



# 스파르타코딩클럽 2주차



매 주차 강의자료 시작에 PDF파일과 영상 링크를 올려두었어요!

## ▼ PDF 강의자료 다운받기

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/d5279eda-f3db-4df1-a60d-a204b229921c/week02.pdf>

## ▼ 영상강의 참고하기

- [2주차 복습용 영상강의 링크](#)



모든 토글을 열고 닫는 단축키

Windows : **Ctrl + alt + t**

Mac : **⌘ + option + t**

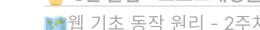
## 목차



수업 목표



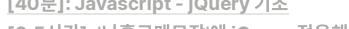
체크인



배우고 적용하기 - 전반부



5분 꿀팁 - 프로그래밍을 한다는 것은?



[0.5시간] Javascript - 기초 문법 (조건문, 반복문)

[0.5시간] Javascript - 패턴 연습하기

[40분]: Javascript - jQuery 기초

[0.5시간]: '나홀로메모장'에 jQuery 적용해보기

[20분]: Javascript + jQuery 연습하기



문제뱅크



배우고 적용하기 - 후반부



5분 꿀팁 - 프로그래밍은 문제 해결#1\_구글링

[1시간] Ajax 통신 기초

[20분] Javascript + jQuery + Ajax 함께하기

[40분] Javascript + jQuery + Ajax 연습하기



문제뱅크

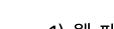


소화 타임

숙제 설명

복습 도우미

[0.5시간] 스스로 소화 타임 시작



체크아웃

[설치] - 다음 시간을 위해 미리 설치해와야 할 것들



## 수업 목표

### 1. 프론트엔드 - Javascript 활용하기

1) 웹 페이지에 동작을 줄 수 있는 Javascript 기초(조건문, 반복문) 연습하기

2) jQuery (Javascript 편하게 쓰기 위한 라이브러리)로 HTML 조작하기

### 2. 나홀로 메모장(링크)에 Javascript 적용하기

### 3. API 사용하기

- 1) 데이터 표현방식 JSON 이해하기
- 2) GET / POST 이해하기
- 3) 비동기 서버 통신방식 Ajax 사용하기

전반 3시간

## ✓ 체크인



튜터님은 체크인과 함께 출석 체크(링크)를 진행해주세요!

스파르타코딩클럽 출석체크

<http://spartacodingclub.shop/attendance>

#### ▼ "15초 체크인"

- 튜터님은 타이머를 띄워주세요! (링크)
- 본인의 감정상태와 오늘 있었던 소소한 일을 공유하는 시간입니다.

## 💡 배우고 적용하기 - 전반부

### 5분 꿀팁 - 프로그래밍을 한다는 것은?

#### ▼ 프로그래밍을 한다는 것

- 프로그래밍(programming)은 '문제를 해결하기 위해 컴퓨터에게 명령을 내리는 것' 입니다.
- 컴퓨터에게 명령을 내리는 것은 외계인蝶과 대화하는 것과 같죠. 외계인蝶들은 지구인의 문화를 모르기 때문에, 특정한 문법을 따라서, 정확히 코드가 적힌대로만 알아듣습니다.
- 단계별로 정확히 적힌대로만 실행하는게 우리가 평소에 이야기하는 거랑 뭐가 다르냐고요? 😊영상(5분 28초)으로 확인하시죠!
  - 📺 THIS "EXACT INSTRUCTIONS CHALLENGE" IS SO HILARIOUS (링크)
  - Youtube 영상 - 한국어 자막으로 보는 방법 : 하단 설정⚙️ - 자막 - 자동번역 - 한국어

<https://youtu.be/Ct-IOOUqmyY>

## 🌐 웹 기초 동작 원리 - 2주차: jQuery, Ajax

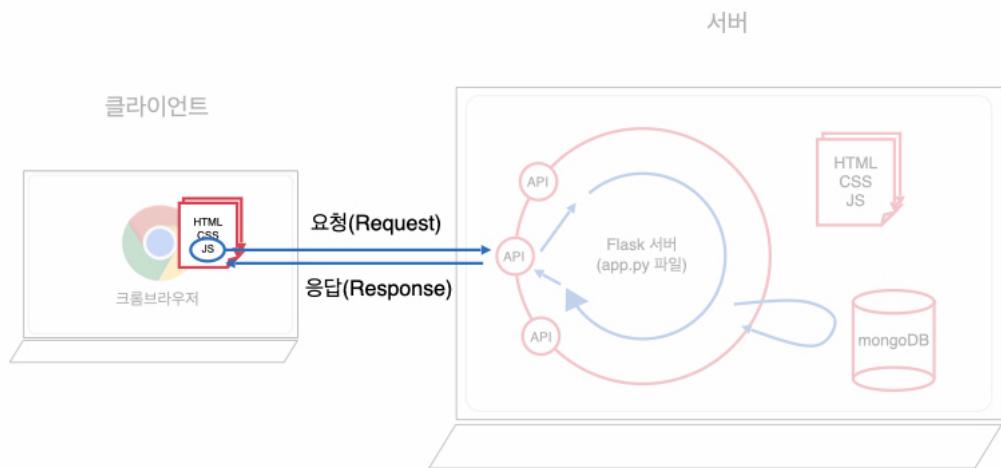
#### ▼ 지난주 복습 - 1주차 개념

- 전반부 주요 키워드
  - HTTP: 웹은 HTTP라는 규약(규칙)을 따릅니다. url에서 `http://` 가 바로 HTTP라는 규약을 따른다는 표시예요.
  - 클라이언트 : HTTP에서 요청을 하는 쪽
  - 서버 : HTTP에서 요청을 받아 응답하는 쪽
  - HTML, CSS, JS(Javascript)는 화면을 구성합니다.
  - HTML은 뼈대, CSS는 예쁘게 꾸미는 디자인을 담당합니다
- 후반부 주요 키워드
  - 부트스트랩(Bootstrap): 웹사이트 화면을 쉽게 만들 수 있도록 도와주는 일종의 디자인 키트
  - JS(JavaScript, 자바스크립트): 프로그래밍 언어로, 웹 브라우저가 해석할 수 있는 언어입니다. JS의 더 많은 기능은 2주차에 배웁니다.
  - JS 프로그래밍 언어 문법 : 변수, 자료형, 함수(요기까지 배웠습니다), 조건문, 반복문(오늘 배웁니다)



오늘은 Javascript로 서버에 데이터를 요청하고, 서버가 응답한 데이터를 받아 조작하는 방법을 배워볼거예요(응답해서 보내준 데이터는 HTML파일이라고 가정).

jQuery를 이용해 Javascript로 HTML을 쉽게 제어하고,  
Ajax를 이용해 서버에 데이터를 요청하고 받아보겠습니다.



## [0.5시간] Javascript - 기초 문법 (조건문, 반복문)

### ▼ 1) 지난주 복습 - 1주차 Javascript 기초 문법

- **변수 & 기본연산**, **리스트 & 딕셔너리**, **기본 함수 사용법**과 **함수**에 대해서 배웠습니다. 아래 코드를 이해하면 복습 끝! 이해가지 않는 부분은 찾아서 복습해보세요(1주차 수업자료 - 19)).

```
// 두 숫자를 입력받으면 더한 결과를 돌려주는(return) 함수
function sum(num1, num2) {
    console.log('num1: ', num1, ', num2: ', num2);
    return num1 + num2;
}

let temp;
let result = [];

temp = sum(3, 5);
console.log(temp); // 8

result.push(temp);
console.log(result[0]); // 8
```

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

let emainInfo = myEmail.split('@') // ['sparta','gmail.com']

console.log(emainInfo[0]) // sparta
console.log(emainInfo[1]) // gmail.com

let domain = emainInfo[1].split('.') // ['gmail','com']

console.log(domain[0]) // gmail -> 우리가 알고 싶었던 것!
console.log(domain[1]) // com

let str = myEmail.split('@')[1].split('.')[0] // gmail -> 간단하게 쓸 수도 있다!
console.log(str);
```



### 【👏튜터와 같이】 표시 잊지 않으셨죠?

튜터와 튜티 함께 한 줄, 한 줄씩 코드를 따라치면서 실습을 해볼 거예요.

#### ▼ 2) 【👏튜터와 같이】 조건문 & 반복문을 배워볼까요?

##### ▼ [크롬 개발자도구]를 열어서, console 탭에 작성

- 💻 단축키! Windows: F12 또는 alt + ctrl + i / Mac : option + command + i



오늘도 출력을 사용해서 값을 확인하며 배우겠습니다! `console.log(변수)`

'출력하기'는 값을 확인하거나, 에러를 찾을 때 자주 쓰입니다. (출력해서 버그찾기 - Debugging By Printing 스클)

##### ▼ 조건문

- 90보다 작으면 작다고, 크면 크다고 알려주는 함수

```
function compareNinety(num){  
    if (num > 90) {  
        alert('90보다 커요!')  
    } else {  
        alert('90보다 작아요!')  
    }  
  
    compareNinety(30);
```

- 응용 - 다음 함수의 기능을 생각해볼까요?

```
// 함수를 정의하기  
function isAdult(person) {  
    if (person['age'] > 19){  
        return true;  
    } else {  
        return false;  
    }  
  
// 함수를 사용하기  
let person = {'name':'Harry', 'age':24}  
let result = isAdult(person);  
console.log(result) // true
```

- AND 조건과 OR 조건!

```
// AND 조건 : 모든 조건들이 true여야 true  
// 나이가 20과 같거나 크고 30보다 작은 경우 true  
function isTwenty(person) {  
    if (person['age'] >= 20 && person['age'] < 30){  
        return true;  
    } else {  
        return false;  
    }  
  
// OR 조건 : 조건 중 하나라도 true이면, true!  
// 나이가 20보다 작거나, 30보다 크면 true  
function isNotTewnty(person) {  
    if (person['age'] < 20 || person['age'] > 30){  
        return true;  
    } else {  
        return false;  
    }  
  
    console.log(person); // person에 값이 있는지 확인하기  
  
    result = isTwenty(person);  
    console.log(result);  
  
    console.log(isNotTewnty(person));
```

- if, else if, else if else if else

```
// 조건을 여러 개 수행하고 싶을 때
function checkGeneration(age){
    if (age > 120) {
        alert('와 19세기에 태어나셨군요!')
    } else if (age >= 80) {
        alert('80세 이상! 인생은 여든부터!')
    } else {
        alert('젊으시군요! 장래희망이 뭔가요?')
    }
}

let myAge = 55;
checkGeneration(myAge);
```

## ▼ 반복문

- 예를 들어, 0부터 99까지 출력해야 하는 상황이라면!

```
console.log(0)
console.log(1)
console.log(2)
console.log(3)
console.log(4)
console.log(5)
...
console.log(99)

// 이렇게 백 줄을 쓰기엔 무리가 있겠죠? 그래서, 반복문이라는 것이 존재합니다!
```

- 반복문을 이용하면 아래와 같이 단 세 줄로, 출력할 수 있습니다. 와우! 😊

```
for (let i = 0; i < 100; i++) {
    console.log(i);
}
```

## ▼ 규칙을 살펴볼까요?

```
for (A_시작조건; B_반복조건; C_매 반복 후에 할 일) {
    D_매 반복마다 실행될 명령어들
}
/*
[실행 순서]
A 로 초기화해서 시작 (i = 0)
-> B가 true 이면 (i < 100) -> D 실행 -> C 실행 (i + 1, i = 1)
-> B가 true 이면 (i < 100) -> D 실행 -> C 실행 (i = 2)
-> B가 true 이면 (i < 100) -> D 실행 -> C 실행 (i = 3)
-> B가 true 이면 (i < 100) -> D 실행 -> C 실행 (i = 4)
...
-> B가 true 이면 (i < 100) -> D 실행 -> C 실행 (i = 99)
-> B가 true 이면 (i < 100) -> D 실행 -> C 실행 (i = 100)
-> B가 false이면 (i가 100보다 작지 않음!) -> (더 이상 실행하지 않고) 종료
*/
```

- i는 반복문에서 횟수를 세어 준다고 해서 '카운터 변수(counter variable)'라고 합니다.
- C에 적혀 있는 조건인 `i++`는 실행 후, i를 +1 한다는 뜻입니다. 즉, `i = i + 1`
- 패턴 ! 반복문은 리스트와 단짝! 리스트와 함께 자주 쓰입니다.
- [크롬 개발자 도구] console 창에서 확인! 함께 코딩해볼게요.

```
let wizards = ['덤블도어', '맥고나걸', '스네이프', '해리', '허마이오니', '론']

// 리스트의 length 기억나시죠? 리스트의 길이!
// i가 1씩 증가하면서, 마법사를 wizards의 원소를 차례대로 불러올 수 있게 됩니다.
// 이렇게 하면 리스트의 모든 원소를 한번에 출력할 수 있겠죠?
for (let i = 0; i < wizards.length ; i++) {
    console.log(wizards[i])
}
```

- 리스트도 그냥 리스트가 아닙니다! 딕셔너리가 들어간 리스트와 찰떡이죠

- [크롬 개발자 도구] console 창에서 확인! 함께 코딩해볼게요.

```
let wizardsInfo = [
  {'name': '덤블도어', 'age': 116},
  {'name': '맥고나걸', 'age': 85},
  {'name': '스네이프', 'age': 60},
  {'name': '해리', 'age': 40},
  {'name': '허마이오니', 'age': 40},
  {'name': '론', 'age': 40},
]

for (let i = 0; i < wizardsInfo.length; i++) {
  console.log(wizardsInfo[i]);
}

// 이렇게 하면 리스트 내의 딕셔너리를 하나씩 출력할 수 있고,
```

- 조건문과 함께 사용할 수도 있겠죠?

```
for (let i = 0; i < wizardsInfo.length; i++) {
  if (wizardsInfo[i]['age'] < 70) {
    console.log(wizardsInfo[i]['name']);
  }
}

// 이렇게 하면 70세 미만인 마법사의 이름만 출력할 수도 있습니다.
```

## [0.5시간] Javascript - 패턴 연습하기

### ▼ 3) [💡튜터와 같이] 패턴 연습하기

- [크롬 개발자 도구]를 열어서, console 탭에 작성. 새로고침(F5 또는 Ctrl + r) 해서 앞선 내용은 지워주세요!



이번 시간에는 튜터님 설명 먼저 듣고, '패턴'에 익숙해지는데 집중해보세요!  
따라치기 / 외우기 보다 이해에 집중!

#### ▼ (1) 반복문 - 합을 구하는 함수 만들기



0부터 n-1까지 더하는 함수를 만들고 싶다면?

```
function calculateSum(n) {
  let sum = 0
  for (let i = 0; i < n; i++) {
    sum += i;           // sum을 i만큼 증가시켜라. sum = sum + i 와 동일!
  }
  return sum
}

let result = calculateSum(10); // return 결과인 sum이 result에 저장
console.log(result)        // 45를 출력
```

#### ▼ (2) 반복문 & 리스트 - 리스트에서 원소의 갯수를 세기



다음에서 '딸기'는 몇 개일까?

```
let fruitsBasket = ['사과', '감', '감', '배', '포도', '포도', '딸기', '포도', '감', '수박', '딸기']

let count = 0;
for (let i = 0; i < fruitsBasket.length; i++) {
  let fruit = fruitsBasket[i];
  if (fruit == '딸기') {
    count += 1;
  }
}
```

```
}
```

```
console.log(count);
```



이것을 함수로 만들어볼까요? 새로고침(F5)를 누르고, 깔끔하게 다시 해보죠!

```
let fruitsBasket = ['사과', '감', '감', '배', '포도', '포도', '딸기', '포도', '감', '수박', '딸기']

function countFruit(name) {
    let count = 0;
    for (let i = 0; i < fruitsBasket.length; i++) {
        let fruit = fruitsBasket[i];
        if (fruit == name) {
            count += 1;
        }
    }
    return count;
}

// 함수로 만들어두면! 같은 코드를 또 적을 필요가 없죠!
let gamCount = countFruit("감");
console.log(gamCount);

let subakCount = countFruit("수박");
console.log(subakCount);
```

▼ (3) 반복문 & 딕셔너리 - 미세먼지(PM10)의 값이 25 미만인 측정소(구 단위, MSRSTE\_NM)와 미세먼지 수치를 출력하기

- [크롬 개발자도구] console 창 사용하기
- Q. "PM10" 수치가  $25\mu\text{g}/\text{m}^3$ 보다 작은 측정소(MSRSTE\_NM)를 모두 출력해보세요.

▼ A. 정답

- Q. 함수 만들기 - 입력받은 미세먼지 수치보다 작은 측정소(MSRSTE\_NM)를 출력하는 함수를 만들어보세요.

▼ A. 정답

- 제대로 만들었는지 테스트하기

```
showUnderMise(30) // 미세먼지  $40\mu\text{g}/\text{m}^3$ 보다 작은 구만 출력!
showUnderMise(25) // 미세먼지  $35\mu\text{g}/\text{m}^3$ 보다 작은 구만 출력!
```



앗, 문법이 어렵다고요? 처음이라 그렇습니다. 괜찮아요!

지금 단계에서는 처음부터 끝까지 문법을 외우기보다 패턴과 사용방법에 집중해주세요!

예를 들어, if문을 어떻게 썼더라- 하는 것은 괜찮습니다. 다시 찾아서 코드를 복사해서 쓰세요. 지금은 구조와 패턴에 익숙해지는 게 중요합니다! 자꾸자꾸 쓰면 익숙해집니다.

## [40분]: Javascript - jQuery 기초

▼ 4) jQuery란

- **jQuery**(제이쿼리)는 HTML의 요소들을 조작하고 화면의 움직임을 처리하는 등 편리한 Javascript 코드를 미리 작성해놓은 뭉입니다. 즉, Javascript의 라이브러리(library) 중 하나입니다.

▼ 라이브러리(library)란?

- 라이브러리(library)는 공통으로 사용할 수 있는 특정 기능들의 모음입니다.
- 1주차에서 Javascript를 배울 때, 소문자를 대문자로(`toUpperCase`), 문자열을 특정 문자로 나눠주는 함수(`split`)를 사용했었죠? 바로 JS에서 기본으로 제공하는 라이브러리가 있기 때문에 사용할 수 있었답니다.
- 라이브러리를 사용하면 내가 모든 걸 다 만들지 않아도, 라이브러리에 있는 함수들을 가져다 쓸 수 있겠죠?



Javascript로도 모든 기능(예 - 버튼 클릭 바꾸기 등)을 구현할 수는 있지만,  
1) 코드가 복잡하고, 2) 브라우저 간 호환성 문제도 고려해야해서,  
jQuery라는 라이브러리가 등장하게 되었답니다.

#### ▼ jQuery와 Javascript - 코드 길이 비교해보기

Javascript로 길고 복잡하게 써야 하는 것을

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

jQuery로 보다 직관적으로 쓸 수 있어요. 편리하죠? :-)

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

#### ▼ 5) jQuery 사용하기 - import(임포트) & 요소 가르키기

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



jQuery는 Javascript와 다른 특별한 것이 아니라 미리 작성된 Javascript 코드입니다.

우리는 전문 개발자들이 짜둔 코드를 가져와서 사용하는 것이죠.

그렇게 때문에, 쓰기 전에 "import(임포트)"를 해야합니다!

- <head> 와 </head> 사이에 아래 코드를 넣으면 import 끝!

- 수업 자료를 잘 따라온 분이라면, 나홀로메모장 코드([week01](#) 폴더에 [memo.html](#) 파일)에 이미 아래처럼 적혀있을 거예요.

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

- [3.5.1](#) 이 jQuery 버전입니다. 우리는 부트스트랩과 호환되는 jquery 버전인 3.5.1을 사용할 거예요.

- 참고! jQuery import하기: ([설명 링크](#)). ([공식 CDN 링크](#))

- jQuery를 사용하는 방법



css와 마찬가지로, jQuery를 쓸 때에도 요소를 "가리켜야(select)" → 조작할 수 있습니다.

예) 특정 인풋박스의 값을 → 가져와줘!

예) 특정 div를 → 안보이게 해줘!

css에서는 선택자로 class를 썼지요?

jQuery에서는 id 값을 통해 특정 버튼/인풋박스/div/.. 등을 가리키는게 일반적입니다.

백문이 불여일견! 자주 쓰는 jQuery들을 함께 다뤄보면서 익혀보죠!

#### ▼ 6) [赖以生存] 패턴 연습하기

- 들어가기 전에 - 모든 jQuery를 외워야 하나요?



절대!! 아닙니다. 예를 들어 스크롤 움직임을 감지한다거나, 화면 사이즈를 안다거나, jQuery가 할 수 있는 일은 무척 많습니다.

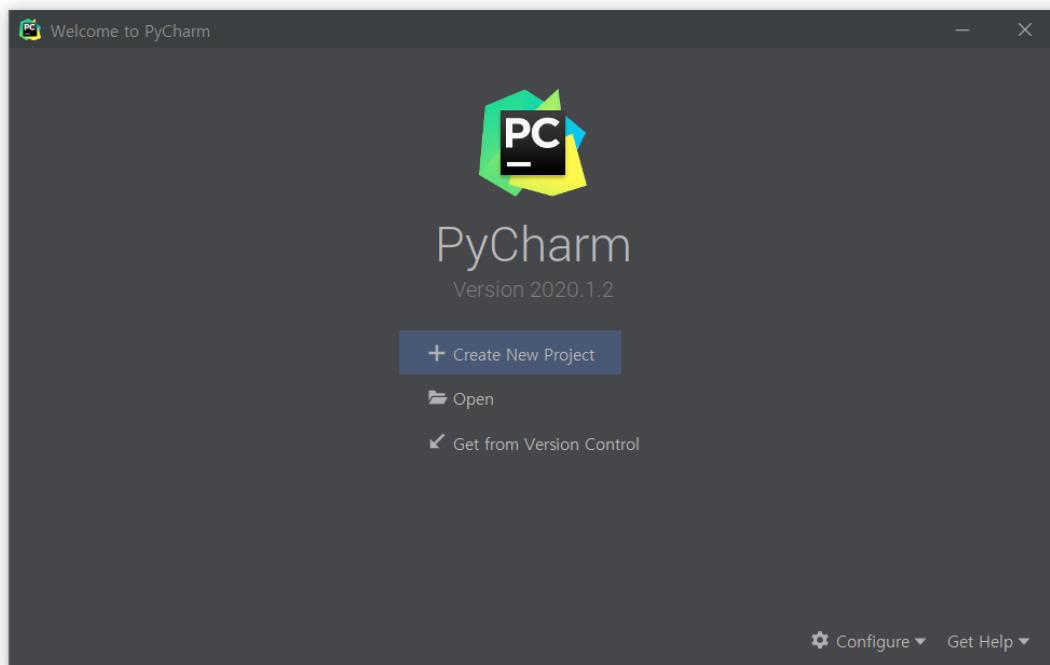
필요할 때에 필요한 기능을 구글링해서 쓰면 됩니다!

#### ▼ 0. 시작하기 - week02 에 [memo.html](#) 파일 만들어두기

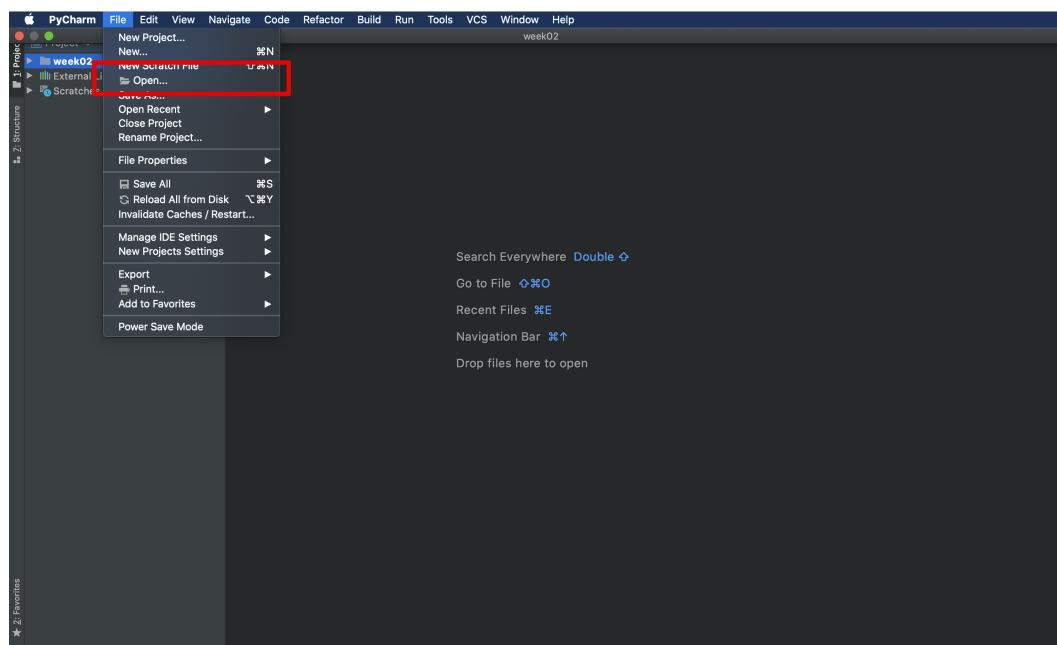
- Pycharm을 열어서, [week02](#) 폴더에 [memo.html](#) 으로 파일을 만들어주세요.

#### ▼ Pycharm에서 week02 폴더 열기

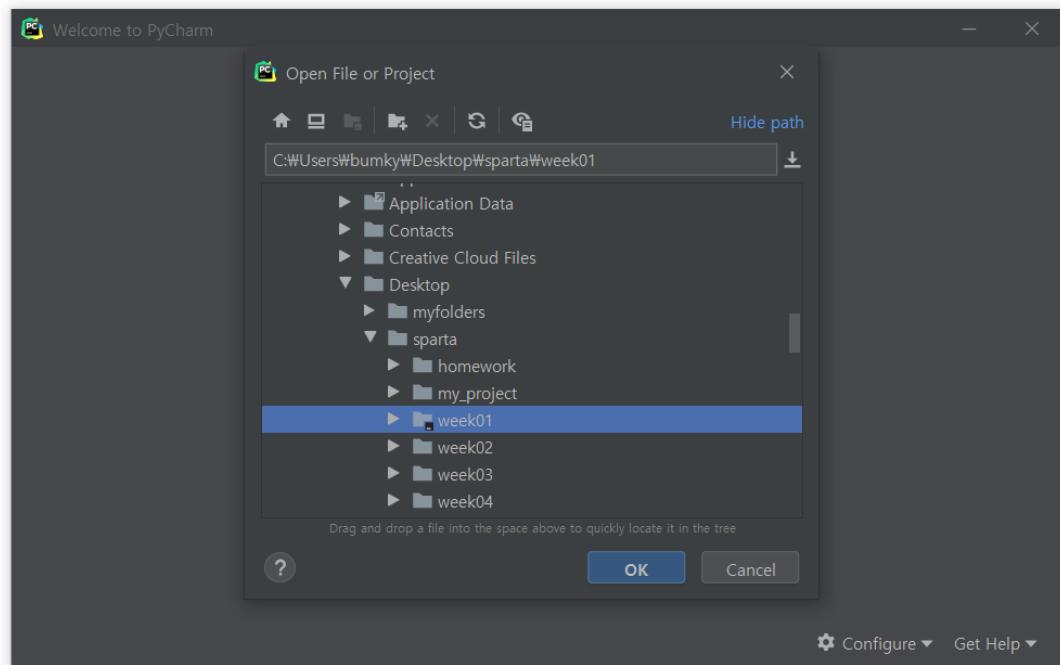
1) Pycharm 화면에서 open 을 클릭



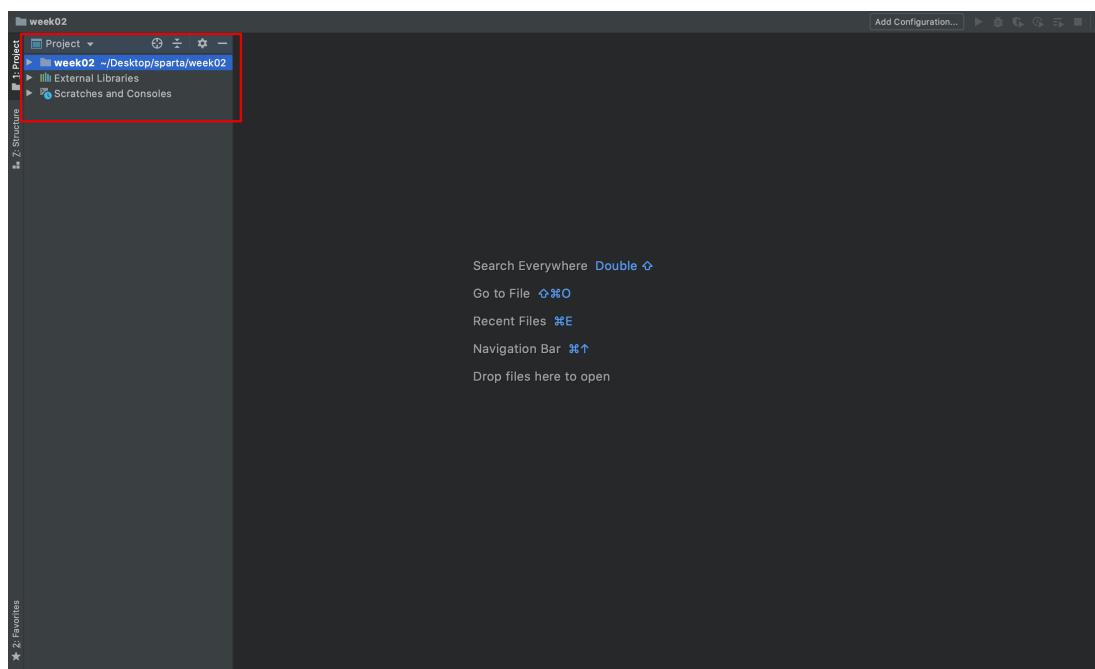
▼ 위 화면이 안 보이면, 상단 File-open 클릭



2) 바탕화면→sparta폴더→week02을 눌러주세요!

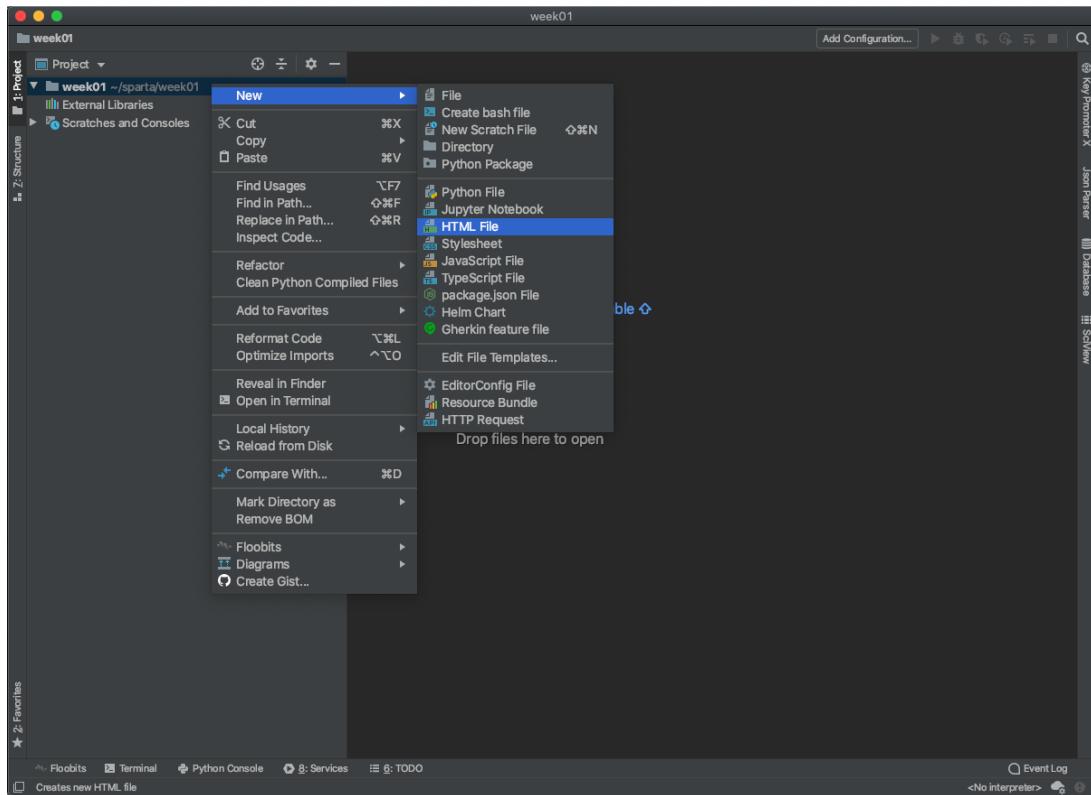


3) 야호! 2주차 준비 완료!



#### ▼ Pycharm에서 HTML 파일 만들기

- 1) 파일을 생성할 폴더(week02)를 마우스 클릭하고 -
- 2) 우클릭 - New - HTML File 클릭하고 이름을 입력하면 아래처럼 파일이 자동 생성됩니다.



- 아래 코드블록에 커서를 두고 전체 선택(ctrl + A 또는 cmd + A) - 복사(ctrl + C 또는 cmd + C) - 붙여넣기 (ctrl+V 또는 cmd + V)

### ▼ 코드!

