

웹과 앱 기초부터 개발자와의 원활한 소통까지  
한 권으로 보는 IT 지식과 개발 용어

민완기 지음

# 아는 만큼

# 보이는

# IT 지식



낮선 IT  
기술과 용어를  
한번에

비전공자의  
마음으로  
쉽고 명쾌하게

개발자와  
잘 소통하는  
법

길벗

## 추천사

이 책은 비전공자가 일을 하다 접하게 되는 IT 기술의 개념을 명확히 이해할 수 있도록 돕습니다. 클라이언트-서버 구조, IP 주소와 도메인 주소 같은 인터넷 기초 용어와 각종 개발 용어를 일상적인 비유를 들어 쉽게 설명하고, 개발자와 원활히 소통하기 위해 알아야 할 개발 용어와 실무에서 활용할 수 있는 IT 지식을 깊이 있게 다룹니다. 이 책은 IT 분야의 기초를 다지고자 하는 모든 이에게 유익한 길잡이가 될 것입니다.

**삼성전자 엔지니어 하지민**

개발 세계의 다양한 분야와 그에 관한 지식을 이 책 한 권으로 정리할 수 있습니다. 웹과 앱 등 IT 관련 용어와 IT의 생리가 생소했던 독자도 ‘아, 이렇게 돌아가는구나!’라고 터득할 수 있고, 복잡하게 느껴지는 개념이 일상의 예시와 함께 설명돼 있어 쉽게 이해할 수 있습니다. IT를 처음 접하는 사람들에게 필독서로 추천하고 싶습니다.

**오가닉비즈니스(주) 대표 현홍수**

기술적 배경이 없는 사람도 이 책을 통해 IT의 본질을 자연스럽게 이해할 수 있습니다. 기초 개념부터 차근차근 설명하며, 이론에 그치지 않고 실무에 바로 적용할 수 있는 팁도 제공해 IT 업무를 처음 하는 사람이 빠르게 적응할 수 있도록 도와줍니다. 이 책은 후배나 팀원에게 IT 지식을 전수해야 하는 IT 전문가에게도 훌륭한 참고서가 될 것입니다.

**리테일 IT 전문기업 황혜진**



웹과 앱 개발에 관심이 있지만 어디서부터 시작해야 할지 모르는 이들을 위한 책입니다. IT 관련 기본 용어는 물론이고 비개발자와 개발자의 원활한 소통 방법까지 다루고 있어 실무에 바로 적용할 수 있는 지식을 얻게 됩니다. IT 용어가 외계어처럼 느껴졌던 독자라면 이 책을 통해 귀가 트이는 경험을 할 수 있습니다.

**(주)비앤케이 영상팀 정낙현**

개발자와 소통을 잘하고자 하는 이들에게 필요한 책입니다. 프로젝트 진행 중 발생하는 기술적 문제에 대해 정확한 질문과 피드백을 주고받을 수 있는 능력을 키워주고, 개발 프로세스와 네트워크 구조를 이해해 비개발자도 기술적 논의에 자신 있게 참여할 수 있도록 돕습니다.

**위밋모빌리티 연구소장 장건웅**

개발자와 소통할 때 한 번이라도 어려움을 겪어본 적이 있다면 어떻게 소통해야 하는지 이 책에서 힌트를 얻을 수 있습니다. 업무를 하다 맞닥뜨리는 문제의 원인과 갈등에 대해 저자의 노하우와 함께 구체적인 해결책까지 제시하기 때문에 개발자와 일하는 독자, 이제 막 개발을 공부하기 시작한 독자에게 유용합니다.

**일루미나리안 프로덕트 디자이너 이중혁**

## 지은이의 말

회의 자리에서 개발자의 말을 알아듣는 척하며 조마조마했던 적이 있나요?  
IT 회사에 취업하고 싶은데 비전공자라 걱정했던 적은요?

전공자가 아니라면 IT 용어가 생소하게 느껴지는 것이 당연합니다. 인터넷에서 검색해보도 어려운 말로 설명돼 있어 더 깊이 파고드는 것을 포기하게 됩니다. 또한 개발자에게 요청하고 싶어도 기획한 서비스가 구현 가능한지 판단이 서지 않아 자신 있게 말하지 못하는 경우도 있습니다.

필자도 그랬습니다. 시각 디자인을 전공했지만 개발자의 연봉이 높다는 말에 개발자로 전향했습니다. 주변에 컴공과 선배도, 개발자 사수도 없어 오랫동안 안 미지의 영역에서 홀로 싸웠습니다. 혼자 해내야 했기에 매일 벼랑 끝에 서 있는 심정이었습니다.

그래서 여러분의 기분이 어떤지 잘 압니다. 이제야 밝히지만, 필자도 신입 시절에 외주 업체에 개발 용역을 맡길 때 개발자의 이야기를 알아듣는 척한 적이 있습니다. 알아보기만 안 된다며 팀장 직함이 찍힌 명함을 받았기 때문입니다. 이는 비단 필자만 겪은 일이 아닐 것입니다.

이 책은 다음과 같은 독자를 위해 쓰였습니다.

- 개발자와 협업한 지 얼마 안 된 마케터, 기획자
- IT 회사에 취업하려고 준비 중인 문과생
- 외주 업체에 웹/앱 개발을 의뢰하려는 회사의 담당자
- 뒤늦게 개발자로 전향하려고 고려 중인 독자
- IT 서비스의 구조와 동작 원리를 알고 싶은 독자
- 자녀를 IT 계열 학과에 진학시키고 싶은 학부모



IT 지식이 부족한 경우 개발자와 잘 소통하려면 코딩을 배워야 한다는 생각에 초보자가 배우기에 좋다는 파이썬부터 공부하곤 합니다. 그런데 이는 집공사를 인테리어 업자에게 맡기기 전에 자신이 직접 타일 시공을 배우는 것과 같습니다. 집을 어떻게 꾸미고 싶은지 인테리어 업자에게 잘 전달하고, 어떤 부분은 왜 고칠 수 없는지 이유를 알면 그만인데 말입니다.

