



[스파르타코딩클럽] 웹개발 종합반 - 2주차



매 주차 강의자료 시작에 PDF파일을 올려두었어요!

▼ PDF 파일

▼ 단축키 모음

▼ 새로고침

- `F5`

▼ 저장

- Windows: `Ctrl` + `S`
- macOS: `command` + `S`

▼ 전체선택

- Windows: `Ctrl` + `A`
- macOS: `command` + `A`

▼ 잘라내기

- Windows: `Ctrl` + `X`
- macOS: `command` + `X`

▼ 콘솔창 줄바꿈

- `shift` + `enter`

▼ 코드정렬

- Windows: `Shift` + `Alt` + `F`
- macOS: `Shift` + `Option` + `F`

▼ 들여쓰기

- `Tab`
- 들여쓰기 취소 : `Shift` + `Tab`

▼ 주석

- Windows: `Ctrl` + `/`
- macOS: `command` + `/`

[수업 목표]

1. Javascript의 사용방식에 대해 이해하고 문법에 익숙해진다
2. JQuery로 HTML을 조작할 수 있다
3. Fetch로 서버 API에 데이터를 주고, 결과를 받아온다

[목차]

- [01. 2주차 오늘 배울 것](#)
- [02. Javascript 맛보기](#)
- [03. Javascript 기초 문법 배우기](#)
- [04. Javascript & JQuery 연습하기 \(1\)](#)

05. Javascript & JQuery 연습하기 (2)

06. JQuery 연습하기

07. 서버-클라이언트 통신 이해하기

08. Fetch 시작하기 (1)

09. Fetch 시작하기 (2)

10. Fetch 연습하기 (1)

11. Fetch 연습하기 (2)

12. Fetch 퀴즈 (1)

13. Fetch 퀴즈 (2)

14. 2주차 끝 & 숙제 설명

HW. 2주차 숙제 해설



모든 토글을 열고 닫는 단축키

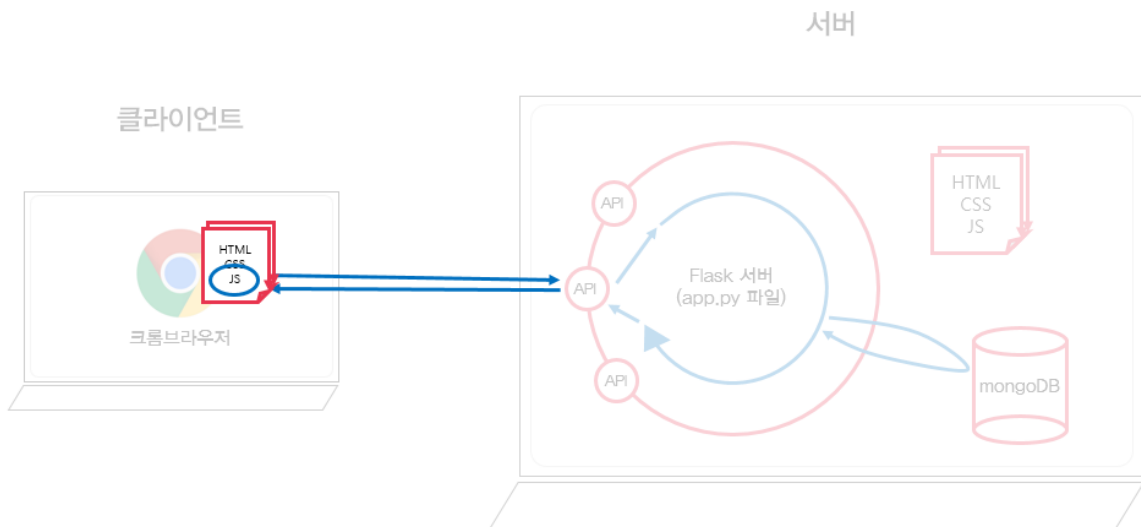
Windows : **Ctrl** + **alt** + **t**

Mac : **⌘** + **⌥** + **t**

01. 2주차 오늘 배울 것



1주차는 뼈대를 만들고 꾸며보았다면, 이번 주차에는 실제로 데이터가 움직이게 만들어 볼거예요!



▼ 1) 2주차 : Javascript, JQuery, Fetch

- **Javascript** 는 웹을 움직이게 하는 코드예요!
- **jQuery** 는 html 뼈대를 선택해서 쉽게 조작할 수 있어요!
- **Fetch** 는 짧은 코드로 요청을 보내고 받아올 수 있어요!



이거 세 개를 하면 뭘 할 수 있는데요?

- 모바일 청첩장으로 예를 들어보자면요!
 - 지난 주차까지는 고정된 모바일 청첩장만 만들어 보낼 수 있어요!
 - 이번 주차까지 배우면, 요청을 보내면 이름만 바뀌어서 재사용할 수 있는 모바일 청첩장을 만들 수 있게 되는거예요!

02. Javascript 맛보기

▼ 1) 자바스크립트란?

- 프로그래밍 언어 중 하나로, 브라우저가 알아들을 수 있는 언어입니다.



이론 설명 때, 클라이언트가 서버에 요청하면, 서버가 클라이언트에게 HTML+CSS+Javascript를 준다고 했던 것, 기억하시나요?



[잠깐 상식]

Q. 왜 브라우저는 Javascript만 알아들어요? HTML안에도 파이썬, Java 같은 언어를 써서 주면 안되나요?

A. 불가능한 이야기는 아닙니다. 다만, 이 "역사적인 이유 & 이미 만들어진 표준"이기 때문에, 모든 브라우저는 기본적으로 Javascript를 알아들게 설계되어있고, 모든 웹서버는 HTML+CSS+Javascript를 주게 되어있죠.

- Java와 Javascript는 어떤 차이가 있나요?



인도와 인도네시아..
바다와 바다코끼리..

아무 관련 없습니다.

▼ 2) 자바스크립트 기초

- 처음 프로그래밍 언어를 배우면 생소한 부분도 존재하기에, 다음 주에 본격적으로 하기 전에! 오늘 먼저 맛보기를 해보겠습니다.



일단 따라쓰기!를 통해,
자바스크립트가 HTML과 어떻게 연동되는지 알아보겠습니다.

▼ 3) 자바스크립트 - HTML 연결. 버튼을 클릭하면 경고창이 뜨게하기

- 함수를 만들어두기

```
function hey(){  
  alert('안녕!');  
}
```



<head> ~ </head> 안에 <script> ~ </script> 로 공간을 만들어 작성합니다.

<script> ~ </script> 내에 자바스크립트를 작성하는 것이죠
아래 코드를 통해 간단한 사용방법을 알아봅시다.

```

82 |         .mybttns > button {
83 |             margin-right: 10px;
84 |         }
85 |     </style>
86 |     <script>
87 |         function hey(){
88 |             alert('안녕!');
89 |         }
90 |     </script>
91 | </head>
92 |
93 | <body>
94 | <div class="mytitle">
95 |     <h1>내 생애 최고의 영화들</h1>
96 |     <button onclick="hey()">영화 기록하기</button>
97 | </div>

```

- 버튼에 함수를 연결하기. 버튼을 누르면 함수가 불립니다.

```
<button onclick="hey()">영화 기록하기</button>
```

03. Javascript 기초 문법 배우기

- ▼ 1) 본격적으로 문법을 배워볼까요?

🕶 예제를 써먹기 위해서 문법을 배워봐요!

- ▼ [크롬 개발자도구]를 열어서, console 탭에 작성합니다!

👉 그냥 쉽게, "마우스 오른쪽 클릭 → 검사 → console"도 가능!

