

# 컴퓨터 프로그래밍 기초 용어 정리

## 1. 프로그래밍 기본 용어

### 프로그램(Program)

컴퓨터에게 일을 시키기 위한 명령어들의 모음.

우리가 컴퓨터에게 원하는 일을 시킬 때, 그 일에 필요한 명령어들을 모아 놓은 것.

게임, 웹사이트서치, 문서를 작성 등의 모든 활동은 프로그램임.

### 프로그래밍 언어(Programming Language)

프로그램을 만들 때 사용되는 언어.

컴퓨터가 이해할 수 있는 언어로 명령을 전달하기 위해 쓰임.

대표적인 프로그래밍 언어 : JavaScript, Python, Java, C++ 등.

### 코드(Code)

프로그램을 만들 때 사용하는 언어로 쓰인 글. 프로그램을 만들려면 우리가 이해할 수 있는

언어로 명령을 내려야 하는데, 이를 컴퓨터가 이해할 수 있는 언어로 바꾼 것이 코드.

### 변수(Variable)

하나의 값(숫자나 글자)을 저장하는 메모리 공간. 변수에는 어떤 값이든 넣을 수 있고, 그 값은 나중에 바꿀 수도 있음.

### 함수(Function)

특정한 작업을 수행하는 코드의 모음.

자주 사용하는 작업들을 함수로 묶어놓고, 그 함수를 호출해서 사용함.

### Algorithm (알고리즘)

문제를 해결하기 위한 명확하고 구체적인 절차나 방법.

(비유: 요리 레시피 - 재료를 언제, 어떻게 썰고 볶아야 하는지 순서대로 알려준다.)

### API (Application Programming Interface, 응용 프로그램 인터페이스)

소프트웨어 간 상호작용을 위한 규칙과 도구들의 집합.

(비유: 레스토랑의 메뉴판 - 주방장이 무엇을 제공할 수 있는지 명확하게 알려준다.)

### Bug (버그)

프로그램이 예상과 다르게 동작하거나 오류를 발생시키는 문제. (비유: 기계 안에 몰래 숨어든 말썹꾸러기 벌레.)

### Compiler (컴파일러)

사람이 이해할 수 있는 고급 언어를 컴퓨터가 이해할 수 있는 기계어로 번역해 주는 프로그

램.(비유: 인간어를 로봇어로 동시통역하는 똑똑한 통역사.)

### Debugging (디버깅)

프로그램의 오류를 찾아 수정하는 과정.

(비유: 숨은 그림 찾기 - 틀린 부분을 찾아내는 집요함이 필요하다.)

### Framework (프레임워크)

개발을 빠르고 체계적으로 할 수 있도록 제공하는 뼈대와 도구 모음.

(비유: 집을 지을 때 사용하는 기본 골조 - 그 위에 방을 만들든 벽지를 바르든 마음껏!)

### IDE (Integrated Development Environment, 통합 개발 환경)

코드 작성, 디버깅, 컴파일, 실행 등을 하나의 환경에서 할 수 있게 해주는 소프트웨어.

(비유: 만능공구세트 - 드라이버, 망치, 톱이 한 세트에 다 들어있는 느낌.)

### Library (라이브러리)

특정 기능을 수행하는 코드의 모음.

(비유: 미리 만들어둔 요리 재료 - 직접 만들 필요 없이 꺼내 쓰기만 하면 된다.)

### Open Source (오픈소스)

소스 코드를 누구나 열람, 수정, 배포할 수 있도록 공개한 소프트웨어.

(비유: 레시피를 공짜로 공유하는 푸드 블로거.)

### Version Control System (버전 관리 시스템)

코드의 변경 이력을 기록하고 관리하는 시스템. 대표적으로 Git이 있다.

(비유: 문서 '저장' 버튼의 진화 버전 - 잘못되면 언제든지 과거로 돌아갈 수 있다.)

### Object-Oriented Programming (OOP)

객체(데이터+기능)를 중심으로 프로그램을 설계하는 방법론. (비유: 레고 블록을 조립하듯 프로그램을 짠다.)

### Database (DB)

대량의 데이터를 체계적으로 저장하고 관리하는 시스템.

