

2일 (금)

1. Codecademy HTML 정리

1) table 만드는 법

`<table>` : 하나의 테이블을 생성

—*semantic table*—

`<thead>` : 테이블의 각 열의 헤딩에 해당함

`<tbody>` : 테이블 내용이 있는 공간

`<tfoot>` : 결과값이나 총합을 표현하는 공간

—*contents*—

`<tr>` : 하나의 로우를 생성함

`<td>` : 하나의 셀에 담을 데이터를 생성

```

<body>
  <table>
    <thead>
      <tr>
        <th></th>
        <th>heading 1</th>
        <th>heading 2</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td colspan="1">mingyu</td>
        <td>row 1</td>
        <td>row 1</td>
      </tr>
      <tr>
        <td colspan="1">haru</td>
        <td>row 2</td>
        <td>row 1</td>
      </tr>
      <tr>
        <td colspan="1">sora</td>
        <td>row 3</td>
        <td>row 1</td>
      </tr>
    </tbody>
    <tfoot>
      <tr>
        <td></td>
        <td>summary</td>
        <td>summary</td>
      </tr>
    </tfoot>
  </table>

```

heading 1 heading 2

mingyu row 1 row 1

haru row 2 row 1

sora row 3 row 1

summary summary

2) Semantic HTML

세맨틱 html 은 각 요소가 어떤 역할을 하는지 개발자 및 브라우저에게 정확히 알려주는 역할을 한다. 보통 <div> 로 구성된 html 은 각 div 가 어떤 역할을 하는지 정확히 아는 데 파악이 힘들다. 하지만 <nav> , <main> , <header> , <footer> 등을 사용하면 각 요소가 어떤 역할을 하는지 정확히 파악이 가능하다.

- Embedding media → <video> , <audio> , <embed> embed 는 어떤 타입의 미디어든 사용이 가능하다. 클로스테그가 없다.

```
<!--Video Tag-->
<video src="4kvideo.mp4">video not supported</video>

<!--Audio Tag-->
<audio src="koreanhiphop.mp3"></audio>

<!--Embed tag-->
<embed src="babyyoda.gif"/>
```

- <figure> and <figcaption>

figure 태그는 이미지, 다이어그램, 작은 코드인 미디어를 캡슐화하기 위해 사용된다. figcaption 은 보통 figure태그 내부에 쓰인다.

```
<figure>
  
  <figcaption>The image shows the layout of a qwerty keyboard.</figcaption>
</figure>
```

- <section> and <article>

section 태그는 챕터나 헤딩, 같은 테마를 가진 문서의 영역을 구분한다.

개발자들은 일반적으로 섹션태그를 웹페이지의 테마를 위해 사용하고, article 태그는 그 테마의 독립적인 콘텐츠를 쓸 때 사용한다.

```
<section>
  <!--defines theme-->
  <h2>Top Sports league in America</h2>
  <!--writes independent content relating to that theme-->
  <article>
    <p>One of the top sports league is the nba.</p>
  </article>
</section>
```

- <aside>

추가적인 정보를 제공하기 위한 사이드바의 개념으로 쓰이는 태그이다.

```
<article>
  <!--Main Content-->
</article>
<aside>
  <!--Additional information-->
</aside>
```

2. Codecademy JavaScript - Scope

스코프 비유 : 내가 서울에 산다면 서울에 있는 한강과 남산은 지역적 즉 지역변수이다. 서울 하늘에 떠있는 별은 전역변수이다. 별은 파리에서도 볼 수 있고 뉴욕에서도 볼 수 있다. 하지만 파리나 뉴욕에서 서울의 한강이나 남산을 볼 수 없다. 이것이 스코프다.

- 블록과 스코프

블록은 { } 중괄호를 포함한 내용을 말한다. 함수나 if 문, 반복문에서 쓰인다.

- 전역 변수 (Global Scope)

블록 밖에 선언된 변수이다. 블록 내외로 모두 호출이 가능하다.

익명함수도 전역변수이다.

```
let satellite = 'The Moon';
let galaxy = 'The Milky Way';
let stars = 'North Star';

function callMyNightSky() {
  return `Night Sky: ${satellite}, ${stars}, and ${galaxy}`
}

console.log(callMyNightSky())
```

- 지역 변수 (블록 스코프)

지역 변수는 블록 안에서만 사용이 가능하다.

익명함수에 들어가는 파라미터도 지역 변수에 해당한다.

- Scope Pollution

변수가 항상 전역변수라면 좋다고 생각할 수도 있지만, 너무 많은 전역변수는 프로그래밍에서 문제를 일으킬 수 있다.

전역 변수는 global namespace 에 저장되는데 스코프 폴루션은 너무 많은 전역 변수가 글로벌 네임스페이스에 들어갈 때 일어나게 된다.

전역변수가 지역변수에 의해 변경되어 다른 곳에서 해당 전역변수가 쓰일 때 예상치 못한 값을 출력할 수 있다. 이러한 점을 유의하여 변수명을 잘 설정하거나 최대한 지역변수 내에서 쓸 수 있도록 하자.

- 좋은 스코핑 연습

좋은 스코핑 연습은 가독성이 높아지며, 이해와 명확도가 증가되며, 유지보수가 쉽고, 메모리를 아낄 수 있을 것이다.

전역변수가 필요한지 생각한 후 정 필요 없다면 지역변수에 넣어버리자.

```
const logSkyColor = () => {
  const dusk = true;
  let color = 'blue';
  if (dusk) {
    let color = 'pink';
    console.log(color); // pink
  }
  console.log(color); // blue
};

console.log(color); // ReferenceError
```

3. 노마드 코더 바닐라 JS로 그림판 만들기

HTML

1. <canvas> 는 html 태그이다.
2. 'input: ' 을 입력하면 input 태그가 자동완성된다.
3. js는 id 로 , css 는 class 로 설정하게 되면 덜 헛갈린가?
4. 가끔 css 가 더러워질 만한 상황에서는 html 에 직접 style = " " 을 넣어 css 를 꾸며줘도 된다.
5. 클래스를 설정할 때 부모 노드 클래스는 복수형으로 분류하면 좋다.
6. 초기에 css, js 를 모두 연결해놓고 프로젝트를 시작하자.

CSS

1. reset.css 사용하자.
2. import 방법
3. padding 프로퍼티 입력순서 상하좌우 순, (상하)(좌우) 순 margin 도 마찬가지
4. css 를 작성할 때는 무작위로 작성하지 말고 html 부모노드 순으로 순서대로 작성한다.
5. 버튼 투명화는 all : unset ;
6. border 순서 (굵기, 선종류, 색상)
7. text-transform 프로퍼티 : 대문자, 소문자 등 글자를 바꿀 수 있다.
8. active 효과 (버튼에 많이 쓰임) transform: scale(변경 크기); → 버튼 누를 때 효과를 준다.
9. 원형 만들기 → width 와 height 을 정사각형 크기로 만든 후 border-radius 를 각 Px 의 절반으로 하면 원형 만들어짐 or 50%
10. cursor 프로퍼티는 마우스를 올리면 나오는 이미지를 설정한다.

4. github 커밋 잔디밭 적용이 안될 때 문제 해결

깃허브 아이디 상 이메일과 커밋 이메일이 다르면 적용이 안된다.

깃허브 상에 이메일을 추가하여 문제 해결 !!

setting → email setting → email add