



1 ~ 5 주차 요약



"이게 어디 있었 더라", "또 어떤 걸 할 수 있습니다?"
해야합니다. 이름으로 치트 시트!

주차 별 코드 요약

HTML, CSS, 자바 스크립트

▼ 1) HTML 기본 골격



아무것도 아닙니다.

```
< html > < head > < 제목 > 스파르타 코딩 클럽의 페이지 </ title > <!--헤드  
및 수신 갈등 및 구성 요소-> </ 헤드 > < 바디 > <!--바디 수신 -> </ body >  
</ html >
```

HTML ▼

▼ 2) HTML 클래스 선택자

📌 선택해야합니다.

1. mytitle이라는 클래스가 스타일입니다.
2. mytitle이라는 클래스와 작은 글꼴이있는 클래스

```
<!DOCTYPE html> <html lang = " en " > <head> <메타 문자 집합 = "
UTF-8 " > <제목> 로그인 !!! </제목> <스타일> 신체 { 배경색 : 검정;
색상 : 흰색; } .mytitle { 색상 : 노랑; } .small-font { 글꼴 크기 : 10px;
} </스타일> </head> <body> <h1 class = " mytitle small-font " >
로그인 페이지입니다. </h1> <div class = " mytitle " > 신분증: <입력
유형 = " 텍스트 " > PW : <입력 유형 = " 텍스트 " > </div> <버튼> 로
그인하기 </버튼> </본문> </html>
```

HTML ▾

▼ 3) 자주 쓰이는 CSS 모음

📌 참고 : <http://www.walterz.net/2017/08/18/ 자주-사용되는-css-속성/>

배경 정보 배경색 배경 이미지 배경 크기 사이즈 폭 신장 따뜻하다 글꼴 크기 글
꼴 무게 서체 가족 색깔 간격 여유 심

CSS ▾

▼ 4) 부트 스트랩 기본 골격

🔗 HTML 페이지를 선택하십시오.

```
<!doctype html> < html lang = " en " > < 헤드 > <!--웹 페이지 제목--> <
제목 > 안녕하세요, 세계! </ 제목 > <!--필수 메타 태그--> < 메타 문자셋 = "
utf-8 " > < 메타 이름 = " 뷰포트 " content = " width = device-width,
initial-scale = 1, shrink-to-fit = no " > <!--부트 스트랩 CSS--> < 링크
rel = " 스타일 시트 " href = "
https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css "
무결성 = " sha384-Gn5384xqQ1aowXA +
058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW / dAiS6JXm " crossorigin = " 익
명 " > <!--JS--> < script src = "
https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js " > </
script > < script src = " https :
//cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js "
무결성 = " SHA384 - ApNbgh9B + Y1QKtv3Rn7W3mgPxhU9K /
ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q " crossorigin = " 익명 " > </ 스크립
트 > < 스크립트 src = "
https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js "
무결성 = " SHA384 - JZR6Spejh4U02d8jOt6vLEHfe /
JQGiRRSQQxSFFWpi1MquVdAyjUar5 + 76PVCmYl " crossorigin = " 익명 " > </
스크립트 > </ head > < body > < h1 > 안녕하세요, 세계! </ h1 > </ 본문 >
</ html >
```

HTML ▾

▼ 5) 부트 스트랩 가져 오기

🔗 참고 : <https://getbootstrap.com/docs/4.0/components/alerts/>
compenents의 해당 코드를 <body> </ body>에 있습니다.

```
< div class = " jumbotron " > < h1 class = " display-4 " > 나홀로 링크
메모장! </ H1 > < P의 클래스 = " 리드 " > 중요한 링크를 저장해두고, 나중에
볼 수있는 공간입니다 </ P > < 열연 클래스 = " MY-4 " > < P의 클래스 = "
리드 " > < btn btn-primary btn-lg " href = " # " role = " button " > 포
스팅 박스 > </ a > </ p > </ div >
```

