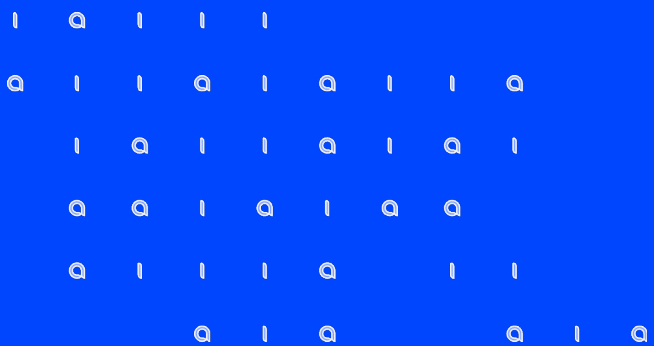


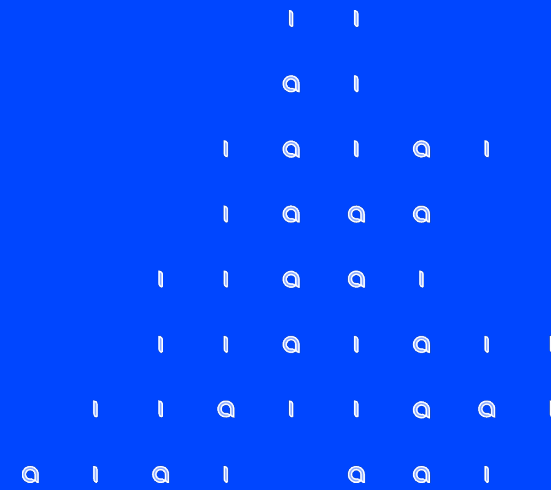
[신한투자증권] 프로 디지털 아카데미

클라우드 기반 금융 ICT개발자 양성과정

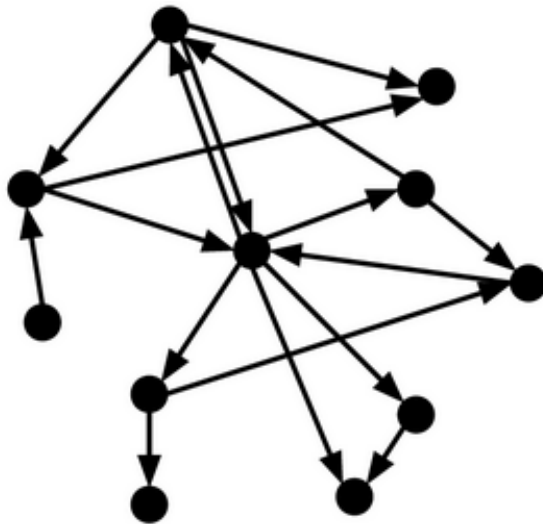
UI/UX 디자인과 구현 (HTML/CSS)



1. 웹 구성요소 알기



- 이를 통해 우리는 전 세계의 문서, 이미지, 비디오, 애플리케이션 등에 접근하고 정보를 공유한다.
- 하이퍼링크를 통해 문서들을 연결하여 네비게이션할 수 있는 구조를 가지고 있다.



2. 웹의 구성 요소

규모가 있는 웹이 만들어지려면 일반적으로 프론트엔드와 백엔드 개발이 필요하다.

프론트엔드

- 사용자가 볼 수 있는 부분
- 디자인, 웹의 동작, 간단한 로직

백엔드

- 사용자에게 보이지 않는 부분
- 데이터 저장/가공, 복잡한 로직

2. 웹의 구성 요소



웹 프론트엔드를 만드는 3가지 언어

HTML

- 웹 페이지를 만드는 언어
- 웹 페이지의 구조를 정의하는 마크업 언어

CSS

- 웹 페이지의 디자인과 스타일을 정의하는 스타일 시트 언어.
- HTML 요소들을 꾸며주는 역할

JavaScript

- 웹 페이지의 동적인 기능을 담당하는 프로그래밍 언어.
- 데이터를 바꿀 수 있다 (백엔드와 소통)
- 복잡한 애니메이션을 추가할 수 있다

3. 웹에서 알아야 하는 기본 개념

웹은

- 클라이언트-서버 모델을 통해 사용자는 웹 브라우저를 통해 정보를 요청하고, 서버는 요청된 정보를 제공
- HTTP 프로토콜을 기반으로, 요청(request)과 응답(response)의 형태로 데이터를 주고 받는다.
- HTTPS은 보안이 강화된 HTTP 프로토콜로, 데이터 암호화를 통해 안전한 통신 환경을 제공한다.
- URL은 리소스 위치 지시자로, 웹에서 문서의 위치를 찾는 데 사용된다.
- 웹 페이지 렌더링은 브라우저가 HTML, CSS, JS를 해석하여 사용자에게 페이지를 보여주는 것을 의미한다

웹 개발자라면

- 웹 표준과 웹 접근성, HTML와 네트워크 기초, 쿠키, 웹 보안, 브라우저 호환성 등의 개념을 알아야한다.

[참고] 웹 사이트의 쿠키와 세션을 사용하는 목적은 무엇인가요?

빈출 면접 질문에 중요한 개념들이 많이 담겨있으니 참고해보자

1. 웹 사이트의 쿠키와 세션을 사용하는 목적은 무엇인가요?
2. 웹 애플리케이션에서의 캐싱 방법과 목적은 무엇인가요?
3. SEO(Search Engine Optimization)에 대해 설명해주세요.
4. SPA(Single Page Application)의 장단점을 설명해주세요.
5. Responsive Web Design과 Mobile-First Design의 차이는 무엇인가요?
6. RESTful API의 특징과 이점은 무엇인가요?

4. 웹 개발자 단골 면접 질문

- 7. HTML, CSS, JavaScript의 차이점은 무엇인가요?
- 8. 웹 표준이란 무엇이며 왜 중요한가요?
- 9. 웹 접근성에 대해 설명해주세요.
- 10. 브라우저 호환성을 위해 어떤 접근 방식을 사용하나요?
- 11. HTTP와 HTTPS의 차이점을 설명해주세요.
- 12. 웹 보안을 위한 HTTPS의 동작 방식을 설명해주세요.

4. 웹 개발자 단골 면접 질문

- 13. DNS(Domain Name System)의 역할은 무엇이며 어떻게 작동하나요?
- 14. CDN(Content Delivery Network)이란 무엇이며 왜 사용하나요?
- 15. HTML5에서 추가된 새로운 기능들은 무엇이 있나요?
- 16. Progressive Web Apps(PWA)의 특징은 무엇인가요?
- 17. 특정 웹사이트의 로딩 속도를 향상시키는 방법을 아는대로 나열해보세요.

4. 웹 개발자 단골 면접 질문

18.웹 개발에 필요한 보안 이슈들을 아는대로 나열해보세요.

19.웹 애플리케이션 보안을 위해 취할 수 있는 조치들은 무엇인가요?

