

10월 27일 (수)

1. 엘리스 SW 엔지니어 트랙 - HTML/CSS -1-

2. HTML 주요 태그 살펴보기

- img 태그

닫힌 태그가 없으며, alt 속성을 꼭 넣어줘야한다.(이미지 출력 x 시 텍스트 정보로 대체)

src 속성에는 삽입할 이미지 파일 경로

- h태그

제목이나 부제목을 표현 , h1 태그는 가장 중요한 정보를 담으므로, 하나의 html 문서에서 한번만 사용된다.

- p 태그 (paragraph 약자)

본문 내용을 표현

- ul 태그

메뉴 버튼 만들 때 사용

- a 태그

링크태그로 href 속성, target 속성이 대표적이다. target="_blank" 를 값을 주면 새로운 페이지가 열리게 된다.

3. 공간을 만들 때 사용하는 태그

- html 태그 구성 요소

책 : 목차, 본문, 부록 == 웹사이트 : header(nav), main, footer

- 목차부분 <header>, <nav> 태그

헤더태그는 웹사이트의 머리글을 담는 공간

내브태그는 메뉴 버튼을 담는 공간 (ul, li, a 와 함께 사용)

Example

```
<header>  <!-- 상단 영역 -->
  
  <nav>    <!-- 메뉴 버튼 영역 -->
    <ul>
      <li>홈</li>
      <li>전체 목록</li>
    </ul>
  </nav>
</header>
```

```
/* elice */   홈   전체 목록
```

- 본문부분 <main>, <article> 태그

Example

```
<main role="main"> <!-- 본문 영역 -->
  <article> <!-- 정보 영역 -->
    ...
  </article>
</main>
```

문서의 **주요 내용**을 담는 태그

IE(Internet Explorer)는 지원하지 않으므로 **role="main"** 속성 필수 입력

/*

main 태그는 익스플로러에서 지원 x **role="main"** 을 입력해줘야한다.

article 태그 안에는 태그 내 구역을 대표하는 **<h#>** 태그가 꼭 존재 해야 한다.

- 부록부분 <footer>태그

Example

```
<footer> <!-- 하단 영역 -->
  <div> <!-- 엘리스 정보 -->
    <p>주소: 대전광역시 유성구 문지로 193 KAIST</p>
    <p>이메일: contact@eclice.io</p>
  </div>
  <div> <!-- 전자상거래소비자보호법 필수 정보 -->
    <p>사업자등록번호: 000-00-00000 | 대표: 엘토끼</p>
    <p>통신판매업신고번호: 제0000-토끼굴-0000호</p>
  </div>
</footer>
```

가장 하단에 들어가는 정보를 표기할 때 사용

/*

div 태그는 깡통태그로 공간설정을 할 때 자주 사용한다.

- nav 를 사용할 때 안에 ul 없이 li 를 바로 써도된다.
- 로고는 html을 대표하므로 헤더태그 안에 h1 안에 감싸서 넣어주었다.
- li 태그 안에 a 태그를 중첩시켜서 넣어주어야 한다.
- main 태그를 사용할 땐 role속성에 role="main" 값을 입력해줘야 한다.

4. HTML의 두 가지 성격

- Block 요소와 Inline 요소



Example

```
<!-- Block 요소 -->
<p>Hello Elice</p>
<p>Hello Elice</p>
<p>Hello Elice</p>
```

Example

```
<!-- Inline 요소 -->
<a>Bye Elice</a>
<a>Bye Elice</a>
<a>Bye Elice</a>
```

. 두 요소를 구분 짓는 세 가지 주요 특징 :

줄바꿈 현상, 가로·세로, 상·하 배치



/*

❶ Block 요소

Example

```
<!-- Block 요소 -->
<p>Hello Elice</p>
<p>Hello Elice</p>
<p>Hello Elice</p>
```



```
Hello Elice  
Hello Elice  
Hello Elice
```



y축 정렬 형태로 출력 (줄바꿈 현상 나타남)

공간을 만들 수 있고, 상하 배치 작업 가능

❷ Inline 요소

Example

```
<!-- Inline 요소 -->
<a>Bye Elice</a>
<a>Bye Elice</a>
<a>Bye Elice</a>
```



```
Bye Elice Bye Elice Bye Elice
```

,

x축 정렬 형태로 출력 (한 줄에 출력)

공간을 만들 수 없고, 상하 배치 작업 불가능

/* elic

5. CSS 주요 속성 살펴보기

- CSS란?