```

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

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

    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
        integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJ1SAwiGFAW/dAiS6JXm" crossorigin="anonymous"

    <!-- JS -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.12.9/umd/popper.min.js"
        integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPsvXusvfa0b4Q"
        crossorigin="anonymous"></script>

    <!-- 구글폰트 -->
    <link href="https://fonts.googleapis.com/css?family=Stylish&display=swap" rel="stylesheet">

    <title>스파르타코딩클럽 | 나홀로 메모장</title>

    <!-- style -->
    <style type="text/css">
        * {
            font-family: 'Stylish', sans-serif;
        }

        .wrap {
            width: 900px;
            margin: auto;
        }

        .comment {
            color: blue;
            font-weight: bold;
        }

        #post-box {
    
```

```

        width: 500px;
        margin: 20px auto;
        padding: 50px;
        border: black solid;
        border-radius: 5px;
    }

```

```

</style>
<script>
    function openClose() {
        // 여기에 jQuery를 이용해 코드를 짤 예정
    }
</script>

```

```

</head>

<body>
<div class="wrap">
    <div class="jumbotron">
        <h1 class="display-4">나홀로 링크 메모장!</h1>
        <p class="lead">중요한 링크를 저장해두고, 나중에 볼 수 있는 공간입니다</p>
        <hr class="my-4">
        <p class="lead">
            <button onclick="openClose()" id="btn-post-box" type="button" class="btn btn-primary">포스팅 박스 열기</button>
        </p>
    </div>
    <div id="post-box" class="form-post">
        <div>
            <div class="form-group">
                <label for="post-url">아티클 URL</label>
                <input id="post-url" class="form-control" placeholder="">
            </div>
            <div class="form-group">
                <label for="post-comment">간단 코멘트</label>
                <textarea class="form-control" rows="2"></textarea>
            </div>
            <button type="button" class="btn btn-primary">기사저장</button>
        </div>
    </div>
    <div id="cards-box" class="card-columns">
        <div class="card">
            
            <div class="card-body">
                <a href="#" class="card-title">여기 기사 제목이 들어가죠</a>
                <p class="card-text">기사의 요약 내용이 들어갑니다. 동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라만세 무궁화 삼
                <p class="card-text comment">여기에서 코멘트가 들어갑니다.</p>
            </div>
        </div>
        <div class="card">
            
            <div class="card-body">
                <a href="#" class="card-title">여기 기사 제목이 들어가죠</a>
                <p class="card-text">기사의 요약 내용이 들어갑니다. 동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라만세 무궁화 삼
                <p class="card-text comment">여기에서 코멘트가 들어갑니다.</p>
            </div>
        </div>
        <div class="card">
            
            <div class="card-body">
                <a href="#" class="card-title">여기 기사 제목이 들어가죠</a>
                <p class="card-text">기사의 요약 내용이 들어갑니다. 동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라만세 무궁화 삼
                <p class="card-text comment">여기에서 코멘트가 들어갑니다.</p>
            </div>
        </div>
        <div class="card">
            
            <div class="card-body">
                <a href="#" class="card-title">여기 기사 제목이 들어가죠</a>
                <p class="card-text">기사의 요약 내용이 들어갑니다. 동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라만세 무궁화 삼
                <p class="card-text comment">여기에서 코멘트가 들어갑니다.</p>
            </div>
        </div>
    </div>

```

```

        </div>
    </div>
    <div class="card">
        
        <div class="card-body">
            <a href="#" class="card-title">여기 기사 제목이 들어가죠</a>
            <p class="card-text">기사의 요약 내용이 들어갑니다. 동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라만세 무궁화 삼&lt;&gt;
            <p class="card-text comment">여기애 코멘트가 들어갑니다.</p>
        </div>
    </div>
</div>
</body>

</html>

```

### ▼ 1. input 박스의 값을 가져오기

- memo.html에서 형광펜 표시된 부분을 주목하세요! (memo.html 내에서 'post-url' 검색 - ctrl + F 또는 cmd + F)

```

<div class="form-post">
    <div>
        <div class="form-group">
            <label for="post-url">아티클 URL</label>
            <input id="post-url" class="form-control" placeholder="">
        </div>
        <div class="form-group">
            <label for="post-comment">간단 코멘트</label>
            <textarea class="form-control" rows="2"></textarea>
        </div>
        <button type="button" class="btn btn-primary">기사저장</button>
    </div>
</div>

```

- [크롬 개발자도구]창 console 탭에서 아래 코드치기
- 화면의 아티클 URL에 www.spartacodingclub.shop을 입력해보세요.

```
// id 값이 post-url인 곳을 가리키고, val()로 값을 가져온다.
$('#post-url').val(); // www.spartacodingclub.shop 이 뜨네요.
```

### ▼ 2. div 요소 보이기 / 숨기기

- memo.html에서 형광펜 표시된 부분을 주목하세요! (memo.html 내에서 'post-box' 검색 - ctrl + F 또는 cmd + F)

```

<div id="post-box" class="form-post">
    <div>
        <div class="form-group">
            <label for="post-url">아티클 URL</label>
            <input id="post-url" class="form-control" placeholder="">
        </div>
        <div class="form-group">
            <label for="post-comment">간단 코멘트</label>
            <textarea class="form-control" rows="2"></textarea>
        </div>
        <button type="button" class="btn btn-primary">기사저장</button>
    </div>
</div>

```

```
// id 값이 post-box인 곳을 가리키고, hide()로 안보이게 한다.
$('#post-box').hide();

// show()로 보이게 한다. (=css의 display 값을 block으로 바꾼다)
$('#post-box').show();
```

### ▼ 3. css의 값 가져와보기 (우리는 display 속성 값을 가져와볼게요!)

- [크롬 개발자도구]창 console 탭에서 아래 코드치기
- \$('#post-box').css('display') 했을 때, 출력값이 어떻게 보이나요?

```

$('#post-box').hide();
$('#post-box').css('display'); // 출력값: none
// hide()를 하면, css의 display 값이 none으로 바꾼다.
// id=post-box 요소의 css display 값이 none 이라서 눈에 안보였군요!

$('#post-box').show();
$('#post-box').css('display'); // 출력값: block
// show()를 하면, css의 display 값을 원래 값이 있던 block으로 바꾼다.
// id=post-box 요소의 css display 값이 block 이라서 눈에 잘 보였군요!

```

#### ▼ 4. 태그 내 텍스트 입력하기

- [크롬 개발자도구]창 console 탭에서 아래 코드치기

##### 1) input box의 경우

```
$('#post-url').val('여기에 텍스트를 입력하면!');
```

##### 2) 그 외 - 예) 버튼의 텍스트 바꾸기

- `memo.html` 에서 형광펜 표시된 부분을 주목하세요! (`memo.html` 내에서 'btn-posting-box' 검색 - **ctrl + F** 또는 cmd + F)

```
// 가리키고 싶은 버튼에 id 값을 준다음
<button id="btn-post-box" type="button" class="btn btn-primary">포스팅 박스 열기</button>
```

```
$('#btn-post-box').text('포스팅 박스 닫기');
```

#### ▼ 5. 특정 HTML 요소 내 동적으로 요소(element) 만들기

- 포스팅하기 버튼을 누르면(동적으로) → `<div> ~ </div>` 안에 카드를 추가(요소 element 만들기)
- `memo.html` 에 아래 위치에 `id="cards-box"` 를 입력해 id 값을 주겠습니다. (`memo.html` 내에서 'card-post' 검색 - **ctrl + F** 또는 cmd + F 한 후, id 값 입력)

```
// 요소 element 를 넣고 싶은 태그에 id 값을 준 다음
<div id="cards-box" class="card-columns">
  <div class="card">
    
    <div class="card-body">
      <a href="https://naver.com/" class="card-title">여기 기사 제목이 들어가죠</a>
      <p class="card-text">여기에 기사 내용이 들어가겠죠</p>
      <p class="card-text comment">여기엔 코멘트가 들어갑니다</p>
    </div>
  </div>
</div>
```

- [크롬 개발자도구]창 console 탭에서 아래 코드치기

##### 1) 버튼을 넣어보기

```
let tempHtml = '<button>나는 추가될 버튼이다!</button>';
$('#cards-box').append(tempHtml);
```

##### 2) 버튼 말고, 카드를 넣어보기

```
let imgUrl = 'https://movie-phinf.pstatic.net/20181019_236/1539926790655oHv5z_JPEG/movie_image.jpg';
let linkUrl = 'https://naver.com/';
let title = '해리포터와 마법사의 돌 영화 정보';
let desc = '기사의 요약 내용이 들어갑니다. 해리 포터(다니엘 래드클리프 분)는 위압적인 버는 속부(리차드 그리피스 분)와 냉담한 이모 페루니아 (...)';
let comment = '여기에 코멘트가 들어갑니다.';

// 주의: 훌따옴표(')가 아닌 backtick(`)으로 감싸야 합니다.
// Windows: 숫자 1번 키 왼쪽의 버튼을 누르면 backtick(`)이 입력됩니다.
// Mac : 영문으로 전환 후, ⇧ 키를 누르면 됩니다.
// backtick을 사용하면 문자 중간에 Javascript 변수를 삽입할 수 있습니다.
let tempHtml = `<div class="card">
```

```

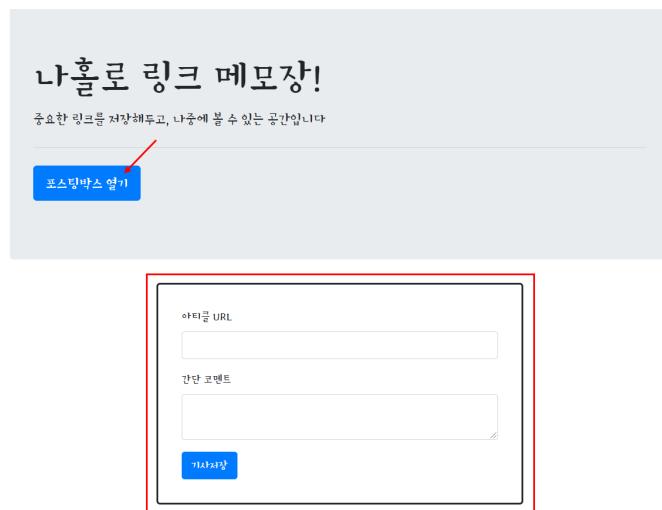
<div class="card-body">
    <a href="${linkUrl}" class="card-title">${title}</a>
    <p class="card-text">${desc}</p>
    <p class="card-text comment">${comment}</p>
</div>
</div>';
$('#cards-box').append(tempHtml);
```

## [0.5시간]: '나홀로메모장'에 jQuery 적용해보기

- ▼ 7) '나홀로메모장'에 포스팅박스 열기/닫기 기능을 함께 붙여보기

- #### ▼ (1) 문제 분석 - 완성된 모습 먼저 보기

링크에서 '포스팅박스 열기' 버튼을 누르면 숨겨진 창이 나타나고 버튼의 내용이 '포스팅박스 닫기'로 바뀝니다. 다시 한 번 버튼을 클릭하면 원래 모습으로 돌아갑니다.



- 6) 에서 만든 `memo.html` 를 계속 사용하겠습니다.
  - ▼  코드! 코드가 없는 사람은 톤을 안 코드 사용해서 시작하세요.

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

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

    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
          integrity="sha384-Gn5384xqQiaowXA+058RXPxPg6fy4IWvTNh0E263XmFcJ1SAwiGgFAW/dAiS6JXm" crossorigin="anonymous"

    <!-- JS -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.2/dist/umd/popper.min.js"
           integrity="sha384-A9yLtXLZGtD1QHwWtOgPQhWxVJZzCZKUkXN0kYDQmZDdGhH3jwM0OQ&lt;!-- crossorigin="anonymous"--></script>

    <!-- 구글폰트 -->
    <link href="https://fonts.googleapis.com/css?family=Stylish&display=swap" rel="stylesheet">

<title>스파르타코딩클럽 | 나홀로 메모장</title>

<!-- style -->
<style type="text/css">
    * {
        font-family: 'Stylish', sans-serif;
    }
</style>
```

```

.wrap {
    width: 900px;
    margin: auto;
}

.comment {
    color: blue;
    font-weight: bold;
}

#post-box {
    width: 500px;
    margin: 20px auto;
    padding: 50px;
    border: black solid;
    border-radius: 5px;
}
</style>
<script>
    function openClose() {
        // 여기에 jQuery를 이용해 코드를 짤 예정
    }
</script>

</head>

<body>
<div class="wrap">
    <div class="jumbotron">
        <h1 class="display-4">나홀로 링크 메모장!</h1>
        <p class="lead">중요한 링크를 저장해두고, 나중에 볼 수 있는 공간입니다</p>
        <hr class="my-4">
        <p class="lead">
            <button onclick="openClose()" id="btn-post-box" type="button" class="btn btn-primary">포스팅 박스 열기</but
        </p>
    </div>
    <div id="post-box" class="form-post">
        <div>
            <div class="form-group">
                <label for="post-url">아티클 URL</label>
                <input id="post-url" class="form-control" placeholder="">
            </div>
            <div class="form-group">
                <label for="post-comment">간단 코멘트</label>
                <textarea class="form-control" rows="2"></textarea>
            </div>
            <button type="button" class="btn btn-primary">기사저장</button>
        </div>
    </div>
    <div id="cards-box" class="card-columns">
        <div class="card">
            
            <div class="card-body">
                <a href="#" class="card-title">여기 기사 제목이 들어가죠</a>
                <p class="card-text">기사의 요약 내용이 들어갑니다. 동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라만세 무궁화 삼
                <p class="card-text comment">여기에 코멘트가 들어갑니다.</p>
            </div>
        </div>
        <div class="card">
            
            <div class="card-body">
                <a href="#" class="card-title">여기 기사 제목이 들어가죠</a>
                <p class="card-text">기사의 요약 내용이 들어갑니다. 동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라만세 무궁화 삼
                <p class="card-text comment">여기에 코멘트가 들어갑니다.</p>
            </div>
        </div>
        <div class="card">
            
            <div class="card-body">
                <a href="#" class="card-title">여기 기사 제목이 들어가죠</a>
                <p class="card-text">기사의 요약 내용이 들어갑니다. 동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라만세 무궁화 삼
                <p class="card-text comment">여기에 코멘트가 들어갑니다.</p>
            </div>
        </div>
        <div class="card">
            
            <div class="card-body">
                <a href="#" class="card-title">여기 기사 제목이 들어가죠</a>
                <p class="card-text">기사의 요약 내용이 들어갑니다. 동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라만세 무궁화 삼
                <p class="card-text comment">여기에 코멘트가 들어갑니다.</p>
            </div>
        </div>
    </div>
</body>

```

```

        <p class="card-text comment">여기에 코멘트가 들어갑니다.</p>
    </div>
<div class="card">
    
    <div class="card-body">
        <a href="#" class="card-title">여기 기사 제목이 들어가죠</a>
        <p class="card-text">기사의 요약 내용이 들어갑니다. 동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라만세 무궁화 삼
        <p class="card-text comment">여기에 코멘트가 들어갑니다.</p>
    </div>
<div class="card">
    
    <div class="card-body">
        <a href="#" class="card-title">여기 기사 제목이 들어가죠</a>
        <p class="card-text">기사의 요약 내용이 들어갑니다. 동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라만세 무궁화 삼
        <p class="card-text comment">여기에 코멘트가 들어갑니다.</p>
    </div>
</div>
</div>
</body>

</html>

```

#### ▼ (2) '포스팅 박스 열기' 버튼에 JS function 을 달기

- memo.html에서 형광펜 표시된 부분을 주목하세요! (memo.html 내에서 'openClose' 검색 - ctrl + F 또는 cmd + F)
- '포스팅 박스 열기' 버튼을 클릭(onclick)하면 openclose 함수가 실행됩니다.

```

<script>
function openClose() {
    // 여기에 jQuery를 이용해 코드를 짤 예정
}
</script>

// onclick 속성(attribute)을 추가합니다.
<button onclick="openClose()" id="btn-post-box" class="btn btn-primary btn-lg">포스팅 박스 열기</button>

```

#### ▼ (3) 클릭 해서 포스팅 박스를 여닫게 하기

- memo.html에서 형광펜 표시된 부분을 주목하세요! (memo.html 내에서 'post-box' 검색 - ctrl + F 또는 cmd + F)
- 포스팅 박스에 id 값을 주기

```

<div id="post-box" class="form-post">
    <div>
        <div class="form-group">
            <label for="post-url">아티클 URL</label>
            <input id="post-url" class="form-control" placeholder="">
        </div>
        <div class="form-group">
            <label for="post-comment">간단 코멘트</label>
            <textarea class="form-control" rows="2"></textarea>
        </div>
        <button type="button" class="btn btn-primary">기사저장</button>
    </div>
</div>

```

- 포스팅 박스 제어하기

• 6) [领悟과 같이] 패턴 연습하기 - 3. css의 값 가져와보기에서 연습했던 것을 이용해보겠습니다. 코드에서 아래 부분을 채워넣어주세요.

```

function openClose() {
    // id 값 post-box의 display 값이 block 이면
    if ($('#post-box').css('display') == 'block') {
        // post-box를 가리고
        $('#post-box').hide();
    } else {

```

```

        // 아니면 post-box를 보여주기
        $('#post-box').show();
    }
}

```

- posting-box를 시작부터 감춰두기 (css의 `display:none` 속성)



HTML 태그 내 속성에 css 를 지정해주는 것을, inline-style (인라인 스타일) 이라고 부릅니다.

HTML 요소마다 각각 inline-style을 적용하면 어떤 스타일이 적용되었는지 한 눈에 보이지 않아 복잡하겠지만, 적절하게 명시적으로 사용한다면, "아, 처음부터 안보이는(display:none) 태그구나!" 하고 알 수 있겠죠!

물론, css(`<style type="text/css">~</style>`)에서 class를 선택(select)해 `display:none` 속성을 주는 방법도 있습니다.

구현하는 방법은 여러가지!