크롬 개발자도구 콘솔창은 어떤 의미?

> 띄워놓은 페이지에서 빠르게 자바스크립트를 테스트해볼 수 있게, 개발자들을 위해 만들어진 도구입니다. 새로고침 하면 모두 사라진다는 사실!

- 윈도우: F12
- 맥: fn(🌐) + F12

👁 Live Server를 활용해 화면을 반으로 나눠 코드를 적으면서 확인해보세요!

- console.log(변수)

--
console.log(변수) 는, 콘솔 창에 괄호 안의 값을 출력해줍니다.
개발자가 결과값을 보기 편하도록!

console.log(변수1, 변수2) 로 여러 변수를 한번에 출력할 수도 있어요.
아래를 복사해서 붙여넣어보세요.

```
console.log("Hello World!");
```

▼ 변수 & 기본연산

- 변수 대입(a = 2)의 의미: "오른쪽에 있는 것을 왼쪽에 넣는 것!"
(2를 a라는 변수에 넣는다)
- let으로 변수를 선언합니다.

```
let num = 20
num = 'Bob'

// 변수는 값을 저장하는 박스예요.
// 한 번 선언했으면, 다시 선언하지 않고 값을 넣습니다.
```

- 사칙연산, 그리고 문자열 더하기가 기본적으로 가능합니다.

```
let a = 1
let b = 2

a+b // 3
a/b // 0.5

let first = 'Bob'
let last = 'Lee'

first+last // 'BobLee'

first+' '+last // 'Bob Lee'

first+a // Bob1 -> 문자+숫자를 하면, 숫자를 문자로 바꾼 뒤 수행합니다.
```

- 변수명은 어떻게하나?

```
let first_name = 'bob' // snake case라고 합니다.

또는,

let firstName = 'bob' // camel case라고 합니다. 회사마다 규칙이 있죠.

과 같이, 쉽게 알아볼 수 있게 쓰는 게 중요합니다.
다른 특수문자 또는 띄워쓰기는 불가능합니다!
```

▼ 리스트 & 딕셔너리

- 리스트: 순서를 지켜서 가지고 있는 형태입니다.

```
let a_list = [] // 리스트를 선언. 변수 이름은 역시 아무렇게나 가능!

// 또는,

let b_list = [1,2,'hey',3] // 로 선언 가능

b_list[1] // 2 를 출력
b_list[2] // 'hey'를 출력

// 리스트에 요소 넣기
b_list.push('헤이')
b_list // [1, 2, "hey", 3, "헤이"] 를 출력

// 리스트의 길이 구하기
b_list.length // 5를 출력
```

- 딕셔너리: 키(key)-밸류(value) 값의 묶음

```
let a_dict = {} // 딕셔너리 선언. 변수 이름은 역시 아무렇게나 가능!

// 또는,

let b_dict = {'name':'Bob','age':21} // 로 선언 가능
b_dict['name'] // 'Bob'을 출력
b_dict['age'] // 21을 출력

b_dict['height'] = 180 // 딕셔너리에 키:밸류 넣기
b_dict // {name: "Bob", age: 21, height: 180}을 출력
```

- 리스트와 딕셔너리의 조합

```
names = [{'name': 'bob', 'age': 20}, {'name': 'carry', 'age': 38}]

// names[0]['name']의 값은? 'bob'
// names[1]['name']의 값은? 'carry'

new_name = {'name': 'john', 'age': 7}
names.push(new_name)

// names의 값은? [{'name': 'bob', 'age': 20}, {'name': 'carry', 'age': 38}, {'name': 'john', 'age': 7}]
// names[2]['name']의 값은? 'john'
```

- 왜 필요할까요?



순서를 표시할 수 있고, 정보를 묶을 수 있습니다.

앞에서 언급한 <스파르타과일가게>가 정말 잘 되어서 전국에서 손님이 찾아오고 있습니다. 대기표를 작성하기 위해서 온 순서대로 이름, 휴대폰 번호를 적도록 하였는데요. 변수만을 사용한 모습은 다음과 같습니다.

```
let customer_1_name = '김스파';
let customer_1_phone = '010-1234-1234';
let customer_2_name = '박르탄';
let customer_2_phone = '010-4321-4321';
...(알아보기 힘듭니다.)
```

👉 딕셔너리를 활용한다면 다음과 같이 고객 별로 정보를 묶을 수 있습니다.

```
let customer_1 = {'name': '김스파', 'phone': '010-1234-1234'};
let customer_2 = {'name': '박르탄', 'phone': '010-4321-4321'};
```

👉 그리고 순서를 나타내기 위해 리스트를 사용하면, 이렇게나 깔끔해집니다.

```
let customer = [
  {'name': '김스파', 'phone': '010-1234-1234'},
  {'name': '박르탄', 'phone': '010-4321-4321'}
]
```

✅ 보기에다 깔끔해지고, 다루기도 쉬워지고, 고객이 새로 한 명 더 오더라도 .push 함수를 이용해 간단하게 대응할 수 있습니다.

▼ 기본 함수들

- 사칙연산 외에도, 기본적으로 제공하는 여러 함수들이 존재합니다.



웬지 이걸 있을 것 같은데?(예 - 특정 문자를 바꾸고 싶다 등) 싶으면 직접 만들지 말고 **구글에 먼저 찾아보세요!**

예를 들면, '나눠셈의나머지'를 구하고 싶은 경우

```
let a = 20
let b = 7

a % b = 6
```

또, 특정 문자로 문자열을 나누고 싶은 경우

```
let myemail = 'sparta@gmail.com'

let result = myemail.split('@') // ['sparta', 'gmail.com']

result[0] // sparta
result[1] // gmail.com
```

```
let result2 = result[1].split('.') // ['gmail','com']


result2[0] // gmail -> 우리가 알고 싶었던 것!
result2[1] // com

myemail.split('@')[1].split('.')[0] // gmail -> 간단하게 쓸 수도 있다!
```

04. Javascript & JQuery 연습하기 (1)


▼ 1) JQuery가 뭐라고요?

- ▼ HTML의 요소들을 조작하는, 편리한 Javascript를 미리 작성해둔 것. 라이브러리!

 Javascript로도 모든 기능(예 - 버튼 글씨 바꾸기 등)을 구현할 수는 있지만,

1) 코드가 복잡하고, 2) 브라우저 간 호환성 문제도 고려해야해서,
jQuery라는 라이브러리가 등장하게 되었습니다.

▼ jQuery와 Javascript - 코드 비교해보기

 jQuery는 Javascript와 다른 특별한 소프트웨어가 아니라 미리 작성된 Javascript 코드입니다. 전문 개발자들이 짜둔 코드를 잘 가져와서 사용하는 것임을 기억해주세요! (그렇게 때문에, 쓰기 전에 "임포트"를 해야합니다!)

Javascript 로 길고 복잡하게 써야 하는 것을


```
document.getElementById("element").style.display = "none";
```

jQuery 로 보다 직관적으로 쓸 수 있어요. 편리하죠? :-)

```
$('#element').hide();
```

▼ jQuery 사용하기


- 미리 작성된 Javascript 코드를 가져오는 것을 '임포트'라고 부릅니다.

 jQuery CDN 부분을 참고해서 임포트하기: ([링크](#)).

▼ [코드스니펫] jQuery CDN

```
https://www.w3schools.com/jquery/jquery_get_started.asp
```

- <head> 와 </head> 사이에 아래를 넣으면 끝!