이 책은 IT 관련 회사 혹은 IT 관련자와 일하는 사람이 알아두면 좋을 ‘IT 기초 지식’ 및 ‘개발자와 소통하는 법’을 다룹니다. 인터넷의 동작 원리부터 웹, 앱, 데이터베이스의 기본 개념을 설명하고 프로그래밍 언어의 종류, 알아두면 좋은 개발 용어, 개발자와의 소통 방법을 알아봅니다.

일반적으로 IT 서적에는 전문 용어가 많이 포함돼 있어 입문자 입장에서는 어렵게 느껴질 수 있습니다. 저 역시 비전공자여서 그 심정을 충분히 이해합니다. 그래서 이 책에서는 누구나 쉽게 이해할 수 있도록 쉬운 표현과 비유, 삽화를 넣어 설명하고, 전문 용어는 한 줄로 정리하는 코너를 마련해 개념을 명확히 짚고 넘어갑니다. 또한 각 장의 마지막에 퀴즈를 실어 배운 내용을 점검할 수 있습니다. 이 책을 통해 일상생활과 업무에 도움이 되는 IT 지식을 갖춰 성과를 향상하기 바랍니다.

민완기

## 베타 리더의 한마디

복잡한 IT 개념을 쉽게 설명해줘 부담 없이 읽을 수 있었고, IT 지식을 이해하는 데 도움이 됐습니다. 이론을 실무와 연관 지어 설명하기 때문에 IT 기술이 실무에 어떻게 적용되는지 감을 잡을 수 있습니다. 평소 IT에 대해 막연한 두려움이 있었지만 이 책을 읽고 나서 IT에 대해 더 알고 싶어졌습니다.

**정석진**

비전공자 입장에서 웹과 앱, 라이브러리와 프레임워크 등 평소에 궁금하면서도 헷갈렸던 주제를 정확히 이해할 수 있게 됐습니다. 전체적으로 쉬우면서도 내용에 깊이가 있어 유익합니다. 개발자와의 소통을 다룬 부분에서는 개발자와 비개발자의 소통이 왜 어려운지, 개발자에게 요청하려면 어떻게 해야 하는지를 구체적인 예시를 통해 이해할 수 있었습니다.

**김수정**

그동안 불명확하게 사용하던 IT 개념을 이 책으로 정리할 수 있었습니다. 왜 그렇게 해야만 하는지에 대한 이유도 명확히 설명돼 있어 좋았습니다. 개발의 전체적인 구조를 파악하는 데 도움이 됐고, 개발자와 비개발자의 관점에서 모두 설명돼 있어 양측의 입장을 다각도로 이해할 수 있었습니다.

**이유진**

이 책은 ‘개발자는 무슨 일을 할까?’라는 궁금증에 대한 답변을 담고 있습니다. 다양한 개발 직무를 주요 프로그래밍 언어와 함께 소개하고 IT의 기초 개념을 쉽게 설명합니다. 생생한 사례를 통해 개발자가 하는 일을 보다 쉽게 이해할 수 있었고, 개발자들이 쓰는 용어와 표현에 담긴 속사정을 알게 됨으로써 개발 세계에 한 걸음 더 다가갈 수 있었습니다.

**임승민**



IT 기초 지식부터 현업에서 알아야 세세한 내용까지 잘 정리돼 있어 기획자, 디자이너, 웹 퍼블리셔, 개발자 등 모든 IT 직군에게 도움이 될 것입니다. 개발은 여러 직군이 모여 협업하는 일이라 커뮤니케이션이 매우 중요하고, 각각의 업무 특성상 다른 직군이 하는 말을 100% 이해하기가 쉽지 않은데, 이 책은 그러한 어려움을 해결해줍니다. 어릴 적 백과사전을 수시로 꺼내 보던 것처럼 앞으로 계속 꺼내 볼 것 같습니다.

**유나연**

이 책은 각 장을 시작할 때 학습할 내용을 키워드로 미리 소개하고 마지막에 서는 한 줄로 다시 요약해 학습 효과를 높입니다. 개발자와 협업하는 비개발자에게 실질적인 도움을 줄 뿐만 아니라, 개발자 입장에서는 비개발 직군과 소통할 때 유의할 점을 간접적으로 알게 됩니다. IT 기획자, 마케터는 물론이고 비전공자나 개발에 관심이 있는 학생도 실무에 대한 이해를 높이는 데 도움이 될 것입니다.

**이지은**

이 책을 읽고 나서 뜻도 모른 채 익숙하기만 했던 IT 용어의 의미를 정확히 알게 됐습니다. 주요 개발 언어가 어떤 배경에서 생겨났는지, 어떤 특징이 있는지, 동향이 어떤지에 대한 설명 덕분에 앞으로 어떤 프로그래밍 언어를 공부해야 할지 계획하는 데 큰 도움을 받았습니다. 소통은 상호 이해를 위한 노력을 전제로 합니다. 개발자와 협업하는 독자에게 이 책은 그러한 노력을 위한 가장 효율적인 도구가 될 것입니다.

**김진욱**

## 베타 리더의 한마디



현재 클라우드 서비스 개발 과정을 수료 중인 저에게 매우 유익한 책이었습니다. SQL과 NoSQL의 이론적인 내용이 잘 정리돼 있어 각 데이터베이스의 유형과 활용 방법을 명확히 이해할 수 있었고, 서드파티 API와 맞춤형 API에 대해 잘 설명돼 있어 개발에 어떻게 활용되는지 알 수 있었습니다. 취업 준비생으로서 실무 역량을 강화하는 데 많은 도움을 얻었습니다.

**이성희**

처음에는 단순히 IT 입문자용 도서로 알고 읽기 시작했습니다. 적절한 비유와 예시를 들면서 설명해줘 이해하기 쉽다고 느끼다가, 읽으면 읽을수록 제가 가지고 있던 지식이 책의 내용과 시너지를 이루며 더 선명하게 흡수되는 느낌을 받았습니다. 책 제목처럼 아는 만큼 보이고 재미있는 주제라는 생각이 듭니다.

**김동현**

이 책은 비개발자의 눈높이에 맞춰 IT 지식을 쉽게 설명해줘 개발자인 저도 많은 팁을 얻었습니다. 알맞은 비유를 들어 개발 용어를 소개하고, 특정 상황에서 비개발자와 개발자가 어떻게 소통해야 하는지 예시를 들어 설명한 점이 특히 인상 깊었습니다. 협업을 잘하고 싶은 개발자와 비개발자 모두에게 도움이 되는 책입니다.