20.CORS(Cross-Origin Resource Sharing)란 무엇인가요?

21.XSS(Cross-Site Scripting) 공격이란 무엇인가요? 방어 방법은?

22.CSRF(Cross-Site Request Forgery) 공격이란 무엇인가요? 방어 방법은?

[알쓸신IT] DNS는 왜 돈주고 사야할까? (1)

웹 사이트에 접속하면 보이는, 도메인 주소는 왜 돈주고 사야할까? ([도메인 가격 검색](#))
(돈은 대체 누구한테 가고, 누가 도메인을 만드는 걸까?)

인터넷에 연결된 웹 사이트는 모두 IP 주소를 갖고 있다.

IP가 웹 사이트에 접속할 수 있는 실제 주소다.

하지만 IP주소는 긴 숫자(223.130.200.104)로 구성되어 있어서 외울 수가 없다.

그래서 생겨난 것이 도메인 시스템이다.

도메인 시스템은 전화번호부처럼 이름과 IP 주소가 짝지어 저장되어 있는 시스템이다.

[알쓸신IT] DNS는 왜 돈주고 사야할까? (2)

그런데 도메인이 수백만 개가 넘어서, 도메인 시스템을 관리하는 시스템이 따로 있다.
전화번호부를 관리하는 서브 전화번호부가 있는 것과 비슷하다.

이러한 서브 전화번호부를 레지스트리(registry)라고 한다.

레지스트리는 기업에서 운영하고, 가장 유명한 곳이 닷컴(.com)이다.
(닷컴 관리는 Verisign이라는 회사에서 한다)

레지스트리는 수익성이 꽤 좋은 사업이지만, 신경쓸 것도 많다.
도메인을 입력하면 이를 IP 주소로 빠르게 연결할 수 있어야 하고
보안 요소를 구축하고 있어야 한다.

[알쓸신IT] DNS는 왜 돈주고 사야할까? (3)

하지만, 레지스트리를 관리하는 기업은 도메인을 만드는 역할을 할 뿐,
도메인을 직접 판매하지는 않는다.

우리가 도메인을 사고 싶을 때 verisign을 방문하지 않는다.

도메인은 GoDaddy, Gabia, Whois같은 리셀러에게서 사야한다

리셀러가 필요한 이유는 '도메인을 레지스트리에 등록하는 과정이 매우 복잡하기 때문'이다.

- 회원 관리, 결제, 도메인 관리 등

그래서 리셀러가 도메인 등록을 대신해주고
유저에게 실제 도메인 비용보다 더 비싼 금액을 받는다.

[알쓸신IT] DNS는 왜 돈주고 사야할까? (4)

그럼 혹시 우리도 새로운 레지스트리를 운영할 수 있을까? (www.***.minjeong?)

할 수 있다!! ICANN(아이캔)에 신청만 하면된다.

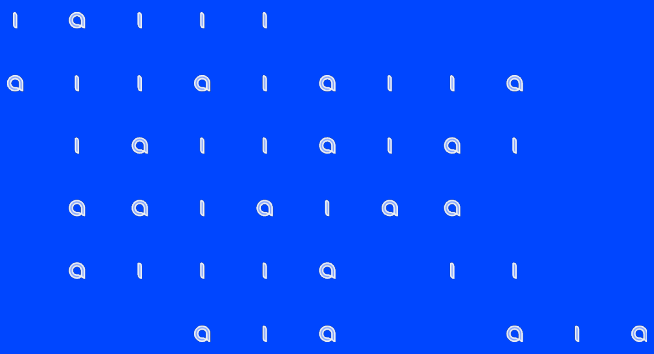
도메인을 관리하는 비영리 기업인데, 최상위 전화번호부 역할을 하는 곳이다.

하지만, 신청 비용이 엄청나다. 18만 5천달러 정도로 한화로 2억 2천만원 쯤 된다.

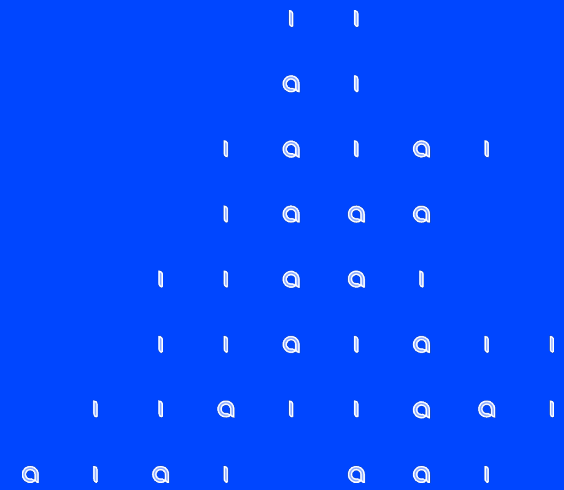
그리고 인터넷 환경, 도메인을 운영할 수 있는 인프라 기술, 고객 관리 능력 등을 갖추고 있는지 ICANN에 증명해야한다.

핵심요약

- 웹은 전 세계적으로 연결된 컴퓨터 네트워크를 통해 정보를 공유하는 공간
- 규모있는 웹을 위해서는 프론트엔드와 백엔드 개발 필요
- 웹 프론트엔드를 만드는 3가지 언어는 HTML, CSS, Javascript



2. HTML 시작하기



학습목표

- ▶ HTML의 기본 개념과 역할을 이해한다.
- ▶ HTML5의 특징과 장점을 이해한다.
- ▶ HTML 문서 구조를 파악하고 작성할 수 있다.