HTML ▾

▼ 6) 1 주차 완성 코드

```
<!doctype html> < html lang = " en " > < 헤드 > <!--웹 페이지 제목--> < 제목
필수 메타 태그--> < 메타 문자셋 = " utf-8 " > < 메타 이름 = " 뷰포트 " conten
initial-scale = 1, shrink-to-fit = no " > <!--부트 스트랩 CSS--> < 링크 rel
https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css " 무
058RXPgPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW / dAiS6JXm " crossorigin = " 익명
https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js " > </ sc
//cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js " 무결
Y1QKtv3Rn7W3mgPxhU9K / ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q " crossorigin =
트 src = " https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.mi
JZR6Spejh4U02d8jOt6vLEHfe / JQGiRRSQQxSfFWpi1MquVdAjyUar5 + 76PVCmYl " crc
> <!-- 구글 폰트 --> < 링크 HREF = " https://fonts.googleapis.com/css?family
" 스타일 시트 " > <!--style--> < style type = " text / css " > * { 글꼴 모음
{ 너비 : 900px; 여백 : 자동; } .comment { 파란색; 글꼴 무게 : 굵게; } .form-
2rem; 마진 : 2rem auto; 경계색 : # e9ecef; 국경 반경 : 0.3rem; 경계 : 고체;
</ head > < body > < div class = " 랩 " > < div class = " jumbotron " > <
링크 메모장! </ h1 > < p class = " lead " > 중요한 링크를 저장해주세요, 어디
hr class = " < P의 클래스 = " 리드 " > < 클래스 = " BTN BTN - 우선 BTN-LG "
스팅 박스 열기 </ > </ P > </ DIV > < DIV 클래스 = " 폼 포스트 " id = " 포스
" 폼 그룹 " > < 라벨 용 = " exampleFormControlInput1 " > 아티클 URL </ 라벨
리 = " " > </ DIV > < DIV 클래스 = " 폼 그룹 " > < 에 대한 레이블 = " exampl
불구하고 </ label > < textarea class = " form-control " rows = " 2 " > </ .
" button " class = " btn btn-primary " > 기사 저장 </ button > </ div > </
>> < div class = " card " > < img class = " card-img-top " src = "
https://www.eurail.com/content/dam/images/eurail/Italy%20CP%20Promo%20Blc
.jpg " alt = " 카드 이미지 캡 " > < div class = " card-body " > < h5 class =
여기 죠 </ h5 > < p 클래스 = " card-text " > 기사의 요약 내용이 들어갑니다. 동
만들어 보우 하사 우리나라 만세 무궁화 삼천리 화려 강산 ... </ p > < p class =
있습니다. </ p > </ div > </ div > < div class = " card " > < img class =
https://www.eurail.com/content/dam/images/eurail/Italy%20CP%20Promo%20Blc
" alt = " 카드 이미지 캡 " > < div 클래스 = " card-body " > < h5 class = "
기 죠 </ h5 > < p class = " card-text " > 기사의 요약 내용이 도착되었습니다.
궁화 삼천리 화려 강산 ... </ p > < p 클래스 = " 카드 텍스트 설명 " > 여기에 나
> </ div > < div class = " card " > < img class = " card-img-top " src = "
https://www.eurail.com/content/dam/images/ eurail / Italy % 20CP % 20Prom
20Block.adaptive.767.1535627244182.jpg " alt = " 카드 이미지 캡 " > < " > <
기 기사 제목이 아닌 죠 </ h5 > < p class = " card-text " > 기사의 요약 내용
이 마르고 하사 우리나라 만세 무궁화 삼천리 화려 강산 ... </ p > < p class = '
던가 들어갑니다. </ p > </ div > </ div > <div 클래스 = " 카드 " > < img cl
https://www.eurail.com/content/dam/images/eurail/Italy%20CP%20Promo%20Blc
" alt = " 카드 이미지 캡 " > < div class = " card-body " > < h5 class = "
디서 죠 </ h5 > < p class = "카드 텍스트 " > 기사의 요약 내용이 들어갑니다.
만들어 보우 하사 우리나라 만세 무궁화 삼천리 화려 강산 ... </ p > < p class =
있습니다. </ p > </ div > </ div > < div class = " card " > < img class =
https://www.eurail.com/content/dam/images/ 유레일 / 이탈리아 % 20CP % 20Pr
20Block.adaptive.767.1535627244182. alt = " 카드 이미지 캡 " > < div 클래스
```

```
" 카드 제목 " > 여기 기사 과목이 있습니다 죠 </ h5 > < p class = " card-text
다. 동해 물과 백두산이 마르고 튼튼하게 만들어 보우 하사 우리나라 만세 무궁화
class = " card-text comment " > 여기에 있습니다. > </ div > </ div > < div
" card-img-top " src = " https://www.eurail.com/content/dam/images/eurail/
20Block.adaptive.767.1535627244182.jpg " alt = " 카드 이미지 캡 " > < div <
class = "카드 제목 " > 여기 기사 제목이 아닙니다 죠 </ h5 > < p class = " ca
어갑니다. 동해 물과 백두산이 마르고 튼튼하게 만들어 보우 하사 우리나라 만세 무
p class = " card-text comment " > 여기에 있습니다. </ p > </ div > </ div >
html >
```

HTML ▾

▼ 7) 자바 스크립트 기초 문법 (다수, 자료형, 함수, 반복문, 조건문, + 결합)

• 변수 선언

```
let first_name = 'bob' // 문자 let age = 24 // 숫자 let is_even = true // 참 거짓
```

자바 스크립트 ▼

• 자료형

```
번호 하자 과일 = [ '감' , '배' , '귤' ] 콘솔 . log ( fruits [ 0 ] )  
// '감'출력 과일 . push ( '수박' ) // 요소 추가 console . log ( fruits [ 3 ] ) // '수박'출력
```

자바 스크립트 ▼

```
덕셔너리 let person = { 'name' : 'bob' , 'age' : 24 } person [ 'height' ] = 178 // 'height'라는 키가 있습니다. 콘솔 . log ( person [ 'name' ] ) // '밥'출력
```

자바 스크립트 ▼

• 함수

```
'두 수를 더합니다' 라는 알림이 먼저 뜨고 , num3에는 7 이 들어감 함수 합 ( num1 , num2 ) { alert ( '두 수를 더합니다' ) return num1 + num2 } num3 = 합 하자 ( 3 , 4 )
```

자바 스크립트 ▼

• 반복문

```
하자 이름 = [ '밥' , '케이' , '주스' ] 대 ( 하자 난 = 0 ; I < 이름 . 길이 ; I ++ ) { 콘솔 . 로그 ( 이름 [ i ] ) ; // i가 0부터 2까지 돌 }
```

자바 스크립트 ▼

• 조건문

```
만약 , 다른 경우 , 다른 function compare_num ( num ) { if ( num > 90 ) { alert ( '90보다 커요!' ) } else if ( num > 80 ) { alert ( '80보다 커요!' ) } else { alert ( '80보다 커요!' ) } }
```

자바 스크립트 ▼

```
AND 조건은 "&&" 로 , OR 조건은 "||" 로 표현 예 ) // 나이가 20보다 크고 30보다 작은 경우에 해당 평션 check_adult ( 연령 ) { 경우 ( 연령 >
```

```
20 && 연령 < 30 ) { 반환 참 ; } else { return false ; } } // 나이가
20보다 작거나, 30보다 크면 true function check_adult ( age ) { if (
나이 < 20 || age > 30 ) { return true ; } else { return false ; } }
```

