2024

한 권으로 마무리하는 직무 면접

제로베이스 면접 실전기출문제

FRONTEND

5,000제 이상의 기출문제 중, 과정 별 **300문제** 엄선
실무에서 물어보는 **빈출 문제+합격자 답변**과 함께 선별

zero-base

제로베이스 소개

안녕하세요. 제로베이스 스쿨입니다.

제로베이스는 대학 교육과 실무의 괴리를 해결하기 위해 바로 실무에 투입이 가능한 인재를 양성하고 있습니다.
취업 · 이직 · 전직을 희망하는 분들을 대상으로 각 과정의 기초부터 결과물을 만드는 것까지 진행하는 취업 집중 교육과정입니다.
현재 제로베이스에선 개발, 데이터, 기획, 마케팅, 디자인 등 다양한 분야에서 취업에 직접 영향을 주는 실무 교육 과정을 운영하고 있으며, 전에 없던 새로운 분야의 취업 교육을 계속해서 선보이고 있습니다.

취업보장 혜택을 자신 있게 내놓을 수 있는 이유

제로베이스에선 과정 졸업 후에도 취업이 되지 않을 경우, 수강료 전액을 환급해 드리는 '취업보장' 혜택을 제공합니다. 어떻게 이런 혜택이 가능할까요?

1. 누적 수강생 10,000명 이상, 데이터로 쌓아 올린 커리큘럼

제로베이스 스쿨은 2016년 패스트캠퍼스 스쿨을 시작으로, 1만명에 달하는 수강생과 함께 국내 최고 수준의 취업 결과를 만들고 있습니다. IT 기업 취업에 필요한 어디에서 찾기 어려운 규모의 온라인 강의 콘텐츠, 과제, 테스트, 최대 규모의 기출문제까지 따라올 수 없는 초격차를 만들어내고 있습니다.

2. 실무 영역을 정의하는 5,000개 이상의 기출문제

입시, 자격증, 공무원 등등. 모두 '기출문제'를 기반으로 공부하는데, 과연 '취업'은 어떨까요? 제로베이스에선 면접 및 채용 현장에서 나온 5,000개 이상의 문제를 분석해, 기업에서 신입 취준생에게 원하는 지식과 역량을 정확하게 정의했습니다.

3. 50만 기업 회원과 함께, 사람인 취업 연계 서비스

서류와 면접만으로는 모든 역량을 검증하기 어려운 시대, 취업을 준비하는 과정까지 들여다 볼 수 있는 '사람인 채용연계' 서비스가 2024년부터 시작되었습니다.
사람인의 50만 기업회원에게 학습 리포트와 포트폴리오, 데모영상을 통해 차별화된 지원자로 먼저 제안을 받을 수 있습니다.
이외에도, 제로베이스의 자세한 취업 서비스가 궁금하다면? 제로베이스 홈페이지를 확인해 보세요!
https://zero-base.co.kr/

기출문제집 활용법

[제로베이스 면접 실전기출문제]는 제로베이스 스쿨에서 다양한 경로를 통해 쌓아 올린 면접 기출문제 은행 자료 중, 일부를 추려 구성한 문제집입니다. 아직 면접을 경험해본 적이 없거나, 당장 눈앞에 면접을 앞둔 분들께는 더욱 큰 도움이 될 수 있을 것 같습니다. 이 자료는 세 가지 파트로 구성되어 있으며, 각 파트별로 용도에 맞게 활용할 수 있습니다.

1. 직무 기출문제 300제

내 직무 면접의 카테고리별 유형을 확인하고, 어떤 질문들이 주로 나오는지 확인해 보세요. 이 면접 질문들은 실제 채용 현장에서 적어도 한 번 이상은 출제된 적 있는 문제들입니다. 양이 너무 많다면? 뒤에 등장하는 빈출문제부터 차근차근 준비해 보세요.

2. 빈출 문제 / 합격 답변 30제

방대한 분량의 면접 질문 중에서도, 특히 높은 빈도로 출제되는 질문입니다. 물론 300가지 질문의 답변을 모두 준비할 수 있다면 좋겠지만, 우리의 시간은 한정되어 있고 때론 선택과 집중을 해야 할 때가 있습니다. 모든 질문에 대비하진 못하더라도, 이 30가지 질문만큼은 꼭 숙지해서 면접장에 들어가세요.

3. 예상문제 10제

당장 면접장에 가야 하는데 준비할 시간이 충분치 않다면? 적어도 이 질문만큼은 대비해 가세요. 기업 입장에선 꼭 확인하고 싶지만, 미리 준비하지 않으면 대답하기 어려운 10가지 문제를 엄선했습니다. 해당 질문들을 출력한 뒤, 아래 빈칸에 자신의 답변 키워드를 메모하여 함께 가지고 가는 것도 추천드려요!

제로베이스는 언제나
언제나 여러분의 취업 성공을 위해 함께 달리겠습니다.

“

본 자료는 아래 기업들을 포함해

총 **232개의 기업** 면접 문제를 바탕으로

제작되었습니다.

”

프론트엔드 개발 직무를 소개할게요.

프론트엔드(Front-end) 사용자가 눈으로 보는 영역을 구축하고, 더 좋은 방향으로 기능을 개발하는 분야입니다. 사용자의 경험 UX(User Experience)을 개선하는 일을 담당합니다. 즉, 프론트엔드 개발자는 사용자가 서비스를 이용하기 위해 웹페이지나 앱 내에서 만나고 경험하는 모든 부분에 관여합니다.

▶ 프론트엔드 개발자는 서비스를 더 흥미로운 방향으로 개선해요.

프론트엔드 개발은 이름 그대로 사용자가 서비스를 이용하기 위해 웹에서 경험을 시작(front)하고 끝(end)내는 모든 부분을 만드는 개발자를 말합니다. 프론트엔드 개발은 크게 웹디자인 구현/성능 최적화/서비스 기능 구현/호환성 관리/보안 관리를 한다고 할 수 있습니다.

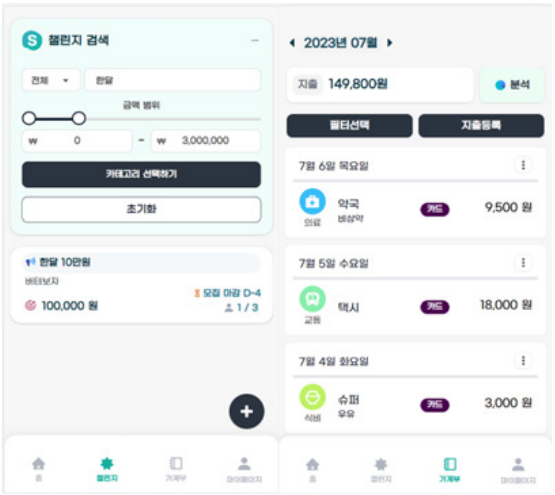


그림 처럼 눈에 보이고 사용자가 경험하는 것을 만드는데요. 이러한 것들을 만들기 위해서 HTML, CSS, JavaScript을 필수로 이용합니다. 이때, HTML과 CSS의 경우 멈춰있는 정적인 화면을 만들 수 있다면 JavaScript는 움직임 까지 세세하게 담은 동적인 화면을 만들 수 있기 때문에 많은 기업에서 필수로 사용하고 있습니다. 또한 리액트를 통해 동적인 화면을 효율적으로 유지보수하고 있는 추세입니다.

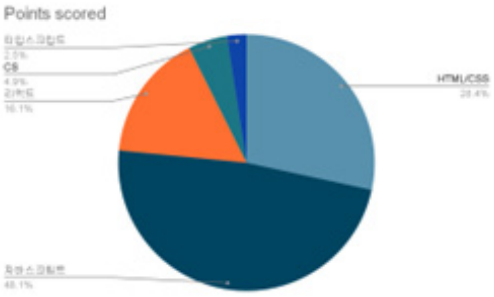
▶ 프론트엔드 채용 시장은 안정적인가요?

웹·앱 어플리케이션의 증가와 사용자 경험 강조로 인해 프론트엔드 개발자에 대한 수요는 꾸준히 늘고 있습니다. 단, 소비자의 눈이 함께 높아지고 있는 만큼 시장에서 나만의 경쟁력을 갖기 위해 사용자 경험 마인드셋을 가진 개발자가 되기 위해 노력해야 합니다.

제가 개발자가 될 수 있을까요?

네! 누구나 개발자가 될 수 있습니다. 제로베이스가 중소기업/중견기업/대기업 면접에 참여한 신입 프론트엔드 개발자를 통해 수집하고 분석한'합격'하는 프론트 개발자의 필수 역량을 공개합니다!

▶ 무스펙으로 취업한 신입 개발자의 합격 노하우

커뮤니케이션	**프론트엔드 개발자는 비개발 직무, 개발 직무의 사람들과 커뮤니케이션을 합니다.** 프론트엔드 개발자는 개발자와 소통하기 위한 기술의 이해가 있어야 할 뿐만 아니라 비개발 직무, 특히 디자이너와의 많은 소통을 필요로 합니다. 따라서 면접 상황에서도 원활한 커뮤니케이션을 하는 모습을 보여준 지원자가 좋은 결과를 보여주었습니다.
나만의 강점	**프론트엔드는 눈에 보이는 것으로 어필해야 하는 만큼 내 능력도 눈에 보이게 어필도 잘 되어야 합니다.** 면접을 볼 때에도 단순히 기술 스택으로 '리액트' 'saas'를 사용해 봤어요가 아닌, 깃허브로 볼 수 있듯이 제가 정말 이걸 만들어 봤는데, 이걸 만들면서 이러한 트러블 슈팅, 문제해결 과정을 겪었습니다. 그로 인해 저는 무엇을 배웠습니다를 보여주는 지원자일 수록 좋은 결과를 보여줬습니다.
좋은 '습관'	**좋은 습관이 곧 좋은 실력을 만듭니다.** 효율적으로 코드를 짜기 위해서는 평소 코딩하는 습관이 중요합니다. 지금 당장은 기술적 능력이 부족하더라도 좋은 습관을 갖고 일을 하게 된다면 나중에 일이 쌓여 갈 때에도 큰 문제가 발생하지 않습니다. 면접관은 이러한 습관을 통해 지원자의 생각의 전개를 파악하는 것입니다.
'기본 개념'의 이해	**JavaScript에 대한 기초 지식은 탄탄해야 합니다.** 프론트엔드 개발자는 퍼블리셔와 협업하는 만큼, 보여지는 것 외의 실현 방안에 대한 깊은 이해가 필요한데요. 면접 질문에서 호이스팅의 개념, 쿠키, HTTP의 개념 등 기본 이해도를 묻는 질문이 면접 시간의 60%를 차지하는 것을 보면 기본 개념을 얼마나 중요시 하는지 더욱 알 수 있습니다. 

좋은 프론트엔드 개발자가 되는 방법

각 역량을 학습하면 무스펙의 '나'도 취업에 성공한 신입 개발자가 될 수 있습니다.

'커뮤니케이션' 능력을 키워보세요.	**협업 프로젝트를 진행해 보세요.** 프론트엔드 개발자는 개발자 중에서도 비개발자와 가장 많은 소통을 하는 사람입니다. 따라서 사이드 프로젝트 경험을 통해 PM, 디자이너 등과 소통하는 법을 배우고, 소통 과정 속 문제를 해결해 가는 능력을 쌓아가세요. "비개발 직군이랑 협업하면서 어려웠던 적이 있나요?" 등의 질문을 미리 준비할 수 있습니다.
'나만의 강점'을 만드세요.	**포트폴리오를 제작해 보세요.** 자신이 흥미있게 생각하는 서비스를 기반으로 제작한 포트폴리오는 직무의 이해도가 높다는 것을 자연스럽게 어필할 수 있습니다. 이를 통해 "성능 최적화 경험을 말해주세요." 등의 질문을 미리 준비할 수 있습니다.
좋은 '습관'을 쌓아가세요.	**오답노트를 만들어 보세요.** 조금씩 이라도 꾸준히 공부하는 습관을 기르고 오답노트 작을을 통해 문제 해결 능력을 쌓아보세요. "만약 데이터를 추가 및 삭제해야할 경우 둘 중 어떤 자료구조가 용이한가요?" 와 같이 문제 해결을 묻는 질문에도 막힘 없이 답변할 수 있습니다.
'기본 개념'을 탄탄히 잡아보세요.	**스케쥴표에 따라 꾸준히 공부하세요.** 프론트엔드 개발자는 년도별로 새로운 기술의 트렌드가 변화하고 있지만 웹접근성, 라이브러리, 구조 관련 학습에 대한 질문엔 흔들림이 없기 때문에 **확실한 준비**가 필요합니다. **"Webpack이란 무엇인가요?", "이벤트 위임에 대해서 설명해 주세요."**와 같은 변함없는 질문을 놓치지 말아야 합니다.

