

# 면접의 정석

The Bible of Computer Science

[CS지식의 정석]

version.23.05.20

해당 교안을 무단으로 복제, 업로드, 배포, 도용할 경우  
저작권법 제 97조의 저작권침해에 해당하며  
5년이하의 징역 또는 5천만원 이하의 벌금에 처해질 수 있습니다.

강의에 있는 설명이 교안에 전부 들어있지 않고 요약본으로 제공되는 경우는 출판권상 책과 중복되는 부분이어서 그렇게 제공되는 점 참고부탁드립니다.  
또한 강의에는 책과 중복되는 내용이라도 모두 다루고 있습니다. :)

## 목차

면접의 기본.....	3
검손.....	3
워딩.....	3
복장.....	4
체크리스트를 만들어 준비하자.....	4
꼬리에 꼬리를 무는 질문을 준비하는 방법.....	5
예시 답변.....	5
1. MongoDB 아키텍처.....	5
2. MongoDB Data Model.....	6
3. MongoDB Storage Engines.....	6
인정하는 자세.....	7
Q. 1분 자기소개 부탁드립니다.....	7
모범답안.....	8
Q. 좋은 개발자란 무엇인가요?.....	8
리스트업.....	8
자세함.....	9
Q. 이 라이브러리가 있는데 이 라이브러리는 왜 썼나요?.....	9
특징.....	10
데이터바인딩.....	11
자유도, 타 플러그인 의존도, DOC, 러닝커브 비교.....	11
성능.....	12
커뮤니티.....	12
모바일로의 확장성.....	13
팀원들의 역량.....	13
Q. 워라벨에 대해 어떻게 생각하나요?.....	13
모범답안.....	14

Q. 의견 충돌이 났을 때 어떻게 해야하나요?.....	14
Q. 마지막으로 하고 싶은 질문이 있나요?.....	14

## 면접의 기본

흔히들 개발자들이 착각할 수 있는게 “실력”만 좋으면 된다는 마음가짐입니다. 저또한 그랬었고 자만했으며 겸손하지 못했습니다. 그래서 면접에서 떨어지기도 했습니다.

그런 사실을 알아야 합니다.

본인의 실력과 똑같은 사람이 2명이 있다면 기본을 갖춘 사람을 뽑겠다라는 사실이요.

여기서 기본이라는 것은 무엇일까요?

필자는 3가지라고 생각합니다. 겸손, 워딩, 복장입니다.

### 겸손

자신의 실력은 뽐내되 겸손해야 합니다. 어떠한 기술에 대해 자신감있게 자신이 말하고자 하는 것은 말하되 자신이 모르는 것은 “~정도까지 압니다.” 라고 자신이 모르는 것에 대해서는 명확하게 말해주어야 합니다. 그리고 “면접관님이 더 잘 아시겠지만..” 하면서 어느정도 면접관의 실력에 대해 존중하는 말도 필요합니다.

### 워딩

자신의 기술, 인성 등을 단 1시간 또는 그보다 더 짧은 시간에 보여주어야 하는 것이 면접입니다. 자신이 가지고 있는 기술에 대해 요약적으로 그리고 잘 설명해야 합니다. 그걸 위해서 자신이 가지고 있는 덕션이나 말의 빠르기를 연습할 필요가 있습니다.

ex) 윤동주의 서시 + 동영상으로 한번 찍어보기

죽는 날까지 하늘을 우러러

한 점 부끄럼이 없기를,

앞새에 이는 바람에도

나는 괴로워했다.

별을 노래하는 마음으로

모든 죽어 가는 것을 사랑해야지

그리고 나한테 주어진 길을

걸어가야겠다.

오늘 밤에도 별이 바람에 스치운다.

## 복장

아무리 가볍게 입으라도 티셔츠 한장 덜렁 입고 면접을 보서는 안됩니다. 어떻게 보이는가는 중요합니다. 스트라이프 와이셔츠를 추천합니다.

<https://www.musinsa.com/app/goods/1550594>

## 체크리스트를 만들어 준비하자.

면접전에 어떤 것을 준비해야할지는 정해져있습니다.