1. HTML란?

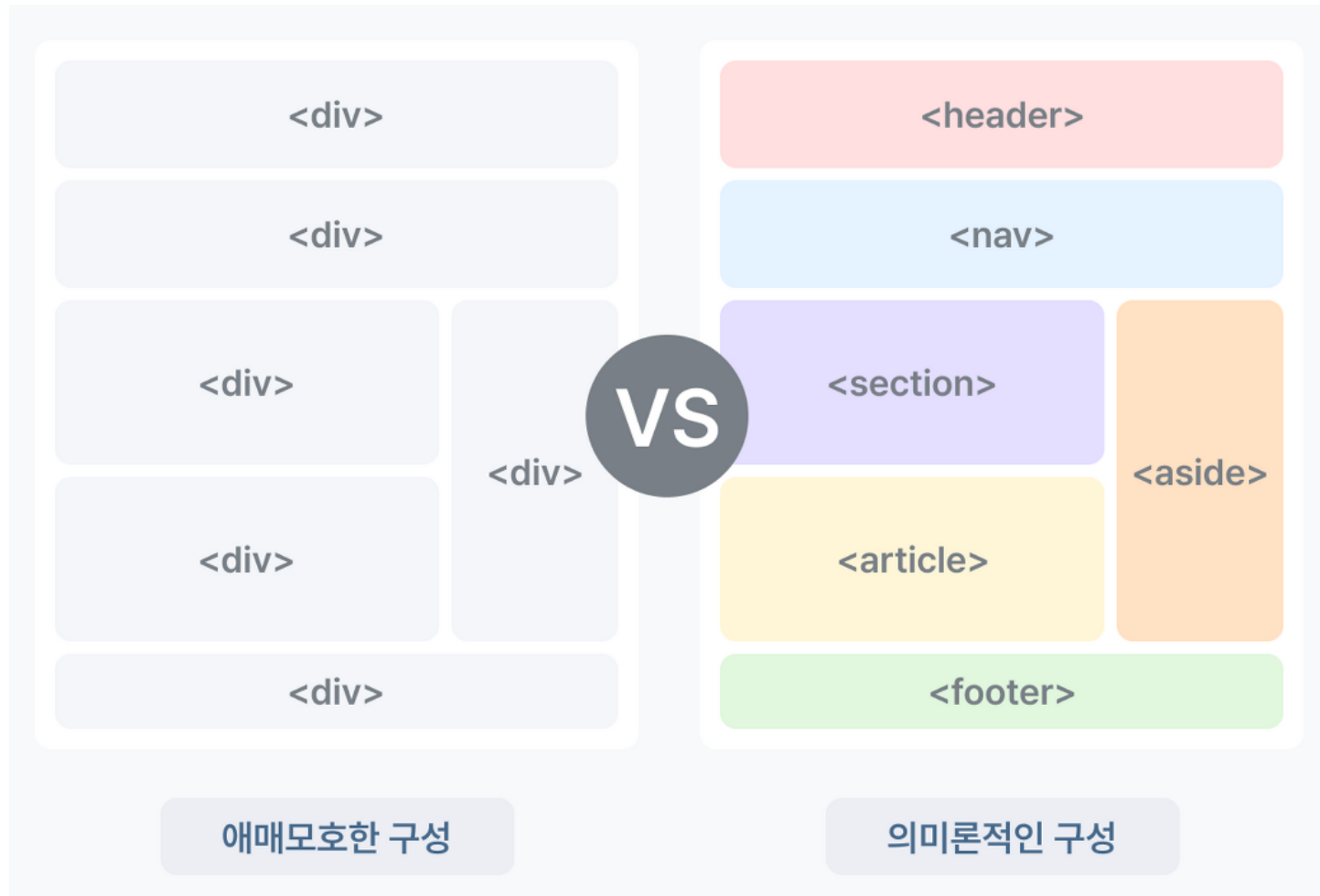
HTML(HyperText Markup Language)이란?

- 웹 페이지를 만들기 위한 기본적인 구조를 정의하는 언어
- 하이퍼텍스트를 작성하고 구조화하는 언어
- 웹 브라우저가 이해하고 해석하여 사용자에게 보여주는 언어



```
<body>  
  <h1>제목 내용</h1>  
  <div>웹 브라우저를 통해 보여지는 내용</div>  
</body>
```

2. 웹 페이지의 구조와 역할



3. HTML5 버전 출시?

HTML5는 정식으로 나온 것이 아니라

여러 개선과 새로운 기능들이 단계적으로 추가되면서 발전해 왔다.

대중적으로 HTML5라고 불리기 시작한 시기는 보통 2009년부터 2014년 사이로 알려져 있다.

그간 웹 표준화 기구인 W3C(World Wide Web Consortium)와

WHATWG(Web Hypertext Application Technology Working Group) 등에서

지속적으로 업데이트와 표준화 작업이 이루어졌고,

이러한 작업들을 통해 HTML5라는 용어가 보다 널리 사용되었다.

따라서 HTML5는 하나의 특정한 날짜에 정식으로 출시된 것이 아니라,

기존의 HTML 표준에 지속적으로 새로운 기능들을 추가하고 업데이트하여 완성된 형태로 사용되고 있다.

4. HTML5로의 변화

HTML5는 현대적인 웹 개발에서 핵심적이고 필수적인 기능들을 위한 변화와 개선이 이루어졌다.

1. 새로운 요소와 API 추가

- HTML5는 시맨틱 요소들(<header>, <footer>, <section>, <article> 등)을 포함하였고, 새로운 요소들을 도입하여 웹 문서의 의미론적 구조를 더 명확히 표현할 수 있다.
- 오디오, 비디오, 그래픽, 로컬 저장소 등을 다루는 API도 추가

2. 멀티미디어 지원 강화

- <video>와 <audio> 태그를 통해 더욱 쉽게 비디오와 오디오를 웹에 삽입할 수 있게 되었다.
- 기존에는 플래시와 같은 외부 플러그인이 필요했던 기능들을 브라우저 자체에서 지원하게 되었다.

3. 폼 관련 개선

- 새로운 입력 유형과 속성이 추가되어 폼 요소들의 사용성과 유효성 검사가 개선되었다.

4. 웹앱 편의성 증가

- 오프라인 웹 애플리케이션 및 Geolocation, Local DB등을 지원하여 웹앱 개발의 편의성이 높아졌다.

5. HTML5로의 변화, 무엇이 좋아졌을까?

1. 모바일 및 웹앱 개발 용이

- HTML5는 모바일 환경에서의 웹앱 개발이 용이하도록 지원
- 반응형 웹 디자인과 모바일 친화적 기능들을 지원
- 다양한 디바이스에서 최적화된 경험을 제공

2. 새로운 기능 및 향상된 성능

- 새로운 태그와 API는 다양한 멀티미디어 요소를 쉽게 조작할 수 있게 함
- 사용자에게 더 풍부하고 동적인 경험을 제공함

3. SEO 및 웹 접근성

- 시맨틱 요소들은 검색 엔진 최적화(SEO)에 도움을 줌
- 웹 접근성을 높여 시각, 청각 장애를 가진 사용자들에게 더 나은 경험을 제공

6. HTML4 vs HTML5 비교

기능/특징	HTML4	HTML5
마크업	<div>와 으로 주로 구분	시맨틱 요소 (<header>, <footer> 등) 도입
멀티미디어	외부 플러그인(Flash 등) 필요	<video>, <audio> 등 멀티미디어 지원
폼	폼 요소 유형 제한	새로운 입력 유형 및 속성 추가
그래픽 및 그래픽 API	부족	<canvas>를 통한 그래픽 처리 및 WebGL 지원
웹 앱	웹 앱 지원 부족	오프라인 웹 앱, 지오로케이션 API 등 지원
퍼포먼스 및 성능	성능 제한	향상된 성능 및 브라우저 호환성
웹 접근성	시맨틱 태그 부재로 낮은 접근성	시맨틱 태그 도입으로 높은 접근성

[참고] “HTML5를 이해하고 잘 활용할 수 있습니다”

HTML5를 이해하고 잘 활용할 수 있습니다.