▶ 제로베이스에선 프론트엔드 개발자와 이렇게 성장합니다.

제로베이스 프론트엔드 스쿨에서

원활한 커뮤니케이션능력자로 변화하세요!
눈에띄는 강점으로 모두에게 주목 받아 보세요!
공부하는 습관을 만들어 믿고 따를 수 있는 개발자로 성장하세요!
기본 개념이 탄탄해 신뢰있는 개발자가 되어보세요!

제로베이스 프론트엔드 스쿨 교육 과정		
기간	PART1. 프론트엔드 기본 학습	PART2. 프론트엔스 심화학습
	1~3개월 차	4~6개월 차
학습 내용	온라인 강의 수강/매주 코딩 테스트/프로젝트 과제 수행	
	HTML/CSS기본 |HTML/CSS활용 | JavaScript	React | Vue
	자료구조 | 알고리즘 | 과제 해설	TypeScript | CS/Git | 과제 해설
교육과정 운영사항	개인 프로젝트 과제 진행 및 평가	개인 or 팀 프로젝트 과제 진행 및 평가
	탈락제도 / 기수 유예제도 / 기수별 수강생 커뮤니티 운영	

제로베이스에선 포기하지 않고 끝까지 함께 갈 수 있는 커리큘럼을 제공합니다.
더 늦지 않게 시작하고 빠른 취업 성공의 길로 나아가세요!

PART

1

기출문제

No	카테고리	문제
문제1	개발경험	최근 자주 사용하는 문법이 있나요?
문제2	개발경험	최신 기술 트렌드를 추적하는 자신만의 방식이 있나요?
문제3	개발경험	프로젝트내에서 협업을 하는 방식이 있나요?
문제4	개발경험	프로젝트 내에서 담당한 부분과 기술적으로 어려웠던 점이 있었나요?
문제5	개발경험	같이 일하기 싫은 팀원은 어떤 팀원인가요?
문제6	개발경험	비동기적인 작업을 동기적으로 코딩하는 방법을 설명해주세요.
문제7	개발경험	협업 경험이 있나요? 적극적으로 문제를 해결한 경험이 있으면 공유해주세요.
문제8	개발경험	고객이 제작한 페이지 화면 로딩 속도가 느리다고 피드백이 들어왔습니다. 어떻게 대처할 것인지 설명해주세요.
문제9	개발경험	git을 사용해 협업 경험이 있을까요? 어떤 규칙을 사용했는지 설명해주세요.
문제10	개발경험	SEO를 최적화하기 위해 어떤 작업을 해봤는지 공유해주세요.
문제11	개발경험	지원자님은 평소 기술 향상을 위해 어떤식으로 노력을 기울이고 있는지 설명해주세요.
문제12	개발경험	Context API에 대해 설명해주세요.
문제13	개발경험	본인은 어떤 개발자가 되고 싶으신가요?
문제14	개발경험	10년 후에도 개발자를 하고 있을 것 같나요?
문제15	개발경험	기억에 남는 프로젝트가 있나요?
문제16	개발경험	프로젝트를 하면서 문제를 해결한 경험이 있나요?
문제17	개발경험	커뮤니케이션에 마찰이 있다면 어떻게 해결하는 편이신가요?
문제18	개발경험	새로운 언어에 대해 배우고 싶은 언어가 있나요?
문제19	개발경험	비전공자로써 전공자와의 차이를 줄이기 위해 노력한 것이 있나요?
문제20	개발경험	즐겨보는 개발 유튜버가 있나요?

No	카테고리	문제
문제21	개발경험	퇴근 후 약속이 있는데 갑작스런 야근을 하게된다면 어떻게 하실건가요?
문제22	개발경험	개발자가 되신 이유가 있으신가요?
문제23	개발경험	본인이랑 잘 맞는 팀원과 그렇지 않는 팀원은 누구인가요?
문제24	개발경험	비개발 직군이랑 협업하면서 어려웠던 적이 있나요?
문제25	개발경험	본인이 사용하는 가장 좋아하는 서비스가 있나요?
문제26	개발경험	현재 우리 웹 페이지의 어떤 부분을 개선시키고 싶나요?
문제27	개발경험	어떤 개발자가 되고 싶나요?
문제28	개발경험	가장 자신 있는 언어는 무엇인가요?
문제29	개발경험	개발자가 왜 되려고 하나요?
문제30	개발경험	본인이랑 잘 맞는 팀원과 그렇지 않는 팀원은 누구인가요?
문제31	개발경험	현재까지 진행한 프로젝트 중 발생한 이슈를 어떻게 해결했는지 하나만 설명해 주세요.
문제32	개발경험	배포한 웹 사이트의 성능을 개선하기 위해 어떠한 노력을 했나요?
문제33	개발경험	가장 최근에 해결한 기술적인 문제는 무엇이었나요?
문제34	개발경험	해당 프로젝트에서 자신의 역할은 무엇이었나요?
문제35	개발경험	다국어 페이지를 제공하는 여러 방법에 관해 설명해주세요.
문제36	CS	브라우저의 동작원리를 설명해 주세요.
문제37	CS	스택과 큐의 차이점을 설명해 주세요.
문제38	CS	프로젝트에서 CI/CD는 어떻게 진행했나요?
문제39	CS	불변성에 대해 어떻게 생각하나요?
문제40	CS	의존성 주입에 대해 알고있나요?

No	카테고리	문제
문제41	CS	프로세스와 스레드의 차이를 설명해 주세요.
문제42	CS	CDN이란 무엇인가요?
문제43	CS	브라우저를 열고 주소를 입력했을 때 일어나는 일을 설명해주세요.
문제44	CS	HTTP와 HTTPS 통신 방식의 차이를 설명해주세요.
문제45	CS	브라우저 저장소의 종류와 차이점을 설명해주세요.
문제46	CS	RESTful API에 대해 설명해주세요.
문제47	CS	서버사이드렌더링(SSR)과 클라이언트사이드렌더링(CSR)의 차이점을 설명해주세요.
문제48	CS	MVVM 모델에 대해 설명해주세요.
문제49	CS	이진 트리를 사용해 보셨나요? 실제 사용 사례를 아신다면 설명해주세요.
문제50	CS	웹 소켓을 사용해본 경험이 있으면 공유해주세요.
문제51	CS	디자인 시스템이 뭔가요? 왜 필요한지 설명해주세요.
문제52	CS	객체지향 프로그래밍이 무엇인지 설명해주세요.
문제53	CS	DNS에 대해 설명해주세요.
문제54	CS	라운드 로빈 알고리즘과 실제 사례를 설명해주세요.
문제55	CS	프로세스와 스레드를 설명해주세요.
문제56	CS	HTTP 응답 상태 코드 중, 대표적인 것들에 대해 설명해주세요.
문제57	CS	CSR과 SSR의 장단점을 설명하세요.
문제58	CS	Reflow, Repaint에 대해 설명해주세요.
문제59	CS	monorepo에 대해 설명하세요.
문제60	CS	브라우저 렌더링 과정을 설명하세요.

No	카테고리	문제
문제61	CS	웹 페이지에서 로딩 속도를 향상시키기 위한 방법에 대해 설명하세요.
문제62	CS	버전 관리 시스템을 사용하는 이유와 장점은 무엇인지 설명하세요.
문제63	CS	스켈레톤 UI에 대한 것과 장점에 대해 설명하세요.
문제64	CS	MySQL의 index에 대해 설명해주세요.
문제65	CS	웹 접근성에 대해서 설명해주세요
문제66	CS	웹 접근성을 개선 시킨 경험이 있는지 설명해주세요.
문제67	CS	Array와 LinkedList 사이의 차이점에 대해 설명해주세요
문제68	CS	만약 데이터를 추가 및 삭제해야할 경우 둘 중 어떤 자료구조가 용이한가요?
문제69	CS	사용자가 입력한 값을 서버에 직접 요청하여 결과를 보여줄 때, 사용자가 많은 입력을 보내게 되면 서버에 부하가 심해질 수도 있는 문제를 해결하려면 어떻게 하실건가요?
문제70	CS	HTTP 프로토콜 메서드 GET과 POST의 차이 점에 대해서 설명해 주세요.
문제71	CS	CSRF나 공격을 막는 방법은?
문제72	CS	TDD란 무엇인가요?
문제73	CS	Static Site Generator에 대해서 아시나요?
문제74	CS	쿠키(Cookies)와 세션저장소(sessionStorage)와 로컬저장소(localStorage)의 차이점을 설명해주세요.
문제75	CS	Progressive rendering이란 무엇인가요?
문제76	CS	인터프러터와 컴파일러는 어떤 점에서 차이가 있나요?
문제77	CS	반응형(Responsive) 디자인은 적응형(Adaptive) 디자인과 어떤 차이점이 있나요?
문제78	CS	전통적으로, 웹사이트의 assets을 여러 도메인으로 서빙했을 때 장점은 무엇인가요?
문제79	CS	URL로 접속했을 때 어떤 플로우로 화면에 웹사이트가 그려지는지 네트워크 관점에서설명해주세요.
문제80	CS	Long-Polling과 Websocket, Server-Sent Event에 대해 설명해주세요.

No	카테고리	문제
문제81	HTML/CSS	css에서 margin과 padding의 차이점에 대해 설명해 주세요.
문제82	HTML/CSS	포지션 static, relative, absolute, fixed 각 특징에 대해서 설명해주세요.
문제83	HTML/CSS	sass, scss를 사용해본적 있나요? css와 차이를 설명해주세요.
문제84	HTML/CSS	CSS 박스 모델에 대해 설명해주세요.
문제85	HTML/CSS	Flex 속성을 설명해주세요.
문제86	HTML/CSS	cascading에 관해 설명해주세요.
문제87	HTML/CSS	크로스 브라우징 경험이 있으신가요? 있다면 말씀해 주세요.
문제88	HTML/CSS	Box Model에 대해 설명하고 웹브라우저에서 어떻게 동작하는지 설명 하세요.
문제89	HTML/CSS	CSS 전처리기의 예시와 장점을 설명하세요.
문제90	HTML/CSS	로컬 스토리지와 세션 스토리지에 대해 설명해주세요
문제91	HTML/CSS	flex의 특징을 설명해주세요
문제92	HTML/CSS	margin-collapse 현상이 무엇인가요?
문제93	HTML/CSS	CSS에 대해서 설명해주세요
문제94	HTML/CSS	HTML과 XHTML의 차이를 설명해주세요
문제95	HTML/CSS	CSS의 성능 최적화에 대해 설명해주세요
문제96	HTML/CSS	WAI-ARIA에 대해 설명해주세요.
문제97	HTML/CSS	HTML과 CSS의 기본적인 차이점을 설명해주세요
문제98	HTML/CSS	CSS의 Flexbox와 Grid 레이아웃에 대해 설명해 주세요.
문제99	HTML/CSS	HTML은 무엇의 약자이고 어떤 걸 의미하는지 또한 HTML 파일에서 <!DOCTYPE html> 이 의미하는 걸 설명해 주세요.
문제100	HTML/CSS	CSS는 무엇의 약자이고 어떤 걸 의미하는지 설명해 주세요.

