



프론트엔드 맛보기 워크북

HTML과 CSS

마크업 언어란? 태그 등을 이용하여 문서나 데이터의 구조를 명기하는 언어의 한 가지입니다. 프로그래밍 언어로 분류되진 않지만, 프론트엔드 개발자들에게엔 필수적으로 요구되는 역량이며, 입문 시 가장 처음으로 접하게 되는 언어이기도 하죠. 언뜻 Javascript나 프레임워크 사용에 비해 중요도가 낮아 보일 수 있지만, 웹 페이지의 뼈대를 이루는 만큼 프론트엔드 개발자라면 완벽하게 다룰 수 있어야 하죠. 제로베이스 프론트엔드 스쿨의 **<5시간 만에 완성하는 핵심 마크업>**의 일부 내용을 정리한 이 맛보기 워크북은 HTML, CSS의 개념과 주요 태그 및 속성에 관해 정리되어 있습니다. 내용을 보고 이해가 되지 않거나 어려워 보인다고 지레 겁먹지는 마세요. 제로베이스 스쿨의 기초 강의부터 차근차근 잘 따라가다 보면, 모두 자연스레 숙달하게 될 내용이니깐요. 제로베이스 스쿨은 새로운 시작을 맞이해 어엿한 한 명의 개발자로 거듭날 여러분의 도전을 응원합니다. 😊

HTML 기초

HTML이란?

HTML은 **Hypertext Markup Language**의 약자입니다. 웹 페이지를 표현하기 위해 사용하는 언어로, 기본적인 뼈대와 내용을 잡는 틀이라고 보시면 될 것 같네요. 웹 브라우저는 HTML 문서를 우리가 볼 수 있는 웹 페이지 형태로 변환해줍니다. 우리가 HTML 코드를 문서로 작성하면 크롬, 사파리 등 여러 브라우저는 눈에 보이는 형태로 출력해주는 역할을 하게 됩니다.

HTML 요소(Element)

요소(Element) = 태그(Tag) + 내용(Content)

태그는 반드시 여는 태그(Opening tag)와 닫는 태그(Closing tag) 한 쌍으로 구성됩니다. 어느 한쪽의 태그가 빠지면, 브라우저는 해당 코드를 인식하지 못합니다. 단, 빈 요소(Empty element)는 예외입니다.

예시:

여는 태그: <h1>, 닫는 태그: </h1>, 내용: This is the title!

```
<h1> This is the title!</h1>
```

빈 요소에 대해선 아래 내용을 확인해보세요.

빈 요소란?

여는 태그 – 닫는 태그 형태가 아닌 Self closing tag 형태로 작성

```
<태그이름 />
```

빈 요소의 예시: ,
, <hr>, <input> 등

HTML 속성(Attribute)

HTML의 요소는 속성을 가질 수 있으며, 속성은 여는 태그 안에 작성합니다.

사용법: 속성 이름="속성 값"

예시: a 태그의 속성

- href: 이동할 주소 지정
- target: 링크가 여리는 방법 지정

```
<a href= https://www.google.com/ target="_blank">구글 바로가기</a>
```

HTML 문서의 주석

주석(Comment) : 주석은 브라우저가 해석하지 않고 무시되는 설명문입니다. 즉, 소스 코드를 이해하기 쉽게 만들고 설명, 메모, 의견 등 협업을 위하여 입력되는 내용이죠. 또는 작성한 코드를 임시로 숨기고 싶을 때 사용하기도 합니다. HTML 주석은 아래와 같은 방식으로 사용됩니다.

```
<!--내용 --> *내용은 여러 줄이 들어갈 수 있음
```

HTML 문서의 구조

- HTML 문서는 **계층적**으로 작성되어야 합니다.

HTML 문서의 최상위 태그(root tag)는 <html>이며, 그 아래에 <head>와 <body>, 두 가지로 나뉘게 됩니다.