먼저 백엔드를 가고 싶다면 백엔드 관련 면접후기 또는 질문을 중심으로 구글링해서 준비하면 되고 여기에다가 가고싶은 회사의 채용공고를 기반으로 공부할것을 체크리스트로 만들어 준비해봅시다.

이 때 신입 같은 경우는 공고자체가 자세하지 않으니 신입의 경우 3년이하의 경력직 채용공고를 기반으로 잡는게 좋습니다.

<https://recruit.navercorp.com/micro/techopen/2022>

Back-end	서비스가 원활하게 작동할 수 있도록 뒷받침합니다. 클라이언트의 요청에 따라 저장된 데이터를 활용하거나 비즈니스 로직을 처리하여 적절한 응답을 전달해 주는 서버를 개발합니다.
AI/ML	의도를 파악하는 기술로 편리한 일상을 만듭니다. 글로벌 수준의 인공지능 모델을 연구 개발하고 서비스 및 로봇에 적용/운영합니다.

앞의 그림처럼 신입공고는 자세하지 않습니다.

### ◆ 세부업무

#### (1) 대용량 데이터베이스 플랫폼 개발

- Database as a Service(DBaaS)를 개발하고 운영합니다.
- 서비스 장애시간을 최소화 시키기 위해 자동화된 고가용성(HA) 방안을 연구하고 이를 위한 플랫폼을 구축합니다.
- DBMS가 성능 튜닝을 수행하고 DBMS의 New Feature들을 검토하여 서비스에 적용합니다.
- DBMS 성능 향상과 안정적 운영을 위한 DBMS Internal 기술을 연구합니다.

앞의 채용공고를 보고 다음과 같이 체크리스트를 만들수 있겠죠?

- DB 튜닝
- HA

## 꼬리에 꼬리를 무는 질문을 준비하는 방법

면접에서 나오는 단골질문 중 하나는 꼬리에 꼬리를 무는 질문입니다. 예를 들어 MongoDB를 안다라고 포트폴리오에 써놓았으면 MongoDB의 인덱싱을 질문하고 인덱싱에 대해 답변을 하면 그러면 인덱싱을 위한 MongoDB의 구조는 무엇이나 어떤 아키텍처를 가지냐라는 식으로 꼬리에 꼬리를 무는 질문이 나오게 됩니다.

어떠한 기술에 대해 이런 꼬리에 꼬리를 무는 질문을 준비하는 방법은 단순합니다.

1. 기술 + pdf를 기반으로 구글링을 해서 얻은 pdf를 중심으로 공부합니다.
2. 관련된 기술에 대한 책 또는 강의로 공부합니다.
3. 해당 기술 공식사이트를 훑어봅니다.

앞의 과정은 시간이 많이 걸리는 단점이 있습니다. 자신이 아는 모든 기술을 이 과정을 통해 공부할 필요는 없습니다. 본인이 잘하고 싶은, 또는 깊이 알고 싶은 기술 하나만을 정해서 공부를 해놓고 면접관이 이 기술 말고 다른 기술에 대해 물어보고 해당 기술에 깊게 답변하지 못했을 때 혹시 ~~한 기술은 정말 자세히 아는데 이 기술에 대해 말씀드려도 될까요? 라는 식으로 자신이 잘하는 답변으로 유도하는게 좋습니다.

ex)

