

수료생 후기

**** 해당 내용은 수료생들로부터 내부공유용으로 전달받은 만큼,
개인적으로만 참고하시고 외부로의 유출은 하지 말아 주세요 ****

Q. 신입으로서 필요한 지식과 기술은 무엇이라고 생각하나요? (ex. JAVA, ORACLE, Linux 등 이 기술은 활용 비율이 높다 등)

- 지원하는 기업이 작은 기업일수록 한 가지 언어에만 특화되어 있지 않고, 여러 기술을 사용할 기회가 많아 기초적인 프로그래밍 지식과 정확한 레퍼런스를 통해 접근하는 연습이 필요할 것 같습니다. 아직까지는 현장에서 Java가 주로 활용되며, jQuery도 많이 쓰이고 있어 css 선택자를 능숙하게 활용하는 연습이 필요합니다. 최근 활용되는 AI 코딩 도구들 역시 강력하지만 정확한 출처를 요구하면 거짓으로 작성하는 경우가 많아, 해결책을 검색 혹은 서적을 탐구하여 찾아내는 기술이 가장 중요하다고 생각합니다.

- 자바에 대한 전반적인 개념과 학원에서 배우는 기술은 모두 학습된 상태로 취업 준비를 하시기를 바랍니다. Stream 클래스, ObjectMapper 클래스 등을 많이 쓰고 DB 관련은 with 구문이 뭔지 한번 보면 좋을 것 같아요. Join은 항상 사용하니 꼭 숙지하시구요.

- 버전 관리 시스템 하나는 반드시 알아야 한다고 생각합니다. 특히 Git(깃)을 알아 놓으면 좋습니다. 깃허브(GitHub)를 통해 프로젝트도 관리하기 좋고 깃허브를 잘 이용하면 취업하는데 도움도 됩니다.

- 신입으로서 교육에서 들었던 기술과 기본적인 알고리즘, 정보처리기사에서 자주 나오는 CS 지식 등이 필요하다고 생각하고, 저도 아직 부족해서 따로 공부 중입니다. 프로그래밍 언어를 두루두루 잘하는 것도 중요하지만 그보다 자신에게

맞는 한 언어를 취업 준비 기간 동안 최대한 깊게 파 보는 것도 좋은 경험이 될 것 같습니다.

- 협업에서 사용되는 기술은 학원에서는 보통 깃이라는 걸 썼는데 막상 회사에 와보니 실루엣이라는 전혀 다른 것을 사용하더라구요. 하지만 기본적인 틀 자체는 비슷했습니다. 신입으로 필요한 지식은 사실 누가 취업을 하던지 간에 학원에서 1차적으로 배우고 간 사람들이라면 다 거기서 거기 인 거 같아요. 저도 회사에 막상 가고 나니 사용하는 툴이나 언어가 다른 부분이 많아서, 사실상 거의 다 처음 배워야 하는 수준이었습니다. 하지만 그래도 자바와 오라클 같은 경우에는 배운 만큼 유용하게 써먹을 수 있었습니다. 툴은 달라져도 기반은 다르지 않았으니까요.

Q. 교육원에서 배운 지식 중 어떤 기술이 실제로 도움이 되었는지, 실무에서 사용하는 기술과 차이가 있는지 궁금합니다.

- JAVA 강의를 들으면서 배운 것은 기반을 다지는 기초 지식이라고 생각합니다. 입사 후 사용하는 언어가 다를 수도 있고(본인의 경우 php), 더 많은 것을 요구할 수도 있는데 그것들을 해내기 위한 가장 기본 틀을 잡는다고 생각하시면 됩니다. 알파벳을 배워야 단어를 읽을 수 있는 것처럼 말입니다.

- Data사업부로 입사를 하게 되어 JAVA보다는 쿼리를 많이 다루게 되었습니다. 그래서 오라클을 배우고 써봤던 경험들이 많이 도움이 되었습니다. Java 개발자(혹은 JavaScript)로 취업하신다면 DOM tree 구조 활용한 페이지를 만드는 프로젝트를 혼자서라도 꼭 진행해 보기를 권합니다.

- 교육원에서 배운 언어는 자바였는데 현재 현업에서는 다른 언어를 쓰고 있습니다. 어떤 언어를 사용하는지 보다 어떻게 공부하는지 배우는 게 더 중요한 것 같습니다. 기본기를 갖추었더니 다른 언어는 금방 익숙해지거든요. 그리고 생각보다 더 기본적인 CS 지식이 중요하다고 많이 느끼고 있습니다.

- SQL은 어떤 회사든 어떤 언어든 기본으로 쓰는 거기 때문에 SQL을 제대로 배워둔 게 큰 도움이 됐어요. 이제 3년차인 제가 느끼는 걸로는 확실히 SQL이 어디를 입사하든 요긴하게 써먹을 수 있다고 생각해요.