 수업 자료를 잘 따라온 분이라면, 코드에 이미 다음과 같이 임포트 되어있을거예요. 다시한번 잘 살펴보세요!

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
```

- jQuery를 사용하는 방법



css와 마찬가지로, jQuery를 쓸 때에도 "가리켜야" → 조작할 수 있습니다.

예) 특정 input박스의 값을 → 가져와줘!

예) 특정 div를 → 안보이게 해줘!

css에서는 선택자로 class를 썼지요?

jQuery에서는 id 값을 통해 특정 버튼/input박스/div.. 등을 가리키게 됩니다.

백문이불여일견! 자주 쓰는 jQuery들을 함께 다뤄보면서 익혀보죠!

▼ 2) 화면에 띄워보면서 이해하기



HTML 뼈대를 실제로 조작해보면서 Javascript와 JQuery를 몸으로 익혀봐요!



새로운 폴더 `jsprac` 을 만들고, 그 안에 `prac.html` 로 시작해봐요!



Live Server를 활용해 화면을 반으로 나눠 코드를 적으면서 확인해보세요!

▼ [코드스니펫] prac.html - JQuery 연습하기 뼈대

```
<!DOCTYPE html>
<html>
  <head>
    <title>자바스크립트 문법 연습하기!</title>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
  </head>
  <style>
    .button-part {
      display: flex;
      height: 50px;
    }
  </style>
  <script>

  </script>
  <body>
    <div class="top-part">
      <h1>자바스크립트 문법 연습하기!</h1>
    </div>
    <hr/>
    <br>
    <h2>1. 함수</h2>
    <div class="button-part">
      <button onclick="checkResult()">결과 확인하기!</button>
    </div>
    <div class="list-part">
      <h2>2. 리스트</h2>
      <div id="q1"></div>
    </div>
    <div class="dict-part">
      <h2>3. 딕셔너리</h2>
      <div id="q2"></div>
    </div>
    <div>
      <h2>4. 리스트 딕셔너리</h2>
      <div id="q3"></div>
    </div>
  </body>
</html>
```

▼ 1) 함수 만들기

- 브라우저에 보이는 결과 확인하기! 버튼을 눌렀을 때, 여러 값들을 보이게 하려면?
 - → 반복적인 작업을 계속 수행할 수 있게 하는 도구, `함수`가 필요해요!

- 주석 처리(//) 코드를 사용해서 설명이나, 실행하고 싶지 않은 코드는 숨겨주세요!
- <script></script> 태그 안에 작성해주세요!

```
function checkResult() {
  // 여기에 코드를 적어요!
}
```

▼ 2) alert

- `alert()` 는 브라우저에서 경고 창 을 띄워주는 자바스크립트 함수예요!
- 로그인에 필요합니다 와 같은 경고 창 보신 적 있지 않나요?

```
function checkResult() {
  alert('함수 연결!')
}
```

▼ 3) console.log

- 개발자 도구를 통해서 볼 수 있는 `console.log()` 기억하시죠?

```
function checkResult() {
  alert('함수 연결!')
  console.log('함수 연결!')
}
```

▼ 3) 자료형

😎 이전에 배웠던 여러 문법들을 활용해 HTML 태그로 붙여봐요!

▼ 1) 변수 선언

😎 여러가지 자료형을 다루기 위해서는 변수를 선언해야 한다는 것 다시 떠올려봅시다!

▼ 1) 리스트 형식

🤔 리스트는 순서를 지켜서 데이터를 저장하는 형태!

▼ [코드스니펫] 자바스크립트 리스트 자료형

```
let fruitList = ['딸기', '수박', '참외', '포도']
```

- 자료형을 먼저 개발자도구에서 한 번 볼까요?

```
let fruitList = ['딸기', '귤', '체리', '포도']
console.log(fruitList)
console.log(fruitList[0])
```

▼ 2) 딕셔너리 형식

🤔 딕셔너리는 키(Key)와 밸류(Value)의 묶음!

▼ [코드스니펫] 자바스크립트 딕셔너리 자료형

```
let nameDict = {
  "name" : "영수",
  "age" : 25
}
```

- 딕셔너리를 개발자도구에서 먼저 봅시다!

```
let nameDict = {
  "name" : "영수",
  "age" : 25
}
console.log(nameDict)
console.log(nameDict['name'])
console.log(nameDict['age'])
```

키 값으로 부르면 딕셔너리의 밸류를 부를 수 있다! 이해되시죠?

```
▶ {name: '영수', age: 25}
영수
25
```

- 또 원하는 값이 있다면 키-밸류로 추가할 수도 있었어요!

```
nameDict['height'] = 178
console.log(nameDict)
```

▼ 3) 리스트-딕셔너리 형식



리스트에 딕셔너리를 연결한 형식까지 가봅시다!

▼ [코드스니펫] 리스트-딕셔너리 자료형

```
let membersFruitList = [
  {
    'name' : 'lee',
    'fav' : '🍇'
  },
  {
    'name' : 'hwang',
    'fav' : '🍒'
  },
  {
    'name' : 'joo',
    'fav' : '🍊'
  }
]
```

- 리스트-딕셔너리를 개발자도구에서 먼저 봅시다!

```
let membersFruitList = [
  {
    'name' : 'lee',
    'fruit' : '포도'
  },
  {
    'name' : 'hwang',
    'fruit' : '체리'
  },
  {
    'name' : 'joo',
    'fruit' : '귤'
  }
]
```

```
}  
console.log(membersFruitList)
```

▼ 2) JQuery로 HTML 조작하기

😎 작성한 자료형 변수를 이제 가져다가 써봅시다!

▼ html 태그 선택하기

😎 선언해준 리스트 변수와 HTML 태그를 연결해서 화면에 붙여줄 겁니다!

- HTML 태그를 선택하기 위해서는 필요한 것이 있어요! 바로 **id 값** 입니다!
- **id** 는 태그에서 **유일한 요소를 가리키기 위해 있는 속성**입니다!
 - → **class** 는 여러 개에 적용할 수 있지만, **id** 는 여러 개를 고를 수 없다는 이야기죠!
- 태그 안에 id 값이 이미 적혀 있어요 한 번 확인해 볼까요?

리스트라는 제목 아래의 id="q1"이라고 적혀있는 div태그에 우리는 값을 넣을겁니다!

```
<div class="list-part">  
  <h2>2. 리스트</h2>  
  <div id="q1"></div>  
</div>
```

- id가 q1이라는 div태그를 선택하는 방법은 아래와 같습니다!

`$('#아이디')` 라고 작성하면 같은 아이디 값에 해당하는 html요소를 선택합니다!

```
$('#q1') // #은 html요소 중 id 속성을 가리키는 문자예요!
```

▼ `.text()` 를 활용해 값을 넣기

😎 JQuery로 HTML요소를 선택했다면, 이제 값을 넣어봐요!

- `.text()` 는 html 요소 안의 값을 바꿔줍니다.