- 분홍 형광펜 부분을 추가해주세요.

```

<div id="post-box" class="form-post" style="display:none">
    <div>
        <div class="form-group">
            <label for="post-url">아티클 URL</label>
            <input id="post-url" class="form-control" placeholder="">
        </div>
        <div class="form-group">
            <label for="post-comment">간단 코멘트</label>
            <textarea class="form-control" rows="2"></textarea>
        </div>
        <button type="button" class="btn btn-primary">기사저장</button>
    </div>
</div>

```

#### ▼ (4) '포스팅박스 열기' 버튼의 글씨 바꿔주기

- `memo.html` 에서 형광펜 표시된 부분을 주목하세요! (`memo.html` 내에서 'btn-post-box' 검색 - `ctrl + F` 또는 `cmd + F`)
- 포스팅박스 열기 버튼에 id 값 있는지 확인하기

```
<button onclick="openClose()" id="btn-post-box" type="button" class="btn btn-primary">포스팅 박스 열기</button>
```

- 버튼 텍스트를 바꿔주기

```

function openClose() {
    // id 값 post-box의 display 값이 block 이면(= 눈에 보이면)
    if ($('#post-box').css('display') == 'block') {
        // post-box를 가리고
        $('#post-box').hide();
        // 다시 버튼을 클릭하면, 박스 열기를 할 수 있게 텍스트 바꿔두기
        $('#btn-post-box').text('포스팅 박스 열기');
    } else {
        // 아니면(눈에 보이지 않으면) post-box를 펴라
        $('#post-box').show();
        // 다시 버튼을 클릭하면, 박스 닫기를 할 수 있게 텍스트 바꿔두기
        $('#btn-post-box').text('포스팅 박스 닫기');
    }
}

```

#### ▼ (5) 완성코드

```

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

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

```

```

<!-- Bootstrap CSS -->
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
      integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJ1SAwiGgFAW/dAiS6JXm" crossorigin="anonymous">

<!-- JS -->
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
       integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
       crossorigin="anonymous"></script>

<!-- 구글폰트 -->
<link href="https://fonts.googleapis.com/css?family=Stylish&display=swap" rel="stylesheet">

<title>스파르타코딩클럽 | 나홀로 메모장</title>

<!-- style -->
<style type="text/css">
    * {
        font-family: 'Stylish', sans-serif;
    }

    .wrap {
        width: 900px;
        margin: auto;
    }

    .comment {
        color: blue;
        font-weight: bold;
    }

    #post-box {
        width: 500px;
        margin: 20px auto;
        padding: 50px;
        border: black solid;
        border-radius: 5px;
    }
</style>
<script>
    function openClose() {
        // id 값 post-box의 display 값이 block 이면(= 눈에 보이면)
        if ($('#post-box').css('display') == 'block') {
            // post-box를 가지고
            $('#post-box').hide();
            // 다시 버튼을 클릭하면, 박스 열기를 할 수 있게 텍스트 바꿔두기
            $('#btn-post-box').text('포스팅 박스 열기');
        } else {
            // 아니면(눈에 보이지 않으면) post-box를 펴라
            $('#post-box').show();
            // 다시 버튼을 클릭하면, 박스 닫기를 할 수 있게 텍스트 바꿔두기
            $('#btn-post-box').text('포스팅 박스 닫기');
        }
    }
</script>

</head>

<body>
<div class="wrap">
    <div class="jumbotron">
        <h1 class="display-4">나홀로 링크 메모장!</h1>
        <p class="lead">중요한 링크를 저장해두고, 나중에 볼 수 있는 공간입니다</p>
        <hr class="my-4">
        <p class="lead">
            <button onclick="openClose()" id="btn-post-box" type="button" class="btn btn-primary">포스팅 박스 열기</button>
        </p>
    </div>
    <div id="post-box" class="form-post" style="display:none">
        <div>
            <div class="form-group">
                <label for="post-url">아티클 URL</label>
                <input id="post-url" class="form-control" placeholder="">
            </div>
            <div class="form-group">
                <label for="post-comment">간단 코멘트</label>
                <textarea class="form-control" rows="2"></textarea>
            </div>
            <button type="button" class="btn btn-primary">기사저장</button>
        </div>
    </div>
    <div id="cards-box" class="card-columns">
        <div class="card">
            
    </div>
</div>

```

```

<div class="card-body">
    <a href="#" class="card-title">여기 기사 제목이 들어가죠</a>
    <p class="card-text">기사의 요약 내용이 들어갑니다. 동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라만세 무궁화 삼천리
    <p class="card-text comment">여기에 코멘트가 들어갑니다.</p>
</div>
</div>
<div class="card">
    
    <div class="card-body">
        <a href="#" class="card-title">여기 기사 제목이 들어가죠</a>
        <p class="card-text">기사의 요약 내용이 들어갑니다. 동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라만세 무궁화 삼천리
        <p class="card-text comment">여기에 코멘트가 들어갑니다.</p>
    </div>
</div>
<div class="card">
    
    <div class="card-body">
        <a href="#" class="card-title">여기 기사 제목이 들어가죠</a>
        <p class="card-text">기사의 요약 내용이 들어갑니다. 동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라만세 무궁화 삼천리
        <p class="card-text comment">여기에 코멘트가 들어갑니다.</p>
    </div>
</div>
<div class="card">
    
    <div class="card-body">
        <a href="#" class="card-title">여기 기사 제목이 들어가죠</a>
        <p class="card-text">기사의 요약 내용이 들어갑니다. 동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라만세 무궁화 삼천리
        <p class="card-text comment">여기에 코멘트가 들어갑니다.</p>
    </div>
</div>
<div class="card">
    
    <div class="card-body">
        <a href="#" class="card-title">여기 기사 제목이 들어가죠</a>
        <p class="card-text">기사의 요약 내용이 들어갑니다. 동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라만세 무궁화 삼천리
        <p class="card-text comment">여기에 코멘트가 들어갑니다.</p>
    </div>
</div>
</div>
</body>
</html>

```

## [20분]: Javascript + jQuery 연습하기

▼ 8) 🎨자꾸 연습해야 빨리 늘어요!



15분 정도 각자 해보고, 퓨터와 함께 풀이해봅시다!  
(퀴즈 코드의 빈 칸을 채워서, 완성본으로 만들기!)

▼ Q. 퀴즈 코드



아래 코드를 Pycharm에서 week02 폴더 안에 `jquery_question.html` 으로 새 파일을 만들어 복사-붙여넣기 해주세요.

퓨터 님이 html 구조를 간략하게 설명해준 뒤, 각자 코드를 완성해봅시다!

```

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

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

    <!-- JQuery를 import 합니다 -->
    <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 showName() {
            // 1. input-name 값을 가져온다.
            // 2. 가져온 값을 이용해 names에 불일 태그를 만든다. (let tempHtml = `<li>박르</li>`)
            // 3. 만들어둔 temp_html을 names에 붙인다.(jQuery의 $('...').append(tempHtml)을 이용하면 굿!)
        }

        function removeNames() {
            // 1. names의 내부 태그를 모두 비운다.(jQuery의 $('...').empty()를 이용하면 굿!)
        }
    </script>
</head>

<body>
    <h1>jQuery + Javascript의 조합을 연습하자!</h1>

    <div class="question-box">
        <h2>HTML 불이기/지우기 연습</h2>
        <h5>1. 이름을 입력하면 아래 나오게 하기</h5>
        <h5>2. 디지우기 버튼을 만들기</h5>
        <input id="input-name" type="text" placeholder="여기에 이름을 입력" />
        <button onclick="showName()">이름 불이기</button>
        <button onclick="removeNames()">디지우기</button>
        <ul id="names">
            <li>김스파</li>
            <li>박르탄</li>
        </ul>
    </div>
</body>
</html>

```

▼ 분석) 완성본으로 결과물 확인하기



단계별로 접근하기!

1,2,3,4 단계를 나눠 한 번에 하나씩 순서대로 접근하면 쉽습니다! 프로그램은 명령어의 집합! 절차적 사고

[https://s3-us-west-2.amazonaws.com/secure.notion-static.com/9c1fe1c3-2930-4930-9dcb-e856252c7de8/jquery\\_answer.html](https://s3-us-west-2.amazonaws.com/secure.notion-static.com/9c1fe1c3-2930-4930-9dcb-e856252c7de8/jquery_answer.html)

▼ A. 완성 코드

```

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

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

    <!-- JQuery를 import 합니다 -->
    <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>

```

```

</style>

<script>
    function showName() {
        // 1. input-name 값을 가져온다.
        let newName = $('#input-name').val();
        if (newName == '') {
            alert('이름을 입력하세요');
            return;
        }
        // 2. 가져온 값을 이용해 names에 불일 태그를 만든다. (let tempHtml = `<li>${가져온 값}</li>`)
        let tempHtml = `<li>${newName}</li>`;
        // 3. 만들어둔 tempHtml을 names에 붙인다.(jQuery의 $('...').append(tempHtml)을 이용하면 굿!)
        $('#names').append(tempHtml);
    }

    function removeNames() {
        // 1. names의 내부 태그를 모두 비운다.(jQuery의 $('...').empty()를 이용하면 굿!)
        $('#names').empty();
    }
</script>

</head>

<body>
    <h1>jQuery + Javascript의 조합을 연습하자!</h1>

    <div class="question-box">
        <h2>HTML 불이기/지우기 연습</h2>
        <h3>1. 이름을 입력하면 아래 나오게 하기</h3>
        <h3>2. 디지우기 버튼을 만들기</h3>
        <input id="input-name" type="text" placeholder="여기에 이름을 입력" />
        <button onclick="showName()">이름 불이기</button>
        <button onclick="removeNames()">디지우기</button>
        <ul id="names">
            <li>김스파</li>
            <li>브루틴</li>
        </ul>
    </div>
</body>
</html>

```



문제를 먼저 푼 사람들은 '😎 문제뱅크'에서 추가로 문제를 골라 풀어보세요!

## 😎 문제뱅크

### ▼ Javascript 패턴

#### ▼ (1) 조건문 - 1,2,3,4 .. 높여가며 누른 횟수를 보여주는 onclick 함수 만들어보기 (난이도 : ★★)



퀴즈 코드의 설명을 보고 비어 있는 함수를 채워 동작하게 만들어보세요

### ▼ Q. 퀴즈 코드

```

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

<head>
    <meta charset="UTF-8">
    <title>Javascript를 연습하자!</title>

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

    <script>
        // 버튼을 누른 횟수를 저장할 변수 선언
        let count = 0;
        function showButtonCount(){
            // 1. 버튼을 누를 때마다(함수를 호출할 때마다) 누른 횟수 증가
            // 2. 누른 횟수가 20 미만이면 "안녕! {{실제 누른 횟수}} 번 누르셨네요!" alert 창 띄우기
            // 3. 누른 횟수가 20번 이상이면, '앗. 그만 누르세요 이제!' alert 창 띄우기
    </script>

```

```

        }
    </script>

</head>

<body>
<h1>Javascript를 연습하자!</h1>

<div class="question-box">
    <h2>누른 횟수를 보여주는 함수 만들기</h2>
    <h5>1. 버튼을 누를 때마다(함수를 호출할 때마다) 누른 횟수 증가</h5>
    <h5>2. 누른 횟수가 20 미만이면 "안녕! {{실제 누른 횟수}} 번 누르셨네요!" alert 창 띄우기</h5>
    <h5>3. 누른 횟수가 20번 이상이면, '앗. 그만 누르세요 이제!' alert 창 띄우기</h5>
    <button onclick="showButtonCount()">클릭</button>
</div>
</body>

</html>

```

#### ▼ A. 완성 코드

```

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

<head>
    <meta charset="UTF-8">
    <title>Javascript를 연습하자!</title>

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

    <script>
        // 버튼을 누른 횟수를 저장할 변수 선언
        let count = 0;
        function showButtonCount(){
            // 1. 버튼을 누를 때마다(함수를 호출할 때마다) 누른 횟수 증가
            count += 1;
            // 2. 누른 횟수가 20 미만이면 "안녕! {{실제 누른 횟수}} 번 누르셨네요!" alert 창 띄우기
            if (count < 20) {
                alert("안녕! " + count + "번 누르셨네요!");
            } else {
                // 3. 누른 횟수가 20번 이상이면, '앗. 그만 누르세요 이제!' alert 창 띄우기
                alert('앗. 그만 누르세요 이제!');
            }
        }
    </script>
</head>

<body>
<h1>Javascript를 연습하자!</h1>

<div class="question-box">
    <h2>누른 횟수를 보여주는 함수 만들기</h2>
    <h5>1. 버튼을 누를 때마다(함수를 호출할 때마다) 누른 횟수 증가</h5>
    <h5>2. 누른 횟수가 20 미만이면 "안녕! {{실제 누른 횟수}} 번 누르셨네요!" alert 창 띄우기</h5>
    <h5>3. 누른 횟수가 20번 이상이면, '앗. 그만 누르세요 이제!' alert 창 띄우기</h5>
    <button onclick="showButtonCount()">클릭</button>
</div>
</body>

</html>

```

#### ▼ (2) 조건문 - 짹/홀수에 따라 다른 얼럿 띄우는 onclick 함수 함께 만들어보기 (난이도 : ★★★★)

- 짹수 번 눌렀을 때는 "쫙쫙쫙👏", 홀수 번 눌렀을 때는 "홀홀홀🥳" 얼럿을 띄우는 버튼을 만들기
- ▼ 힌트! 짹/홀수 판단하는 방법

```

let even = 4;
console.log(even % 2); // 2로 나눈 나머지가 0입니다.
let odd = 5;
console.log(odd % 2); // 2로 나눈 나머지가 1입니다.

```

#### ▼ Q. 퀴즈 코드

```

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

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>스파르타코딩클럽 | 자바스크립트 연습</title>
    <script>
        let num = 0;
        function handleClick() {
            num += 1;
            // 클릭에 따라 다른 알럿을 띄웁니다.
        }
    </script>
</head>

<body>
    <button>이 버튼을 눌러보세요</button>
</body>

</html>

```

#### ▼ A. 완성 코드

```

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

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>스파르타코딩클럽 | 자바스크립트 연습</title>
    <script>
        let num = 0;
        function handleClick() {
            num += 1;
            if (num % 2 == 0) {
                alert('짝짝짝👏');
            } else {
                alert('홀홀홀🎉');
            }
        }
    </script>
</head>

<body>
    <button onclick="handleClick()">이 버튼을 눌러보세요</button>
</body>

</html>

```

#### ▼ (3) 반복문 & 딕셔너리 - 자전거(parkingBikeTotCnt)가 5개 이하인 정류장의 이름을 출력하기 (난이도 : ★★★★)

- [크롬 개발자도구] console 창에서 풀어보세요.



먼저, 우리가 사용할 데이터를 입력해보겠습니다. 아래 토글 안, 따릉이 데이터(bikes)를 [크롬 개발자도구] console 창에 복사-붙여넣기 해주세요.

#### ▼ 따릉이 데이터(bikes)

```

let bikes = [
    {
        rackTotCnt: "7",
        stationName: "101. (구)합정동 주민센터",
        parkingBikeTotCnt: "4",
        shared: "14",
        stationLatitude: "37.54956055",
        stationLongitude: "126.90575409",
        stationId: "ST-3",
    },
    {
        rackTotCnt: "22",
        stationName: "102. 망원역 1번출구 앞",
        parkingBikeTotCnt: "17",
        shared: "5",
        stationLatitude: "37.55564880",
        stationLongitude: "126.91062927",
    }
]

```

```

        stationId: "ST-4",
    },
{
    rackTotCnt: "16",
    stationName: "103. 망원역 2번출구 앞",
    parkingBikeTotCnt: "11",
    shared: "13",
    stationLatitude: "37.55495071",
    stationLongitude: "126.91083527",
    stationId: "ST-5",
},
{
    rackTotCnt: "15",
    stationName: "104. 합정역 1번출구 앞",
    parkingBikeTotCnt: "11",
    shared: "0",
    stationLatitude: "37.55062866",
    stationLongitude: "126.91498566",
    stationId: "ST-6",
},
{
    rackTotCnt: "7",
    stationName: "105. 합정역 5번출구 앞",
    parkingBikeTotCnt: "1",
    shared: "0",
    stationLatitude: "37.55000687",
    stationLongitude: "126.91482544",
    stationId: "ST-7",
},
{
    rackTotCnt: "12",
    stationName: "106. 합정역 7번출구 앞",
    parkingBikeTotCnt: "8",
    shared: "8",
    stationLatitude: "37.54864502",
    stationLongitude: "126.91282654",
    stationId: "ST-8",
},
{
    rackTotCnt: "7",
    stationName: "107. 신한은행 서교동금융센터점 앞",
    parkingBikeTotCnt: "5",
    shared: "14",
    stationLatitude: "37.55751038",
    stationLongitude: "126.91850281",
    stationId: "ST-9",
},
{
    rackTotCnt: "12",
    stationName: "108. 서교동 사거리",
    parkingBikeTotCnt: "9",
    shared: "8",
    stationLatitude: "37.55274582",
    stationLongitude: "126.91861725",
    stationId: "ST-10",
},
{
    rackTotCnt: "12",
    stationName: "109. 제일빌딩 앞",
    parkingBikeTotCnt: "8",
    shared: "33",
    stationLatitude: "37.54769135",
    stationLongitude: "126.91998291",
    stationId: "ST-11",
},
{
    rackTotCnt: "22",
    stationName: "110. 사천교",
    parkingBikeTotCnt: "16",
    shared: "5",
    stationLatitude: "37.56819916",
    stationLongitude: "126.91784668",
    stationId: "ST-13",
},
{
    rackTotCnt: "12",
    stationName: "111. 상수역 2번출구 앞",
    parkingBikeTotCnt: "9",
    shared: "25",
    stationLatitude: "37.54787064",
    stationLongitude: "126.92353058",
    stationId: "ST-15",
},
{
    rackTotCnt: "12",
    stationName: "112. 극동방송국 앞",
    parkingBikeTotCnt: "8",

```

```

        shared: "25",
        stationLatitude: "37.54920197",
        stationLongitude: "126.92320251",
        stationId: "ST-16",
    },
    {
        rackTotCnt: "27",
        stationName: "113. 홍대입구역 2번출구 앞",
        parkingBikeTotCnt: "24",
        shared: "22",
        stationLatitude: "37.55749893",
        stationLongitude: "126.92380524",
        stationId: "ST-18",
    },
    {
        rackTotCnt: "17",
        stationName: "114. 홍대입구역 8번출구 앞",
        parkingBikeTotCnt: "14",
        shared: "129",
        stationLatitude: "37.55706024",
        stationLongitude: "126.92442322",
        stationId: "ST-20",
    },
    {
        rackTotCnt: "17",
        stationName: "115. 사루비아 빌딩 앞",
        parkingBikeTotCnt: "1",
        shared: "0",
        stationLatitude: "37.55893326",
        stationLongitude: "126.92711639",
        stationId: "ST-12",
    },
    {
        rackTotCnt: "7",
        stationName: "116. 일진아이월아파트 옆",
        parkingBikeTotCnt: "1",
        shared: "0",
        stationLatitude: "37.56454086",
        stationLongitude: "126.92707062",
        stationId: "ST-14",
    },
    {
        rackTotCnt: "27",
        stationName: "117. 홍은사거리",
        parkingBikeTotCnt: "9",
        shared: "0",
        stationLatitude: "37.59115982",
        stationLongitude: "126.94132996",
        stationId: "ST-17",
    },
    {
        rackTotCnt: "12",
        stationName: "118. 광흥창역 2번출구 앞",
        parkingBikeTotCnt: "9",
        shared: "67",
        stationLatitude: "37.54773331",
        stationLongitude: "126.93176270",
        stationId: "ST-19",
    },
    {
        rackTotCnt: "12",
        stationName: "119. 서강나루 공원",
        parkingBikeTotCnt: "9",
        shared: "17",
        stationLatitude: "37.54528427",
        stationLongitude: "126.93105316",
        stationId: "ST-21",
    },
    {
        rackTotCnt: "7",
        stationName: "120. 신수동 사거리",
        parkingBikeTotCnt: "3",
        shared: "0",
        stationLatitude: "37.54524231",
        stationLongitude: "126.93411255",
        stationId: "ST-22",
    },
    {
        rackTotCnt: "17",
        stationName: "121. 미포소방서 앞",
        parkingBikeTotCnt: "11",
        shared: "24",
        stationLatitude: "37.54976654",
        stationLongitude: "126.93317413",
        stationId: "ST-23",
    },

```

```

        rackTotCnt: "12",
        stationName: "122. 신성기사식당 앞",
        parkingBikeTotCnt: "6",
        shared: "0",
        stationLatitude: "37.54745865",
        stationLongitude: "126.93837738",
        stationId: "ST-24",
    },
    {
        rackTotCnt: "22",
        stationName: "123. 문화촌 공원",
        parkingBikeTotCnt: "7",
        shared: "0",
        stationLatitude: "37.59432983",
        stationLongitude: "126.94738770",
        stationId: "ST-25",
    },
    {
        rackTotCnt: "22",
        stationName: "124. 서강대 정문 건너편",
        parkingBikeTotCnt: "7",
        shared: "0",
        stationLatitude: "37.55113983",
        stationLongitude: "126.93698883",
        stationId: "ST-26",
    },
    {
        rackTotCnt: "17",
        stationName: "125. 서강대 남문 옆",
        parkingBikeTotCnt: "13",
        shared: "0",
        stationLatitude: "37.54948425",
        stationLongitude: "126.93894958",
        stationId: "ST-27",
    },
    {
        rackTotCnt: "22",
        stationName: "126. 서강대 후문 옆",
        parkingBikeTotCnt: "17",
        shared: "5",
        stationLatitude: "37.55041122",
        stationLongitude: "126.94384766",
        stationId: "ST-28",
    },
    {
        rackTotCnt: "22",
        stationName: "128. 신촌역(2호선) 1번출구 옆",
        parkingBikeTotCnt: "14",
        shared: "5",
        stationLatitude: "37.55549622",
        stationLongitude: "126.93634033",
        stationId: "ST-30",
    },
    {
        rackTotCnt: "17",
        stationName: "129. 신촌역(2호선) 6번출구 옆",
        parkingBikeTotCnt: "8",
        shared: "0",
        stationLatitude: "37.55505371",
        stationLongitude: "126.93756866",
        stationId: "ST-31",
    },
    {
        rackTotCnt: "12",
        stationName: "130. 신촌역(2호선) 7번출구 앞",
        parkingBikeTotCnt: "8",
        shared: "17",
        stationLatitude: "37.55485916",
        stationLongitude: "126.93615723",
        stationId: "ST-32",
    },
    {
        rackTotCnt: "25",
        stationName: "131. 증산2교",
        parkingBikeTotCnt: "17",
        shared: "4",
        stationLatitude: "37.58417130",
        stationLongitude: "126.91110229",
        stationId: "ST-33",
    },
    {
        rackTotCnt: "17",
        stationName: "133. 해담는다리",
        parkingBikeTotCnt: "11",
        shared: "12",
        stationLatitude: "37.58203125",
        stationLongitude: "126.90899658",
    }

```

```

        stationId: "ST-35",
    },
{
    rackTotCnt: "10",
    stationName: "134. 연세로 명물길",
    parkingBikeTotCnt: "6",
    shared: "20",
    stationLatitude: "37.55789185",
    stationLongitude: "126.93807983",
    stationId: "ST-36",
},
{
    rackTotCnt: "12",
    stationName: "135. 명물길 원형무대 앞",
    parkingBikeTotCnt: "10",
    shared: "0",
    stationLatitude: "37.55910110",
    stationLongitude: "126.93917847",
    stationId: "ST-37",
},
{
    rackTotCnt: "9",
    stationName: "136. 대흥동 주민센터",
    parkingBikeTotCnt: "1",
    shared: "11",
    stationLatitude: "37.55600357",
    stationLongitude: "126.94229889",
    stationId: "ST-38",
},
{
    rackTotCnt: "12",
    stationName: "137. NH농협 신촌지점 앞",
    parkingBikeTotCnt: "4",
    shared: "0",
    stationLatitude: "37.55681229",
    stationLongitude: "126.94318390",
    stationId: "ST-39",
},
{
    rackTotCnt: "12",
    stationName: "138. 신촌동 제1공영주차장 앞",
    parkingBikeTotCnt: "7",
    shared: "25",
    stationLatitude: "37.55917740",
    stationLongitude: "126.93452454",
    stationId: "ST-40",
},
{
    rackTotCnt: "15",
    stationName: "139. 연세대 정문 건너편",
    parkingBikeTotCnt: "13",
    shared: "7",
    stationLatitude: "37.55979538",
    stationLongitude: "126.93447876",
    stationId: "ST-43",
},
{
    rackTotCnt: "22",
    stationName: "140. 이화여대 후문",
    parkingBikeTotCnt: "6",
    shared: "0",
    stationLatitude: "37.56000900",
    stationLongitude: "126.94073486",
    stationId: "ST-41",
},
{
    rackTotCnt: "22",
    stationName: "141. 연대 대운동장 옆",
    parkingBikeTotCnt: "13",
    shared: "5",
    stationLatitude: "37.56238174",
    stationLongitude: "126.93264771",
    stationId: "ST-42",
},
{
    rackTotCnt: "13",
    stationName: "142. 아현역 4번출구 앞",
    parkingBikeTotCnt: "1",
    shared: "0",
    stationLatitude: "37.55720139",
    stationLongitude: "126.95566559",
    stationId: "ST-200",
},
{
    rackTotCnt: "11",
    stationName: "143. 공덕역 2번출구",
    parkingBikeTotCnt: "7",

```

```

        shared: "0",
        stationLatitude: "37.54457855",
        stationLongitude: "126.95021820",
        stationId: "ST-201",
    },
    {
        rackTotCnt: "12",
        stationName: "144. 공덕역 8번출구",
        parkingBikeTotCnt: "6",
        shared: "0",
        stationLatitude: "37.54357910",
        stationLongitude: "126.95132446",
        stationId: "ST-202",
    },
    {
        rackTotCnt: "11",
        stationName: "145. 공덕역 5번출구",
        parkingBikeTotCnt: "8",
        shared: "36",
        stationLatitude: "37.54425049",
        stationLongitude: "126.95163727",
        stationId: "ST-203",
    },
    {
        rackTotCnt: "14",
        stationName: "146. 마포역 2번출구 뒤",
        parkingBikeTotCnt: "6",
        shared: "0",
        stationLatitude: "37.53993607",
        stationLongitude: "126.94582367",
        stationId: "ST-204",
    },
    {
        rackTotCnt: "9",
        stationName: "147. 마포역 4번출구 뒤",
        parkingBikeTotCnt: "4",
        shared: "0",
        stationLatitude: "37.53927231",
        stationLongitude: "126.94591522",
        stationId: "ST-205",
    },
    {
        rackTotCnt: "17",
        stationName: "150. 서강대역 2번출구 앞",
        parkingBikeTotCnt: "13",
        shared: "65",
        stationLatitude: "37.55295563",
        stationLongitude: "126.93434143",
        stationId: "ST-207",
    },
    {
        rackTotCnt: "12",
        stationName: "151. 망원1동주민센터",
        parkingBikeTotCnt: "11",
        shared: "58",
        stationLatitude: "37.55568695",
        stationLongitude: "126.90554810",
        stationId: "ST-208",
    },
    {
        rackTotCnt: "32",
        stationName: "152. 마포구민체육센터 앞",
        parkingBikeTotCnt: "8",
        shared: "31",
        stationLatitude: "37.55661011",
        stationLongitude: "126.89801788",
        stationId: "ST-209",
    },
    {
        rackTotCnt: "12",
        stationName: "153. 성산2교 사거리",
        parkingBikeTotCnt: "7",
        shared: "17",
        stationLatitude: "37.56469727",
        stationLongitude: "126.91261292",
        stationId: "ST-210",
    },
    {
        rackTotCnt: "15",
        stationName: "154. 마포구청역 ",
        parkingBikeTotCnt: "9",
        shared: "0",
        stationLatitude: "37.56090927",
        stationLongitude: "126.90549469",
        stationId: "ST-211",
    },