No	카테고리	문제
문제101	HTML/CSS	웹 페이지의 성능을 고려할 때 HTML 파일에서 <script>태그를 어디에 위치시키는 게 좋을지 설명해 주세요.
문제102	HTML/CSS	<head>태그 내에 <script>태그를 위치시키는 건 잘못된 건가요?
문제103	HTML/CSS	Flex와 Grid의 차이점에 대해서 설명해보세요.
문제104	HTML/CSS	MVVM패턴과 Flux패턴의 차이점에 대해서 설명해보세요.
문제105	HTML/CSS	DOCTYPE이 무엇을 하는 것인가요?
문제106	HTML/CSS	표준 모드(Standards Mode)와 쿼스 모드(Quirks Mode)의 다른 점은 무엇인가요?
문제107	HTML/CSS	XML과 XHTML의 다른 점은 무엇인가요?
문제108	HTML/CSS	XHTML을 이용한 페이지의 한계점은 무엇이 있나요?
문제109	HTML/CSS	application/xhtml+xml으로 지정한 페이지에 어떠한 문제가 있나요?
문제110	HTML/CSS	data-* 속성은 무엇을 하는 것인가요? 사용했을 때 이점은 무엇인가요?
문제111	HTML/CSS	HTML5를 오픈 웹 플랫폼(Open Web Platform)으로 생각해 본다면, 어떤 것들로 구성돼 있을까요?
문제112	HTML/CSS	CSS <link> 를 <head></head> 사이에 쓰는 것과 JS <script> 를 </body> 닫기 태그 전에 사용하는 것은 좋은 사용법일까요? 어디에 배치하는 게 좋을까요?
문제113	HTML/CSS	이미지 태그에 srcset 속성을 사용하는 이유는 무엇인가요? 브라우저가 이 속성을 가진 콘텐츠를 평가할 때 사용하는 과정을 설명해보세요.
문제114	HTML/CSS	id와 class의 차이점에 관해서 설명해주세요.
문제115	HTML/CSS	reset CSS가 무엇인지, 어떻게 유용한지 설명해주세요.
문제116	HTML/CSS	Floats가 어떻게 동작하는지 설명해주세요.
문제117	HTML/CSS	z-index에 관해 설명해주세요.
문제118	HTML/CSS	BFC(Block Formatting Context)에 관해 설명해주세요.
문제119	HTML/CSS	클리어링(Clearing) 기술에는 어떤 것들이 있으며, 어떨 때 어떻게 사용하는 것이 적절한지 설명하세요.
문제120	HTML/CSS	CSS 스프라이트(CSS Sprites)를 설명하고, 페이지나 사이트를 어떻게 향상하는지 설명하세요.

No	카테고리	문제
문제121	HTML/CSS	Image Replacement를 사용해야 할 때, 선호하는 기술과 언제 사용하는지를 설명해주세요.
문제122	HTML/CSS	브라우저 스펙 차이에 따른 스타일링 이슈를 수정하기 위해서 어떻게 접근하나요?
문제123	HTML/CSS	기능이 제약된 브라우저를 위해서 어떤 방식으로 페이지를 만드나요?
문제124	HTML/CSS	시각적으로 보이지 않고 스크린 리더에서만 가능하게 하는 방법에 관해 설명해주세요
문제125	HTML/CSS	미디어 쿼리(media queries)를 사용한 적이 있나요? 혹은 모바일에 맞는 layout과 CSS를 사용한 적이 있나요?
문제126	HTML/CSS	인쇄하기 위해 웹페이지를 어떻게 최적화 하나요?
문제127	HTML/CSS	효율적인 CSS를 작성하기 위한 비법(gotchas)은 어떤 게 있나요?
문제128	HTML/CSS	페이지에서 표준 폰트가 아닌 폰트 디자인을 사용할 때 어떤 방식으로 처리하시나요? (웹폰트를 제외하고)
문제129	HTML/CSS	CSS Selector가 어떠한 원리로 동작하는지 설명해주세요.
문제130	HTML/CSS	pseudo-elements에 관해서 설명하고 어디에서 사용되는지 이야기해보세요.
문제131	HTML/CSS	box model에 관해 설명하고 브라우저에서 어떻게 동작하는지 설명해주세요.
문제132	HTML/CSS	* { box-sizing: border-box; }은 무엇이고 사용했을때 이점은 무엇인가요?
문제133	HTML/CSS	inline과 inline-block의 차이점은 무엇인가요
문제134	HTML/CSS	요소를 배치하는 방법(relative, fixed, absolute, static) 간의 차이는 무엇인가요?
문제135	HTML/CSS	CSS에서 'C'는 Cascading을 의미합니다. Cascading에 관해서 설명해주세요. 또 cascading system의 장점은 무엇인가요?
문제136	HTML/CSS	display 속성에 대해 아는 만큼 이야기해 주세요.
문제137	HTML/CSS	display block 과 inline 요소의 차이점은 무엇인가요?
문제138	HTML/CSS	시맨틱 마크업이란 무엇인가요?
문제139	HTML/CSS	input type 종류에 대해 8개 이상 이야기해 주세요.
문제140	HTML/CSS	input 입력값 형식을 제어하는 방법은 무엇이 있을까요?

No	카테고리	문제
문제141	HTML/CSS	button type 종류 3가지를 이야기해 주세요.
문제142	HTML/CSS	a 태그에 이메일을 연결할 경우 href를 적는 방법을 설명해 주세요.
문제143	HTML/CSS	a 태그에 전화번호를 연결할 경우 href를 적는 방법을 설명해 주세요.
문제144	HTML/CSS	a 태그의 target 속성 값에 대해 아는 대로 설명해 주세요.
문제145	HTML/CSS	strong 과 b 태그의 차이점은 무엇인가요?
문제146	HTML/CSS	Tabnabbing 을 알고 있나요?
문제147	HTML/CSS	Tabnabbing을 예방하기 위한 방법은 무엇이 있나요?
문제148	HTML/CSS	ol 과 ul 의 차이점은 무엇일까요?
문제149	HTML/CSS	box-model 에 대하여 설명해 주세요.
문제150	HTML/CSS	border-box 와 content-box 의 차이점은 무엇인가요?
문제151	HTML/CSS	reset.css 에 대해 알고 있나요? 어떻게 만들어 사용하는지 설명해 주세요.
문제152	HTML/CSS	기본 input 스타일을 제거 하기 위한 방법은 무엇이 있을까요?
문제153	HTML/CSS	홀수번째의 li 만 선택하는 선택자는 무엇인가요?
문제154	HTML/CSS	nth-child와 nth-of-type의 차이점은 무엇인가요?
문제155	HTML/CSS	자식요소가 없을 경우를 선택하는 선택자는 무엇인가요?
문제156	HTML/CSS	클래스명에 ui 가 포함된 모든 button을 선택하는 선택자는 무엇인가요?
문제157	HTML/CSS	http://로 시작하는 경로를 가진 모든 a를 선택하는 선택자는 무엇인가요?
문제158	HTML/CSS	이미지 src 확장자가 .png 끝나는 img를 선택하는 선택자는 무엇인가요?
문제159	HTML/CSS	p 옆 div를 가리키는 형제 셀렉터는 무엇인가요?
문제160	HTML/CSS	가상 클래스 선택자란 무엇인가요?

No	카테고리	문제
문제161	HTML/CSS	가상 요소 선택자란 무엇인가요?
문제162	HTML/CSS	disabled 와 readonly의 차이점은 무엇인가요?
문제163	HTML/CSS	em과 rem의 차이점은 무엇인가요?
문제164	HTML/CSS	vw, vh에 대해 설명해 주세요.
문제165	HTML/CSS	visibility: hidden과 display:none 의 차이점은 무엇인가요?
문제166	HTML/CSS	화면이 스크롤이 되더라도 배경이미지는 고정되어 있게 하려면 어떻게 해야 할까요?
문제167	HTML/CSS	한 줄을 넘지 않도록 말줄임처리를 하는 방법은 무엇인가요?
문제168	HTML/CSS	2줄 이상일 때, 말줄임처리를 하는 방법을 설명해 주세요.
문제169	HTML/CSS	position: fixed와 sticky의 차이점은 무엇인가요?
문제170	HTML/CSS	background-size contain과 cover의 차이점은 무엇인가요?
문제171	HTML/CSS	값이 상속되는 css 요소에는 어떤 것들이 있나요?
문제172	HTML/CSS	브라우저 호환성 및 벤더프리픽스 사용여부를 확인하는 방법이 있을까요?
문제173	HTML/CSS	transition이 가능한 css 프로퍼티는 무엇일까요?
문제174	HTML/CSS	CSS로 100px 정방형 사이즈의 빨간 박스가 5초 동안 left: 0 → 100px 로 무한 반복으로 이동하는 애니메이션을 구현하여 주세요.
문제175	HTML/CSS	한글/영문 폰트를 따로 지정할 때 font-family 선언 방법을 설명해 주세요.
문제176	HTML/CSS	가로폭 375 미만의 해상도일 경우 미디어쿼리는 어떻게 될까요?
문제177	HTML/CSS	justify-content: space-between과 space-around의 차이점은 무엇인가요?
문제178	HTML/CSS	sass 7-1 pattern에 대해 알고 있나요?
문제179	HTML/CSS	sass mixin, function 은 무엇인가요?
문제180	HTML/CSS	div 안에 div 요소가 하나만 있다고 가정 했을 때, 항상 중앙에 위치하도록 하는 방법은 무엇인가요? (flexbox를 사용하여 대답하세요. 혹시 모르겠다면 다른 방법으로도 가능합니다.)

No	카테고리	문제
문제181	HTML/CSS	aria-label 에 대해 아는가? 기타로 aria-* 에 대해 알고 있는 것이 있다면 설명해 주세요.
문제182	Javascript	실행컨텍스트의 관점에서 클로저에 대해 이야기 해 주세요.
문제183	Javascript	실행 컨텍스트에 대해 설명해 주세요.
문제184	Javascript	실행 컨텍스트를 시작하는 방법을 설명하세요.
문제185	Javascript	메서드 또는 함수를 호출했을때 일어나는 일을 간략히 설명하세요.
문제186	Javascript	this값이 정해지는 방법을 설명해보세요.
문제187	Javascript	var,let,const 차이점을 설명해 주세요.
문제188	Javascript	호이스팅이란 무엇인지 설명해 주세요.
문제189	Javascript	atomic design에 대해 아시나요.
문제190	Javascript	OOP에 대해 아시나요?
문제191	Javascript	비동기 처리를 해본 적 있나요?
문제192	Javascript	Promise에 대해 설명해 주세요.
문제193	Javascript	debounce , throttle에 대해 설명해 주세요.
문제194	Javascript	AJAX에 대해 설명해 주세요.
문제195	Javascript	ES6에 추가된 문법으론 어떤 게 있나요?
문제196	Javascript	클로저는 무엇인가요? 원리와 왜 사용하는지 설명해주세요.
문제197	Javascript	Async, Await과 Promise의 차이를 설명해주세요.
문제198	Javascript	this 용법을 아는대로 설명해주세요.
문제199	Javascript	스크립트 언어란 무엇인지 설명해주세요.
문제200	Javascript	Event Loop에 대해 설명해주세요.

No	카테고리	문제
문제201	Javascript	javascript 성능 최적화를 위해 시도한 경험을 공유해주세요.
문제202	Javascript	CORS를 대처하는 방법과 우회하는 방법을 설명해주세요.
문제203	Javascript	ESLint에 대해 설명해주세요.
문제204	Javascript	JWT 개념과 사용하는 이유를 설명해주세요.
문제205	Javascript	use strict 개념과 이것을 사용하는것의 장점과 단점을 설명해주세요.
문제206	Javascript	forEach와 map의 차이점이 무엇인지 설명해주세요.
문제207	Javascript	Null, undefined, undeclared, NaN 에 대해 설명해주세요.
문제208	Javascript	event.preventDefault() 함수가 어떤 상황에 쓰이는지 설명하세요.
문제209	Javascript	Intersection Observer가 무엇인지 설명하세요.
문제210	Javascript	이벤트 버블링이란 무엇이며, 막을 수 있는 방법을 설명하세요.
문제211	Javascript	클로저가 무엇인지, 장점에대해 설명하세요.
문제212	Javascript	가비지 콜렉터란 무엇인지 설명하고, 메모리 누수에 예시를 들어보세요.
문제213	Javascript	Babel과 Webpack이 무엇인지 설명하세요.
문제214	Javascript	`data-` Attribute에 대해 설명해 주세요. 사용했을 때의 이점은 무엇인가요?
문제215	Javascript	자바스크립트에서 가비지 컬렉터란 무엇인가요?
문제216	Javascript	자바스크립트에서 변수와 할당에 대해서 설명해주세요.
문제217	Javascript	자바스크립트는 클래스 기반 객체 지향 언어인가요? 프로토 타입 기반 객체 지향 언어인가요?
문제218	Javascript	프로토타입에 대해서 설명해주세요
문제219	Javascript	클래스에 대해 간략히 설명해주세요
문제220	Javascript	디바운스와 스로틀의 차이에 대해서 설명해주세요