자바 스크립트 ▾

• 결합

만약에 , 0 부터 N - 1 까지 더하는 함수를 만들고 싶다면 ? 함수

```
get_sum ( N ) { 하자 결과 = 0 에 대한 ( 하자 난 = 0 ; I < N ; I를 ++
) { 결과 = 결과 + I ; // 반복문을 돌면서, 결과 에다 // 결과 + = i; 그
래도 쓸쓸하다. } 결과 반환 } 콘솔 . log ( get_sum ( 10 ) ) // 45를 출
력
```

자바 스크립트 ▾

리스트의 값을 모두 출력하십시오 . let names = ['Bob' , 'Kay' ,
'Juice'] for (let j = 0 ; j < names . length ; j ++) { // names
리스트의 범위가 돌아 왔습니다! 콘솔 . 로그 (이름 [j]) ; }

자바 스크립트 ▾

반복문과 조건 문의 결합 리스트의 값을 돌면서 나이가 20 세 가지 하자
사람들 = [{ '이름' : '밥' , '나이' : 24 } , { '이름' : '제임스' ,
'연령' : 18 } , { '이름' : '케이' , '나이' : 30 } , { 'name' :
'john' , 'age' : 35 }] for (let i = 0 ; i < people . length ; i
++) { if (people [i] ['age'] < 20) { 콘솔 . 로그 (사람 [i]
) ; } }

자바 스크립트 ▾

자바 스크립트, jQuery, Ajax

▼ 1) HTML 버튼과 자바 스크립트 연결해보기

👉 버튼의 onclick 함수에 JavaScript 스크립트의 hey () 함수를 부르고 있습니다.

```
1 , 2 , 3 , 4 . . . 함수 클릭보기 // 자바 중요 부분 < script > num = 0으로
하자; 함수 hey () { num + = 1 ; if ( num < 20 ) { alert ( "안녕!" + num
+ "번 누르는데 네요!" ) ; } else { alert ( '앗. 그만 누르세요 이제!' ) ;
} } </ 스크립트 > // 버튼 부분 < button onclick = " hey () " class = "
btn btn-primary " > 기사 저장 </ button >
```

자바 스크립트 ▾

▼ 2) HTML 로딩 후 자바 스크립트 실행하기

```
< 스크립트 > $ (document) .ready (function () { 목록 ( ) ; } ); function  
listing () { alert ( '이 함수는 HTML 로딩 후 불러있다.' ) } </ 스크립트 >
```

자바 스크립트 ▼

▼ 3) jQuery 임포트하기

👉 참고 : https://www.w3schools.com/jquery/jquery_get_started.asp
<head>와 </ head> 사이에 있습니다.
(부트 스트랩 기본 템플릿에 이미 다음과 같이 임포트 해두 템플릿)

```
< script src = "  
https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js " > </  
스크립트 >
```

HTML ▼

▼ 4) 자주 사용하는 jQuery 모음

▼ 1. 입력 박스의 값을 가져와보기

```
// 조작하고 싶은 부분에 ID 값을주기 < DIV 클래스 = " 폼 포스트 " > <
형식 > < DIV 클래스 = " 폼 그룹 " > < 라벨 에 대한 = "
exampleFormControlInput1 " > 아티클 URL </ 라벨 > < 입력 id = "
posting-url " class = " 양식 제어 " 자리 표시 자 = " " ></ div > <
DIV 클래스 = " 폼 그룹 " > < 라벨 용 = " exampleFormControlTextarea1
" > 간단 코멘트 </ 라벨 > < 입력 클래스 = " 폼 제어 " 자리 = " " > </
DIV > < 버튼 타입 = " 버튼 " 클래스 = " btn btn-primary " > 기사 저장
</ 버튼 > </ 양식 > </ div >
```

자바 스크립트 ▼

```
// 크롬 개발자 도구 콘솔 창에서 쳐보기 // id 값이 posting-url 인 위치
를 메뉴 고, val ()로 값을 선택 다. $ ( '# posting-url' ) . val ( ) ;
```

자바 스크립트 ▼

▼ 2. div 보이기 / 자극

```
// 조작하고 싶은 부분에 ID 값을주기 < DIV ID = " 게시 박스 " 클래스 =
" 폼 포스트 " > < 형식 > < DIV 클래스 = " 폼 그룹 " > < 라벨 에 대한
= " exampleFormControlInput1 " > 아티클 URL </ label > < input id =
" posting-url " class = " form-control " 자리 표시 자 = " " > </ DIV
> < DIV 클래스 = " 폼 그룹 " > < 라벨 에 대한 = "
exampleFormControlTextarea1 " > 간단 코멘트 </ 라벨 > < 입력 ID = "
게시-코멘트 " 클래스 = " 폼 컨트롤 " 자리 = " " > </ div > < 버튼
onclick = " posting () " type = " button " class = " btn btn-
primary " > 기사 저장 </ button > </ form > </ div >
```

자바 스크립트 ▼

```
// 크롬 개발자 도구 콘솔 창에 쳐보기 // ID 값이 게시 박스 인 곳을 가리
키고, 숨기기 ()로 안보이게한다. (= CSS의 표시 값을 없음으로 바꾼다) $
( '# 게시 박스' ) . 숨기기 ( ) ; // show ()로 보이게한다. (= css의 표
시 값을 블록으로 바꾼다) $ ( '# posting-box' ) . 표시 ( ) ;
```

자바 스크립트 ▼

▼ 3. css의 값 가져와보기 (여기선 표시 속성 값을 가져와 볼게요!)

```
$ ( '# posting-box' ) . 숨기기 ( ) ; $ ( '# posting-box' ) . css (
'디스플레이' ) ; $ ( '# posting-box' ) . 표시 ( ) ; $ ( '# posting-
box' ) . css ( '디스플레이' ) ;
```

자바 스크립트 ▼

▼ 4. 태그 내 텍스트 입력하기

1) 입력 상자

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

자바 스크립트 ▼

2) 다른 것들-예) 버튼의 텍스트 메뉴

```
// 가리키고 싶은 버튼에 ID 값을 준 다음 < 버튼 ID = " BTN-게시 박스 "
유형 = " 버튼 " 클래스 = " BTN BTN-차 " > 포스팅 박스 열기 </ 버튼 >
```

