

3. 텍스트 콘텐츠 구성

3.1 제목 태그 (<h1> ~ <h6>)

HTML에서 제목 태그는 문서의 구조적 계층을 표현하고, 정보의 중요도와 의미를 시각적으로 드러내는 역할을 한다. 이들은 단순히 글자를 굵게/크게 만드는 것이 아니라, 시맨틱 구조(Semantic Structure)를 명확히 하여 사용자와 검색엔진, 스크린 리더 모두에게 문서의 흐름을 이해하게 하는 핵심 요소다.

✓ 기본 개념

태그	의미	기본 스타일 (브라우저)	시맨틱 계층
<h1>	가장 중요한 제목	가장 큼, 굵음	최상위 제목
<h2>	하위 제목	<h1> 보다 작음	2단계 제목
<h3> ~ <h6>	점점 덜 중요한 제목	점점 작아짐	3~6단계 제목

✓ 예제

```
1 <h1>HTML 배우기</h1>
2 <h2>1. 소개</h2>
3 <h3>1.1 HTML이란?</h3>
4 <h4>1.1.1 HTML의 역사</h4>
5 <h5>1.1.1.1 HTML4 vs HTML5</h5>
6 <h6>부록: 참고 자료</h6>
```

☞ 위와 같이 사용할 경우, 문서 전체가 논리적 구조 트리로 정리됨.

✓ 시맨틱 구조와 SEO(검색엔진 최적화)

기능	효과
<h1> 사용	페이지 제목으로 검색엔진 인식
<h2>, <h3> 분류	하위 주제의 구분 및 문맥 부여
올바른 제목 구조 유지	스크린 리더가 문서를 음성으로 읽을 때 사용자가 방향을 잡을 수 있음
제목 내 키워드 포함	검색 랭킹 최적화에 도움

✓ 사용 시 유의 사항

원칙	설명
<code><h1></code> 은 한 페이지에 1개 사용 권장	페이지 주제를 명확히 하기 위함
제목의 순서 건너뛰지 않기	<code><h1></code> → <code><h3></code> 이런 식의 건너뛴은 바람직하지 않음
시각적 크기와 무관하게 시맨틱 우선	크기가 작다고 <code><h6></code> 를 무조건 쓰지 말고 의미에 따라 사용

✓ 스타일을 변경하고 싶을 때

제목의 의미를 유지한 채 시각적으로 다르게 표현하려면 CSS 사용

```
1 <h2 class="sub-title">부제목</h2>
2
3 <style>
4   .sub-title {
5     font-size: 1.2rem;
6     color: darkslateblue;
7     font-weight: normal;
8   }
9 </style>
```

✦ 시각을 위한 `div + font-size` 남용보다 의미가 있는 `h` 태그 사용 + CSS 변경이 바람직함.

✓ 실제 사용 예시

```
1 <body>
2   <h1>오정석의 개발 노트</h1>
3
4   <h2>자바스크립트 입문</h2>
5   <p>변수, 조건문, 반복문 등의 기초를 배웁니다.</p>
6
7   <h3>const와 let의 차이</h3>
8   <p>재할당 가능 여부로 구분됩니다.</p>
9
10  <h2>HTML 시맨틱 구조</h2>
11  <p>웹 문서를 구조화하고 접근성을 개선합니다.</p>
12 </body>
```

✓ 접근성과 도구 연동

- 스크린리더: 제목 계층을 따라 읽기 → 문서 길더라도 빠르게 위치 탐색 가능
- Braille 장치: 제목 태그를 따로 구분하여 표시
- 검색 엔진: `<h1>` ~ `<h3>` 에 포함된 키워드를 문서 요약에 반영

✓ 한 줄 요약

`<h1>` ~ `<h6>` 태그는 문서의 계층적 구조를 나타내며, 단순한 글자 크기가 아니라 의미(시맨틱)를 전달하는 핵심 태그이다.

3.2 단락 태그 (`<p>`)

태그는 HTML에서 "문단(Paragraph)"을 표현하는 대표적인 블록 요소이다. 웹 페이지에서 본문 텍스트의 기본 단위를 구획짓는 용도로 사용되며, 자동 줄바꿈, 위아래 간격(margin), 가독성 확보 등 다양한 기능을 내포한다.