```
$('#q1').text(fruitList) // 아이디가 q1인 html 요소의 값을 fruitList로
```

▼ 리스트, 딕셔너리, 리스트+딕셔너리 조합

⇒ JQuery를 이용하여 아래와 같이 만들어볼까요?

1. 함수

결과 확인하기!

2. 리스트

배

3. 딕셔너리

영수

4. 리스트 딕셔너리

35

▼ [코드스니펫] 완성 코드

```
<!DOCTYPE html>
<html>
  <head>
    <title>자바스크립트 문법 연습하기!</title>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
  </head>
  <style>
    .button-part {
      display: flex;
      height: 50px;
    }
  </style>
  <script>
    function checkResult(){
      let a = ['사과', '배', '감', '귤']
      $('#q1').text(a[1])

      let b = {'name': '영수', 'age': 30}
      $('#q2').text(b['name'])

      let c = [
        {'name': '영수', 'age': 30},
        {'name': '철수', 'age': 35}
      ]
      $('#q3').text(c[1]['age'])
    }
  </script>
  <body>
    <div class="top-part">
      <h1>자바스크립트 문법 연습하기!</h1>
    </div>
    <hr/>
    <br>
    <h2>1. 함수</h2>
    <div class="button-part">
      <button onclick="checkResult()">결과 확인하기!</button>
    </div>
    <div class="list-part">
      <h2>2. 리스트</h2>
      <div id="q1">테스트</div>
    </div>
    <div class="dict-part">
      <h2>3. 딕셔너리</h2>
      <div id="q2">테스트</div>
    </div>
    <div>
      <h2>4. 리스트 딕셔너리</h2>
      <div id="q3">테스트</div>
    </div>
  </body>
</html>
```

05. Javascript & JQuery 연습하기 (2)

▼ 1) 반복문



리스트, 리스트-딕셔너리 형식의 자료는 하나하나 뽑아써야 하나요?

- 리스트 자료형의 데이터를 일일이 뽑아보는데..

```
let fruits = ['사과', '배', '감', ... , '귤']
console.log('사과')
console.log('배')
console.log('감')
...
console.log('귤')
```

// 이렇게 100개 씩 쓰기엔 무리가 있겠죠? 그래서, 반복문이라는 것이 존재합니다!

- 우리는 그 중에서도 forEach라는 친구로 간단하게 뽑아볼 거예요!

```
fruits.forEach((a) => {
  console.log(a)
})
// fruits 의 요소를 하나씩 확인하는데 이름은 a로 할 것
```

▼ 2) 조건문



반복문과 더불어 조건에 맞춰 실행을 다르게 해주는 “조건문”도 프로그래밍에선 빼놓을 수 없습니다!

```
if (조건) {
  // 조건에 맞다면~
} else {
  // 아니라면~
}
```

⇒ 만약 20살보다 크면 **성인입니다** 작으면 **청소년입니다** 를 출력하려면?

```
let age = 35

if (age > 20) {
  console.log('성인입니다')
} else {
  console.log('청소년입니다')
}
```

⇒ 반복문+조건문 합치기!

```
let ages = [20,15,12,25,33,40]

ages.forEach((a)=> {
  if (a > 20) {
    console.log('성인입니다')
  } else {
    console.log('청소년입니다')
  }
})
```

06. JQuery 연습하기

▼ 1) JQuery - append

▼ [코드스니펫] 새로운 HTML 코드

```
<!DOCTYPE html>
<html>
```

```

<head>
  <title>자바스크립트 문법 연습하기!</title>
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
</head>
<script>
  function checkResult() {

  }
</script>

<body>
  <div class="top-part">
    <h1>자바스크립트 문법 연습하기!</h1>
  </div>
  <hr />
  <br>
  <h2>1. 함수</h2>
  <div class="button-part">
    <button onclick="checkResult()">결과 확인하기!</button>
  </div>
  <div class="list-part">
    <h2>2. 불이기</h2>
    <div id="q1">
      <p>사과</p>
      <p>귤</p>
      <p>감</p>
    </div>
  </div>
  <div class="list-part">
    <h2>3. 불이기</h2>
    <div id="q2">
      <p>영수는 24살입니다.</p>
      <p>세종은 30살입니다.</p>
      <p>수영은 20살입니다.</p>
    </div>
  </div>
</body>

</html>

```

▼ [코드스니펫] people

```

let people = [
  { 'name': '서영', 'age': 24 },
  { 'name': '현아', 'age': 30 },
  { 'name': '영환', 'age': 12 },
  { 'name': '서연', 'age': 15 },
  { 'name': '지용', 'age': 18 },
  { 'name': '예지', 'age': 36 }
]

```

▼ [코드스니펫] fruits

```

let fruits = ['사과', '배', '감', '귤', '수박']

```

▼ .append() 를 활용하기

```

let temp_html = `<p>영수는 24살</p>`
$('#q3').append(temp_html)

```

🤔 어? 🍎 에 \${ } 까지 너무 어려운데요?

- 걱정마세요! 왜 이렇게 생겼는지 이해하려고 하지 마시고, 규칙이 있다고만 생각하고 작성하시면 얼마든지 따라 쓸 수 있어요!
- 1) 원하는 html 태그를 백틱(` `)으로 묶어 주세요!
- 2) 태그 안에 들어갈 값은 \${ } 로 표시하고, 그 안에는 자료가 있는 변수를 넣어주세요!
- 3) 통째로 넣을 html 요소를 \$('#아이디') 로 골라주시고 append(변수) 를 넣어주세요!



아직 너무 어렵다면, 계속 반복해서 연습할 테니 걱정마세요!

▼ 문자와 변수를 같이 쓰는 백틱(` `)

- 아까 리스트를 HTML에 추가할 때 썼던 백틱(` `) 기억하시나요? HTML요소를 통째로 넣을 때 묶어주는 기호였었습니다!
- 이 **백틱**은 문자와 변수를 함께 써줄 수 있도록 하는 특수기호입니다! 이렇게도 쓸 수 있죠!

```
let profile = `${nameDict['name']}의 나이는 ${nameDict['age']}살 입니다`
```

- 이러면 한꺼번에 변수와 문자를 쓸 수 있게 되었습니다. `\${}` 자리에 변수만 넣어줘 볼까요?
- 우리가 원하는 이름 값은 nameDict['name']으로 가져올 수 있었죠?
- 우리가 원하는 나이 값은 nameDict['age']로 가져올 수 있었습니다!

```
let profile = `${nameDict['name']}의 나이는 ${nameDict['age']}살 입니다`
```

07. 서버-클라이언트 통신 이해하기

▼ 1) 서버 → 클라이언트: "JSON"을 이해하기

▼ 서울시 OpenAPI에 접속해보기

▼ [코드스니펫] 서울시 미세먼지 OpenAPI

```
http://openapi.seoul.go.kr:8088/6d4d776b466c656533356a4b4b5872/json/RealtimeCityAir/1/99
```

▼ 크롬 익스텐션 JSONView를 설치해볼까요? 그럼 좀 더 예쁘게 JSON을 볼 수 있습니다.

▼ [코드스니펫] Jsonview

```
https://chrome.google.com/webstore/detail/jsonview/chklaanhfefbnpoihckbnefhakgolnmc?hl=ko
```

▼ JSON은, Key:Value로 이루어져 있습니다. 자료형 Dictionary와 아주- 유사하죠



위 예제에서는 RealtimeCityAir라는 키 값에 딕셔너리 형 value가 들어가있고, 그 안에 row라는 키 값에는 리스트형 value가 들어가있습니다.

