



# 컴퓨터 개론

비고	
예상 출시일	@Jan 31, 2020

## 코스 요약

---

### 수강 기간

4주

### 선수 과목

없음

### 수업 목표

- 초심자가 프로그래밍 세계에 입문할 수 있는 가이드 역할을 한다.
- 어떤 프로그래밍 언어가 있고, 어떤 프로그램이 있는지, 프로그래밍 세계의 큰 그림을 이해한다.
- 올바르고 효율적인 공부 방향을 제시한다.
- 컴퓨터의 구조, 소프트웨어 등 프로그래밍에 있어 기본적인 개념들을 이해한다.

### 수업 구성

동영상 + 노트 + 퀴즈 + 실습 과제

### 사용 기술

파이썬

### 수강 대상

- 개발자들과 수월하게 소통하고 싶은 분
- 개발자로서의 커리어를 꿈꾸는 분

## 강사

---



## 최지웅

Instructor / Content Producer

- 동국대학교 수학교육과 전공
- 에이릭스 교육연구소 전임연구원
- KBS ㄱㄷㄱㄷ 코딩TV 스토리가 살아있는 코딩교육(아트테마, 게임테마) 집필
- 아주대학교 SW 가치확산프로그래밍 기획 및 운영



## 손동욱

Content Director

- 고려대학교 전기전자 공학부/심화 전공
- 상명대학교 뮤직테크놀로지학과 석사 과정
- 코드잇 Content Director
- LG전자 Software Engineer

## 커리큘럼 요약

### 토픽 1: 파이썬 프로그래밍 시작하기

#### ▼ Chapter 1: 파이썬 첫 걸음

- 파이썬: 쉽고 간결한 프로그래밍 언어
- 파이썬은 어디에 쓰이나요?

#### ▼ Chapter 2: 프로그래밍 기본 개념

- 코멘트: 내 코드에 노트 달기
- 자료형 개요: 숫자형, 문자열, 불린형 등
- 추상화 개요: 변수와 함수
- return문: 함수의 결괏값 활용하기

## 토픽 2: 프로그래밍 언어 이해하기

### ▼ Chapter 1: 프로그래밍 언어 살펴보기

- "컴퓨터 개론" 코스 소개 영상 (프로그래밍을 배우기 힘든 이유)
- "프로그래밍 언어 이해하기" 토픽 소개
- 프로그래밍 언어란?
- 프로그래밍 언어의 생김새

### ▼ Chapter 2: 프로그래밍 언어의 다양성

- 프로그래밍 언어가 너무 많아요!
- 첫 번째 패러다임: 객체 지향 프로그래밍
- 객체 지향 프로그래밍 이해하기
- 객체 지향 프로그래밍 정리하기
- 두 번째 패러다임: 변수의 데이터 타입
- Dynamic typing vs. Static typing
- 프로그래밍 언어의 스펙트럼
- 저수준 언어와 고수준 언어 이해하기
- 저수준 언어와 고수준 언어 비교하기
- 프로그래밍 언어의 종류 정리하기
- 특수 목적 언어

### ▼ Chapter 3: 프로그래밍 언어 제대로 사용하기

- 좋은 코드란?
- 읽기 쉬운 코드
- 스타일 가이드 적용하기
- 구조화가 잘 된 코드
- 라이브러리
- 프레임워크
- 남의 코드에서 배우기

## 토픽 3: 프로그래머의 세계 이해하기

#### ▼ Chapter 1: 프로그래밍의 다양한 분야

- “프로그래머의 세계 이해하기” 토픽 소개 (로드맵의 필요성)
- 컴퓨터 사이언스
- 소프트웨어 공학
- 프로세스 관리
- 테스트 프로세스
- 버전 관리
- 개발자들이 많이 쓰는 프로그램

#### ▼ Chapter 2: 트렌드를 읽는 능력

- 프로그래밍은 선택의 연속
- 트렌드를 읽는 방법 1: 활발한 기술
- 트렌드를 읽는 방법 2: 인정받는 기술
- 트렌드를 읽는 방법 3: 유망한 기술
- 트렌드를 읽는 방법 (정리 노트)
- 꿀팁 노트: 프로그래머에게 영어는 필수인가요?