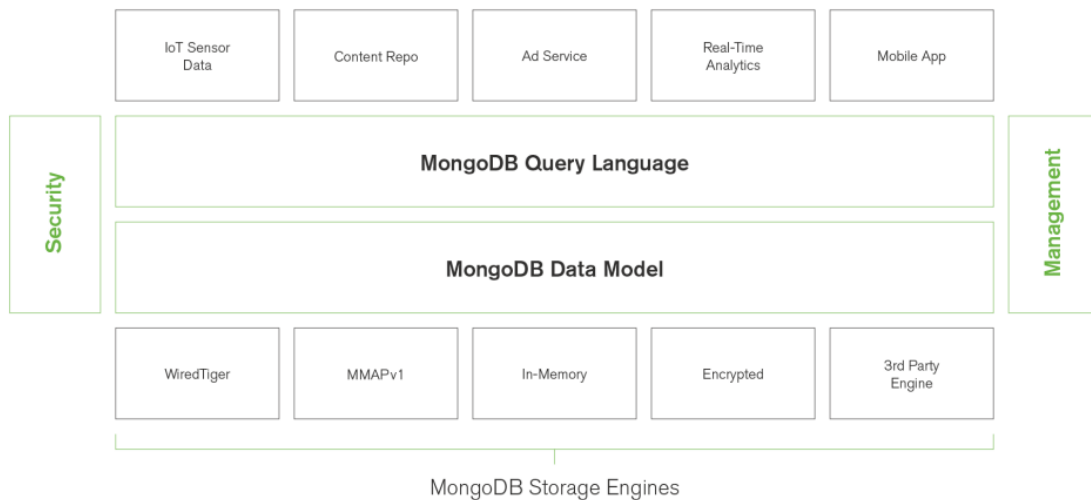
[https://jira.mongodb.org/secure/attachment/112939/MongoDB\\_Architecture\\_Guide.pdf](https://jira.mongodb.org/secure/attachment/112939/MongoDB_Architecture_Guide.pdf)

제가 MongoDB에 대해 공부하며 준비했던 답변은 다음과 같습니다.

### 예시 답변

#### 1. MongoDB 아키텍처

MongoDB는 C, C++, Go, Java, JavaScript 등 수많은 언어로 쿼리를 보낼 수 있습니다. 그중 Node.js에서 MongoDB에 접근하는 데 사용하는 모듈에는 Mongoose와 MongoDB 드라이버 2개가 있으며 MongoDB에 쿼리라는 요청을 통해 원하는 데이터의 수정, 조회작업 등을 할 수 있습니다.



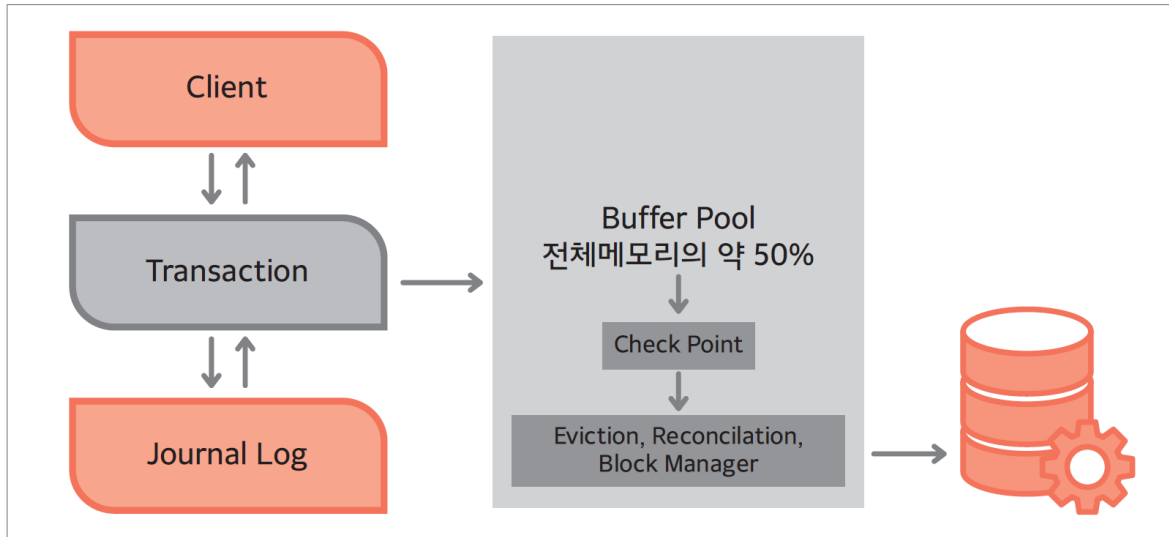
## 2. MongoDB Data Model

RDBMS에서는 데이터베이스 - 테이블 - 데이터로 계층화되어 있습니다. 반면 MongoDB에서는 데이터베이스 - 컬렉션 - 문서로 계층화되어 있습니다. 흔히 데이터(data)라고 하는 것을 MongoDB에서는 문서(document)라고 합니다. RDBMS의 row와 같은 개념입니다.