- SQL 열심히 공부하시면 도움이 많이 될 것 같습니다. 배운 지식 중에서 순위를 매겨 보자면 첫 번째는 SQL, 두 번째로는 Spring framework, 마지막으로는 jQuery입니다. 시스템의 유지보수를 하다 보면 jQuery가 사용되는 솔루션 제품이 많아서 여러모로 도움이 됩니다.

교육원 과정 중에는 javascript prototype 내용이 별로 없었는데 해당 방식으로 프로젝트를 진행해보면 도움이 될 듯합니다. 물론 신규 프로젝트에서는 es6, typescript, react, spring boot로 구성되는 빈도가 높아 해당 기술에 대한 지식도 갖출 수 있다면 좋을 것 같습니다. (일단 교육원에서 배우는 내용을 소화하는 게 우선입니다.)

- 회사는 복잡한 비즈니스 로직이 적용되어 학원에서 진행한 모의 팀프로젝트보다 훨씬 큰 규모와 코드양, 인프라를 경험하게 되는 것이지 배운 것과 실무진에 엄청난 차이가 존재하진 않다고 생각합니다. 이에 빠르게 적응하기 위해선 CS, 인프라적 지식도 반드시 기본 소양으로 공부해야 하며 자신이 작성하는 코드에 대해 데이터 흐름이 어떻게 진행되는지 철두철미하게 파악된 상태에서 코드를 작성하는 습관을 들여야 합니다.

- 개인적으로는 MVC MODEL2 구조의 이해와 SQL쿼리 짜는 능력, 자바스크립트 (제이쿼리 등) 능력이 많이 도움이 됐습니다. 각 업무마다 사용하는 툴도 다르고 개발 환경도 다 달라서 온통 새로 배워야 되는 것 투성입니다. 하지만 MVC 구조로 안되어 있는 소스는 없으며, 개발환경이 다 달라도 SQL 쿼리는 변하지 않습니다. 또한 서버 재기동을 최대한 줄이기 위해 JSP단에서 자바스크립트로 많은걸 제어 하더라구요. 스프링을 이용한 기본적인 CRUD 능력은 반드시 갖추셔야 하며, 앞서 말씀드린 3가지의 스킬을 좀 더 연마하신다면 좋을 것 같습니다. 실제 운영

되는 시스템의 소스를 보면, 소스 하나하나는 충분히 이해하고 짜볼 수 있는 소스였으나 전체적인 구조가 새로운 것들이 많았습니다. 사용자가 입력한 데이터를 DB에 저장할 때 PK값을 생성하는 것이 아닌 사용자 화면이 로드 될 때 PK를 생성하고, 그 키 값을 이용해서 중간 테이블에 필요한 데이터를 먼저 삽입 한다던지, 서버 재기동을 줄이기 위해 JSP에서 컨트롤 한다 던지 등 색다른 구조가 많이 있었던 것 같습니다.

Q. 지원 직무 기술습득 시 추천해주고 싶은 도서, 영상 등(유튜브, 인터넷 강의 등)이 있을까요?

- 학원에서 수행하는 업무에 대해서 정확한 레퍼런스를 참조하도록 노력한 것이 큰 도움이 되었고, 현업에서도 도움이 되고 있습니다. 자바의 로우 레벨을 참조할 때는 오라클의 Java API를 활용하고, Mybatis의 경우에도 공식 튜토리얼 문서가 큰 도움이 되었습니다. 실무를 진행하며 느낀 점은 화려한 기술이나 API 등 보다는 정확하게 자신이 원하는 기능을 기존 코드에서 분리해 낼 수 있는 리팩토링 기술(Refactoring) 이 제일 평가받는 느낌이 들었습니다.

- 코테 준비를 위해 프로그래머스 많이 풀었고, 티스토리나 블로그 등에 꼭 정리 해두세요. 기업에서도 프로그래머스 문제 그대로 내는 경우가 많습니다. 그 밖에 토비의 spring 책과 인프런 강의 추천하고, 자료 구조론과 알고리즘 공부하면 좋습니다.

- 제가 지원한 곳들 중엔 코테를 안 보는 회사도 많았습니다. 대신 코테를 안 보는 회사는 내가 아는 기술을 말로 설명해야 합니다. 코테를 안 봤지만 기술면접 합격했던 한 회사를 예를 들자면, 페이지징에 대해 프론트부터 백단까지 설명해 달라고 한 면접관이 있었습니다. 그래서 프론트 단에서 받아야 하는 파라미터 값부터 쿼리까지 일련의 과정을 전부 설명했던 기억이 납니다.

기술습득 시 추천하는 방법은 공식 홈페이지를 가까이하세요. 책은 공식 홈페이지

지를 요약하고 번역한 것입니다. 또한 기술을 외우지 말고 이해하세요. 제가 공부할 때 많이 실수했던 부분이, 사용 방법과 기술 패턴을 외우곤 했다는 겁니다. 외워야 하는 부분도 분명 있지만, 이해의 영역이 더 중요합니다. 어떻게 되는 거지? 왜 이렇게 할까? 항상 되물어보고 찾아보세요.

