

개발자가

비전공자도 할 수 있는 개발자 취업의 모든 것

되고 싶습니다



개발자 전망과
개발 분야 소개

학원 선택과
공부 방법

이력서,
포트폴리오,
면접 준비
방법

어디서도 이야기되지 않았던
개발자 도전과 성공에 대한 현실적인 이야기

앨런 지음

길벗

개발자가 되고 싶습니다

I want to be a developer

초판 발행 · 2023년 4월 26일

지은이 · 앨런

발행인 · 이종원

발행처 · (주)도서출판 길벗

출판사 등록일 · 1990년 12월 24일

주소 · 서울시 마포구 월드컵로 10길 56(서교동)

대표 전화 · 02)332-0931 | 팩스 · 02)323-0586

홈페이지 · www.gilbut.co.kr | 이메일 · gilbut@gilbut.co.kr

기획 및 책임 편집 · 변소현(sohyun@gilbut.co.kr) | 디자인 · 박상희 | 제작 · 이준호, 손일순, 이진혁, 김우식

마케팅 · 임태호, 전선하, 차명환, 박민영, 지운집, 박성용 | 영업관리 · 김명자 | 독자지원 · 윤정아, 최희창

교정교열 · 이미연 | 전산편집 · design KEY | 출력 및 인쇄 · 금강인쇄 | 제본 · 금강인쇄

- ▶ 잘못 만든 책은 구입한 서점에서 바꿔 드립니다.
- ▶ 이 책은 저작권법에 따라 보호받는 저작물이므로 무단전재와 무단복제를 금합니다. 이 책의 전부 또는 일부를 이용하려면 반드시 사전에 저작권자(©앨런, 2023)와 (주)도서출판 길벗의 서면 동의를 받아야 합니다.

ISBN 979-11-407-0427-9 13000

(길벗 도서코드 080352)

정가 20,000원

독자의 1초를 아껴주는 정성 길벗출판사

(주)도서출판 길벗 | IT교육서, IT단행본, 경제경영서, 어학&실용서, 인문교양서, 자녀교육서
길벗스쿨 | 국어학습, 수학학습, 어린이교양, 주니어 어학학습, 학습단행본

페이스북 · <https://www.facebook.com/gbitbook>

개발자가

비전공자도 할 수 있는 개발자 취업의 모든 것

되고 싶습니다



앨런 지음

길벗

비전공자로 개발 공부를 시작한 저자의 모든 고민과 시행착오, 진실된 조언들이 녹아 있습니다. 개발자가 되기 위해 고민하거나 이제 막 개발 공부를 시작했다면 망설이지 마세요. 당신의 고민 끝에 갈 길을 보여 주고 당신의 노력에 확신을 줄 것입니다.

김도엽 | 전 투자운용사 대표

개발자를 꿈꾸는 사람들을 위한 전략적인 취업 안내서입니다. 특히 비전공자들을 대상으로 프론트엔드, 백엔드, 모바일 분야에서 개발자로서 첫발을 내딛는 데 필요한 실용적인 조언을 하고 있습니다. 취업 준비를 위한 공부 과정과 면접에 관한 팁들도 유용할 것입니다.

김푸르외 | 카카오 모빌리티 개발자

대한민국에서 취업 준비생은 조급할 수밖에 없고 다른 경쟁자들보다 더 좋은 조건을 갖추기 위해 항상 노력해야 합니다. 그런데 그런 무게감을 내려 놓고 이 책에 제시되어 있는 길을 따라가 보세요. 많은 개발자 중에서도 나만의 특별한 존재감을 보여 줄 수 있는 방법이 보일 것입니다.

김은상 | 대학생

비전공자인 저자가 개발자가 되어 많은 이를 교육하면서 얻은 공부 방법과 취업 노하우를 고스란히 담고 있습니다. 개발자가 되고 싶지만 방법을 모르거나 어려움을 겪고 있는 분이 있다면 이 책을 가이드 삼으세요. 자신감 부여와 성장에 도움이 될 것입니다.

전은석 | 라인프렌즈 개발자

추천사

저도 개발자로 커리어를 전환하면서 여러 시행착오를 겪었습니다. 당시 ‘이런 책이 있었다면 얼마나 도움이 됐을까, 불필요한 시간 소비를 줄일 수 있지 않았을까’라는 생각이 듭니다. 프로젝트 매니저로 일하는 지금도 이 책은 많은 도움이 됩니다. 개발자와 소통하기 원하거나 혹은 개발자를 이해하고 싶은 현업의 디자이너, 프로젝트 매니저, 스타트업 CEO 분들에게도 이 책을 추천합니다.

김천수 | 하이퍼하이퍼 프로젝트 매니저

이 책은 개발에 대해 폭넓게 다루면서 저자의 독특한 도전 정신과 실제 경험에서 배운 교훈들을 통해 독자에게 열정과 자신감을 불어넣습니다. 더불어 기술 면접 대처법, 개발에 적합한 공부법, 흔들리지 않고 공부하는 방법 등을 저자의 경험과 삶에서 묻어 나오는 조언과 함께 소개합니다. 그저 자신의 성공담을 얘기하는 것이 아니라 독자들이 자신의 인생을 살면서 실제로 적용할 수 있는 가치를 전달합니다. 저자의 용기와 열정을 배워 이 책을 보는 모든 분이 성공적인 도전을 이어갈 수 있기를 바랍니다.

