[ITEC] JS Study

> 4 week

> strings, array, event listener, time handler

Contents

0. 첼린지 코드 리뷰

1. String

- Template Literals
- includes method
- repeat method
- split method

Template Literals

```
const wrapper = document.querySelector(".wrap");

const addWelcome = () => {
  // const helloBox = document.createElement("div");
  // const h1 = document.createElement("h1");
  // h1.innerText = "hello";
  // helloBox.append(h1);
  // wrapper.append(helloBox);

  const helloBox = `
    <div class="hello">
      <h1 class="title">Hello</h1>
    </div>
  `;

  wrapper.innerHTML = div;
};

addWelcome();
```

includes method

```
const isEmail = (email) => email.includes("@");

console.log(isEmail("amamov@amamov.com")); // true
```

repeat method

```
const CC_NUMBER = "6060";

const displayName = `${"*".repeat(10)}${CC_NUMBER}`;

console.log(displayName); // **********6060
```

split method

```
const data = "django@gmail.com";

// @ 기준으로 django와 gmail.com 분리하기

const result = data.split("@");
```

2. Array

Array 생성자

- `Array()` : Array 객체를 생성한다.

Array 정적 메서드

- `Array.from()` : 전달인자를 타입에 관계 없이 새로운 Array를 만든다.
- `Array.of()` : array-like object나 iterable object을 얕게 복사해서 새로운 Array 객체를 만든다.

Array 속성 메서드

- `Array.prototype.length`

Array 변경자 메서드

- `Array.prototype.pop()`
- `Array.prototype.push()`
- `Array.prototype.shift()`
- `Array.prototype.unshift()`
- `Array.prototype.reverse()`
- `Array.prototype.sort()`
- `Array.prototype.splice()` : 배열 **수정**, 제거
- `Array.prototype.fill()`

Array 접근자 메서드 (새로운 객체를 반환)

- `Array.prototype.includes()` : `true` / `false` 반환
- `Array.prototype.slice()`
- `Array.prototype.join()`

Array 순회 메서드

- `Array.prototype.find()`
- `Array.prototype.filter()`
- `Array.prototype.forEach()`
- `Array.prototype.map()`

Array.from()

```
const buttons = document.getElementsByClassName("btn");

console.log(buttons); // HTMLCollection Type 이므로 Array Type으로 바꿔주고 싶다.

Array.from(buttons).forEach((button) => console.log(button));
```

Array.prototype.find()

doc

- 조건으로 만족하는 첫 번째 item을 반환한다.

```
const myArray = [
  "amamov@gmail.com",
  "naver@gmail.com",
  "joy@naver.com",
  "hello@daum.net",
  "world@wow.com",
];

// 조건으로 만족하는 첫 번째 item을 반환한다.
const foundItem = myArray.find((item) => item.includes("@gmail.com"));

console.log(foundItem); // amamov@gmail.com
```

Array.prototype.filter()

doc

- 조건으로 만족하는 모든 item을 반환한다.

```
const data = [
  "yss@gmail.com",
  "django@gmail.com",
  "react@kakao.com",
  "summer@amamov.con",
  "joy@naver.com",
  "hello@daum.net",
  "yyr@gmail.com",
  "world@wow.com",
];

// filter method를 사용하여 gmail을 사용하는 유저를 담은 배열을 뽑기
// [hint] filter, includes
// [출력] ['yss', 'django', 'yyr']
```

Array.prototype.forEach()

doc

- 배열을 순회하면서 각 요소에 대해 액션을 **수행한다.**

```
const data = [
  "yss",
  "django",
  "react",
  "summer",
  "joy",
  "hello",
  "yyr",
  "world",
];

// forEach method를 사용하여 각각의 유저들을 출력하되, 뒤에 @gmail.com을 붙여서 출력하기
// [hint] forEach, +
// [출력] yss@gmail.com django@gmail.com ...
```

