

7. 테이블 구성

7.1 기본 구조 (<table>, <tr>, <td>, <th>)

HTML 테이블은 표 형태의 데이터를 표현할 때 사용된다.
예: 시간표, 가격표, 통계표, 순위표, 데이터 목록 등.
표를 구성하는 기본 요소는 <table>, <tr>, <td>, <th> 네 가지이다.

✓ 전체 구조 요약

```
1 <table>
2   <tr>
3     <th>제목1</th>
4     <th>제목2</th>
5   </tr>
6   <tr>
7     <td>데이터1</td>
8     <td>데이터2</td>
9   </tr>
10 </table>
```

✓ 태그별 설명

태그	의미
<table>	표 전체를 감싸는 루트 요소
<tr>	table row : 한 줄(행)을 의미
<td>	table data : 행의 일반 데이터 셀
<th>	table header : 제목 셀 (기본적으로 굵은 글씨 + 가운데 정렬)

✓ 기본 예제: 2행 2열 표

```
1 <table border="1">
2   <tr>
3     <th>이름</th>
4     <th>나이</th>
5   </tr>
6   <tr>
7     <td>홍길동</td>
8     <td>25</td>
9   </tr>
10  <tr>
11    <td>김철수</td>
```

```

12     <td>30</td>
13 </tr>
14 </table>

```

🔴 `border="1"` 은 임시로 테두리를 보이게 하기 위한 HTML 속성. CSS로 꾸미는 것이 현대적인 방식이다.

✅ 시맨틱 강조: `th` vs `td`

```

1 <tr>
2   <th>항목</th>    <!-- 헤더 셀 -->
3   <td>값</td>      <!-- 데이터 셀 -->
4 </tr>

```

- `<th>` 는 열 또는 행의 헤더를 의미하여 시맨틱하게 구분
- 브라우저/스크린리더는 `<th>` 를 특별히 인식해 읽기 도움 제공

✅ 구조 시각화

```

1 <table>
2   └─ <tr>   ← 첫 번째 행
3     └─ <th> ← 헤더 셀
4         └─ <th>
5   └─ <tr>   ← 두 번째 행
6     └─ <td> ← 일반 셀
7         └─ <td>
8         ...
9 </table>

```

✅ CSS와 함께 쓰는 예

```

1 <table>
2   <tr>
3     <th>과목</th>
4     <th>점수</th>
5   </tr>
6   <tr>
7     <td>수학</td>
8     <td>90</td>
9   </tr>
10 </table>
11
12 <style>
13 table {
14   border-collapse: collapse;
15   width: 300px;
16 }
17 th, td {

```

```

18 border: 1px solid #ccc;
19 padding: 8px;
20 text-align: center;
21 }
22 th {
23     background-color: #f5f5f5;
24 }
25 </style>

```

✓ 한 줄 요약

HTML 테이블은 `<table>` 을 루트로 하며, `<tr>` 로 행을 만들고, `<th>` 는 제목 셀, `<td>` 는 일반 셀을 구성하는 기본 구조를 가진다.

7.2 헤더, 바디, 풋 (<thead>, <tbody>, <tfoot>)

HTML의 `<table>` 요소는 시각적으로만 표를 만들 수 있는 것이 아니라, 논리적 구조를 세분화해서 의미를 명확히 구분하고, 스타일 적용이나 스크립트 처리도 효율적으로 할 수 있다.

바로 이 때 사용하는 구조화 태그가:

- `<thead>` (테이블의 헤더 행)
- `<tbody>` (테이블의 본문 데이터)
- `<tfoot>` (테이블의 요약 또는 합계 행)

이다.

✓ 기본 구조

```

1 <table>
2   <thead>
3     <tr>
4       <th>상품명</th>
5       <th>가격</th>
6     </tr>
7   </thead>
8   <tbody>
9     <tr>
10      <td>책</td>
11      <td>10,000원</td>
12    </tr>
13    <tr>
14      <td>연필</td>
15      <td>1,000원</td>
16    </tr>
17  </tbody>
18  <tfoot>
19    <tr>
20      <td>합계</td>
21      <td>11,000원</td>

```

```
22     </tr>
23   </tfoot>
24 </table>
```

✅ 태그별 설명

태그	의미 및 용도
<code><thead></code>	표의 제목 행. <code><tr></code> + <code><th></code> 조합으로 사용하며 일반적으로 열 제목을 포함
<code><tbody></code>	실제 데이터(내용) 행들. 반복되는 <code><tr></code> + <code><td></code> 구조
<code><tfoot></code>	하단 요약 행 (예: 합계, 평균, 요약 텍스트 등)