✓ 기본 문법

```
1 | <p>이것은 하나의 문단입니다.</p>
```

- 시작 태그 `<p>` 와 종료 태그 `</p>` 로 감싼 부분이 하나의 문단으로 인식됨
- 하나의 문단 안에는 텍스트, 인라인 태그(`span`, `a`, `strong` 등) 를 포함할 수 있음
- 중첩 불가: `<p><p>중첩</p></p>` → 잘못된 문법

✓ 기본 동작 특징

속성	설명
블록 요소	줄바꿈이 자동으로 발생함 (<code>display: block</code>)
위아래 마진 기본 적용	브라우저 기본 스타일: <code>margin-top</code> , <code>margin-bottom</code> 존재
텍스트 자동 줄바꿈	글이 너비를 넘으면 자동으로 줄바꿈 됨 (word-wrap)

✓ 예제

```
1 | <p>HTML은 웹페이지의 구조를 만드는 언어입니다.</p>
2 | <p>CSS는 스타일을, JavaScript는 동작을 담당합니다.</p>
```

- 위 예제는 두 개의 단락으로 출력됨
- 두 문장 사이에는 줄바꿈과 기본 간격이 생김

✓ 잘못된 사용 예시

```
1 | <p>첫 번째 문단
2 | <p>두 번째 문단</p>
```

- 첫 번째 `<p>` 가 닫히지 않았고 중첩 구조도 어긋남 → HTML 파서가 자동 수정할 수 있으나, 표준 위반

✓ 인라인 요소와의 사용

내부에는 인라인 요소를 자유롭게 삽입할 수 있다:

```
1 <p>
2   <strong>중요:</strong> HTML 문서는 항상 구조를 먼저 생각해야 합니다.
3   <a href="#">이 문장을 클릭하세요.</a>
4 </p>
```

✦ 하지만 블록 요소(`div`, `section`, `p`)는 내부에 직접 넣으면 안됨

→ `<p><div>내용</div></p>` → ✗ 문법 오류

✓ CSS 스타일 적용 예시

```
1 <p class="warning">이 문장은 경고 메시지입니다.</p>
2
3 <style>
4   .warning {
5     color: red;
6     background-color: #fff3f3;
7     padding: 10px;
8     border-left: 4px solid red;
9   }
10 </style>
```

✦ 단락 태그는 클래스/아이디를 이용해 시각적 표현도 다양하게 제어 가능

✓ <p> vs

구분	<p>	
역할	문단 전체를 구획	강제 줄바꿈
구조적 의미	있음 (시맨틱 블록)	없음
블록 요소 여부	✓ block	✗ inline
사용 예시	단락 구분	주소, 시, 가사 등 줄 나눔

예:

```
1 <p>나는 오늘<br>하늘을 보았다.</p>
```

✓ 접근성과 SEO 관점

- 스크린리더는 `<p>` 를 만날 때 잠시 멈추며 문장 단위로 읽음
- 검색엔진도 `<p>` 내부 텍스트를 본문으로 간주하여 인덱싱

✓ 실제 예시

```
1 <body>
2   <h1>HTML 소개</h1>
3   <p>HTML은 HyperText Markup Language의 약자입니다.</p>
4   <p>웹 문서를 작성할 때 가장 기본이 되는 마크업 언어입니다.</p>
5 </body>
```

✓ 한 줄 요약

`<p>` 태그는 웹 문서의 "문장 단위 논리 구조"를 구분하는 핵심 요소로, 단락을 명확히 구분하고 읽기 좋은 구조를 만드는 데 필수적이다.

3.3 줄 바꿈, 수평선 (`
`, `<hr>`)

HTML에서 `
` 과 `<hr>` 는 본문 텍스트 내에서 시각적 구분을 위한 중요한 요소다.

이 두 태그는 모두 빈 태그(Self-closing Tag)이며, 별도의 닫는 태그 없이 단독으로 사용된다.