정보와 디자인을 분리할 수 있도록 함

속성값에 세미콜론 필수 입력

- CSS 연동 방법 세 가지

Inline Style Sheet , Internal Style Sheet, External Style Sheet

각 태그에 직접 , html 파일 안에 style 태그 사용, <link> 태그로 불러오기(닫힌태그 없음)

- CSS 선택자

Type(태그), Class, ID 로 이루어져 있다.

- 부모 자식 관계 1

✓ 부모 자식 관계 1

Example

```
<header>
  <h1>Header h1</h1>
  <p>Header p</p>
</header>
```

```
// style.css 문서
header { color: red; }
h1 { color: blue; }
p { color: green; }
```

<header>과 <h1><p> : 부모 자식 관계

<h1>과 <p> : 형제 관계

/* elic

부모는 자식에게 상속을 해줌 즉, header 태그에 red 속성값을 주면 h1, p 태그 또한 red로 바뀐다.

- 부모 자식 관계 2

❸ 부모 자식 관계 2

Example

```
<header>
  <h1>Header h1</h1>
  <p>Header p</p>
</header>
<footer>
  <h1>Footer h1</h1>
  <p>Footer p</p>
</footer>
```

// style.css 문서

```
header { color: red; }
header h1 { color: blue; }
header p { color: green; }
```

원하는 지역에만 css 속성을 적용하기 위해 부모를 **구체적으로** 표기

/* elic

위와 같이 구체적인 명시 가능하다.

- CSS 캐스케이딩

의미: CSS의 우선순위를 결정하는 세 가지 요소 ⇒ 순서, 디테일, 선택자

❹ 순서에 의한 캐스케이딩

Example

```
<p>Hello World</p>
```

// style.css 문서

```
p { color: red; }
p { color: blue; }
```

나중에 적용한 속성값의 우선순위가 높음

/* elic

✓ 디테일에 의한 캐스케이딩

Example

```
<header> // style.css 문서  
    <p>Hello World</p>  
</header>    header p { color: red; }  
                  p { color: blue; }
```

더 구체적으로 작성된 선택자의 우선순위가 높음

✓ 선택자에 의한 캐스케이딩

Example

```
<h3 style="color: pink" id="color" class="color"> color </h3>

// style.css 문서
#color { color: blue; }
.color { color: red; }
h3 { color: green; }
```

style > id > class > type 순으로 우선순위가 높음

- CSS 주요 속성

❷ font

Example

```
<p class="paragraph"> 즐거운 웹프로그래밍!</p>

.paragraph {
    font-size: 50px; /* 글자 크기 */
    font-family: Arial, sans-serif; /* 글꼴 */
    font-style: italic; /* 글자 기울기 */
    font-weight: bold; /* 글자 두께 */
}
```

.font-family

브라우저마다 지원하는 폰트가 다름.

입력한 순서대로 우선순위 적용.

sans-serif는 마지막에 작성하는 디폴트 값.

.font-weight

100~900 사이의 숫자를 입력할 수도 있음.

font-weight 는 100단위로 입력한다.

❸ border

Example

```
<p class="paragraph"> 즐거운 웹프로그래밍!</p>

.paragraph {
    width: 500px;
    height: 500px;
    border-style: solid;
    border-width: 10px;
    border-color: red;
    /* border: solid 10px red; */
}
```

.border-style

실선 : solid

점선 : dotted

주석과 같이 한 줄에 이어 쓸 수도 있음.

이때, 쉼표는 작성하지 않고 띄어쓰기만 함.

❹ background



Example

```
<p class="paragraph"> 즐거운 웹프로그래밍!</p>

.paragraph {
    background-color: yellow;
    background-image: url(이미지 경로);
    background-repeat: no-repeat;
    background-position: left;
    /* background: yellow url(이미지 경로)
       no-repeat left; */
}
```

.background-repeat

x축으로 반복 : repeat-x

y축으로 반복 : repeat-y

반복하지 않음 : no-repeat

.background-position

공간 안에서 이미지의 좌표를 변경할 때.

top, bottom, center, left, right 등.

