[효준] 바닐라JS

Property	
만든 사람	Ⓗ HYOJUN KIM
생성 일시	@Mar 30, 2020 9:23 PM
수정 일시	@Apr 12, 2020 9:06 PM
태그	Ⓗ HYOJUN KIM

Nomad Coders의 바닐라 자바스크립트 강좌

▼ 강의를 통해 기초적인 자바스크립트를 이해할 수 있다.

function, variable, object가 무엇인지 얘기할 수 있다.

바닐라 JS로 크롬 앱 만들기

9살부터 프로그래밍을 혼자 독학해서 배운 천재 코더, 13살에 이미 개발자로 입사제의를 받기 시작하였으나, 본인의 자유가 더욱 중요하다며 뻥 차버린 자유로운 영혼의 소유자, f**k이 들어가는 언어를 자유롭게 구사한다. 콜롬비아를 떠나 현재 4년여 넘게 세계를 떠돌아다니면서 디지털노마드로 살고있다. Youtube

https://academy.nomadcoders.co/courses/enrolled/435558

1. Vanilla JS(JavaScript)

Vanilla JS is a fast, lightweight, cross-platform framework for building incredible, powerful JavaScript applications.

- Vanilla JS는 웹사이트를 들어가서 보면 알 수 있듯이, 그냥 순수한 JavaScript를 일컫는 말이다.
- Vanilla는 비격식으로 평범한, 특별할 것 없는 이라는 뜻을 가진 형용사이다.
- 웹에 쓰이는 하나뿐인 프로그래밍 언어 (옵션이 하나뿐)
- 웹사이트를 만들때, 웹사이트를 interactive로 만들고 싶을 때 사용

Speed Comparison

Here are a few examples of just how fast *Vanilla JS* really is:

Retrieve DOM element by ID

	Code	ops / sec
Vanilla JS	document.getElementById('test-table');	12,137,211
Dojo	dojo.byId('test-table');	5,443,343
Prototype JS	$('test-table')	2,940,734
Ext JS	delete Ext.elCache['test-table']; Ext.get('test-table');	997,562
jQuery	$jq('#test-table');	350,557
YUI	YAHOO.util.Dom.get('test-table');	326,534
MooTools	document.id('test-table');	78,802

Retrieve DOM elements by tag name

	Code	ops / sec
Vanilla JS	document.getElementsByTagName("span");	8,280,893
Prototype JS	Prototype.Selector.select('span', document);	62,872
YUI	YAHOO.util.Dom.getElementsBy(function(){return true;},'span');	48,545
Ext JS	Ext.query('span');	46,915
jQuery	$jq('span');	19,449
Dojo	dojo.query('span');	10,335
MooTools	Slick.search(document, 'span', new Elements);	5,457

- Vanilla JS는 120만개가 넘는 Operation(ops)을 1초(sec)에 할 수 있다. (속도가 매우 빠름)

2. let vs const

▼ 기본형 변수(number, string, boolean, null, undefined)를 선언할 때 사용

값의 변경이 필요할 경우 ***let***을 사용

값의 변화가 필요없는 상수로 사용 될 경우 ***const***로 선언 (기본적으로 사용할 것!!)

```
const a = 221;
let b = a-5;
a= 4;
console.log(b,a);
```

3. Data Types on JS

▼ 기본 자료형 (Primitive) 인 여섯가지 데이터 타입

- Boolean
- Null
- Undefined
- Number

- String
- Symbol (ECMAScript 6 에 추가됨)

4. Array, Object

▼ Array

- 데이터를 저장하는 곳으로써 리스트 같이 저장하는 것

```
const daysOfWeek = ["Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun"];

console.log(daysOfWeek)
```

▼ Object

- 실제 객체를 만드는 것 (label을 저장하고 싶은 data에 줄 수 있는)

```
const HyoInfo = {
  name : "HYOJUN",
  age : 24,
  gender : "Male",
  favFood : [
    {
      name: "kimchi",
      fatty: false
    },
    {
      name: "hamburger",
      fatty : true
    }
   ]
}
```

5. Function

▼ Function 과 argument

- Function : 코드 조각으로 원하는 만큼 쓸 수 있는 코드
- Argument (인자) : 변수와 유사, 외부에 있는 데이터를 읽는 함수를 만드는 방법

```
function sayHello(name, age){
   //string 안에 변수 넣어 주는   ${argument}
   console.log("hello! ${name} you are ${age} years old.");
}

const greetHyo = sayHello("hyo", 24)

console.log(greetHyo)
```

```
const calculator = {
 plus: function(a, b){
    return a+b;
  }
}

const plus = calculator.plus(5,5)
console.log(plus)
```