정창식 | 유어코드 대표

개발자로 전향하고자 고민할 때 궁금했던 내용을 저자의 경험을 살려 구체적이고 세세히 조언해 주어 좋았습니다.

김은지

‘비전공자의, 비전공자에 의한, 비전공자를 위한’ 이 한 문장으로 이 책을 설명할 수 있습니다.

황태현

300페이지가 안 되는 책에 개발자가 되기 위한 모든 내용이 함축되어 있습니다. 저자는 자신해서 길잡이가 되길 희망합니다.

김하은

개발자가 되기 위해 어떻게 방향을 잡을지 A to Z로 1:1 컨설팅을 받는 느낌이었습니다.

황윤주

코딩을 시작하는 누구나 시행착오를 겪을 수밖에 없는데, 저자는 자신의 경험을 토대로 시행착오를 줄여 주는 가이드라인을 제시합니다.

이상엽

저도 비전공자로 개발을 시작하면서 쉽지 않았던 기억이 떠올라 저자의 경험과 충고가 10년 전에 있었다면 얼마나 좋았을까 하면서 읽었습니다.

이호철

엔지니어에서 개발자로 이직하려는 지금 제 상황에서 많은 용기와 위로를 받을 수 있었고 제 자신을 더 믿기로 했습니다.

김혜진

부트캠프 수료를 앞두고 앞으로 나아가야 할 명확한 가이드를 받았습니다.

김유원

1부 > 개발자 전망과 개발 분야 소개

1장			
왜 개발자가 되어야 하는가	01	변하는 세상, 다가올 미래	017
	02	개발을 배워야 하는 이유	021
	03	개발자에 대해 자주 묻는 질문과 답변	029
2장			
개발 분야 선택하기	01	들어가기 전에	037
	02	프로그래밍 언어와 프레임워크	039
	03	프론트엔드와 백엔드	046
	04	개발 분야의 종류	048
	05	개발 분야별 기술	055
	06	나에게 맞는 분야 찾기	063
		• 정리하기	067

2부 > 개발자가 되는 과정과 공부 방법

3장			
공부 방법 선택하기	01	독학, 국비지원 학원, 사설 학원(부트캠프)	071
	02	공부 방법별 특징과 장단점	074
	03	학원 6개월 커리큘럼	077
		• 정리하기	079

4장		
학원 6개월 경험과 공부 팁	01 들어가기 전에	081
	02 학원 6개월 회고	083
	03 학원 수료 후 느낀 점	093
	04 학원의 명확한 장단점	099
	05 만약 그때로 돌아간다면	103
	• 정리하기	109

3부 > 개발자 취업을 위한 이력서, 포트폴리오, 면접 준비 방법

5장		
신입 개발자 취업 과정	01 신입으로 취업할 수 있는 수준	113
	02 개발자 채용 정보 사이트	117
	03 개발자 취업 과정	120
	04 알고리즘, 공부해야 할까	123
	05 알고리즘 공부 방법	127
	• 정리하기	132

6장		
이력서와 포트폴리오 작성 방법	01 작성 방법에 따른 제출 방법	135
	02 이력서 작성 방법	136
	03 포트폴리오 작성 방법	140
	04 이력서/포트폴리오 실제 사례	146
	05 그밖에 준비할 수 있는 것들	151
	• 정리하기	153

7장			
포트폴리오	01	직접 기획해 만들기	155
작성을 위한	02	클론 코딩하기	165
프로젝트 관리		• 정리하기	170
8장			
취업문 바로 앞,	01	기술 면접 준비 방법	173
면접 준비	02	기술 면접 대처법	177
	03	적절한 면접 태도	180
		• 정리하기	184

4부 > 시작하기 전에 알았더라면

9장			
시행착오를 줄이는	01	들어가기 전에	189
개발 공부 방법	02	개발 공부의 속성을 이해하자	191
	03	직접 쳐 보고 그려 보고 찾아보자	196
	04	흔들리지 않는 마음을 갖자	202
		• 정리하기	205
10장			
학습 자료 소개	01	코딩 강의 사이트	207
	02	개발자 양성 무료 아카데미	211
	03	부트캠프와 국비지원 학원	213
	04	개발 관련 유튜브 채널	216
	05	CS 참고 자료	218

06 개발 관련 추천 도서	220
07 개발 관련 깃허브	222
08 모바일 개발 정보	224

5부 > 마지막으로 당부의 말

11장	01 회사 선택과 멘토 찾기	229
취업 전후 업무, 경력 관리	02 취업 후 1년 차 수준	231
	03 취업 후 업무 태도	235
	04 취업 후 자기 계발	239
	05 마지막으로	247
	에필로그	249
	찾아보기	260

나는 경영학을 전공해 대학 시절에는 회계사를 준비했었다. 하지만 은행원으로 첫 직장 생활을 시작하게 됐고 중소기업에서 재무/회계 업무를 2년간 했다. 그러다가 늘 꿈꾸던 스타트업을 공동 창업했다.

- **첫 번째 창업**: 기획자이자 공동 창업자로 SNS 앱 출시
- **두 번째 창업**: 스타트업과 투자자를 이어주는 크라우드펀딩 서비스의 초기 멤버로 참여

부푼 꿈으로 시작한 스타트업은 실패와 시행착오의 연속이었다. 그래도 4년 동안 여러 개발자와 협업할 수 있었다.