```
{
  - RealtimeCityAir: {
    + list_total_count: 25,
    + RESULT: { ... },
    + row: [ ... ]
  }
}
```

key (딕셔너리 형) value

```
{
  - RealtimeCityAir: {
    + list_total_count: 25,
    + RESULT: { ... },
    + row: [ ... ]
  }
}
```

key value (리스트 형)

▼ 2) 클라이언트 → 서버: GET 요청 이해하기

👉 API는 은행 창구와 같은 것!

같은 예금 창구에서도 개인 고객이나 기업 고객이나에 따라 가져와야 하는 것 / 처리해주는 것이 다른 것처럼,

클라이언트가 요청 할 때에도, "타입"이라는 것이 존재합니다.

* GET → 통상적으로! 데이터 조회(Read)를 요청할 때
예) 영화 목록 조회

* POST → 통상적으로! 데이터 생성(Create), 변경(Update), 삭제>Delete) 요청 할 때
예) 회원가입, 회원탈퇴, 비밀번호 수정

이 중에서 오늘은 GET 방식에 대해 배워보겠습니다. (POST는 4주차에 배웁니다)

• GET

기생충

"폐 끼치고 싶진 않았어요" 전원백수로 살 길 막막하지만 사이는 좋은 기택(송강호) 가족. 장남 기우(최우식)에게 명문대생 친구가 연결시켜 준 고액 과외 자리는 모처럼 싹튼 고정수입의 희망이다. 온 가족의 도움과 기대 속에 박사장(이선균) 집으로 향하는 기우. 글로벌 IT기업 CEO인 박사장의 저택에 도착하자 절고 아름다운 사

🌐 <https://movie.naver.com/movie/bi/mi/basic.nhn?code=161967>



<https://movie.naver.com/movie/bi/mi/basic.nhn?code=161967>

위 주소는 크게 두 부분으로 쪼개집니다. 바로 "?"가 쪼개지는 지점인데요.
"?" 기준으로 앞부분이 <서버 주소>, 뒷부분이 [영화 번호] 입니다.

* 서버 주소: <https://movie.naver.com/movie/bi/mi/basic.nhn>
* 영화 정보: code=161967

👉 GET 방식으로 데이터를 전달하는 방법

? : 여기서부터 전달할 데이터가 작성된다는 의미입니다.

& : 전달할 데이터가 더 있다는 뜻입니다.

예시) google.com/search?q=아이폰&sourceid=chrome&ie=UTF-8

위 주소는 google.com의 search 창구에 다음 정보를 전달합니다!

q=아이폰	(검색어)
sourceid=chrome	(브라우저 정보)
ie=UTF-8	(인코딩 정보)

👉 여기서 잠깐, 그럼 code라는 이름으로 영화번호를 주자!는 것은 누가 정하는 것일까요?

→ 네, 바로 프론트엔드 개발자와 백엔드 개발자가 미리 정해둔 약속입니다.

프론트엔드: "code라는 이름으로 영화번호를 주면 될까요?"

백엔드: "네 그렇게 하시죠. 그럼 code로 영화번호가 들어온다고 생각하고 코딩하고 있을게요"

08. Fetch 시작하기 (1)

▼ 1) Fetch 연습을 위한 파일

 `prac2.html` 를 만들고 시작합니다!

▼ [코드스니펫] fetch 연습하기

```
<!doctype html>
<html lang="ko">
<head>
  <meta charset="UTF-8">
  <title>Fetch 시작하기</title>
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
</script>
</head>
<body>
  Fetch 연습을 위한 페이지
</body>
</html>
```

▼ 2) Fetch 시작하기

- 크롬 개발자 도구에 다음과 같이 써보세요



참고! Fetch를 쓰면서 jQuery를 사용할 것이기 때문에
jQuery를 임포트한 페이지에서만 동작 할 거예요!

즉, <http://google.com/> 과 같은 화면에서 개발자도구를 열면, jQuery가 임포트 되어있지 않기 때문에 아래와 같은 에러가 뜹니다.

Uncaught TypeError: \$ is not a function
→ jQuery 라는 게 없다는 뜻

▼ [코드스니펫] 미세먼지 OpenAPI

```
http://spartacodingclub.shop/sparta_api/seoulair
```

▼ [코드스니펫] Fetch 기본 골격

```
fetch("여기에 URL을 입력").then(res => res.json()).then(data => {
  console.log(data)
})
```

Fetch 코드 해설

```
fetch("여기에 URL을 입력") // 이 URL로 웹 통신을 요청한다. 괄호 안에 다른 것이 없다면 GET!
.then(res => res.json()) // 통신 요청을 받은 데이터는 res라는 이름으로 JSON화 한다
.then(data => {
  console.log(data) // 개발자 도구에 찍어보기
}) // JSON 형태로 바뀐 데이터를 data라는 이름으로 불러 사용한다
```

• Fetch 코드 설명

- `fetch("여기에 URL을 입력")` ← 이 URL로 웹 통신 요청을 보낼 거야!
 - ← 이 괄호 안에 **URL밖에 들어있지 않다**면 기본상태인 **GET!**
- `.then()` ← 통신 요청을 받은 다음 이렇게 할 거야!
- `res => res.json()`
 - ← 통신 요청을 받은 데이터는 res 라는 이름을 붙일 거야(변경 가능)

- ← res는 JSON 형태로 바뀌어서 조작할 수 있게 할 거야!
- `.then(data => {})` ← JSON 형태로 바뀐 데이터를 data 라는 이름으로 붙일거야



리마인드

GET 요청은, url뒤에 아래와 같이 붙여서 데이터를 가져옵니다.
`http://naver.com?param=value¶m2=value2`

POST 요청은, `data : {}` 에 넣어서 데이터를 가져옵니다.
`data: { param: 'value', param2: 'value2' },`

- success: 성공하면, response 값에 서버의 결과 값을 담아서 함수를 실행한다.



결과가 어떻게 res에 들어가나요? then이 뭐예요?

→ 일단 모르는 느낌이 있어도 받아들이셔야 합니다...! 형태를 암기하세요!
 (대부분의 개발자들도 내부 원리는 코드를 안 뜯어봐서 몰라요.^^;;)

09. Fetch 시작하기 (2)

▼ 1) Fetch 통신의 결과값을 이해해보기

- 위에서 했던 Fetch 통신을 발전시켜볼게요!

```
fetch("http://spartacodingclub.shop/sparta_api/seoulair") // 기본 요청(GET)
.then(res => res.json()) // 요청해서 받은 데이터를 JSON화
.then(data => { // JSON화 한 데이터를 다시 data로 이름짓기
  console.log(data) // 조작할 수 있는 결과물을 가지고 코드에서 작성
})
```

- 개발자도구 콘솔에 찍어보기

```
fetch("http://spartacodingclub.shop/sparta_api/seoulair") // 기본 요청(GET)
.then(res => res.json()) // 요청해서 받은 데이터를 JSON화
.then(data => { // JSON화 한 데이터를 다시 data로 이름짓기
  console.log(data) // 조작할 수 있는 결과물을 가지고 코드에서 작성

  // 값 중 도봉구의 미세먼지 값만 가져와보기
  let ["RealtimeCityAir"]["row"][11];
  let gu_name = dobong['MSRSTE_NM'];
  let gu_mise = dobong['IDEX_MVL'];
  console.log(gu_name, gu_mise);
})
```

- 모든 구의 미세먼지 값을 찍어보기

```
fetch("http://spartacodingclub.shop/sparta_api/seoulair") // 기본 요청(GET)
.then(res => res.json()) // 요청해서 받은 데이터를 JSON화
.then(data => { // JSON화 한 데이터를 다시 data로 이름짓기
  console.log(data) // 조작할 수 있는 결과물을 가지고 코드에서 작성

  let mise_list = response["RealtimeCityAir"]["row"];
  mise_list.forEach((i) => {
    let mise = mise_list[i];
    let gu_name = mise["MSRSTE_NM"];
    let gu_mise = mise["IDEX_MVL"];
    console.log(gu_name, gu_mise);
  })
})
```

- ▼ 복습할 때 참고! - 해설) 모든 구의 미세먼지 값을 찍어보기



천천히 따라해볼 수 있도록 복습자료를 첨부해요!

1. 미세먼지 데이터가 어디에 있는지 찾기

```
{
  - RealtimeCityAir: {
    list_total_count: 25,
    - RESULT: {
      CODE: "INFO-000",
      MESSAGE: "정상 처리되었습니다"
    },
    - row: [
      - {
        MSRDT: "202004201400",
        MSRRGN_NM: "도심권",
        MSRSTE_NM: "중구",
        PM10: 17,
        PM25: 14,
        O3: 0.032,
        NO2: 0.018,
        CO: 0.4,
        SO2: 0.003,
        IDEX_NM: "보통",
        IDEX_MVL: 52,
        ARPLT_MAIN: "O3"
      },
      ,
    ],
  },
}
```

위 그림과 같이 RealtimeCityAir > row 에 미세먼지 데이터가 들어있습니다. 이걸 꺼내볼까요?