✅ CSS 및 JS에서 유용한 이유

- CSS에서 영역별 스타일링 가능

```
1 thead {
2    background-color: #f0f0f0;
3    font-weight: bold;
4  }
5  tfoot {
6    background-color: #e0e0e0;
7    text-align: right;
8  }
```

- JavaScript로 동적으로 행 추가/제거할 때 구조를 쉽게 분리해서 처리 가능

```
1 const tbody = document.querySelector('tbody');
2 tbody.innerHTML += '<tr><td>지우개</td><td>500원</td></tr>';
```

✅ 브라우저 렌더링 순서

- HTML 코드상 `<tfoot>` 이 `<thead>` 보다 뒤에 오지만,
브라우저는 항상 `<thead>` → `<tfoot>` → `<tbody>` 순서로 렌더링한다.

🔴 따라서 `<tfoot>` 은 HTML 상에서 어디에 위치하든, 브라우저는 **상단(head)** - **하단(foot)** - **본문(body)** 순서로 인식한다.

✅ 접근성(Accessibility) 측면

- 시맨틱 구조로 나뉘어져 있어, **스크린 리더**가 더 잘 읽을 수 있음
- 복잡한 표에서 데이터와 제목 행을 명확히 구분 가능
- `scope` 속성과 함께 사용 시 가독성 향상

```
1 <th scope="col">상품명</th>
2 <th scope="col">가격</th>
```

✓ 요약

태그	위치	사용 목적
<code><thead></code>	표의 상단	열 제목을 지정
<code><tbody></code>	표의 중간	실제 데이터 목록
<code><tfoot></code>	표의 하단	합계, 요약 등 결과

✓ 한 줄 요약

`<thead>`, `<tbody>`, `<tfoot>`은 HTML 테이블을 논리적으로 구분하여 구조화, 스타일 적용, 접근성 향상에 모두 도움이 되는 표 구성 방식이다.

7.3 셀 병합 (colspan, rowspan)

HTML 테이블에서는 표를 만들 때 열을 가로로 병합하거나, 행을 세로로 병합할 수 있다. 이 기능은 `<td>` 또는 `<th>` 태그에 `colspan`, `rowspan` 속성을 부여해서 사용한다. 시간표, 성적표, 통계표처럼 복잡한 테이블을 만들 때 반드시 필요한 구조 기능이다.

✓ colspan (열 병합: 가로 병합)

- 한 셀이 여러 열(column)에 걸쳐도록 만든다

```
1 <tr>
2   <th colspan="2">이름 및 연락처</th>
3 </tr>
4 <tr>
5   <td>홍길동</td>
6   <td>010-1234-5678</td>
7 </tr>
```

✦ "이름 및 연락처" 셀이 가로로 두 열을 차지함

✓ rowspan (행 병합: 세로 병합)

- 한 셀이 여러 행(row)에 걸쳐도록 만든다

```

1 <tr>
2   <th rowspan="2">학번</th>
3   <td>20240101</td>
4 </tr>
5 <tr>
6   <td>20240102</td>
7 </tr>

```

🔴 "학번" 셀이 세로로 두 행을 차지함

✅ colspan + rowspan 함께 쓰는 예

```

1 <table border="1">
2   <tr>
3     <th rowspan="2">지역</th>
4     <th colspan="2">인구수</th>
5   </tr>
6   <tr>
7     <th>남성</th>
8     <th>여성</th>
9   </tr>
10  <tr>
11    <td>서울</td>
12    <td>500만</td>
13    <td>530만</td>
14  </tr>
15 </table>

```

🔴 구조:

- "지역" 은 왼쪽 열에서 2행을 병합 (rowspan="2")
- "인구수" 는 상단에서 2열을 병합 (colspan="2")

✅ 잘못된 병합 예시

```

1 <tr>
2   <td colspan="2">합계</td>
3 </tr>
4 <tr>
5   <td>10</td>
6   <td>20</td>
7   <td>30</td> <!-- 오류: 위에 colspan으로 열이 2개로 지정되었는데, 3열이 있음 -->
8 </tr>

```

🔴 colspan 또는 rowspan 사용 시, 전체 셀의 개수 일치 를 항상 고려해야 함

→ 안 맞으면 레이아웃이 깨짐

✓ 표의 전체 열 수 계산 방법

열 병합과 행 병합이 있는 테이블에서는, 표 전체에서 가장 넓은 행을 기준으로 열 수 계산해야 한다.