- **head** : HTML 문서의 메타데이터(HTML 문서에 대한 정보성 데이터) 포함

- `<title>` : 페이지 제목
- `<meta charset="utf-8">` : 여러가지 메타 정보
- `<link>` : favicon 적용, css 파일 연결 등
- `<style>` : css 작성
- `body` : 브라우저 화면에 표시되는 실제 콘텐츠

블록 요소와 인라인 요소

HTML 요소의 종류는 크게 두 가지로 나눌 수 있습니다.

1. 블록 요소
2. 인라인 요소

블록 요소(Block element or Block level element)

- 페이지의 구조적인 요소를 나타낼 때 사용합니다.
- 예시: `<p>`, `<div>`, `<form>` 등.
- 블록 요소 다음에는 반드시 **줄바꿈(New line)**이 들어갑니다.
- 블록 요소를 입력하게 되면, 반드시 한 줄 전체를 차지하며 표시하게 됩니다.

인라인 요소(Inline element)

- 문장, 단어 등 작은 부분에 주로 사용합니다.
- 예: ``, `<a>`, `<button>`, `` 등
- 인라인 요소 다음에는 줄바꿈이 들어가지 않습니다.
- 한 줄 모두 사용 x, 콘텐츠 크기만큼의 공간만 차지함.
- 블록 요소 사이에 인라인 요소가 들어갈 수 있음
 - 인라인 요소 안에는 블록 요소가 들어갈 수 없음(a태그 제외).
 - 인라인 요소 안에 인라인 요소가 들어갈 수 있음.

블록 요소와 인라인 요소의 가장 큰 차이점은 **줄바꿈 여부** 라고 할 수 있습니다. 당장은 태그를 사용하며 이 태그가 블록 요소인지, 인라인 요소인지 구분하기 어려울 수 있어요. 그래도 조금만 HTML을 사용해 문서를 작성하다 보면, 금방 감을 잡으실 수 있을 겁니다.

HTML 핵심태그

- **Heading**

- Heading 태그는 제목을 나타냅니다. `<h1>` 부터 `<h6>` 까지 총 여섯 개의 레벨이 있으며, h1이 가장 작고 h6이 가장 크게 표시됩니다. 단, 레벨을 중간에 건너 뛰는 건 일반적으로 그리 권장되지 않죠. Heading 태그는 블록 레벨 요소이기 때문에, 한 라인에 하나만 출력되게 됩니다.
- **Paragraph(p태그)**
 - P 태그는 문단과 문단을 구분짓는 태그입니다. P 태그 없이 본문 내용을 작성하면, Enter로 내용 구분을 시도해도 문단 구분이 되지 않죠. 그럴 때 p태그를 사용하면, 엔터를 친 것처럼 문단을 구분해 작성할 수 있다. P 태그 역시 블록 레벨 요소이기 때문에, 새로운 라인으로 시작하게 됩니다.
- **Image**
 - `` 태그로 웹 페이지에 이미지를 삽입할 수 있습니다. 위에 소개해 드렸던 빈 요소 중 하나이며, 추가로 `src 속성` 이 반드시 함께 사용되어야 하죠.
 - `src 속성` 이란? 삽입하고자 하는 이미지를 가리키는 속성입니다. Src에는 절대 경로, 상대 경로, 웹 주소 등의 요소가 들어가게 됩니다 .
 - `Alt` : 이미지가 보여지지 않을 경우, 사용자에게 이미지를 설명할 수 있는 글입니다. 경로가 잘못되었거나 네트워크에 문제가 있을 경우, `alt="The head a dinosaur skeleton">`과 같은 식으로 사용하게 됩니다.
 - `Width, height` : 넓이와 높이를 설정해, 이미지의 크기를 지정할 수 있습니다.
- **Anchor 태그(<a>) - 하이퍼링크**
 - 주요 속성
 - `Href 속성` (이동할 링크의 주소)을 필수로 함께 사용해야 합니다. Href에는 프로젝트 내의 다른 HTML 파일, 웹 페이지 주소 등이 들어갈 수 있습니다.
 - `Title` : 마우스를 hover했을 때 표시되는 추가적인 제목이나 설명입니다.
 - `Target` : 페이지를 열 방법 지정하는 속성입니다.
 - `_self` : 현재 페이지에서 열기(Default)
 - `_blank` : 새 탭 or 새 창으로 열기
 - `_parent` , `_top` 등등
 - **이미지 링크**
 - `a 태그` 는 텍스트뿐 아니라, 어떤 HTML 요소든 올 수 있습니다. 주로 이미지를 사용하는 경우가 많은 편입니다.
 - `List`

