

[ITEC] JS Study

5 week

event listener, time handler, object

© yoon sang seok all rights reserved.

Contents

- 0. 챌린지 코드 리뷰
- 1. Event Listener
- 2. Time Handler
- 3. object

1. 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();
```

[실습] 구구단 게임

```
<form id="form">
  <span>1999 x 17 = ?</span>
  <input type="text" id="input" />
</form>
<div id="output"></div>
```

- [1단계] 1999 x 17 의 값에 대한 답을 input에 적었을 때 맞으면 정답! 틀리면 다시 풀어보세요! 를 #output에 출력
- [2단계] 문제를 랜덤으로 계속 내고 맞추고 반복
- [3단계] 해당 문제를 맞출때마다 점수 올라가기

2. Time Handler

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

setInterval(callbackFunction, milliseconds)

callbackFunction 0 | 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.

time event 원리

3. Object

- Object를 만드는 방법
- computed properties
- in operator
- `Object.values()`
- `Object.keys()`
- Prototype

```
// Javascript
const yoonSangSeok = {
  name: "yoon-sang-seok",
  email: "amamov@kakao.com",
  age: 24,
};
```

```
# Python
yoon_sang_seok = {
  "name": "yoon-sang-seok",
  "email": "amamov@kakao.com",
  "age": 24,
};
```

```
// go lang
var yoonSangSeok = map[string]string {
  "name": "yoon-sang-seok",
  "email": "amamov@kakao.com",
}
```

key should be always string


```
const myObject = { key : value, ...}
```

Object를 만드는 방법

```
const obj1 = {}; // literal
```

```
const obj2 = new Object(); // prototype
```

1. literal표기법을 사용해 객체를 만드는 방법

```
const amamov = {  
  name: "sangSeok", // "name" : "sangSeok", 과 같다.  
  age: 24, // "age" : 24, 과 같다.  
};  
  
const amamov = {  
  ["first" + "name"]: "yoon", // key 값을 연산하는 경우 []로 묶어준다.  
};  
  
//객체에 프로퍼티와 값 추가  
amamov.isGood = true;  
console.log(amamov);  
//{name: "sangSeok", age: 23, isGood: true}  
  
//객체의 element 삭제  
delete amamov.isGood;  
console.log(amamov);  
//{name: "sangSeok", age: 23}
```

2. constructor function : 함수를 사용해 객체를 만드는 방법

```
function Book(author, pages, price, title) {  
  // this = {};  
  this.author = author;  
  this.pages = pages;  
  this.price = price;  
  this.title = title;  
  this.showAuthor = function () {  
    console.log("Author is ", this.author);  
    console.log(this);  
  };  
  // return this;  
}  
  
const pythonBook = new Book("yss", 50, 1, "fcing Python");  
const jsBook = new Book("sangseok", 100, 1, "fcing JS");  
  
console.log(pythonBook);  
jsBook.showAuthor();  
/*  
[출력 결과]  
Book {  
  author: 'yss',  
  pages: 50,  
  price: 1,  
  title: 'fcing Python',  
  showAuthor: [Function (anonymous)]  
}  
Author is  sangseok  
*/
```

3. class를 이용하여 객체를 만드는 방법

```
class Person {  
  //Constructor(생성자)  
  constructor(name, age) {  
    this.name = name;  
    this.age = age;  
  }  
  
  //Methods  
  speak() {  
    console.log(`${this.name} : Hellow !`);  
  }  
}  
  
const amamov = new Person("sangSeok", 23);  
  
console.log(amamov.name); //sangSeok  
console.log(amamov.age); //23
```

Computed properties

```
const amamov = {
  name: "sangSeok",
  age: 23,
};

// Computed properties
console.log(amamov.name); // sangSeok
console.log(amamov["name"]); // sangSeok
// key should be always string
// 대괄호를 쓰면 변수를 넣어서 쓸 수 있다.

function printValue(obj, key) {
  console.log(obj.key); //undefined
  console.log(obj[key]); //sangSeok
}

printValue(amamov, "name");
```

in operator

```
const amamov = {  
  name: "sangSeok",  
  age: 23,  
};  
  
// in operator: property existence check (key in obj)  
console.log("name" in amamov); // true  
console.log("sexy" in amamov); // false
```

values

```
// Object.values(object) : 해당 객체의 모든 value를 배열로 만든다.
```

```
const object1 = {  
  a: "somestring",  
  b: 42,  
  c: false,  
};
```

```
console.log(Object.values(object1));  
// expected output: Array ["somestring", 42, false]
```


keys

```
const object1 = {  
  a: "somestring",  
  b: 42,  
  c: false,  
};  
  
console.log(Object.keys(object1));  
// expected output: Array ["a", "b", "c"]
```

Prototype

JS는 프로토타입 기반 언어이다. 클래스 기반 언어에서는 '상속'을 사용하지만 프로토타입 기반 언어에서는 어떤 객체를 원형(prototype)으로 삼고 이를 복제(참조)함으로써 상속과 비슷한 효과를 얻는다.

챌린지

1. 모던 JS 읽기, object 공부

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 : class, localStorage CRUD

다음주 챌린지 : 시한폭탄 로그 쌓기 (로그 기록, 삭제 기능)



© yoon sang seok all rights reserved.