그리고 BSON(Binary JSON)으로 저장되며 유연하게 저장할 수 있습니다. 데이터 스키마는 그래프, key-value, 테이블 형태든 뭐든 쉽게 저장하고 데이터를 결합해서 쓸 수 있습니다.

## 3. MongoDB Storage Engines

스토리지 엔진은 데이터 서버의 디스크에서 사용자가 요청한 데이터를 어떻게 가져오고 어떻게 저장할지 결정합니다. 3.2 이상부터 기본(default)값으로 설정되는 wiredTiger엔진, In-Memory 등이 있는데, 이를 선택해 쓸 수 있습니다. wiredTiger 엔진은 LSMTree(로그 기반 병합트리)를 이용하여 읽기 성능을 포기하고 그만큼 저장 성능을 향상시키고 느린 읽기 성능을 보완하기 위해 블룸 필터를 사용한 엔진입니다. 그렇다면 MongoDB에서 트랜잭션이 일어나면 어떤 일이 발생할까요? 즉, insert나 find 등 쿼리를 보내면 MongoDB는 어떻게 그 쿼리들을 처리하게 될까요?



앞의 그림처럼 Journal Log에 저장됨과 동시에 Buffer Pool로 들어가게 됩니다. Journal Log에 저장되는 것을 저널링이라고 하는데, 쿼리 수행 실패 시 롤백을 하기 위해 먼저 저널 파일에 로그를 남기는 것을 뜻합니다. 그리고 버퍼풀(Buffer Pool)에 들어가 체크포인트(Check Point), 즉 메모리 버퍼와 디스크 간의 데이터 불일치를 해소하기 위해 메모리에서 디스크로 데이터를 동기화하는 작업이 일어나고 이빅션(eviction), 즉 메모리 버퍼에서 필요 없는 데이터를 삭제하는 작업이 일어나게 됩니다.

## 인정하는 자세

이렇게 준비했음에도 불구하고 모르는 것은 분명히 나올겁니다. 이 때 ~~정도까지 압니다. ~~는 모르는 것 같습니다. 혹시 ~~ 이부분은 자세히 아는데 설명해도 될까요?라고 하는 자세와 또 내가 말한 부분이 틀릴 수도 있습니다. 만약 그 틀림이 확실하다면 “아 제가 잘못알고 있는 거 같습니다. 해당부분은 다시 한번 확인해보도록 하겠습니다.”라고 하는 자세가 필요합니다.

## Q. 1분 자기소개 부탁드립니다.

면접에서 가장 중요한 부분입니다.

그냥 무엇을 잘한다 보다는 무엇을 몇%로 잘한다하면서 조금은 수학적으로 알려주는게 좋습니다.

예시는 다음과 같습니다.

## 모범답안

안녕하세요. 끊임없이 도전하는 개발자, 주홍철입니다.

대표적으로 현 대한민국 공항 및 공군에서 쓰는 실시간 기상상황 모니터링시스템을 개발했으며 오픈소스 mocha.js 의 contributor이기도 합니다.

1700이상의 알고리즘 문제를 푼 경험을 바탕으로 프론트에서 세그먼트를 이용해 서버요청을 줄여 67% 이상 성능을 향상시켰으며 15 배의 자료증가가 되었지만 불필요한 로직, 요청을 제거해 30% 의 성능을 향상시킨 경험이 있습니다.

이외에도 AI, IOT 등 다양한 프로젝트를 진행했으며 카카오 프런티엔드 챌린지 6위, 스타트업 해커톤 1위와 대한민국 국방부 등으로부터 개발 관련 공로를 인정받아 참모총장상, 사령관상, 단장상 등을 수상한 경험이 있습니다.