#### ▼ Chapter 3: 실무 이야기

- 스타트업이 일하는 방식과 대기업에서 일하는 방식
- 개발자와 함께 일하는 사람들
- 현직자 인터뷰 1 (스타트업 개발자)
- 현직자 인터뷰 2 (대기업 개발자)
- 현직자 인터뷰 3
- 현직자 인터뷰 4 (디자이너가 바라본 개발자)

## 토픽 4: 소프트웨어 이해하기

#### ▼ Chapter 1: 컴파일러와 인터프리터

- “소프트웨어 이해하기” 토픽 소개 (보이는 프로그램과 보이지 않는 프로그램)
- 다양한 애플리케이션
- 애플리케이션을 만드는 프로그램

- 프로그래밍 언어의 번역기, 컴파일러
- 실시간 코드 실행기, 인터프리터
- 컴파일러 vs. 인터프리터 정리 노트
- 언어의 종류에 따른 실행 방식
- 꿀팁 노트: 첫 번째 컴파일러는 어떻게 만들었을까?

#### ▼ Chapter 2: 운영 체제 개요

- 애플리케이션을 도와주는 프로그램, 운영 체제
- 운영 체제의 역할 1: 입력과 출력
- 운영 체제의 역할 2: 저장과 불러오기
- 운영 체제의 역할 3: 여러 프로그램 동시에 빠르게 실행하기
- 운영 체제의 종류
- 운영 체제의 발전
- 꿀팁 영상: 리눅스 이해하기

#### ▼ Chapter 3: 네이티브 애플리케이션과 웹 애플리케이션

- 플랫폼의 파편화
- 표준의 필요성
- 크로스 플랫폼과 가상 머신
- 네이티브 앱과 웹 앱
- 웹을 활용하는 여러 가지 방법
- 웹의 전망
- 웹 표준과 웹 브라우저

## 토픽 5: 컴퓨터 이해하기

#### ▼ Chapter 1: 컴퓨터 들여다보기

- "컴퓨터 이해하기" 토픽 소개 (컴퓨터의 구조, 꼭 알아야 할까?)
- 컴퓨터 내부 살펴보기
- 꿀팁 노트: HDD와 SSD

#### ▼ Chapter 2: 컴퓨터의 언어, 0과 1

- 이진법을 알아야 하는 이유
- 0과 1의 힘
- 두 손으로 1000까지 세는 법
- 0과 1로 숫자를 표현하는 방법
- 이진법으로 변환하기
- 이진법과 십진법 퀴즈
- 이진법의 덧셈
- 0과 1로 문자를 표현하는 방법
- 16진법
- 16진법 퀴즈
- 비트와 바이트

#### ▼ Chapter 3: 칩의 기본 구조

- 0과 1의 전달 방법
- 논리 연산 개요
- 일곱 가지 논리 연산 pt. 1
- 일곱 가지 논리 연산 pt. 2
- 일곱 가지 논리 연산 정리 노트
- 나의 첫 번째 회로 설계
- 회로의 추상화
- 여러 자리 덧셈
- 칩 조립하기

#### ▼ Chapter 4: CPU

- 프로그램을 실행하는 칩, CPU
- 하나의 칩에 여러 기능 넣기
- CPU와 저장 장치 I
- CPU와 저장 장치 II
- 머신 코드 실행하기

- 꿀팁 영상: CPU 패밀리
- 컴퓨터의 추상화
- 어셈블리 언어
- 32비트 시스템과 64비트 시스템

▼ Chapter 5: 컴퓨터의 퍼포먼스

- 컴퓨터의 속도
- CPU 성능 지표
- 프로그램을 빠르게 실행하는 두 가지 방법
- 프로그램 빠르게 실행하기 I: Instruction 개수 줄이기
- 프로그램 빠르게 실행하기 II: Instruction 실행 속도 올리기
- 더 작게, 더 빠르게!