⇒ 기본 HTML 문법을 이해하고 있고, 특히 시맨틱 태그, 멀티미디어(video, audio) 태그, form 태그의 문법, 유효성 검사들을 알고 있고 잘 활용할 수 있다.

⇒ 새로운 기술을 익힐 때, 이론/문법/역할/특징을 열심히 공부한다.
그리고 공부한 것을 실제로 코드에 잘 적용한다.

⇒ 어떻게 공부하나요?

⇒ 공식 튜토리얼 문서로 공부한다.

7. HTML 문법 : 태그

〈태그명 속성명1="속성값1" 속성명2="속성값2"〉 콘텐츠 〈/태그명〉

태그는 콘텐츠를 감싸서 그 정보의 성격과 의미를 정의 한다.

`생활코딩`

태그

`생활코딩`

콘텐츠

열리는 태그가 있으면 닫히는 태그가 있어야 한다.

`생활코딩`

열리는 태그

닫히는 태그

닫히는 태그는 태그 명 앞에는 '/'가 붙는다.

`생활코딩`

8. HTML 기본 템플릿

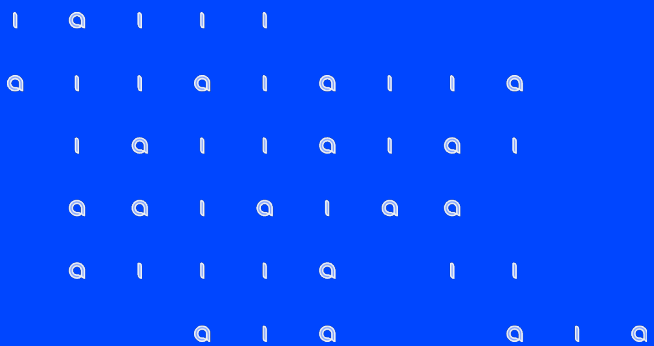
```
<html>  
  <head>  
    웹 브라우저의 정보  
  </head>  
  <body>  
    웹 브라우저를 통해 보여지는 내용  
  </body>  
</html>
```

[참고] HTML 기본 템플릿

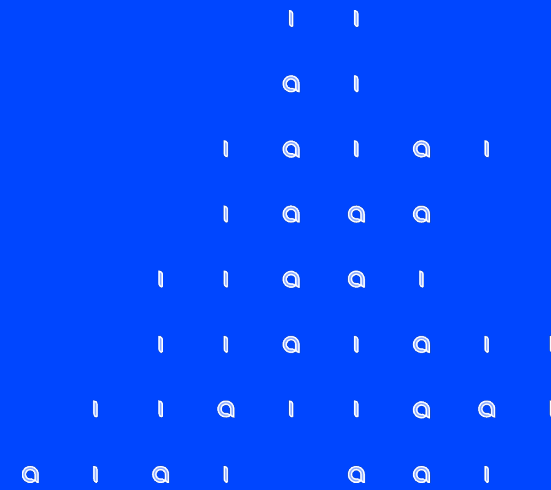
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title> 상태바</title>
</head>
<body>
  웹 브라우저를 통해 보여지는 내용
</body>
</html>
```

핵심요약

- HTML은 웹 페이지를 만들기 위한 기본적인 구조를 정의하는 언어
- HTML5로 변화하며 현대 웹 개발에 많이 쓰이는 기능들이 생겨났다
- HTML은 태그를 기반으로 작성된다



3. 개발 환경 구성



학습목표

- ▶ HTML/CSS를 개발할 수 있는 환경을 준비한다,
- ▶ 생산성에 도움이 되는 VS CODE 익스텐션을 설치한다.
- ▶ 크롬 개발자 도구 사용법을 익힌다.

1. VS CODE 설치하기

공식 사이트

- <https://code.visualstudio.com/>

추천 VS CODE 익스텐션

- Live Preview : HTML을 미리보기 서버로 볼 수 있음
- Material Icon Theme : 파일 아이콘을 깔끔하고 보기 좋게 변경
- Prettier : 코드 포매팅 정리
- Auto Rename Tag : 시작 태그 이름 변경 시 닫는 태그 이름 자동 변경
- CSS Peek : html에서 CSS 선택 시 찾기 기능

2. Github 저장소 만들기

- 1) Github repository에 프로젝트를 생성합니다. (README.md 포함)
- 2) Git clone을 하고 VS Code로 열어줍니다.
- 3) 그 폴더에 index.html 파일을 하나 만들어줍니다.
- 4) Git commit & Push 합니다.

[참고] 유다시티 커밋 메시지 추천

제목은 모두 현재형으로 작성

영어라면 명령조, 한글이라면 구문으로 작성

예) 한글 커밋 메시지

- feat: Home 지역별 여행지 소개 섹션 추가
- docs: README.md 데모 링크 추가

예) 영문 커밋 메시지

- feat: Add migration to update empty routerLinks
- refactor: remove unused import

[참고] 유다시티 커밋 메시지 추천

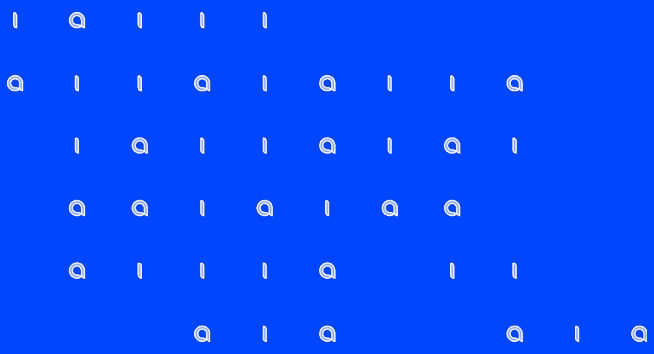
‘유다시티 커밋 컨벤션 가이드’는 7가지 타입 중, 하나를 사용할 것을 권장

태그 이름	설명
feat	새로운 기능 추가
fix	오류 또는 버그 수정
docs	문서 관련 변경사항
style	코드 스타일 변화 (product code와 관련 없는 포맷이나 세미콜론을 놓친 것 등)
refactor	기능과 관계없는 코드의 리팩토링
test	test를 추가하거나 수정했을 때
chore	build와 관련된 부분, 패키지 매니저 설정 등

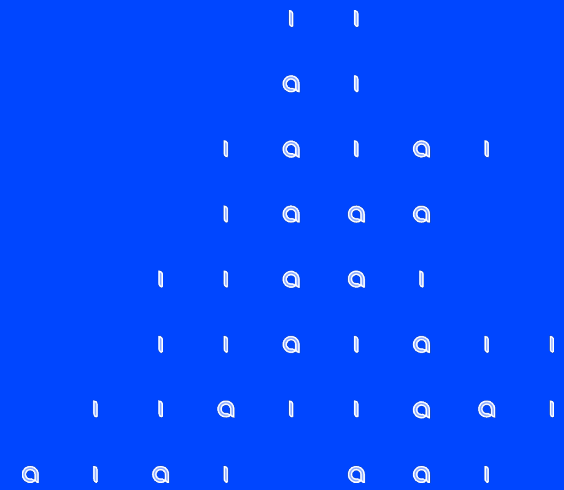
실습

- VS Code를 설치하고 필요한 익스텐션을 설치합니다.
- Github 레파지토리를 생성하고 폴더에 연결합니다.





4. HTML 문법 시작



학습목표

- ▶ HTML의 태그의 구성을 이해하고 활용할 수 있다.
- ▶ 주석, 속성 등을 이해하고 활용할 수 있다.

1. HTML 태그

HTML은 웹 페이지에 콘텐츠를 표시하기 위해 사용하는 언어
어느 부분에 텍스트가 있어야 할지, 어느 부분에 이미지가 있어야 할지 등을 나타내야한다.

이러한 역할을 수행하기 위해 HTML은 '태그(tag)'라는 표기법을 사용한다.

<태그이름> 콘텐츠 </태그의이름>

1. HTML 태그

여는 태그와 닫는 태그를 짝 지어 사용하는 것은, 그 사이에 콘텐츠를 표시하기 위해서이다.