- 목록(List)을 나타내는 태그는 `` 과 ``, 두 가지가 있습니다. 차이점은 아래와 같습니다.

``: 순서가 있는 리스트로, 기본 스타일은 아라비아 숫자

``: 순서가 없는 리스트로, 기본 스타일은 Bullet poin(점)

목록을 만들 땐 우선 상위에 리스트의 타입을 명시하고, 하위에 `` 태그를 통해 리스트의 항목을 나열하는 식으로 사용합니다. 더불어 리스트는 중첩, 교차해서 작성할 수 있습니다.

- **Table**
 - HTML로 표를 표현할 때 사용하는 태그입니다. 관련 핵심 태그로는 `<table>`, `<tr>`, `<td>`가 있으며, 선택적으로 `<thead>`, `<tbody>`, `<tfoot>`를 사용할 수 있습니다.
 - 사용 방법
 1. `<table>` 태그로 테이블을 선언한다.(전체 테이블을 감싸는 태그)
 2. `<tr>` 로 테이블의 row(행, 가로)을 한 줄씩 추가한다.
 3. `<tr>` 안에 테이블의 셀(Cell, Column, 열)을 채워준다.
 4. 맨 첫째줄이라면, `<th>` 로 테이블 헤더를 만들어준다.
 5. 헤더 아니라면, `<td>` 로 일반 데이터를 채워준다.
- **Div & Span**
 - 레이아웃을 구성하기 위해 사용하는 태그입니다. 아무것도 표현하지 않아 눈으로 보이는 변화는 없죠. 단, HTML 코드에서 영역을 나누기 위해 사용하는 일종의 Container 역할을 하게 됩니다.
 - ***Div는 블록 레벨 요소이며, span은 인라인 요소**입니다. 블록 레벨 요소와 인라인 요소에 대해선 위에서 이미 배웠죠?
- **전역 속성 (Global attributes)**
 - 전역 속성은 거의 모든 HTML 요소에서 공통적으로 사용할 수 있는 속성입니다. 대표적으로 `class`, `id`, `style`, `hidden`, `title` 등이 있습니다.
- **시멘틱 태그**
 - HTML5에서 등장한 개념이며, 의미를 갖고 있는 요소를 의미합니다. 코드를 보는 개발자에게 명확하게 해당 요소의 의미, 목적을 전달하며 검색 엔진 최적화(SEO)와 접근성 등을 위해 사용합니다. 시멘틱 태그가 등장하기 이전에

는 구조를 구분하기 위해 div 태그에 id 또는 class 등이 사용되었으나, 시멘틱 태그가 등장하며 보다 보기 쉽고 직관적인 구조를 만들 수 있게 되었습니다.

CSS 기초

CSS란? **Cascading Style Sheet**의 약자로, HTML로 작성한 웹 페이지에 스타일을 적용하기 위해 사용되는 언어입니다. HTML이 내용과 구조를 만드는 언어라면, CSS는 이를 색칠하고 위치나 크기를 바꾸는 식으로 적용됩니다.

캐스케이딩이란 계단식이라는 뜻을 포함하고 있는데요. CSS는 위에서 아래로 흐르는 형태로 작성해 사용됩니다. 이를 위해 아래의 내용을 고려해 작성해야 하죠.

- 상속: 상위 요소의 스타일이 하위 요소에게 상속됨
- 우선순위: 여러 개의 CSS rule을 동시에 선언했을 때, CSS 적용의 우선순위가 있음
- 디자인(색, 폰트 등)과 위치, 배치, 레이아웃, 다양한 기기/스크린 크기 대응 등을 위해 사용
-

CSS의 주요 문법으로 고려해야 할 것은 아래와 같습니다.