**김효진**



## 이 책의 구성



이 책은 IT 업계에서 일하는 비전공자를 위해 업무에 필요한 IT 지식 및 개발자와 소통하는 법을 알려줍니다. IT 회사에서 일하는 마케터, 기획자뿐만 아니라 IT 회사에 취업하려고 준비 중인 문과생, 외주 개발사와 소통하는 회사 대표, 개발에 관심을 가진 학생, IT 지식이 부족하다고 느끼는 독자에게 도움이 됩니다.

이 책은 총 2개 파트, 8개 장으로 구성돼 있습니다. 파트 1에서는 웹과 앱을 중심으로 IT 기초 지식을 쌓고, 파트 2에서는 개발자와 소통하는 데 필요한 개발 용어 및 개발자와의 소통 방법을 배웁니다. 책을 읽고 나면 IT 서비스의 전반적인 동작과 개발자가 하는 일을 이해할 수 있습니다.

### · 1 · 이것만 알면 된다! IT 기초 지식

- 인터넷의 동작 원리
- 인터넷에서 데이터를 안전하게 지키는 법
- 웹과 앱 이해하기
- 데이터베이스 이해하기



### · 2 · 개발자와 소통하기

- 개발 분야별 프로그래밍 언어
- 알아두면 좋은 개발 용어
- 안 된다고 하는 개발자의 속사정
- 개발자와 잘 소통하는 법

\_이 책의 그림에 사용된 'Bazzi' 폰트는 무료로 제공되는 넥슨 배찌체로, 해당 폰트의 저작권은 (주)넥슨 코리아에 있습니다.

## 1장 · 인터넷의 동작 원리

|                             |    |
|-----------------------------|----|
| <b>1.1 클라이언트-서버 구조</b>      | 20 |
| 1.1.1 클라이언트와 서버의 개념         | 20 |
| 1.1.2 서버에서 데이터를 받아오는 이유     | 22 |
| 1.1.3 서버의 역할                | 23 |
| <b>1.2 프론트엔드 개발과 백엔드 개발</b> | 25 |
| 1.2.1 프론트엔드 개발의 개념          | 25 |
| 1.2.2 백엔드 개발의 개념            | 28 |
| 1.2.3 개발 범위 정리              | 30 |
| <b>1.3 IP 주소와 도메인 주소</b>    | 32 |
| 1.3.1 IP 주소의 개념             | 32 |
| 1.3.2 고정 IP 주소와 유동 IP 주소    | 33 |
| 1.3.3 도메인 주소의 개념            | 35 |
| <b>1.4 인터넷 통신 과정</b>        | 38 |
| 1.4.1 TCP/IP 모델             | 38 |
| 1.4.2 HTTP 통신               | 41 |
| 1.4.3 웹 소켓                  | 42 |
| 1.4.4 라우팅                   | 43 |
| <b>1.5 인터넷 오류 메시지 보는 법</b>  | 45 |
| 1.5.1 주요 HTTP 상태 코드         | 46 |
| 1.5.2 HTTP 상태 코드별 개발자와의 소통법 | 47 |
| <b>정리하기 퀴즈</b>              | 55 |

## 2장 · 인터넷에서 데이터를 안전하게 지키는 법

|                        |    |
|------------------------|----|
| <b>2.1 방화벽을 세우는 이유</b> | 58 |
| 2.1.1 방화벽의 개념          | 58 |



|                        |    |
|------------------------|----|
| 2.1.2 방화벽의 형태          | 59 |
| 2.1.3 내부망과 프록시 서버      | 59 |
| 2.1.4 DMZ              | 60 |
| <b>2.2 암호화 기술</b>      | 63 |
| 2.2.1 암호화의 개념          | 63 |
| 2.2.2 암호화 알고리즘         | 64 |
| 2.2.3 일방향 암호화와 양방향 암호화 | 65 |
| 2.2.4 암호화를 적용한 네트워크 기술 | 67 |
| <b>정리하기 퀴즈</b>         | 72 |

### 3장 • 웹과 앱 이해하기

|                         |    |
|-------------------------|----|
| <b>3.1 웹의 개념과 발전 과정</b> | 76 |
| 3.1.1 웹의 개념             | 76 |
| 3.1.2 웹의 발전 과정          | 77 |
| 3.1.3 반응형 웹             | 81 |
| <b>3.2 앱의 개념과 종류</b>    | 84 |
| 3.2.1 앱의 개념             | 84 |
| 3.2.2 웹 앱               | 84 |
| 3.2.3 모바일 앱             | 88 |
| <b>정리하기 퀴즈</b>          | 95 |

### 4장 • 데이터베이스 이해하기

|                           |     |
|---------------------------|-----|
| <b>4.1 데이터베이스</b>         | 98  |
| 4.1.1 데이터베이스의 개념          | 98  |
| 4.1.2 데이터 저장 방식           | 100 |
| 4.1.3 데이터베이스의 특징          | 102 |
| 4.1.4 트랜잭션                | 107 |
| <b>4.2 데이터베이스 언어: SQL</b> | 109 |
| 4.2.1 SQL의 개념             | 109 |

|  |     |
|--|-----|
| 4.2.2 SQL 문 알아보기                       | 110 |
| <b>4.3 데이터베이스 관리 시스템: RDBMS와 NoSQL</b> | 116 |
| 4.3.1 RDBMS                            | 116 |
| 4.3.2 NoSQL                            | 121 |
| 4.3.3 RDBMS와 NoSQL 정리                  | 128 |
| <b>정리하기 퀴즈</b>                         | 130 |

## PART 2 개발자와 소통하기

...