자바 스크립트 ▼

```
$ ( '# btn-posting-box' ) . 텍스트 ( '포스팅 박스' ) ;
```

자바 스크립트 ▼

▼ 5. 태그 내 html 입력하기

- <div> ~ </div> 실험,
동적으로 html을 부르고 싶을 땐? (예를 들어, 포스팅하지만 → 카드 추가)

```
// 사이에 HTML을 넣고 싶은 태그에 ID 값을 준 다음 < DIV ID = " 카드
박스 " 클래스 = " 카드 열에 카드 포스트 " > < DIV 클래스 = " 카드 " >
< IMG 클래스 = " 카드 - img-top " src = " https://www.fodors.com/wp-
content/uploads/2018/10/4_UltimateRome_PiazzaNavona-975x650.jpg "
alt = " 카드 이미지 캡 " > <div class = " 카드 본체 " > < h5 class =
" 카드 제목 " > 예를 들어 카드가 나옵니다 !! </ h5 > < p class =
" card-text " > 여기에 기사 내용이 있겠어요 </ p > < p class = " 카드
텍스트 설명 " > 여기 엔가요가 들어갑니다 </ p > </ div > </ div > </
div >
```

자바 스크립트 ▼

1) 버튼을 누르면보기

```
let temp_html = '<button> 나는 추가 될 버튼이다! </ button>' ; $ (
'# cards-box' ) . 추가 ( temp_html ) ;
```

자바 스크립트 ▼

2) 버튼 말고

```
// 버튼과 같은 것인데, 좀 더 긴 html 임. 줄 바꿈 엔 \를 넣어 줘야 이어
질 수 있음 하자 temp_html = '<DIV 클래스 = "카드"> \ <img class =
```

```
"card-img-top"src = "https://www.fodors.com/wp-
content/uploads/2018/10/4_UltimateRome_PiazzaNavona-975x650.jpg"alt
= "카드 이미지 캡"> \ <div class = "card-body"> \ <h5 class = "card-
title"> 예를 들어 줘 카드가 나겠다 !! </ h5> \ <p class = "card-
text"> 여기에 기사 내용이 많 겠죠 </ p> \ <p class = "card-text
comment"> 여기 엔 가로가 들어갑니다 </ p> \ </ div> \ </ div> \ </
div> ' ; $ ( '# cards-box' ) . 추가 ( temp_html ) ;
```

자바 스크립트 ▾

▼ 5) 아약스 API 호출하기

👉 데이터를 담은 POST 요청

```
let url =
'https://www.nytimes.com/interactive/2016/08/25/universal/ko/36-hours-
seoul-korean.html' ; 하자 코멘트를 = '코멘트 테스트' ; 하자 저자 = '스파
르타 코딩 클럽' ; $ . ajax ( { type : "POST" , url : "/ post" , data : {
url_give : url , comment_give : comment , author_give : author } ,
success : function ( response ) { 콘솔 . 로그 ( 응답 ) } } )
```

자바 스크립트 ▾

👉 데이터를 담은 GET 요청

```
$ . ajax ( { 유형 : "GET" , url : "http://spartacodingclub.shop/post?
author_give= 스파르타 코딩 클럽" , 데이터 : { } , 성공 : 함수 ( 응답 ) {
콘솔 . 로그 ( 응답 ) } } )
```

자바 스크립트 ▾

▼ 6) 2 주차 완성 코드

```

<!doctype html> < html lang = " en " > < 헤드 > <!--웹 페이지 제목--> < 제목
타 태그--> < meta charset = " utf-8 " > < meta name = " viewport " content
= 1, shrink-to-fit = no " > <!--부트 스트랩 CSS--> < 링크를 확인해 = " 스타일
https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css " 무
058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW / dAiS6JXm " crossorigin = " 익명
https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js " > </ sc
https://cdnjs.cloudflare.com /ajax/libs/popper.js/1.12.9/umd/popper.min.js
Y1QKtv3Rn7W3mgPxhU9K / ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q " crossorigin =
" https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js " 무
JZR6Spejh4U02d8jOt6vLEHfe / JQGiRRSQQxSfFWpi1MquVdAyjUar5 + 76PVCmYl " crc
구글 폰트--> < 링크 HREF = " https://fonts.googleapis.com/css?family=Stylish
<script> function openclose() { if ( $('#posting-box').css('display') == '
$('#btn-posting-box').text('포스팅 박스 열기') } else { $('#posting-box').s
팅 박스 닫기') } } function posting() { // 읽기 let url = $('#posting-url')
comment').val(); let author = '스파르타코딩클럽'; // 우리가 크롬 콘솔창에서
POST 방식으로 요청하겠다. url: "http://spartacodingclub.shop/post", // /pos
url, comment_give: comment, author_give:author}, // 데이터를 주는 방법 succ
(response['result'] == 'success') { alert('포스팅 성공!'); window.location
} }) } $(document).ready(function(){ $('#cards-box').html(''); listing();
"GET", url: "http://spartacodingclub.shop/post?author_give=스파르타코딩클럽
{ if (response['result'] == 'success') { let articles = response['articles
i++) { make_card(articles[i]['comment'],articles[i]['desc'],articles[i]['i
['url']) } } else { alert('기사를 받아오지 못했습니다'); } } }) } function m
url) { let temp_html = '<div class="card">\ <img class="card-img-top" src=
class="card-body">\ <a href="'+url+'" class="card-title">'+title+'</a>\ <p
class="card-text comment">'+comment+'</p>\ </div>\ </div>'; $('#cards-box'
style --> <style type="text/css"> * { font-family: 'Stylish', sans-serif;
.comment { color: blue; font-weight: bold; } .form-post { max-width: 500px
border-color: #e9ecef; border-radius: 0.3rem; border: solid; display: bloc
</style> </head> <body> <div class="wrap"> <div class="jumbotron"> <h1 cla
<p class="lead">중요한 링크를 저장해두고, 나중에 볼 수 있는 공간입니다</p> <h
id="btn-posting-box" onclick="openclose()" class="btn btn-primary btn-lg"
</a> </p> </div> <div class="form-post" id="posting-box"> <div> <div class
for="exampleFormControlInput1">아티클 URL</label> <input id="posting-url"
</div> <div class="form-group"> <label for="exampleFormControlTextarea1">코
id="posting-comment" class="form-control" rows="2"></textarea> </div> <but
primary">기사저장</button> </div> </div> <div class="card-columns" id="card
class="card-img-top"
src="https://www.eurail.com/content/dam/images/eurail/Italy%200CP%20Promo%
alt="Card image cap"> <div class="card-body"> <a href="#" class="card-titl
class="card-text">기사의 요약 내용이 들어갑니다. 동해물과 백두산이 마르고 닳도
삼천리 화려강산...</p> <p class="card-text comment">여기에 코멘트가 들어갑니
 <div class="card-body"> <h5 class="card-title">여기