✓ 1. `
` — 줄 바꿈 (line break)

✓ 의미

- 강제 줄 바꿈(line break) 을 생성하는 인라인 요소
- 문단을 나누는 것이 아니라, 한 문단 안에서 **줄만 바꾸고 싶을 때** 사용

✓ 문법

```
1 | 문장1<br>문장2
```

→ 출력 시:

```
1 | 문장1
2 | 문장2
```

✓ 사용 예시

```
1 <p>
2   오정석<br>
3   서울특별시 강남구<br>
4   전화: 010-1234-5678
5 </p>
```

📌 주소, 연락처, 시, 노래 가사 등 포맷 유지가 중요한 텍스트에서 유용

✓ <p> 와의 차이

구분	<p>	
의미	문단 구분	줄 구분
줄바꿈	위아래 여백 포함	줄만 내려감
시맨틱	있음	없음 (순수 표현)
블록/인라인	블록	인라인

✓ 2. <hr> — 수평선 (horizontal rule)

✓ 의미

- 문단이나 내용의 구분선을 삽입하는 태그
- 시각적으로 구획을 나누며, 의미상 전환점을 나타냄
- 브라우저 기본 스타일: 가로줄 회색 실선

✓ 문법

```
1 <hr>
```

또는 XHTML 스타일:

```
1 <hr />
```

✓ 사용 예시

```
1 <p>소개글입니다.</p>
2 <hr>
3 <p>이제 본문 내용이 시작됩니다.</p>
```

📌 페이지 구역 전환, 글의 주제 구분, 각주 구분 등에 사용

✓ 기본 스타일 변경 (CSS)

```
1 <hr class="custom-line">
2
3 <style>
4   .custom-line {
5     border: none;
6     height: 2px;
7     background-color: #444;
8     width: 50%;
9     margin: 20px auto;
10  }
11 </style>
```

→ 중앙 정렬된 두꺼운 회색 줄 생성

✓ 접근성과 시맨틱 의미

- `<hr>` 는 "이전 내용과 다음 내용이 의미상 구분됨" 을 나타냄
- `
` 는 단순히 화면상 줄을 바꾸는 표현 도구
- 둘 다 시맨틱 요소로 활용 가능하지만, 잘못 남용하면 구조 흐름이 깨질 수 있음

✓ `
`, `<hr>` 함께 쓰인 예

```
1 <h2>오시는 길</h2>
2 <p>
3   서울특별시 강남구 테헤란로 123<br>
4   지하철 2호선 강남역 5번 출구<br>
5   도보 5분 거리
6 </p>
7
8 <hr>
9
10 <h2>연락처</h2>
11 <p>
12   이메일: contact@example.com<br>
13   전화: 010-1234-5678
14 </p>
```

✓ 실수 방지 포인트

실수 내용	설명
<code>
</code> 남용	문단 구분에도 <code>
</code> 만 반복 → 구조 파악 어려움
<code><hr></code> 디자인용 남용	단순한 선을 CSS로 그릴 수 있음에도 의미 없는 <code><hr></code> 사용

실수 내용	설명
빈 태그에 <code></br></code> 등 잘못된 종료 사용	HTML5에선 <code>
</code> 만으로 충분

✓ 요약 비교

항목	<code>
</code>	<code><hr></code>
기능	줄 바꿈	구획 나눔
요소 타입	인라인 요소	블록 요소
의미(시맨틱)	낮음 (단순 표현)	있음 (내용 분리 의미)
닫는 태그	없음 (<code>
</code>)	없음 (<code><hr></code>)

✓ 한 줄 요약

`
` 는 줄을 바꾸고, `<hr>` 는 의미를 나눈다.

두 태그 모두 콘텐츠 흐름을 시각적으로 보완하는 데 사용되며, 구조적 의미를 고려해 적절히 활용해야 한다.

3.4 강조 태그 (``, ``, ``, `<i>`, `<mark>`)

HTML에서 강조(emphasis) 는 단순히 글자를 굵게 또는 기울이는 시각적 효과뿐만 아니라,

의미(semantic) 와 강조의 목적성에 따라 다양한 태그를 사용해야 한다.

이 항목에서는 강조 태그들의 의미적 차이, 시각적 차이, 접근성, 사용 예제 등을 상세히 설명한다.