2번의 스타트업 실패 후, 궁극적으로 내 서비스를 만들기 위해 이번에는 직접 개발을 배워 보자고 결심했다. 개발자에 도전하기로 마음먹은 그때, 내 나이는 서른 중반을 넘기고 있었다. 2019년 3월부터 8월까지 6개월 과정의 iOS 개발 과정을 수료하고 그 이후에는 독학했다.

같이 학원을 다녔던 친구들은 개발자로 취업했다. 나는 직접 앱을 개발하며 다시 스타트업을 창업했다. 정부 지원금을 신청했는데 못 받게 되며 사업이 중단됐다. 그러던 중 우연한 기회에 개발 과외를 해 내가 가르친 친구를 개발자로 취업시켰고 그 일을 계기로 온라인 강의까지 출시하게 되었다.

현재는 스타트업에 대한 꿈을 잠시 보류하고 강의에 매진하고 있다. 비전공자로서 개발자 준비 과정에서 겪은 시행착오 경험을 살려 ‘개발자 취업’ 강의와 내 이름을 건 ‘iOS 앱 개발 부트캠프’를 운영 중이다.

책을 집필한 이유

비전공자로서 개발자가 되기 위해 4년 넘게 공부하며 직접 개발해 보니 처음 공부하는 사람은 누구나 나와 비슷한 상황에 놓인다는 것을 알게 됐다. 같이 공부했던 비전공자 친구들도 똑같은 어려움을 겪었고, 이야기를 나눠 본 개발자 지인들도 크게 다르지 않았다.

이런 힘든 경험을 하고 난 지금, 결론은 무엇일까?

“개발은 생각보다 어렵지 않다.”

개발자가 되려고 시도했다가 포기한 사람이 있다면 아마도 잘 가르치지 못하는 강사를 만났고 진도까지 빨리 수업 대부분이 이해되지 않았을 가능성이 높다. 내용을 이해하지 못한 상태에서 재미를 느낄 틈도 없으니 급기야 ‘개발이 적성에 맞지 않는다’는 결론에 도달해 포기하게 된다. 많은 초심자가 이처럼 잘못된 방식으로 공부하며, 개발에 재미를 찾지 못하고 스트레스만 받다가 중도에 포기한다. 나 또한 그렇게 될 뻔한 사람 중 한 명이다.

개발을 공부할 때는 개발에 적합한 방법으로 접근해야 한다. 누군가에게는 사소할 수 있지만, 이런 점을 미리 알고 시작해야 비전공자도 ‘실패하지 않는’ 개발 공부를 할 수 있다.

내가 깨달은 점을 여러분에게도 공유하고 싶다. 아주 조그마한 팁을 알려 주려는 것이 목적이다. 이 팁들을 안다면 덜 고생하며, 중간에 포기하지 않을 수 있다고 믿는다. 이제 4년 차 개발자가 된 나는 개발자가 되는 것은 ‘누구나 가능’하며, 절대 어렵지 않다고 생각한다.

개발에 전혀 소질이 없다고 생각했던 나도 해냈다. 심지어 나처럼 비전

공자인 이들을 개발자로 취업할 수 있게 가르쳤으며, 이제는 온라인 강의
를 만들어 전공자까지 가르치는 나름 인기 강사가 됐다. 현재까지 누적 수
강생 수백 명에 유료 강의 수강평은 지난 몇 년 간 만점을 유지하고 있다.
내 강의를 정말로 좋아해 주는 팬들도 있다. 내가 가르친 1~3년 차 현직
개발자들이 ‘네카라쿠배(개발자 대우가 좋은 회사인 네이버, 카카오, 라
인, 쿠팡, 배달의민족을 묶어 부르는 말)’로 이직에 성공하기도 했다.

이런 결과가 과연 내가 잘해서일까? 아니다. 나는 남들보다 더 많이 사
집했고, 그래서 개발 공부의 방향성만큼은 누구보다 많이 고민했다. 물론
지난 4년간 쉽 없이 ‘좋은’ 개발자로 성장하기 위해 전공자 이상으로 공부
하고 노력했다.

이런 나의 경험을 공유하기 위해 이 책을 집필했다. 이 책에 나의 삽질
과 고민을 모두 담았다. 학원에 가기 전에는 무엇을 준비해야 하는지, 개
발 공부를 할 때 겪는 시행착오에는 어떤 것이 있는지, 올바른 공부 방향
은 무엇인지를 미리 알고 시작하자. 그러면 적어도 나와 같은 삽질, 즉 시
행착오를 겪지 않거나 겪게 되더라도 그 정도를 확실히 낮출 수 있다.

누군가는 ‘개발이 어렵다’, ‘내 적성에 맞지 않는다’는 결론에 빠르게 도
달해 쉽게 포기했을지 모른다. 하지만 꽤 높은 연봉을 받으며 어떤 기업에
서나 채용하고 싶어 하는 개발자로 사는 것, 그것은 누구라도 가능하다.
이제 그 이야기를 시작해 보자.

3장

공부 방법 선택하기





독학, 국비지원 학원, 사설 학원(부트캠프)

비전공자라도 개발자가 되는 방법은 무궁무진하다. 혼자서 독학할 수도 있고, 좀 더 짧은 기간에 효율적으로 공부하기 위해 학원에 들어갈 수도 있다. 일반적으로 신입 개발자가 되기 위해 공부하는 기간은 평균 8개월 ~1년이다. 이 책에서는 이 기간을 최대한 줄일 수 있는 방법도 이야기해 보겠다. 우선 개발자가 되기 위해 선택할 수 있는 공부 방법에는 어떤 것들이 있는지 살펴보자.