```

```

    rackTotCnt: "17",
    stationName: "155. 가좌역1 번출구 뒤",
    parkingBikeTotCnt: "14",
    shared: "0",
    stationLatitude: "37.56855011",
    stationLongitude: "126.91451263",
    stationId: "ST-212",
},
{
    rackTotCnt: "12",
    stationName: "156. 서울서부지방법원 앞",
    parkingBikeTotCnt: "9",
    shared: "0",
    stationLatitude: "37.54990387",
    stationLongitude: "126.95514679",
    stationId: "ST-213",
},
{
    rackTotCnt: "14",
    stationName: "157. 애오개역 4번출구 앞",
    parkingBikeTotCnt: "1",
    shared: "0",
    stationLatitude: "37.55300140",
    stationLongitude: "126.95668793",
    stationId: "ST-214",
},
{
    rackTotCnt: "17",
    stationName: "158. 독립문 어린이 공원",
    parkingBikeTotCnt: "1",
    shared: "0",
    stationLatitude: "37.57125854",
    stationLongitude: "126.96047974",
    stationId: "ST-215",
},
{
    rackTotCnt: "9",
    stationName: "159. 0대역 4번 출구",
    parkingBikeTotCnt: "1",
    shared: "0",
    stationLatitude: "37.55695343",
    stationLongitude: "126.94634247",
    stationId: "ST-216",
},
{
    rackTotCnt: "22",
    stationName: "160. 북아현동 가구거리",
    parkingBikeTotCnt: "15",
    shared: "0",
    stationLatitude: "37.55754852",
    stationLongitude: "126.95938110",
    stationId: "ST-217",
},
{
    rackTotCnt: "10",
    stationName: "161. 무악재역1번 출구",
    parkingBikeTotCnt: "0",
    shared: "0",
    stationLatitude: "37.58224487",
    stationLongitude: "126.95064545",
    stationId: "ST-218",
},
{
    rackTotCnt: "17",
    stationName: "162. 복원고가차도 밑",
    parkingBikeTotCnt: "8",
    shared: "0",
    stationLatitude: "37.56526947",
    stationLongitude: "126.94624329",
    stationId: "ST-219",
},
{
    rackTotCnt: "9",
    stationName: "163. 명지전문대학교 정문 앞",
    parkingBikeTotCnt: "0",
    shared: "0",
    stationLatitude: "37.58369827",
    stationLongitude: "126.92496490",
    stationId: "ST-220",
},
{
    rackTotCnt: "12",
    stationName: "164. 북가좌1동 주민센터 ",
    parkingBikeTotCnt: "7",
    shared: "25",
    stationLatitude: "37.57447815",
    stationLongitude: "126.91004944",
}

```

```

        stationId: "ST-221",
    },
    {
        rackTotCnt: "22",
        stationName: "165. 중앙근린공원",
        parkingBikeTotCnt: "9",
        shared: "0",
        stationLatitude: "37.57513809",
        stationLongitude: "126.91394043",
        stationId: "ST-222",
    },
    {
        rackTotCnt: "22",
        stationName: "166. 가재울 초등학교",
        parkingBikeTotCnt: "6",
        shared: "0",
        stationLatitude: "37.57327652",
        stationLongitude: "126.91967773",
        stationId: "ST-223",
    },
    {
        rackTotCnt: "17",
        stationName: "167. 연가초등학교 옆",
        parkingBikeTotCnt: "12",
        shared: "0",
        stationLatitude: "37.57946014",
        stationLongitude: "126.91712952",
        stationId: "ST-224",
    },
    {
        rackTotCnt: "17",
        stationName: "169. 북가좌 삼거리",
        parkingBikeTotCnt: "6",
        shared: "0",
        stationLatitude: "37.57300186",
        stationLongitude: "126.90779877",
        stationId: "ST-226",
    },
    {
        rackTotCnt: "12",
        stationName: "170. 가재울 뉴타운 주유소 옆",
        parkingBikeTotCnt: "9",
        shared: "33",
        stationLatitude: "37.57311249",
        stationLongitude: "126.92244720",
        stationId: "ST-227",
    },
    {
        rackTotCnt: "12",
        stationName: "171. 임광빌딩 앞",
        parkingBikeTotCnt: "9",
        shared: "8",
        stationLatitude: "37.56472397",
        stationLongitude: "126.96727753",
        stationId: "ST-228",
    },
    {
        rackTotCnt: "10",
        stationName: "173. 서대문역 8번출구 앞",
        parkingBikeTotCnt: "4",
        shared: "0",
        stationLatitude: "37.56477737",
        stationLongitude: "126.96614838",
        stationId: "ST-230",
    },
    {
        rackTotCnt: "22",
        stationName: "175. 흥연2교옆",
        parkingBikeTotCnt: "6",
        shared: "0",
        stationLatitude: "37.57807159",
        stationLongitude: "126.93081665",
        stationId: "ST-231",
    },
    {
        rackTotCnt: "12",
        stationName: "176. 명지대학교 도서관",
        parkingBikeTotCnt: "0",
        shared: "0",
        stationLatitude: "37.58109665",
        stationLongitude: "126.92402649",
        stationId: "ST-555",
    },
    {
        rackTotCnt: "10",
        stationName: "177. 북가좌 초등학교",
        parkingBikeTotCnt: "1",
    }

```

```

        shared: "0",
        stationLatitude: "37.57767487",
        stationLongitude: "126.90980530",
        stationId: "ST-345",
    },
    {
        rackTotCnt: "12",
        stationName: "178. 증산3교 앞",
        parkingBikeTotCnt: "0",
        shared: "0",
        stationLatitude: "37.57987595",
        stationLongitude: "126.90634918",
        stationId: "ST-349",
    },
    {
        rackTotCnt: "17",
        stationName: "179. 가좌역 4번출구 앞",
        parkingBikeTotCnt: "14",
        shared: "47",
        stationLatitude: "37.56912231",
        stationLongitude: "126.91479492",
        stationId: "ST-232",
    },
    {
        rackTotCnt: "12",
        stationName: "180. 충정로역 7번출구 아래",
        parkingBikeTotCnt: "10",
        shared: "8",
        stationLatitude: "37.55996704",
        stationLongitude: "126.96246338",
        stationId: "ST-233",
    },
    {
        rackTotCnt: "17",
        stationName: "181. 망원초록길 입구",
        parkingBikeTotCnt: "9",
        shared: "0",
        stationLatitude: "37.55134201",
        stationLongitude: "126.90267181",
        stationId: "ST-339",
    },
    {
        rackTotCnt: "12",
        stationName: "182. 망원2빌립프장 앞",
        parkingBikeTotCnt: "7",
        shared: "0",
        stationLatitude: "37.55156708",
        stationLongitude: "126.90284729",
        stationId: "ST-340",
    },
    {
        rackTotCnt: "17",
        stationName: "183. 하늘채코오롱아파트 건너편",
        parkingBikeTotCnt: "10",
        shared: "0",
        stationLatitude: "37.56516647",
        stationLongitude: "126.91939545",
        stationId: "ST-341",
    },
    {
        rackTotCnt: "11",
        stationName: "184. SK망원동주유소 건너편",
        parkingBikeTotCnt: "4",
        shared: "0",
        stationLatitude: "37.55894852",
        stationLongitude: "126.90775299",
        stationId: "ST-342",
    },
    {
        rackTotCnt: "17",
        stationName: "185. 마포 신수공원 앞",
        parkingBikeTotCnt: "5",
        shared: "0",
        stationLatitude: "37.54254532",
        stationLongitude: "126.93429565",
        stationId: "ST-343",
    },
    {
        rackTotCnt: "42",
        stationName: "186. 월드컵공원",
        parkingBikeTotCnt: "22",
        shared: "10",
        stationLatitude: "37.56396484",
        stationLongitude: "126.89820862",
        stationId: "ST-344",
    },

```

```

    rackTotCnt: "12",
    stationName: "188. 흥은동 정원여중 입구",
    parkingBikeTotCnt: "2",
    shared: "0",
    stationLatitude: "37.58638763",
    stationLongitude: "126.93512726",
    stationId: "ST-346",
},
{
    rackTotCnt: "12",
    stationName: "191. 서우빌딩(바른학원)",
    parkingBikeTotCnt: "6",
    shared: "0",
    stationLatitude: "37.57889175",
    stationLongitude: "126.91073608",
    stationId: "ST-347",
},
{
    rackTotCnt: "12",
    stationName: "192. 연서어린이공원",
    parkingBikeTotCnt: "0",
    shared: "0",
    stationLatitude: "37.57222748",
    stationLongitude: "126.92306519",
    stationId: "ST-348",
},
{
    rackTotCnt: "12",
    stationName: "194. 증산교 앞",
    parkingBikeTotCnt: "2",
    shared: "0",
    stationLatitude: "37.57731628",
    stationLongitude: "126.90296936",
    stationId: "ST-350",
},
{
    rackTotCnt: "12",
    stationName: "195. 모래내고가차도",
    parkingBikeTotCnt: "6",
    shared: "42",
    stationLatitude: "37.56765747",
    stationLongitude: "126.91780853",
    stationId: "ST-351",
},
{
    rackTotCnt: "12",
    stationName: "196. 연희교차로 인근",
    parkingBikeTotCnt: "1",
    shared: "0",
    stationLatitude: "37.56612015",
    stationLongitude: "126.92589569",
    stationId: "ST-352",
},
{
    rackTotCnt: "17",
    stationName: "198. 충정2교",
    parkingBikeTotCnt: "15",
    shared: "0",
    stationLatitude: "37.56213760",
    stationLongitude: "126.96377563",
    stationId: "ST-354",
},
{
    rackTotCnt: "32",
    stationName: "199. 서울 월드컵 경기장",
    parkingBikeTotCnt: "7",
    shared: "0",
    stationLatitude: "37.56684494",
    stationLongitude: "126.89644623",
    stationId: "ST-443",
},
{
    rackTotCnt: "22",
    stationName: "200. 국회의원회관",
    parkingBikeTotCnt: "8",
    shared: "0",
    stationLatitude: "37.52841568",
    stationLongitude: "126.91391754",
    stationId: "ST-45",
},
{
    rackTotCnt: "17",
    stationName: "201. 진미파라곤 앞",
    parkingBikeTotCnt: "9",
    shared: "6",
    stationLatitude: "37.53123856",
    stationLongitude: "126.92133331",
}

```

```

        stationId: "ST-46",
    },
{
    rackTotCnt: "32",
    stationName: "202. 국민일보 앞",
    parkingBikeTotCnt: "21",
    shared: "19",
    stationLatitude: "37.52881622",
    stationLongitude: "126.92453003",
    stationId: "ST-47",
},
{
    rackTotCnt: "17",
    stationName: "203. 국회의사당역 3번출구 옆",
    parkingBikeTotCnt: "14",
    shared: "76",
    stationLatitude: "37.52805710",
    stationLongitude: "126.91870117",
    stationId: "ST-51",
},
{
    rackTotCnt: "15",
    stationName: "204. 국회의사당역 5번출구 옆",
    parkingBikeTotCnt: "10",
    shared: "53",
    stationLatitude: "37.52816391",
    stationLongitude: "126.91702271",
    stationId: "ST-50",
},
{
    rackTotCnt: "22",
    stationName: "205. 산업은행 앞",
    parkingBikeTotCnt: "13",
    shared: "0",
    stationLatitude: "37.52626419",
    stationLongitude: "126.92050934",
    stationId: "ST-52",
},
{
    rackTotCnt: "37",
    stationName: "206. KBS 앞",
    parkingBikeTotCnt: "24",
    shared: "11",
    stationLatitude: "37.52466583",
    stationLongitude: "126.91802216",
    stationId: "ST-53",
},
{
    rackTotCnt: "42",
    stationName: "207. 여의나루역 1번출구 앞",
    parkingBikeTotCnt: "16",
    shared: "0",
    stationLatitude: "37.52698898",
    stationLongitude: "126.93209839",
    stationId: "ST-73",
},
{
    rackTotCnt: "14",
    stationName: "209. 유진투자증권빌딩 앞",
    parkingBikeTotCnt: "12",
    shared: "14",
    stationLatitude: "37.52461243",
    stationLongitude: "126.92783356",
    stationId: "ST-55",
},
{
    rackTotCnt: "23",
    stationName: "210. IFC몰",
    parkingBikeTotCnt: "16",
    shared: "13",
    stationLatitude: "37.52606583",
    stationLongitude: "126.92553711",
    stationId: "ST-56",
},
{
    rackTotCnt: "15",
    stationName: "211. 여의도역 4번출구 옆",
    parkingBikeTotCnt: "2",
    shared: "0",
    stationLatitude: "37.52222824",
    stationLongitude: "126.92463684",
    stationId: "ST-57",
},
{
    rackTotCnt: "37",
    stationName: "212. 여의도역 1번출구 옆",
    parkingBikeTotCnt: "9",

```

```

        shared: "0",
        stationLatitude: "37.52136230",
        stationLongitude: "126.92346191",
        stationId: "ST-58",
    },
];

```

- Q. 5대 이하 정류장의 이름을 출력해주세요.

▼ A. 정답

```

for (let i = 0; i < bikes.length; i++) {
    if (bikes[i]['parkingBikeTotCnt'] <= 5) {
        let station = bikes[i]['stationName'];
        console.log(station);
    }
}

```



마찬가지로 여러 번 코드를 '중복해서 입력하지 않게' , 함수로 만들어볼까요?

- Q. 입력한 자전거 대수 이하 정류장의 이름을 출력하는 함수를 만들어주세요.

▼ A. 정답

```

function showBikeCount(num){
    for (let i = 0; i < bikes.length; i++) {
        if (bikes[i]['parkingBikeTotCnt'] <= num) {
            let station = bikes[i]['stationName'];
            console.log(num + "대 이하 정류장 : " + station);
        }
    }
}

// 이라면 아래와 같은 것이 가능!
showBikeCount(10) // 10개 이하 주차된 정류소만 출력!
showBikeCount(5) // 5개 이하 주차된 정류소만 출력!

```

▼ JQuery 연습

▼ (1) 입력받은 값 확인하기 (난이도 : ★★)

- 입력받은 칸이 빈 칸이면 경고 메시지를 띄우고, 그렇지 않으면 `alert('입력값')` 띄우기

▼ Q. 퀴즈 코드

```

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

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

    <!-- JQuery를 import 합니다 -->
    <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 isEmpty() {
            // 1. input-value의 입력값을 가져온다.
            // 2. 만약 입력값이 빈칸이면 if(입력값==' ')
            // 3. alert(입력값) 띄우기
        }
    </script>
</head>

<body>
<h1>jQuery + Javascript의 조합을 연습하자!</h1>

```

```

<div class="question-box">
    <h2>빈칸 체크 함수 만들기</h2>
    <h5>1-1. 버튼을 눌렀을 때 입력한 글자로 얼럿 띄우기</h5>
    <h5>[완성본]1-2. 버튼을 눌렀을 때 칸에 아무것도 없으면 "입력하세요!" 얼럿 띄우기</h5>
    <input id="input-value" type="text"/>
    <button onclick="isEmpty()">클릭</button>
</div>
</body>

</html>

```

▼ A. 완성 코드

```

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

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

    <!-- JQuery를 import 합니다 -->
    <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 isEmpty() {
            // 1. input-value의 입력값을 가져온다.
            let value = $('#input-value').val();
            // 2. 만약 입력값이 빈칸이면 if(입력값=='')
            if (value == '') {
                // 3. alert('입력하세요!') 띄우기
                alert('입력하세요!');
            } else {
                // 4. alert(입력값) 띄우기
                alert(value);
            }
        }
    </script>
</head>

<body>
<h1>jQuery + Javascript의 조합을 연습하자!</h1>

<div class="question-box">
    <h2>1. 빈칸 체크 함수 만들기</h2>
    <h5>1-1. 버튼을 눌렀을 때 입력한 글자로 얼럿 띄우기</h5>
    <h5>[완성본]1-2. 버튼을 눌렀을 때 칸에 아무것도 없으면 "입력하세요!" 얼럿 띄우기</h5>
    <input id="input-value" type="text"/>
    <button onclick="isEmpty()">클릭</button>
</div>
</body>

</html>

```

▼ (2) 입력받은 이메일 형식이 유효한지 확인하기 (난이도 : ★★★★)

▼ Q. 퀴즈 코드

```

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

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

    <!-- JQuery를 import 합니다 -->
    <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 validateEmail() {
        // 1. input-email 값을 가져온다.
        // 2. 만약 가져온 값에 @가 있으면 (includes 이용하기 - 찾아보자!)
        // 3. contact@gmail.com -> gmail 만 추출해서
        // 4. alert(도메인 값);으로 띄우기
        // 5. 만약 이메일이 아니면 '이메일이 아닙니다.' 라는 얼럿 띄우기
    }
</script>

</head>

<body>
<h1>jQuery + Javascript의 조합을 연습하자!</h1>

<div class="question-box">
    <h2>이메일 판별 함수 만들기</h2>
    <h5>1. 버튼을 눌렀을 때 입력받은 이메일로 얼럿 띄우기</h5>
    <h5>2. 이메일이 아니면(@가 없으면) '이메일이 아닙니다'라는 얼럿 띄우기</h5>
    <h5>[완성본]3. 이메일 도메인만 얼럿 띄우기</h5>
    <input id="input-email" type="text"/>
    <button onclick="validateEmail()">클릭</button>
</div>
</body>

</html>

```

#### ▼ A. 완성 코드

```

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

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

    <!-- JQuery를 import 합니다 -->
    <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 validateEmail() {
            // 1. input-email 값을 가져온다.
            let email = $('#input-email').val();
            // 2. 만약 가져온 값에 @가 있으면 (includes 이용하기 - 찾아보자!)
            if (email.includes('@')) {
                // 3. contact@gmail.com -> gmail 만 추출해서
                // 4. alert(도메인 값);으로 띄우기
                let domainWithDot = email.split('@')[1];
                let onlyDomain = domainWithDot.split('.')[0];
                alert(onlyDomain);
            } else {
                // 5. 만약 이메일이 아니면 '이메일이 아닙니다.' 라는 얼럿 띄우기
                alert('이메일이 아닙니다.');
            }
        }
    </script>
</head>

<body>
<h1>jQuery + Javascript의 조합을 연습하자!</h1>

<div class="question-box">
    <h2>이메일 판별 함수 만들기</h2>
    <h5>1. 버튼을 눌렀을 때 입력받은 이메일로 얼럿 띄우기</h5>
    <h5>2. 이메일이 아니면(@가 없으면) '이메일이 아닙니다'라는 얼럿 띄우기</h5>
    <h5>[완성본]3. 이메일 도메인만 얼럿 띄우기</h5>
    <input id="input-email" type="text"/>
    <button onclick="validateEmail()">클릭</button>
</div>
</body>

</html>

```

후반 3시간

[월수/화목반] : 체크인 & 출석체크



튜터님은 체크인과 함께 출석 체크(링크)를 진행해주세요!

스파르타코딩클럽 출석체크

<http://spartacodingclub.shop/attendance>

▼ "15초 체크인" 진행합니다

- 튜터는 타이머를 띄워주세요! ([링크](#))
- 본인의 감정상태와 오늘 있었던 소소한 일을 공유하는 시간입니다.

## 배우고 적용하기 - 후반부

### 5분 꿀팁 - 프로그래밍은 문제 해결#1 \_구글링

▼ 구글링 - 구글에 모르는 것 검색하기

- 앞 5분 꿀팁에서 '프로그래밍은 문제를 해결해나가는 과정' 이라고 한거 기억나시나요? 모르거나 막히는 부분이 있을 때 여러 가지 방법을 써서 문제를 해결해나가면 됩니다. 앞으로 개발자들이 문제 해결을 위해 많이 쓰는 방법인 [구글링\(Googling\)](#), [질문](#) [질하기](#), [디버깅\(에러 해결해나가기\)](#) 를 차근차근 알아가볼게요!
- 첫번째 순서로 구글링을 소개하고, 바로 써먹을 수 있는 꿀팁을 드리겠습니다!
- ! 구글링이 무엇인가요?
  - 여러분이 모르는 것, 전 세계 사람들 다 모릅니다. 이럴 땐 필요한 건 구글! 구글에 검색한다고 해서 [구글링\(Googling\)](#) 이라고 합니다.
  - ! 구글(Google) 예만 검색해야하나요? 네이버, 다음은 안되나요?
  - 구글로 검색하면 국내 블로그, 사이트 중심이 아니라 전 세계 풍부하고 방대한 자료가 나옵니다. 개발은 전 세계 사람이 하니까 보다 많은 자료를 접할 수 있겠죠?
- 아래 내용을 **나중에 스스로 읽어보면서**, 구글 검색 잘하는 방법을 구글링해보세요! 😊
  - 추천 키워드 : [구글링하는 법](#), [how to googling for programmer](#), [구글 검색 팁](#)
  - 검색 결과 예: 개발자를 위한 Google 검색 노하우 - 김환규 hosted by OKKY ([1편 링크](#) / [2편 링크](#))

▼ 검색어 팁

- 아래처럼 검색어를 조합해 입력해보세요!
- 기술을 처음 배우고 싶을 때 : '기술이름' + 'tutorial' (예. [Javascript tutorial](#))
- 기능을 찾을 때 : '기술이름' + 'how to' + '찾을 내용' (예. [Javascript how to hide div](#))
- 어떻게 사용하는지 예제를 보고 싶을 때 : '기술이름' + '내용' + 'example' (예. [Javascript onclick tutorial](#))
- 원하는 사이트명 포함해 검색할 수도 있습니다. (예: [stackoverflow Javascript how to hide div](#) - stackoverflow라는 사이트에서 검색)

▼ 검색 결과 중에, 좋은 자료 고르기

- 좋은 자료를 찾으려면 경험치가 필요해요. 많이 검색해보면 자연스레 나만의 검색 노하우와 자료 판단하는 눈이 길러질 거에요.
- 검증과정을 거쳐 잘못된 내용이 금방 수정되거나 오류 자체가 적은 사이트
  - MDN([링크](#)) - 참고. MDN을 신뢰할 수 있는 이유([링크](#))
  - 기술 공식 문서(예를 들면, [부트스트랩 컴퓨터 페이지](#))
  - 신뢰할 수 있는 블로그(tech 회사의 기술 블로그, IT 전문 매거진)
  - [stackoverflow\(\[링크\]\(#\)\)](#) - 개발 QnA 사이트입니다. 전 세계적으로 많이 쓰입니다. 질문과 답변, 댓글에 사용자들이 [vote](#) 할 수 있어요. 좋은 질문과 답변에는 [vote](#) 수가 높습니다. ([링크](#))

- 그 외에

- 해결책뿐만 아니라 문제(에러)의 이유까지 적어두어서 내 문제와 같은지 판단할 수 있는 정보를 제공하는 글
- 오래되지 않은 자료 - 웹의 경우, 기술이 빠르게 발전하기 때문에, 몇 년이 지난 오래된 자료는 버전 등의 문제로 내용이 달라질 수 있어요.

▼ 검색 결과글에 모르는 용어가 너무 많다면?

1. 자, 일단 심호흡합시다. 침착하세요! 내가 이해할 수 있는 만큼만 부분부분 찬찬히 읽어봅시다.
  2. 읽어보고 내가 따라서 하는게 가능한가요? 그럼 한 번 따라서 해보죠! 부분적으로 안되거나, 미심쩍은 부분, 더 알고 싶은 내용은 슬랙 채널에 질문해보세요. (단, 참고한 자료 링크도 포함해서 질문해야겠죠?)
  3. 따라하기 어렵나요? 내용을 전혀 알 수가 없나요? 좋아요. 그럼 다른 검색결과로 넘어가죠! 내 수준에 맞는 자료를 찾으면 됩니다.
- 이렇게 해도 되냐고요? 그럼요. 우리는 프로그래밍을 배우고 있는 과정이니까 모르는 내용이 많아도 괜찮아요. 처음부터 끝까지 하나도 빠짐없이 꼭 알아야만 하는 건 아니랍니다.

▼ 만약 검색 결과가 잘 나오지 않는다면?

- 여러 번 검색하기 : 처음에는 문제를 잘 모르기 때문에, 검색하면서 정보를 얻게 됩니다. 얻은 정보를 가지고 검색어를 수정해보세요.
- 검색어 자동완성 & 연관검색어 사용 : '자동완성 되는 검색어' 와 검색 결과 하단에 보이는 '연관 검색어' 로 검색해보세요. 구글에서는 현재 검색 결과에서 사람들이 많이 검색한 검색어와 결과가 많은 검색어를 추천해줍니다.
- 검색어 검토하기 : 한 발 물러서서, 내가 입력한 검색어를 살펴보세요. 너무 검색어 범위가 넓지 않은지 / 여러가지 해결책 중에 내가 생각한 특정 해결책으로만 검색하고 있지 않은지!

## [1시간] Ajax 통신 기초

▼ 9) 서버→클라이언트: 데이터 표현방식 "JSON" 이해하기

- 먼저 크롬 익스텐션 JSONView를 설치해볼까요? 그럼 좀 더 예쁘게 크롬 화면에서 JSON을 볼 수 있습니다. ([링크](#))

▼ 서울시 OpenAPI(실시간 대기 환경정보) 에 접속해보기

- 아래 링크를 클릭!



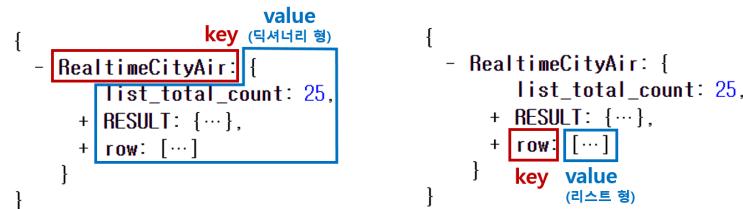
<http://openapi.seoul.go.kr:8088/6d4d776b466c656533356a4b4b5872/json/RealtimeCityAir/1/99>

- 위 링크를 클릭한 것은 '클라이언트'로 서버에 데이터를 '요청(Request)'한 것입니다. 요청에 대해서는 아래에서 더 자세하게 배워보겠습니다.