### 5장 · 개발 분야별 프로그래밍 언어

|   |     |
|---|-----|
| <b>5.1 인공지능 개발과 빅데이터 분석 언어: 파이썬</b>       | 134 |
| 5.1.1 입문용으로 파이썬이 좋은 이유                    | 134 |
| 5.1.2 파이썬 인터프리터                           | 136 |
| 5.1.3 프로그래밍 언어 버전의 중요성                    | 138 |
| <b>5.2 프론트엔드 개발 언어: HTML, CSS, 자바스크립트</b> | 142 |
| 5.2.1 HTML                                | 142 |
| 5.2.2 CSS                                 | 150 |
| 5.2.3 자바스크립트                              | 153 |
| <b>5.3 백엔드 개발 언어: 자바, 코틀린, PHP</b>        | 157 |
| 5.3.1 자바                                  | 157 |
| 5.3.2 코틀린                                 | 163 |
| 5.3.3 PHP                                 | 164 |
| <b>5.4 모바일 앱 개발 언어: 자바, 코틀린, 스위프트</b>     | 169 |
| 5.4.1 안드로이드 앱 개발: 자바, 코틀린                 | 169 |
| 5.4.2 iOS 앱 개발: 스위프트                      | 173 |
| <b>5.5 크로스 플랫폼 개발 언어: 닷넷, 자바스크립트</b>      | 175 |
| 5.5.1 닷넷                                  | 175 |
| 5.5.2 자바스크립트                              | 179 |
| 5.5.3 크로스 플랫폼 개발 언어 선택 방법                 | 184 |
| <b>정리하기 퀴즈</b>                            | 186 |

## 6장 • 알아두면 좋은 개발 용어

|                                 |     |
|---------------------------------|-----|
| <b>6.1 자주 쓰는 코드의 모음: 라이브러리</b>  | 190 |
| 6.1.1 라이브러리의 개념                 | 190 |
| 6.1.2 라이브러리의 적합성 판단 기준          | 191 |
| 6.1.3 라이브러리 관련 에피소드             | 194 |
| 6.1.4 라이브러리 라이선스                | 195 |
| <b>6.2 개발을 도와주는 도구: 프레임워크</b>   | 197 |
| 6.2.1 프레임워크의 개념                 | 197 |
| 6.2.2 풀 리퀘스트와 머지                | 199 |
| 6.2.3 개발 분야별 라이브러리/프레임워크        | 199 |
| <b>6.3 소프트웨어 개발 키트: SDK</b>     | 201 |
| 6.3.1 SDK의 개념                   | 201 |
| <b>6.4 서비스 인터페이스: API</b>       | 204 |
| 6.4.1 API의 개념                   | 204 |
| 6.4.2 API 설계: REST API          | 206 |
| 6.4.3 API 인증과 토큰                | 208 |
| 6.4.4 API의 이용 유형                | 209 |
| <b>6.5 구조화된 데이터 포맷: JSON</b>    | 212 |
| 6.5.1 JSON 데이터 표기법              | 212 |
| 6.5.2 JSON을 활용한 API 통신 과정       | 213 |
| <b>6.6 사이트맵과 RSS 작성 언어: XML</b> | 219 |
| 6.6.1 XML 데이터 표기법               | 219 |
| 6.6.2 XML 사용 분야                 | 221 |
| <b>정리하기 퀴즈</b>                  | 225 |

## 7장 • 안 된다고 하는 개발자의 속사정

|                            |     |
|----------------------------|-----|
| <b>7.1 업무 상황별 담당자 찾기</b>   | 230 |
| 7.1.1 웹 페이지의 데이터를 수정할 때    | 230 |
| 7.1.2 웹 페이지 표시 속도가 느릴 때    | 235 |
| 7.1.3 모바일 앱 화면 표시 속도가 느릴 때 | 240 |
| 7.1.4 화면에 나와야 할 것이 안 나올 때  | 242 |

|                                       |     |
|---------------------------------------|-----|
| <b>7.2 개발자가 안 된다고 하는 진짜 이유</b>        | 248 |
| 7.2.1 새 기능 개발이 안 된다고 하는 이유            | 248 |
| 7.2.2 효율성과 이익이 납득된 기능 개발도 안 된다고 하는 이유 | 250 |
| 7.2.3 기존 기능 수정이 안 된다고 하는 이유           | 252 |
| 7.2.4 기존 기능 삭제가 안 된다고 하는 이유           | 256 |
| 7.2.5 개발자가 안 된다고 하는 진짜 이유 정리          | 258 |
| <br><b>8장 · 개발자와 잘 소통하는 법</b>         |     |
| <b>8.1 요구 사항을 문서로 정리하기</b>            | 262 |
| 8.1.1 사용자 시나리오 작성하기                   | 263 |
| 8.1.2 기능별 입출력 작성하기                    | 265 |
| 8.1.3 비기능적 요구 사항 작성하기                 | 267 |
| 8.1.4 코드 번호 달고 마무리하기                  | 270 |
| <b>8.2 못 알아들었는데 어물쩍 넘어가지 않기</b>       | 273 |
| 8.2.1 회의 자료 만들어 논의하기                  | 273 |
| 8.2.2 재차 물어볼 때의 요령                    | 274 |
| <b>8.3 개발이 불가능하면 원인을 찾아 방법 바꾸기</b>    | 276 |
| 8.3.1 기술력과 자금이 부족한 경우                 | 276 |
| 8.3.2 로직 변경이 어려운 경우                   | 277 |
| 8.3.3 데이터 간에 연결점이 없는 경우               | 278 |
| 8.3.4 자격이 없어서 불가능한 경우                 | 279 |
| <b>8.4 상황에 맞게 문서 버전 업데이트하고 공유하기</b>   | 281 |
| 8.4.1 문서 버전 표기하는 법                    | 282 |
| 8.4.2 문서 변경 이력 관리하는 법                 | 283 |
| <b>8.5 정확한 데이터로 말하기</b>               | 285 |
| 8.5.1 정확하게 요청 범위 말하기                  | 285 |
| 8.5.2 모호한 표현 쓰지 않기                    | 286 |
| <br><b>정리하기 퀴즈 정답</b>                 | 288 |
| <b>찾아보기</b>                           | 290 |

PART

1

이것만  
알면 된다!  
IT 기초 지식



CHAPTER

# 1

## 인터넷의 동작 원리

1.1 클라이언트-서버 구조

1.2 프론트엔드 개발과 백엔드 개발

1.3 IP 주소와 도메인 주소

1.4 인터넷 통신 과정

1.5 인터넷 오류 메시지 보는 법

정리하기 퀴즈



인터넷은 어떤 구조로 동작하고, 개발자는 인터넷에서 돌아가는 서비스를 어떻게 만들까요?  
오류가 날 때 웹 페이지에 뜨는 404, 500과 같은 숫자는 무엇을 의미할까요? 이 장에서는  
인터넷이 어떤 구조로 동작하는지, 개발자가 인터넷에서 어떤 일을 하는지 알아봅니다.