```
fetch("http://spartacodingclub.shop/sparta_api/seoulair")
  .then(res => res.json())
  .then(data => {
    let mise_list = response["RealtimeCityAir"]["row"]; // 꺼내는 부분!
    console.log(mise_list);
  })
```

2. 반복문으로 구 데이터를 출력해보기

```
- row: [
  - {
    MSRDT: "202004201400",
    MSRRGN_NM: "도심권",
    MSRSTE_NM: "중구",
    PM10: 17,
    PM25: 14,
    O3: 0.032,
    NO2: 0.018,
    CO: 0.4,
    SO2: 0.003,
    IDEX_NM: "보통",
    IDEX_MVL: 52,
    ARPLT_MAIN: "O3"
  },
  - {
    MSRDT: "202004201400",
    MSRRGN_NM: "도심권",
    MSRSTE_NM: "중로구",
  },
]
```

row의 값을 mise_list에 담았으니, 반복문을 이용해보겠습니다!

```
fetch("http://spartacodingclub.shop/sparta_api/seoulair")
  .then(res => res.json())
  .then(data => {
    let mise_list = response["RealtimeCityAir"]["row"];
    mise_list.forEach((i) => {
      let mise = mise_list[i]
      console.log(mise)
    }) // 반복문을 사용해서 미세먼지 리스트를 개발자도구에 찍어보기!
  })
```

3. 구 데이터에서 구 이름, 미세먼지 수치를 골라내어 출력하기


```
- {
  MSRD: "202004201400",
  MSRRGN_NM: "도심권",
  MSRSTE_NM: "중구",
  PM10: 17,
  PM25: 14,
  O3: 0.032,
  NO2: 0.018,
  CO: 0.4,
  SO2: 0.003,
  IDEX_NM: "보통",
  IDEX_MVL: 52,
  ARPLT_MAIN: "O3"
},
```

구 이름 키 값인 "MSRSTE_NM", 미세먼지 수치 키값인 "IDEX_MVL"의 밸류를 출력

```
fetch("http://spartacodingclub.shop/sparta_api/seoulair")
  .then(res => res.json())
  .then(data => {
    let mise_list = response["RealtimeCityAir"]["row"];
    mise_list.forEach((i) => {
      let mise = mise_list[i];
      let gu_name = mise["MSRSTE_NM"];
      let gu_mise = mise["IDEX_MVL"];
      console.log(gu_name, gu_mise);
    })
  })
```

10. Fetch 연습하기 (1)

- ▼ 1) 서울시 OpenAPI(실시간 미세먼지 상태)를 이용하기

 `prac3.html` 를 만들고 시작합니다!

▼ [코드스니펫] fetch 기본 골격

```
fetch("여기에 URL을 입력").then(res => res.json()).then(data => {
  console.log(data)
})
```

▼ [코드스니펫] 미세먼지 OpenAPI

```
http://spartacodingclub.shop/sparta_api/seoulair
```

▼ [코드스니펫] fetch 퀴즈1

```
<!doctype html>
<html lang="ko">

<head>
  <meta charset="UTF-8">
  <title>미세먼지 API로Fetch 연습하고 가기!</title>

  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

  <style type="text/css">
    div.question-box {
      margin: 10px 0 20px 0;
    }
  </style>

  <script>
    function q1() {
      // 여기에 코드를 입력하세요
    }
  </script>

</head>
```

```

<body>
  <h1>Fetch 연습하자!</h1>

  <hr/>

  <div class="question-box">
    <h2>1. 서울시 OpenAPI(실시간 미세먼지 상태)를 이용하기</h2>
    <p>모든 구의 미세먼지를 표기해주세요</p>
    <p>업데이트 버튼을 누를 때마다 지웠다 새로 씌여져야 합니다.</p>
    <button onclick="q1()">업데이트</button>
    <ul id="names-q1">
      <li>종구 : 82</li>
      <li>종로구 : 87</li>
      <li>용산구 : 84</li>
      <li>은평구 : 82</li>
    </ul>
  </div>
</body>

</html>

```

▼ [코드스니펫] fetch 퀴즈(보기)

<http://spartacodingclub.shop/ajaxquiz/01>

▼ [코드스니펫] fetch 퀴즈1(완성)

```

<!doctype html>
<html lang="ko">

<head>
  <meta charset="UTF-8">
  <title>미세먼지 API로Fetch 연습하고 가기!</title>

  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

  <style type="text/css">
    div.question-box {
      margin: 10px 0 20px 0;
    }
  </style>

  <script>
    function q1() {
      fetch("http://spartacodingclub.shop/sparta_api/seoulair")
        .then((response) => response.json())
        .then((data) => data['RealtimeCityAir']['row'])
        .then((data) => {
          // console.log(data)
          $('#names-q1').empty()
          data.forEach(element => {
            let sector = element.MSRSTE_NM
            let mise = element.IDEX_MVL
            // console.log(sector, ":", mise)
            let temp_html = `<li>${sector} : ${mise}</li>`
            $('#names-q1').append(temp_html)
          });
        })
    }
  </script>

</head>

<body>
  <h1>Fetch 연습하자!</h1>

  <hr/>

  <div class="question-box">
    <h2>1. 서울시 OpenAPI(실시간 미세먼지 상태)를 이용하기</h2>
    <p>모든 구의 미세먼지를 표기해주세요</p>
    <p>업데이트 버튼을 누를 때마다 지웠다 새로 씌여져야 합니다.</p>
    <button onclick="q1()">업데이트</button>
    <ul id="names-q1">
      <li>종구 : 82</li>
      <li>종로구 : 87</li>
      <li>용산구 : 84</li>
      <li>은평구 : 82</li>
    </ul>
  </div>
</body>

```

```
</html>
```

11. Fetch 연습하기 (2)



[한걸음 더]

미세먼지 수치가 40이상인 곳은 빨강게 보여줄까요?

▼ [코드스니펫] fetch 연습(완성·한걸음더)

```
<!doctype html>
<html lang="ko">

<head>
  <meta charset="UTF-8">
  <title>미세먼지 API로Fetch 연습하고 가기!</title>

  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

  <style type="text/css">
    div.question-box {
      margin: 10px 0 20px 0;
    }
    .red {
      color: red;
    }
  </style>

  <script>
    function q1() {
      fetch("http://spartacodingclub.shop/sparta_api/seoulair")
        .then((response) => response.json())
        .then((data) => data['RealtimeCityAir']['row'])
        .then((data) => {
          // console.log(data)
          $('#names-q1').empty()
          data.forEach(element => {
            let sector = element.MSRSTE_NM
            let mise = element.IDEX_MVL
            // console.log(sector, ":", mise)
            let temp_html = `<li>${sector} : ${mise}</li>`
            if (mise > 40) {
              temp_html = `<li class="red">${sector} : ${mise}</li>`
            }
            $('#names-q1').append(temp_html)
          });
        })
    }
  </script>