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

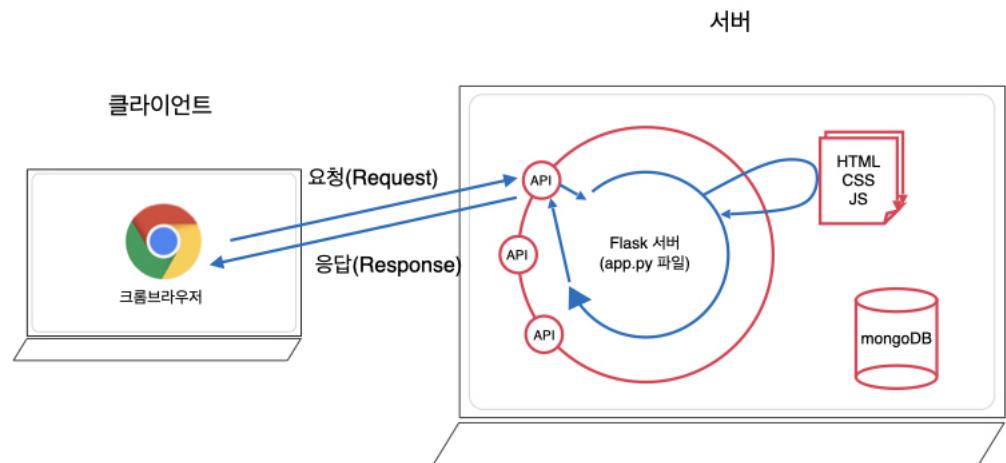


위 예제에서는 RealtimeCityAir라는 키 값에 딕셔너리 형 value가 들어가있고,  
그 안에 row라는 키 값에는 리스트형 value가 들어가있습니다.



▼ 10) 클라이언트→서버: GET 요청(Request) 이해하기

- 클라이언트는 서버의 API에 요청(Request)을 보내어 데이터(HTML, CSS, JS 포함)를 응답(Response)받습니다. 아래 그림은 HTML,CSS,JS를 응답해주는 것을 나타내고 있습니다.



은행의 창구가 API와 같은 것을 기억하시나요?

같은 예금 창구에서도 개인 고객이냐 기업 고객이냐에 따라 처리하는 것이 다른 것처럼,

**클라이언트가 요청 할 때에도, "방식(Type)"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 방식

- 아래 사이트의 URL 을 유심히 관찰해보죠.

#### 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 짹른 고정수입의 희망이다. 온 가족의 도움과 기대 속에 박사장(이선균) 집으로 향하는 기우. 글로벌 IT기업 CEO인 박사장의 저택에 도착하자 짧고 아름다운 사모

☞ <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 ? 를 기준으로 두 부분으로 쪼개집니다.  
? 기준으로 앞부분이 <서버 주소>, 뒷부분이 <영화 번호> 입니다.
  - 서버 주소: <https://movie.naver.com/movie/bi/mi/basic.nhn>
  - 영화 정보: code=161967
- 한 번 더 관찰해볼까요?

### 해리 포터와 마법사의 돌

해리 포터(다니엘 래드클리프 분)는 위압적인 버논 숙부(리차드 그리피스 분)와 냉담한 이모 페트니아 (피오나 쇼 분), 욕심 많고 벼룩없는 사촌 더즐리(해리 멜링 분) 밑에서 갖은 구박을 견디며 계단 밑 벽장에서 생활한다. 이모네 식구들 역시 해리와의 동거가 불편하기는 마찬가지. 이모 페트니아에겐 해리가 이상한(?) 언니 부부에

☞ <https://movie.naver.com/movie/bi/mi/basic.nhn?code=30688>



<https://movie.naver.com/movie/bi/mi/basic.nhn?code=30688>

- URL 분석

- 서버 주소: <https://movie.naver.com/movie/bi/mi/basic.nhn>
- 영화 정보: code=30688
- 아! 같은 서버 주소라도 <영화 번호>를 다르게 주면 다른 결과가 나오는군요! 같은 서버에 요청을 할 때에도 영화번호라는 데이터를 추가로 전달해주는 것을 알 수 있습니다.

### (GET 방식으로 데이터를 전달하는 방법)

? : 여기서부터 전달할 데이터가 작성된다는 의미입니다.

& : 전달할 데이터가 더 있다는 뜻입니다.

딕셔너리처럼 key와 value가 쌍을 이루고 있죠?

예시) [google.com/search?q=해리포터&sourceid=chrome&ie=UTF-8](http://google.com/search?q=해리포터&sourceid=chrome&ie=UTF-8)

위 주소는 google.com의 search 칭구에 다음 정보를 전달합니다!

q=해리포터 (검색어 q의 값은 해리포터)

sourceid=chrome (브라우저 정보 sourceid의 값은 chrome)

ie=UTF-8 (인코딩 정보 ie의 값은 UTF-8)

### 잠깐, 그럼 'code'라는 이름으로 영화번호 데이터를 전달해주자'는 것은 누가 정하는 것일까요?

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

백엔드 : "서버는 code로 영화번호 정보를 받는다고 정해둘게요."

프론트엔드 : "네 알겠습니다. 그러면 클라이언트에서는 code라는 이름으로 영화번호를 포함해 요청할게요."

이렇게 API를 포함해 미리 설계해 둔 약속을 따라야합니다.

### ▼ 11) Ajax(에이잭스)로 서버와 통신하기

- Ajax는 '비동기 서버 통신방식'입니다.

여기서, '서버 통신'은 웹 기초 동작원리에서 배운 요청-응답을 이야기합니다.

그리고 '비동기' 덕분에 Ajax를 사용하면, 웹 페이지 전체를 새로고침 하지 않고서도 작업을 수행할 수 있게 됩니다. 즉, 동작(event)이 일어났을 때, 전체 페이지가 아닌 일부분만을 업데이트 할 수 있게 되는 것이지요.

### Ajax로 하는 주요 작업

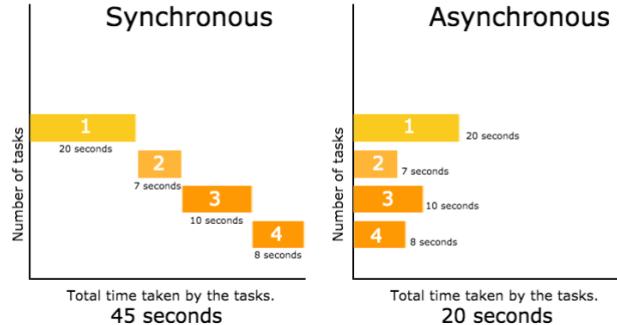
1) 웹 페이지 새로고침 없이 서버에 요청(Request)

2) 서버로부터 데이터를 받고 작업을 수행

- 그럼 이제 비동기(Asynchronous)가 무엇인지에 대해 살펴보겠습니다.

### ▼ 비동기(Asynchronous)란?

- 작업 요청을 해두고 작업 결과가 오기 전까지 다른 작업을 처리하는 것입니다. 비동기로 하게 되면 자원을 효율적으로 활용할 수 있다는 장점이 있습니다.
- Ajax를 사용하게 되면, 서버에 요청(Request)해놓고 응답이 오기 전까지 다른 작업을 처리합니다. 응답이 오면, 그 때 정해놓은 함수(콜백함수 callback Function)를 호출해 실행합니다.
- 그림으로 '비동기'를 비유와 그림으로 이해해볼까요?



위에서, 1,2,3,4번 일이 꼭 순서대로 실행될 필요가 없다면, (즉 반드시 1이 완료되고 2가 실행돼야 할 필요가 없다면) 어떻게 될까요?

요청에 대한 결과를 기다리는 동안 다른 작업을 하면 일이 더 빨리 끝나겠죠?

이직 할 때 A회사 서류 넣고, 면접 보고, 결과 나올 때까지 기다렸다가 B회사 넣으면  
→ 동기

A,B,C,D 다 서류 넣어놓고 면접 잡히는대로 처리한 뒤 아무데나 먼저 붙으면 취업뽀개기  
→ 비동기

- 이제, Ajax를 사용하면서 이해해보겠습니다!

#### ▼ 12) [💡튜터와 같이] Ajax 사용하기

- 우리는 Ajax를 jQuery를 통해 사용하겠습니다. jQuery를 사용하기 위해서는 웹 페이지에 jQuery가 import 되어야 합니다.

##### ▼ 잠깐! JQuery 로 Ajax 사용할 수 있는지 확인하기

###### 1) jQuery 가 페이지에 import 되어있지 않은 경우

- <http://google.com/> 처럼 jQuery가 import 되어있지 않은 화면에서 코드를 사용하면, jQuery가 임포트 되어있지 않기 때문에 아래와 같은 에러가 됩니다.

```
Uncaught TypeError: $ ajax is not a function
```

###### 2) jQuery 가 slim 버전인 경우

- jQuery slim 버전을 import해둔 경우엔 Ajax가 작동하지 않아요.

- 내 코드처럼 코드 내용을 수정할 수 있는 경우 아래처럼 jQuery 풀버전으로 import 해주세요.

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

- 앞서 사용한 나홀로메모장 코드([memo.html](#))에는 jQuery가 import되어있으므로, 나홀로 메모장 화면에서 [크롬 개발자 도구]에 console 탭을 사용해보겠습니다.

##### ▼ 코드 💻 ! 나홀로메모장 [memo.html](#) (파일이 없으면 이걸 사용하세요)

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

```

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

    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
        integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous"

    <!-- JS -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"
        integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPsvXusvfa0b4Q"
        crossorigin="anonymous"></script>

    <!-- 구글폰트 -->
    <link href="https://fonts.googleapis.com/css?family=Stylish&display=swap" rel="stylesheet">

    <title>스파르타코딩클럽 | 나홀로 메모장</title>

    <!-- style -->
    <style type="text/css">
        * {
            font-family: 'Stylish', sans-serif;
        }

        .wrap {
            width: 900px;
            margin: auto;
        }

        .comment {
            color: blue;
            font-weight: bold;
        }

        #post-box {
            width: 500px;
            margin: 20px auto;
            padding: 50px;
            border: black solid;
            border-radius: 5px;
        }
    </style>
    <script>
        function openClose() {
            // id 값 post-box의 display 값이 block 이면(= 눈에 보이면)
            if ($('#post-box').css('display') == 'block') {
                // post-box를 가리고
                $('#post-box').hide();
                // 다시 버튼을 클릭하면, 박스 열기를 할 수 있게 텍스트 바꿔두기
                $('#btn-post-box').text('포스팅 박스 열기');
            } else {
                // 아니면(눈에 보이지 않으면) post-box를 펴라
                $('#post-box').show();
                // 다시 버튼을 클릭하면, 박스 닫기를 할 수 있게 텍스트 바꿔두기
                $('#btn-post-box').text('포스팅 박스 닫기');
            }
        }
    </script>
</head>

<body>
    <div class="wrap">
        <div class="jumbotron">
            <h1 class="display-4">나홀로 링크 메모장!</h1>
            <p class="lead">중요한 링크를 저장해두고, 나중에 볼 수 있는 공간입니다</p>
            <hr class="my-4">
            <p class="lead">
                <button onclick="openClose()" id="btn-post-box" type="button" class="btn btn-primary">포스팅 박스 열기</button>
            </p>
        </div>
        <div id="post-box" class="form-post" style="display:none">
            <div>
                <div class="form-group">
                    <label for="post-url">아티클 URL</label>
                    <input id="post-url" class="form-control" placeholder="">
                </div>
                <div class="form-group">
                    <label for="post-comment">간단 코멘트</label>
                    <textarea class="form-control" rows="2"></textarea>
                </div>
                <button type="button" class="btn btn-primary">기사저장</button>
            </div>
        </div>
    </div>
</body>

```

```

</div>
<div id="cards-box" class="card-columns">
  <div class="card">
    
    <div class="card-body">
      <a href="#" class="card-title">여기 기사 제목이 들어가죠</a>
      <p class="card-text">기사의 요약 내용이 들어갑니다. 동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라만세 무궁화 삼&lt;br&gt;여기 기사 제목이 들어가죠</p>
      <p class="card-text comment">여기에 코멘트가 들어갑니다.</p>
    </div>
  </div>
  <div class="card">
    
    <div class="card-body">
      <a href="#" class="card-title">여기 기사 제목이 들어가죠</a>
      <p class="card-text">기사의 요약 내용이 들어갑니다. 동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라만세 무궁화 삼&lt;br&gt;여기 기사 제목이 들어가죠</p>
      <p class="card-text comment">여기에 코멘트가 들어갑니다.</p>
    </div>
  </div>
  <div class="card">
    
    <div class="card-body">
      <a href="#" class="card-title">여기 기사 제목이 들어가죠</a>
      <p class="card-text">기사의 요약 내용이 들어갑니다. 동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라만세 무궁화 삼&lt;br&gt;여기 기사 제목이 들어가죠</p>
      <p class="card-text comment">여기에 코멘트가 들어갑니다.</p>
    </div>
  </div>
  <div class="card">
    
    <div class="card-body">
      <a href="#" class="card-title">여기 기사 제목이 들어가죠</a>
      <p class="card-text">기사의 요약 내용이 들어갑니다. 동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라만세 무궁화 삼&lt;br&gt;여기 기사 제목이 들어가죠</p>
      <p class="card-text comment">여기에 코멘트가 들어갑니다.</p>
    </div>
  </div>
  <div class="card">
    
    <div class="card-body">
      <a href="#" class="card-title">여기 기사 제목이 들어가죠</a>
      <p class="card-text">기사의 요약 내용이 들어갑니다. 동해물과 백두산이 마르고 닳도록 하느님이 보우하사 우리나라만세 무궁화 삼&lt;br&gt;여기 기사 제목이 들어가죠</p>
      <p class="card-text comment">여기에 코멘트가 들어갑니다.</p>
    </div>
  </div>
</div>
</body>

</html>

```



9)에서 접속한 서울시 OpenAPI(실시간 대기 환경정보)를 ajax로 사용해보겠습니다. ([링크](#))

GET 방식으로 요청(Request)해서 API가 응답(Response)해주는 데이터 사용하기!

- 아래 코드를 [크롬 개발자 도구]에 console 탭에 적고 실행시키기

```

$.ajax({
  type: "GET", // GET 방식으로 요청한다.
  url: "http://openapi.seoul.go.kr:8088/6d4d776b466c656533356a4b4b5872/json/RealtimeCityAir/1/99",
  data: {}, // 요청하면서 함께 줄 데이터 (GET 요청시엔 비워두세요)
  success: function(response){ // 서버에서 준 결과를 response라는 변수에 담음
    console.log(response) // 서버에서 준 결과를 이용해서 나머지 코드를 작성
  }
});

```

```
}
```

#### ▼ 코드 상세 설명

- `type : "GET"` → GET 방식으로 요청한다.
- `url`: 요청할 url
- `data`: 요청하면서 함께 줄 데이터 (GET 요청 시엔 비워두세요!)



#### 다시 한번, 요청 데이터를 주는 방식

GET 요청은, url 뒤에 아래와 같이 붙여서 데이터를 줍니다.

`http://naver.com?param=value&param2=value2`

앞으로 배울 POST 요청은, `data : {}`에 넣어서 데이터를 줍니다.

`data: { param: 'value', param2: 'value2' }`

- `success`: 콜백함수(=요청에 대한 결과가 오면 실행할 함수). 요청에 성공해 응답이 정상적으로 오면, `response`에 서버의 결과 값을 담아서 함수 실행

▼ 참고. 서버 결과가 어떻게 response에 들어가나요? → 그 방식으로 미리 구현이 되어있습니다!

jQuery는 라이브러리입니다. 라이브러리는 함수 등을 편하게 사용할 수 있게 미리 코드를 구현해 놓은 것이라는 거 기억나시죠? jQuery 내부에 구현되어있는 코드([링크](#))를 살펴보면 파악할 수 있습니다. 하지만 이런 것은 보다 전문적인 개발 영역으로 우리가 우선적으로 배우고 집중해야 할 것과 거리가 있는 내용입니다. 이렇게 구현이 되어있구나! 하고 넘어가시는게 좋습니다. 앞으로 충분한 공부를 더 한 후에 탐구해보세요 😊

```
success: function(response){ // 서버에서 준 결과를 response라는 변수에 담음
    console.log(response)
}
```



API는 어떻게 요청하고 어떤 응답 데이터를 보내줄 것인지 모두 약속되어있습니다.

약속이 적힌 페이지 : 서울시 권역별 실시간 대기환경 현황([링크](#))

그 외에도 여러 데이터를 API로 사용할 수 있습니다. 이렇게 공개된 API를 OpenAPI(공개 API)라고 합니다. 서울시 OpenAPI([링크](#))처럼 여러 OpenAPI가 있습니다.

#### ▼ 13)[💡튜터와 같이] Ajax 통신 결과값(response data)을 이용하기

- 위에서 했던 Ajax 통신을 발전시켜볼게요! 아래 코드를 [크롬 개발자 도구]에 console 탭에 적고 실행시켜보세요. 튜터님이 단계별로 차근차근 설명해주실 거예요.

```
$.ajax({
    type: "GET", // GET 방식으로 요청한다.
    url: "http://openapi.seoul.go.kr:8088/6d4d776b466c656533356a4b4b5872/json/RealtimeCityAir/1/99",
    data: {}, // 요청하면서 함께 줄 데이터 (GET 요청시엔 비워두세요)
    success: function(response){ // 서버에서 준 결과를 response라는 변수에 담음
        console.log(response) // 서버에서 준 결과를 이용해서 나머지 코드를 작성
    }
})
```

- 도봉구의 미세먼지 값을 출력해보기 (관측소 도봉구는 12번째에 위치)

```
$.ajax({
    type: "GET",
    url: "http://openapi.seoul.go.kr:8088/6d4d776b466c656533356a4b4b5872/json/RealtimeCityAir/1/99",
    data: {},
    success: function(response){
        // 도봉구의 미세먼지 값만 가져와보기
        let dobong = response["RealtimeCityAir"]["row"][11];
        let guName = dobong['MSRSTE_NM'];
        let guMise = dobong['PM10'];
        console.log(guName, guMise);
    }
})
```

```
}
```

- 모든 구의 미세먼지 값을 출력해보기

```
$.ajax({
  type: "GET",
  url: "http://openapi.seoul.go.kr:8088/6d4d776b466c656533356a4b4b5872/json/RealtimeCityAir/1/99",
  data: {},
  success: function (response) {
    let cityAir = response["RealtimeCityAir"]["row"];
    for (let i = 0; i < cityAir.length; i++) {
      let mise = cityAir[i];
      let guName = mise["MSRSTE_NM"];
      let guMise = mise["PM10"];
      console.log(guName, guMise);
    }
  },
});
```

#### ▼ 복습할 때 참고! [단계별 해설] 모든 구의 미세먼지 값을 찍어보기

##### 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,
        INDEX_NM: "보통",
        INDEX_MVL: 52,
        ARPLT_MAIN: "O3"
      },
      ...
    ]
  }
}
```

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

```
$.ajax({
  type: "GET",
  url: "http://openapi.seoul.go.kr:8088/6d4d776b466c656533356a4b4b5872/json/RealtimeCityAir/1/99",
  data: {},
  success: function(response){
    let cityAir = response["RealtimeCityAir"]["row"]; // 꺼내는 부분!
    console.log(cityAir);
  }
});
```

##### 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,
    INDEX_NM: "보통",
    INDEX_MVL: 52,
    ARPLT_MAIN: "O3"
  },
  - {
    MSRDT: "202004201400",
    MSRRGN_NM: "도심권",
    MSRSTE_NM: "종로구",
    PM10: 17,
    PM25: 14,
    O3: 0.032,
    NO2: 0.018,
    CO: 0.4,
    SO2: 0.003,
    INDEX_NM: "보통",
    INDEX_MVL: 52,
    ARPLT_MAIN: "O3"
  }
]
```

row의 값을 cityAir에 담았으니, 반복문을 이용해보겠습니다!

```
$.ajax({
  type: "GET",
  url: "http://openapi.seoul.go.kr:8088/6d4d776b466c656533356a4b4b5872/json/RealtimeCityAir/1/99",
  data: {},
  success: function (response) {
    let cityAir = response["RealtimeCityAir"]["row"];
    for (let i = 0; i < cityAir.length; i++) {
      let mise = cityAir[i];
      console.log(mise);
    }
  },
});
```

3. 구 데이터에서 구 이름, 미세먼지 수치를 골라내어 출력하기

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

구 이름 키 값인 "MSRSTE\_NM", 미세먼지 수치 키값인 "PM10"의 값을 출력

```
$.ajax({
  type: "GET",
  url: "http://openapi.seoul.go.kr:8088/6d4d776b466c656533356a4b4b5872/json/RealtimeCityAir/1/99",
  data: {},
  success: function (response) {
    let cityAir = response["RealtimeCityAir"]["row"];
    for (let i = 0; i < cityAir.length; i++) {
      let mise = cityAir[i];
      let guName = mise["MSRSTE_NM"];
      let guMise = mise["PM10"];
      console.log(guName, guMise);
    }
  },
});
```

## [20분] Javascript + jQuery + Ajax 함께하기

▼ 14) [💡튜티와 같이] 서울시 OpenAPI(실시간 미세먼지 상태)를 이용하기



사용할 API :

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

API 페이지 : 서울시 권역별 실시간 대기환경 현황([링크](#))

▼ Q. 퀴즈 코드

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

<head>
  <meta charset="UTF-8">
  <title>jQuery+Ajax의 조합을 연습하자!</title>

  <!-- jQuery를 import 합니다 -->
  <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 showMiseInfo() {
            // 여기에 코드를 입력하세요
        }
    </script>

</head>

<body>
<h1>jQuery+Ajax의 조합을 연습하자!</h1>

<hr/>

<div class="question-box">
    <h2> 서울시 OpenAPI(실시간 대기환경 정보)를 이용하기</h2>
    <p>모든 관측소(MSRSTE_NM)의 미세먼지 수치(PM10)를 표기해주세요</p>
    <p>업데이트 버튼을 누를 때마다 새로 씌여져야 합니다.</p>
    <button onclick="showMiseInfo()">업데이트</button>
    <ul id="mise-info">
        <li>종구 : 82</li>
        <li>종로구 : 87</li>
        <li>용산구 : 84</li>
        <li>은평구 : 82</li>
    </ul>
</div>
</body>
</html>

```

▼ 참고) 완성본으로 결과물 확인하기

- [완성본 보기 \(링크\)](#).

▼ A. 완성 코드

```

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

<head>
    <meta charset="UTF-8">
    <title>jQuery+Ajax의 조합을 연습하자!</title>

    <!-- jQuery를 import 합니다 -->
    <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 showMiseInfo() {
            // 여기에 코드를 입력하세요
            $('#mise-info').empty();
            $.ajax({
                type: "GET",
                url: "http://openapi.seoul.go.kr:8088/6d4d776b466c656533356a4b4b5872/json/RealtimeCityAir/1/99",
                data: {},
                success: function (response) {
                    let rows = response["RealtimeCityAir"]["row"];
                    for (let i = 0; i < rows.length; i++) {
                        let guName = rows[i]['MSRSTE_NM'];
                        let guMise = rows[i]['PM10'];
                        let tempHtml = `<li>${guName} : ${guMise}</li>`;
                        $('#mise-info').append(tempHtml);
                    }
                }
            })
        }
    </script>
</head>

<body>
<h1>jQuery+Ajax의 조합을 연습하자!</h1>

<hr/>

```

```

<div class="question-box">
    <h2> 서울시 OpenAPI(실시간 대기환경 정보)를 이용하기</h2>
    <p>모든 관측소(MSRSTE_NM)의 미세먼지 수치(PM10)를 표기해주세요</p>
    <p>업데이트 버튼을 누를 때마다 지웠다 새로 씌여져야 합니다.</p>
    <button onclick="showMiseInfo()">업데이트</button>
    <ul id="mise-info">
    </ul>
</div>
</body>

</html>

```



[한걸음 더]

미세먼지 수치가 10이상인 곳은 빨갛게 보여줄까요?

▼ 참고) 완성본으로 결과물 확인하기

- [완성본 보기 \(링크\)](#)

▼ A. 완성 코드

```

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

<head>
    <meta charset="UTF-8">
    <title>jQuery+Ajax의 조합을 연습하자!</title>

    <!-- jQuery를 import 합니다 -->
    <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;
        }
        .bad {
            color: red;
        }
    </style>

    <script>
        function showMiseInfo() {
            // 여기에 코드를 입력하세요
            $('#mise-info').empty();
            $.ajax({
                type: "GET",
                url: "http://openapi.seoul.go.kr:8088/6d4d776b466c656533356a4b4b5872/json/RealtimeCityAir/1/99",
                data: {},
                success: function (response) {
                    let rows = response["RealtimeCityAir"]["row"];
                    for (let i = 0; i < rows.length; i++) {
                        let guName = rows[i]['MSRSTE_NM'];
                        let guMise = rows[i]['PM10'];
                        let tempHtml = '';

                        if (guMise > 10) {
                            tempHtml = `<li class="bad">${guName} : ${guMise}</li>`;
                        } else {
                            tempHtml = `<li>${guName} : ${guMise}</li>`;
                        }

                        $('#mise-info').append(tempHtml);
                    }
                }
            })
        }
    </script>

</head>

<body>
<h1>jQuery+Ajax의 조합을 연습하자!</h1>
<hr/>

<div class="question-box">
    <h2> 서울시 OpenAPI(실시간 대기환경 정보)를 이용하기</h2>
    <p>모든 관측소(MSRSTE_NM)의 미세먼지 수치(PM10)를 표기해주세요</p>
    <p>업데이트 버튼을 누를 때마다 지웠다 새로 씌여져야 합니다.</p>
    <button onclick="showMiseInfo()">업데이트</button>
</div>
</body>

```

```

<ul id="mise-info">
</ul>
</div>
</body>

</html>

```

## [40분] Javascript + jQuery + Ajax 연습하기

- 먼저 튜터님의 문제 설명을 듣고, 시간을 가지고 스스로 문제를 풀어보겠습니다. 그 후, 튜터님이 완성 코드 해설을 해주실 거에요.
- ▼ 15) 🖌 서울시 OpenAPI(실시간 따릉이 현황)을 이용하기



사용할 API : <http://openapi.seoul.go.kr:8088/6d4d776b466c656533356a4b4b5872/json/bikeList/1/99>

API 페이지 : 서울특별시 공공자전거 실시간 대여정보([링크](#))

- API에서 정보를 가져오는데 오래 걸릴 수도 있으니 업데이트 버튼을 누르고 조금 기다려주세요!
- ▼ Q. 퀴즈 코드