✓ 전체 태그 요약표

태그	의미 중심 강조	시각적 강조	기본 스타일	의미
<code></code>	✓	✓	굵은 글씨 (bold)	강력한 의미 강조
<code></code>	✓	✓	기울임 (italic)	강조, 감정적 뉘앙스 전달
<code></code>	✗	✓	굵은 글씨 (bold)	단순한 시각 강조
<code><i></code>	✗	✓	기울임 (italic)	단순한 스타일 변경, 외래어 등
<code><mark></code>	✓	✓	형광펜 배경	강조 표시, 검색 결과 강조

◆ 1. — 의미 강조 + 굵게 표시

✓ 사용 목적

- 중요한 내용임을 명시적으로 강조
- 스크린리더가 보통 "강조된 말입니다" 등으로 음성 강조

✓ 예시

1 | <p>주의: 이 작업은 되돌릴 수 없습니다.</p>

✦ 단순히 굵게만 보이지만, 의미상 강조가 있다는 것이 핵심

◆ 2. — 의미 강조 + 기울임 표시

✓ 사용 목적

- 말의 강조, 감정 전달, 억양 등을 나타냄
- 스크린리더는 억양을 다르게 읽음

✓ 예시

1 | <p>정말 꼭 확인하고 제출하세요.</p>

✦ 텍스트 내용 중 강세나 뉘앙스를 표현할 때 적절

◆ 3. — 단순 굵게 (의미 없음)

✓ 사용 목적

- 단순한 시각적 강조용
- 의미적 중요성은 없음

✓ 예시

1 | <p>이 항목은 신규 상품입니다.</p>

✦ 강조 효과는 주고 싶지만 의미 강조는 원치 않을 때 사용

◆ 4. `<i>` — 단순 기울임 (의미 없음)

✓ 사용 목적

- 외래어, 학명, 생각, 용어, 속삭임 등 **기울임으로 표현**
- 의미 강조는 아님

✓ 예시

```
1 | <p>이탈리아어로 '사랑'은 <i>amore</i>입니다.</p>
```

✦ 기술용어, 영화 제목, 단어 정의 등 **비일상적 용어** 강조에 활용

◆ 5. `<mark>` — 강조 표시 (형광펜 효과)

✓ 사용 목적

- 검색 결과 강조, 중요 키워드 강조 등에 사용
- 기본 스타일은 노란 배경 + 검정 글씨

✓ 예시

```
1 | <p>검색어 <mark>HTML</mark>이 포함된 결과입니다.</p>
```

✦ 시각적으로 눈에 띄게 만들며, **콘텐츠 내 특정 구간을 하이라이트**할 때 유용

✓ 시맨틱 vs 비시맨틱 비교

구분	시맨틱 강조 (<code></code> , <code></code>)	비시맨틱 강조 (<code></code> , <code><i></code>)
의미 전달	✓ 강조 의미 있음	✗ 단순한 스타일 효과
접근성	✓ 스크린리더가 강조로 인식	✗ 단순 기울임/굵기만 반영
스타일 변경	CSS로 가능	CSS로 가능

✦ 가능하면 의미 있는 강조는 ``, `` 을 쓰고,
단순 스타일만 필요할 때 ``, `<i>` 를 쓰자.

✓ CSS로 강조 스타일 커스터마이징

```
1 <mark class="important">중요 단어</mark>
2
3 <style>
4   .important {
5     background-color: orange;
6     font-weight: bold;
7     color: white;
8   }
9 </style>
```

🔥 `<mark>` 를 꾸며서 브랜드 컬러 강조 등에도 활용 가능

✓ 실전 예제

```
1 <p>
2   <strong>경고:</strong> <em>절대</em> 전원을 끄지 마세요!<br>
3   이 제품은 <i>prototype</i> 단계이며, <b>기능이 제한적</b>입니다.<br>
4   <mark>베타테스트</mark>에 참여해주셔서 감사합니다.
5 </p>
```

✓ 한 줄 요약

``, `` 은 의미 강조 / ``, `<i>` 는 시각 강조 / `<mark>` 는 시각적 하이라이트용 강조 도구이다.
목적에 따라 구분해서 써야 웹의 접근성과 시맨틱 구조가 유지된다.

