

중학생 맞춤형 컴퓨팅 과 프로그래밍

2025 Summer

Overview

단순히 코드를 작성하는 것을 넘어, Computational Thinking 기반의 문제 해결 능력과 체계적인 알고리즘 설계 능력을 함양하는 것을 목표로 합니다. 중학생의 인지 발달 단계를 고려한 맞춤형 커리큘럼을 통해, 컴퓨팅 과학의 기본 원리를 깊이 있게 이해하고 실질적인 프로그래밍 능력을 배양합니다.

Objectives

실습 중심의 몰입형 학습 경험

- 학습자의 능동적인 참여와 흥미 유발을 최우선으로 설계
- 이론적 배경과 Hands-on 실습을 병행하여 학습 효과 극대화
- 실제 작동하는 소프트웨어를 구현하는 경험을 통해 자기 주도 학습 능력 향상

핵심 도구: Python 프로그래밍 언어

- 높은 가독성과 간결한 문법 구조를 특징으로 하는 Python을 주요 교육 언어로 채택
- 데이터 과학, 인공지능, 웹 개발 등 다양한 기술 분야에서 활용되는 범용적인 언어 학습
- 학습자가 미래 기술 변화에 유연하게 적응할 수 있는 기반 제공

전문적인 교육 목표:

- 컴퓨팅 과학의 핵심 개념 (정보, 데이터, 알고리즘)에 대한 정확한 이해 확립
- 변수, 자료형, 제어 구조 (조건문, 반복문) 등 프로그래밍의 기본 문법 체계 심층 학습
- 함수를 활용한 코드 모듈화 및 재사용성 증대 기법 체계적 습득
- 표준 라이브러리 및 외부 라이브러리 활용을 통한 효율적인 소프트웨어 개발 능력 배양
- 간단한 코드 분석을 통해 프로그램의 작동 원리 및 논리적 흐름에 대한 통찰력 확보
- 소프트웨어 개발 생명 주기 (SDLC)에 대한 기본적인 이해 구축

Structure

1. 컴퓨팅 과학 개론 및 개발 환경
 - 컴퓨팅 과학의 정의와 응용
 - 정보, 데이터, 알고리즘의 관계
 - 하드웨어/소프트웨어 구조
 - 프로그래밍 패러다임과 Python 특징
2. 기본 자료형 및 연산자
 - 자료형 (Integer, Float, String, Boolean) 특징
 - 변수와 메모리 할당
 - 연산자 (산술, 비교, 논리) 활용
3. 조건문으로 제어 흐름 구현
 - Boolean 논리와 연산
 - if, elif, else 사용법
 - 중첩/복합 조건 처리
4. 반복문으로 자동화
 - for 루프와 range() 활용
 - List 구조와 슬라이싱
 - while 루프와 종료 조건
 - break/continue로 흐름 제어
5. 함수로 코드 모듈화
 - 함수 정의와 호출
 - 매개변수와 반환 값
 - 내장 함수 활용
 - 사용자 정의 함수 설계
6. 표준 라이브러리 활용
 - 모듈과 패키지 개념
 - import 사용법
 - random, math, time 활용
 - 외부 라이브러리 기초
7. 코드 분석 및 디버깅
 - 코드 실행 흐름 추적
 - 오류 유형과 해결법
 - print()로 디버깅
 - 복잡도 개념 이해
8. 소프트웨어 개발과 프로젝트
 - 간단한 프로젝트 아이디어 구상