## Q. 좋은 개발자란 무엇인가요?

저는 좋은 개발자란 자세하고 좋은 질문을 하는 사람이라고 생각합니다.

## 리스트업

질문은 요약하고 정리해서 해야 하고 리스트업하는게 좋습니다.

ex)

안녕하세요 큰돌님. MongoDB서버하나 만들려고 하는데 질문하나 드려도 될까요? 남는 유후 서버 하나를 DB전용 서버로 만들건데 관리자만 아는 비밀키(해시키)를 만들어서 HTTP요청으로 header - authorization 키 확인해서 해당 해시키가 아니면 요청거부해서 접근제한두게 할 계획인데요. 혹시 추가적으로 해야될 보안조치가 있을까요?정리하자면 이렇게 할 계획이에요.

- 작업 : insert, select 만 가능.



- 사용자가 접근할 인터페이스 X, 관리자만 가능.
- IP 누출됨을 방지(해시키 기반 private API) : 해시는 SHA256 또는 bcrypt로 생성
- HTTPS 예정(추가적으로 가능하다면)
- 계정권한제한(서버 내부에서 API요청받았을 때 처리하는 계정은 관리자권한 X)
- 민감한 collection에 접근 불가(API를 기반으로 통신 및 해당 collection에서만 작업수행)
- 접근 IP에 대한 로깅 수행(간단한 FILE 기반)

## 자세함



0



! 질문이 모두 해결되었다면, 왼쪽 체크 표시를 눌러 질문의 상태를 바꿔보세요!

<< 코드는 하단에 링크로 첨부하였습니다. >>

안녕하십니까 큰돌 선생님

백준 3273 두 수의 합 문제를 30분 여유 시간 안으로 생각하고 먼저 풀어보려고 했는데 틀렸다고 나와서 질문 드리려 합니다.

문제에서 보여준 예제 경우는 정답이 맞게 나왔지만 3% 째에서 틀렸다고 나옵니다. 그런데 왜 틀렸는지 모르겠습니다.

수열을 정렬하고, 오른쪽 j인덱스를 줄여나가면서 왼쪽 i의 0번째 부터 탐색하면서 x와 값이 같으면 cnt++ 하는 로직을 떠올리고 구현했습니다.

선생님이 작성하신 코드의 로직과 어떤 차이가 있는 것인지 파악을 못해서 틀린 것 같습니다. 제 코드를 봐주시고 피드백 주시면 감사하겠습니다.

Q1. 위에서 설명한 저의 로직이 어떤 부분에서 부족한가요?

Q2. 선생님의 코드와 비교해서 분석한다면 어떤 부분이 다른건가요?

읽어주셔서 감사합니다..

<http://boj.kr/ee88cf1f8b544d1cb9dfb0617b018d9a>

## Q. 이 라이브러리가 있는데 이 라이브러리는 왜 썼나요?

면접질문 중 단골질문입니다. 프로젝트내에 어떤 기술이 있고 어떤 기능에 대한 여러 기술이 있다면 그 기술을 쓴 이유가 반드시 있어야 합니다.

그저 팀장이 시켜서가 아닌, 팀원이 하자고 해서가 아닌 왜 그 기술을 사용하는가에 대한 이유가 반드시 있어야 합니다.

기술을 도입하는 것은 개발의 시작이며 신발끈을 묶는 행위입니다. 신발끈을 제대로 묶지 않으면 마라톤이라는 개발을 제대로 수행할 수 없습니다.

화면을 표출하는 “뷰”라는 기능에 대해 어떠한 기술을 써야 할까요? 뷰에 관해 유명한 라이브러리이자 프레임워크로 비교가 되는 것은 바로 Vue.js와 React.js입니다. 어떤 것을 써야 할까요? 해당 부분에 대해 제가 고민했던 사례를 공유합니다.



제가 고민했던 부분들은 다음과 같습니다.