```
<h1> 점심 메뉴는 훌륭했다 </h1>
```

콘텐츠가 필요하지 않은 태그의 경우 '단일 태그' 형태로도 사용한다.

- 단일 태그는 닫는 태그를 사용하지 않는 태그이다.
- 단일 태그는 태그 그 자체만으로도 콘텐츠가 되거나, 별도의 기능을 수행하는 경우 사용한다.
 - 일반적인 태그 : <태그이름></태그이름>
 - 단일 태그 : <태그이름/>

```
<input type="text" />  

```

2. 태그 작성 전 알아야 할 것

- 태그는 소문자로 쓴다.
- 여는 태그와 닫는 태그를 정확히 입력해야 한다 (단일 태그는 예외)
- 태그 안에 태그를 넣을 수 있다.
- 들여쓰기를 적절하게 사용하는 것이 좋다

3. 속성

- 속성(attribute)은 태그에 부가 기능을 추가하기 위해 작성하는 것이며, 선택 사항이다.
- 하나의 태그에 속성의 개수 제한이 없다.
- 속성, 태그 사이에 공백을 반드시 써줘야한다.

<태그의이름 속성명="속성값">콘텐츠</태그의이름>

<태그의이름 속성명="속성값">

```

```

3. 주석

- 주석(comment)은 사람에게에는 보이지만, 웹 브라우저에게는 보이지 않는 코드이다.
- 주석은 주로 코드 작성자가 자신이 작성한 코드에 설명 또는 메모를 추가하기 위해 사용한다.

주석 문법

<!-- 이 안에 쓰인 내용은 웹 브라우저에게 아무런 영향도 주지 못합니다 -->

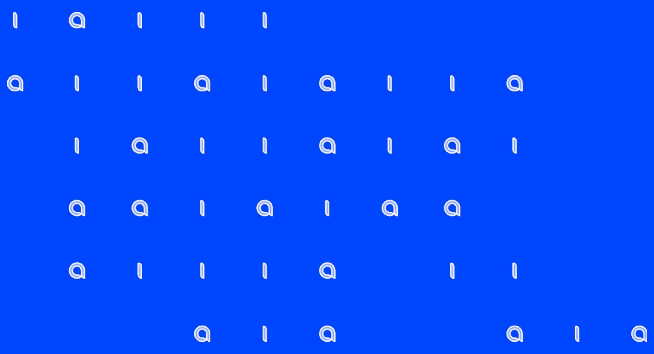
```
<!-- 이 안에 쓰인 내용은 웹 브라우저에게 아무런 영향도 주지 못합니다 -->
```

4. HTML 문서 기본 구조

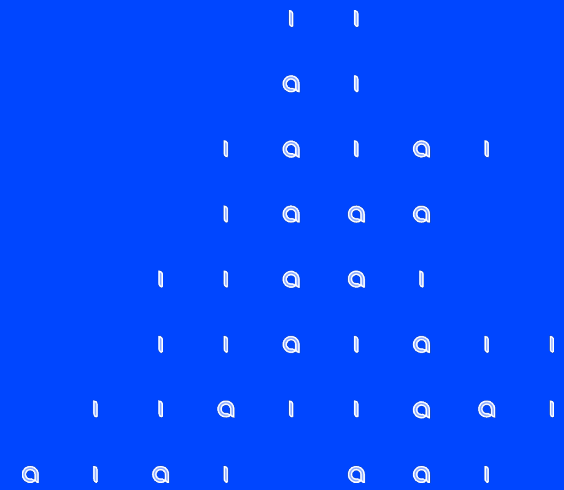
- 문서 형식 선언, `<!DOCTYPE html>`
- 문서의 시작과 끝, `<html lang="ko">`
- 문서의 정보, `<head>`
- 인코딩 방식 설정, `<meta charset="utf-8">`
- 문서의 제목, `<title>`
- 화면에 표시될 콘텐츠, `<body>`

4. HTML 문서 기본 구조

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>상태바</title>
</head>
  <body class="hello">
    웹 브라우저를 통해 보여지는 내용
  </body>
</html>
```



5. HTML 문법 - 글



1. 글 관련 태그

제목

- <h1> ~ <h6>
- 숫자가 1일 때 가장 크고 6일 때 가장 작다.

숫자가 1일 때의 제목 텍스트

숫자가 2일 때의 제목 텍스트

숫자가 3일 때의 제목 텍스트

숫자가 4일 때의 제목 텍스트

숫자가 5일 때의 제목 텍스트

숫자가 6일 때의 제목 텍스트

2. 글 관련 주요 태그

문단

- <p>
- 글의 본문을 표시할 때 주로 사용

<p> 문단을 나타낼 때 사용하는 태그입니다.</p>
<p> 하나의 문단에는 일정한 여백이 형성되어 화면 상
에서 구분하기가 용이합니다.</p>

3. 글 관련 주요 태그

인용구

- <blockquote>
- 일반적인 텍스트보다 안쪽으로 들여 써짐

[cite 속성의 경우]

- 인용문의 출처가 있으면 항상 추가하는 것이 좋은 습관이다.
- 주요 브라우저에서 특별한 것으로 렌더링되지 않지만 검색 엔진에서 인용에 대한 추가 정보를 얻기 위해 사용될 수 있다.

```
<blockquote cite="https://naver.com">  
이제부터는 간헐적으로 문서 기본 구조를 생략하고  
</blockquote>
```

이제부터는 간헐적으로 문서 기본 구조를 생략하고

4. 글 관련 주요 태그

수평선

- <hr>
- 수평선을 표시할 때 사용하는 태그

```
<p> 작년 12월은 추운 겨울을 보냈습니다.</p>  
<hr>  
<p>2024년은 맑은 날이 이어질 것으로 보입니다.</p>
```