No	카테고리	문제
문제221	Javascript	DOM이란 무엇인지 설명해주세요.
문제222	Javascript	SPA에 대해 설명해주세요
문제223	Javascript	이벤트 버블링과 캡처링에 대해 설명해주세요.
문제224	Javascript	Babel이란 무엇인가요?
문제225	Javascript	자바스크립트에서 데이터 타입에 대해 설명해주세요.
문제226	Javascript	자바스크립트에서 함수란 무엇인가요?
문제227	Javascript	스코프란 무엇인가요?
문제228	Javascript	스코프 체인에 대해서 설명해주세요.
문제229	Javascript	종속성 배열에 대해 설명해주세요.
문제230	Javascript	리액트에서 key를 왜 사용하는지 설명해주세요.
문제231	Javascript	자바스크립트에서 원시(primitive) 타입과 참조(reference) 타입에 대해서 설명해 주세요.
문제232	Javascript	자바스크립트에서 클로져(Closures)에 대해서 설명해주세요.
문제233	Javascript	얕은 복사(Shallow Copy)와 깊은 복사(Deep Copy)에 대해서 설명해 주세요.
문제234	Javascript	완벽한 깊은 복사(Deep Copy)를 하는 방법에 대해서 설명해 주세요.
문제235	Javascript	콜백 지옥과 콜백 지옥 해결법에 대해서 설명 해주세요.
문제236	Javascript	어떤 프레임워크를 사용하여 개발을 시작했나요? 왜 그 프레임워크를 선택했나요?
문제237	Javascript	== 와 ===의 차이는 무엇인가요?
문제238	Javascript	클래스형 컴포넌트와 함수형 컴포넌트의 차이는 무엇일까요?
문제239	Javascript	비동기 함수에 대해서 설명해 보세요.
문제240	Javascript	<script> , <script async> 와 <script defer> 의 차이점에 관해 설명해 주세요

No	카테고리	문제
문제241	Javascript	자바스크립트와 ECMAScript의 차이점은 무엇인가요?
문제242	Javascript	변수란 무엇이고 왜 필요한가요?
문제243	Javascript	리터럴이란 무엇인가요?
문제244	Javascript	표현식과 문의 차이는 무엇인가요?
문제245	Javascript	자바스크립트의 데이터타입을 모두 나열해 주세요.
문제246	Javascript	자바스크립트는 동적 타입 언어입니다. 정적 타입 언어와 동적 타입 언어의 차이는 무엇인가요?
문제247	Javascript	for 문과 while 문의 차이는 무엇이며 어떤 경우 사용하나요?
문제248	Javascript	객체란 무엇인가요?
문제249	Javascript	원시값과 객체의 차이는 무엇인가요?
문제250	Javascript	익명함수(anonymous functions)는 주로 어떤 상황에서 사용하나요?
문제251	Javascript	호스트 객체(Host Objects)와 네이티브 객체(Native Objects)의 차이점은 무엇인가요?
문제252	Javascript	Function.prototype.call, Function.prototype.apply, Function.prototype.bind의 차이점은 무엇인가요?
문제253	Javascript	DOM이란 무엇인가요?
문제254	Javascript	이벤트 위임(event delegation)에 대해서 설명해주세요.
문제255	Javascript	load 이벤트와 DOMContentLoaded event의 차이점은 무엇인가요?
문제256	Javascript	AMD와 CommonJS는 무엇이고, 이것들에 대해 어떻게 생각하시나요?
문제257	Javascript	그리드 시스템(Grid system)을 사용한 적이 있나요? 있다면 어떠한 것을 선호하나요?
문제258	React	클래스 컴포넌트와 함수형 컴포넌트의 차이가 무엇인가요?
문제259	React	React Hooks 에 대해 자세히 설명해주세요
문제260	React	리액트에서 메모이제이션을 어떻게 활용할수 있나요?

No	카테고리	문제
문제261	React	상태관리 라이브러리에 대해 설명해주세요
문제262	React	선호하는 상태관리가 있나요? 있다면, 이유와 함께 이야기해 주세요.
문제263	React	redux에 대하여 설명해 주세요
문제264	React	Virtual DOM 작동 원리를 설명해주세요.
문제265	React	React 라이프 사이클을 간단하게 설명해 주세요.
문제266	React	useMemo와 useCallback에 대해 설명해주세요.
문제267	React	리액트에서 왜 useState를 사용하는지 설명해주세요.
문제268	React	Redux, Recoil 등, 상태관리 라이브러리 사용한 경험을 공유해주세요.
문제269	React	Flux 패턴에 대해 설명하세요.
문제270	React	React의 주요 특징에 대해 설명하세요.
문제271	React	React에서 말하는 컴포넌트에 대해 설명하세요.
문제272	React	State와 Props의 차이점에 대해 설명하세요.
문제273	React	컴포넌트 간의 통신 방법에는 어떤 것들이 있는지 설명하세요.
문제274	React	Props Drilling 에 대해 설명하세요.
문제275	React	React Hook에 대해 설명하세요.
문제276	React	state를 직접 변경하지 않고 setState를 사용하는 이유를 설명하세요.
문제277	React	함수형 컴포넌트에서 언마운트된 때를 감지하려면 어떻게 해야할까요?
문제278	React	React Router를 사용하는 이유에 대해 설명하세요.
문제279	React	virtual DOM이 무엇인가요? virtual DOM이 좋은 이유에 대해서 설명하세요.
문제280	React	"React에 성능 향상을 위해 최적화를 해 본 경험이 있나요? 혹은 useMemo와 useCallback 메소드를 활용해 최적화하는 원리에 대해서 설명하세요."

No	카테고리	문제
문제281	React	useCallback의 동작원리에 대해 설명하세요.
문제282	React	Redux와 Mobx 중에 보통 Redux를 사용하는 이유에 대해 생각해보시고 설명하세요.
문제283	React	React 에서 key 속성은 어떤 역할을 하는지 설명하세요.
문제284	React	React Hooks의 사용에 대한 권장 사항과 주의할 점은 무엇인가요?
문제285	React	React 에서 리렌더링이 일어나는 경우에 대해 설명하세요.
문제286	React	React의 suspense와 lazy loading을 사용하여 어떻게 컴포넌트를 비동기적으로 로딩하는지 설명해주세요.
문제287	React	React의 Error Boundary가 무엇이며 어떻게 오류 처리를 담당하나요?
문제288	React	React의 Portals가 무엇이며 어떻게 사용하나요?
문제289	React	React의 Portals의 주의사항을 설명하세요.
문제290	React	리액트 컴포넌트가 무엇인지 설명해주세요
문제291	React	Next.js의 특징을 설명해주세요.
문제292	React	리액트에서 만약 세미나를 열게된다면, 어떤 주제로 열고 싶나요?
문제293	React	React 컴포넌트 생명주기에 대해 설명해주세요.
문제294	React	props 드릴링에 대해 설명해 주세요.
문제295	React	스토리북에 대해서 아시나요?
문제296	Typescript	타입스크립트의 특징과 사용 경험을 공유해주세요.
문제297	Typescript	TypeScript를 사용해본 적이 있나요? 사용해보셨다면 어떠셨나요?
문제298	Typescript	타입 스크립트를 사용하는 이유에 대해서 설명해 주세요.
문제299	Vue.js	Vue.js에서 사용하는 양방향 데이터 바인딩의 원리를 설명해주세요.
문제300	Vue.js	Vite를 사용해보셨나요? 기존 웹팩과 차이점을 설명해주세요.

PART

2

빈출문제

Javascript

문제1 ▶ Event Loop에 대해 설명해주세요.

합격 답변

Event Loop는 JavaScript의 비동기 동작 원리를 이해하는 데 핵심적인 개념입니다. 여기에 대한 설명은 다음과 같습니다:

JavaScript는 싱글 스레드 언어입니다. 즉, 한 번에 하나의 작업만 처리할 수 있습니다. 하지만 웹 애플리케이션이나 서버에서 동시에 많은 작업을 처리해야 하는 경우가 많은데, 이때 Event Loop가 핵심적인 역할을 합니다.

Event Loop의 주된 역할은, Call Stack과 Callback Queue 사이에서 실행될 코드를 조율하는 것입니다. Call Stack에는 현재 실행 중인 코드의 함수 호출이 쌓여 있고, Callback Queue에는 비동기 처리가 완료되었을 때 실행될 콜백 함수들이 대기하고 있습니다.

작동 방식은 다음과 같습니다:

Call Stack에 실행할 함수가 있으면, 그 함수를 실행합니다.
Call Stack이 비어있으면, Event Loop는 Callback Queue에서 대기 중인 콜백 함수를 Call Stack으로 옮겨 실행합니다.
이 과정은 반복되면서 비동기 이벤트 처리, 타이머, HTTP 요청의 결과 등을 순차적으로 처리합니다.
이 과정 덕분에 JavaScript는 비동기 이벤트를 효율적으로 관리하며, 싱글 스레드이면서도 비동기 작업을 동시에 처리할 수 있는 것처럼 보이게 합니다. Event Loop는 JavaScript의 Non-Blocking 특성을 가능하게 하며, 사용자 인터페이스가 멈추지 않고 부드럽게 동작할 수 있도록 합니다.

특히, 웹 브라우저나 Node.js와 같은 환경에서 이벤트 루프는 중요한 구성 요소이며, 비동기 API의 결과를 관리하고 적절한 시점에 콜백 함수를 실행시켜 줍니다.

Javascript

문제2 ▶ var let const 의 차이점을 설명해주세요.

합격 답변

변수를 선언할 때 var, let, const를 사용하는 것은 JavaScript에서 변수의 scope(유효 범위)와 재할당 가능성을 제어하기 위함입니다. 간단하게 이 세 가지의 차이점은 다음과 같습니다:

var:
var로 선언된 변수는 함수 스코프를 가집니다. 이는 해당 변수가 함수 내에서 정의되었을 때, 함수 내부 어디에서든 접근 가능하다는 것을 의미합니다.
var로 선언된 변수는 재선언 가능하며, 함수 밖에서 선언된 경우 전역 스코프를 가지게 됩니다.
호이스팅되는 특징이 있어, 선언 전에 사용할 수 있습니다(단, 값은 undefined로 처리됩니다).
let:
let으로 선언된 변수는 블록 스코프를 가집니다. 즉, 선언된 블록({}) 내부에서만 유효합니다.
let으로 선언된 변수는 재할당이 가능하지만 재선언은 불가능합니다.
호이스팅은 발생하지만 var와 달리 선언 전에 접근하면 참조 에러(ReferenceError)가 발생합니다.
const:
const도 let처럼 블록 스코프를 가집니다.
const로 선언된 변수는 반드시 선언과 동시에 초기화가 이루어져야 하며, 한 번 할당된 값은 변경할 수 없습니다(불변성).
객체나 배열을 const로 선언하는 경우, 객체나 배열 자체는 변경할 수 없지만, 그 내용은 수정할 수 있습니다.
이 세 가지 키워드를 이해하는 것은 변수의 유효 범위와 상태 관리에 있어 중요합니다. var는 이제 구식이 되었고, let과 const는 더 명확하고 예측 가능한 코드를 작성하기 위해 ES6에서 도입되었습니다. let은 변수가 변할 수 있는 경우에, const는 변하지 않을 변수에 사용합니다.

React

문제3 ▶ useMemo와 useCallback에 대해 설명해주세요.

합격 답변

React에서 useMemo와 useCallback은 성능 최적화를 위해 사용되는 훅입니다. 둘 다 의존성 배열을 기반으로 값이나 함수를 메모이제이션(캐싱)하는 데 사용되며, 불필요한 렌더링을 방지할 수 있습니다.
useMemo는 계산량이 많은 함수의 결과값을 메모이제이션하는데 사용됩니다. 이 훅은 의존성 배열에 있는 값이 변경될 때만 메모이제이션된 값을 다시 계산합니다. 만약 의존성 값이 변경되지 않았다면, 이전에 메모이제이션된 값을 재사용하여 성능을 향상시킵니다.

예를 들어:
const memoizedValue = useMemo(() => computeExpensiveValue(a, b), [a, b]);
위 코드는 computeExpensiveValue 함수가 복잡한 계산을 수행한다고 할 때, a 또는 b가 바뀌지 않는 한 이전에 계산된 값을 재사용합니다.
useCallback은 특정 함수를 메모이제이션하는 데 사용됩니다. 함수는 컴포넌트가 렌더링될 때마다 기본적으로 새로 생성되지만, useCallback을 사용하면 의존성 배열에 있는 값이 변경될 때만 함수를 다시 생성합니다. 이는 자식 컴포넌트에 props로 함수를 전달할 때 특히 유용하며, 자식 컴포넌트가 불필요하게 리렌더링되는 것을 방지할 수 있습니다.

예를 들어:
const memoizedCallback = useCallback(() => { doSomething(a, b); }, [a, b]);
위 코드에서 doSomething 함수는 a 또는 b가 바뀌었을 때만 새로 생성됩니다. 그러므로, 메모이제이션된 memoizedCallback을 자식 컴포넌트에 전달하면, 자식 컴포넌트가 prop이 변경되지 않는 한 같은 함수를 계속해서 사용할 수 있습니다.
useMemo와 useCallback은 성능 최적화를 위해 필요할 때만 사용해야 합니다. React 팀은 이 훅들을 필요 이상으로 사용하지 않을 것을 권장합니다. 불필요하게 메모이제이션을 사용하면, 메모리 사용량이 증가하고, 성능 최적화에 역효과를 낼 수 있기 때문입니다.