독학

비전공자가 독학으로 개발자가 된 사례가 많다. 책과 인터넷 강의를 활용하고 모르는 내용은 블로그 검색 등을 통해 꾸준히 공부한다면 혼자서도 가능하다. 공부에 대한 의지도 있으며, 개발 이외 다른 분야도 꾸준히 공부한 경험이 있어 혼자 공부하는 것에 익숙하다면 독학도 충분히 하나의 선택지가 될 수 있다. 주변에 가까운 개발자 친구나 선후배 등 물어볼 사람이 있으면 더욱 좋다.

다만, 아무래도 완전히 새로운 학문을 공부하는 것이라 위험 부담이 있

다. 예를 들어, 코드에서 세미콜론(;)이나 콤마(,) 같은 간단한 기호 하나만 빠져도 코드가 동작하지 않고 에러가 발생하기 때문에 혼자 공부할 경우 시행착오를 자주 겪게 될 확률이 크다. 그러면 평균적으로 공부 기간이 늘어날 수 있다. 혼자서는 방향성을 명확하게 인지하고 시행착오 없이 공부하기가 힘들다.

국비지원 학원

다음으로는 국가에서 지정한 학원, 즉 국비지원 학원에서 공부할 경우가 있다. 국비지원 학원은 고용노동부에서 학원비를 지원해 주는 제도로 운영되는 곳이다. 국가에서 특정 산업 분야의 인재를 양성하고 직무 능력을 향상하기 위해 학원비를 지원하는데, IT 개발 분야의 과목도 개설돼 있다. 실업 상태에서 취업을 준비한다면 이런 혜택을 누릴 수 있다. 다만, 학원비를 지원받으려면 여러 조건이 필요하므로 본인이 조건에 해당하는지 확인한 후 등록해야 한다.

일반 사설 학원의 6개월 과정이 500~600만 원 선인 것과 비교해 국비지원 학원은 분명 큰 장점이 있다. 그러나 개설된 과정이 백엔드 개발자 과정인 자바-스프링과, 요즘 뜨고 있는 AI 분야에 집중돼 있다는 한계가 있다. 국비지원 학원이 많고 학원에 따라 강의 질이 차이가 날 수 있어 사전에 좋은 학원을 고르기 위해 발품을 팔아야 한다. 개발자 커뮤니티인 OKKY에서 강의나 학원에 대한 수강평을 사전에 조사해 보고 선택하는 것도 하나의 방법이다.

시설 학원(부트캠프)

일반 시설 학원도 선택지가 될 수 있다. 최근 이런 학원을 부트캠프라고 부르는데, 부트캠프는 ‘신병훈련소’를 뜻한다. 신입 병사를 훈련시키는 것처럼 단기간에 개발자를 키워낸다는 뜻에서 미국, 유럽 등지에서 몇 개월 이상으로 된 개발자 양성 과정을 개설한 데서 유래한 말이다.

기간은 국비지원 학원과 비슷하게 6개월 과정이 많고, 학원비는 최소 500만 원부터이고 1,000만 원이 넘는 과정도 심심치 않게 볼 수 있다. 그러나 비싸다고 해서 무조건 좋은 것은 아니다. 그러니 상술에 넘어가지 않았으면 좋겠다. 최근에는 3~4개월 단기 과정도 개설되고 있는데 최소 6개월 이상 과정을 추천한다.

인터넷에서 부트캠프를 검색해 보면 많은 학원이 검색되는데, 개인적으로는 거의 차이가 없다고 본다. 6개월 동안 배울 수 있는 내용이 한정적이기도 하고, 친절하고 자세하게 가르쳐 주기에는 한계가 있다.

학원에서는 단기간에 많은 내용을 가르쳐 개발자로 취업시켜야 하므로 진도를 빨리 나갈 수밖에 없다. 어느 정도 예습하지 않고 학원에 갔다간 후회하거나 중간에 그만두는 일이 생길 수 있다. 따라서 반드시 예습하고 들어가야 한다. 이런 내용에 대해서는 4장에서 자세하게 설명한다.



공부 방법별 특징과 장단점

앞에서 독학, 국비지원 학원, 사설 학원(부트캠프)에 대해 간략하게 살펴봤다. 이 절에서는 공부 방법별 특징과 장단점을 표로 비교해 보며 좀 더 일목요연하게 정리해 보자. 장단점을 알면 시행착오를 조금이나마 덜고 각자 상황에 맞게 선택할 수 있다.

경제적으로 여유가 된다면 당연히 검증된 사설 학원(부트캠프)의 6개월 이상 과정을 추천한다. 여러 가지로 공부하기 좋은 환경이 마련돼 있고 시스템도 잘 갖춰져 있기 때문이다. 모르는 것을 상시로 물어볼 수 있는 질의응답 시스템(조교 등)도 있어 공부에만 집중할 수 있다. 또한 큰돈을 들이면서까지 개발을 공부하기로 결정한 사람들이 모여 있기 때문에 학습에 대한 열의도 높다.