```

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

<head>
    <meta charset="UTF-8">
    <title>Javascript + jQuery + Ajax 연습하기</title>
    <!-- JQuery를 import 합니다 -->
    <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;
        }

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

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

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

```

</head>

```

<body>
<h1>Javascript + jQuery + Ajax 연습하기</h1>

<hr/>

<div class="question-box">
    <h2>서울시 OpenAPI(실시간 따릉이 현황)를 이용하기</h2>
    <p>모든 거치대(stationName)와 거치대 수(rackTotCnt), 남은 따릉이 수(parkingBikeTotCnt)를 보여주세요</p>
    <p>업데이트 버튼을 누를 때마다 데이터가 지웠다 새로 씌여져야 합니다.</p>
    <button onclick="showBikeInfo()">업데이트</button>
    <table>
        <thead>
            <tr>
                <td>거치대 위치</td>
                <td>거치대 수</td>
                <td>현재 거치된 따릉이 수</td>
            </tr>
        </thead>
        <tbody id="bike-info">
            <tr>
                <td>102. 망원역 1번출구 앞</td>
                <td>22</td>
                <td>0</td>
            </tr>
        </tbody>
    </table>

```

```

<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>

```

▼ 참고) 완성본으로 결과물 확인하기

- [완성본 보기 \(링크\)](#).

▼ A. 완성 코드

```

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

<head>
    <meta charset="UTF-8">
    <title>Javascript + jQuery + Ajax 연습하기</title>
    <!-- JQuery를 import 합니다 -->
    <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;
        }

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

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

    <script>
        function showBikeInfo() {
            $('#bike-info').empty();
            $.ajax({
                type: "GET",
                url: "http://openapi.seoul.go.kr:8088/6d4d776b466c656533356a4b4b5872/json/bikeList/1/99",
                data: {},
                success: function (response) {
                    let rows = response["rentBikeStatus"]["row"];
                    for (let i = 0; i < rows.length; i++) {
                        let rackName = rows[i]['stationName'];
                        let rackCnt = rows[i]['rackTotCnt'];
                        let bikeCnt = rows[i]['parkingBikeTotCnt'];
                        let tempHtml = `<tr>
                            <td>${rackName}</td>
                            <td>${rackCnt}</td>
                            <td>${bikeCnt}</td>
                        </tr>`;
                        $('#bike-info').append(tempHtml);
                    }
                }
            })
        }
    </script>
</head>

<body>
<h1>Javascript + jQuery + Ajax 연습하기</h1>
<hr/>

<div class="question-box">
    <h2>서울시 OpenAPI(실시간 따릉이 현황)를 이용하기</h2>
    <p>모든 거치대(stationName)와 거치대 수(rackTotCnt), 남은 따릉이 수(parkingBikeTotCnt)를 보여주세요</p>

```

```

<p>업데이트 버튼을 누를 때마다 데이터가 지웠다 새로 써여져야 합니다.</p>
<button onclick="showBikeInfo()">업데이트</button>
<table>
    <thead>
        <tr>
            <td>거치대 위치</td>
            <td>거치대 수</td>
            <td>현재 거치된 따릉이 수</td>
        </tr>
    </thead>
    <tbody id="bike-info">
    </tbody>
</table>
</div>
</body>

</html>

```



[한걸음 더]

따릉이 대수가 5대 미만인 정거장은 빨갛게 보여주면 어떨까요?

▼ 참고) 완성본으로 결과물 확인하기

- [완성본 보기 \(링크\)](#).

▼ A. 완성 코드

```

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

<head>
    <meta charset="UTF-8">
    <title>Javascript + jQuery + Ajax 연습하기</title>
    <!-- JQuery를 import 합니다 -->
    <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;
        }

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

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

        .urgent {
            color: red;
            font-weight: bold;
        }
    </style>

    <script>
        $(document).ready(function () {
            // 창을 새로고침할 때마다 실행할 함수를 여기에 입력합니다.
            $('#bike-info').empty();
        });

        function showBikeInfo() {
            $('#bike-info').empty();
            $.ajax({
                type: "GET",
                url: "http://openapi.seoul.go.kr:8088/6d4d776b466c656533356a4b4b5872/json/bikeList/1/99",
                data: {},
                success: function (response) {
                    let rows = response["rentBikeStatus"]["row"];
                    for (let i = 0; i < rows.length; i++) {
                        let rackName = rows[i]['stationName'];
                        let rackCnt = rows[i]['rackTotCnt'];
                        let bikeCnt = rows[i]['parkingBikeTotCnt'];
                        let tempHtml = '';
                        if (bikeCnt < 5) {
                            tempHtml = '<tr class="urgent">';
                            <td>${rackName}</td>
                            <td>${rackCnt}</td>
                        } else {
                            tempHtml = '<tr>';
                            <td>${rackName}</td>
                            <td>${rackCnt}</td>
                        }
                    }
                }
            });
        }
    </script>

```

```

        <td>${bikeCnt}</td>
    </tr>';
} else {
    tempHtml = `<tr>
        <td>${rackName}</td>
        <td>${rackCnt}</td>
        <td>${bikeCnt}</td>
    </tr>`;
}
$(`#bike-info`).append(tempHtml);
}
})
}
</script>
</head>

<body>
<h1>Javascript + jQuery + Ajax 연습하기</h1>
<hr/>

<div class="question-box">
    <h2>서울시 OpenAPI(실시간 따릉이 현황)를 이용하기</h2>
    <p>모든 거치대(stationName)와 거치대 수(rackTotCnt), 남은 따릉이 수(parkingBikeTotCnt)를 보여주세요</p>
    <p>업데이트 버튼을 누를 때마다 데이터가 지웠다 새로 씌여져야 합니다.</p>
    <button onclick="showBikeInfo()">업데이트</button>
    <table>
        <thead>
            <tr>
                <td>거치대 위치</td>
                <td>거치대 수</td>
                <td>현재 거치된 따릉이 수</td>
            </tr>
        </thead>
        <tbody id="bike-info">
        </tbody>
    </table>
</div>
</body>
</html>

```

#### ▼ 16) 🖌 일반 API 활용하기

- API에서 정보를 가져오는데 오래 걸릴 수도 있으니 업데이트 버튼을 누르고 조금 기다려주세요!
- ▼ 랜덤 고양이 사진 API를 이용하기 (난이도 : ★★)



사용 할 API : <https://api.thecatapi.com/v1/images/search>

API 정보 : thecatapi Doc ([링크](#))

#### 💡 힌트

- 1) 응답결과가 딕셔너리 리스트네요! 첫번째 요소에 접근하려면 `response[0][key이름]`을 사용하면 되겠군요.
- 2) "jQuery 이미지태그 src 바꾸기" 라고 구글링!

#### ▼ Q. 퀴즈 코드

```

<!Doctype html>
<html lang="ko">
    <head>
        <meta charset="UTF-8">
        <title>Javascript + jQuery + Ajax 연습하기</title>
        <!-- JQuery를 import 합니다 -->
        <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;
        }
        div.question-box > div {
            margin-top: 30px;
        }
    </style>

```

```

        }

    </style>

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

</head>
<body>
    <h1>Javascript + jQuery + Ajax 연습하기</h1>

    <hr/>

    <div class="question-box">
        <h2>랜덤 고양이 사진 API를 이용하기</h2>
        <p>예쁜 고양이 사진을 보여주세요</p>
        <p>업데이트 버튼을 누를 때마다 화면에 데이터가 지웠다 새로 써여져야 합니다.</p>
        <button onclick="showCat()">고양이를 보자</button>
        <div>
            
        </div>
    </div>
</body>
</html>

```

▼ 참고) 완성본으로 결과물 확인하기

- [완성본 보기\(링크\)](#).

▼ A. 완성 코드

```

<!Doctype html>
<html lang="ko">
<head>
    <meta charset="UTF-8">
    <title>Javascript + jQuery + Ajax 연습하기</title>
    <!-- JQuery를 import 합니다 -->
    <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;
        }

        div.question-box > div {
            margin-top: 30px;
        }
    </style>

    <script>
        function showCat() {
            $.ajax({
                type: "GET",
                url: "https://api.thecatapi.com/v1/images/search",
                data: {},
                success: function (response) {
                    let imgUrl = response[0]['url'];
                    $("#img-cat").attr("src", imgUrl);
                }
            })
        }
    </script>

</head>
<body>
    <h1>Javascript + jQuery + Ajax 연습하기</h1>

    <hr/>

    <div class="question-box">
        <h2>랜덤 고양이 사진 API를 이용하기</h2>
        <p>예쁜 고양이 사진을 보여주세요</p>
        <p>업데이트 버튼을 누를 때마다 화면에 데이터가 지웠다 새로 써여져야 합니다.</p>
        <button onclick="showCat()">고양이를 보자</button>
        <div>
            
        </div>
    </div>
</body>
</html>

```



문제를 먼저 푼 사람들은 '😎 문제뱅크'에서 추가로 문제를 골라 풀어보세요!



API를 사용하면, 데이터를 가져다 쓸 수 있거나 기능을 편하게 사용할 수 있어요.

대표적인 OpenAPI(공개API)로는

구글 API, 네이버 API, 카카오 API, 서울 열린 데이터 광장([링크](#)), 공공데이터 포털([링크](#)) 가 있습니다.

무료로 사용할 수 있는 API 모음([링크](#))

## 😎 문제뱅크

### ▼ API 활용하기

#### ▼ (1) Harry Potter 에 나오는 주문 보기 (난이도 : ★★)



사용 할 API 요청 URL : <https://www.potterapi.com/v1/spells?>

key=\$2a\$10\$LiNkiQtS86DQ8.NxC9G95.NN3.KkhNa917y/RZ6EjDILkjBaAJSL\$

API 정보: <https://www.potterapi.com/#spell-routes>

### ▼ 퀴즈 코드

```
<!DOCTYPE html>
<html lang="ko">
<head>
    <meta charset="UTF-8">
    <title>해리 포터 주문보기</title>
    <!-- JQuery를 import 합니다 -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <style type="text/css">
        div.question-box {
            color: white;
            margin: 10px 0 20px 0;
            background-image: url('https://images.pexels.com/photos/1900185/pexels-photo-1900185.jpeg?cs=srgb&dl=4k');
            background-position: center;
        }

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

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

    <script>
        $(document).ready(function () {
            // 창을 새로고침할 때마다 실행할 함수를 여기에 입력합니다.
        });

        function showSpell() {
            // 여기에 함수를 입력하세요
        }
    </script>

</head>
<body>
<h1>Javascript + jQuery + Ajax 연습하기</h1>

<hr/>
```

```

<div class="question-box">
    <h2>해리 포터 시리즈에 나오는 마법 주문 API 이용하기</h2>
    <p>모든 주문 타입(type), 주문(spell)과 효과(effect)를 화면에 보여주세요</p>
    <p>업데이트 버튼을 누를 때마다 데이터가 지웠다 새로 씌여져야 합니다.</p>
    <button onclick="showSpell()">업데이트</button>
    <table>
        <thead>
            <tr>
                <td>종류</td>
                <td>주문</td>
                <td>효과</td>
            </tr>
        </thead>
        <tbody id="spell-info">
        </tbody>
    </table>
</div>

</body>
</html>

```

- 완성본 보기(링크).

▼ 완성본 코드

```

<!DOCTYPE html>
<html lang="ko">
<head>
    <meta charset="UTF-8">
    <title>해리 포터 주문보기</title>
    <!-- JQuery를 import 합니다 -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <style type="text/css">
        div.question-box {
            color: white;
            margin: 10px 0 20px 0;
            background-image: url('https://images.pexels.com/photos/1900185/pexels-photo-1900185.jpeg?cs=srgb&dl=4k&q=80');
            background-position: center;
        }

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

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

    <script>
        $(document).ready(function () {
            // 칭을 새로고침할 때마다 실행할 함수를 여기에 입력합니다.
            $('#spell-info').empty();
        });

        function showSpell() {
            $('#spell-info').empty();
            $.ajax({
                type: "GET",
                url: 'https://www.potterapi.com/v1/spells?key=$2a$10$LiNkIQtS86DQ8.NxC9G95.NN3.KkhNa917y/RZ6EjDILkj',
                data: {},
                success: function (response) {
                    let spells = response;
                    for (let i = 0; i < spells.length; i++) {
                        let spellType = spells[i]['type'];
                        let spell = spells[i]['spell'];
                        let effect = spells[i]['effect'];

                        tempHtml = `<tr>
                            <td>${spellType}</td>
                            <td>${spell}</td>
                            <td>${effect}</td>
                        </tr>`;
                        $('#spell-info').append(tempHtml);
                    }
                }
            })
        }
    </script>

```

```

</script>

</head>
<body>
<h1>Javascript + jQuery + Ajax 연습하기</h1>

<hr/>

<div class="question-box">
    <h2>해리 포터 시리즈에 나오는 마법 주문 API 이용하기</h2>
    <p>모든 주문 타입(type), 주문(spell)과 효과(effect)를 화면에 보여주세요</p>
    <p>업데이트 버튼을 누를 때마다 데이터가 자랐다 새로 씌여져야 합니다.</p>
    <button onclick="showSpell()">업데이트</button>
    <table>
        <thead>
            <tr>
                <td>종류</td>
                <td>주문</td>
                <td>효과</td>
            </tr>
        </thead>
        <tbody id="spell-info">
        </tbody>
    </table>
</div>

</body>
</html>

```



[한걸음 더]

위험한 저주 주문은 빨갛게 보여주면 어떨까요?

#### ▼ 퀴즈 코드

```

<!DOCTYPE html>
<html lang="ko">
<head>
    <meta charset="UTF-8">
    <title>해리 포터 주문보기</title>
    <!-- JQuery를 import 합니다 -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <style type="text/css">
        div.question-box {
            color: white;
            margin: 10px 0 20px 0;
            background-image: url('https://images.pexels.com/photos/1900185/pexels-photo-1900185.jpeg?cs=srgb&dl=4k');
            background-position: center;
        }

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

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

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

    <script>
        $(document).ready(function () {
            // 창을 새로고침할 때마다 실행할 함수를 여기에 입력합니다.
        });

        function showSpell() {
            // 여기에 작성하세요
        }
    </script>
</head>

```

```

<body>
<h1>Javascript + jQuery + Ajax 연습하기</h1>

<hr/>

<div class="question-box">
    <h2>해리 포터 시리즈에 나오는 마법 주문 API 이용하기</h2>
    <p>모든 주문 타입(type), 주문(spell)과 효과(effect)를 화면에 보여주세요</p>
    <p>업데이트 버튼을 누를 때마다 데이터가 지웠다 새로 씌여져야 합니다.</p>
    <p>주문 타입이 저주(curse)면 빨간 글씨로 뜨도록 업데이트 해주세요.</p>
    <button onclick="showSpell()">업데이트</button>
    <table>
        <thead>
            <tr>
                <td>종류</td>
                <td>주문</td>
                <td>효과</td>
            </tr>
        </thead>
        <tbody id="spell-info">
        </tbody>
    </table>
</div>

</body>
</html>

```

- 완성본 보기 (링크)

▼ 완성본 코드

```

<!DOCTYPE html>
<html lang="ko">
<head>
    <meta charset="UTF-8">
    <title>해리 포터 주문보기</title>
    <!-- JQuery를 import 합니다 -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <style type="text/css">
        div.question-box {
            color: white;
            margin: 10px 0 20px 0;
            background-image: url('https://images.pexels.com/photos/1900185/pexels-photo-1900185.jpeg?cs=srgb&dl=4k&q=80');
            background-position: center;
        }

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

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

        .curse {
            color: red;
        }
    </style>
<script>
    $(document).ready(function () {
        // 창을 새로고침할 때마다 실행할 함수를 여기에 입력합니다.
        $('#spell-info').empty();
    });

    function showSpell() {
        $('#spell-info').empty();
        $.ajax({
            type: "GET",
            url: 'https://www.potterapi.com/v1/spells?key=$2a$10$LiNkiQtS86DQ8.NxC9G95.NN3.KkhNa917y/RZ6EjDILkj',
            data: {},
            success: function (response) {
                let spells = response;
                for (let i = 0; i < spells.length; i++) {
                    let spellType = spells[i]['type'];
                    let spell = spells[i]['spell'];
                    let effect = spells[i]['effect'];

                    if(spellType === 'Curse'){

```

```

        tempHtml = `<tr class="curse">\n            <td>${spellType}</td>\n            <td>${spell}</td>\n            <td>${effect}</td>\n        </tr>`;\n\n    }else {\n        tempHtml = `<tr>\n            <td>${spellType}</td>\n            <td>${spell}</td>\n            <td>${effect}</td>\n        </tr>`;\n    }\n    $('#spell-info').append(tempHtml);\n}\n})\n}\n</script>\n</head>\n<body>\n<h1>Javascript + jQuery + Ajax 연습하기</h1>\n<hr/>\n\n<div class="question-box">\n    <h2>해리 포터 시리즈에 나오는 마법 주문 API 이용하기</h2>\n    <p>모든 주문 타입(type), 주문(spell)과 효과(effect)를 화면에 보여주세요</p>\n    <p>업데이트 버튼을 누를 때마다 데이터가 지웠다 새로 씌여져야 합니다.</p>\n    <p>주문 타입이 저주(curse)면 빨간 글씨로 뜨도록 업데이트 해주세요.</p>\n    <button onclick="showSpell()">업데이트</button>\n    <table>\n        <thead>\n            <tr>\n                <td>종류</td>\n                <td>주문</td>\n                <td>효과</td>\n            </tr>\n        </thead>\n        <tbody id="spell-info">\n        </tbody>\n    </table>\n</div>\n\n</body>\n</html>\n
```

## ▼ (2) 사랑에 관한 책 보기 (난이도 : ★★★★)



사용 할 API 요청 URL : [https://openlibrary.org/subjects/love.json?published\\_in=1900-2000](https://openlibrary.org/subjects/love.json?published_in=1900-2000)

API 정보: <https://openlibrary.org/dev/docs/api/subjects>

## ▼ 퀴즈 코드

```

<!DOCTYPE html>
<html lang="ko">
<head>
    <meta charset="UTF-8">
    <title>20세기 사랑 주제의 책 보기</title>
    <!-- JQuery를 import 합니다 -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <style type="text/css">
        div.question-box {
            color: white;
            margin: 10px 0 20px 0;
            background-image: url("https://p0.pikist.com/photos/714/496/book-book-pages-novel-paperback-pitched-ope
        }
        table {
    
```

```

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

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

```

```

</style>

<script>
    $(document).ready(function () {
        // 창을 새로고침할 때마다 실행할 함수를 여기에 입력합니다.
        $('#book-info').empty();
    });

    function showBookInfo() {
        // 여기에 코드를 적어주세요
    }
</script>

```

```

</head>
<body>
<h1>Javascript + jQuery + Ajax 연습하기</h1>

<hr/>

<div class="question-box">
    <h2>20세기 사랑 주제의 책 보기</h2>
    <p>모든 책의 제목(title), 작가 이름(authors)을 화면에 보여주세요</p>
    <p>업데이트 버튼을 누를 때마다 데이터가 지웠다 새로 써여져야 합니다.</p>
    <button onclick="showBookInfo()">업데이트</button>
    <table>
        <thead>
            <tr>
                <td>책 제목</td>
                <td>작가명</td>
            </tr>
        </thead>
        <tbody id="book-info">
        </tbody>
    </table>
</div>

</body>
</html>

```

- 완성본 보기 (링크).

▼ 완성본 코드

```

<!DOCTYPE html>
<html lang="ko">
<head>
    <meta charset="UTF-8">
    <title>20세기 사랑 주제의 책 보기</title>
    <!-- JQuery를 import 합니다 -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <style type="text/css">
        div.question-box {
            color: white;
            margin: 10px 0 20px 0;
            background-image: url("https://p0.pikist.com/photos/714/496/book-book-pages-novel-paperback-pitched-opened-1000x1000.jpg");
            background-position: center;
        }

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

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

```

```

<script>
    $(document).ready(function () {
        // 창을 새로고침할 때마다 실행할 함수를 여기에 입력합니다.
        $('#book-info').empty();
    });

```

```

        });

        function showBookInfo() {
            $('#book-info').empty();
            $.ajax({
                type: "GET",
                url: "https://openlibrary.org/subjects/love.json?published_in=1900-2000",
                data: {},
                success: function (response) {
                    let works = response["works"];
                    for (let i = 0; i < works.length; i++) {
                        let title = works[i]['title'];
                        let author = works[i]['authors'][0]['name'];

                        tempHtml = `<tr> \
                            <td>${title}</td>
                            <td>${author}</td>
                        </tr>`;
                        
                        $('#book-info').append(tempHtml);
                    }
                }
            })
        }

    </script>

</head>
<body>
<h1>Javascript + jQuery + Ajax 연습하기</h1>

<hr/>

<div class="question-box">
    <h2>20세기 사랑 주제의 책 보기</h2>
    <p>모든 책의 제목(title), 작가 이름(authors)을 화면에 보여주세요</p>
    <p>업데이트 버튼을 누를 때마다 데이터가 지웠다 새로 써여져야 합니다.</p>
    <button onclick="showBookInfo()">업데이트</button>
    <table>
        <thead>
        <tr>
            <td>책 제목</td>
            <td>작가명</td>
        </tr>
        </thead>
        <tbody id="book-info">
        </tbody>
    </table>
</div>

</body>
</html>

```



[한걸음 더]

위험한 저주 주문은 빨갛게 보여주면 어떨까요?

#### ▼ 퀴즈 코드

```

<!DOCTYPE html>
<html lang="ko">
<head>
    <meta charset="UTF-8">
    <title>해리 포터 주문보기</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 {
            color: white;
            margin: 10px 0 20px 0;
            background-image: url('https://images.pexels.com/photos/1900185/pexels-photo-1900185.jpeg?cs=srgb&dl=4k');
            background-position: center;
        }

        table {
            border: 1px solid;
            border-collapse: collapse;
        }
    </style>

```

```

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

        .curse {
            color: red;
        }
    
```

```

</style>

<script>
    $(document).ready(function () {
        // 창을 새로고침할 때마다 실행할 함수를 여기에 입력합니다.
    });

    function showSpell() {
        // 여기에 작성하세요
    }
</script>

```

```

</head>
<body>
<h1>Javascript + jQuery + Ajax 연습하기</h1>

<hr/>

<div class="question-box">
    <h2>해리 포터 시리즈에 나오는 마법 주문 API 이용하기</h2>
    <p>모든 주문 타입(type), 주문(spell)과 효과(effect)를 화면에 보여주세요</p>
    <p>업데이트 버튼을 누를 때마다 데이터가 지웠다 새로 씌여져야 합니다.</p>
    <p>주문 타입이 저주(curse)면 빨간 글씨로 뜨도록 업데이트 해주세요.</p>
    <button onclick="showSpell()">업데이트</button>
    <table>
        <thead>
        <tr>
            <td>종류</td>
            <td>주문</td>
            <td>효과</td>
        </tr>
        </thead>
        <tbody id="spell-info">
        </tbody>
    </table>
</div>

</body>
</html>

```

- 완성본 보기 (링크).

▼ 완성본 코드

```

<!DOCTYPE html>
<html lang="ko">
<head>
    <meta charset="UTF-8">
    <title>해리 포터 주문보기</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 {
            color: white;
            margin: 10px 0 20px 0;
            background-image: url('https://images.pexels.com/photos/1900185/pexels-photo-1900185.jpeg?cs=srgb&dl=4k');
            background-position: center;
        }

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

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

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

```

```

        }
    </style>

    <script>
        $(document).ready(function () {
            // 창을 새로고침할 때마다 실행할 함수를 여기에 입력합니다.
            $('#spell-info').empty();
        });

        function showSpell() {
            $('#spell-info').empty();
            $.ajax({
                type: "GET",
                url: 'https://www.potterapi.com/v1/spells?key=$2a$10$LINKiQtS86DQ8.NxC9G95.NN3.KkhNa917y/RZ6EjDILkj',
                data: {},
                success: function (response) {
                    let spells = response;
                    for (let i = 0; i < spells.length; i++) {
                        let spellType = spells[i]['type'];
                        let spell = spells[i]['spell'];
                        let effect = spells[i]['effect'];

                        if(spellType === 'Curse'){
                            tempHtml = `<tr class="curse">\n                                <td>${spellType}</td>\n                                <td>${spell}</td>\n                                <td>${effect}</td>\n                            </tr>`;
                        } else {
                            tempHtml = `<tr>\n                                <td>${spellType}</td>\n                                <td>${spell}</td>\n                                <td>${effect}</td>\n                            </tr>`;
                        }
                        $('#spell-info').append(tempHtml);
                    }
                }
            });
        }
    </script>

</head>
<body>
<h1>Javascript + jQuery + Ajax 연습하기</h1>

<hr/>

<div class="question-box">
    <h2>해리 포터 시리즈에 나오는 마법 주문 API 이용하기</h2>
    <p>모든 주문 타입(type), 주문(spell)과 효과(effect)를 화면에 보여주세요</p>
    <p>업데이트 버튼을 누를 때마다 데이터가 자랐다 새로 씌여져야 합니다.</p>
    <p>주문 타입이 저주(curse)면 빨간 글씨로 뜨도록 업데이트 해주세요.</p>
    <button onclick="showSpell()">업데이트</button>
    <table>
        <thead>
        <tr>
            <td>종류</td>
            <td>주문</td>
            <td>효과</td>
        </tr>
        </thead>
        <tbody id="spell-info">
        </tbody>
    </table>
</div>

</body>
</html>

```

▼ (3) 컴퓨터에 관한 책 정보 보기 (난이도 : ★★★★★)



API 정보: <https://openlibrary.org/dev/docs/api/subjects>

API 정보를 보고 요청 URL 을 수정해보세요.

사용할 API 요청 URL 형식 : [https://openlibrary.org/subjects/주제명.json?published\\_in=%E1%BA%80%EB%A1%9C%ED%95%91%ED%8A%A4%ED%8A%A1](https://openlibrary.org/subjects/주제명.json?published_in=%E1%BA%80%EB%A1%9C%ED%95%91%ED%8A%A4%ED%8A%A1)

▼ 퀴즈 코드