3.5 의미 없는 태그 (``, `<div>`)

HTML에서 `` 과 `<div>` 는 "의미 없는(Non-semantic)" 태그로 분류된다.

이들은 본질적으로 구조나 시맨틱한 의미를 부여하지 않으며,
오직 레이아웃 구획 및 스타일 적용용 컨테이너 역할만 수행한다.

하지만 HTML 문서에서 구성의 유연성을 높이는 핵심 요소이기도 하다.

✓ 개요 비교

태그	블록/인라인	의미 (시맨틱)	용도
<code><div></code>	블록 요소	✗ 없음	레이아웃 구획, 그룹화
<code></code>	인라인 요소	✗ 없음	텍스트 일부 스타일 지정, 인라인 그룹

◆ 1. <div> — 레이아웃 그룹화용 블록 요소

✓ 특징

- **블록 요소**: 줄바꿈을 동반하며 세로 레이아웃 구성에 적합
- 콘텐츠 구역을 **논리적으로 묶기 위한 컨테이너**
- 스타일 지정, JavaScript 제어, 컴포넌트 구조화에 활용

✓ 예시

```
1 <div class="card">
2   <h2>제품명</h2>
3   <p>이 제품은 매우 훌륭합니다.</p>
4 </div>
```

✦ 의미는 없지만, CSS와 JS 적용을 위한 **기준 영역** 역할을 함

✓ 사용 사례

- 페이지 레이아웃 구획 (header, main, footer)
 - 그리드 시스템 구현 (`display: grid` / `flex`)
 - 컴포넌트 단위 구분
 - 반복 항목 묶기
-

✗ 남용 주의

```
1 <div class="header">
2   <div class="nav">
3     <div class="logo">Logo</div>
4   </div>
5 </div>
```

→ 시맨틱 태그로 대체 가능:

```
1 <header>
2   <nav>
3     <h1>Logo</h1>
4   </nav>
5 </header>
```

✦ 의미가 있는 태그(`header`, `nav`, `article`)가 있다면 `div` 대신 사용하는 것이 좋음

◆ 2. `` — 인라인 스타일 지정용 컨테이너

✓ 특징

- 인라인 요소: 줄바꿈 없이 텍스트 흐름 내에서 사용
- 문장의 일부를 감싸 스타일 지정하거나 JS 제어

✓ 예시

```
1 | <p>가격: <span class="price">₩49,000</span></p>
```

→ ``으로 감싼 부분에만 글자색이나 강조 스타일 부여 가능

✓ CSS와 함께 활용

```
1 | <span class="highlight">강조할 부분</span>
2 |
3 | <style>
4 |   .highlight {
5 |     background-color: yellow;
6 |     font-weight: bold;
7 |   }
8 | </style>
```

✓ 함께 사용한 예제

```
1 | <div class="product">
2 |   <h2>에너지 드링크</h2>
3 |   <p>카페인 함량: <span class="caffeine">80mg</span></p>
4 | </div>
```

- `<div class="product">`: 하나의 상품 단위 구획
- ``: 특정 단어 강조

✓ 시맨틱 태그와의 차이점

구분	의미 있는 태그 (<code><header></code> , <code><article></code> 등)	의미 없는 태그 (<code><div></code> , <code></code>)
의미 전달	문서 구조에 대한 힌트 제공 (SEO, 접근성)	없음. 오직 스타일/JS 목적
스크린리더	읽을 때 구역이나 목적 알려줌	아무 의미 없이 읽음
CSS/JS 용도	✓ 사용 가능	✓ 사용 가능

◆ 구조적 의미가 필요한 곳엔 **시맨틱 태그**,
의미보다 스타일이나 기능만 필요한 곳엔 `div / span`을 쓰자.

✓ 실수 방지 포인트

실수	권장 방식
모든 걸 <code><div></code> 로 감쌘	의미가 있는 구조에는 시맨틱 태그 사용
<code></code> 안에 <code><div></code> 넣음	✗ 블록 요소는 인라인 안에 못 들어감
시맨틱 태그를 <code><div></code> 로 대체	SEO와 접근성 손상

✓ 한 줄 요약

`<div>` 와 `` 은 의미는 없지만, 구조와 스타일 제어를 위해 HTML에서 가장 자주 사용되는 유틸리티 태그다.
다만 의미가 필요한 곳에서는 시맨틱 태그를 우선 사용해야 한다.