```

```

text">기사의 요약 내용이 들어갑니다. 동해물과 백두산이 마르고 닳도록 하느님이
산...</p> <p class="card-text comment">여기에 코멘트가 들어갑니다.</p> </div>
class="card-img-top"
src="https://www.eurail.com/content/dam/images/eurail/Italy%20CP%20Promo%
alt="Card image cap"> <div class="card-body"> <h5 class="card-title">여기
text">기사의 요약 내용이 들어갑니다. 동해물과 백두산이 마르고 닳도록 하느님이
산...</p> <p class="card-text comment">여기에 코멘트가 들어갑니다.</p> </div>
class="card-img-top" src = "
https://www.eurail.com/content/dam/images/eurail/Italy%20CP%20Promo%20Blc
= " 카드 이미지 캡 " > < div class = " card-body " > < h5 class = " 카드 제
p class = " card-text " > 기사 요약 내용이 들어갑니다. 동해 물과 백두산이
만세 무궁화 삼천리 화려 강산 ... </ p > < p class = " card-text comment " >
> < div class = " card " > < img class = " card-img-top " src = "
https://www.eurail.com/content/dam/images/eurail/Italy%20CP%20Promo%20Blc
= " 카드 이미지 캡 " > < div 클래스 = " card-body " > < h5 class = " card-t
< p class = " card-text " > 기사 요약 내용이 도착되었습니다. 동해 물과 백두
... </ p > <p 클래스 = " 카드 텍스트 설명 " > 여기에 나오는가 들어갑니다. </
</ body > </ html >

```

자바 스크립트 ▾

Python, MongoDB, 크롤링

▼ 1) 문법 기초 문법

▼ 변수 및 기본 연산

```
a = 3 # 3을 a에 넣는다 b = a # a를 b에 넣는다 a = a + 1 # a+1을 다시
a에 넣는다 num1 = a*b # a*b의 값을 num1이라는 변수에 넣는다 num2 = 99
# 99의 값을 num2이라는 변수에 넣는다 # 변수의 이름은 마음대로 지을 수
있음! # 진짜 "마음대로" 짓는 게 좋을까? var1, var2 이렇게?
```

Python ▼

▼ 자료형

• 숫자, 문자형

```
name = 'bob' # 변수에는 문자열이 들어갈 수도 있고, num = 12 # 숫
자가 들어갈 수도 있고, is_number = True # True 또는 False ->
"Boolean"형이 들어갈 수도 있습니다. ##### # 그리고 List,
Dictionary 도 들어갈 수도 있죠. 그게 뭔지는 아래에서!
```

Python ▼

• 리스트 형 (Javascript의 배열형과 동일)

```
a_list = [] a_list.append(1) # 리스트에 값을 넣는다
a_list.append([2,3]) # 리스트에 [2,3]이라는 리스트를 다시 넣는다
# a_list의 값은? [1,[2,3]] # a_list[0]의 값은? 1 # a_list[1]의
값은? [2,3] # a_list[1][0]의 값은? 2
```

Python ▼

• Dictionary 형 (Javascript의 dictionary형과 동일)

```
a_dict = {} a_dict = {'name':'bob','age':21} a_dict['height'] =
178 # a_dict의 값은? {'name':'bob','age':21, 'height':178} #
a_dict['name']의 값은? 'bob' # a_dict['age']의 값은? 21 #
a_dict['height']의 값은? 178
```

Python ▼

• Dictionary 형과 List형의 조합

```
people = [{'name':'bob','age':20},{'name':'carry','age':38}] #
people[0]['name']의 값은? 'bob' # people[1]['name']의 값은?
'carry' person = {'name':'john','age':7} people.append(person)
# people의 값은? [{'name':'bob','age':20},
{'name':'carry','age':38},{'name':'john','age':7}] # people[2]
['name']의 값은? 'john'
```

Python ▼

▼ 함수

- 함수의 정의 - 이름은 마음대로 정할 수 있음!

```
def f(x): return 2*x+3
```

Python ▾

- 함수의 응용

```
def sum_all(a,b,c): return a+b+c def mul(a,b): return a*b
result = sum_all(1,2,3) + mul(10,10) # result라는 변수의 값은?
106 def minus(a,b): return a-b result2 =
minus(mul(10,10),sum_all(1,2,3)) # result2라는 변수의 값은? 94
```

Python ▾

▼ 조건문

```
def oddeven(num): # oddeven이라는 이름의 함수를 정의한다. num을 변수
로 받는다. if num % 2 == 0: # num을 2로 나눈 나머지가 0이면 return
True # True (참)을 반환한다. else: # 아니면, return False # False
(거짓)을 반환한다. def checkbob(name): if name == 'bob': # name이
'bob'이면 True를, 아니면 False를 반환해라 return True else: return
False
```