```
<!DOCTYPE html>
<html lang="ko">
<head>
    <meta charset="UTF-8">
    <title>21세기 컴퓨터 주제 출판 책 보기</title>
    <!-- JQuery를 import 합니다 -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <style type="text/css">
        div.question-box {
            color: white;
            margin: 10px 0 20px 0;
            background-image: url("https://p1.pxfuel.com/preview/62/205/156/technology-hands-agreement-ok-screen-co");
            background-position: center;
        }

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

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

    <script>
        $(document).ready(function () {
            // 창을 새로고침할 때마다 실행할 함수를 여기에 입력합니다.
        });

        function showBookInfo() {
            // 여기에 코딩하세요
        }
    </script>
</head>
<body>
<h1>Javascript + jQuery + Ajax 연습하기</h1>

<hr/>

<div class="question-box">
    <h2>21세기 컴퓨터 주제 출판 책 보기</h2>
    <p>모든 책의 제목(title), 작가 이름(authors)을 화면에 보여주세요</p>
    <p>업데이트 버튼을 누를 때마다 데이터가 지웠다 새로 씌여져야 합니다.</p>
    <!-- button 을 눌렀을 때 업데이트하려면 어떻게 해야할까요? -->
    <button onclick="">업데이트</button>
    <table>
        <thead>
            <tr>
                <td>책 제목</td>
                <td>작가명</td>
            </tr>
        </thead>
        <!-- tbody 에 정보를 업데이트 하려면 어떻게 해야할까요? 힌트! 가리킨다(select) -->
        <tbody>
        </tbody>
    </table>
</div>

</body>
</html>
```

- 완성본 보기 (링크).

▼ 완성본 코드

```
<!DOCTYPE html>
<html lang="ko">
<head>
    <meta charset="UTF-8">
    <title>21세기 컴퓨터 주제 출판 책 보기</title>
    <!-- JQuery를 import 합니다 -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <style type="text/css">
        div.question-box {
            color: white;
```

```

margin: 10px 0 20px 0;
background-image: url("https://p1.pxfuel.com/preview/62/205/156/technology-hands-agreement-ok-screen-co");
background-position: center;
}

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

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

<script>
$(document).ready(function () {
    // 창을 새로고침할 때마다 실행할 함수를 여기에 입력합니다.
    $('#book-info').empty();
});

function showBookInfo() {
    $('#book-info').empty();
    $.ajax({
        type: "GET",
        url: "https://openlibrary.org/subjects/computer.json?published_in=2000-2020",
        data: {},
        success: function (response) {
            let works = response["works"];
            for (let i = 0; i < works.length; i++) {
                let title = works[i]['title'];
                let author = works[i]['authors'][0]['name'];

                tempHtml = `<tr> \
                    <td>${title}</td>
                    <td>${author}</td>
                </tr>`;
                $('#book-info').append(tempHtml);
            }
        }
    })
}

</script>

</head>
<body>
<h1>Javascript + jQuery + Ajax 연습하기</h1>

<hr/>

<div class="question-box">
    <h2>21세기 컴퓨터 주제 출판 책 보기</h2>
    <p>모든 책의 제목(title), 작가 이름(authors)을 화면에 보여주세요</p>
    <p>업데이트 버튼을 누를 때마다 데이터가 자렸다 새로 써여져야 합니다.</p>
    <button onclick="showBookInfo()">업데이트</button>
    <table>
        <thead>
        <tr>
            <td>책 제목</td>
            <td>작가명</td>
        </tr>
        </thead>
        <tbody id="book-info">
        </tbody>
    </table>
</div>

</body>
</html>

```

#### ▼ (4) 랜덤 숫자 의미부여 API를 이용하기 (난이도 : ★★)

- 가끔 API가 제대로 동작되지 않는 경우가 있어요. API 동작 확인하고 문제를 풀어보세요!
- <http://numbersapi.com/99> 를 클릭해서 화면에 JSON이 보이는지 확인



사용할 API : <http://numbersapi.com/숫자>

예) <http://numbersapi.com/99>

<http://numbersapi.com/76>

API 정보 : NUMBERS API- Let your metrics tell tales with our API of number facts([링크](#))



힌트:

url을 잘 만드는 게 중요하겠네요! 이렇게 만들면 어떨까요?

```
let number = 76
let url = "http://numbersapi.com/" + number
```

이 url 변수를 ajax 요청할 때 사용하면 되겠죠?

▼ Q. 퀴즈 코드

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

<head>
    <meta charset="UTF-8">
    <title>Javascript + jQuery + Ajax 연습하기</title>
    <!-- JQuery를 import 합니다 -->
    <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 showNumMeaning() {
            // 여기에 코드를 입력하세요
        }
    </script>
</head>

<body>
<h1>Javascript + jQuery + Ajax 연습하기</h1>

<hr/>

<div class="question-box">
    <h2>랜덤 숫자 의미부여 API를 이용하기</h2>
    <p>모든 숫자는 어떤 의미가 있습니다. 함께 살펴볼까요?</p>
    <p>아무것도 입력하지 않으면, alert를 띄워주세요.</p>
    <input type="number" id="magic-num">
    <button onclick="showNumMeaning()">이 숫자의 의미는?</button>
    <h5 id="meaning"></h5>
</div>
</body>
</html>
```

▼ 참고) 완성본으로 결과물 확인하기

- [완성본 보기\(링크\)](#).

▼ A. 완성 코드

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

<head>
    <meta charset="UTF-8">
    <title>Javascript + jQuery + Ajax 연습하기</title>
    <!-- JQuery를 import 합니다 -->
    <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 showNumMeaning() {
        let targetNum = $('#magic-num').val();

        if (targetNum == '') {
            alert('숫자를 입력하세요');
            return;
        }

        let targetUrl = 'http://numbersapi.com/' + targetNum;
        $.ajax({
            type: "GET",
            url: targetUrl,
            data: {},
            success: function (response) {
                $("#meaning").text(response);
            }
        })
    }
</script>

</head>

<body>
<h1>Javascript + jQuery + Ajax 연습하기</h1>

<hr/>

<div class="question-box">
    <h2>랜덤 숫자 의미부여 API를 이용하기</h2>
    <p>모든 숫자는 어떤 의미가 있습니다. 함께 살펴볼까요?</p>
    <p>아무것도 입력하지 않으면, alert을 띄워주세요.</p>
    <input type="number" id="magic-num">
    <button onclick="showNumMeaning()">이 숫자의 의미는?</button>
    <h5 id="meaning"></h5>
</div>
</body>

</html>

```

## 소화 타임



소화타임 = 복습 & 숙제 시작 타임!

### ▼ 리마인드! 소화 타임이 왜 중요할까요?

- 5시간의 내용을 한참 있다가 다시 보려면, 이해도 잘 안가고 까먹기도 쉬워요.
- 그래서 오늘 이 자리를 뜨기 전에! 내 것으로 만들어보는 시간을 갖습니다.
- 복습 또는 숙제를 시작하고, 튜터에게 질문해서 아직 맞춰지지 않은 퍼즐 을 맞춰가세요.
- **가장 좋은 방법은, 수업 내용을 다시 보는 게 아니라, (의외로) 숙제를 바로 시작하는 것**입니다. 숙제를 하다보면, 수업 내용은 자연스럽게 들춰보게 될거예요!

### 숙제 설명



달달한 맛 vs 짭짤한 맛 중 하나만 골라서 하시면 됩니다! :-)

[숙제 - 달달한 맛 🍫] - 웹페이지 쇼핑몰에 JS 함수 붙이기 (30분 예상)



아래 웹페이지 쇼핑몰 코드에, (1)빈칸 alert을 띄워보세요.  
- 완성본 보기: <http://spartacodingclub.shop/homework>

## ▼ 💻 코드! 원페이지 쇼핑몰

```
<!DOCTYPE html>
<html lang="ko">
<head>
    <!-- Webpage Title -->
    <title>나홀로 쇼핑몰</title>

    <!-- Required meta tags -->
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <link rel="shortcut icon" href="/static/images/favicon.png">

    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"
        integrity="sha384-Gn5Q8v8JXGibCkZcDdH+gXyW9JwYmXgIgJyUvE9xPcKX&lt;!-- JS -->
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.12.9/dist/umd/popper.min.js"
        integrity="sha384-ApNbgh8BfQ4RGVUcItVcU8E4g196E6k/P9LXu9F+q50fblK2L47jN4Oe7T4vN2" crossorigin="anonymous"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"
        integrity="sha384-JZR6Spejh4U0zWm8Jf4eFyW26eOFQ6LwQ1Z0Lc7h9S100" crossorigin="anonymous"></script>

    <!-- 구글폰트 -->
    <link href="https://fonts.googleapis.com/css?family=Stylish&display=swap" rel="stylesheet">

    <style type="text/css">
        * {
            font-family: 'Stylish', sans-serif;
        }

        .wrap {
            width: 500px;
            margin: auto;
        }

        .img {
            background-image: url('https://www.conscious-skincare.com/wp-content/uploads/2016/02/glc-candle-lit-with-new-gi');
            background-size: cover;
            background-position: center;
            width: 500px;
            height: 300px;
        }

        .info {
            margin-top: 20px;
            margin-bottom: 20px;
        }

        .user-info {
            margin: 20px 5px auto 5px;
        }

        h1, h5 {
            display: inline;
        }

        .order {
            text-align: center;
        }

        .orders {
            margin-top: 100px;
        }

        .info-temp {
            color: blue;
        }
    </style>

    <script>
        function order() {
            // 여기에 코드를 작성하세요.
            // 1-1. 이름(order-name) 부분의 값을 가져오고
            // 1-2. 만약에 빈 칸이면 alert을 띄워주세요.

            // 2-1. 수량(order-count) 부분의 값을 가져오고
            // 2-2. 만약에 빈 칸이면 alert을 띄워주세요.
        }
    </script>

```

```

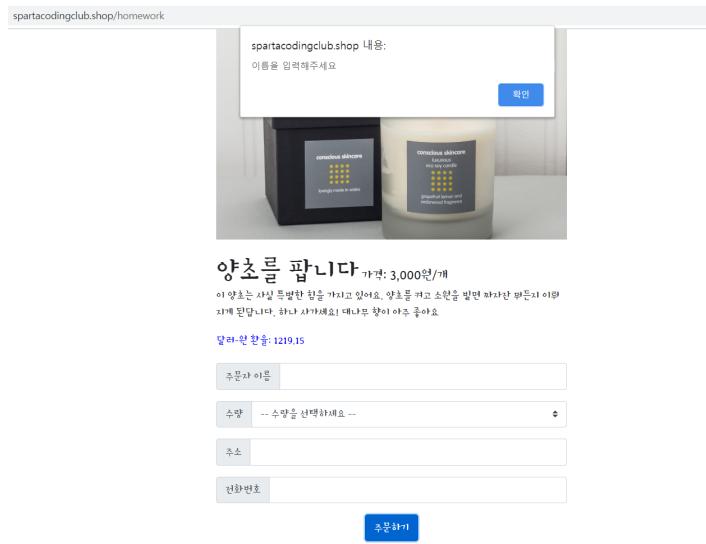
</head>
<body>
<div class="wrap">
    <div class="img"></div>
    <div class="info">
        <h1>양초를 팝니다</h1>
        <h5>가격: 3,000원/개</h5>
        <p>이 양초는 사실 특별한 힘을 가지고 있어요. 양초를 켜고 소원을 빌면 짜잔 워든지 이뤄지게 된답니다. 하나 사가세요! 대나무 향이 아주 좋아요</p>
    </div>
    <div class="info">
        <div class="input-group mb-3">
            <div class="input-group-prepend">
                <span class="input-group-text" id="basic-addon3">주문자 이름</span>
            </div>
            <input type="text" class="form-control" id="order-name" aria-describedby="basic-addon3">
        </div>
        <div class="input-group mb-3">
            <div class="input-group-prepend">
                <label class="input-group-text" for="inputGroupSelect01">수량</label>
            </div>
            <select class="custom-select" id="order-count">
                <option selected value=""> -- 수량을 선택하세요 --</option>
                <option value="1">1</option>
                <option value="2">2</option>
                <option value="3">3</option>
            </select>
        </div>
        <div class="input-group mb-3">
            <div class="input-group-prepend">
                <span class="input-group-text" id="basic-addon3">주소</span>
            </div>
            <input type="text" class="form-control" id="order-address" aria-describedby="basic-addon3">
        </div>
        <div class="input-group mb-3">
            <div class="input-group-prepend">
                <span class="input-group-text" id="basic-addon3">전화번호</span>
            </div>
            <input type="text" class="form-control" id="order-phone" aria-describedby="basic-addon3">
        </div>
        <div class="order">
            <button onclick="order()" type="button" class="btn btn-primary">주문하기</button>
        </div>
    </div>
    <div class="orders">
        <table class="table">
            <thead>
                <tr>
                    <th scope="col">이름</th>
                    <th scope="col">수량</th>
                    <th scope="col">주소</th>
                    <th scope="col">전화번호</th>
                </tr>
            </thead>
            <tbody id="orders-box">
                <tr>
                    <td>Otto</td>
                    <td>@mdo</td>
                    <td>Otto</td>
                    <td>@mdo</td>
                </tr>
            </tbody>
        </table>
    </div>
</div>
</body>
</html>

```

### (1) 빈칸 alert띄우기

- '주문하기' 버튼을 눌렀을 때 이름, 수량 중 빈 것이 있으면 alert를 띄워주세요

▼ 결과 예시

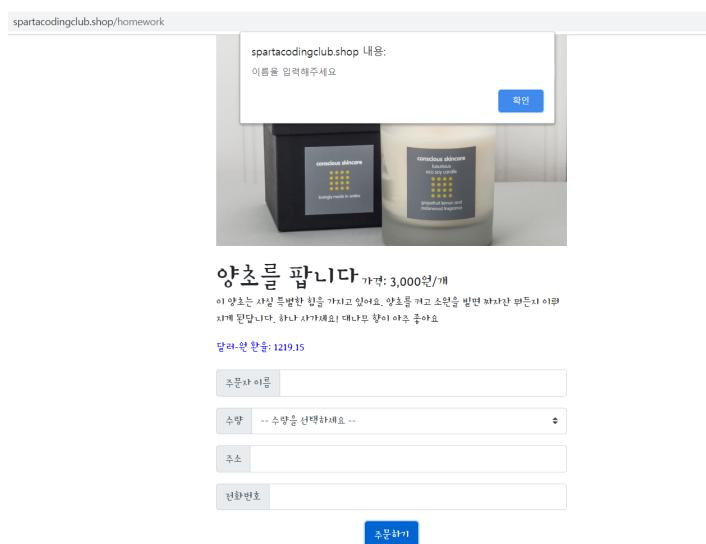


### [숙제 - 짭짤한 맛 🍜] - 나만의 원페이지 쇼핑몰에 JS함수 붙이기

 1주차에 완성한 쇼핑몰에,  
 (1)빈칸 alert를 띄우고,  
 (2)환율 정보를 넣어주세요!  
 - 완성본 보기: <http://spartacodingclub.shop/homework>

#### (1) 빈칸 alert띄우기

- '주문하기' 버튼을 눌렀을 때 이름, 수량, 주소, 전화번호 중 빈 것이 있으면 alert를 띄워주세요
- ▼ 결과 예시



#### (2) 환율 정보 넣기

- 로딩이 완료되면, 환율 API를 이용해서 환율을 표시해주세요.
- ▼ 환율 API 참고: <https://api.manana.kr/exchange/rate.json>

 쇼핑몰에 환율을 표시해주세요. 빨간 박스가 달려-원 환율입니다! (\$1 = 1,227원)

```
[  
  - {  
      date: "2020-05-07 01:33:19",  
      name: "KRW=X",  
      rate: 1,  
      timestamp: "1588782799"  
    },  
  - {  
      date: "2020-05-07 01:00:43",  
      name: "USDKRW=X",  
      rate: 1227,  
      timestamp: "1588780843"  
    },  
  - {  
      date: "2020-05-07 01:40:00",  
      name: "JPYKRW=X",  
      rate: 11.572414833617481,  
      timestamp: "1588783200"  
    }  
]
```

▼ 결과 예시



**양초를 팝니다** 가격: 3,000원/개

이 양초는 사실 특별한 힘을 가지고 있어요. 양초를 켜고 소원을 빌면 짜자잔 떠듬치 이루어지게 된답니다. 하나 사가세요! 대나무 향이 아주 좋아요.

달려-원 환율: 1219.15

주문자 이름		
수량	-- 수량을 선택하세요 --	▼
주소		
전화번호		
<b>주문하기</b>		

▼ 힌트: 로딩이 완료되면 → Ajax를 실행하는 방법

 힌트:

로딩 완료 후 환율을 써주려면 어떻게 해야할까요?  
먼저, "javascript 로딩 후 실행"이라고 검색해볼까요?

아래와 같은 친구를 찾았다면 굿! 거의 다 왔어요!

```

<script>
$(document).ready(function(){
    alert('다 로딩됐다!');
    // 여기에 활용 API Ajax 요청을 하면 되겠죠?
});