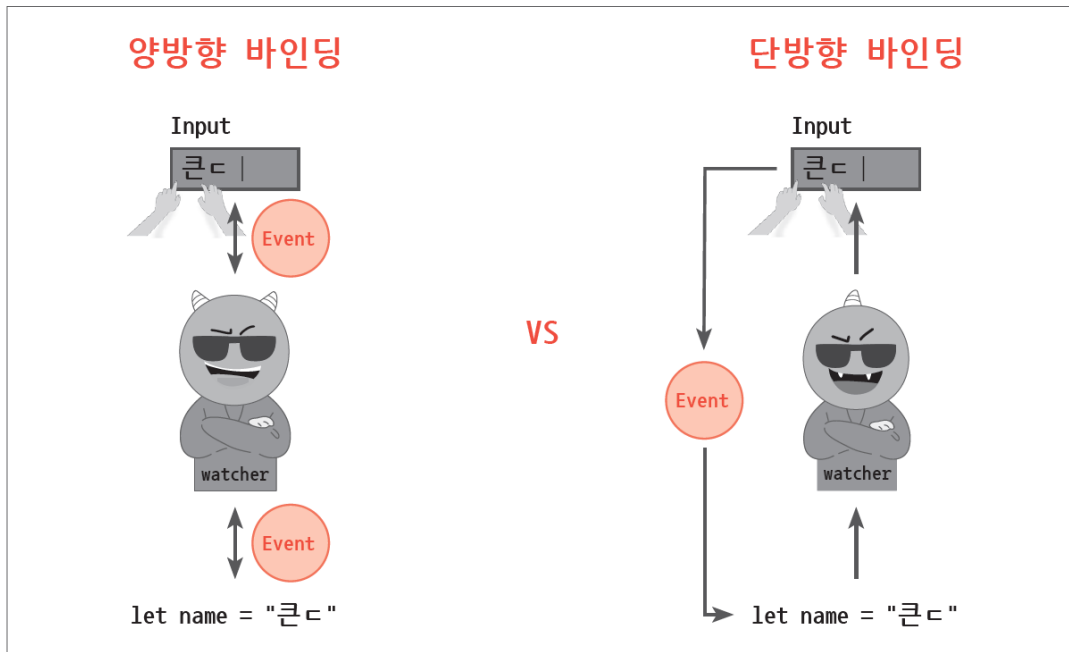
특징, 자유도, 타플러그인 의존도, 러닝커브, 성능, DOC, 커뮤니티, 모바일로의 확장성, 팀원들의 역량이었습니다.

## 특징

Vue.js	React.js
반응형 <ul style="list-style-type: none"><li>- watch, computed가 있으며 값 대입만으로 값을 수정가능.</li><li>- <a href="#">Try it in the PlaygrVue SFC Playground</a></li></ul>	함수형 <ul style="list-style-type: none"><li>- immutable, setState(함수를 통해서 값을 바꿀 수 있음.) mapStatetoProps, pureComponent</li><li>- <a href="https://codepen.io/e_me/pen/NdPa_bd">https://codepen.io/e_me/pen/NdPa_bd</a></li></ul>
양방향 바인딩	단방향 바인딩
html	JSX

## 데이터바인딩

먼저 데이터바인딩은 화면에 보이는 데이터와 브라우저 메모리상의 데이터가 일치하게 만드는 것입니다.



데이터바인딩에는 위의 그림처럼 2가지, 즉 양방향 바인딩(two-way binding)과 단방향 바인딩(one-way binding)이 있는데, Vue.js는 양방향 바인딩 모델을 사용하고 React.js는 단방향 바인딩 모델을 사용합니다.

데이터가 변했을 때 뷰에 반영되는 것은 둘 다 동일하지만, 뷰에서 이벤트가 일어났을 때 양방향 바인딩은 watcher를 통해 반영되는 반면에 단방향 바인딩은 watcher에 의해 반영되는게 아니라 해당 이벤트에 대한 핸들러함수를 만들어야 하고 그 함수에 의해 반영된다는 점에서 다릅니다.