이 장을 읽고 나면 다음 키워드를 이해할 수 있습니다.

클라이언트 / 서버 / 프론트엔드 개발 / 백엔드 개발 / IP 주소 / 도메인 주소 / TCP/IP /  
HTTP / 웹 소켓 / HTTP 상태 코드



## 클라이언트-서버 구조

일상생활에서 매일 보는 유튜브, 카카오톡, 네이버 웹툰 같은 서비스를 이용하려면 보고, 듣고, 만질 수 있는 컴퓨터나 스마트폰이 있어야 합니다. 이러한 장치를 통틀어 ‘클라이언트’라고 합니다.

그런데 클라이언트만 있다고 해서 원하는 콘텐츠를 즐길 수 있는 것은 아닙니다. 클라이언트를 통해 이용하는 콘텐츠는 대부분 ‘서버’에서 가져와야 합니다. 그렇다면 클라이언트와 서버가 무엇이고 어떤 구조로 동작하는지 알아봅시다.

### 1.1.1 클라이언트와 서버의 개념

클라이언트(client)는 사용자가 원하는 작업을 하기 위해 직접 손으로 조작하는 컴퓨터 장치 또는 응용 프로그램을 말합니다.

- **컴퓨터 장치:** 데스크톱, 노트북, 스마트폰 등의 전자 기기를 말합니다.
- **응용 프로그램:** 특정 작업을 하기 위해 사용하는 프로그램을 말합니다. 인터넷에 접속할 때 사용하는 **웹 브라우저**(web browser)와 카카오톡, 인스타그램, 유튜브 등의 **애플리케이션**(application)이 있습니다.

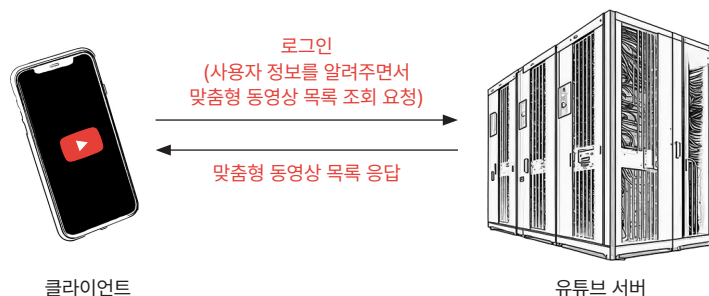
그림 1-1 클라이언트의 종류



클라이언트의 종류가 많아서 헷갈린다면 ‘클라이언트는 손님이다’라고 생각하세요. 컴퓨터 장치나 응용 프로그램은 모두 사용자가 뭔가를 요청하기 위해 이용하기 때문에 클라이언트를 손님처럼 생각하면 쉽게 이해할 수 있습니다. 다시 말해 클라이언트는 서비스를 이용하는 데 필요한 데이터를 서버에 요청하고, 서버로부터 응답받은 데이터를 화면에 표시합니다.

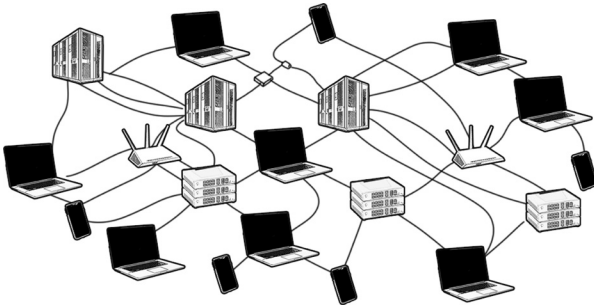
**서버(server)**는 클라이언트의 요청을 받아 처리한 후 응답하는 컴퓨터 장치 또는 그러한 역할을 하는 응용 프로그램을 말합니다. 예를 들어 스마트폰으로 유튜브 앱에 로그인하면 자신의 취향에 맞는 동영상 목록이 뜹니다. 이는 서버에 로그인하면서 “나는 누구인데 내 취향에 맞는 동영상 목록을 보여줘”라고 요청했기 때문입니다.

그림 1-2 클라이언트-서버 간 통신



이렇게 클라이언트와 서버가 서로 통신하는 구조를 **클라이언트-서버 구조**라고 합니다. 클라이언트와 서버는 네트워크에 연결돼 서로 데이터를 주고받습니다. 여기서 **네트워크(network)**는 각종 컴퓨터 장치가 두 대 이상 그물망처럼 연결돼 서로 통신할 수 있도록 만든 체계를 말합니다.

그림 1-3 네트워크



## 1.1.2 서버에서 데이터를 받아오는 이유

그런데 클라이언트는 왜 서버에서 데이터를 받아올까요? 클라이언트가 자체적으로 데이터를 가지고 있다면 서버의 응답을 기다릴 필요 없이 사용자가 요청한 데이터를 바로 보여줄 수 있지 않을까요?

클라이언트가 서버에서 데이터를 받아오는 것은 저장 장치의 용량 문제와 보안 문제 때문입니다. 개인이 사용하는 클라이언트에 모든 데이터와 이를 처리하기 위한 프로그램을 저장하기에는 용량의 한계가 있습니다. 유튜브 동영상처럼 방대한 데이터와 프로그램을 컴퓨터나 스마트폰에 다 저장할 수는 없습니다. 그래서 서버에 프로그램과 데이터를 저장해두고 필요할 때 데이터를 요청해 사용합니다.

또한 민감한 개인 정보를 클라이언트에 가지고 있으면 문제가 됩니다. 만약 기기를 잃어버리거나 보안을 소홀히 하면 개인 정보가 유출될 수도 있

습니다. 따라서 서버에 데이터를 저장하고 있다가 데이터 전송 시 각종 암호화 기술이 적용된 보안 네트워크를 통해 데이터를 받아와 사용하는 것입니다.

### 1.1.3 서버의 역할

서버의 역할을 이해하기 위해 유튜브에 접속해 동영상을 보는 과정을 예로 살펴보겠습니다.