차선책은 국비지원 학원이다. 국비지원을 받을 조건이 될 경우 6개월 동안 수업을 무료로 들을 수 있다는 장점이 있기 때문이다. 다만, 강의 질 차이가 크기 때문에 좋은 학원을 잘 알아보고 신중하게 선택해야 한다.

▼ 표 3-1 독학, 국비지원 학원, 사설 학원(부트캠프) 비교

구분	독학	국비지원 학원	사설 학원(부트캠프)
특징	<ul style="list-style-type: none"> 인터넷 강의, 책으로 공부 의지와 방향성이 가장 중요 주변 도움이 있으면 좋음 비전공자가 성공하기 쉽지 않음 	<ul style="list-style-type: none"> 선택하는 학원, 강사에 따라 크게 좌지우지될 수 있음(강의 질 차이 가 큼) 과정, 분야 선택이 중요(중간에 바꾸기 어려움) 	<ul style="list-style-type: none"> 선택하는 학원, 강사에 따라 크게 좌지우지될 수 있음 과정, 분야 선택이 중요(중간에 바꾸기 어려움) 취업 관련 정보 접근성이 높음
장점	<ul style="list-style-type: none"> 원하는 시간을 알차게 활용 가능 자신에게 맞는 강의와 강사 선택 가능 취업까지 시간이 충분하고, 방향성이 올바르다면 추천(대학생 등) 	<ul style="list-style-type: none"> 그나마 체계적으로 공부할 수 있음 국비지원 조건이 된다면 학원비 무료(500~600만 원 상당) 약간의 교통비 지원(10~40만 원) 같이 공부하는 동기가 생김(향후 업계 정보 공유) 	<ul style="list-style-type: none"> 시스템이 좋음 주변 공부 환경이 좋음 취업 확률이 높은 편(반드시는 아님) 공부하는 주변 친구들의 열의가 높음 같이 공부하는 동기가 생김(향후 업계 정보 공유)
단점	<ul style="list-style-type: none"> 취업 관련 정보 부족 (물어볼 사람이 없다면)문제 해결 어려움 어떻게 공부해야 할지 방향성을 고민하게 될 가능성 높음 	<ul style="list-style-type: none"> 취업 관련 정보가 부족할 수도 있음 최신 기술을 배우지 못할 수 있음 강사와 학생들의 열의가 떨어질 수 있음(Case by Case) 반드시 취업이 보장되는 것은 아님 	<ul style="list-style-type: none"> 학원비가 비쌌(최소 500만 원 이상, 생활비 포함하면 1,000만 원 이상 필요) 중간에 포기할 경우 높은 기회비용이 발생함 반드시 취업이 보장되는 것은 아님

물론 사설 학원에 다니든 국비지원 학원에 다니든 6개월 만에 모든 준비가 완벽하게 된 개발자는 될 수 없다. 6개월간의 커리큘럼을 마치고 나면 ‘아, 개발이 이런 것이구나!’ 정도를 파악할 수 있는 최소한의 수준은

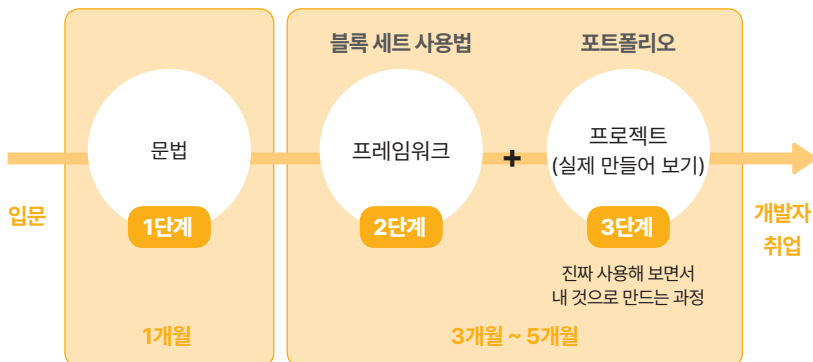
된다. 학원에 다니며 부족한 부분은 추가로 온라인 강의나 책으로 채우면 된다. 여러 온라인 강의를 미리 듣고 어느 정도 연습한 상태로 국비지원 학원에 가서 실력을 키울 수 있다면 일반 사설 학원보다 효과가 더 클 수도 있다. 어느 정도 연습했다면 국비지원 학원을 고려하는 것도 추천한다.



학원 6개월 커리큘럼

국비지원 학원이나 사설 학원(부트캠프)은 신입 개발자로 취업하기 위해 필요한 최소한의 내용을 바탕으로 커리큘럼이 짜여 있다. 다음 그림은 일반적인 신입 개발자 6개월 과정의 커리큘럼이다.

▼ 그림 3-1 신입 개발자 6개월 과정의 커리큘럼



[1단계] 문법 학습

모든 학원에서 첫 달은 해당 분야에서 사용하는 프로그래밍 언어를 배우는 과정으로 이루어져 있다. 국비지원 학원에서 가장 많이 개설된 자바-스프링 백엔드 과정을 기준으로 보면 첫 1개월 동안은 자바를 배운다.

[2단계] 프레임워크 학습

문법을 배운 후 3개월 정도는 프레임워크를 배운다. 앞에서 프레임워크는 장난감 블록 세트와 유사한 개념이라고 설명했던 것처럼 이미 누군가가 잘 만들어 놓은 블록 세트의 사용법을 배우는 것이다. 첫 달에 배웠던 문법을 기반으로 코드 블록을 적절하게 조립하는 방법과 노하우를 훈련한다.