5. 글 관련 주요 태그

줄내림

- `
`

```
<p> 오늘은 즐거운 일이 펼쳐질꺼야.  
벌써 오후잖아? </p>
```

```
<p> 오늘은 즐거운 일이 펼쳐질꺼야. <br> 벌써 오후  
잖아? </p>
```

6. 글 관련 주요 태그

그대로 보여주기

- `<pre>`

```
<p>  
오늘은 즐거운 일이 펼쳐질꺼야.  
    벌써 오후잖아?  
</p>
```

```
<pre>  
오늘은 즐거운 일이 펼쳐질꺼야.  
    벌써 오후잖아?  
</pre>
```

7. 글 관련 주요 태그

강조

- `<i>` : 텍스트를 *이탤릭체로 표시*할 때 사용 (Italic)
- `` : 텍스트를 *이탤릭체로 강조*할 때 사용 (Emphasis)
- `` : 텍스트를 **진하게** 표시할 때 사용 (Bold)
- `` : 텍스트를 **진하게 강조**할 때 사용

```
<i> 웹은</i>  
<em> 정말</em>  
<b> 재미있고</b>  
<strong> 신난다</strong>
```

Q. `` 와 ``, `<i>` 와 `` 차이는 뭘까?

8. display 요소

Display 요소는 두개의 특징을 가짐

- Block-level Element
 - 왼쪽에서 오른쪽으로 하나의 가로 영역(Block) 전체를 차지하는 태그들
- Inline-level Element
 - 자기의 고유 영역이 없이 그저 자신이 둘러싸고 있는 내용들의 양에 따라 길이가 결정되어 지는 태그들

9. 엔티티 코드

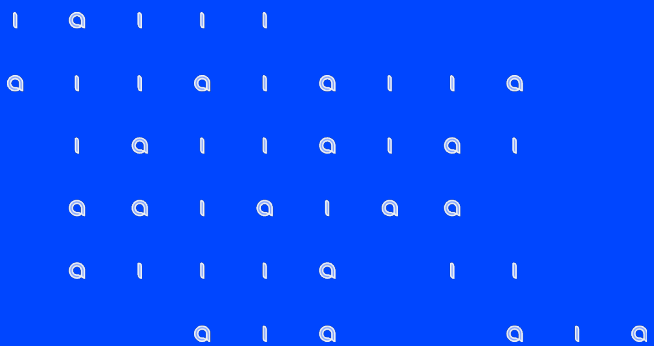
HTML 예약어(<> 등) 를 HTML 코드에서 사용하면, 웹 브라우저는 그것을 평소와는 다른 의미로 해석한다. HTML 예약어를 기존에 사용하던 의미 그대로 사용하기 위해 별도로 만든 문자셋을 **엔티티(entity)**라고 한다.

Entity Name	Entity Code	Description	Example
<	<	Less than	<
>	>	Greater than	>
&	&	Ampersand	&
"	"	Double quotation mark	"
'	'	Single quotation mark	'
	 	Non-breaking space	
©	©	Copyright symbol	©
®	®	Registered trademark	®

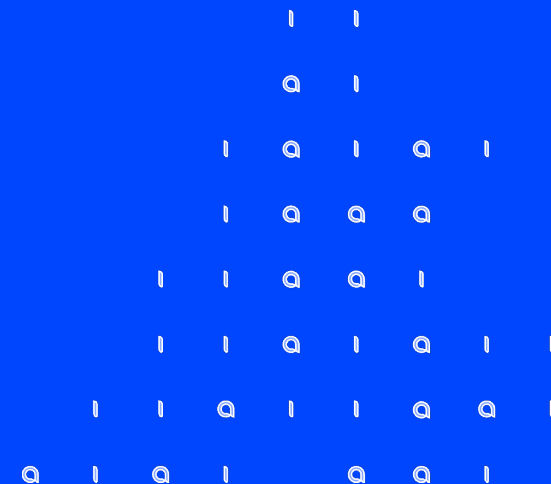
실습

- 글 관련 HTML 문법들을 직접 타이핑해보고, 동작을 확인합니다.
- 현재 레벨이 너무 쉬우시면 아래 페이지를 활용해 실습해주세요~!
 - https://www.w3schools.com/tags/tag_comment.asp
 - https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web





6. HTML 문법 – 리스트, 테이블, 동작



학습목표

- ▶ 리스트 관련 HTML 문법을 이해하고 활용할 수 있다.
- ▶ 테이블 관련 HTML 문법을 이해하고 활용할 수 있다.
- ▶ 동작 관련 HTML 문법을 이해하고 활용할 수 있다.

1. 동작 관련 태그

<a>: 앵커 태그로, 하이퍼링크를 생성. href 속성을 통해 연결될 주소를 지정한다.

```
<a href="https://www.example.com">방문하기</a>
```

<button>: 클릭 가능한 버튼. 주로 JavaScript와 함께 사용되어 특정 동작을 수행한다.

```
<button onclick="alert('Hello!')">클릭하세요</button>
```

1. 동작 관련 태그

`<input>`: 사용자 입력을 받는 입력 필드. `type` 속성을 통해 다양한 입력 필드를 만들 수 있다.

```
<input type="text"/>
```

`type` 속성으로 입력 필드의 종류를 지정할 수 있다

1. `text`: 일반 텍스트
2. `password`: 비밀번호
3. `number`: 숫자
4. `email`: 이메일 주소
5. `checkbox`: 체크박스
6. `radio`: 라디오 버튼
7. `file`: 파일을 업로드하는 필드

1. 동작 관련 태그

<input>: 사용자 입력을 받는 입력 필드. type 속성을 통해 다양한 입력 필드를 만들 수 있다.

```
<input type="text" name="username " maxlength=20 placeholder="이름"/>
```

- 1.value: 입력 필드의 값을 설정
- 2.placeholder: 입력 필드에 입력할 값의 예시
- 3.name: 입력 필드의 이름을 정의. 서버로 전송될 때 해당 이름으로 데이터가 전송됨
- 4.disabled: 입력 필드를 비활성화함. 사용자가 값을 입력할 수 없음
- 5.readonly: 읽기 전용으로 설정. 사용자는 값을 읽을 수는 있지만 변경할 수 없음
- 6.required: 필수 입력 필드로 설정합니다. 제출하기 전에 반드시 값을 입력해야 함
- 7.maxlength: 입력 필드에 입력 가능한 최대 문자 수를 지정
- 8.min와 max: 숫자 입력 필드의 최소 및 최대값을 지정
- 9.pattern: 입력 필드에 입력할 수 있는 정규 표현식 패턴을 지정

1. 동작 관련 태그

<label>: 입력 요소에 대한 설명을 제공하거나, 사용자가 클릭 가능한 레이블을 생성한다.

```
<label for="chk">동의합니다</label>  
<input type="checkbox" id="chk" name="chk">
```

<textarea>: 여러 줄의 텍스트를 입력할 수 있는 입력 필드를 생성한다.

```
<textarea rows="4" cols="50" placeholder="메시지를 입력하세요"></textarea>
```

1. 동작 관련 태그

`<form>`: 데이터를 서버로 제출하는 양식을 생성함. 주로 버튼이나 입력 필드들을 담는다.

```
<form action="/submit" method="post">
  <input type="text" name="username" placeholder="사용자명">
  <input type="password" name="password" placeholder="비밀번호">
  <button type="submit">제출</button>
</form>
```

action 속성: 입력된 데이터가 전송될 서버 쪽 스크립트 파일의 URL을 지정한다.

method 속성: 입력된 데이터를 서버로 전송하는 방식을 결정한다.