6. Document

html을 dom 객체로 바꿀 수 있다.

```
const title = document.querySelector("#title");
title.innerHTML = "hi! from js"
title.style.color = "Red"
document.title = "hyo"
console.log(title);
```

```
const title = document.querySelector("#title");

function handleClick() {
    title.style.color = "red"
}

title.addEventListener("click", handleClick);
```

7. if, else, and , or

- if(20>5 && "김효준"==="김효준")
- python과 다른점 & && , == ===

8. addEventListener

- 자바스크립트에서는 특정 이벤트가 발생했을 시 특정 함수를 실행할 수 있게 해주는 **addEventListener**라는 기능이 존재

▼ **addEventListener 등으로 등록할 수 있는 이벤트 중 자주 쓰이는 이벤트 목록**

이벤트 목록

Aa 이벤트 명	설명
change	변동이 있을 때 발생
click	클릭시 발생
focus	포커스를 얻었을 때 발생
keydown	키를 눌렀을 때 발생
keyup	키에서 손을 땟을 때 발생
load	로드가 완료 되었을 때 발생
mousedown	마우스를 클릭 했을 때 발생
mouseout	마우스가 특정 객체 밖으로 나갔을 때 발생
mouseover	마우스가 특정 객체 위로 올려졌을 때 발생
mousemove	마우스가 움직였을 때 발생
mouseup	마우스에서 손을 땟을 때 발생
select	option 태그 등에서 선택을 햇을 때 발생

```
const title = document.querySelector("#title");

const BASE_COLOR = "rgb(52, 73, 94)";
const OTHER_COLOR = "#7f8c8d";
function handleClick() {
    const currentColor = title.style.color;
    if (currentColor === BASE_COLOR) {
        title.style.color = OTHER_COLOR;
```

```
    } else {
        title.style.color = BASE_COLOR;
    }
}

function init() {
    title.style.color = BASE_COLOR;
    title.addEventListener("click", handleClick);
}
init();
```

```
const title = document.querySelector("#title");

const CLICKED_CLASS = "clicked";

function handleClick() {
    const hasClass = title.classList.contains(CLICKED_CLASS);
    if (!hasClass) {
        title.classList.add(CLICKED_CLASS);
    } else {
        title.classList.remove(CLICKED_CLASS);
    }
}

function init() {
    title.addEventListener("click", handleClick);
}
init();
```

```
function handleClick() {
  title.classList.toggle(CLICKED_CLASS);
}
```

8. 백틱이란?

키보드 탭 위, 1의 왼쪽, '~'(물결표) 쯤에 있는 '와 비슷하게 생긴 문자다.

백틱(`)을 사용하면 ${}를 사용해서 문자열과 변수를 적절히 같이 사용할 수 있다.

ex) 이전코드

```
const num1 = 10;
const num2 = 20;

console.log(num1 + ' + ' + num2 + ' = ' + (num1+num2) +  " 입니다.");
```

ex) 백틱을 이용한 코드

```
const num1 = 10;
const num2 = 20;

console.log(`${num1} + ${num2} = ${num1+num2} 입니다.`);
```

9. 자바스크립트 주기적 실행

▼ **setInterval 함수**

일정한 시간 간격으로 작업을 수행하기 위해서 사용

clearInterval 함수를 사용하여 중지 가능하지만 일정한 시간 간격으로 실행되는 작업이 그 시간 간격보다 오래걸릴 경우 문제가 발생할 수 있다.

▼ **setTimeout 함수**

:일정한 시간 후에 작업을 한번 실행, 보통 재귀적 호출을 사용하여 작업을 반복

기본적으로 setInterval 과는 달리 지정된 시간을 기다린후 작업을 수행하고, 다시 일정한 시간을 기다린후 작업을 수행하는 방식

지정된 시간 사이에 작업 시간이 추가 되는 것.

clearTimeout() 을 사용해서 작업을 중지

10. Local Storage

- 웹에서 데이터를 저장하는 방법 중 서버가 아닌 클라이언트에서 가능한 방법
- local storage 에는 data를 저장할 수가 없다
- 오직 string만 저장 가능
- boolean 타입인 true 도 string으로 처리되어 저장된다.

! localStorage, sessionStorage를 어떤 경우에 사용할까

대부분의 데이터의 경우 데이터베이스에 저장하므로 로컬 및 세션 스토리지가 사용되는 곳은 그리 많지 않다. 대부분 임시적인 용도 또는 캐시, history 기능을 위하여 사용된다. 아래의 경우가 그 예이다.