## 자유도, 타 플러그인 의존도, DOC, 러닝커브 비교

	Vue.js	React.js
자유도	낮음	높음
의존도	낮음	높음
DOC	낮음	보통

러닝커브	낮음	보통
------	----	----

## 성능

[https://krausest.github.io/js-framework-benchmark/2022/table\\_chrome\\_105.0.5195.102.html](https://krausest.github.io/js-framework-benchmark/2022/table_chrome_105.0.5195.102.html)

Name	vue-v3.2.37	react-v17.0.2
Duration for...		
Implementation notes		
<b>create rows</b> creating 1,000 rows (5 warmup runs).	45.6 ± 0.4 (1.00)	48.8 ± 0.2 (1.07)
<b>replace all rows</b> updating all 1,000 rows (5 warmup runs).	44.0 ± 0.3 (1.00)	48.6 ± 0.3 (1.10)
<b>partial update</b> updating every 10th row for 1,000 rows (3 warmup runs). 16x CPU slowdown.	107.0 ± 3.4 (1.00)	123.4 ± 3.0 (1.15)
<b>select row</b> highlighting a selected row. (5 warmup runs). 16x CPU slowdown.	18.9 ± 1.0 (1.00)	36.7 ± 1.1 (1.95)
<b>swap rows</b> swap 2 rows for table with 1,000 rows. (5 warmup runs). 4x CPU slowdown.	27.7 ± 0.5 (1.00)	164.6 ± 1.2 (5.94)
<b>remove row</b>		

- ☐ radix-vv.0.7  
☒ react-v17.0.2  
☐ react-easy-state-v17.0.1 + 6.3.0  
☐ react-focal-v17.0.1 + 0.8.5  
☐ react-hooks-v18.2.0  
☐ react-lab-v17.0.3  
☐ react-mlyn-v0.5.11  
☐ react-mobX-v17.0.1 + 5.15.4
- ☐ varimajs  
☐ vanillajs-1  
☐ vanillajs-wc  
☐ vidom-v0.11.0  
☐ voby-v0.37.0  
☒ vue-v3.2.37  
☐ vuex-jsx-v0.2.0  
☐ wasm-bindgen-v0.2.47

테이블의 row를 선택하는 것, 바꾸는 것만 react.js가 뒤쳐지며 대부분 Vue.js가 이깁니다. 그러나 거의 비슷하게 성능이 좋은 것을 알 수 있습니다.

## 커뮤니티

이 기술을 사용하는데 있어서 이 기술을 만드는 사람이 얼마나 꾸준히 만드는지, 얼마나 많은 사람이 만드는지, 지속적으로 업데이트가 되는지 등은 중요합니다.

	React.js	Vue.js
커밋수	15213	3510
스타수	195k	200k
포크수	40.4k	32.8k
기여자	1579	345

기여자수, 커밋수, 포크수는 React.js가 많으며 Vue.js는 스타수가 앞서는 것을 볼 수 있습니다. 둘 다 훌륭한 지표이지만 React.js가 앞섭니다.

- 2022.09 기준

## 모바일로의 확장성

서비스마다 다르지만 모바일앱으로도 확장할 수 있다고 생각이 되면 생각해야 합니다.

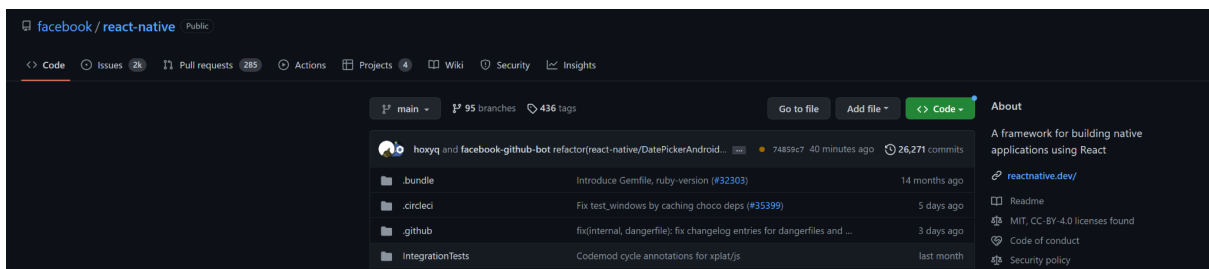
React는 React-native가 있으며 Vue.js는 Native script 또는 Capacitor가 있습니다.