선택자(selector) : 해당 CSS를 적용할 타겟 요소

속성(property) : CSS 속성의 이름(color, font-size, display ...)

값(property value) : 속성의 값(속성마다 허용되는 값의 형태가 정해져있음)

여러 속성을 작성하기 위해 세미콜론(;)으로 구분

CSS의 주요 속성

- **Color**
 - CSS의 색상을 표현하는 방법은 여러 가지가 있습니다.
- 1. 미리 정해진 색상 이름(blue, red, yellow...)으로 표시하는 방법입니다. 단, 현업에서는 디자인에서 정확한 색상 코드(16진수)로 넘어오는 경우가 많기 때문에, 잘 사용되는 방법은 아닙니다.
- 2. Rgb or rgba 값으로 표시하는 방법입니다. 각 색깔 별 0~255까지의 값으로 각각의 수치를 표현하게 됩니다.

3. 16진수 코드로 표하는 방법입니다. 두 자리씩 각각 R, G, B를 표현하며 #000000 : black / #FFFFFF : white / #FF0000 : Red / #00FF00 : Green / #0000FF : Blue과 같은 식으로 표시됩니다.

- **Background**

- 배경 스타일링을 지정하는 방법입니다. 배경 스타일에 관련된 CSS 속성은 아래와 같습니다.

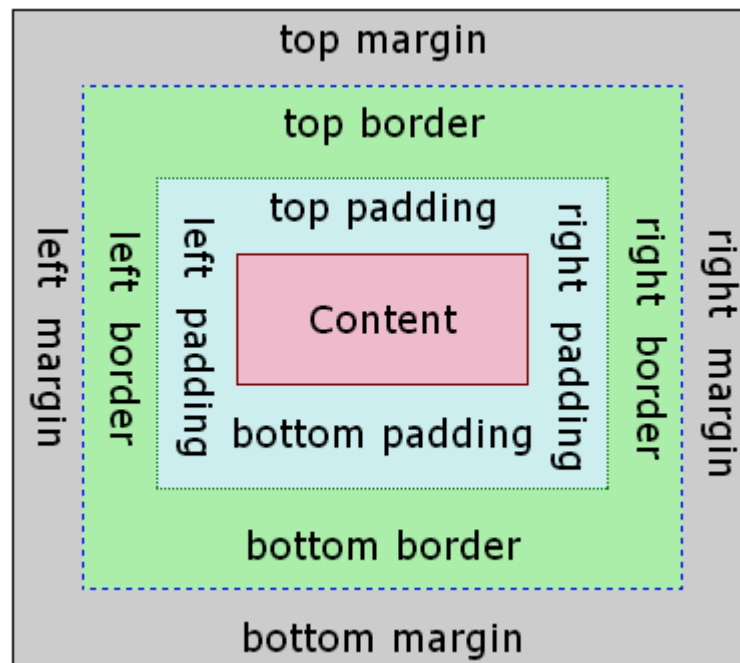
- `background-color` : 배경색
- `background-image` : 배경 이미지
 - 값 작성 방법 : url("이미지주소")
 - 절대경로, 상대경로, 웹주소
- `background-repeat` : 배경 이미지 반복 여부
- `no-repeat`
- `repeat`
- `repeat-x` (가로 반복), `repeat-y` (세로 반복)
- `background-position` : 배경 이미지의 위치 지정
 - 기본 키워드 : top, bottom, left, right, center
 - 퍼센트, 길이 등 지정
 - 예) right 20%, right top, center top

- **Fonts**

- 폰트와 관련된 CSS 속성입니다. 주로 폰트 사이즈와 색깔, 굵기, 서체 등을 지정하게 됩니다. 자세한 내용은 아래와 같습니다.