Python ▾

▼ 반복문

👉 파이썬에서의 반복문은, 리스트의 요소들을 하나씩 꺼내쓰는 형태입니다.

```
people = [{'name':'bob','age':20},{'name':'carry','age':38},
{'name':'john','age':7}] # 모든 사람의 이름과 나이를 출력해봅시다. for p
in people: print (person['name'] + ' / ' + person['age']) # 이번엔, 반
과 조건문을 응용한 함수를 만들어봅시다. # 이름을 받으면, age를 리턴해주는
def get_age(myname): for person in people: if person['name'] == myname
return person['age'] return '해당하는 이름이 없습니다' print
(get_age('bob')) print (get_age('kay'))
```

Python ▾

▼ 2) 웹 스크래핑 기본 성능

패키지 설치하기

beautifulsoup4, requests - beautifulsoup4 -> HTML 파싱을 편리하게. -
requests -> 브라우저에서 엔터를 치는 효과. URL에 요청하기

Python ▾

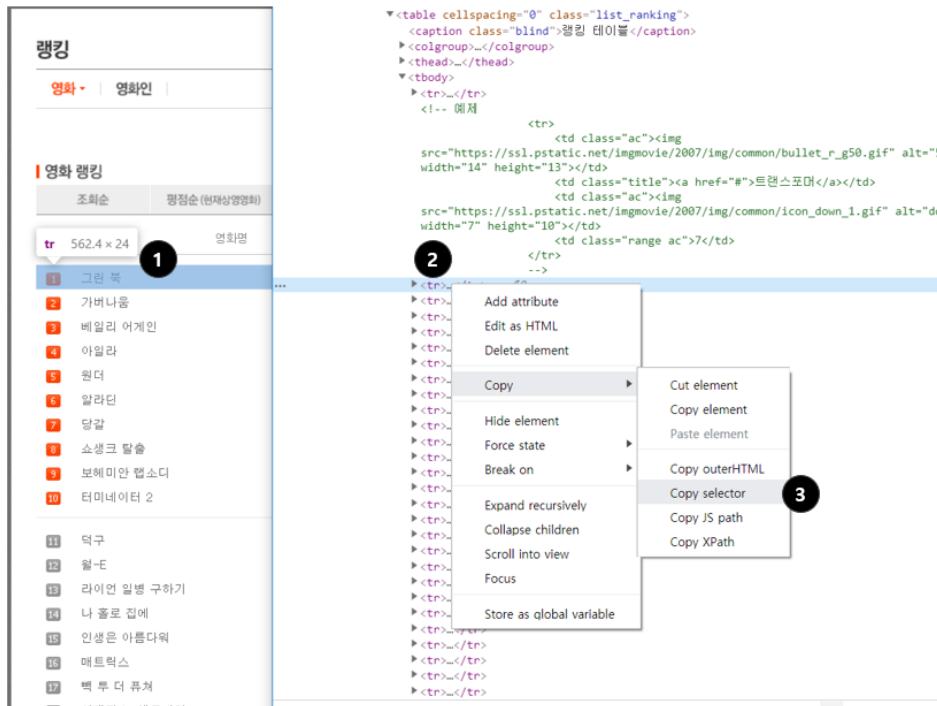
스크래핑 코드 기본 세팅

```
import requests from bs4 import BeautifulSoup headers = {'User-Agent' :
'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/73.0.3683.86 Safari/537.36'} data =
requests.get('https://movie.naver.com/movie/sdb/rank/rmovie.nhn?
sel=pnt&date=20190909', headers=headers) soup = BeautifulSoup(data.text,
'html.parser') ##### # (입맛에 맞게 코딩)
#####
```

Python ▾

▼ 3) beautifulsoup 라이브러리 내 선택 / 선택

크롬 개발자도구의 Copy selector를 활용



select는 여러개가 리스트에 담겨서, select_one은 한 개만 출력

```
# 여러개를 가져오고 싶은 경우 soup.select('태그명') soup.select('.클래스명') soup.select('#아이디명') soup.select('상위태그명 > 하위태그명 > 하위태그명') soup.select('상위태그명.클래스명 > 하위태그명.클래스명')
soup.select('태그명[속성="값"]') # 한 개만 가져오고 싶은 경우
soup.select_one('위와 동일')
```

Python ▼

- 항상 정확하지는 않으나, 크롬 개발자도구를 참고할 수도 있습니다.

1. 원하는 부분에서 마우스 오른쪽 클릭 → 검사
2. 원하는 태그에서 마우스 오른쪽 클릭
3. Copy → Copy selector로 선택자를 복사할 수 있음

▼ 4) 네이버 영화 쓰기 래핑 코드

```
import requests from bs4 import BeautifulSoup # URL을 읽어서 HTML을 받아  
오고, headers = {'User-Agent' : 'Mozilla/5.0 (Windows NT 10.0; Win64;  
x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.86  
Safari/537.36'} data =  
requests.get('https://movie.naver.com/movie/sdb/rank/rmovie.nhn?  
sel=pnt&date=20190909', headers=headers) # HTML을 BeautifulSoup이라는 라  
이브러리를 활용해 검색하기 용이한 상태로 만들 soup =  
BeautifulSoup(data.text, 'html.parser') # select를 이용해서, tr들을 불러  
오기 movies = soup.select('#old_content > table > tbody > tr') # movies  
(tr들) 의 반복문을 돌리기 for movie in movies: # movie 안에 a 가 있으면,  
a_tag = movie.select_one('td.title > div > a') if not a_tag == None: #  
a의 text를 찍어본다. print (a_tag.text)
```