✓ CSS와 병합 셀 꾸미기

```
1 td[colspan], td[rowspan] {  
2   background-color: #f9f9f9;  
3   font-weight: bold;  
4 }
```

✓ 접근성 고려

- 병합 셀에는 `scope`, `headers`, `id` 속성을 조합하여 스크린 리더에 정확한 정보 제공 가능
- 특히 복잡한 통계 테이블에서는 병합 위치를 명확하게 표현해 주는 것이 중요함

✓ 한 줄 요약

`colspan`은 가로로 셀을 병합하고, `rowspan`은 세로로 셀을 병합하는 속성으로, 복잡한 테이블 구조를 설계할 때 반드시 필요한 도구이다.

7.4 테이블 속성 (`border`, `width`, `align` 등)

HTML의 `<table>`, `<tr>`, `<td>`, `<th>` 요소는 여러 속성을 통해 기본적인 시각적 스타일을 지정할 수 있다.

HTML4까지는 직접 속성을 통해 스타일을 지정했으나, HTML5 이후에는 **CSS를 사용하는 것이 표준**이다.

하지만 여전히 HTML 속성의 동작 원리와 의미를 이해하는 것은 테이블 처리에 매우 중요하다.

✓ 주요 속성 정리 (구형 방식 / CSS 이전)

속성 이름	적용 대상	설명
<code>border</code>	<code><table></code>	테이블 및 셀의 테두리 두께 (픽셀 단위)
<code>cellspacing</code>	<code><table></code>	셀과 셀 사이의 간격
<code>cellpadding</code>	<code><table></code>	셀 내부 내용과 테두리 사이 여백
<code>width</code>	<code><table></code> , <code><td></code>	너비 지정 (px 또는 %)
<code>height</code>	<code><table></code> , <code><td></code>	높이 지정
<code>align</code>	<code><table></code> , <code><tr></code> , <code><td></code>	정렬 방식 (left, center, right)
<code>valign</code>	<code><td></code> , <code><th></code>	수직 정렬 방식 (top, middle, bottom)
<code>bgcolor</code>	<code><table></code> , <code><tr></code> , <code><td></code>	배경색 지정

속성 이름	적용 대상	설명
<code>colspan</code>	<code><td></code> , <code><th></code>	열 병합 (앞서 설명)
<code>rowspan</code>	<code><td></code> , <code><th></code>	행 병합 (앞서 설명)

✅ 예시: HTML 속성을 직접 사용하는 방식 (구버전 방식)

```

1 <table border="1" cellspacing="5" cellpadding="10" width="500" align="center"
  bgcolor="#f5f5f5">
2   <tr>
3     <th>이름</th>
4     <th>나이</th>
5   </tr>
6   <tr>
7     <td align="center" valign="top">홍길동</td>
8     <td align="right" valign="bottom">25</td>
9   </tr>
10 </table>

```

🚩 위 방식은 여전히 동작하지만 **비표준**이며, HTML5에서는 **CSS로 대체**하는 것이 원칙이다.

✅ 현대적 방식: CSS로 대체하는 예

```

1 <table class="styled-table">
2   <tr>
3     <th>과목</th>
4     <th>점수</th>
5   </tr>
6   <tr>
7     <td>수학</td>
8     <td>95</td>
9   </tr>
10 </table>
11
12 <style>
13 .styled-table {
14   width: 500px;
15   border-collapse: collapse;
16   margin: auto;
17   background-color: #f5f5f5;
18 }
19 .styled-table th,
20 .styled-table td {
21   border: 1px solid #ccc;
22   padding: 10px;
23   text-align: center;
24   vertical-align: middle;
25 }

```


✓ border-collapse 속성의 의미

값	설명
separate	각 셀마다 테두리가 따로 있음 (기본값)
collapse	셀 사이의 테두리를 하나로 합침 (더 깔끔한 테이블)

```
1 table {
2   border-collapse: collapse;
3 }
```

✓ 정렬 관련 CSS 속성

CSS 속성	설명
text-align	수평 정렬 (left, center, right)
vertical-align	수직 정렬 (top, middle, bottom)

```
1 td {
2   text-align: right;
3   vertical-align: bottom;
4 }
```

✓ width와 %의 차이

```
1 <table width="100%">  <!-- 부모 영역 전체 차지 -->
2 <table width="500">   <!-- 고정 픽셀 크기 -->
```

HTML 속성보다 CSS로 명확하게 지정하는 것이 더 유연하다:

```
1 table {
2   width: 100%;
3   max-width: 800px;
4 }
```

✓ 한 줄 요약

HTML 테이블의 `border`, `width`, `align`, `valign` 등은 시각적 스타일을 지정하지만, 현대적인 웹에서는 CSS로 처리하는 것이 필수이며, `border-collapse`, `text-align`, `padding`, `width` 등을 적절히 조합해 구조적이고 아름다운 표를 만들 수 있다.