Javascript

문제4 ▶ 크로스 브라우징 경험이 있으신가요?

합격 답변

Null: null은 JavaScript에서 의도적으로 변수에 할당된 '아무것도 없음'을 나타내는 값입니다. 변수가 '빈 상태'임을 명시적으로 나타내고 싶을 때 사용합니다.

Undefined: undefined는 변수가 선언되었으나 아직 값을 할당받지 않았을 때 기본적으로 주어지는 값입니다. 또한, 객체의 존재하지 않는 속성에 접근하려고 할 때도 undefined가 반환됩니다.

Undeclared: 'undeclared'는 선언되지 않은 변수를 의미합니다. 즉, 해당 변수에 대해 var, let, const 등을 사용하여 선언하지 않고 값을 할당하려 할 때 JavaScript에서 참조 오류(ReferenceError)가 발생합니다.

NaN: 'NaN'은 'Not-a-Number'의 약자로, 수학적 연산이 불가능할 때 반환되는 값입니다. 예를 드어, 숫자가 아닌 문자열을 숫자로 변환하려 할 때, 또는 0으로 나누는 등의 연산에서 발생할 수 있습니다.

이들은 JavaScript에서 자주 혼동될 수 있는 개념들이며, 각각은 다른 의미와 용도를 가지고 있어 프로그래밍 시 주의를 요합니다.

Javascript

문제5 ▶ Null, undefined, undeclared, NaN 에 대해 설명해주세요.

합격 답변

Null: null은 JavaScript에서 의도적으로 변수에 할당된 '아무것도 없음'을 나타내는 값입니다. 변수가 '빈 상태'임을 명시적으로 나타내고 싶을 때 사용합니다.

Undefined: undefined는 변수가 선언되었으나 아직 값을 할당받지 않았을 때 기본적으로 주어지는 값입니다. 또한, 객체의 존재하지 않는 속성에 접근하려고 할 때도 undefined가 반환됩니다.

Undeclared: 'undeclared'는 선언되지 않은 변수를 의미합니다. 즉, 해당 변수에 대해 var, let, const 등을 사용하여 선언하지 않고 값을 할당하려 할 때 JavaScript에서 참조 오류(ReferenceError)가 발생합니다.

NaN: 'NaN'은 'Not-a-Number'의 약자로, 수학적 연산이 불가능할 때 반환되는 값입니다. 예를 드어, 숫자가 아닌 문자열을 숫자로 변환하려 할 때, 또는 0으로 나누는 등의 연산에서 발생할 수 있습니다.

이들은 JavaScript에서 자주 혼동될 수 있는 개념들이며, 각각은 다른 의미와 용도를 가지고 있어 프로그래밍 시 주의를 요합니다.

Javascript

문제6 ▶ 서버사이드렌더링(SSR)과 클라이언트사이드렌더링(CSR)의 차이점을 설명해주세요.

합격 답변

서버사이드 렌더링(SSR)과 클라이언트사이드 렌더링(CSR)은 웹 페이지의 컨텐츠를 사용자에게 보여주는 방식에 대한 두 가지 접근 방식입니다.

서버사이드 렌더링 (SSR):

SSR에서는 웹 페이지의 초기 렌더링이 서버에서 이루어집니다.
사용자가 웹페이지에 접근할 때, 서버는 HTML, CSS, 그리고 초기 상태의 JavaScript를 포함하는 완전한 페이지를 생성하여 사용자의 브라우저로 보냅니다.
이 방식은 사용자가 페이지를 빠르게 볼 수 있게 하지만, 서버에 부담을 줄 수 있으며, 페이지의 상호작용이 가능해지기까지의 시간이 늘어날 수 있습니다.
검색엔진 최적화(SEO)에 유리하며, 초기 로딩 시 사용자에게 콘텐츠를 빠르게 제공할 수 있는 이점이 있습니다.

클라이언트사이드 렌더링 (CSR):

CSR에서는 초기 렌더링이 사용자의 브라우저에서 이루어집니다.
서버는 HTML과 초기 로딩을 위한 JavaScript 파일만을 보내고, JavaScript가 브라우저에서 실행되어 페이지를 동적으로 생성합니다.
이 방식은 서버의 부하를 줄일 수 있으며, 사용자가 한 번의 초기 로딩 후에는 매끄러운 페이지 전환 경험을 할 수 있도록 합니다.
하지만 초기 로딩 시간이 길어질 수 있고, 검색엔진 최적화(SEO) 측면에서 불리할 수 있습니다.
두 방식은 각각의 장단점을 가지고 있으며, 웹 애플리케이션의 목적과 요구사항에 따라 적절한 방식을 선택할 수 있습니다. 최근에는 이 두 방식의 장점을 결합한 하이브리드 렌더링이나 서버사이드 렌더링을 보완하기 위한 방식들이 사용되고 있습니다.

Javascript

문제7 ▶ this 용법을 아는대로 설명해주세요.

합격 답변

JavaScript에서 this 키워드는 함수나 메소드의 현재 실행 컨텍스트를 참조합니다. this의 값은 함수가 호출되는 방식에 따라 달라집니다.
글로벌 컨텍스트:
- 브라우저에서: this는 글로벌 객체인 window를 참조합니다.
- Node.js에서: this는 모듈의 글로벌 객체를 참조합니다.
함수 컨텍스트:
- 일반 함수에서 this는 기본적으로 글로벌 객체를 참조합니다. 하지만 strict mode에서는 undefined가 됩니다.
- 메소드 내에서 this는 메소드를 호출한 객체를 참조합니다.
생성자 함수에서:
- 생성자 함수 내부에서 this는 새롭게 생성되는 인스턴스를 참조합니다.
클래스 컨텍스트에서:
- 클래스의 메소드 내에서 this는 메소드를 호출한 인스턴스를 참조합니다.
화살표 함수에서:
- 화살표 함수는 자신의 this를 가지지 않습니다. 대신, 화살표 함수는 자신을 둘러싼 외부 함수의 this 값을 상속받습니다.
이벤트 핸들러에서:
- DOM 이벤트 핸들러에서 this는 이벤트를 받는 DOM 요소를 참조합니다.
명시적 바인딩:
- call, apply, bind 함수를 사용하면 함수의 this 값을 명시적으로 설정할 수 있습니다.
this는 JavaScript에서 매우 강력하면서도 혼란스러운 부분 중 하나입니다. 그래서 this의 값은 함수를 어떻게 호출하느냐에 따라 다양하게 결정될 수 있으며, 이는 때로는 예상치 못한 버그의 원인이 되기도 합니다. 따라서 this를 사용할 때는 그 컨텍스트가 무엇인지 항상 주의해서 확인해야 합니다.

CS

문제8 ▶ HTTP와 HTTPS 통신 방식의 차이를 설명해주세요.

합격 답변

우선 HTTP는 HyperText Transfer Protocol의 약자로, 인터넷에서 데이터를 주고받는 프로토콜입니다. 웹 서버와 클라이언트, 즉 브라우저 간에 통신할 때 사용하는 기본적인 규칙이죠. 하지만 HTTP의 큰 단점은 정보가 평문 형태로 전송된다는 것입니다. 이는 누군가 네트워크를 감시하고 있다면, 전송되는 정보를 쉽게 읽을 수 있다는 의미입니다.

이를 보완하기 위해 나온 것이 HTTPS입니다. 'S'는 Secure, 즉 보안을 의미하죠. HTTPS는 HTTP에 보안 계층을 추가한 것으로, SSL 또는 TLS라는 기술을 통해 데이터를 암호화하여 전송합니다. 이 암호화 과정은 데이터가 제 3자에 의해 도청되거나 변경되는 것을 방지해줍니다. 그래서 개인 정보나 민감한 거래를 다루는 웹사이트에서 주로 HTTPS를 사용하게 됩니다.

HTTP는 기본적으로 80번 포트를 사용하는 반면, HTTPS는 443번 포트를 사용합니다. 이는 웹 서버가 각 프로토콜을 구분하고 적절히 처리할 수 있게 해줍니다.

요약하자면, HTTP는 빠르고 간단하지만 보안에 취약하고, HTTPS는 약간의 처리 시간이 더 걸리지만 훨씬 안전합니다. 그래서 요즘 웹 환경에서는 대부분의 사이트가 사용자의 보안을 위해 HTTPS를 채택하고 있습니다.

Javascript

문제9 ▶ 클로저는 무엇인가요? 원리와 왜 사용하는지 설명해주세요.

합격 답변

클로저는 함수가 자신이 생성될 때의 환경을 기억하는 JavaScript의 한 기능입니다. 함수가 자신의 외부 변수에 접근할 수 있게 해주는데, 이는 함수가 실행되는 순간이 아닌, 함수가 선언되는 순간의 스코프에 영향을 받습니다.

클로저의 원리는 실행 컨텍스트와 렉시컬 스코프의 개념에 근거합니다. 자바스크립트에서 모든 함수는 각자의 실행 컨텍스트를 가지며, 이 컨텍스트 내에는 변수 객체, 스코프 체인, this 등의 정보가 포함됩니다. 함수가 실행될 때, 자바스크립트 엔진은 현재 함수의 스코프와 상위 함수의 스코프를 체인으로 연결하여 스코프 체인을 형성하고, 이를 통해 현재 함수에서 상위 함수의 변수에 접근할 수 있게 됩니다. 클로저는 이 스코프 체인을 통해 외부 함수의 변수를 기억하고 접근합니다.

클로저를 사용하는 이유는 여러 가지가 있습니다. 첫째, 데이터 캡슐화를 위해 사용됩니다. 클로저를 통해 공개 API를 제공하면서도 내부 변수를 외부로부터 숨길 수 있기 때문에, 모듈 패턴 같은 디자인 패턴을 구현할 때 유용합니다. 둘째, 클로저는 상태를 유지할 수 있습니다. 예를 들어, 이벤트 핸들러나 콜백 함수에서 특정 데이터를 참조할 필요가 있을 때 클로저를 이용하면 그 상태를 유지하면서도 동작을 정의할 수 있습니다.

이러한 클로저의 특성은 강력하지만, 메모리 누수의 가능성을 열어두기 때문에 주의해서 사용해야 합니다. 스코프 내의 변수가 클로저에 의해 참조되는 한, 가비지 컬렉터는 그 메모리를 회수할 수 없습니다. 그래서 불필요한 클로저가 계속 유지되지 않도록 주의해야 합니다.

개발경험

문제10 ▸ 고객이 제작한 페이지 화면 로딩 속도가 느리다고 피드백이 들어왔습니다. 어떻게 대처할 것인지 설명해주세요.

합격 답변

고객으로부터 페이지 로딩 속도가 느리다는 피드백을 받았을 때, 저는 먼저 문제의 원인을 체계적으로 파악하는 데 집중할 것입니다. 웹 성능 점검 도구를 사용하여 페이지의 로딩 프로세스를 분석하고, 크롬 개발자 도구의 네트워크 탭을 통해 어떤 자원이 로딩에 시간을 많이 소요하는지 확인할 것입니다.

제 경험에 따르면, 일반적인 원인으로는 불필요한 대용량 이미지, 미최적화된 스크립트 실행, 또는 서버의 응답 시간 지연 등이 있습니다. 이러한 문제들을 해결하기 위해 이미지는 적절한 형식(예: WebP)으로 변환하고 압축을 적용할 수 있으며, 자바스크립트는 비동기 로딩으로 변경하거나 필요한 부분만 로드하도록 코드 분할을 고려할 수 있습니다. 또한, 사용자가 전 세계 어디에 있더라도 빠른 접근을 보장하기 위해 콘텐츠 전송 네트워크(CDN)를 사용하는 방안도 고려할 것입니다.

이러한 기술적 조치 외에도, 서버 설정의 개선을 통해 캐싱 전략을 재검토하고, 필요하다면 서버의 성능을 향상시키는 것도 중요합니다. 문제를 해결한 이후에는 Google PageSpeed Insights와 같은 도구를 통해 성능이 실제로 개선되었는지 지속적으로 모니터링하여, 향후 비슷한 문제가 발생하지 않도록 예방하려고 합니다.

CS

문제11 ▸ SEO를 최적화하기 위해 어떤 작업을 해봤는지 공유해주세요.