2. 프로그래머스 자바스크립트 12기 스터디 3주차

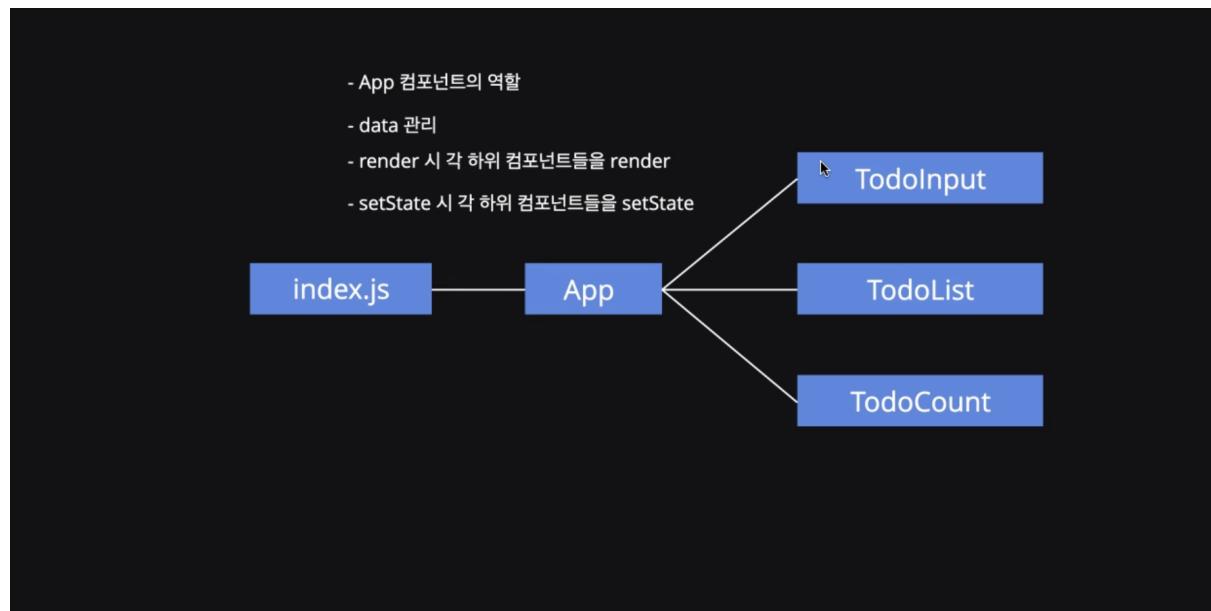
최소한의 책임

⇒ 각 컴포넌트는 자신이 담당하는 화면을 그리기 위해 가장 최소한의 책임과 의존성을 가져야 한다.

즉, TodoCount를 그리기 위해 TodoList에 접근해와서는 안된다는 말

만약 위의 두 개를 둑어버리면, TodoList의 변화가 TodoCount에 영향을 주는 구조가 되어 복잡도가 올라가고 예기치 못한 버그를 만나게 된다.

⇒ 자바스크립트는 함수의 파라미터로 함수를 만들어서 넣을 수 있다. 즉, 이를 이용해서 각 컴포넌트가 최소한의 의존성을 갖도록 해야한다.



TodoInput의 입장이 되어보기

- TodoInput의 역할에 대해 생각해봅시다.

TodoInput의 입장이 되어보기

- TodoInput의 역할에 대해 생각해봅시다.
 - 이름에서 유추해보면, Todo를 입력 받는 컴포넌트입니다.
 - 입력 받아서 이걸 어디에 추가한다거나 어디에 보낸다거나 하는 것을 이 친구가 알아야 할까요?
 - 입력된 Todo가 어디에 추가되고 어떻게 가공되고 하는지는 이 컴포넌트를 쓰는 쪽에서 알면 되지 않을까요?

↳

html에 script 태그의 모듈 순서가 중요하다.