

10월 16일 (토)

1. 프로그래머스 자바스크립트 스터디 12기 - 1주차 Mission

- 필수 구현사항

```
<html>

<head>
  <title>Mission 1</title>
  <meta charset="utf-8" />
</head>

<body>
  <div id="todo-list"></div>
  <script>
    var data = [
      {
        text: 'JS 공부하기'
      },
      {
        text: 'JS 복습하기'
      }
    ]
    // 이곳에서 코딩을 시작하세요!
  </script>
</body>
</html>
```

필수 구현사항 설명

- function style의 문법, class style의 문법 어느 것을 사용해도 좋습니다.
- new 키워드를 통해 생성해서 사용됩니다.
- 파라미터로 아래와 같은 형태의 data를 넘겨받습니다.

```
var data = [  
  {  
    text: 'JS 공부하기'  
  },  
  {  
    text: 'JS 복습하기'  
  }  
]
```

`var todoList = new TodoList(data);` 와 같은 형태로 파라미터를 넘기고, 생성해서 사용합니다.

- 해당 컴포넌트에 `render` 함수를 작성합니다.
 - 이 함수는 파라미터로 넘겨받은 data를 순회하며 각 배열첨자의 text를 html string으로 만든 뒤, `todo-list` 라는 id를 가진 div에 `innerHTML` 을 이용해 렌더링 되도록 합니다.
 - 이 함수는 별도의 파라미터 없이 `todoList.render()` 형태로 실행되도록 만듭니다.

Tip

- new 키워드와 this 키워드가 생소하신 분을 위해 간단한 샘플을 준비했습니다.

- ```
function Band(name) {
 // 파라미터로 받은 name을 this.name으로 할당
 this.name = name;
 // Band 함수에 perform 함수를 만들고 그것을 this.perform 에 할당
 this.perform = function(){
 console.log(`${this.name} 밴드가 공연을 합니다!`);
 }
 this.performHTML = function() {
 document.querySelector('#stage').innerHTML = `<div>${this.name}</div>`;
 }
}

// new 키워드를 이용해 Band 여러개를 만들기
// new를 붙이고 함수가 실행될 때마다, 함수 내 this는 새 함수 인스턴스를 가리킨다.
var band1 = new Band('Cold Play'); // band1.name은 무엇일까요?

// 나머지 band2, band3, band4에 대해서도 생각해봅시다.
var band2 = new Band('Foo Fighters');
var band3 = new Band('Crying Nut');
var band4 = new Band('Jaurim');

// 무슨 일이 일어날까요?
band1.performHTML();
// 무슨 일이 일어날까요?
band2.performHTML();
```

- data와 html string을 결합하기 위해 + 를 연산자를 이용하는 방법

- ```
var html = '<div>' + data[0].text + '</div>';
```

- string template을 이용하는 방법

- ```
// 문자열을 '이 아닌 ` (탭 위에 있는 키)로 묶는 것에 유의하세요!
var html = `<div>${data[0].text}</div>`
```

- 특정한 div 내에 html을 넣으려면 아래의 문법을 활용하세요.

- ```
document.querySelector('#todo-list').innerHTML = 'html string';
```

읽으면 도움이 될 글들

- [JavaScript에서 사용자정의 생성자함수](#) - 생성자 함수에 대해 정리가 된 글입니다.
- [Template literals](#)
 - ES6의 Template Literals 문법에 대한 글입니다. 이번 미션을 수행하면서 html 을 string 형태로 생성하게 되는데, 적절히 활용하면 data와 html string을 붙이는 작업을 편하게 할 수 있습니다.
- [this는 어렵지 않습니다.](#) - this에 대해 정리된 글입니다.

```
<html>
  <head>
    <title>Mission 1</title>
    <meta charset="utf-8" />
  </head>

  <body>
    <div id="todo-list"></div>
    <div id="todo-list-2"></div>
    <div id="todo-list-3"></div>
    <script>
      const data = [
        {
          text: 'JS 공부하기',
          isCompleted: true,
        },
        {
          text: 'JS 복습하기',
          isCompleted: false,
        },
        {
          text: '아침 먹기',
          isCompleted: false
        }
      ];

      // 이곳에서 코딩을 시작하세요!

      // TodoList 생성자 함수
      function TodoList(data) {

        this.data = data;
```

```
this.render = function() {  
  
    const contents = document.getElementById('todo-list-1');  
  
    this.data.forEach(function(item) {  
  
        const toDoThing = document.createElement('div'); // 새로운 div태그 생성  
        toDoThing.innerHTML = item.text; // 생성한 div태그에 text 넣어주기  
        contents.appendChild(toDoThing); // HTML내 연결 시켜 렌더링  
  
    })  
}  
  
var todoList = new TodoList(data);  
todoList.render();  
  
</script>  
</body>  
</html>
```