→ 루트(/) 기준으로 /images/logo.png 탐색

상대경로 방식 (index.html에서 사용 시)

```
1 | 
```

→ index.html 기준으로 images 폴더 참조

✓ **외부 URL vs 내부 경로**

구분	예시
외부 절대경로	https://cdn.example.com/lib.js
내부 절대경로	/css/main.css
내부 상대경로	../images/bg.jpg

✓ **정리 비교표**

항목	절대경로	상대경로
시작 기준	사이트 루트 또는 전체 URL	현재 문서 위치 기준
URL 변경 영향	큼 (전체 수정 필요)	작음 (문서 위치만 맞으면 OK)
유지보수	불편 (긴 경로 반복)	편함 (모듈식 파일 구조에 적합)
외부 링크 가능	✓ 가능 (https://...)	✗ 불가
SEO 영향	동일	동일

✓ 실전 팁

상황	추천 경로
외부 사이트 링크	절대 URL (<code>https://...</code>)
사이트 내 이미지, 스타일시트 연결	상대경로 or <code>/</code> 루트기준경로
사이트 전체 구조 고정	루트 기반 절대경로 (<code>/img/...</code>)
프로젝트를 서브폴더에 둘 경우	상대경로 (<code>../</code>) 활용

✓ 한 줄 요약

절대경로는 '고정된 위치'를, 상대경로는 '현 위치 기준 상대적 이동'을 의미하며, 링크 구조와 유지보수 전략에 큰 영향을 미친다.

5.3 북마크(anchor) 링크 (``)

HTML에서 `` 형식의 북마크 링크(Anchor Link)는

같은 페이지 내 특정 위치로 점프(scroll) 하게 하는 기능이다.

긴 문서에서 특정 섹션으로 빠르게 이동하거나

목차, 탭, 사이드바 내비게이션 등에서 매우 유용하게 사용된다.

✓ 기본 개념

```
1 <a href="#section1">1장으로 이동</a>
2
3 <h2 id="section1">1장 제목</h2>
```

- ``: 이동 트리거 (링크)
- `<h2 id="section1">`: 이동 대상 (앵커)

🔍 클릭 시 브라우저가 해당 요소 위치로 스크롤 이동함

✓ 작동 원리

- `#id`는 HTML 요소의 `id` 속성 값을 참조
- `id`가 일치하는 요소가 있으면 해당 위치로 이동

✓ 예제

```
1 <h1>문서 목차</h1>
2 <ul>
3   <li><a href="#intro">소개</a></li>
4   <li><a href="#features">특징</a></li>
5   <li><a href="#download">다운로드</a></li>
6 </ul>
7
8 <h2 id="intro">소개</h2>
9 <p>이 프로젝트는...</p>
10
11 <h2 id="features">특징</h2>
12 <p>다음과 같은 특징이 있습니다...</p>
13
14 <h2 id="download">다운로드</h2>
15 <p>여기서 다운로드 가능합니다.</p>
```

🔥 목차를 누르면 각각의 제목으로 **페이지 내 점프**

✓ 시맨틱 태그와 함께 활용

```
1 <nav>
2   <a href="#top">맨 위로</a>
3 </nav>
4
5 <footer id="bottom">
6   <a href="#top">▲ TOP</a>
7 </footer>
8
9 <header id="top">
10  <h1>페이지 제목</h1>
11 </header>
```

🔥 사용자가 긴 페이지를 스크롤한 뒤 **"TOP" 버튼**으로 되돌아오게 하는 패턴

✓ 외부 문서 + 북마크

```
1 <a href="help.html#faq">FAQ로 바로 가기</a>
```

- `help.html` 문서에서 `id="faq"` 요소로 이동

✅ 스타일 조절 (부드러운 스크롤)

```
1 html {  
2   scroll-behavior: smooth;  
3 }
```

🔴 `scroll-behavior` 속성은 앵커 점프를 부드러운 스크롤 애니메이션으로 전환

✅ 앵커 위치 여백 조정 (고정 헤더 문제 해결)

앵커 요소가 고정 헤더에 가려질 경우:

```
1 [id]::before {  
2   content: "";  
3   display: block;  
4   height: 80px; /* 헤더 높이 */  
5   margin-top: -80px;  
6   visibility: hidden;  
7 }
```

✅ 접근성과 주의점

항목	설명
<code>id</code> 는 유일해야 함	동일한 <code>id</code> 가 여러 개면 이동 오류 발생
링크 텍스트는 명확하게	"여기 클릭" 대신 "다운로드 섹션으로 이동" 사용
자바스크립트와 함께 사용 가능	<code>onclick</code> 이벤트로 스크롤 제어 가능

✅ 잘못된 예시

```
1 <a href="#section1">섹션 1로</a>  
2 <!-- 아래 id가 없음 -->  
3 <!-- <h2>섹션 1</h2> -->
```

🔴 `id="section1"`이 없으면 스크롤 이동 실패

✅ 한 줄 요약

``는 동일 페이지 내 특정 위치로 스크롤 이동시키는 북마크 링크로, 목차, 탭, Top 버튼 등 사용자 편의 기능 구현에 핵심적인 HTML 기능이다.