</head>

<body>
  <h1>Fetch 연습하자!</h1>

  <hr/>

  <div class="question-box">
    <h2>1. 서울시 OpenAPI(실시간 미세먼지 상태)를 이용하기</h2>
    <p>모든 구의 미세먼지를 표기해주세요</p>
    <p>업데이트 버튼을 누를 때마다 지웠다 새로 써여져야 합니다.</p>
    <button onclick="q1()">업데이트</button>
    <ul id="names-q1">
      <li>종로구 : 82</li>
      <li>중로구 : 87</li>
      <li>용산구 : 84</li>
      <li>은평구 : 82</li>
    </ul>
  </div>
</body>

</html>
```

12. Fetch 퀴즈 (1)

▼ 1) 서울시 OpenAPI(실시간 따릉이 현황)을 이용하기

 `prac4.html` 를 만들고 시작합니다!

▼ [코드스니펫] 따릉이 OpenAPI

`http://spartacodingclub.shop/sparta_api/seoulbike`

▼ [코드스니펫] fetch 퀴즈2

```
<!doctype html>
<html lang="ko">

<head>
  <meta charset="UTF-8">
  <title>Fetch 연습하고 가기!</title>
  <!-- JQuery를 import 합니다 -->
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>

  <style type="text/css">
    div.question-box {
      margin: 10px 0 20px 0;
    }

    table {
      border: 1px solid;
      border-collapse: collapse;
    }

    td,
    th {
      padding: 10px;
      border: 1px solid;
    }
  </style>

  <script>
    function q1() {
      // 여기에 코드를 입력하세요
    }
  </script>
</head>

<body>
  <h1>Fetch 연습하자!</h1>

  <hr />

  <div class="question-box">
    <h2>2. 서울시 OpenAPI(실시간 따릉이 현황)를 이용하기</h2>
    <p>모든 위치의 따릉이 현황을 보여주세요</p>
    <p>업데이트 버튼을 누를 때마다 지웠다 새로 씌여져야 합니다.</p>
    <button onclick="q1()">업데이트</button>
    <table>
      <thead>
        <tr>
          <td>거치대 위치</td>
          <td>거치대 수</td>
          <td>현재 거치된 따릉이 수</td>
        </tr>
      </thead>
      <tbody id="names-q1">
        <tr>
          <td>102. 망원역 1번출구 앞</td>
          <td>22</td>
          <td>0</td>
        </tr>
        <tr>
          <td>103. 망원역 2번출구 앞</td>
          <td>16</td>
          <td>0</td>
        </tr>
        <tr>
          <td>104. 합정역 1번출구 앞</td>
          <td>16</td>
          <td>0</td>
        </tr>
      </tbody>
    </table>
  </div>
</body>
</html>
```

```

        </tr>
      </tbody>
    </table>
  </div>
</body>

</html>

```

▼ [코드스니펫] fetch 퀴즈2(보기1)

http://spartacodingclub.shop/ajaxquiz/02_1

▼ [코드스니펫] fetch 퀴즈2(완성1)

```

<!doctype html>
<html lang="ko">

<head>
  <meta charset="UTF-8">
  <title>Fetch 연습하고 가기!</title>
  <!-- JQuery를 import 합니다 -->
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>

  <style type="text/css">
    div.question-box {
      margin: 10px 0 20px 0;
    }

    table {
      border: 1px solid;
      border-collapse: collapse;
    }

    td,
    th {
      padding: 10px;
      border: 1px solid;
    }
  </style>

  <script>
    function q1() {
      // 여기에 코드를 입력하세요
      fetch("http://spartacodingclub.shop/sparta_api/seoulbike")
        .then(response => response.json())
        .then(data => data['getStationList']['row'])
        .then(data => {
          console.log(data)
          $('#names-q1').empty()
          data.forEach(ele => {
            let sector = ele.stationName
            let bike = ele.parkingBikeTotCnt
            let rack = ele.rackTotCnt
            // console.log(sector, rack, bike)

            let temp_html = `<tr>
                                <td>${sector}</td>
                                <td>${bike}</td>
                                <td>${rack}</td>
                              </tr>`
            $('#names-q1').append(temp_html)
          })
        })
    }
  </script>

</head>

<body>
  <h1>Fetch 연습하자!</h1>

  <hr />

  <div class="question-box">
    <h2>2. 서울시 OpenAPI(실시간 따릉이 현황)를 이용하기</h2>
    <p>모든 위치의 따릉이 현황을 보여주세요</p>
    <p>업데이트 버튼을 누를 때마다 지웠다 새로 씌여져야 합니다.</p>
    <button onclick="q1()">업데이트</button>
    <table>
      <thead>

```



```

        <tr>
            <td>거치대 위치</td>
            <td>거치대 수</td>
            <td>현재 거치된 따릉이 수</td>
        </tr>
    </thead>
    <tbody id="names-q1">
        <tr>
            <td>102. 망원역 1번출구 앞</td>
            <td>22</td>
            <td>0</td>
        </tr>
        <tr>
            <td>103. 망원역 2번출구 앞</td>
            <td>16</td>
            <td>0</td>
        </tr>
        <tr>
            <td>104. 합정역 1번출구 앞</td>
            <td>16</td>
            <td>0</td>
        </tr>
    </tbody>
</table>
</div>
</body>
</html>

```

13. Fetch 퀴즈 (2)



[한걸음 더]

따릉이 대수가 5대 미만인 곳은 빨강게 보여주면 어떨까요?

▼ [코드스니펫] fetch 퀴즈2(보기2)

http://spartacodingclub.shop/ajaxquiz/02_2

▼ [코드스니펫] fetch 퀴즈2(완성2)

```

<!doctype html>
<html lang="ko">

<head>
    <meta charset="UTF-8">
    <title>Fetch 연습하고 가기!</title>
    <!-- JQuery를 import 합니다 -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>

    <style type="text/css">
        div.question-box {
            margin: 10px 0 20px 0;
        }

        table {
            border: 1px solid;
            border-collapse: collapse;
        }

        td,
        th {
            padding: 10px;
            border: 1px solid;
        }

        .red {
            color: red;
        }
    </style>

    <script>
        function q1() {
            // 여기에 코드를 입력하세요
            fetch("http://spartacodingclub.shop/sparta_api/seoulbike")

```

```

        .then(response => response.json())
        .then(data => data['getStationList']['row'])
        .then(data => {
            console.log(data)
            $('#names-q1').empty()
            data.forEach(ele => {
                let sector = ele.stationName
                let bike = ele.parkingBikeTotCnt
                let rack = ele.rackTotCnt
                // console.log(sector, rack, bike)

                let temp_html = `<tr>
                    <td>${sector}</td>
                    <td>${bike}</td>
                    <td>${rack}</td>
                </tr>`

                if (bike < 5) {
                    temp_html = `<tr class="red">
                        <td>${sector}</td>
                        <td>${bike}</td>
                        <td>${rack}</td>
                    </tr>`
                }
                $('#names-q1').append(temp_html)
            })
        })
    }
</script>