- ❶ 크롬 브라우저를 실행합니다.
- ❷ 유튜브에 접속합니다.
- ❸ 마음에 드는 썸네일을 클릭합니다.
- ❹ [재생] 버튼을 눌러 동영상을 시청합니다.

❶번은 유튜브 동영상을 보기 위해 클라이언트(크롬 브라우저)를 준비하는 것입니다.

❷번은 클라이언트가 서버에 접속하는 과정으로, 클라이언트는 서버에 “나는 누구인데 내 취향에 맞는 동영상 목록을 보여줘”라고 **조회 요청**을 합니다. 요청을 받은 서버는 사용자의 취향에 맞는 동영상으로 응답하고, 클라이언트는 이를 받아 화면에 표시합니다.

❸번에서 흥미를 끄는 동영상 썸네일을 클릭하는 것도 마찬가지로 “해당 동영상의 제목, 설명글, 댓글 목록 등 상세 정보를 보여줘”라고 서버에 **조회 요청**을 하는 것입니다.

그리고 ❹번에서 [재생] 버튼을 누르는 것은 동영상을 끊김 없이 보기 위해 **조회 요청**을 하는 것입니다. 즉 스트리밍(streaming)으로 동영상을 보여달라고 요청하는 것입니다. 스트리밍이란 큰 데이터를 잘게 쪼개 연속으로 보냄

으로써 동영상을 실시간으로 끊김 없이 볼 수 있게 하는 데이터 전송 방식을 말합니다.

이처럼 클라이언트가 서버에 원하는 데이터를 보여달라고 조회 요청을 하면 서버는 그에 따라 데이터로 응답합니다.

한편 클라이언트가 서버에 보내는 요청에는 조회 요청뿐만 아니라 저장 요청, 수정 요청, 삭제 요청도 있습니다. 예를 들어 유튜브 동영상을 보다가 화면을 이탈하거나 [정지] 버튼을 누르면 서버에 동영상의 마지막 시청 시점을 저장하라는 **저장 요청**이 전송됩니다. 그래서 다음에 동영상을 재생할 때 어디까지 봤는지 알 수 있습니다. 저장 요청은 어떤 채널의 [구독] 버튼을 누르거나, 댓글을 작성한 후 [저장] 버튼을 누를 때도 발생합니다.

**수정 요청**은 댓글을 수정할 때를 생각하면 쉽게 이해할 수 있습니다. 댓글을 수정하고 [수정] 버튼을 누르면 서버에 수정 요청이 전송되고 댓글이 수정됩니다. 또한 마음에 들지 않는 채널의 구독을 취소할 때는 서버에 **삭제 요청**을 보냅니다.

이처럼 서버는 클라이언트의 요청에 따라 데이터 **저장(Create)**, **조회(Read)**, **수정(Update)**, **삭제>Delete** 작업을 수행하며, 이를 약자로 **CRUD**라고 합니다.

#### 한 줄 정리



- **클라이언트**: 서버에 데이터를 요청하고 제공받는 컴퓨터 장치 또는 응용 프로그램을 말합니다.
- **서버**: 클라이언트의 요청을 받아 처리한 후 응답하는 컴퓨터 장치 또는 응용 프로그램을 말합니다.
- **서버에서 데이터를 받아오는 이유**: 클라이언트에 모든 데이터와 프로그램을 저장하기에는 용량의 한계가 있고, 보안상 데이터를 안전하게 지키기 위해서입니다.
- **서버의 역할**: 클라이언트의 요청에 따라 데이터 저장, 조회, 수정, 삭제 작업을 수행합니다.



## 프론트엔드 개발과 백엔드 개발

클라이언트와 서버는 역할이 다르기 때문에 한 개발자가 둘 다 배워 개발하기보다 따로 개발해야 생산성을 높일 수 있습니다. 그래서 웹 개발은 클라이언트를 개발하는 프론트엔드 개발과 서버를 개발하는 백엔드 개발로 나뉩니다.

### 1.2.1 프론트엔드 개발의 개념

**프론트엔드 개발**(front-end development)이란 눈으로 볼 수 있고 손으로 조작할 수 있는 화면을 개발하는 것을 말합니다. 쉽게 말해 웹 사이트와 모바일 앱 화면을 개발하는 것입니다. 이는 눈에 보이는 영역인 앞단(front-end)을 개발한다는 의미에서 프론트엔드 개발이라고 불리게 되었습니다.

자동차의 핸들처럼 어떤 기능을 쉽게 이용할 수 있도록 도와주는 장치를 보통 **인터페이스**(interface)라고 합니다. 인터페이스는 각기 다른 장치가 소통할 수 있도록 도와주는 매개체 역할을 합니다. 웹 개발에서 화면에 보이는 그래픽 요소, 즉 화면을 구성하는 각종 입력창, 버튼, 체크박스, 스크롤바처럼 사람과 컴퓨터가 소통하게 해주는 인터페이스를 **사용자 인터페이스**(UI, User Interface)라고 합니다.

프론트엔드 개발자는 디자이너로부터 각종 인터페이스가 배치된 웹/앱 디

자인 시안을 전달받아 개발 가능 여부를 논의합니다. 그리고 실제로 각 인터페이스가 화면에 나타나게 만든 후 그 기능을 구현합니다(서버와 통신해 데이터를 받아오고 화면에 보여주는 것까지 모두 구현합니다).

그럼 프론트엔드 개발자는 사용자 인터페이스를 개발하기 위해 어떤 프로그래밍 언어를 사용할까요? 일반적으로 HTML, CSS, 자바스크립트를 사용합니다. 이 세 가지 언어에 대해서는 **5.2 프론트엔드 개발 언어: HTML, CSS, 자바스크립트**에서 자세히 알아보겠습니다.

재미있는 사실은, 이 세 가지 언어만 알면 **웹 퍼블리셔(web publisher)**라는 직업을 가질 수 있다는 것입니다. 자신이 몸담고 있는 회사에 웹 퍼블리셔가 있다면 지금부터 설명할 내용이 업무 요청에 도움이 될 것입니다. 웹 퍼블리셔의 역할이 무엇이고, 프론트엔드 개발자와 어떤 차이가 있는지 알아보겠습니다.