5.4 네비게이션 메뉴 구성 (<nav>)

HTML에서 `<nav>` 태그는 문서 전체에서 주요 탐색 링크들을 그룹화할 때 사용하는 시맨틱(semantic) 태그이다. 탐색 영역은 웹사이트의 주 메뉴, 사이드 메뉴, 풋터 내 링크 그룹, 목차 링크 집합 등 웹페이지 간 또는 문서 내 중요한 이동 수단을 담당한다.

✓ <nav> 태그란?

- `<nav>` 는 navigation(탐색)의 약자로, 사이트 간 이동을 위한 링크 집합을 명시적으로 표시함
- 스크린리더나 검색엔진이 "이곳은 메뉴구역" 이라고 이해할 수 있게 도와줌
- WAI-ARIA, SEO(검색 최적화), 접근성 측면에서 매우 중요

✓ 기본 사용법

```
1 <nav>
2   <ul>
3     <li><a href="/index.html">홈</a></li>
4     <li><a href="/about.html">소개</a></li>
5     <li><a href="/services.html">서비스</a></li>
6     <li><a href="/contact.html">문의</a></li>
7   </ul>
8 </nav>
```

▼ 출력 결과:

- 홈 | 소개 | 서비스 | 문의
(보통 CSS로 수평 정렬되도록 스타일링함)

✓ 예제: 헤더 내 네비게이션

```
1 <header>
2   <h1>My website</h1>
3   <nav>
4     <a href="/home.html">Home</a>
5     <a href="/portfolio.html">Portfolio</a>
6     <a href="/blog.html">Blog</a>
7     <a href="/contact.html">Contact</a>
8   </nav>
9 </header>
```

🔥 `<nav>` 는 반드시 `` 로 감싸지 않아도 되며, `<a>` 태그 여러 개로 구성 가능

✓ 여러 개의 <nav> 사용

```
1 <aside>
2   <nav>
3     <h3>문서 목차</h3>
4     <ul>
5       <li><a href="#section1">소개</a></li>
6       <li><a href="#section2">설치</a></li>
7       <li><a href="#section3">사용법</a></li>
8     </ul>
9   </nav>
10 </aside>
11
12 <footer>
13   <nav>
14     <a href="/privacy.html">개인정보처리방침</a>
15     <a href="/terms.html">이용약관</a>
16   </nav>
17 </footer>
```

🔴 문서 내 여러 위치에 탐색 블록이 있을 수 있음

✓ 접근성과 시맨틱 구조

항목	설명
✓ 시맨틱	<nav> 은 명확한 탐색 영역 의미 전달
✓ 접근성	스크린 리더가 탐색 영역으로 인식
✓ SEO	검색엔진이 메뉴 링크 구조를 인식 가능

🔴 <nav> 내부가 아니더라도 링크는 만들 수 있지만, 의미와 구조적 명확성은 떨어짐

✓ 스타일링 예시 (CSS)

```
1 nav ul {
2   list-style: none;
3   display: flex;
4   gap: 1rem;
5 }
6
7 nav a {
8   text-decoration: none;
9   color: #333;
10  padding: 0.5em 1em;
11 }
12
13 nav a:hover {
```

```
14 background-color: #eee;
15 }
```

🔴 메뉴를 가로로 배치하고, 호버 시 시각적 효과 추가

✅ 실전 예: 반응형 네비게이션 바

```
1 <nav class="navbar">
2   <a href="/">🏠</a>
3   <a href="/docs">문서</a>
4   <a href="/blog">블로그</a>
5   <a href="/login">로그인</a>
6 </nav>
```

```
1 .navbar {
2   display: flex;
3   justify-content: space-around;
4   background-color: #222;
5   padding: 1em;
6 }
7
8 .navbar a {
9   color: white;
10  text-decoration: none;
11  font-weight: bold;
12 }
```

✅ <nav> vs <div>

비교 항목	<nav>	<div>
의미	탐색 영역	의미 없음
SEO	탐색 영역 인식	불명확
접근성	탐색으로 구분	일반 블록
용도	메뉴, 목차, 쿼링크	레이아웃용

✅ 한 줄 요약

<nav>는 사이트의 주요 이동 수단을 담은 영역을 의미하는 시맨틱 태그로, 접근성과 SEO를 향상시키며 웹 구조를 명확히 한다.