- 주로 GET과 POST를 사용하는데
- GET은 URL에 데이터를 추가해서 전송하고
- POST는 숨겨진 방식으로 데이터를 전송한다

1. 동작 관련 태그

<select>: 드롭다운 목록을 생성. 사용자가 옵션을 선택할 수 있다.

```
<select>  
  <option value="1">항목 1</option>  
  <option value="2">항목 2</option>  
  <option value="3">항목 3</option>  
</select>
```

1. 동작 관련 태그

<fieldset>: 여러 관련된 입력 요소들을 그룹화한다. 주로 <legend>와 함께 사용된다.

```
<fieldset>
  <legend>개인 정보</legend>
  <label for="name">이름:</label>
  <input type="text" id="name" name="name"><br><br>
  <label for="email">이메일:</label>
  <input type="email" id="email" name="email"><br><br>
</fieldset>
```


1. 동작 관련 태그

`<iframe>`: 다른 HTML 페이지를 현재 페이지에 삽입하는데 사용한다.

```
<iframe src="https://www.example.com" width="600" height="400"></iframe>
```

2. 미디어 관련 태그들

: 이미지를 삽입하는 태그로, src 속성을 통해 이미지 파일의 경로를 지정

```

```

<audio>: 오디오를 재생하는 태그로, 소리 파일을 삽입하고 컨트롤

```
<audio controls>  
  <source src="오디오_파일.mp3" type="audio/mpeg">  
  Your browser does not support the audio element.  
</audio>
```

2. 미디어 관련 태그들

<audio>: 오디오를 재생하는 태그로, 소리 파일을 삽입하고 컨트롤

```
<audio controls>
  <source src="오디오_파일.mp3" type="audio/mpeg">
  Your browser does not support the audio element.
</audio>
```

<video>: 비디오를 재생하는 태그로, 동영상 파일을 삽입하고 컨트롤

<track>: 비디오나 오디오의 자막, 자막 파일을 지정

```
<video controls width="500">
  <source src="비디오_파일.mp4" type="video/mp4">
  <track src="자막_파일.vtt" kind="subtitles" srclang="en" label="English">
  Your browser does not support the video element.
</video>
```

2. 미디어 관련 태그들

<picture>: 반응형 이미지를 제공하기 위해 여러 이미지 리소스를 포함하는 컨테이너

```
<picture>
  <source media="(min-width: 768px)" srcset="큰_이미지.jpg">
  <source media="(min-width: 320px)" srcset="작은_이미지.jpg">
  
</picture>
```

3. 목록 태그

: 순서가 없는 리스트. 각 아이тем은 점이나 기호로 표시된다.

: 순서가 있는 리스트. 각 아이тем은 숫자로 표시된다.

: 리스트 아이тем을 가리킨다. 이나 안에 들어가는 태그이다.

<dl>: 설명 리스트를 만들 때 사용된다. 주로 용어나 정의들을 표시할 때 쓰인다.

<dt>: 정의 리스트에서 용어를 정의할 때 사용된다.

- 보통 <dl> 안에서 <dd>와 함께 사용된다.

<dd>: 정의 리스트에서 정의에 해당하는 내용을 표시할 때 사용된다.

- <dl> 안에서 <dt>와 함께 사용된다.

4. 테이블 태그

<table>: 테이블을 정의하는 태그

<tr>: 테이블의 행을 나타내는 태그

<th>: 테이블의 헤더 셀을 정의하는 태그

<td>: 테이블의 일반 셀을 정의하는 태그

<thead>: 테이블의 헤더를 그룹화하는 태그

<tbody>: 테이블의 본문을 그룹화하는 태그

<tfoot>: 테이블의 푸터를 그룹화하는 태그

<caption>: 테이블의 캡션(제목)을 정의하는 태그

The diagram illustrates a 3x4 table structure. It features a table with the following content:

이름	국어	수학	코딩
홍길동	80	100	100
짐코딩	100	80	100

Labels and arrows pointing to the table components:

- 열(col)**: Points to the column headers.
- 셀(ce)**: Points to a specific cell (80).
- 표(table)**: Points to the entire table structure.
- 행(row)**: Points to the first data row.

3행 4열의 테이블

4. 테이블 태그

`<colgroup>`: 열을 그룹화하는 태그

`<col>`: 각 열을 정의하는 태그로, `<colgroup>` 안에서 사용된다.

`<tbody>`: 테이블의 본문을 그룹화하는 태그

`<thead>`: 테이블의 헤더를 그룹화하는 태그

`<tfoot>`: 테이블의 푸터를 그룹화하는 태그

5. 리소스 경로

[참고] 절대경로 VS 상대경로

절대경로: 리소스(이미지)의 절대경로는 말 그대로 절대적인 고유한 경로를 지정한다.

- 웹 이미지 절대경로 (<http://www.naver.com/apple.png>)
 - http 프로토콜로 시작해서 전체 경로를 입력함
- 웹 이미지 절대경로 (2) - /apple.png 루트
 - ('/') 디렉토리 부터 시작하는 경우 현재 도메인이 자동으로 앞에 붙음
- PC 컴퓨터 절대경로
 - 예) - C:\user\gymcoding\apple.png

절대 경로를 이용하면 웹에서 이미지가 사라지거나,
내 컴퓨터에서 만든 파일을 다른 곳으로 옮길 때 해당 절대경로를 다시 수정해야 하는 불편함이 있다.

작업 중인 폴더에 이미지를 옮기고 상대적인 위치를 가리킴으로써 이러한 불편함을 해소할 수 있다.

5. 리소스 경로

[참고] 절대경로 VS 상대경로

상대경로: 상대경로는 현재 문서를 기준으로 경로를 인식하는 방법이다.

- index.html 에서 동일한 위치에 있는 apple.png를 가져오는 방법
 - src="apple.png" 또는 src="./apple.png"
- index.html 의 상위 폴더에 이미지가 있는 경우
 - src="../apple.png"
- index.html 의 하위 폴더에 이미지가 있는 경우
 - src="하위폴더/apple.png"

6. 그래픽 태그

`<canvas>`: JavaScript를 사용하여 그래픽을 그리고 이미지를 생성하는 데 사용한다.