<https://github.com/facebook/react-native>

<https://github.com/ionic-team/capacitor>

<https://github.com/NativeScript/NativeScript>

앞의 링크를 타고 가면 보이는 것처럼 React-native가 커뮤니티 적인 측면에서 앞승을 거두고 있습니다.



## 팀원들의 역량

이렇게 여러가지 것들을 비교해봤는데요. 저 같은 경우는 둘 다 비슷하다고 생각했고 모바일로의 확장성이 필요없는 프로젝트라 생각했습니다. 그렇기 때문에 팀원들의 역량으로 결정하자 라고 생각했고 팀원들이 Vue.js에 대한 숙련도가 더 높아서 Vue.js로 기술을 선택하게 되었습니다.

물론 앞서 살펴본 것 외에도 개발자마다 중요시여기는 것은 다르기 때문에 해당 부분또한 고려해야 합니다. 그러나 앞서 설명한 특징, 자유도, 의존도, 성능, 커뮤니티, DOC, 러닝커브는 필수적으로 분석하고 기술을 도입해야 합니다.

## Q. 워라벨에 대해 어떻게 생각하나요?

이 때 그저 워라벨을 “쉬는 삶은 중요하고 잘 개발하고 잘 쉬는게 좋습니다.”라고 했을 때 좋아하지 않을 면접관들이 많습니다. 이 때문에 조금 더 정형화된 답변이 어쩔 수 없이 필요합니다.

모범답안은 다음과 같습니다.

## 모범답안

저는 오히려 워라벨이라는 워딩이 안좋다 라고 생각합니다. 제가 좋아하는 개발을 하는 것 자체가 삶이기 때문에 오히려 워라벨이라는 워딩 때문에 나누려하다가 제 개발에 방해가 되는 경우가 많았습니다. 예를 들어 6시까지 일하고 그 이후에는 쉬어야지 나는 저녁이 있는 삶이 중요하니까! 라는 마인드는 오히려 일을 끝내는데 방해가 되더라구요.

그래서 그 이후로는 해당 이슈나 어떠한 일을 끝내고 쉰다. 이런 마인드로 개발을 진행하고 있습니다. 제게 워라벨이란 워킹과 라이프를 분리하는 것이 아닌 워킹&라이프로써 이 일을 잘하기 위해 틈틈히 잘 쉬는 것이라 생각합니다.

## Q. 의견 충돌이 났을 때 어떻게 해야하나요?

일단 감정적이지 않아야 합니다. 자신이 어떠한 큰 노력을 했음에도 불구하고 그 노력이 팀적으로 봤을 때 도움이 안되는 경우도 발생하기 때문에 자신이 한 노력이 물거품이 될 수도 있다는 것을 깔고 들어가야 한다고 생각합니다. 무조건 팀적으로 좋은 방향, 예를 들어 UX를 위한다던가 성능을 위한다던가 회사에서 중요시여기는 지표를 기반으로 해당 지표를 기반으로 봤을 때 나의 의견이 더 좋은 건지 그 분의 의견이 더 좋은건지를 기반으로 판단해서 해당 의견을 채택하는게 좋습니다.

## Q. 마지막으로 하고 싶은 질문이 있나요?

해당 회사가 어떤 일을 하고 있는지 구글링을 해서 어떤 프로젝트가 어떻게 되고 있는지 물어보는 것 또는 회사의 문화에 대해 “이런 문화가 참 좋다고 생각하는데요. 혹시 이런 문화는 꼭 지켜야 한다는 문화가 있을까요?” 라는 식으로 회사에 대한 관심을 보여주는 질문이 좋습니다.

<https://story.baemin.com/3359/>

여기까지 오시느라 수고하셨습니다.  
질문 있으시면 언제든지 질문 부탁드립니다.  
좋은 수강평과 별점 5점은 제가 큰 힘이 됩니다. :)  
감사합니다.