추천하는 도서는, 자바는 자바의 정석(저자 남궁성), 자바스크립트는 모던 자바스크립트 딥다이브(저자 이웅모), vue는 공식 홈페이지 튜토리얼과 가이드 봤습니다.(영어를 못하시나요? 괜찮습니다. 저도 번역기 돌려서 봤는데 이해했습니다.)

- 너무 기본이라고 생각될 수 있겠지만, 수업 시간에 했던 것들 위주로 공부했습니다. 저는 아날로그를 좋아해서 노트필기를 했었는데, 그때 쓴 필기 위주로 봤습니다.

Q. 포기하고 싶었던 순간, 슬럼프가 있었다면 어떻게 극복하셨나요? 또한 다시 과정을 다시 과정을 수강하던 때로 돌아간다면 어떤 부분을 더 공부하거나, 어떤 마인드로 임할 것인지 궁금합니다.

- 다시 과정을 수강하던 때로 돌아간다면 단순 기능 구현에 목 매지 않고 제가 배웠던 언어들이 어떠한 방식으로 작동하며, 어떤 특징을 가지는지를 조금 더 면밀히 파악하며 공부해보고 싶습니다.

- 프로젝트 때 포기하고 싶었습니다. 하지만 조원들에게 힘을 많이 얻었습니다. 다시 돌아간다면 자바 공부를 더 열심히 할 것 같습니다.

취업하고 나서 spring 프레임워크 공부를 다시 했는데 토비의 spring책이 정말 좋았습니다. 자바 웹 개발자에게 spring은 정말 중요합니다. 해당 책은 스프링을 위시하였으나 자바와 디자인 패턴, aop 등 많은 객체 지향 기술 관련 내용을 담고 있으니 반드시 보시길 추천합니다.

- 너무 어려워서 포기하고 싶을 때는 정말 내 머리를 쥐어박고 싶을 정도였는데

요. 그럴 때면 이해될 때까지 계속해서 코드를 짜보는 수밖에 없었습니다. 머리가 돌아가지 않는다면 그냥 손으로라도 익혀야지 하는 마음으로 했던 것 같습니다.

- 지금 하는 거 모른다고 해서 기죽지 말고 이런 기능이 있구나~ 라는 생각만 하면 됩니다. 시간 지나면 다 까먹어서 구글링하고, 다음날 또 까먹어서 구글에 물어봅니다. 학원에서 프로젝트 하다가 "이래서 CS 지식이 필요했구나" 라는 생각이 들어 다시 돌아간다면 더 열심히 할 거 같아요.

- 돌아설 곳이 없다고 생각하고 시작했기 때문에 포기하고 싶었던 순간은 단 한번도 없었습니다. 벽을 만날 때마다 '이 문제를 어떻게 해결하지'만 생각했습니다. 슬럼프는 주변사람들의 격려와 공부로 견뎌냈습니다.

다시 과정을 수강하던 때로 돌아간다 해도 크게 달라질 건 없을 것 같습니다. 조급한 마음을 공부로 풀어내서 당시에 이미 무리하게 공부했습니다. 말은 그럴 듯하지만 제 의지와 다르게 머리는 협조를 해주지 않아 공부가 잘 안되는 날이 훨씬 많았습니다. '대충이라도 하자', '누워서라도 하자', '10번 봐도 모르면 100번 보자' 그런 마음으로 공부했고 지금도 합니다.

제가 주니어 개발자라 이런 생각을 할 수도 있지만 제게 개발은 산 넘어 산입니다. 정상에 오르면(문제를 해결하면) 등산하면서 힘들었던 걸 잊는 사람, 그리고 다시 산에 가고 싶은 사람 그런 마음가짐으로 해야 버틸 수 있는 것 같습니다.

- 저는 취업준비를 하다가 왔던 케이스였기 때문에, 여기가 최후의 배수진이라고 생각해서 포기하고 싶다는 생각 자체를 하지 않았습니다. 대신 강의가 너무 힘들었을 땐 강의와 관련된 포스트를 인터넷에서 찾아보거나 했던 것 같습니다.

- 제 슬럼프는 전공생으로 대학을 다닐 때였는데 코딩을 처음 접하고 진짜 이게 맞는지 내가 이런 감감한 코드 속에서 갇혀서 평생 사는게 맞는건지 싶어서 다른 공부를 했습니다. 하지만 해당 공부에서도 역시 슬럼프가 와서 어쩔 수 없이 다시 웹 코딩을 공부하게 됐는데, 한 줄 한 줄 치면서 만들어지는 결과물에 대해

성취감을 느꼈고 결국은 웹 개발자로 취업을 하게 되었습니다. 다시 과정을 수강 하던 때로 돌아간다면 기초적인 개념 부분에 대한 설명을 놓치지 않도록 신경 써 보고 싶습니다.