보통 백엔드 과정에서는 서버를 만드는 데 필요한 데이터베이스를 구성하는 방법, 프론트엔드와 의사소통하기 위해 API를 만드는 방법 등을 배운다. 앱 개발 과정이라면 기본 앱에서 화면을 만드는 법, 버튼을 만드는 법, 화면 배치를 하는 법 등을 배우게 된다.

[3단계] 개인 또는 팀 프로젝트

4개월 동안 기본적인 문법-프레임워크 과정을 마치면 마지막 단계로 개인 프로젝트 또는 팀 프로젝트를 2개월 동안 진행한다. 1, 2단계에서 배웠던 내용을 바탕으로 쇼핑몰, 게시판, SNS, 동영상 스트리밍 서비스 등 실제 취업 후 실무에서 만들 정도의 프로젝트에 도전한다.

총 6개월 과정을 마치면 어느 수준이 될까? 기본적으로 본인이 원하는 분야의 신입 개발자로 취업할 수 있는 최소한의 지식을 한 번 훑은 정도가 된다. 이후 학원을 졸업하면 그동안 배운 내용을 복습하고, 이력서와 포트폴리오에 들어갈 나만의 프로젝트를 추가로 진행하면서 면접을 준비하는 등 최소 2~3개월을 보낸다. 보통 학원 과정을 6개월 동안 밟는다고 가정하면 평균적으로 8~9개월 정도가 실제 학원에 입학하는 시점부터 취업하기까지 걸리는 기간이라고 보면 된다.

1 개발자가 되기 위해 선택할 수 있는 공부 방법에는 3가지가 있다.

- **독학:** 책과 인터넷 강의를 활용하고 모르는 내용은 블로그 검색 등을 통해 공부하는 방법이다. 원하는 시간에 자신에게 맞는 강의와 강사를 선택해 공부할 수 있지만, 어떻게 공부해야 할지 방향을 잡지 못하면 시간이 오래 걸릴 수 있다.
- **국비지원 학원:** 국가에서 지정한 학원에서 학원비를 지원받으며 공부하는 방법이다. 체계적으로 공부하면서 무료로 다닐 수 있으나, 선택하는 학원과 강사에 따라 강의 질이 크게 좌우되므로 사전에 좋은 학원을 고르기 위한 노력이 필요하다.
- **사설 학원(부트캠프):** 개발자 양성 집중 과정을 운영하는 사설 학원에 다니며 공부하는 방법이다. 시스템이 좋고 졸업 후 취업 연계, 업계 정보 공유 등의 이점이 있으나 학원비가 비싸고 중도에 포기할 경우 높은 기회비용이 발생한다.

2 국비지원 학원이나 사설 학원(부트캠프)의 신입 개발자 과정은 보통 6개월 과정으로 짜여 있으며 문법 학습, 프레임워크 학습, 프로젝트 순으로 공부한다.

5 장

신입 개발자 취업 과정





신입으로 취업할 수 있는 수준

어느 정도 실력이 되어야 신입 개발자로 취업할 수 있을까? 이 질문에 대한 완벽한 정답은 없다. 신입 개발자의 실력도 천차만별이다. 같은 신입 개발자라도 1~2년 차 이상의 실력을 가진 사람도 있고, 개발자로 취업하기에는 아직 모자란 사람도 있다.

여기서는 평균적으로 신입 개발자 취업에 성공하는 최소한의 기준에 대해 이야기해 보겠다.

회사가 신입 개발자에게 바라는 최소한의 수준

일반적으로 회사가 신입 개발자에게 바라는 최소한의 수준은 ‘클라이언트(웹, 앱)와 서버 간에 데이터를 주고받는 내용이 포함된 프로젝트를 해 본 적이 있는가’다. 즉 회사는 네트워크 통신을 활용한 프로젝트 경험 여부를 본다.

왜 네트워크 통신이 중요할까? 실무에서는 클라이언트(프론트엔드)와 서버(백엔드) 간에 데이터를 주고받는 일을 빼놓고 그 어떤 프로젝트도 완성할 수 없다. 아무리 단순한 홈페이지라도 해당 인터넷 주소에 접속하면

서버에 데이터를 요청하고 응답받는 과정을 거친다. 어떤 서비스도 예외일 수 없다.

한 쇼핑몰이 있다고 가정해 보자. 제품마다 품목명, 사진, 상세 옵션, 재고 수량 등이 있을 것이다. 서버는 이러한 정보를 데이터베이스에 가지고 있다가 클라이언트가 “A 제품의 상세 정보를 보여 줘.”라고 요청하면 적절한 데이터를 찾아 응답한다. 클라이언트는 서버의 응답을 받아 처리한 후 웹 페이지(또는 앱)에 표시한다. 또 사용자가 A 제품을 주문하면 주문 품목명, 개수, 주소, 연락처 등의 정보를 받아 서버에 전달하고, 서버는 또다시 내부 로직을 거쳐 결과를 클라이언트에 응답한다. 이렇게 클라이언트와 서버 처리가 원활하게 이루어져야 쇼핑몰이 제대로 동작한다.

물론 이런 구현이 말로 표현했을 때는 간단하게 보일 수 있다. 하지만 서버와 클라이언트 간에 데이터를 주고받는 프로그램을 만드는 일은 개발을 처음 하는 사람에겐 높은 진입 장벽이다.