Python ▼

▼ 5) mongoDB 기능하기

패키지 설치하기

```
pymongo
```

Python ▾

DB연결하기

```
from pymongo import MongoClient # pymongo를 임포트 하기(패키지 인스톨 먼저 해야겠죠?)
client = MongoClient('localhost', 27017) # mongoDB는 27017 포트로 돌아갑니다.
db = client.dbsparta # 'dbsparta'라는 이름의 db를 만듭니다.
```

Python ▾

Insert 하기

```
# MongoDB에 insert 하기 doc = { 'name':'bobby', 'age':21 } # 'users'라는 collection에 {'name':'bobby','age':21}를 넣습니다.
db.users.insert_one(doc)
```

Python ▾

Update 하기

```
# 생김새 db.people.update_many(찾을조건,{ '$set': 어떻게바꿀지 })
db.users.update_one({'name':'bobby'},{'$set':{'age':19}})
```

Python ▾

Find 하기

```
# 여러개 뽑아보기 # 리스트형태. _id는 출력 안함 same_ages =
list(db.users.find({'age':21},{'_id':0}))
```

Python ▾

```
# 한 개 뽑아보기 # 딕셔너리형태. _id는 출력 안함 user =
db.users.find_one({'name':'bobby'},{'_id':0})
```

Python ▾

▼ 6) 네이버 영화 스크래핑 → mongoDB 저장 코드

```
import requests from bs4 import BeautifulSoup from pymongo import
MongoClient # pymongo를 임포트 하기(패키지 인스톨 먼저 해야겠죠?) client
= MongoClient('localhost', 27017) # mongoDB는 27017 포트로 돌아갑니다.
db = client.dbsparta # 'dbsparta'라는 이름의 db를 만듭니다. # URL을 읽어
서 HTML를 받아오고, headers = {'User-Agent' : 'Mozilla/5.0 (Windows NT
10.0; Win64; x64)AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/73.0.3683.86 Safari/537.36'} data =
requests.get('https://movie.naver.com/movie/sdb/rank/rmovie.nhn?
sel=pnt&date=20190909',headers=headers) # HTML을 BeautifulSoup이라는 라
이브러리를 활용해 검색하기 용이한 상태로 만듦 soup =
BeautifulSoup(data.text, 'html.parser') # select를 이용해서, tr들을 불러
오기 movies = soup.select('#old_content > table > tbody > tr') # movies
(tr들) 의 반복문을 돌리기 rank = 1 for movie in movies: # movie 안에 a 가
있으면, a_tag = movie.select_one('td.title > div > a') if not a_tag ==
None: title = a_tag.text star = movie.select_one('td.point').text doc =
{ 'rank' : rank, 'title' : title, 'star' : star }
db.movies.insert_one(doc) rank += 1
```

Python ▼

▼ 7) 메타 태그

🔗 meta태그 중 property가 og:image인 태그의 content 값을 가져오기

```
import requests from bs4 import BeautifulSoup url =
'https://platum.kr/archives/120958' headers = {'User-Agent' :
'Mozilla/5.0 (Windows NT 10.0; Win64; x64)AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/73.0.3683.86 Safari/537.36'} data =
requests.get(url,headers=headers) soup = BeautifulSoup(data.text,
'html.parser') og_image = soup.select_one('meta[property="og:image"]')
url_image = og_image['content'] print(url_image, url_title,
url_description)
```

Python ▼

API 만들기

▼ 1) 플라스크 : 기본 구조 + URL 스위치

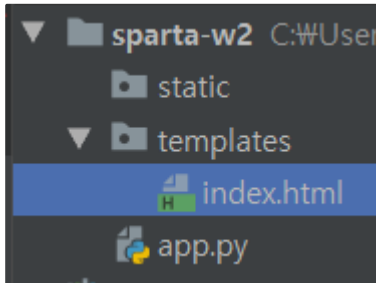
```
from flask import Flask app = Flask(__name__) @app.route('/') def
home(): return 'This is Home!' @app.route('/mypage') def mypage():
return 'This is My Page!' if __name__ == '__main__':
app.run('0.0.0.0',port=5000,debug=True)
```

Python ▼

▼ 2) 플라스크 : html 파일 건네주기

- 기본 폴더구조

👉 flask 프레임워크를 쓸 때의 기본 폴더구조입니다. 오타에 유의하세요!
반드시 templates 폴더내에 html 파일을 뒤야 불러올 수 있습니다.



- html 파일 불러오기

👉 render_template를 import 에 추가하고, home()내에 써보았습니다.

```
from flask import Flask, render_template app = Flask(__name__) ##
URL 별로 함수명이 같거나, ## route('/') 등의 주소가 같으면 안됩니다.
@app.route('/') def home(): return render_template('index.html') if
__name__ == '__main__': app.run('0.0.0.0',port=5000,debug=True)
```

Python ▼

▼ 3) CSS, 이미지가 깨질 때

👉 flask가 실행되는 위치를 기준으로 가져와야해서 그렇습니다.
아래와 같이 바꿔주면, 깨지지 않습니다.