합격 답변

SEO 최적화를 위한 작업은 웹사이트의 가시성을 높이고 검색 엔진의 결과 페이지에서 더 높은 순위를 차지하기 위해 필수적입니다. 여기에 몇 가지 접근 방식과 구체적인 작업들을 소개하겠습니다:

1. 메타 태그 최적화: 각 페이지의 `title`, `description`, 그리고 `keyword` 메타 태그를 검색 엔진이 콘텐츠를 정확히 이해할 수 있도록 명확하고, 관련성 높은 키워드로 구성했습니다. 이는 검색 엔진 결과에서 사용자의 클릭을 유도하는데 중요한 역할을 합니다.
2. 콘텐츠 최적화: 품질 높은 원본 콘텐츠를 제작하고 키워드 밀도를 최적화하여 검색 결과에서 더 나은 순위를 얻을 수 있도록 했습니다. 또한, 정기적으로 콘텐츠를 업데이트하고 확장하여 웹사이트의 신선도를 유지했습니다.
3. 사이트 구조와 URL 최적화: 명확하고 직관적인 URL 구조를 사용하여 사용자와 검색 엔진 모두에게 사이트의 내비게이션을 용이하게 만들었습니다. 이를 통해 사이트의 '크롤링'이 용이해지고, 페이지 간의 연관성을 높일 수 있습니다.
4. 모바일 최적화: 반응형 웹 디자인을 적용하여 모든 기기에서 사용자 경험이 일관되게 유지되도록 했습니다. 구글과 다른 검색 엔진들은 모바일 친화성을 랭킹 요소로 사용하기 때문에 이는 중요한 최적화 작업입니다.
5. 속도 최적화: 페이지 로딩 시간을 줄이기 위해 이미지 최적화, 캐싱 정책 설정, CDN 사용 등을 통해 사이트의 성능을 향상시켰습니다.
6. 백링크 전략: 고품질의 백링크를 확보하여 도메인 권위를 높이고 검색 순위에 긍정적인 영향을 미치도록 노력했습니다. 이를 위해 게스트 블로깅, 파트너십, 인플루언서 아웃리치 등의 방법을 사용했습니다.
7. 애널리틱스 및 모니터링: Google Analytics와 Google Search Console을 활용하여 트래픽과 사용자 행동을 분석하고, SEO 전략을 미세 조정했습니다. 또한, 주기적으로 키워드 순위를 확인하고 경쟁 분석을 수행하여 전략을 지속적으로 개선했습니다.

이러한 작업을 통해 저희 웹사이트는 검색 엔진의 가시성이 크게 향상되었고, 유입 트래픽 또한 지속적으로 증가하고 있습니다. SEO는 지속적인 과정이기 때문에 앞으로도 이러한 노력을 지속할 계획입니다.

Javascript

문제12 ▸ CORS를 대처하는 방법과 우회하는 방법을 설명해주세요.

합격 답변

CORS, 즉 Cross-Origin Resource Sharing은 웹 보안의 중요한 부분으로, 웹 페이지가 다른 도메인의 리소스에 액세스할 수 있도록 브라우저가 사용하는 메커니즘입니다. 이 메커니즘은 다른 출처(도메인, 프로토콜, 포트)의 스크립트가 발생시킨 요청에 대한 응답을 서버가 허용해야 할 때 어떤 조건들을 충족해야 하는지를 정의합니다.
CORS 문제를 대처하고 우회하는 방법은 여러 가지가 있습니다:
대처 방법:
서버 설정 변경:
가장 흔하고 권장되는 방법은 서버에서 Access-Control-Allow-Origin 헤더를 설정하여 요청을 보낸 출처를 허용하는 것입니다. 서버 설정에서 특정 도메인 또는 모든 도메인(*)에 대해 이 헤더를 추가할 수 있습니다.
프록시 서버:
서버 사이드에 프록시를 설정하여 요청을 중계함으로써 CORS 정책을 우회할 수 있습니다. 이 방식은 실제 요청을 프록시 서버로 보내고, 프록시 서버가 타겟 리소스에 대한 요청을 대신 보낸 후 결과를 반환합니다.
서버 사이드 CORS 처리: 서버 언어(예: Node.js, Python, Java 등)에서 요청을 받았을 때 CORS 관련 헤더를 추가하여 응답할 수 있도록 처리하는 것입니다.
우회 방법:
JSONP: JSON with Padding은 스크립트 태그를 통해 다른 도메인에 있는 서버로부터 데이터를 가져올 때 사용되는 기술입니다. 하지만 이 방법은 오직 GET 요청에만 사용할 수 있고, 보안상의 위험이 있어 현대의 웹 개발에서는 권장되지 않습니다.
브라우저 확장 기능 사용: 특정 브라우저 확장 기능을 사용하여 CORS 정책을 우회할 수 있습니다. 이러한 확장 기능은 개발 단계에서 유용하지만, 보안 문제로 인해 일반 사용자에게는 권장되지 않습니다. 브라우저 설정 변경: 브라우저의 보안 기능을 비활성화하면 CORS 문제를 우회할 수 있지만, 이는 오직 개발 목적으로만 사용해야 하며, 일반적으로 사용자에게는 적용될 수 없는 방법입니다.CORS 정책에 대처하고 우회하는 방법을 선택할 때는, 보안과 웹 애플리케이션의 무결성을 해치지 않는 선에서 적절한 방법을 선택해야 합니다. 개발 중에는 이러한 제한을 우회하는 방법들이 유용할 수 있지만, 실제 프로덕션 환경에서는 정당한 서버 설정을 통해 문제를 해결하는 것이 가장 좋습니다.

HTML/CSS

문제13 ▸ CSS 박스 모델에 대해 설명해주세요.

합격 답변

CSS 박스 모델은 웹페이지를 렌더링할 때, HTML 요소들이 어떻게 레이아웃되고 크기가 결정되는지를 정의하는 CSS의 기본적인 디자인 개념입니다. 각 HTML 요소는 마치 상자처럼 취급되며, 여러 개의 다른 상자들이 쌓이거나 나란히 배열되어 웹페이지를 구성합니다. 이 모델은 각 요소의 마진(margin), 테두리(border), 패딩(padding), 그리고 실제 내용(content)의 네 부분으로 구성됩니다.

내용(Content): 실제 텍스트, 이미지 또는 다른 미디어가 표시되는 영역입니다.
패딩(Padding): 내용과 테두리 사이의 공간으로, 내용 영역을 테두리 안쪽으로부터 밀어냅니다.
테두리(Border): 패딩을 둘러싸고 있는 실제 선으로, 요소의 외곽선을 정의합니다.
마진(Margin): 요소와 주변 요소 사이의 외부 공간입니다. 이 공간은 투명하며, 다른 요소와의 거리를 조정하는 데 사용됩니다.
모델의 각 부분은 CSS를 사용하여 스타일링할 수 있으며, 각 요소의 width와 height 속성은 기본적으로 내용 영역만을 말합니다. 하지만, box-sizing 속성을 border-box로 설정하면, 테두리와 패딩을 포함한 크기를 조절할 수 있습니다.

Javascript

문제14 ▸ use strict 개념과 이것을 사용하는것의 장점과 단점을 설명해주세요.

합격 답변

use strict는 ECMAScript 5 (ES5)에 도입된 지시어로, 스크립트나 함수가 '엄격 모드(strict mode)'로 실행되도록 합니다. 이 모드에서는, 예를 들어, 암묵적 전역 변수 생성, 삭제 불가능한 프로퍼티 삭제 시도, 중복된 매개변수 이름, 안전하지 않은 동작(예: with 문 사용) 등 일반적인 JavaScript 코드에서 허용되는 몇몇 행위들이 금지됩니다.

장점:
버그 조기 발견: 엄격 모드는 일반적으로 무시되거나 조용히 실패하는 많은 오류들을 실제 오류로 전환합니다. 이는 개발자가 버그를 더 빨리 발견하고 수정할 수 있게 도와줍니다.
보안 개선: 안전하지 않은 동작의 금지로 인해 보안에 더 나은 보장을 제공합니다.
최적화 가능성: 엄격 모드에서 실행되는 코드는 일부 엔진에서 더 최적화되어 더 빠르게 실행될 가능성이 있습니다.
더 나은 에러 체킹: 더 엄격한 에러 체킹으로 인해, 잠재적으로 문제를 일으킬 수 있는 코드를 방지할 수 있습니다.
미래 지향적: ECMAScript의 미래 버전과의 호환성을 위해, 오늘날의 좋은 습관을 채택하도록 도와줍니다.
단점:
호환성 문제: 엄격 모드가 적용되지 않은 기존의 라이브러리나 스크립트와의 호환성 문제가 발생할 수 있습니다.
학습 곡선: 새로운 개발자가 엄격 모드의 규칙을 이해하고 적응하는 데 시간이 필요할 수 있습니다.
개발자의 오류: 일부 개발자는 엄격 모드에서 발생하는 오류에 대처하기 위해 코드를 잘못 수정할 수도 있습니다.
엄격 모드를 사용하기 위해서는 스크립트나 함수의 맨 위에 use strict; 선언을 추가합니다. 이렇게 하면 해당 스크립트 또는 함수는 엄격 모드로 실행되며, 이로 인해 발생하는 어떠한 오류도 콘솔에 출력됩니다. 이 지시어는 JavaScript를 좀 더 안정적이고 예측 가능한 언어로 만들기 위한 한 단계로 볼 수 있습니다.

CS

문제15 ▸ 객체지향 프로그래밍이 무엇인지 설명해주세요.

합격 답변

객체지향 프로그래밍(Object-Oriented Programming, OOP)은 컴퓨터 프로그래밍의 패러다임 중 하나로, '객체'라는 기본 단위로 프로그램의 요소들을 구성하는 방법입니다. 객체는 데이터와 이 데이터를 처리하는 함수(객체의 '메서드'라고 함)를 포함하는 '클래스'라는 설계도로부터 생성됩니다. 객체지향 프로그래밍의 주요 목표는 코드의 재사용성, 확장성, 관리 용이성을 높이는 것입니다.

객체지향 프로그래밍의 주요 개념에는 다음 네 가지가 있습니다:

1. 캡슐화(Encapsulation): 객체는 데이터와 함수를 하나의 단위로 묶고, 객체의 세부 사항을 외부로부터 숨기며(은닉), 외부 코드가 객체 내의 데이터를 직접 조작하는 것을 방지합니다.

2. 상속(Inheritance): 한 클래스가 다른 클래스의 속성과 메서드를 상속받을 수 있습니다. 이를 통해 기존 코드의 재사용성을 높이고, 관계를 통해 클래스 계층을 구성할 수 있습니다.

3. 다형성(Polymorphism): 같은 이름의 메서드가 다른 행동을 할 수 있도록 하여, 다양한 타입의 객체를 일관된 방식으로 처리할 수 있습니다.

4. 추상화(Abstraction): 복잡한 실제 상황을 단순화된 모델로 표현합니다. 클래스를 통해 고수준의 추상화를 제공하고, 구체적인 구현을 추상 클래스 또는 인터페이스 뒤에 숨길 수 있습니다.

객체지향 프로그래밍은 소프트웨어를 유연하고, 변경에 쉽게 대응하며, 대규모 프로젝트와 복잡한 시스템을 관리하기 위한 강력한 도구로 광범위하게 사용됩니다.

Javascript

문제16 ▶ '=='와 '===' 차이를 설명해주세요.

합격 답변

`==`는 동등 연산자로, 두 값이 같은지 비교할 때 사용합니다. 하지만, `==`는 타입 변환을 수행하여, 다른 타입의 두 값을 비교할 때 자동으로 한 쪽을 다른 쪽 타입으로 변환한 후 값을 비교합니다. 예를 들어, `0 == '0'`은 `true`를 반환합니다, 왜냐하면 문자열 '0'이 숫자 0으로 변환되기 때문입니다.

반면, `===`는 일치 연산자로, 값과 타입이 모두 같은지를 엄격하게 비교합니다. 이 연산자는 타입 변환을 수행하지 않으므로, `3 === '3'`은 `false`를 반환합니다, 왜냐하면 한 쪽은 숫자이고 다른 쪽은 문자열이기 때문입니다. `===`를 사용하는 것은 타입에 대한 예측 가능성을 높이고, 불필요한 오류를 방지하기 위해 권장됩니다.

React

문제17 ▸ Context API에 대해 설명해주세요.