### 웹 퍼블리셔의 역할

웹 퍼블리셔는 프론트엔드 개발자가 하는 업무의 일부를 수행합니다. 자동차 제작에 비유하자면 내비게이션 시스템, 자동 주차 기능, 엔진 등 복잡한 시스템 장치를 다루는 것이 아니라 자동차의 형태, 색상, 질감을 결정하고 운전석, 시트, 대시보드 등 외관과 내장을 아름답고 편리하게 만드는 일을 한다고 볼 수 있습니다.

웹 사이트를 개발할 때는 HTML 5라는 웹 표준을 따라야 합니다. 또한 지능정보화 기본법에 따라 의무적으로 웹 접근성(web accessibility)을 준수해야 합니다. 웹 퍼블리셔는 이러한 규격에 맞게 웹 화면을 만듭니다. 예컨대 다음과 같은 기능을 구현함으로써 장애인이나 노인도 차별 없이 웹 사이트의 정보를 제공받을 수 있게 합니다.

- **스크린 리더 기능:** 시각장애인 또는 저시력자를 위해 텍스트를 음성으로 읽어주는 기능입니다.



- **화면 확대 기능:** 시각장애인 또는 저시력자를 위해 작은 글씨를 크게 보여주는 기능입니다.
- **음성 인식 기능:** 키보드를 사용할 수 없는 사람을 위해 음성으로 검색하는 것을 돕는 기능입니다.
- **키보드 오버레이 기능:** 어떤 키가 입력되고 있는지 화면에 보여주는 기능입니다.

웹 퍼블리셔는 이 밖에도 색맹을 위해 화면에 데이터를 표시할 때 색상만으로 데이터를 구분하면 안 된다거나, 콘텐츠를 쉽게 식별할 수 있도록 텍스트 및 이미지와 배경 간 명도가 4.5:1 이상이어야 한다는 지침을 준수합니다.

웹 퍼블리셔가 수행하는 웹 퍼블리싱은 프론트엔드 개발자가 사용자 인터페이스를 개발하는 것보다 상대적으로 난이도는 낮지만 업무량이 많습니다. 웹 퍼블리셔와 프론트엔드 개발자의 업무는 다음과 같이 요약할 수 있습니다.

#### ● 웹 퍼블리셔의 업무

- HTML 5 웹 표준과 장애인 및 노인도 이용할 수 있도록 웹 접근성을 준수해 웹 화면을 만듭니다.
- 클라이언트(사용하는 컴퓨터 장치 또는 웹 브라우저)마다 동일한 디자인으로 보이도록 웹 화면을 만듭니다.

#### ● 프론트엔드 개발자의 업무

- 웹 퍼블리셔의 업무를 똑같이 수행합니다.
- 웹 사이트 기획에 따라 서버와 통신하는 기능을 구현합니다.

실무에서는 웹 퍼블리셔가 디자인 시안을 받아 웹 표준과 접근성에 부합하게 정리한 후 최종 디자인 시안을 확정하고, 이를 토대로 HTML, CSS, 자바스크립트로 웹 페이지의 뼈대를 만듭니다. 그러면 그 뼈대 위에 프론트엔드

개발자가 기능을 개발하는 방식으로 일합니다. 웹 퍼블리셔는 서버와 소통하는 코드를 작성하지 않는데, 이는 웹 퍼블리셔와 프론트엔드 개발자의 뚜렷한 차이라고 할 수 있습니다.

자금이 부족한 스타트업에서는 웹 퍼블리셔 없이 프론트엔드 개발자만 채용하는 경우가 많습니다. 프론트엔드 개발에 웹 퍼블리싱이 포함되기 때문입니다.

## 1.2.2 백엔드 개발의 개념

**백엔드 개발**(back-end development)이란 서버가 클라이언트의 요청을 받아 처리하는 프로그램을 개발하는 것을 말합니다. 사용자의 눈에 보이지 않는 영역인 뒷단(back-end)을 개발한다는 의미에서 백엔드 개발이라고 불리게 됩니다.

서버가 무슨 일을 하는지 레스토랑에 비유해 살펴보겠습니다.

- ① 손님이 메뉴판을 보고 웨이터에게 음식을 주문합니다.
- ② 주문을 받은 웨이터는 주방장에게 주문서를 전달합니다.
- ③ 주방장이 음식을 만들어 웨이터에게 건넵니다.
- ④ 웨이터는 손님에게 음식을 서빙(serving)하고, 손님은 음식을 맛있게 먹습니다.

이를 클라이언트-서버 구조에서 이뤄지는 웹 서비스의 동작으로 나타내면 다음과 같습니다.

- ① 클라이언트(손님)가 사용자 인터페이스(메뉴판)를 통해 웹 서버(웨이터)에 데이터(음식)를 요청합니다.

- ❷ 웹 서버(웨이터)는 웹 애플리케이션 서버(주방장)에 클라이언트의 요청 사항(주문서)을 전달합니다.
- ❸ 웹 애플리케이션 서버(주방장)는 요청받은 데이터(음식)를 처리한 후 결과를 웹 서버(웨이터)에 전달합니다. 이때 데이터의 저장과 처리는 데이터베이스(냉장고)가 담당합니다.
- ❹ 웹 서버(웨이터)는 클라이언트(손님)에게 결과를 전달하고, 클라이언트는 이를 화면에 표시합니다.

그림 1-4 웹 서비스의 동작

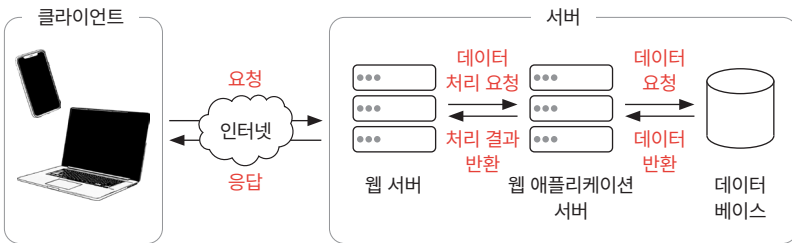


그림 1-4에서 웹 서버(web server)는 클라이언트의 요청을 받아 비교적 간단히 처리할 수 있는 응답은 바로 하고, 처리하기 어려운 요청은 웹 애플리케이션 서버(web application server)로 보냅니다. 웹 애플리케이션 서버는 웹 서버로부터 받은 복잡한 요청을 처리한 후 이를 웹 서버에 반환합니다.