- 글 작성 중간 중간에 잃어버리지 않기 위한 임시 저장용도
- 장바구니나 좋아하는 콘텐츠 등 수시로 변경되는 경우
- 방문자의 이동 경로를 저장하였다가 이동할 경우
- 그 외 서버에 반드시 저장할 필요가 없는 경우

11. JSON.stringify()

- JSON은 JavaScript Object Notation의 약자로, 브라우저와 서버사이에서 오고가는 데이터의 형식이다.
- **JSON.stringify(value, replacer, space)**

  value(필수): JSON 문자열로 변환할 값이다.(배열, 객체, 또는 숫자, 문자 등이 될 수 있다.)

  replacer(선택): 함수 또는 배열이 될 수 있다. 이 값이 null 이거나 제공되지 않으면, 객체의 모든 속성들이 JSON 문자열 결과에 포함된다.
- JSON.stringify() 자바스크립트 object를 string으로 바꿔줌

12. JavaScript Array - forEach, map, filter 내장 메소드

▼ forEach

배열의 각 요소에 대해 한번씩 순서대로 콜백함수를 실행

▼ map

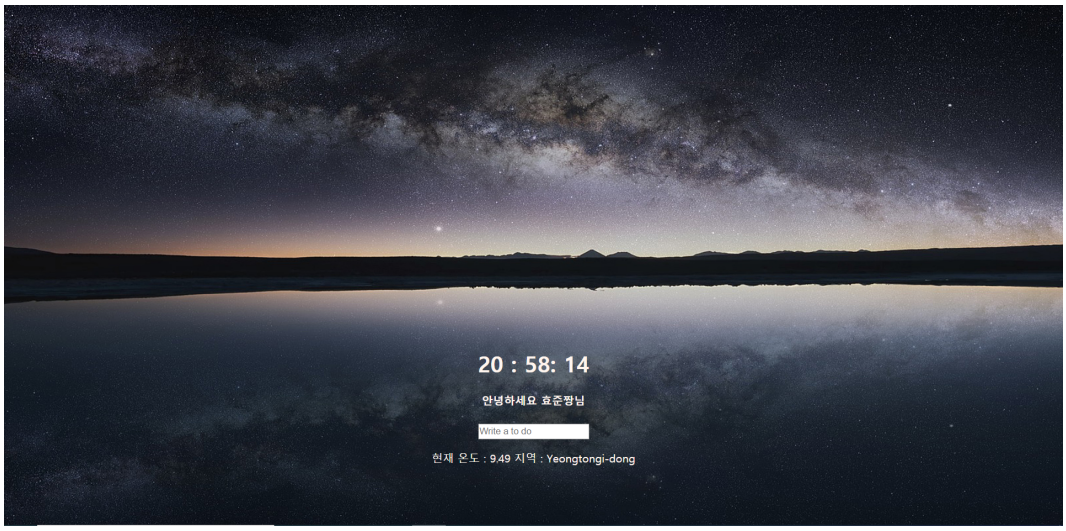
배열의 각각의 요소에 대해 한번씩 순서대로 콜백함수를 실행하고,
그 결과를 모아서 새로운 배열을 반환

▼ filter

배열 내 각 요소에 대해 제공된 콜백함수에서 true를 반환하는 모든 값이 있는 새로운 배열을 반환

즉, 콜백 테스트를 통과한 배열 요소만 반환되고, 통과하지 못한 배열 요소는 건너뛰어 새로운 배열에 포함되지 않는다.

Code



▼ index.html

```
<!DOCTYPE html>
<meta charset="utf-8">
<html>

<head>
    <title>Something</title>
    <style>
        body {
            text-align: center;
            margin: 0px auto;

        }
    </style>
    <link rel="stylesheet" href="index.css" />
</head>

<body>
    <div class="js-clock">
        <h1>00:00</h1>
    </div>
    <form class="js-form form">
        <input type="text" placeholder="What is your name?" />
    </form>
    <h4 class="js-greetings greetings"></h4>
    <form class="js-toDoForm">
        <input type="text" placeholder="Write a to do" />
    </form>
    <ul class="js-toDoList">
    </ul>
    <span class="js-weather"></span>
    <script src="clock.js"></script>
    <script src="gretting.js"></script>
    <script src="todo.js"></script>
    <script src="bg.js"></script>
    <script src="weather.js"></script>
</body>

</html>
```

▼ index.css

```
body{
    background-color:#2c3e50;
}
```

```
body {
    color: #34495e;
}
.btn {
    cursor: pointer;
}

.clicked{
    color : #7f8c8d;
}

.form,
.greetings {
    display : none;
}

.showing{
    display: block;
}

@keyframes fadeIn{
    from{
        opacity:0;
    } to{
        opacity: 1
    }
}


.bgImage{
    position : absolute;
    top:0;
    left:0;
    width: 100%;
    height: 100%;
    z-index: -1;
    animation: fadeIn .5s linear;
}
```