3.6 특수 문자 (, < , > , & 등)

HTML은 태그를 사용하는 마크업 언어이기 때문에,
일부 문자 자체를 그대로 쓰면 브라우저가 오해하거나 파싱 오류를 일으킬 수 있다.
이런 경우 문자 그대로의 의미를 표현하기 위해 HTML 엔티티(entity)를 사용한다.

✓ HTML 엔티티란?

HTML에서 태그로 오해될 수 있는 문자를 기호가 아닌 코드로 표현하는 방식
예: `<` → `<`, `>` → `>`

✓ 기본 사용법

```
1 | &엔티티이름;
```

또는

```
1 | &#코드번호;
```

예:

- `<` == `<`
- `<` == `<`

→ 둘 다 같은 결과지만 엔티티 이름 방식이 더 직관적

✓ 주요 특수 문자 목록

표시할 문자	엔티티 이름	출력 결과	의미 설명
공백	<code>&nbsp;</code>		Non-breaking space (띄어쓰기 유지)
<	<code>&lt;</code>	<	Less than 기호 (<code><div></code> 방지용)
>	<code>&gt;</code>	>	Greater than 기호
&	<code>&amp;</code>	&	Ampersand 기호 (@, 등에 사용됨)
"	<code>&quot;</code>	"	큰따옴표 (속성 내 문자열에 사용)
'	<code>&apos;</code>	'	작은따옴표 (HTML5부터 일부 브라우저만 지원)
©	<code>&copy;</code>	©	저작권 기호
®	<code>&reg;</code>	®	등록상표 기호
™	<code>&tm;</code>	™	상표 기호
€	<code>&euro;</code>	€	유로화
¥	<code>&yen;</code>	¥	엔화
→	<code>&rarr;</code>	→	오른쪽 화살표
←	<code>&larr;</code>	←	왼쪽 화살표

✓ 예제: 코드 출력 시

```
1 | <p>&lt;div>는 블록 요소입니다.&lt;/div></p>
```

출력 결과:

```
1 | <div>는 블록 요소입니다.</div>
```

🔴 실제 태그로 인식되지 않고 텍스트로 출력

✓ 예제: 공백 유지

```
1 | <p>안녕하세요,&nbsp;&nbsp;&nbsp;&nbsp;오정석입니다.</p>
```

출력 결과:

```
1 | 안녕하세요,     오정석입니다.
```

- HTML은 연속된 공백을 하나의 공백으로 축소하므로 여백 유지가 필요할 때 ` ` 사용

✓ 예제: 속성 값에서의 안전 처리

```
1 <a href="search.html?q=Tom&Jerry">Tom & Jerry 검색</a>
```

- `&` 기호는 속성 값에서 혼동되므로 `&` 로 대체해야 함
- 잘못 쓸 경우 `q=Tom` 으로만 인식되고 `Jerry` 는 무시됨

✓ 문자 엔티티 vs JavaScript 이스케이프

상황	사용 방식	예시
HTML 안에서	HTML 엔티티	<code><</code> , <code>></code> , <code>&</code>
JS 문자열 안에서	백슬래시 이스케이프	<code>\</code> , <code>'</code> , <code>\"</code>

✦ 혼동하지 말자 — HTML은 엔티티, JS는 백슬래시

✓ 실수 방지 포인트

실수 상황	해결 방법
<code><</code> , <code>></code> 를 그대로 출력하고 싶을 때	<code><</code> , <code>></code> 사용
<code>&</code> 를 URL이나 텍스트로 쓸 때	<code>&</code> 사용
연속 공백이 사라짐	<code>&nbsp;</code> 사용
속성 내 인용문 겹침 문제	<code>"</code> 또는 <code>'</code> 으로 구분

✓ 한 줄 요약

HTML에서 태그와 구분되는 특수 기호를 정확히 출력하고 싶다면, 문자 엔티티 (`<`, `&`, ` ` 등)를 반드시 사용해야 한다.