- `font-size`
 - **px**
 - 고정값, 절대 단위
 - **em**
 - 가변 크기(반응형 웹), 상대 단위
 - 상위 요소의 폰트 사이즈를 상속받음
 - 1em == 현재 브라우저에서 표시되는 폰트 크기

- 브라우저의 default 1em == 16px
- rem
- `font-style`
 - normal
 - italic
- `font-weight`
 - normal, bold
 - 숫자 값
- `font-family` : 글꼴(서체) 지정.
- `Text-align`
 - 텍스트의 가로 정렬을 위해 사용하는 속성으로, `left` (기본), `right`, `center`, `justify` : 양쪽 정렬 등이 있습니다. 이외에도 `text-align-last` 을 통해, 텍스트 마지막 줄의 text-align을 지정할 수 있습니다.
- `Box model`
 - 모든 HTML 요소는 box로 표현됩니다. 즉, 모든 요소는 네모로 표현되어 있으며, 다른 모양을 갖는 요소는 없죠. 기본적인 구조는 아래 이미지와 같습니다.



◦

디자인을 정확하게 적용하기 위해선 반드시 요소의 총 크기를 계산할 수 있어야 합니다. 박스 모델의 총 크기는 content 크기 + 사방 border + 사방 padding으로 구성되죠. Box sizing 속성의 값을 어떻게 주느냐에 따라 요소(박스)의 총 크기를 화면에 표시하는 방식이 달라지게 됩니다.

- **content-box** (default) : 오직 content의 크기만 width, height로 지정
 - 총 width(height) = content 크기(width, height 속성으로 준 값) + padding + border
- **border-box** : padding, border를 포함하여 크기 지정. simple!
 - 총 width(height) = width, height 속성으로 준 값(content, padding, border가 다 포함되어있음)
 - (각 조직마다 다르지만) 최근에는 개발자가 디자인에 맞춰서 코드를 작성하고 계산하기 편하기 때문에, 거의 필수로 border-box로 지정해주는 경우가 많음
- **Border**
 - 테두리 스타일을 지정하는 속성입니다. **border-style** 을 통해 **dotted** , **dashed** , **solid** (실선) 등의 모양을, **border-width** 을 통해 테두리 길이를, 이외에도 **color** 을 통해 색상, **radius** 를 통해 모서리 둥글기 등을 설정할 수 있죠. 또한 **border-right** , **border-left** , **border-bottom** , **border-top** 을 통해 사방 중 하나를 지정하여 스타일을 따로 지정할 수도 있습니다.
- **Margin**
 - **Margin** 은 border 외부, 즉 요소와 요소 사이의 여백을 지정하는 속성입니다. Border와 마찬가지로 사방 중 하나를 지정해서 여백을 줄 수도 있죠. Margin 속성에 들어가는 값은 아래와 같습니다.
 - 수치(length) : **px** , **em** , **rem** , **pt** , **cm** ...
 - 기본값은 0 : margin을 아무것도 지정하지 않았을 때
 - 양수, 음수 모두 사용 가능
 - 퍼센트(%) : 감싸고 있는 컨테이너(부모 컨테이너)의 대비 몇%인지
 - auto : 브라우저가 계산한 값 자동 적용 (대부분 균등하게 분배)
- **Padding**
 - **Padding** 은 border 내부의 여백, 즉, content 주위를 둘러싸는 여백을 뜻합니다. Padding 역시 마찬가지로 사방 중 하나를 지정해 설정할 수 있으며, padding 속성에 들어가는 값은 아래와 같습니다.
 - padding : border 내부 여백(content 주위를 둘러싸는 여백)

- 사방 중 하나를 지정해서 줄 수 있음 : **padding-top, padding-bottom, padding-left, padding-right**
- padding 속성에 들어가는 값
 - 수치(length)
 - 음수 양수만
 - 퍼센트(%)

참고로, margin, padding의 shorthand는 자주 사용하므로 **top right bottom left** 로 순서를 외워두는 것이 좋습니다. 값을 하나만 적으면 네 면 모두, 두 개만 적으면 위아래, 양옆으로 적용되며 세 개를 적으면 위, 양옆, 아래 순으로 적용되게 됩니다.