▼ **weather.js**

```
const weather = document.querySelector(".js-weather");

const API_KEY = "ec3b230171ebee645663e55f32eb82e2";
const COORDS = 'coords';

function getWeather(lat, lng) {
    fetch(`https://api.openweathermap.org/data/2.5/weather?lat=${lat}&lon=${lng}&appid=${API_KEY}&units=metric`
    ).then(function (response) {
        return response.json();
    }).then(function (json) {
        const temperature = json.main.temp;
        const place = json.name;
        weather.innerText = `현재 온도 : ${temperature} 지역 : ${place}`;
    });
}


function saveCoords(coordsObj) {
    localStorage.setItem(COORDS, JSON.stringify(coordsObj));
}

function handleGeoSuccess(position) {
    const latitude = position.coords.latitude;
    const longitude = position.coords.longitude;
    const coordsObj = {
        latitude,
        longitude
    };
    saveCoords(coordsObj);
    getWeather(latitude, longitude)
}
function handleGeoError() {
    console.log("cant");
}
```

```
function askForCoords() {
    navigator.geolocation.getCurrentPosition(handleGeoSuccess, handleGeoError)
}
function loadCoords() {
    const loadedCoords = localStorage.getItem(COORDS);
    if (loadedCoords === null) {
        askForCoords();
    } else {
        const parseCoords = JSON.parse(loadedCoords);
        getWeather(parseCoords.latitude, parseCoords.longitude);
    }
}


function init() {
    loadCoords();
}

init();
```

▼ **gretting.js**

```
const form = document.querySelector(".js-form"),
    input = form.querySelector("input"),
    greeting = document.querySelector(".js-greetings");

const USER_LS = "currentUser",
    SHOWING_ON = "showing";

function saveName(text) {
    localStorage.setItem(USER_LS, text);
}

function handleSubmit(event) {
    event.preventDefault();
    const currentValue = input.value;
    paintGreeting(currentValue);
    saveName(currentValue);
}

function askForName() {
    form.classList.add(SHOWING_ON);
    form.addEventListener("submit", handleSubmit);
}

function paintGreeting(text) {
    form.classList.remove(SHOWING_ON);
    greeting.classList.add(SHOWING_ON);
    greeting.innerText = `안녕하세요 ${text}님`;

}
function loadName() {
    const currentUser = localStorage.getItem(USER_LS);
    if (currentUser === null) {
        askForName();
    } else {
        // she is
        paintGreeting(currentUser);
    }
}


function init() {
    loadName();
}

init();
```

▼ **todo.js**

```
const toDoForm = document.querySelector(".js-toDoForm"),
    toDoInput = toDoForm.querySelector("input"),
    toDoList = document.querySelector(".js-toDoList");
```

```
const TODOS_LS = 'toDos';

let toDos = [];

function deleteToDo(event) {
    const btn = event.target;
    const li = btn.parentNode;
    toDoList.removeChild(li);
    const cleanToDos = toDos.filter(function (toDo) {
        return toDo.id !== parseInt(li.id);
    });
    toDos = cleanToDos;
    saveToDos();
}
//local storage에 저장
function saveToDos() {
    localStorage.setItem(TODOS_LS, JSON.stringify(toDos));
}

function paintTodo(text) {
    const li = document.createElement("li"); //li 태그 생성
    const delBtn = document.createElement("button");
    const span = document.createElement("span");
    const newId = toDos.length + 1;
    delBtn.innerText = "X";
    delBtn.addEventListener("click", deleteToDo);
    span.innerText = text
    li.appendChild(span);
    li.appendChild(delBtn);
    li.id = newId;
    toDoList.appendChild(li);
    const toDoObj = {
        text: text,
        id: newId
    };
    toDos.push(toDoObj);
    saveToDos();
}

function handleSubmit(event) {
    event.preventDefault();
    const currentValue = toDoInput.value;
    paintTodo(currentValue);
    toDoInput.value = "";
}
function something(toDo) {
    paintTodo(toDo.text);
}
function loadToDos() {
    const loadedToDos = localStorage.getItem(TODOS_LS);
    if (loadedToDos !== null) {
        const parsedToDos = JSON.parse(loadedToDos);
        parsedToDos.forEach(something);
    }
}
function init() {
    loadToDos();
    toDoForm.addEventListener("submit", handleSubmit);

}

init();
```

▼ bg.js

```
const body = document.querySelector("body");

const IMG_NUMBER = 3;

function paintImage(imgNumber) {
    const image = new Image();
    image.src = `images/${imgNumber + 1}.jpg`;
    image.classList.add("bgImage");
    body.appendChild(image);
}
```

```
function genRandom() {
    const number = Math.floor(Math.random() * IMG_NUMBER);
    return number;
}

function init() {
    const randomNumber = genRandom();
    paintImage(randomNumber);
}

init();
```