웹 개발이든, 앱 개발이든 프론트엔드 개발자는 ❶ 서버에 적절한 요청 보내기 ❷ 서버가 응답한 데이터를 제대로 받고 있는지 확인하기 ❸ 받은 데이터를 변환해서 화면에 적절하게 표시하기 등 일련의 과정을 구현해야 하는데, 초심자에겐 만만치 않다.

왜 그럴까? 프론트엔드 개발을 위한 프로그래밍 언어도 이해해야 하고, 이를 바탕으로 인터넷 세계의 약속을 코드로 구현해 응용할 줄도 알아야 하기 때문이다. 여기서 인터넷 세계의 약속이란, 서로 다른 컴퓨터 간에 데이터를 주고받기 위해 약속한 체계인 HTTP 프로토콜을 말한다.

이러한 상황은 백엔드 개발자라고 하더라도 별반 다르지 않다. 프론트엔드에서 요청한 내용을 적절하게 처리하려면 백엔드 개발에 사용하는 프로그래밍 언어와 인터넷 통신의 주요 과정을 알아야 한다. 기업은 적어도 이런 프로젝트 경험이 있는 사람을 원한다.

신입 개발자 채용 공고를 자주 확인하자

개발자 채용 공고를 보면 어떤 회사든 ‘자격 요건’이 있다. 지금 당장 지원하지 않더라도 내가 가고 싶은 회사에서 원하는 기술에는 어떤 것들이 있는지 구체적으로 확인해 보길 바란다. 자격 요건은 회사가 사용하는 기술인 동시에 회사가 자사 기술 중에서 중요하게 생각하는 부분이다.

물론 자격 요건이라고 해서 반드시 해당 기술을 ‘완벽하게 잘’ 알고 있어야 하는 것은 아니다. 회사가 원하는 기술 5개 중에 4개 또는 3개만 잘 알아도 나머지 부족한 기술은 다른 부분에서 본인의 장점을 통해 만회할 수 있다.

그리고 필수 항목과 우대 항목도 잘 구별해야 한다. 필수 항목은 반드시 구현할 줄 알아야 하는 것이고, 우대 항목은 말 그대로 구현할 줄 알면 우대를 받을 수 있는 항목이다. 때로는 우대 항목에 너무 많은 요구 사항들이 적혀 있어 주눅이 들 수도 있다. 하지만 필수 항목 위주로 자신의 수준을 객관적으로 판단했을 때 필수 항목에 대부분 만족한다면 지원할 자격이 충분하다. 우대 항목을 어설픈다고 알고 있는 것보다 필수 항목에 대한 기본기가 탄탄한 것이 훨씬 낫다.

채용 공고가 학습 가이드라인이 되기도 하니 해당 기술들을 잘 참고해 다음에 무엇을 공부할지 목표를 세우는 데 활용하기 바란다.

그래도 감이 안 오면? 일단 지원해 보자

나는 취업 준비생을 많이 봐 왔고 그들이 실제 취업에 성공하는 것도 옆에서 지켜봤다. 그들이 공통적으로 두려워하는 것이 무엇인지도 잘 알고

있다. 바로 본인의 실력이 부족하다고 생각해 실제 채용 과정에 뛰어들지 않고 계속 공부만 하려고 한다는 점이다.

하지만 공부만 한다고 해서 취업이 될 리 없다. 최소한 관심 있는 회사나 그에 준하는 회사에 서류를 넣어 봐야 하고 그 과정에서 탈락도 경험해 봐야 한다. 운이 좋게 서류에 통과했다면 면접도 경험해 봐야 한다. 그래야 본인의 실력을 객관적인 잣대로 판단하고 제대로 판가름해 볼 수 있다.

면접 과정에서 본인의 부족한 점을 깨닫는다면 그런 부분들을 좀 더 채워서 다음 면접을 보면 된다. 그러면 대부분은 면접을 더 잘 볼 수 있게 된다. 면접을 보다 보면 면접 문제가 회사마다 크게 차이 나지 않는다는 점도 알게 된다. 그래서 경험이 필요하다. 한두 번 면접을 경험해 보면 그 이후부터는 훨씬 더 무덤덤하게 본인의 실력을 뽐낼 수 있다.

또한, 채용 과정에서 단 한 번이라도 합격해 본 사람은 그 이후 2번, 3번 쉽게 합격한다. 다시 말하지만 처음이 어려울 뿐이다. 탈락을 두려워해서는 안 된다. 탈락도 경험이고 이런 경험을 많이 해 봐야 다음에 합격할 확률이 높아진다. 어느 정도 준비했다면 무작정 도전해 보자.



개발자 채용 정보 사이트

개발자 채용 공고는 일반적으로 모든 구직자가 이용하는 잡코리아 JobKorea, 사람인Saramin 같은 사이트에서도 확인할 수 있지만 개발자 채용과 구직에 특화돼 있는 사이트를 활용할 수도 있다. 개발자 일자리에 특화된 검색 기능이 있기 때문에 검색하기도 쉽고 여러 회사의 조건을 비교해 보기도 편하다.