합격 답변

React의 Context API는 컴포넌트 트리 전반에 걸쳐 데이터를 전달하는 방법을 제공합니다. 이는 특정 범위의 컴포넌트에 걸쳐 전역적인 데이터(예를 들어, 현재 로그인한 사용자, 테마 설정 등)를 공유할 때 유용합니다. Context API를 사용하면, 컴포넌트를 통해 명시적으로 props를 전달하지 않고도, 컴포넌트가 필요한 데이터에 접근할 수 있습니다.
Context를 사용하는 일반적인 단계는 다음과 같습니다:

Context 생성: React.createContext() 함수를 호출하여 Context를 생성합니다. 이 함수는 Provider와 Consumer 컴포넌트를 포함하는 객체를 반환합니다.

```
const MyContext = React.createContext(defaultValue);
```

Context Provider: Provider 컴포넌트는 Context를 구독하는 컴포넌트들에게 Context의 변화를 알립니다. Provider는 value 속성을 통해 전달되는 값을 통해 자신의 하위 컴포넌트에게 Context 데이터를 제공합니다.

```
<MyContext.Provider value={/* 어떤 값 */}>
{/* 하위 컴포넌트들 */}
</MyContext.Provider>
```

Context 소비: 하위 컴포넌트에서는 Consumer 컴포넌트를 사용하거나 useContext 훅을 통해 Context 데이터를 소비할 수 있습니다.

```
// 클래스 컴포넌트에서
<MyContext.Consumer>
{value ⇒ /* 컨텍스트 값을 이용한 렌더링 */}
</MyContext.Consumer>
// 함수형 컴포넌트에서
const value = useContext(MyContext);
```

Context API를 사용함으로써, 컴포넌트의 재사용성을 높이고, prop drilling(여러 계층에 걸쳐 props를 전달하는 것) 문제를 해결할 수 있습니다. 그러나, Context는 주의해서 사용해야 합니다. Context의 값이 변경되면, 이를 구독하는 모든 컴포넌트가 리렌더링되기 때문에 성능 문제가 발생할 수 있습니다. 따라서 상태 관리가 필요한 경우에만 사용해야 하며, 전역 상태를 관리하는 데 좀 더 집중된 상태 관리 라이브러리(예: Redux)와 함께 사용하는 것이 좋을 수 있습니다.

Javascript

문제18 ▸ 클래스형 컴포넌트와 함수형 컴포넌트의 차이를 설명해주세요.

합격 답변

클래스형 컴포넌트와 함수형 컴포넌트는 React에서 UI를 구성하는 두 가지 방식입니다. 이 두 방식의 주요 차이점은 다음과 같습니다:

문법과 선언 방식:

클래스형 컴포넌트는 ES6의 클래스를 사용해 컴포넌트를 정의하며, this 키워드를 사용하여 상태(state)와 생명주기(lifecycle) 메소드에 접근합니다.
함수형 컴포넌트는 단순한 JavaScript 함수를 사용하여 컴포넌트를 정의하며, Hooks를 사용하여 상태와 다른 React의 기능을 사용합니다.
생명주기 메소드(Lifecycle methods):

클래스형 컴포넌트는 componentDidMount, componentDidUpdate, componentWillUnmount와 같은 생명주기 메소드를 제공합니다.
함수형 컴포넌트는 기존의 생명주기 메소드에 대응하는 useEffect Hook을 사용하여 부수 효과(side effects)를 처리합니다.
상태(State) 관리:
클래스형 컴포넌트는 this.state와 this.setState를 통해 내부 상태를 관리합니다.
함수형 컴포넌트는 useState Hook을 통해 상태를 관리합니다.
this 키워드:
클래스형 컴포넌트에서는 this를 사용해 현재 컴포넌트 인스턴스에 접근할 수 있습니다.
함수형 컴포넌트에서는 this가 필요 없으며, 모든 것이 함수의 인자 또는 Hook을 통해 접근 가능합니다.
React 팀과 커뮤니티는 함수형 컴포넌트와 Hooks 사용을 권장하고 있으며, 함수형 컴포넌트는 선언적이고 간결해 코드의 가독성과 유지 관리성을 향상시키는 장점이 있습니다. 그러나 일부 복잡한 상황에서는 클래스형 컴포넌트가 여전히 유용할 수 있습니다.

Vue.js

문제19 ▸ Vue.js에서 사용하는 양방향 데이터 바인딩의 원리를 설명해주세요.

합격 답변

Vue.js에서 양방향 데이터 바인딩은 v-model 디렉티브를 사용하여 구현됩니다. 이것은 주로 폼 입력과 애플리케이션 데이터를 양방향으로 연결하는 데 사용됩니다. v-model은 내부적으로 데이터를 업데이트하는 이벤트 리스너와 데이터를 표시하는 방법을 결합하여 작동합니다.

양방향 데이터 바인딩의 기본 원리는 다음과 같습니다:

데이터를 입력 필드에 바인딩하기: Vue 인스턴스의 데이터 속성을 입력 필드에 연결하면, 해당 필드의 값이 데이터 속성과 동기화됩니다. 사용자가 입력 필드의 값을 변경하면, 바인딩된 Vue 데이터도 자동으로 업데이트됩니다.
Vue 인스턴스 데이터 변경 반영하기: 반대로, Vue 인스턴스의 데이터 속성이 변경되면, 이 변경 사항이 연결된 입력 필드에 자동으로 반영됩니다.

이 과정은 두 가지 주요한 기능으로 구성됩니다:

DOM에서 데이터로의 흐름: 사용자가 입력 필드에 값을 입력하면, input 이벤트가 발생합니다. v-model은 이 이벤트를 감지하고, Vue 인스턴스의 데이터를 업데이트합니다.
데이터에서 DOM으로의 흐름: Vue 인스턴스의 데이터가 변경되면, Vue는 반응형 시스템을 사용하여 변경 사항을 감지하고, 관련된 DOM을 업데이트합니다. 이것은 DOM을 항상 최신 상태로 유지합니다.

이러한 메커니즘을 통해, Vue는 사용자 인터페이스와 데이터 사이의 일관된 상태를 유지할 수 있게 해주며, 복잡한 데이터 동기화 코드를 작성할 필요 없이, 자연스럽게 데이터의 양방향 흐름을 관리할 수 있습니다. Vue의 양방향 데이터 바인딩은 효율적인 사용자 경험을 제공하며, 폼 기반의 인터랙션을 더욱 간단하게 만들어 줍니다.

React

문제20 ▸ React 라이프 사이클을 간단하게 설명해주세요.

합격 답변

React의 라이프사이클은 주로 생성(Creation), 업데이트(Updating), 제거(Unmounting)의 세 단계로 나눌 수 있습니다.

1. 생성(Creation):
* constructor(): 컴포넌트가 생성될 때 호출되어 초기 state를 설정할 수 있습니다.
* getDerivedStateFromProps(props, state): 컴포넌트가 마운트되기 전과 컴포넌트가 새 props를 받을 때 호출됩니다.
* render(): 컴포넌트의 필수 메소드로, React 요소를 반환하여 DOM에 렌더링합니다.
* componentDidMount(): 컴포넌트가 처음 DOM에 마운트된 직후에 호출되며, 여기서 API 호출과 같은 사이드 이펙트를 일으킬 수 있습니다.
2. 업데이트(Updating):
* getDerivedStateFromProps(props, state): 컴포넌트가 새로운 props를 받았을 때 호출됩니다.
* shouldComponentUpdate(nextProps, nextState): 컴포넌트가 업데이트되어야 할지 여부를 결정합니다. 성능 최적화에 사용됩니다.
* render(): 컴포넌트가 업데이트를 반영하기 위해 다시 렌더링합니다.
* getSnapshotBeforeUpdate(prevProps, prevState): 실제 DOM 변경 사항이 적용되기 직전에 호출됩니다.
* componentDidUpdate(prevProps, prevState, snapshot): 업데이트가 DOM에 반영된 후에 호출됩니다.
3. 제거(Unmounting):
* componentWillUnmount(): 컴포넌트가 DOM에서 제거되기 전에 호출되어, 필요한 정리(clean-up) 작업을 수행합니다. 예를 들어, 타이머를 제거하거나 네트워크 요청을 취소할 수 있습니다.

이 라이프사이클 메소드들은 컴포넌트가 어떻게 생성, 업데이트, 제거되는지를 관리하며, 각 단계에서 개발자가 특정 동작을 실행할 수 있는 훅(hook)을 제공합니다.

Javascript

문제21 ▸ javascript 성능 최적화를 위해 시도한 경험을 공유해주세요.

합격 답변

제가 일하던 한 프로젝트에서 웹 애플리케이션의 로딩 시간이 상당히 느린 문제가 있었습니다. 이를 해결하기 위해 여러 성능 최적화 전략을 시도했습니다.
첫 번째로, 크롬 개발자 도구를 활용하여 성능 프로파일링을 수행했습니다. 이를 통해 메모리 누수가 있음을 확인하고, 이벤트 리스너가 제대로 해제되지 않아 발생하는 메모리 누수를 해결했습니다.
두 번째로, 웹팩(bundle)을 사용하여 코드 스플리팅을 적용했습니다. 사용자가 필요로 하는 자원만을 로딩할 수 있도록 하여 초기 로딩 시간을 대폭 줄였습니다.
세 번째로, 렌더링 성능을 향상시키기 위해 Virtual DOM을 활용하는 React로 프론트엔드를 재작성했습니다. 이를 통해 불필요한 DOM 조작을 줄이고, 결과적으로 사용자 인터페이스의 반응성을 개선했습니다.
마지막으로, 이미지와 같은 정적 자원들에 대해서는 지연 로딩(lazy loading)을 구현하여, 사용자에게 필요한 내용만 우선적으로 로딩하도록 했습니다.
이러한 최적화 작업을 통해 애플리케이션의 성능이 크게 향상되었고, 페이지 로딩 시간은 50% 이상 단축되었습니다. 사용자 경험 또한 상당히 개선되었으며, 이는 구글 애널리틱스의 사용자 체류 시간 증가와 같은 지표로도 명확히 확인할 수 있었습니다.
이 경험을 통해 저는 성능 최적화가 단순히 코드를 빠르게 하는 것을 넘어서, 사용자 경험을 직접적으로 개선하고 비즈니스 목표에 기여할 수 있음을 깨닫게 되었습니다.

Typescript

문제22 ▶ 타입스크립트의 특징과 사용 경험을 공유해주세요.

합격 답변

타입스크립트는 자바스크립트에 타입 시스템을 추가하여 개발한 언어입니다. 이는 코드의 안정성을 높이고, 대규모 애플리케이션을 개발할 때 발생할 수 있는 오류를 사전에 방지하는 데 큰 도움이 됩니다. 타입스크립트는 자바스크립트의 상위 집합이기 때문에 모든 자바스크립트 코드가 타입스크립트에서도 그대로 동작합니다. 또한, IDE의 지원을 받아 더 나은 자동 완성과 인터페이스를 제공하여 개발 생산성을 향상시킵니다.

개인적인 경험으로는, 타입스크립트를 사용함으로써 코드의 가독성이 향상되고, 팀원 간의 협업 시 코드의 의도를 명확히 전달할 수 있었습니다. 컴파일 시점에 타입 체크가 이루어져 런타임 오류의 가능성을 줄였고, 복잡한 객체나 함수의 인터페이스를 정의함으로써 프로젝트의 구조를 명확히 할 수 있었던 것이 큰 장점이었습니다.

CS

문제23 ▶ 스크립트 언어란 무엇인지 설명해주세요.

합격 답변

스크립트 언어는 프로그래밍 언어의 한 유형으로, 복잡한 컴파일 과정 없이 바로 실행할 수 있으며, 주로 자동화된 작업 수행, 웹 페이지의 동적인 동작 구현, 빠른 프로토타이핑 및 개발 과정을 용이하게 하기 위해 사용됩니다. 이들은 대체로 인터프리터에 의해 실행되며, 높은 수준의 추상화를 제공하여 개발자가 생산성을 향상시킬 수 있게 해줍니다. 예를 들어, JavaScript, Python, Ruby와 같은 언어들이 이 범주에 속합니다.

CS

문제24 ▶ RESTful API에 대해 설명해주세요.

합격 답변

RESTful이란, 'Representational State Transfer'의 약어로, 웹에서 사용되는 아키텍처 스타일 중 하나입니다. 이 아키텍처를 따르는 웹 서비스를 RESTful 서비스라고 합니다. RESTful 아키텍처의 핵심 개념은 다음과 같습니다:

자원(Resource)의 표현(Representation): 웹에서 모든 것은 자원으로 표현되며, 각 자원은 고유한 URI(Uniform Resource Identifier)를 가집니다. 자원의 상태(데이터)는 JSON, XML 등으로 표현될 수 있습니다.