</script>

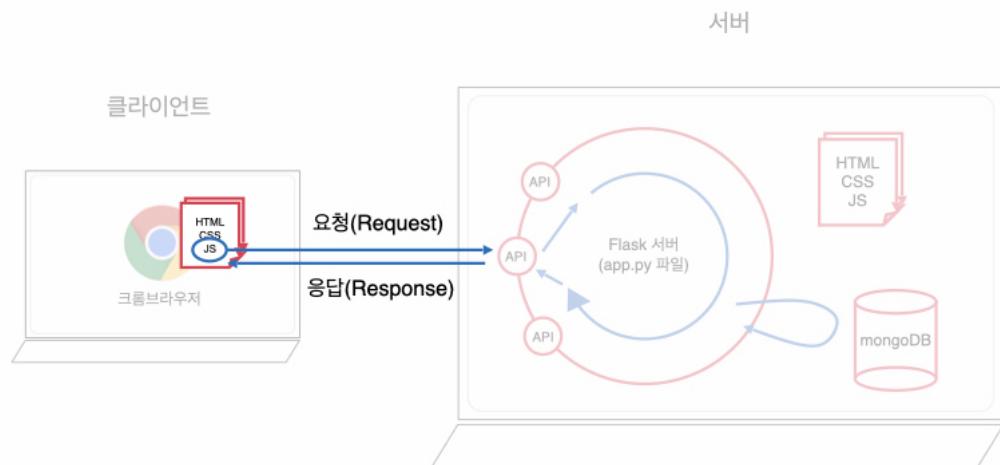
```

### [숙제 제출]

- 숙제 파일을 슬랙에 올리면 끝!

## 복습 도우미

- 오늘 배운 내용을 빠르게 복습할 수 있게 준비했어요!
- ▼ 웹 기초 동작 원리
  - HTTP: 웹은 HTTP라는 규약(규칙)을 따릅니다. url에서 `http://` 가 바로 HTTP라는 규약을 따른다는 표시예요.
  - 클라이언트 : HTTP에서 요청을 하는 쪽
  - 서버 : HTTP에서 요청을 받아 응답하는 쪽
  - HTML, CSS, JS(Javascript) 는 화면을 구성합니다.



- Ajax를 이용해 Javascript로 서버에 데이터를 요청하고, 서버가 응답한 데이터를 받아 조작하는 방법을 배웠습니다. 이 때 사용한 것이 API였습니다.

### ▼ ★ 전반부 주요 키워드

- Javascript 조건문, 반복문을 배웠습니다. 사용 패턴에 익숙해집시다!
- jQuery : 사용 패턴에 익숙해집시다! jQuery를 이용해 Javascript로 HTML을 쉽게 제어합니다.
  - jQuery는 Javascript의 라이브러리(library)로 HTML의 요소들을 조작하고 화면의 움직임을 처리하는 등 편리한 Javascript 코드를 미리 작성해놓은 뮤음. 공식 사이트([링크](#))
  - HTML 요소에 id를 부여하고, 이걸 가리켜서 조작합니다. 예. `$('#post-box').val()`
- 라이브러리(library) : 공통으로 사용할 수 있는 특정 기능들의 모음. 라이브러리를 사용하면 내가 모든 걸 다 만들지 않아도, 라이브러리에 있는 함수들을 가져다 쓸 수 있겠죠?

### ▼ ★ 후반부 주요 키워드

- GET 요청(Request)

- 클라이언트가 요청할 때, "방식(Type)"이라는 것이 존재합니다. GET 방식은 통상적으로 데이터 조회(Read)를 요청할 때 사용합니다.
- GET 요청에서 데이터를 추가로 넘겨주려고 할 때 URL에 ? 를 사용해 덧붙이고 key=value 형태로 표시합니다.
  - <https://movie.naver.com/movie/bi/mi/basic.nhn?code=161967>
  - <https://movie.naver.com/movie/bi/mi/basic.nhn?code=30688>
  - <https://google.com/search?q=%ed%95%91%ec%9d%94&sourceid=chrome&ie=UTF-8>
- JSON : 데이터 표현방식. API로 요청하면 JSON형태로 데이터를 전달해줍니다. Key:Value로 이루어져 있습니다. 자료형 Dictionary와 아주- 유사하죠.
- Ajax : 비동기 서버 통신방식. jQuery를 사용해 Ajax를 사용해봤습니다.
  - 1) 웹 페이지 새로고침 없이 서버에 요청(Request)
  - 2) 서버로부터 데이터를 받고 작업을 수행 할 수 있습니다.
- API 를 사용할 땐, 미리 정해둔 약속 을 따라야 작동합니다. 약속들은 API 페이지(문서)에 적혀있습니다.
  - 오늘 수업시간에 사용한 API 페이지 ([서울시 권역별 실시간 대기환경 현황](#) / [서울시 공공자전거 실시간 대여정보](#) / [랜덤 고양이 사진 API Doc](#) )

## [0.5시간] 스스로 소화 타임 시작

- 오늘 배운 것을 복습하고, 본격적으로 숙제를 시작합니다. 모르는 것이 있으면 튜터에게 질문합니다.

### 체크아웃

▼ "15초 체크아웃"을 진행합니다.

- 튜터는 타이머를 띄워주세요! ([링크](#)).
- 체크인처럼, 현재 본인의 감정상태와 수업후기에 관해 이야기합니다.
- 자유롭게 해주셔도 좋고, **KPT**에 맞춰해주셔도 좋아요.
  - Keep : 오늘 수업을 하면서 좋았던 것, 앞으로도 할 행동 / Problem : 아쉬워서 고쳐보면 좋을 것 / Try : 고치기 위해 내가 할 시도
  - 예) (Keep)오늘 다같이 자기소개를 해서 좋았어요. 친해진 기분! (Problem) 그런데 0주차 사전과제 중에 안한 게 있어서 쉬는 시간에 했어요. (Try)수업 전날 지난주 강의자료 확인하고 숙제하할게요!
- 튜터님은 체크아웃과 함께 출석 체크([링크](#))를 진행해주세요!
  - 출석체크 - (변동이 있다면 다시 제출해주세요!)

스파르타코딩클럽 출석체크  
<http://spartacodingclub.shop/attendance>

## [설치] - 다음 시간을 위해 미리 설치해와야 할 것들

▼ Python

▼ 윈도우

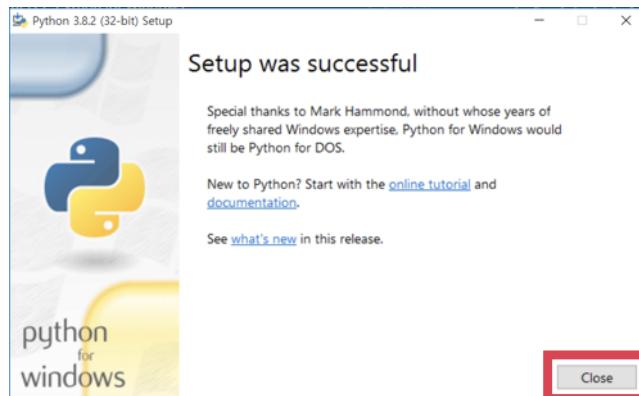
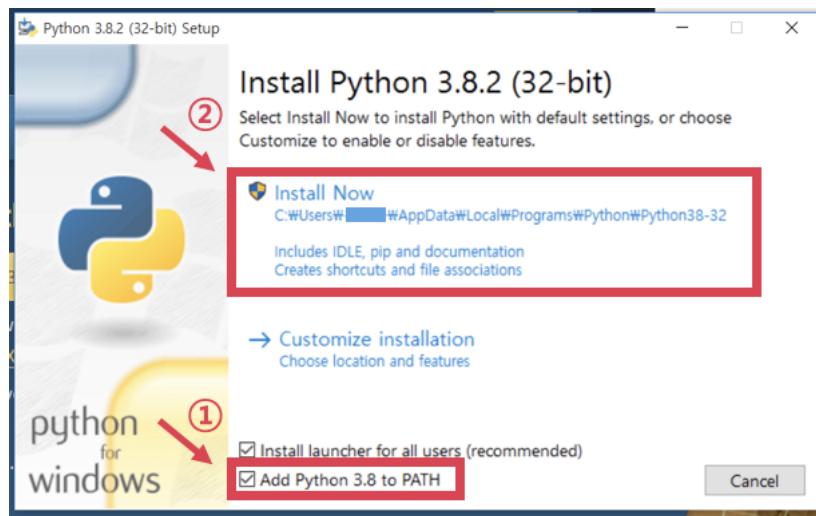
1. 버튼을 눌러 [다운로드](#) 해주세요. (3.8.x 이상의 최신 버전을 설치!)  
[설치페이지 바로가기>>](#)



2. 다운로드된 파일을 열어 설치합니다.



주의) Add Python 3.8 to PATH에 체크해야 합니다!!



▼ 맥



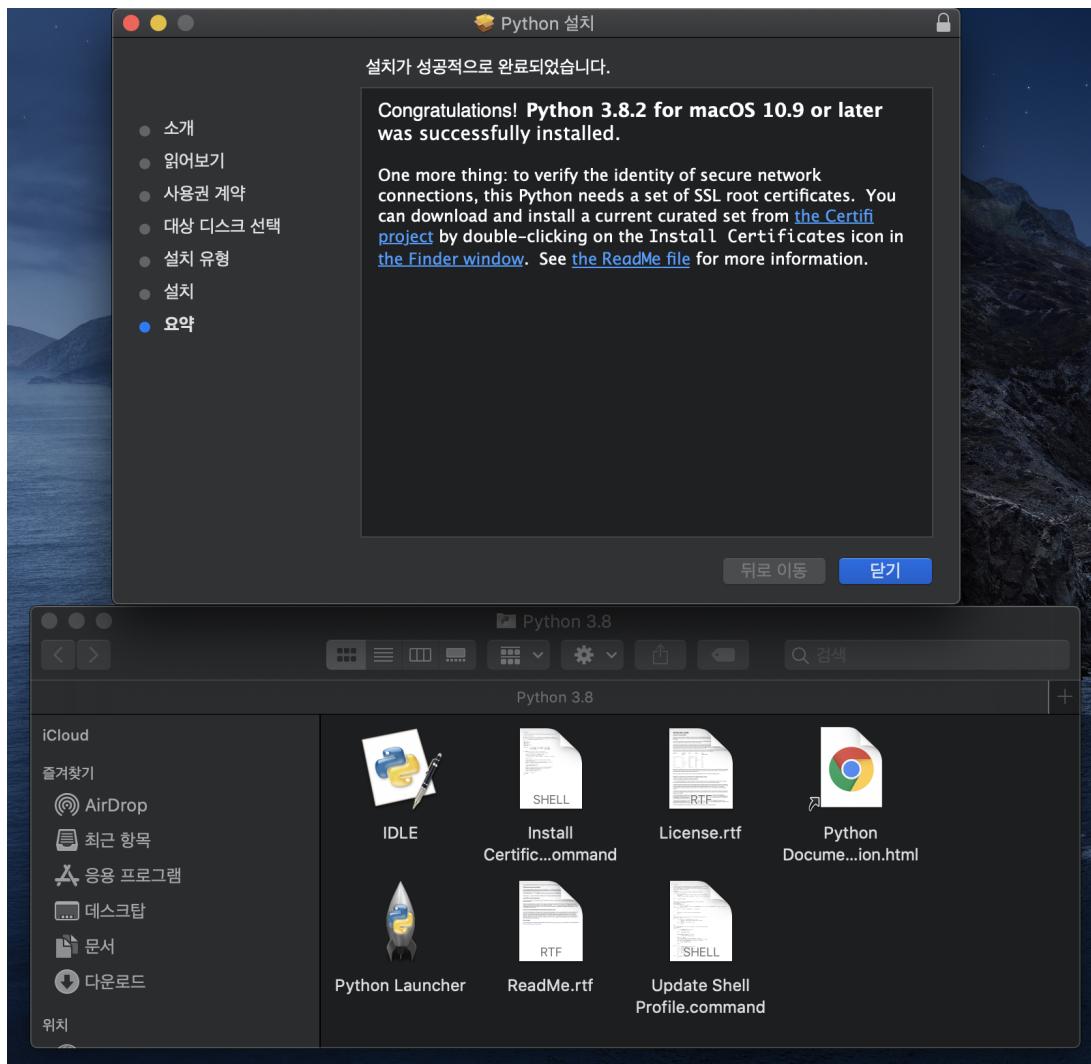
Mac에는 기본적으로 파이썬 2.x 버전이 설치되어 있지만, 버전3으로 업데이트되면서 많은 변화가 있었습니다. 본 강의에서는 3.x 버전의 파이썬이 필요하니 아래 가이드에 따라 꼭 설치해주세요.

- 버튼을 눌러 다운로드 해주세요. (3.8.x 이상의 최신 버전을 설치!)

[설치페이지 바로가기>>](#)



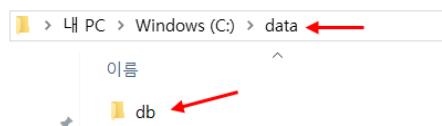
- 다운로드된 파이썬 파일을 열어 설치합니다. 아래와 같은 폴더가 나타나면 설치가 완료된 것입니다.



## ▼ mongoDB

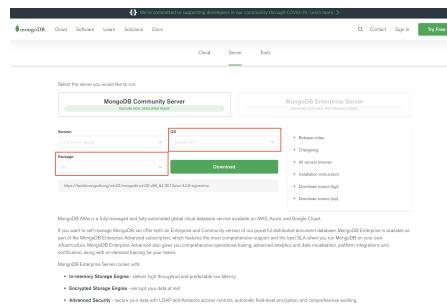
### ▼ 원도우

1. C드라이브에, 그림과 같이 data 라는 폴더를 만들고, 그 안에 db 라는 폴더를 만듭니다.

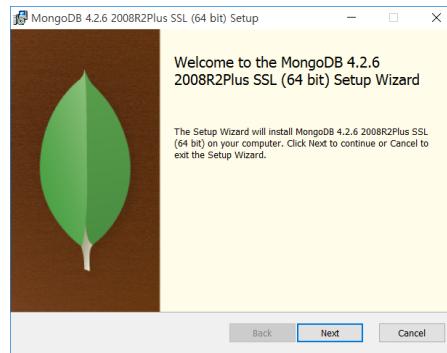


2. 링크로 가셔서, Server 탭 클릭 > MongoDB Community Server 를 클릭하시고 다음 사항을 선택한 뒤 다운로드합니다. (캡처 이미지를 클릭하면 크게 보실 수 있어요!)

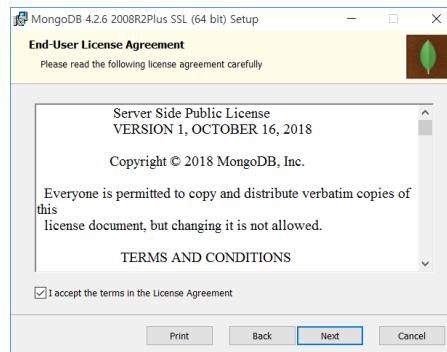
- OS : Windows x64
- Package : MSI



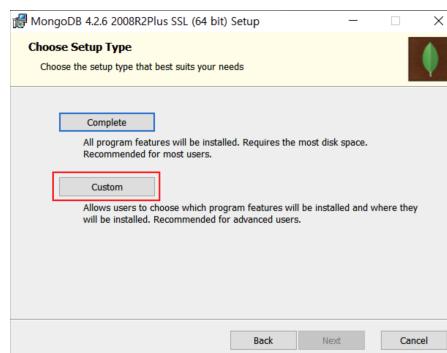
3. Next 를 클릭합니다.



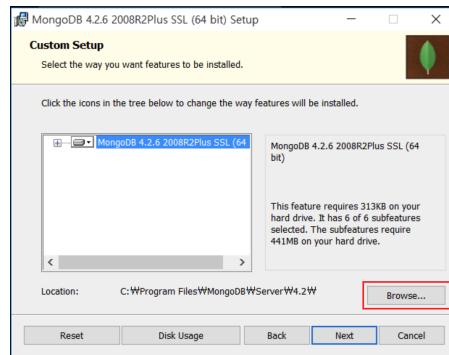
4. 약관에 동의한 뒤 Next를 클릭합니다.



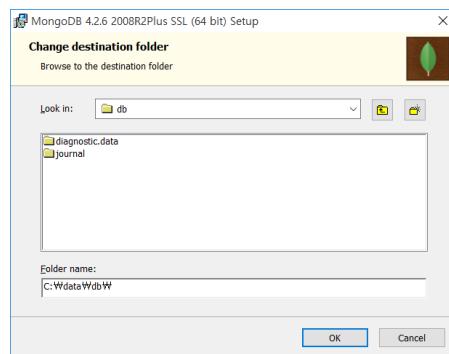
5. Custom을 클릭합니다.



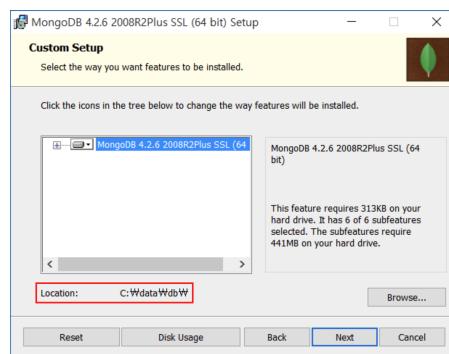
6. Browse를 클릭합니다.



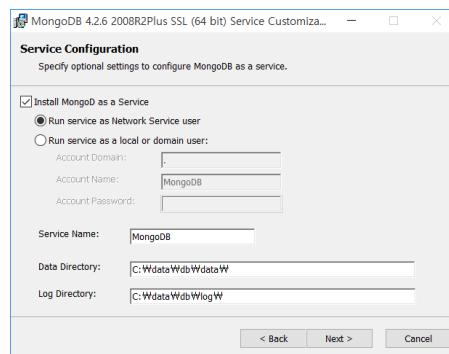
7. C:\data\db\ 를 찾아 선택하고 OK를 클릭합니다.



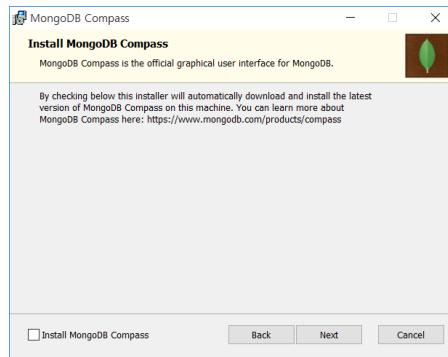
8. Location 항목이 C:\data\db\ 로 변경된 것을 확인하고 Next 를 클릭합니다.



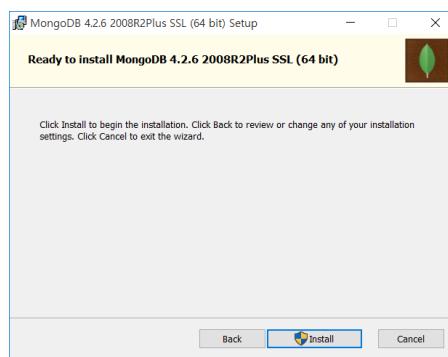
9. Next 를 클릭합니다.



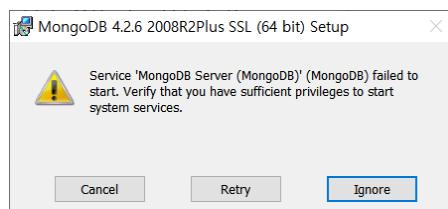
10. Install MongoDB Compass 선택을 해제하고 Next를 클릭합니다.



11. Install을 선택합니다.



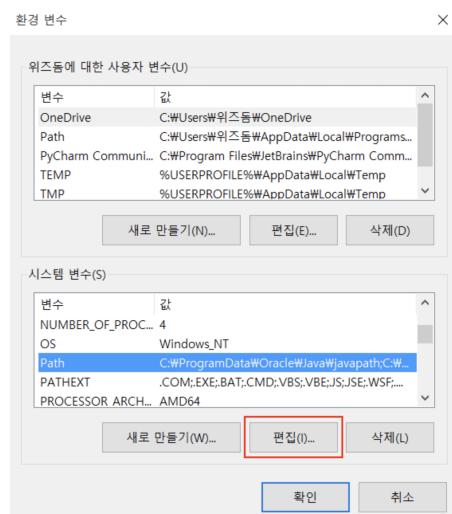
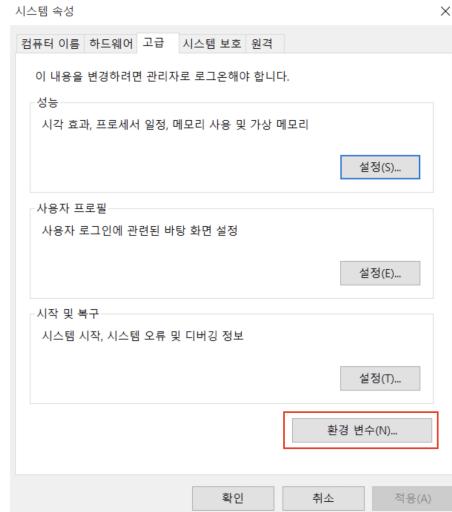
12. 아래와 같은 알림창이 뜨면, 'ignore'를 클릭해주세요.



13. 제어판 > 시스템 및 보안 > 시스템에서 [고급 시스템 설정]을 클릭합니다.



14. 환경 변수 > [시스템 변수]의 변수 목록에서 Path 선택 후 편집을 클릭한 뒤 다음 내용을 추가합니다.



C:\data\db\bin

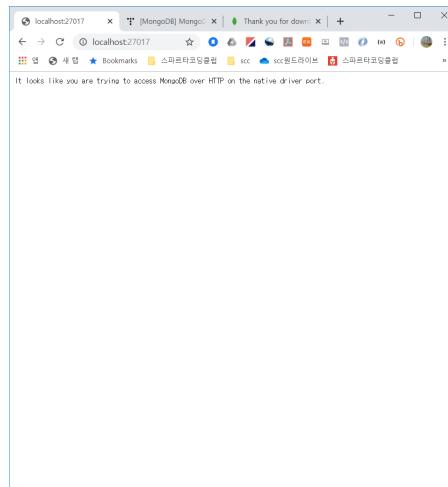
15. 원도우 키 + R 을 누른 후 cmd 를 입력하고 엔터를 누릅니다. 그리고 다음 명령어를 복사/붙여넣기하여 입력합니다.

mongod

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\scs89>mongod
2018-06-09T09:59:12.689-0700 | CONTROL  [initandlisten] MongoDB starting : pid=7504 port=27017 dbpath=C:\data\db\ 64-bit
host=BEOMLN-PARK
2018-06-09T09:59:12.689-0700 | CONTROL  [initandlisten] targetMinOS: Windows 7/Windows Server 2008 R2
2018-06-09T09:59:12.690-0700 | CONTROL  [initandlisten] db version v3.6.5
2018-06-09T09:59:12.690-0700 | CONTROL  [initandlisten] git version: a20ecdf3e3a174162052ff99913bc2ca9a839d618
```

16. 브라우저에서 localhost:27017 주소로 이동한 뒤 다음과 같은 화면이 또는 것을 확인하면 설치 완료!



## ▼ 맥

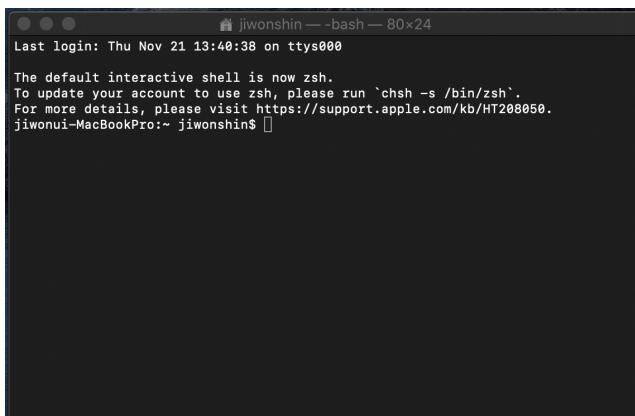
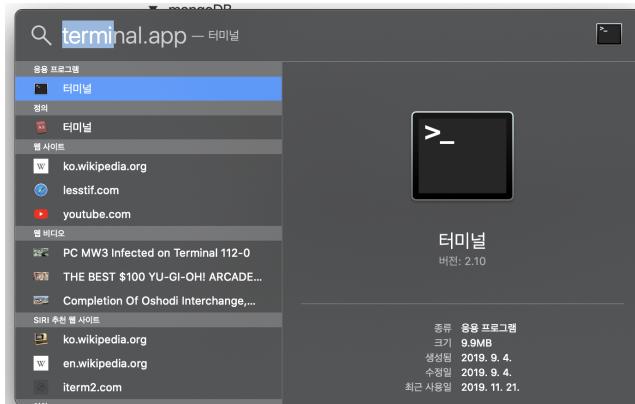
### 1. 맥북 비밀번호 설정

brew라는 프로그램을 이용해서 MongoDB를 설치합니다. brew를 설치하실 때 반드시 컴퓨터 비밀번호가 있어야 합니다. 비밀번호가 없으신 분들은 맥북의 비밀번호를 먼저 설정하시고 진행해주세요.

### 2. 터미널을 연다. spotlight에서, "terminal" 또는 "터미널"로 검색하면 됩니다!

👉 터미널이 무엇인가요?

우리는 주로 마우스를 써서 컴퓨터에 명령을 내리지만, 컴퓨터가 더 익숙한 건 '명령어'를 통한 명령입니다. 터미널은, '명령어를 쓸 수 있는 창'으로 생각해주시면 되겠습니다! 잘 여셨다면 아래와 같은 창을 만날 수 있습니다!



### 3. Homebrew 설치하기

👉 Homebrew는 무엇인가요?

새로운 것이 참 많죠? 지금부터 맥의 편리함에 빠져드신 겁니다.

Homebrew는 '다운로드 패키지'를 관리할 수 있는 툴이에요.

*brew install* 프로그램이름

을 입력하면, 프로그램을 자동으로 다운로드 받아 설치해준답니다. 엄청나죠?

터미널 창에 아래 코드를 복사, 붙여넣기 하고 엔터!

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install.sh)"
```

그러면 아래와 같은 화면이 나옵니다. 이 화면에서 다시 엔터!

```
● ● ● jiwonshin — ruby -e #!/usr/bin/ruby\012# This script installs to /usr/local only. T...
Last login: Thu Nov 21 13:40:38 on ttys000

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
[jiwonui-MacBookPro:~ jiwonshin$ /usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
==> This script will install:
/usr/local/bin/brew
/usr/local/share/doc/homebrew
/usr/local/share/man/man1/brew.1
/usr/local/share/zsh/site-functions/_brew
/usr/local/etc/bash_completion.d/brew
/usr/local/Homebrew
==> The following new directories will be created:
/usr/local/bin
/usr/local/etc
/usr/local/include
/usr/local/lib
/usr/local/sbin
/usr/local/share
/usr/local/var
/usr/local/opt
/usr/local/share/zsh
/usr/local/share/zsh/site-functions
/usr/local/var/homebrew
/usr/local/var/homebrew/linked
/usr/local/Cellar
/usr/local/Caskroom
/usr/local/Homebrew
/usr/local/Frameworks
==> The Xcode Command Line Tools will be installed.

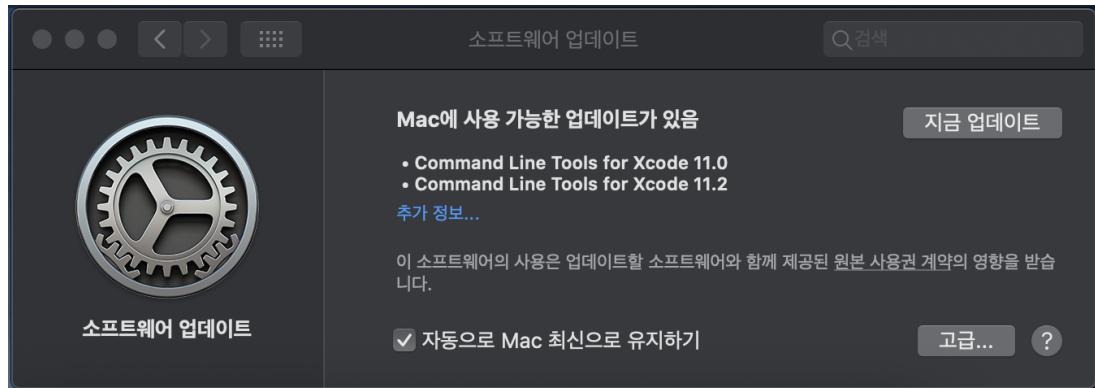
Press RETURN to continue or any other key to abort
█
```

그러면 아래와 같이 패스워드 입력칸이 나옵니다. 내 맥북 패스워드를 입력하고 엔터!

(패스워드를 입력하더라도 화면에 나타나진 않아요. 사실 잘 입력되고 있으니 걱정 마세요.)

```
Press RETURN to continue or any other key to abort
==> /usr/bin/sudo /bin/mkdir -p /usr/local/bin /usr/local/etc /usr/local/include
/usr/local/lib /usr/local/sbin /usr/local/share /usr/local/var /usr/local/opt /
usr/local/share/zsh /usr/local/share/zsh/site-functions /usr/local/var/homebrew
/usr/local/var/homebrew/linked /usr/local/Cellar /usr/local/Caskroom /usr/local/
Homebrew /usr/local/Frameworks
Password: [REDACTED]
```

엔터 누른 후에 만약 맥 소프트웨어(Xcode) 업데이트 알림이 뜨면 업데이트를 해주세요.



모든 설치가 완료되면 아래와 같은 화면이 나옵니다. Homebrew 설치 끝!

```
jiwonshin — -bash — 80x39
* [new tag]      2.1.12    -> 2.1.12
* [new tag]      2.1.13    -> 2.1.13
* [new tag]      2.1.14    -> 2.1.14
* [new tag]      2.1.15    -> 2.1.15
* [new tag]      2.1.16    -> 2.1.16
* [new tag]      2.1.2     -> 2.1.2
* [new tag]      2.1.3     -> 2.1.3
* [new tag]      2.1.4     -> 2.1.4
* [new tag]      2.1.5     -> 2.1.5
* [new tag]      2.1.6     -> 2.1.6
* [new tag]      2.1.7     -> 2.1.7
* [new tag]      2.1.8     -> 2.1.8
* [new tag]      2.1.9     -> 2.1.9
HEAD is now at fb978a786 Merge pull request #6759 from daviddrapp/patch-1
==> Homebrew is run entirely by unpaid volunteers. Please consider donating:
https://github.com/Homebrew/brew#donations
==> Tapping homebrew/core
Cloning into '/usr/local/Homebrew/Library/Taps/homebrew/homebrew-core'...
remote: Enumerating objects: 5087, done.
remote: Counting objects: 100% (5087/5087), done.
remote: Compressing objects: 100% (4880/4880), done.
remote: Total 5087 (delta 47), reused 319 (delta 10), pack-reused 0
Receiving objects: 100% (5087/5087), 4.15 MiB | 2.99 MiB/s, done.
Resolving deltas: 100% (47/47), done.
Tapped 2 commands and 4868 formulae (5,129 files, 12.8MB).
Already up-to-date.
==> Installation successful!

==> Homebrew has enabled anonymous aggregate formulae and cask analytics.
Read the analytics documentation (and how to opt-out) here:
https://docs.brew.sh/Analytics

==> Homebrew is run entirely by unpaid volunteers. Please consider donating:
https://github.com/Homebrew/brew#donations
==> Next steps:
- Run `brew help` to get started
- Further documentation:
  https://docs.brew.sh
jiwonui-MacBookPro:~ jiwonshin$ 
```

#### 4. mongoDB 설치하기

터미널 창에 아래 코드 입력 후 엔터 (한줄씩 복사-붙여넣기 하세요)

```
brew tap mongodb/brew
```

```
brew install mongodb-community
```

몽고DB 설치가 완료되면 다음 화면이 나옵니다.

```

jaronshin — -bash — 80x39
Receiving objects: 100% (5087/5087), 4.15 MiB | 2.99 MiB/s, done.
Resolving deltas: 100% (47/47), done.
Tapped 2 commands and 4868 formulae (5,129 files, 12.8MB).
Already up-to-date.
==> Installation successful!

==> Homebrew has enabled anonymous aggregate formulae and cask analytics.
Read the analytics documentation (and how to opt-out) here:
https://docs.brew.sh/Analytics

==> Homebrew is run entirely by unpaid volunteers. Please consider donating:
https://github.com/Homebrew/brew#donations
==> Next steps:
- Run `brew help` to get started
- Further documentation:
  https://docs.brew.sh
jiwonui-MacBookPro:~ jiwonshin$ brew tap mongodb/brew
==> Tapping mongodb/brew
Cloning into '/usr/local/Homebrew/Library/Taps/mongodb/homebrew-brew'...
remote: Enumerating objects: 13, done.
remote: Counting objects: 100% (13/13), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 13 (delta 4), reused 5 (delta 1), pack-reused 0
Unpacking objects: 100% (13/13), done.
Tapped 6 formulae (45 files, 59.8KB).
jiwonui-MacBookPro:~ jiwonshin$
jiwonui-MacBookPro:~ jiwonshin$ brew install mongodb-community
==> Installing mongodb-community from mongodb/brew
==> Downloading https://fastdl.mongodb.org/osx/mongodb-macos-x86_64-4.2.1.tgz
#####
100.0%
==> Caveats
To have launchd start mongodb/brew/mongodb-community now and restart at login:
  brew services start mongodb/brew/mongodb-community
Or, if you don't want/need a background service you can just run:
  mongod --config /usr/local/etc/mongod.conf
==> Summary
🍺 /usr/local/Cellar/mongodb-community/4.2.1: 21 files, 273.5MB, built in 1 minute 26 seconds
jiwonui-MacBookPro:~ jiwonshin$ 
```

## 5. mongoDB 실행해보기

```
brew services start mongodb-community
```

아래와 같은 화면이 나옵니다.

```
(base) Byeongkwanui-MacBook-Pro:~ byeongkwan$ brew services start mongodb-community
==> Successfully started `mongodb-community` (label: homebrew.mxcl.mongodb-commu
```

**!** 앗, 혹시 brew services start mongodb-community 실행이 안되시나요?

### ▼ 안될 경우 대처 방안

아래 내용을 차례대로 실행해보세요

```
brew update
```

```
brew services start mongodb-community
```

## 6. 마지막으로, mongoDB 실행이 잘 되었는지 확인하기



<http://localhost:27017>에 접속했을 때,  
아래와 같은 화면을 만나면 잘 된 것입니다!

← → ⌛ ⓘ localhost:27017

It looks like you are trying to access MongoDB over HTTP on the native driver port.

이제 mongoDB 설치가 모두 완료되었습니다.

#### ▼ Robo3T

- [다운로드\(링크\)](#) 후 설치
- Download Robo 3T 를 받으세요! ( ⇒ Studio 3T를 받지 마세요!)

The most powerful option  
Meet Studio 3T, Robo 3T's big brother.

Studio 3T is the professional GUI and IDE for MongoDB available for Windows, Mac, and Linux. Explore and manage your data faster with features like query building, data exploration, aggregation and data comparison, import/export, code generation, and more – with or without the knowledge of the MongoDB query language.

Download Studio 3T

The latest version  
Robo 3T (formerly Robomongo)

Robo 3T 1.3 brings you support for MongoDB 4.0 and SCRAM-SHA-256, an upgraded mongo shell, support for importing from MongoDB SRV connection strings, among many other fixes and improvements.

Download Robo 3T

#### ▼ [윈도우만] Git bash 설치하기

- 다음 주 수업에서 Git bash라는 프로그램이 필요합니다. 꼭 설치해주세요!
- 설치하기: ([링크](#))