</head>

<body>
    <h1>Fetch 연습하자!</h1>

    <hr />

    <div class="question-box">
        <h2>2. 서울시 OpenAPI(실시간 따릉이 현황)를 이용하기</h2>
        <p>모든 위치의 따릉이 현황을 보여주세요</p>
        <p>업데이트 버튼을 누를 때마다 지웠다가 새로 써져야 합니다.</p>
        <button onclick="q1()">업데이트</button>
        <table>
            <thead>
                <tr>
                    <td>거치대 위치</td>
                    <td>거치대 수</td>
                    <td>현재 거치된 따릉이 수</td>
                </tr>
            </thead>
            <tbody id="names-q1">
                <tr>
                    <td>102. 망원역 1번출구 앞</td>
                    <td>22</td>
                    <td>0</td>
                </tr>
                <tr>
                    <td>103. 망원역 2번출구 앞</td>
                    <td>16</td>
                    <td>0</td>
                </tr>
                <tr>
                    <td>104. 합정역 1번출구 앞</td>
                    <td>16</td>
                    <td>0</td>
                </tr>
            </tbody>
        </table>
    </div>
</body>

</html>

```

14. 2주차 끝 & 숙제 설명



지난 주차에 만들었던 스파르타피디이에 실시간 서울 날씨 API 를 적용해봅시다!

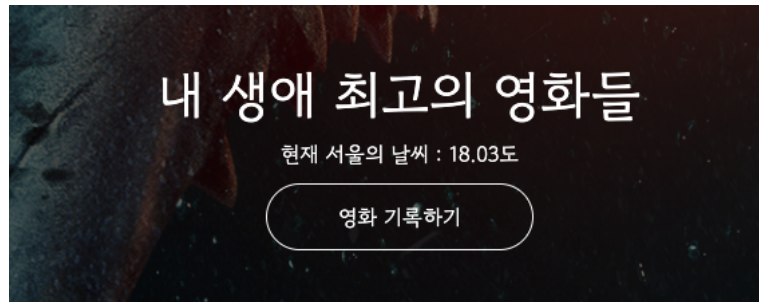
▼ [코드스니펫] 자동 로딩 함수 & 날씨 API

```
$(document).ready(function() {
  fetch("http://spartacodingclub.shop/sparta_api/weather/seoul").then(res => res.json()).then(data => {
    console.log(data)
  })
})
```

▼ [코드스니펫] 추가 HTML

```
<div>현재 서울의 날씨 : <span id="temp">20</span>도</div>
```

▼ 이렇게 만들면 완료!



▼ 힌트!

- **내 생애 최고의 영화들** 제목과 **영화 기록하기** 버튼 사이에 뼈대를 하나 만들어보세요!
- fetch를 사용해 값을 가져와보세요!
- 백틱을 사용해, 뼈대/문자/변수를 모두 섞어서 붙여보세요!

HW. 2주차 숙제 해설

▼ [코드스니펫] 2주차 숙제 답안 코드

```
<!doctype html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet"
    integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"
    integrity="sha384-Mrcw6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
    crossorigin="anonymous"></script>

  <title>스파르타 피디아</title>

  <link href="https://fonts.googleapis.com/css2?family=Gowun+Dodum&display=swap" rel="stylesheet">
  <script src="prac.js"></script>
  <script src="mise.js"></script>

  <style>
    * {
      font-family: 'Gowun Dodum', sans-serif;
    }

    .mytitle {
      width: 100%;
      height: 250px;

      background-image: linear-gradient(0deg, rgba(0, 0, 0, 0.5), rgba(0, 0, 0, 0.5)), url('https://movie-phinf.pstatic.net/

      color: white;

      display: flex;
```

```

        flex-direction: column;
        align-items: center;
        justify-content: center;
    }

    .mytitle > button {
        width: 200px;
        height: 50px;

        background-color: transparent;
        color: white;

        border-radius: 50px;
        border: 1px solid white;

        margin-top: 10px;
    }

    .mytitle > button:hover {
        border: 2px solid white;
    }

    .mycomment {
        color: gray;
    }

    .mycards {
        margin: 20px auto 0px auto;
        width: 95%;
        max-width: 1200px;
    }

    .mypost {
        width: 95%;
        max-width: 500px;
        margin: 20px auto 0px auto;
        padding: 20px;
        box-shadow: 0px 0px 3px 0px gray;
    }

    .mybtns {
        display: flex;
        flex-direction: row;
        align-items: center;
        justify-content: center;

        margin-top: 20px;
    }
    .mybtns > button {
        margin-right: 10px;
    }
</style>
<script>
    $(document).ready(function() {
        fetch("http://spartacodingclub.shop/sparta_api/weather/seoul")
            .then(res => res.json())
            .then(data => {
                // console.log(data)
                let temp = data['temp']
                $('#temp').text(temp)
            })
    })
</script>
</head>

<body>
<div class="mytitle">
    <h1>내 생애 최고의 영화들</h1>
    <div>현재 서울의 날씨 : <span id="temp">20</span>도</div>
    <button>영화 기록하기</button>
</div>
<div class="mypost">
    <div class="form-floating mb-3">
        <input type="email" class="form-control" id="floatingInput" placeholder="name@example.com">
        <label for="floatingInput">영화URL</label>
    </div>
    <div class="input-group mb-3">
        <label class="input-group-text" for="inputGroupSelect01">별점</label>
        <select class="form-select" id="inputGroupSelect01">
            <option selected>-- 선택하기 --</option>
            <option value="1">*</option>
            <option value="2">*</option>
            <option value="3">*</option>
            <option value="4">*</option>
            <option value="5">*</option>
        </select>
    </div>
</div>

```

```

<div class="form-floating">
  <textarea class="form-control" placeholder="Leave a comment here" id="floatingTextarea2"
    style="height: 100px"></textarea>
  <label for="floatingTextarea2">코멘트</label>
</div>
<div class="mybtns">
  <button type="button" class="btn btn-dark">기록하기</button>
  <button type="button" class="btn btn-outline-dark">닫기</button>
</div>
</div>
<div class="mycards">
  <div class="row row-cols-1 row-cols-md-4 g-4">
    <div class="col">
      <div class="card h-100">
        
        <div class="card-body">
          <h5 class="card-title">영화 제목이 들어갑니다</h5>
          <p class="card-text">여기에 영화에 대한 설명이 들어갑니다.</p>
          <p>***</p>
          <p class="mycomment">나의 한줄 평을 씁니다</p>
        </div>
      </div>
    </div>
    <div class="col">
      <div class="card h-100">
        
        <div class="card-body">
          <h5 class="card-title">영화 제목이 들어갑니다</h5>
          <p class="card-text">여기에 영화에 대한 설명이 들어갑니다.</p>
          <p>***</p>
          <p class="mycomment">나의 한줄 평을 씁니다</p>
        </div>
      </div>
    </div>
    <div class="col">
      <div class="card h-100">
        
        <div class="card-body">
          <h5 class="card-title">영화 제목이 들어갑니다</h5>
          <p class="card-text">여기에 영화에 대한 설명이 들어갑니다.</p>
          <p>***</p>
          <p class="mycomment">나의 한줄 평을 씁니다</p>
        </div>
      </div>
    </div>
    <div class="col">
      <div class="card h-100">
        
        <div class="card-body">
          <h5 class="card-title">영화 제목이 들어갑니다</h5>
          <p class="card-text">여기에 영화에 대한 설명이 들어갑니다.</p>
          <p>***</p>
          <p class="mycomment">나의 한줄 평을 씁니다</p>
        </div>
      </div>
    </div>
  </div>
</div>
</body>

</html>

```

◀ 이전 주차

■ [스파르타코딩클럽] 웹개발 종합반 - 2주차

다음 주차 ▶

■ [스파르타코딩클럽] 웹개발 종합반 - 3주차

Copyright © TeamSparta All rights reserved.