### Server / Client

서버는 요청을 받아 처리하고, 클라이언트는 요청을 보내는 주체.

### Frontend / Backend

프론트엔드는 사용자에게 보이는 부분, 백엔드는 보이지 않는 서버 처리 부분.

### Cloud Computing

인터넷을 통해 서버, 스토리지, 데이터베이스 등을 제공하는 기술.

(비유: 내 컴퓨터가 아니라 '구름 속' 컴퓨터를 빌려 쓰는 것.)

## 웹 개발(Web Development) 주요 용어 정리

HTML (HyperText Markup Language)

웹페이지의 구조를 작성하는 언어.

(비유: 건물의 철골 구조 - 형태는 만들지만, 색이나 기능은 아니다.)

CSS (Cascading Style Sheets)

웹페이지의 스타일(색, 글꼴, 레이아웃 등)을 지정하는 언어.

(비유: 철골 구조 위에 바르는 페인트와 인테리어.)

JavaScript (JS)

웹페이지에 동적인 기능을 추가하는 프로그래밍 언어.

(비유: 정적인 집에 조명 스위치나 자동문을 설치하는 것.)

Responsive Web Design (반응형 웹 디자인)

화면 크기에 따라 자동으로 레이아웃이 조정되는 웹 디자인 기법.

Frontend (프론트엔드)

사용자가 직접 보고, 상호작용하는 웹사이트의 화면 부분.

Backend (백엔드)

서버, 데이터베이스 등 사용자가 보지 못하는 웹사이트의 기능 부분.

(비유: 식당 주방 - 맛있는 요리가 완성되는 곳.)

HTTP (HyperText Transfer Protocol)

웹 브라우저와 서버가 정보를 주고받는 약속된 통신 규칙.

(비유: 손님과 웨이터가 사용하는 공식 주문서.)

HTTPS (HyperText Transfer Protocol Secure)

HTTP에 보안을 추가한 버전. 데이터가 암호화되어 전송된다.

(비유: 평범한 택배가 아니라 보안 봉투에 담긴 비밀 택배.)

REST API (Representational State Transfer API)

웹의 자원을 이름으로 구분하고, HTTP 방식(GET, POST 등)으로 자원에 접근하는 설계 방식.

DOM (Document Object Model)

웹페이지를 트리 구조로 표현한 모델. JS를 통해 DOM을 조작하면 페이지 내용을 실시간으로 바꿀 수 있다. (비유: 웹페이지를 나무로 본다면, DOM은 가지 하나하나를 직접 만지는 것.)

Framework (프레임워크) Angular, React, Vue.js 같은 도구로, 웹개발을 체계적으로 빠르게 도와준다. (비유: 골조와 기본 설계도가 포함된 집짓기 세트.)

Library (라이브러리)

필요한 기능을 추가하는 도구 모음. (예: jQuery, Lodash)

CDN (Content Delivery Network)

전 세계 서버에 파일을 저장해 사용자에게 빠르게 제공하는 시스템. (비유: 동네마다 분점이 있는 맛집.)

SEO (Search Engine Optimization)

검색 엔진에서 웹사이트를 상위에 노출되게 최적화하는 작업. (비유: 바다 속에 숨은 진주를 구글에게 확성기로 알리는 것.)

Web Hosting (웹 호스팅) 웹사이트를 인터넷에 올려 누구나 접근할 수 있도록 저장 공간을 제공하는 서비스.

SSL Certificate (SSL 인증서)

웹사이트의 통신을 암호화하고 보안성을 증명하는 전자 인증서. (비유: "이 가게는 믿을 수 있습니다" 인증 마크.)

요약

-HTML = 뼈대, CSS = 옷 입히기, JS = 움직임

-프론트엔드는 사용자와 맞는 부분, 백엔드는 보이지 않는 작동 원리

-HTTP/HTTPS는 소통 방법, REST API는 주문과 배달 체계

웹 개발은 단순히 "멋진 화면을 만든다"가 아니라,

정보를 잘 전달하고, 빠르고, 안전하게, 효율적으로 제공해야한다.