개발자 구인구직에 최적화된 사이트

원티드

최근 몇 년 동안 급격하게 성장한 사이트다. 최근에 잘 나가는 IT 스타트업의 채용 공고는 대부분 이곳에 올라오므로 잘 확인해 보기 바란다. 근무환경이 좋은 회사인지, 연봉이 높은 회사인지, 기타 다른 부분에 특화된 회사인지 등을 테마별 검색 기능으로 확인할 수 있다.

- 원티드: wanted.co.kr

점핏

국내 구인구직 플랫폼 중 가장 규모가 큰 사람인Saramin에서 만든 개발자 전용 구인구직 플랫폼이다. 회사가 원하는 기술, 즉 자격 요건을 확인하기 편하며, 다른 기업들의 유사 포지션도 추천해 주는 기능이 있어 원하는 일자리를 검색하기 좋다.

- 점핏: jumpit.co.kr

렐릿

개발 관련 강의 사이트 중 국내에서 규모가 가장 큰 인프런이 만든 개발자 전용 구인구직 플랫폼이다. 개발자 구인구직에 특화돼 있어 일자리 카테고리도 잘 갖춰져 있고, 검색이 편하다. 또한 비슷한 채용 공고를 추천해 주는 기능이 있어 원하는 일자리를 검색하기 수월하다.

- 렐릿: rallit.com

프로그래머스

프로그래머스는 국내에서 가장 유명한 코딩 테스트 플랫폼이다. 개발자 취업 준비를 위해 코딩 테스트 문제를 풀기도 하고 코딩 테스트 관련 강의도 수강할 수 있다. 취업을 준비하는 예비 개발자들이 모여 있는 곳이다 보니 개발자 채용 관련 섹션도 있다. 여기에도 채용 공고가 많이 올라오니 잘 활용해 보길 바란다.

- 프로그래머스: programmers.co.kr

스타트업 채용에 특화된 사이트

로켓펀치

이제 막 창업해 미래가 촉망되는 스타트업에서 일하길 원하는 경우 로켓펀치를 이용하면 된다. 로켓펀치는 스타트업 정보를 공유하고 인맥을 연결하는 플랫폼으로, 스타트업 채용 섹션에 개발자 구인구직 정보가 활발하게 올라온다.

소규모 스타트업에 지원하는 것은 장단점이 아주 명확한 일이다. 평생 개발자로서 커리어를 발전시키는 것에 관심이 있다면 스타트업에서 일하는 것을 권하지는 않는다. 아무래도 적은 인원이 일하고, 체계가 잡혀 있지 않을 가능성이 크다. 사수가 없는 경우도 많다. 그래서 개발자로서 체계적인 과정을 밟으며 성장하지 못할 수도 있다.

그렇다고 해서 단점만 있는 것은 아니다. 소규모 인원이다 보니 가족 같은 분위기에서 일할 수 있는 경우도 있고, 신입이라 하더라도 많은 부분에 권한이 있는 경우도 있다. 사수 없이도 개발과 관련해 스스로 문제를 해결하는 것을 즐긴다면 스타트업도 괜찮은 대안이 될 수 있다. 또 향후 개발자로서 커리어보다는 개발 지식을 활용해서 사업하는 것을 고려하고 있다면 사업을 간접적으로 경험해 볼 수 있는 좋은 기회가 될 것이다.

그 외에 정말 드물긴 하지만, 정말 마음에 들고 향후 몇 년 안에 급격하게 성장할 것 같은 서비스의 초기 멤버로 합류해 회사와 같이 성장하면서 성취감을 느낄 수도 있다. 토스(비바리퍼블리카), 하이퍼커넥트, 지그재그, 오늘의집과 같이 요즘 대세로 떠오르는 서비스도 몇 년 전만 해도 작은 스타트업이었다는 것을 명심하자. 드물지만 이런 회사의 초기 멤버들은 스톡 옵션을 받아 몇십 억을 번 경우도 있다.

- 로켓펀치: rocketpunch.com



서른 중반의 나이에 공부를 시작해 개발자가 되었다

서비스 기획자로 일하다가 직접 개발자가 되기 위해 공부를 시작했습니다. 퇴직금의 절반은 학원 비로, 절반은 생활비로 쓰며 6개월 간 학원에 다녔고 그 이후엔 독학해 개발자가 되었습니다. 이 책은 어디서도 이야기되지 않았던 개발자 도전과 성공에 대한 현실적인 이야기를 합니다. 비전 공자가 개발자로 취업하기까지 공부 방법, 취업 준비 과정과 팁을 자세히 알려줍니다. 누구나 시작할 수 있지만 아무나 될 수 없는 개발자. 이 책으로 시행착오를 줄이며 시작해 보세요.

개발자 전망

학원 선택

공부 방법

이력서 작성법

멘탈 관리

포트폴리오 작성법

프로젝트하는 법

코딩 테스트 준비

면접 준비

개발자로 전향하려고 고민할 때 궁금했던 내용을 저자의 경험을 살려
구체적이고 세세히 조언해 주어 좋았습니다.

— 김은지

개발자가 되기 위해 어떻게 방향을 잡을지 A to Z로
1:1 컨설팅을 받는 느낌이었습니다.

— 황윤주

코딩을 시작하는 누구나 시행착오를 겪을 수밖에 없는데,
저자는 자신의 경험을 토대로 시행착오를 줄여 주는 가이드라인을 제시합니다.

— 이상엽



개발자가 되고 싶습니다

I want to be a developer

20,000원



9 791140 704279
ISBN 979-11-407-0427-9