Array.prototype.map()

doc

- 배열을 순회하면서 각 요소에 대한 함수의 반환 값을 리턴한다.

```
const myArray = [
  "yss@gmail.com",
  "django@gmail.com",
  "react@kakao.com",
  "summer@amamov.con",
  "joy@naver.com",
  "hello@daum.net",
  "yyr@gmail.com",
  "world@wow.com",
];

// map method를 사용하여 유저 이름을 뽑아내기
// [hint] map, split
// [출력] [ 'yss', 'django', 'react', 'summer', 'joy', 'hello', 'yyr', 'world' ]
```

3. Event Listener

- click event
- submit event
- mouseover event

click event

```
<div id="show"></div>
<input type="button" id="bt" value="클릭!" />
```

```
const div = document.getElementById("show");
const button = document.getElementById("bt");

const handleClick = () => {
  div.innerText = "hello world!!!";
};

const init = () => {
  button.addEventListener("click", handleClick);
};

init();
```

submit event

```
<form id="form">
  <input type="text" id="input" />
</form>
<div id="output"></div>
```

```
const form = document.getElementById("form");
const input = document.getElementById("input");
const output = document.getElementById("output");

const handleSubmit = (event) => {
  event.preventDefault();
  const inputValue = input.value;
  output.innerText = inputValue;
  input.value = "";
};
const init = () => {
  form.addEventListener("submit", handleSubmit);
};
init();
```

mouseover event

```
<style>
  .box {
    background: skyblue;
    height: 100px;
  }
</style>
<div id="box" class="box"></div>
```

```
const box = document.getElementById("box");

const handleMouseover = () => {
  console.log("hello");
};

const init = () => {
  box.addEventListener("mouseover", handleMouseover);
};

init();
```

4. Time Handler

- `setInterval`
- `clearInterval`
- `setTimeout`
- `clearTimeout`

setInterval(callbackFunction, milliseconds)

> `callbackFunction`이 `milliseconds` 시간 마다 실행된다.

```
<div id="timer"></div>
```

```
const timer = document.getElementById("timer");
let count = 0;

const addTime = () => {
  count += 1;
  timer.innerText = count;
};

const init = () => {
  setInterval(addTime, 1000);
};

init();
```

clearInterval(clearFunction)

> `clearFunction`을 중지 시킨다.

```
<div id="timer"></div>
<input type="button" id="bt" value="중지" />
```

```
const timer = document.getElementById("timer");
const stopBt = document.getElementById("bt");
let count = 0;
let intervalFunction;
const addTime = () => {
  count += 1;
  timer.innerText = count;
};
const stopTime = () => clearInterval(intervalFunction);
const init = () => {
  intervalFunction = setInterval(addTime, 1000);
  stopBt.addEventListener("click", stopTime);
};
init();
```

setTimeout(callbackFunction, milliseconds)

> `callbackFunction`이 `milliseconds` 시간 후에 실행된다.

```
<div id="app"></div>
```

```
const app = document.getElementById("app");

const sayHello = () => {
  app.innerText = "hello world!!!";
};

const init = () => {
  setTimeout(sayHello, 3000);
};

init();
```

clearTimeout(timeoutFunction)

> `timeoutFunction`을 중지 시킨다.

첼린지

1. 모던 JS 읽기
2. **시한폭탄 만들기**
    - `01시 30분 21초` 형식으로 시간 입력
    - start 버튼을 누르면 `01시 30분 21초` 에서 1초씩 감소
    - 감소되다가 `00시 00분 00초` 가 되면 폭탄 이미지
    - stop 버튼을 누르면 타이머가 일시정지됨
    - reset 버튼을 누르면 다시 시간을 입력하도록 유도

> 다음주 내용 : object, localStorage CRUD
> 다음주 챌린지 : 시한폭탄 로그 쌓기 (로그 기록, 삭제 기능)

> 다다음주까지 class 배우고 그 다음부터는 계속 토이 플젝 느낌으로 간단한 앱들 개발 -> 연습
> 그후에 p5.js 맛보기 후에 리액트 바로 넘어갑니다