단, static 폴더 내 mystyle.css 파일을 위치시켜야 한다는 점!

```
<link rel="stylesheet" href="{ url_for('static',
filename='mystyle.css') }}">
```

HTML ▾

▼ 4) 플라스크 : GET, POST API 만들기

👉 title_give라는 키 값으로 들어온 데이터를 받는 방법입니다.

```
from flask import Flask, render_template, jsonify, request app =
Flask(__name__) ## HTML을 주는 부분 @app.route('/') def home(): return
render_template('index.html') ## API 역할을 하는 부분
@app.route('/test', methods=['POST']) def test_post(): title_receive =
request.form['title_give'] print(title_receive) return
jsonify({'result':'success', 'msg': '이 요청은 POST!'})
@app.route('/test', methods=['GET']) def test_get(): title_receive =
request.args.get('title_give') print(title_receive) return
jsonify({'result':'success', 'msg': '이 요청은 GET!'}) if __name__ ==
'__main__': app.run('0.0.0.0',port=5000,debug=True)
```

Python ▾

▼ 5) 플라스크 : 네이버 영화 쓰기 래핑

▼ pymongo 기본 세팅

```
from flask import Flask, render_template, jsonify, request app =
Flask(__name__) from pymongo import MongoClient # pymongo를 임포트
하기(패키지 인스톨 먼저 해야겠죠?) client = MongoClient('localhost',
27017) # mongoDB는 27017 포트로 돌아갑니다. db = client.dbsparta #
'dbsparta'라는 이름의 db를 만듭니다. ## HTML을 주는 부분
@app.route('/') def home(): return render_template('index.html')
(중략..)
```

Python ▾

▼ GET연습: 받은 rank와 일치하는 영화 정보 가져오기

• API

👉 클라이언트가 준 값은, 모두 문자입니다. (즉, 숫자 2로 줘도 "2"로 인식)
반면, DB에 저장된 rank의 값들은 숫자죠! robo3T에서 확인해볼까요?

그래서 비교하려면, 받은 rank_give 값을 숫자로 꼭 바꿔줘야합니다!

```
@app.route('/test', methods=['GET']) def test_get(): #
rank_give로 클라이언트가 준 rank를 가져오기 rank_receive =
request.args.get('rank_give') # rank_receive를 숫자로 만들어주기
(db엔 숫자로 저장되어있으니까!) rank_receive = int(rank_receive)
# rank의 값이 받은 rank와 일치하는 document 찾기 & _id 값은 출력
에서 제외하기 movie_info =
db.movies.find_one({'rank':rank_receive},{ '_id':0}) # info라는
키 값으로 영화정보 내려주기 return jsonify({'result':'success',
'info':movie_info})
```

Python ▾

• ajax로 잘 작동하는 지 확인

```
$.ajax({ type: "GET", url: "/test?rank_give=2", data: {},
success: function(response){ console.log(response) } })
```

JavaScript ▾

▼ POST연습: 받은 rank의 평점을 받은 star로 업데이트하기

•

```

## API 역할을 하는 부분 @app.route('/test', methods=['POST']) def
test_post(): # rank_give로 클라이언트가 준 rank를 가져오기 & 숫자변환
rank_receive = request.form['rank_give'] rank_receive = int(rank_r
# star_give로 클라이언트가 준 star를 가져오기 & 숫자변환 star_receiv
request.form['star_give'] star_receive = int(star_receive) # 해당
영화를 받은 score로 업데이트 해주기
db.movies.update_one({'rank':rank_receive},{'$set':{'star':star_re
# 다했으면 성공여부만 보냄 return jsonify({'result':'success'})

```

- ajax로 잘 작동하는 지 확인

```

$.ajax({ type: "POST", url: "/test", data:
{rank_give:2,star_give:5}, success: function(response){
console.log(response) } })

```

JavaScript ▾

▼ POST연습: rank, title, star를 받아서 새로운 영화를 DB에 넣어주기

- API

```

@app.route('/new', methods=['POST']) def new_post():
rank_receive = int(request.form['rank_give']) star_receive =
request.form['star_give'] title_receive =
request.form['title_give'] db.movies.insert_one({'rank':
rank_receive, 'star': star_receive, 'title':title_receive})
return jsonify({'result': 'success'})

```

Python ▾

- ajax로 잘 작동하는 지 확인

```

$.ajax({ type: "POST", url: "/new", data: { rank_give:'99',
star_give:'0', title_give:'청년경찰' }, success:
function(response){ console.log(response) } })

```

JavaScript ▾

▼ 6) 플라스크 : 나홀로 메모장 전체 코드

```

from flask import Flask, render_template, jsonify, request
app = Flask(__name__)
import requests
from bs4 import BeautifulSoup
from pymongo import MongoClient # pymongo를 임포트 하기(패키지 인스톨 먼저 해야겠죠?)
client = MongoClient('localhost', 27017) # mongoDB는 27017 포트로 돌아갑니다.
db = client.dbsparta # 'dbsparta'라는 이름의 db를 만듭니다.
## HTML을 주는 부분
@app.route('/')
def home():
    return render_template('index.html')
@app.route('/post', methods=['GET'])
def listing():
    # author_give 클라이언트가 준 author를 가져오기
    author_receive = request.args.get('author_give') # author의 값이 받은 author와 일치하는 document 찾기 & _id 값은 출력에서 제외하기
    result = list(db.articles.find({'author':author_receive},{'_id':0})) # articles라는 키 값으로 영화정보 내려주기
    return jsonify({'result':'success', 'articles':result})
## API 역할을 하는 부분
@app.route('/post', methods=['POST'])
def saving():
    # 클라이언트로부터 데이터를 받는 부분
    url_receive = request.form['url_give']
    comment_receive = request.form['comment_give']
    author_receive = request.form['author_give']
    # meta tag를 스크래핑 하는 부분
    headers = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/73.0.3683.86 Safari/537.36'}
    data = requests.get(url_receive, headers=headers)
    soup = BeautifulSoup(data.text, 'html.parser')
    og_image = soup.select_one('meta[property="og:image"]')
    og_title = soup.select_one('meta[property="og:title"]')
    og_description = soup.select_one('meta[property="og:description"]')
    url_image = og_image['content']
    url_title = og_title['content']
    url_description = og_description['content']
    # mongoDB에 넣는 부분
    article = {'author':author_receive, 'url': url_receive, 'comment': comment_receive, 'image': url_image, 'title': url_title, 'desc': url_description}
    db.articles.insert_one(article)
    return jsonify({'result': 'success'})
if __name__ == '__main__':
    app.run('0.0.0.0', port=5000, debug=True)

```

Python ▼