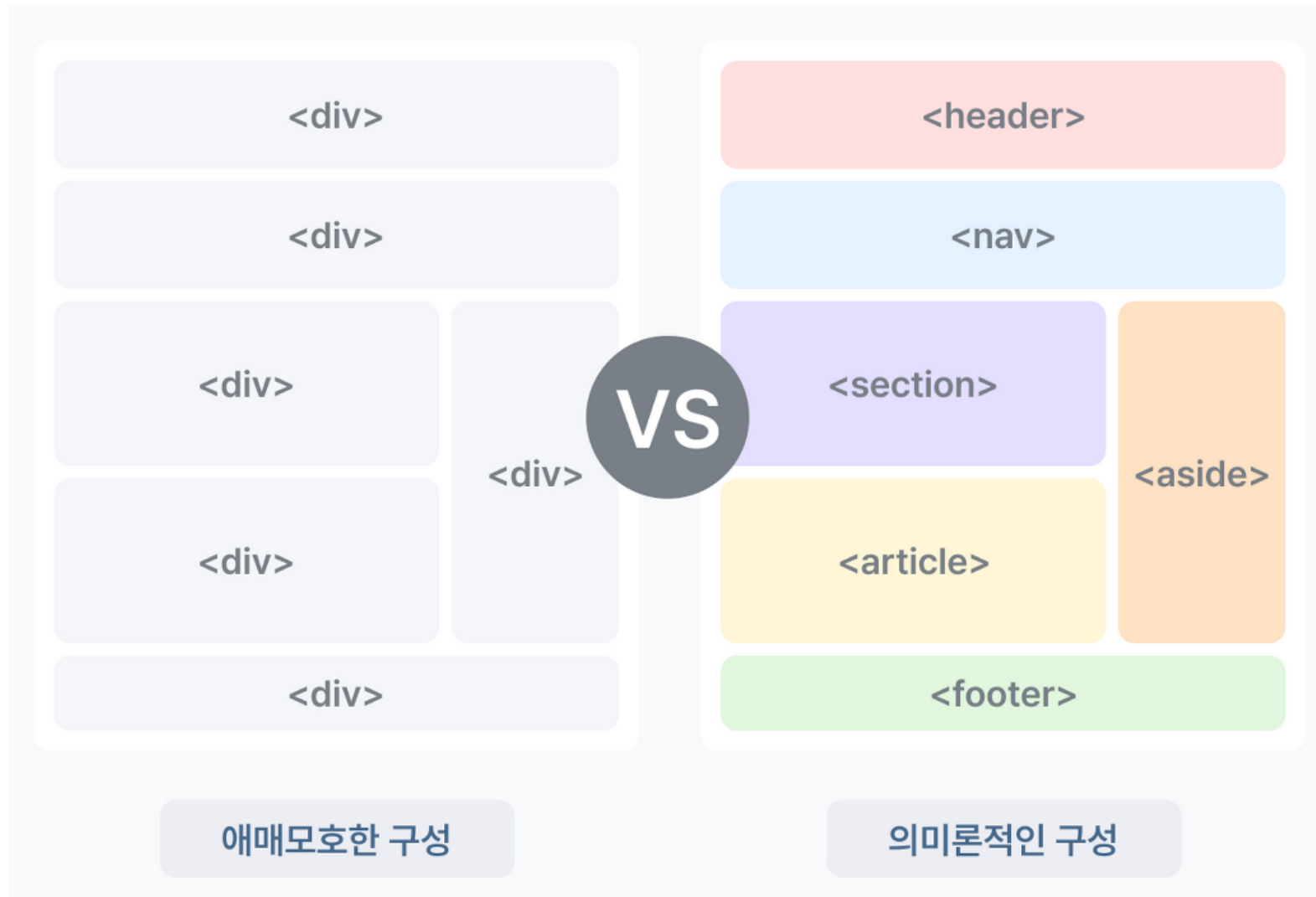
```
<canvas id="myCanvas" width="200" height="100" style="border:1px solid #000;"></canvas>
```

```
<script>
```

```
  const canvas = document.getElementById('myCanvas');  
  const ctx = canvas.getContext('2d');  
  ctx.fillStyle = 'rgb(200, 0, 0)'; ctx.fillRect(10, 10, 150, 80);
```

```
</script>
```

7. 시멘틱 태그



7. 시멘틱 태그

<header>: 웹 페이지나 섹션의 헤더를 정의

<footer>: 웹 페이지나 섹션의 푸터를 정의

<nav>: 네비게이션 링크를 담는 태그

<article>: 독립적인 콘텐츠 영역을 정의

<section>: 문서의 섹션을 정의하며 주제별로 그룹화함

<aside>: 본문 콘텐츠와는 관련성이 떨어지는 사이드바를 정의

<main>: 문서의 주요 콘텐츠를 감싸는 태그로, 페이지에 단 하나만 사용

<figure>: 이미지, 차트, 캡션 등의 그룹을 정의

<figcaption>: <figure> 요소의 캡션을 정의

<time>: 날짜, 시간 등을 나타냄

7. 시멘틱 태그

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>시멘틱 태그 예제</title>
  </head>
  <body>
    <header>
      <h1>웹 페이지 제목</h1>
      <nav>
        <ul>
          <li>메뉴 1</li>
          <li>메뉴 2</li>
        </ul>
      </nav>
    </header>

    <main>
      <section>
        <h2>섹션 제목</h2>
        <p>섹션의 내용을 여기에</p>
      </section>

      <section>
        <h2>또 다른 섹션</h2>
        <article>
          <h3>글 제목</h3>
          <p>글 내용을 작성합니다.</p>
        </article>

        <aside>
          <h4>사이드바</h4>
          <p>부가적인 정보를 여기에</p>
        </aside>

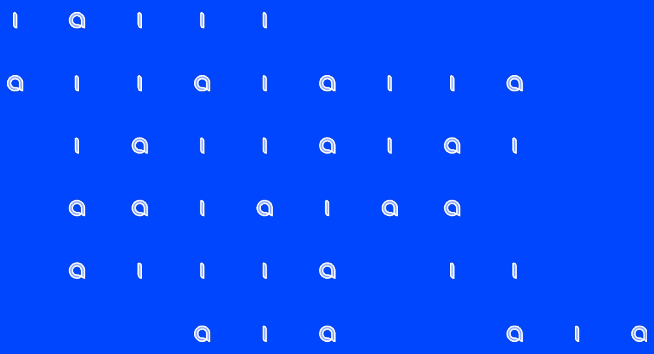
      </section>
    </main>

    <footer>
      <p>&copy; 2023 웹 페이지</p>
    </footer>
  </body>
</html>
```

실습

- 요소들을 직접 타이핑 하고 생김새와 동작을 확인해보세요.
- 주어진 디자인 시안을 확인하고, 코드로 구현해보세요.
- 시맨틱 태그를 활용하여 작성해보세요





THANK YOU.