7.5 접근성 고려한 테이블 마크업

접근성(Accessibility)은 장애를 가진 사용자도 웹 콘텐츠에 동등하게 접근하고 이해할 수 있도록 만드는 것을 말한다.

특히 시각 장애인을 위한 스크린 리더(Screen Reader)는 테이블의 구조를 이해하기 위해 시맨틱한 HTML 마크업과 명확한 관계 지정을 요구한다.

복잡한 표일수록 접근성을 고려한 마크업이 필수이다.

✓ 접근성 향상을 위한 핵심 요소

요소	설명
<code><caption></code>	표의 제목을 지정해줌 (스크린 리더에서 유용)
<code>scope</code>	헤더 셀(<code>th</code>)이 어떤 셀을 설명하는지 지정
<code>headers</code>	각 데이터 셀(<code>td</code>)이 어떤 헤더와 연관되는지 ID로 지정
<code>id</code>	헤더 셀의 식별자 정의 (<code>headers</code> 속성과 연결)
<code>summary</code>	예전 HTML4의 설명 속성 (HTML5에서는 폐지됨)
<code><thead></code> , <code><tbody></code> , <code><tfoot></code>	구조를 명확히 나눠줌

✓ 예제 1: 기본적인 접근성 마크업

```
1 <table>
2   <caption>학생 성적표</caption>
3   <thead>
4     <tr>
5       <th scope="col">이름</th>
6       <th scope="col">수학</th>
7       <th scope="col">영어</th>
8     </tr>
9   </thead>
10  <tbody>
11    <tr>
12      <td>홍길동</td>
13      <td>90</td>
14      <td>85</td>
15    </tr>
16    <tr>
17      <td>김영희</td>
18      <td>95</td>
```

```

19     <td>92</td>
20   </tr>
21 </tbody>
22 </table>

```

✅ `scope="col"` 은 해당 `<th>` 가 열 제목이라는 뜻 → 스크린 리더가 셀을 읽을 때 연관된 헤더를 같이 안내함.

✅ 예제 2: 복잡한 표에서 `headers` 와 `id` 사용

```

1 <table>
2   <caption>병원 진료 기록</caption>
3   <thead>
4     <tr>
5       <th id="name">환자명</th>
6       <th id="date">진료일</th>
7       <th id="dept">진료과</th>
8       <th id="diag">진단명</th>
9     </tr>
10  </thead>
11  <tbody>
12    <tr>
13      <td headers="name">이철수</td>
14      <td headers="date">2025-05-01</td>
15      <td headers="dept">내과</td>
16      <td headers="diag">감기</td>
17    </tr>
18    <tr>
19      <td headers="name">박민지</td>
20      <td headers="date">2025-05-02</td>
21      <td headers="dept">정형외과</td>
22      <td headers="diag">골절</td>
23    </tr>
24  </tbody>
25 </table>

```

🔴 이 구조는 특히 행/열 병합이 많은 복잡한 표에서 사용됨.
스크린 리더가 `headers` 속성을 따라가 해당 `td` 가 어떤 `th` 에 대응하는지 정확히 안내함.

✅ `caption` 태그의 활용

- 표의 제목/설명을 의미하며, 스크린 리더는 가장 먼저 이 요소를 읽음
- 디자인 상 보이지 않게 할 수도 있음 (`.sr-only` 클래스 등)

```

1 <caption class="sr-only">2025년 1학기 중간고사 성적표</caption>

```

✓ scope 속성 값 종류

값	의미
<code>col</code>	열 방향으로 적용되는 헤더
<code>row</code>	행 방향으로 적용되는 헤더
<code>colgroup</code>	열 그룹에 대한 헤더 (복잡한 구조용)
<code>rowgroup</code>	행 그룹에 대한 헤더

✓ ARIA 접근성 보조 속성 (선택적)

속성	설명
<code>role="table"</code>	사용자 정의 구조에서 명시적 역할 부여 가능
<code>aria-label</code>	<code>caption</code> 대체 텍스트
<code>aria-describedby</code>	표 전체에 설명 연결 가능 (ID 기반)

✓ 스크린 리더 동작 예시

화면에 표시된 내용	스크린 리더가 읽는 방식
85	영어, 홍길동, 85 점
내과	진료과, 이철수, 내과

→ 이것이 가능하려면 정확한 `scope`, `headers`, `id` 구조가 필요

✓ 한 줄 요약

테이블에서의 접근성은 `caption`, `scope`, `headers`, `id`를 통해 셀과 헤더의 논리적 관계를 명확히 지정함으로써 스크린 리더 사용자에게 올바른 정보를 제공하는 데 핵심이다.