무상태(Stateless) 통신: 각 요청은 독립적이며, 요청 사이에 클라이언트 정보가 저장되지 않습니다. 즉, 이전 요청의 정보를 기억하지 않습니다. 서버는 각 요청을 완전히 독립적으로 처리합니다.

메소드(Methods): HTTP 표준 메소드를 사용하여 자원을 처리합니다. 주로 GET (자원 조회), POST (자원 생성), PUT (자원 갱신), DELETE (자원 삭제) 메소드가 사용됩니다.

일관성 있는 인터페이스(Uniform Interface): 일관된 인터페이스를 통해 정보가 상호 작용하므로, REST API를 사용하기 위한 복잡성이 줄어들고, 플랫폼과 언어에 독립적으로 작동할 수 있습니다.

시스템 분리(Separation of Concerns): RESTful 아키텍처는 클라이언트와 서버의 관심사를 분리합니다. 이로 인해 클라이언트는 사용자 인터페이스에, 서버는 데이터 스토리지와 보안에 집중할 수 있습니다.

캐시 가능(Cacheable): RESTful 서비스는 캐시를 사용하여 응답을 재사용할 수 있도록 하여 네트워크 효율을 높이고 성능을 개선할 수 있습니다.
이러한 원칙들을 통해, RESTful은 분산 시스템에서 확장성이 높고, 유연하며, 여러 플랫폼 간에 호환 가능한 서비스를 구축할 수 있도록 해줍니다.

Javascript

문제25 ▸ Async, Await과 Promise의 차이를 설명해주세요.

합격 답변

Async/Await는 Promise를 더 간결하고 동기적인 코드 흐름으로 작성할 수 있게 하는 ES2017(ES8)에 도입된 문법입니다.
async 함수는 항상 Promise를 반환하며, 함수 내부에서 await 키워드를 사용하여 Promise의 결과를 기다립니다.
await은 비동기 작업이 끝날 때까지 함수 실행을 일시 중단시키고, 결과값이 나오면 다시 실행을 계속합니다.

Promise는 .catch() 메소드를 사용하여 에러를 처리하는 반면, Async/Await은 전통적인 try/catch 문을 사용하여 에러를 핸들링할 수 있습니다. Async/Await을 사용하면 중첩된 .then() 호출이 필요한 복잡한 비동기 코드를 피할 수 있어, 가독성이 좋은 코드를 작성할 수 있습니다. Async/Await은 일반적인 동기 코드의 흐름을 가지고 있어 디버깅이 더 수월할 수 있습니다. Promise 체인에서 발생한 에러는 때때로 추적하기 어려울 수 있습니다.

Javascript

문제26 ▸ 이벤트 버블링에 대해 설명해주세요.

합격 답변

이벤트 버블링(Event Bubbling)은 웹 브라우저의 이벤트 전파 방식 중 하나입니다. DOM 요소에서 이벤트가 발생했을 때, 그 이벤트가 발생한 요소로부터 시작하여 상위 요소로 전파되는 특성을 말합니다.

예를 들어, HTML에서는 한 요소가 다른 요소 안에 중첩되어 있을 수 있습니다. 만약 하위 요소에서 클릭 이벤트가 발생하면, 그 이벤트는 먼저 해당 요소에 할당된 이벤트 핸들러를 실행한 후, 부모 요소의 이벤트 핸들러를 차례로 실행하며, 가장 상위의 요소까지 전파됩니다.

이러한 이벤트 전파 메커니즘은 개발자가 여러 요소에 대해 이벤트를 처리할 때 유용하게 활용됩니다. 예를 들어, 여러 버튼이 하나의 부모 요소 안에 있을 때, 각각의 버튼에 이벤트 리스너를 붙이는 대신, 부모 요소에 하나의 이벤트 리스너를 붙여서 모든 버튼의 클릭 이벤트를 관리할 수 있습니다. 이를 이벤트 위임(Event Delegation)이라고 합니다.

이벤트 버블링은 자동으로 발생하지만, 때로는 버블링을 중지시켜야 할 필요가 있습니다. 이를 위해 JavaScript에서는 event.stopPropagation() 메소드를 제공합니다. 이 메소드는 현재 이벤트가 더 이상 부모 요소로 전파되지 않도록 합니다. 하지만 이 메소드의 사용은 주의를 요하며, 이벤트 버블링이 필요한 다른 이벤트 핸들러에 영향을 줄 수 있기 때문에 필요한 경우에만 제한적으로 사용해야 합니다.

이벤트 버블링의 이해는 돔 이벤트를 다루는데 있어서 필수적이며, 복잡한 인터페이스를 효과적으로 관리하고 이벤트를 효율적으로 처리할 수 있는 기반을 마련합니다.

React

문제27 ▸ Virtual DOM 작동 원리를 설명해주세요.

합격 답변

"Virtual DOM은 현대 프론트엔드 프레임워크에서 중요한 개념 중 하나로, 특히 React와 같은 라이브러리에서 핵심적인 역할을 합니다. Virtual DOM의 주된 목적은 브라우저의 실제 DOM을 직접 조작하는 것보다 효율적으로 UI를 업데이트 하는 것입니다. Virtual DOM은 기본적으로 실제 DOM의 가벼운 복사본입니다. 실제 DOM을 직접 조작하는 것은 비용이 많이 들기 때문에, Virtual DOM을 사용하면 이러한 비용을 줄일 수 있습니다.

작동 원리는 다음과 같습니다:

초기 렌더링: 애플리케이션이 처음 로드될 때, Virtual DOM은 실제 DOM과 일치하는 상태로 생성됩니다. 이 때, UI의 각 요소는 Virtual DOM 노드로 표현됩니다.
UI 변경: 애플리케이션에서 데이터가 변경되어 UI를 업데이트해야 할 때, 이 변경사항은 먼저 Virtual DOM에 적용됩니다. 이는 실제 DOM에 직접 적용하는 것보다 훨씬 빠릅니다.
Diffing 알고리즘: 변경사항을 실제 DOM에 반영하기 전에, Virtual DOM에 있는 모든 요소를 이전 Virtual DOM과 비교하는 과정을 거칩니다. 이를 통해 실제로 변경된 부분만 파악합니다.
배치 업데이트: 변경이 감지되면, 그 변경사항만을 모아서 실제 DOM에 한 번에 반영합니다. 이 과정을 배치 업데이트라고 하며, DOM 조작 횟수를 최소화해 성능을 향상시킵니다.
UI 반영: 최종적으로 실제 DOM이 업데이트되고, 사용자는 변경된 UI를 볼 수 있습니다.
이러한 프로세스는 사용자 인터랙션 또는 데이터의 변경이 있을 때마다 반복됩니다. Virtual DOM을 사용함으로써 불필요한 DOM 조작을 피하고, 결과적으로 더 빠른 UI 업데이트를 가능하게 합니다. 이는 특히 복잡한 애플리케이션과 대규모 업데이트가 잦은 경우에 성능상의 큰 이점을 제공합니다.

HTML/CSS

문제28 ▶ 브라우저 열고 OO 주소를 입력했을 때 일어나는 일을 설명해주세요.

합격 답변

브라이저 주소창에 url을 작성하면 해당 서버를 찾아간다. DNS에서 실제 서버가 어딨는지 주소를 찾고 나서 HTTP/HTTPS 프로토콜 방식으로 통신을 시도한다. 서버의 기본 설정이 대부분 index.html 파일로 되어 있어서 해당 파일을 클라이언트로 보낸다. 브라우저는 텍스트로 이뤄진 Index.html 파일을 파싱한다. 한줄씩 리딩하면서 DOM 트리를 완성한다. 중간에 link 태그를 만나서 css 요청이 발생하면 요청과 응답 과정을 거쳐 css를 파싱한다. 최종적으로 html을 모두 읽어내 DOM 트리를 완성한다. 완성된 DOM 트리와 CSSOM트리를 합쳐서 Render 트리를 완성한다. 중간에 script 태그를 만나면 자바스크립트 코드를 실행하기 위해 파싱을 중단하고 자바스크립트 엔진을 통해 관련 코드 혹은 파일을 로드해서 실행한다.

React

문제29 ▸ Redux, Recoil 등, 상태관리 라이브러리 사용한 경험을 공유해 주세요.

합격 답변

"네, 상태관리 라이브러리를 사용한 경험이 있습니다. 특히 Redux와 Recoil을 사용해 본 바 있습니다.

Redux는 React 애플리케이션에서 가장 인기 있는 상태 관리 라이브러리 중 하나입니다. 저는 큰 규모의 프로젝트에서 Redux를 사용하여 애플리케이션의 상태를 중앙에서 관리하는 아키텍처를 구축했습니다. Redux의 '단방향 데이터 흐름', '상태 불변성', '액션'을 통한 상태 업데이트 등의 원칙을 적용하여 예측 가능하고 유지보수가 쉬운 코드베이스를 만들 수 있었습니다. 또한, Redux DevTools를 사용하여 상태 변화를 시각적으로 모니터링하고 디버깅하는 데 큰 도움을 받았습니다.

한편, Recoil은 React에 좀 더 밀접하게 통합된 상태 관리 라이브러리로, 훅 기반의 API를 제공하여 React의 기본적인 훅 패턴과 잘 어우러집니다. 저는 특히 동적 데이터 흐름이 많은 대화형 UI를 가진 프로젝트에서 Recoil을 활용했습니다. Recoil의 'atom'과 'selector' 개념을 사용하여 글로벌 상태를 잘게 쪼개고, 각 컴포넌트에 필요한 데이터만을 구독하게 함으로써 성능 최적화에 기여했습니다.

Redux와 Recoil을 사용하며, 상황에 따라 적합한 상태 관리 전략을 선택하는 것의 중요성을 배웠습니다. Redux는 대규모 상태를 가진 복잡한 앱에 적합한 반면, Recoil은 상태의 변화를 더 세밀하게 다루어야 할 때 유리하다는 것을 경험했습니다."

개발경험

문제30 ▶ 지원자님은 평소 기술 향상을 위해 어떤식으로 노력을 기울이고 있는지 설명해주세요.

합격 답변

기술 향상을 위해 저는 주로 온라인 코스를 수강하고, 개인 프로젝트를 통해 배운 내용을 실제로 적용해 봅니다. 예를 들어, 최근에는 'React Hooks'에 대한 깊은 이해를 목표로 Udemy의 고급 강좌를 완료했고, 그 지식을 사용하여 한 사이드 프로젝트의 상태 관리를 개선했습니다. 또한, 새로운 기술 동향에 대해 최신 상태를 유지하기 위해 기술 블로그를 구독하고, 관련 기술 커뮤니티에 참여하여 다른 개발자와 지식을 나누고 있습니다. 이러한 노력은 최근에 수행한 프로젝트에서 시간 효율성을 높이는 데 크게 기여했습니다.

PART

3

예상문제

Javascript

문제 1 호이스팅(Hoisting)이 무엇인지 설명해 주세요.

Javascript

문제 2 var, let, const 차이를 설명해 주세요.

Javascript

문제 3 자바스크립트에서 클로져(Closures)에 대해서 설명해주세요.

개발경험

문제 4 성능 최적화 경험이 있으시면 설명해 주세요.

개발경험

문제 5 협업 경험이 있나요? 적극적으로 문제를 해결한 경험이 있으면 공유해주세요.

개발경험

문제 6 배포한 웹 사이트의 성능을 개선하기 위해 어떠한 노력을 했나요?

개발경험

문제 7 비개발 직군이랑 협업하면서 어려웠던 적이 있나요?

개발경험

문제 8 현재 우리 웹 페이지의 어떤 부분을 개선시키고 싶나요?

개발경험

문제 9 커뮤니케이션에 마찰이 있다면 어떻게 해결하는 편이신가요?

개발경험

문제 10 현재까지 진행한 프로젝트 중 발생한 이슈를 어떻게 해결했는지 하나만 설명해 주세요.

제로베이스에서 드리는

100% 취업 보장 혜택!

취업 실패 시 수강료는 전액 돌려드려요.

EVENT

제로베이스 스쿨 전과정 10만원 할인 쿠폰

100,000원

쿠폰코드 : 2024합격

사용 방법 : 수강신청 → 수강료 선택, 결제 페이지에서
쿠폰 코드 **2024합격** 를 입력하세요.

※ 쿠폰 수량 소진시 조기 종료될 수 있음

100%
현업 강사진

누구나 끝까지
취업 성공할 수 있도록
100% 현업 강사진 구성

100%
포트폴리오 무한첨삭

취업할 때까지 무제한
피드백으로 나만의
포트폴리오 준비 가능

100%
취업 보장

졸업 후 1년 이내
미취업시 전액 환불