# AI/머신러닝 주요 용어 정리

Artificial Intelligence (AI, 인공지능)

인간의 지능적 행동(학습, 추론, 문제 해결 등)을 컴퓨터가 모방하는 기술.  
(비유: 로봇이 "이 상황에서는 이렇게 행동해야지!"를 스스로 생각하는 것.)

Machine Learning (ML, 머신러닝)

명시적 프로그래밍 없이 데이터로부터 스스로 학습하는 기술.  
(비유: 처음에는 서툴지만, 경험을 통해 점점 농구를 잘하게 되는 로봇.)

Deep Learning (DL, 딥러닝)

인간의 뇌 신경망을 모방한 '인공 신경망(ANN)'을 사용해 학습하는 머신러닝의 한 분야.  
(비유: 뇌세포를 본떠 만든 초고속 학습 시스템.)

Neural Network (신경망)

수많은 노드(뉴런)를 연결해 입력 데이터를 처리하고 학습하는 구조.  
(비유: 수천 개의 전구를 켜고 끄면서 패턴을 맞추는 퍼즐 놀이.)

Supervised Learning (지도 학습)

정답이 포함된 데이터를 기반으로 학습하는 방법.  
(비유: 선생님이 "이건 고양이야!"라고 알려주면서 사진을 보여주는 것.)

Unsupervised Learning (비지도 학습)

정답 없이 데이터의 패턴이나 구조를 스스로 찾아내는 학습 방법.  
(비유: 외국어를 아무 설명 없이 듣고, 문법과 단어를 스스로 터득하는 것.)

Reinforcement Learning (강화 학습)

행동에 따른 보상이나 벌을 통해 학습하는 방법.  
(비유: 개가 간식을 받기 위해 "손!"을 배워가는 과정.)

Overfitting (과적합)

학습 데이터에는 정확하지만, 새로운 데이터에는 잘 작동하지 않는 현상.  
(비유: 기출 문제만 달달 외워서, 시험에 응용 문제가 나오면 망하는 경우.)

Underfitting (과소적합)

학습 자체가 부족해서 데이터의 패턴을 제대로 파악하지 못하는 현상.

Model (모델)

학습된 규칙과 패턴을 바탕으로 예측이나 분류를 수행하는 결과물.  
(비유: 세상의 법칙을 이해하고 문제를 푸는 '머신의 두뇌'.)

Feature (특징) 데이터 안의 의미 있는 속성이나 변수. (비유: 사람의 키, 몸무게, 나이 같은 것.)

Label (레이블) 지도 학습에서 정답 역할을 하는 값. (비유: "이 사진은 고양이다.")

Training Data (학습 데이터)

모델을 훈련시키는 데 사용하는 데이터.

Validation Data (검증 데이터) 모델의 성능을 튜닝하기 위해 사용하는 별도의 데이터.

Test Data (테스트 데이터)

Loss Function (손실 함수)

학습이 끝난 후 모델 성능을 최종 평가하는 데이터.

모델의 예측이 정답과 얼마나 다른지를 측정하는 함수. (비유: 정답과 오답 사이 거리 측정기.)

Gradient Descent (경사 하강법) 손실을 최소화하기 위해 모델을 조금씩 조정하는 최적화 방법. (비유: 산을 내려가면서 가장 낮은 골짜기를 찾는 것.)

Epoch (에포크) 전체 학습 데이터를 한 번 다 사용하는 학습 주기. (비유: 헬스장에서 한 번 운동 루틴을 다 돌고 오는 것.)

Hyperparameter (하이퍼파라미터) 학습 전에 설정해야 하는 변수(예: 학습률, 신경망 층 수 등).

Transfer Learning (전이 학습) 기존에 학습한 모델의 일부를 새로운 문제에 적용하는 방법. (비유: 영어를 배운 사람이 프랑스어를 더 빨리 배우는 것.)

요약

AI는 '지능 모방', ML은 '경험으로 배우기', DL은 '뇌처럼 배우기'

지도/비지도/강화 학습은 학습 방식의 차이

과적합/과소적합은 모델이 데이터를 얼마나 잘 이해했는지를 나타내는 문제

Feature, Loss, Gradient 같은 단어들은 '모델이 더 똑똑해지기 위한 핵심 부품'들