백엔드 개발자는 웹 서비스가 잘 돌아가도록 웹 서버와 웹 애플리케이션 서버를 개발하는 사람입니다. 웨이터와 주방장이 손님의 주문을 잘 처리하도록 관리하고 교육하는 매니저와 같은 존재라고 볼 수 있죠. 따라서 백엔드 개발자는 네트워크, 데이터베이스, 인프라에 대한 지식을 갖추고 있어야 합니다.

- **네트워크 지식:** 네트워크에 연결된 클라이언트와 서버 간에 어떤 방식(프로토콜)으로 소통하는지 알고, 이를 구현합니다.

- **데이터베이스 지식:** 클라이언트가 요청한 데이터를 처리하기 위해 데이터베이스를 구축하고 데이터를 저장·관리합니다.
- **인프라 지식:** 서버를 운영하는 데 필요한 하드웨어와 소프트웨어를 관리합니다. 예를 들어 클라이언트의 요청이 서버가 감당하기 어려울 정도로 증가하면 서버 장비를 증설하는 등 인프라를 관리합니다.

백엔드 개발자는 ‘A라는 상황에서는 B를 하라’고 서버에 하나씩 정해줍니다. 상황과 요청에 맞는 임무를 수행하도록 꼼꼼히 지시를 내리지 않으면 서버가 엉뚱하게 처리하거나 같은 작업을 반복할 수도 있습니다. 매니저가 매뉴얼을 세심하게 정해주지 않아 웨이터와 주방장이 갈팡질팡하면 그만큼 시간과 비용이 낭비되는 것과 마찬가지입니다.

한편 직원들의 동선을 효율적으로 짜고 업무를 최적화하면 100명이 할 일을 50명이 할 수도 있습니다. 즉 백엔드 개발자가 어떻게 하느냐에 따라 서버 비용(회사의 지출)이 줄어들 수도 있고, 낭비될 수도 있습니다. 그래서 백엔드 개발자는 프론트엔드 개발자에 비해 성과를 측정하기가 쉽습니다.

### 1.2.3 개발 범위 정리

지금까지 프론트엔드 개발과 백엔드 개발에 대해 살펴보았습니다. 프론트엔드 개발자는 웹 사이트나 모바일 앱 화면의 사용자 인터페이스를 통해 서버에 데이터를 요청하는 **프레젠테이션 로직**(presentation logic)을 개발합니다. 그리고 백엔드 개발자는 클라이언트의 요청을 받아 처리하는 서버의 **비즈니스 로직**(business logic)을 개발합니다.

**그림 1-5**를 토대로 자신이 회사에서 개발자에게 요청하려는 업무가 프론트엔드 개발에 관한 것인지, 백엔드 개발에 관한 것인지 구분할 수 있을 것입니다.



## 정리하기 퀴즈



### 1. 다음 각각의 설명이 맞으면 ○, 틀리면 ×로 표시하세요.

- ① 클라이언트는 사용자가 원하는 작업을 하기 위해 직접 손으로 조작하는 컴퓨터 장치 또는 응용 프로그램을 말한다. .... ( )
- ② 데이터를 클라이언트가 아니라 서버에 저장하는 것은, 개인 기기에 방대한 데이터를 저장하기에는 한계가 있고 보안 측면에서도 서버에 보관하는 것이 더 안전하기 때문이다. .... ( )
- ③ 백엔드 개발자는 프레젠테이션 로직을 개발하는 사람이다. .... ( )
- ④ 서버나 클라이언트가 사용하는 IP 주소는 항상 고정돼 있다. .... ( )
- ⑤ 서버에 요청하지 않고도 데이터를 실시간으로 전달받고 싶으면 웹 소켓 통신을 이용하면 된다. .... ( )
- ⑥ TCP/IP 모델로 데이터를 주고받을 때는 최종 목적지에 직통으로 데이터를 보낸다. .... ( )

### 2. 다음 각각의 상황에 맞는 HTTP 상태 코드를 쓰세요.

- ① 요청한 웹 페이지를 찾을 수 없을 때 .... ( )
- ② 서버에 너무 많이 요청했을 때 .... ( )
- ③ 서버 내부에 오류가 발생했을 때 .... ( )
- ④ 로그인에 필요한 요청인데 로그인을 하지 않았을 때 .... ( )

▶ 정답 288쪽

IT 용어를 못 알아들겠어요. 코딩을 배워야 할까요?

## 낮선 IT 기술과 개발 용어를 한번에 정리하는 올인원 기술 교양서

전공자가 아닌 이상 IT 용어는 생소합니다. 잘 몰라서 검색해봐도 어려운 용어로 설명돼 있으니 금새 찾아보는 걸 포기하게 되고, 기획한 서비스가 구현 가능한지 판단이 서질 않아 개발자에게 자신 있게 말하지 못하는 경우도 있습니다.

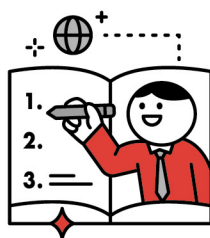
이 책은 IT 업계에서 일하는 비전공자를 위해 업무에 필요한 IT 지식과 개발자와 소통하는 법을 알려줍니다. 웹/앱 마케터 및 기획자, IT 계열 취업 준비생, 개발자에 관심 있는 학생, IT 지식이 부족하다고 느끼는 분들이 보면 좋습니다. 책을 읽고 나면 IT 서비스의 전반적인 동작과 개발자가 하는 일을 이해할 수 있습니다.

### 1. 이것만 알면 된다! IT 기초 지식: 인터넷의 동작 원리 / 인터넷에서 데이터를

안전하게 지키는 법 / 웹과 앱 이해하기 / 데이터베이스 이해하기

### 2. 개발자와 소통하기: 개발 분야별 프로그래밍 언어 / 알아두면 좋은 개발 용어

/ 안 된다고 하는 개발자의 속사정 / 개발자와 잘 소통하는 법



아는 만큼 보이는 IT 지식

Easy to Learn IT Knowledge  
and Development Terms

정가 22,000원



ISBN 979-11-407-1141-3