

# HTML & CSS 빠른 참조

요소, 선택자, Flexbox, Grid, 반응형

## HTML 구조

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport"
  content="width=device-width, initial-scale=1.0">
<title>Page Title</title>
<link rel="stylesheet" href="style.css">
</head>
<body>
<h1>Hello World</h1>
<script src="app.js"></script>
</body>
</html>
```

## 주요 요소

### 텍스트 & 제목

```
<h1>Heading 1</h1> ... <h6>Heading 6</h6>
<p>Paragraph text</p>
<strong>Bold</strong> <em>Italic</em>
<br> <!-- line break -->
<hr> <!-- horizontal rule -->
```

### 링크 & 이미지

```
<a href="https://example.com">Link text</a>
<a href="#section">Anchor link</a>

```

### 목록

```
<ul> <!-- unordered -->
<li>Item</li>
</ul>
<ol> <!-- ordered -->
<li>First</li>
</ol>
```

## 폼

### 입력 타입

```
<form action="/submit" method="POST">
<input type="text" name="user"
  placeholder="Name" required>
<input type="email" name="email">
<input type="password" name="pw">
<input type="number" min="0" max="100">
<input type="checkbox" name="agree">
<input type="radio" name="size" value="M">
<button type="submit">Send</button>
</form>
```

### Select & Textarea

```
<select name="color">
<option value="r">Red</option>
<option value="g" selected>Green</option>
</select>
<textarea name="msg" rows="4"
  cols="30"></textarea>
```

## 시맨틱 HTML

```
<header> 소개 콘텐츠 또는 nav 컨테이너
<nav> 내비게이션 링크
<main> 페이지 주요 콘텐츠 (페이지당 하나)
<section> 주제별 콘텐츠 그룹화
<article> 독립적인 콘텐츠
<aside> 사이드바 또는 부가 콘텐츠
<footer> 섹션 또는 페이지 하단
<figure> 캡션이 있는 이미지/다이아그램
<figcaption> <figure>의 캡션
```

## CSS 선택자

### 기본 선택자

```
element p { } - 모든 <p> 요소
.class .card { } - 클래스 선택자
#id #main { } - id 선택자
* * { } - 전체 선택자 (모든 요소)
```

### 결합자

```
A B 자손 (A 안의 B)
A > B 직접 자식만
A + B 인접 형제 (A 바로 뒤의 B)
A ~ B 일반 형제 (A 뒤의 B)
```

### 의사 클래스

```
:hover 마우스 오버
:focus 요소에 포커스
:first-child 부모의 첫 번째 자식
:nth-child(n) n번째 자식 (2n=짝수, odd=홀수)
```

## 박스 모델

```
.box {
margin: 10px; /* outside */
border: 1px solid #333; /* edge */
padding: 15px; /* inside */
width: 200px;
box-sizing: border-box; /* include pad+border */
}
```

### 단축 표기

```
margin: 10px 모든 방향
margin: 10px 20px 상하|좌우
margin: 10px 20px 15px 상|좌우|하
margin: 10px 20px 15px 5px 상|우|하|좌
```

## Flexbox

### 컨테이너

```
.flex-container {
display: flex;
flex-direction: row; /* row | column */
justify-content: center; /* main axis */
align-items: center; /* cross axis */
gap: 10px;
flex-wrap: wrap;
}
```

## 항목 속성

```
flex: 1 사용 가능한 공간을 차지하도록 늘어남
flex-grow: 2 형제 대비 늘어나는 비율
flex-shrink: 0 기본 크기 이하로 줄어들지 않음
flex-basis: 200px 늘어나기/줄어들기 전 초기 크기
align-self: flex-end 컨테이너 정렬 재정의
order: -1 시각적 순서 (기본값 0)
```

## Grid

### 컨테이너

```
.grid {
display: grid;
grid-template-columns: 1fr 2fr 1fr;
grid-template-rows: auto;
gap: 10px;
}
```

### 항목 배치

```
.item {
grid-column: 1 / 3; /* span col 1-2 */
grid-row: 1 / 2;
}
/* shorthand: repeat, minmax */
grid-template-columns: repeat(3, 1fr);
grid-template-rows:
  repeat(auto-fit, minmax(200px, 1fr));
```

## 포지셔닝

```
static 기본 흐름 (offset 속성 없음)
relative 일반 위치에서 이동
absolute 가장 가까운 positioned 조상에서 이동
fixed 뷰포트에서 이동 (스크롤해도 고정)
sticky 스크롤 임계점까지 relative, 그 후 fixed
```

## 예시

```
.parent { position: relative; }
.child {
position: absolute;
top: 0; right: 0; /* top-right corner */
}
```

## 미디어 쿼리

```
/* Mobile-first: base styles for small screens */
.container { padding: 10px; }

@media (min-width: 768px) {
.container { max-width: 720px; }
}
@media (min-width: 1024px) {
.container { max-width: 960px; }
}
```

## 일반 중단점

```
480px 소형 스마트폰
768px 태블릿
1024px 소형 데스크톱 / 가로 태블릿
1280px 데스크톱
```