

KECLA Coding Class

General Information

Term: Fall 2026

Audience: Grades 5-12

Course Description:

This 15-week introductory course is designed for students in grades 5–12 with no prior coding experience. Through Python Turtle, students will explore what coding is and gain hands-on experience with how the games they enjoy are created. By learning fundamental programming concepts through fun and creative activities, students will develop the skills and confidence to design and build their own simple games. The course provides a valuable opportunity to learn practical coding skills that students may not have many opportunities to experience in a traditional school setting.

Course Objectives:

By the end of the course, students will be able to:

- Understand basic programming concepts and Python syntax.
- Use Turtle to create drawings, shapes, and patterns.
- Use loops, variables, conditionals, and functions in Python programs.
- Use randomness and colors to create dynamic projects.
- Create interactive programs using keyboard events.
- Develop simple games with score, lives, and game mechanics.
- Plan, create, debug, and present an original Python Turtle project.

Week	Contents	Activities
1	Introduction to Coding & Python • What is coding? • What is Python?	
2	Introduction to Python Turtle • Running a simple Python program • Turtle shapes and size • Colors and pen settings	Create a simple Turtle drawing using lines, points, colors, and basic shapes.
3	Drawing with Turtle • Combining Turtle commands and coordinates • Using shapes, colors, and basic geometric shapes • Planning and creating a drawing step by step	Create a simple scene by drawing a house, tree, garden, and car using Turtle.
4	Loops: Introduction to Repetition • Understanding repetition and iteration • Using for loops and range() • Repeating Turtle commands with loops	Create repeated lines and shapes using loops.
5	Patterns & Shapes • Using loops with angles, left(), and right()	Create stars, flowers, spirals, and original patterns using loops and angles.

	• Creating geometric patterns by changing repetition counts and angles • Exploring how small code changes create different results	
6	Variables & Calculations • Understanding and using variables • Performing basic calculations with numbers • Using variables to control Turtle size, distance, and angles	Create a drawing that can be resized or modified using variables.
7	Colors & Randomness • Understanding randomness and using the random module • Generating random numbers, positions, and RGB colors • Creating dynamic and unpredictable drawings	Draw Korean letters using Python Turtle to celebrate Hangul Day.
8	If Statements • Understanding conditions and making decisions with if, elif, and else • Using comparison operators and Boolean logic • Changing Turtle behavior based on conditions	Create a program where the Turtle reacts differently to different conditions
9	Functions • Understanding functions and creating them with def • Calling functions and using parameters • Using functions to organize and reuse code	Create reusable functions such as draw_square() and draw_tree()
10	Events I: Keyboard Input • Understanding events and user interaction • Using onkey() and listen() for keyboard input • Connecting keyboard keys to functions	Make the Turtle move when the Space key is pressed
11	Events II: Arrow Key Controls • Using arrow keys to control the Turtle • Combining keyboard events with functions • Building a simple interactive program with directional movement	Control the Turtle using the arrow keys
12	Game Mechanics • Using variables, loops, if statements, functions, and events to build a game • Creating Game Over conditions and a complete game experience	Create a simple Turtle game
13	Final Project Planning • Plan new Shapes, colors, movements, and game elements to add to the program created in Week 12 • Add score, lives, collision detection, and game rules • Begin developing the project	Create a Final Project Plan and start coding

14	Final Project Development Implement the ideas planned in Week 13 using the concepts learned in class.	Develop and refine a final project using Python Turtle.
15	Final Project Presentation - Complete the project and conduct final testing - Demonstrate the project and share the first coding experience	Present and play your original Python Turtle project

LA 한국 교육원 코딩 수업

수업 정보

학기: 2026 년 가을학기

대상: 5-12 학년

수업 소개

이 1 주 입문 과정은 코딩 경험이 전혀 없는 5-12 학년 학생들을 대상으로 합니다. 학생들은 Python Turtle 을 활용하여 코딩이 무엇인지 배우고, 평소 즐겨 하는 게임이 어떻게 만들어지는지 직접 경험하게 됩니다. 재미있고 창의적인 활동을 통해 프로그래밍의 기본 개념을 배우면서, 학생들은 자신만의 간단한 게임을 직접 설계하고 제작할 수 있는 능력과 자신감을 키우게 됩니다. 또한 이 과정은 일반적인 학교 수업에서는 접하기 어려운 실용적인 코딩 경험을 직접 해볼 수 있는 소중한 기회를 제공합니다.

수업 목표

강의가 끝날 때까지 학생들은 다음을 할 수 있게 됩니다.

- 프로그래밍의 기본 개념과 Python 문법을 이해할 수 있다.
- Turtle 을 활용하여 그림, 도형, 패턴을 만들 수 있다.
- Python 프로그램에서 반복문, 변수, 조건문, 함수를 사용할 수 있다.
- 랜덤 기능과 색상을 활용하여 다양한 프로젝트를 만들 수 있다.
- 키보드 이벤트를 활용하여 사용자와 상호작용하는 프로그램을 만들 수 있다.
- 점수, 목숨(Lives) 등의 게임 요소를 포함한 간단한 게임을 만들 수 있다.
- 자신만의 Python Turtle 프로젝트를 기획하고, 제작하고, 디버깅하고, 발표할 수 있다.

주차	수업 내용	수업 활동
1	코딩과 Python 소개 • 코딩이란 무엇인가? • Python 이란 무엇인가?	Python 코딩 환경 설치하기
2	Python Turtle 소개 • 간단한 Python 프로그램 실행하기 • Turtle 의 모양과 크기 설정하기 • 색상과 펜 설정하기	선, 점, 색상, 기본 도형을 활용하여 간단한 Turtle 그림 만들기

3	Turtle 을 활용한 그림 그리기 • Turtle 명령어와 좌표를 조합하여 활용하기 • 도형과 색상을 활용하여 기본적인 그림 그리기 • 그림을 단계별로 계획하고 완성하기	Turtle 을 활용하여 집, 나무, 정원, 자동차가 있는 장면 그리기
4	반복문 – 반복의 이해 • 반복과 반복 실행의 개념 이해하기 • for 반복문과 range() 사용하기 • 반복문을 활용하여 Turtle 명령어 반복하기	반복문을 활용하여 반복되는 선과 도형 만들기
5	패턴과 도형 만들기 • 반복문과 각도, left(), right() 활용하기 • 반복 횟수와 각도를 변경하여 다양한 기하학적 패턴 만들기 • 코드의 작은 변화가 결과에 미치는 영향 탐색하기	반복문과 각도를 활용하여 별, 꽃, 나선형 및 나만의 패턴 만들기
6	변수와 계산 • 변수의 개념 이해하고 활용하기 • 숫자를 이용한 기본적인 계산하기 • 변수를 활용하여 Turtle 의 크기, 이동 거리, 각도 조절하기	변수를 활용하여 크기나 모양을 변경할 수 있는 그림 만들기
7	색상과 랜덤 기능 • 랜덤의 개념과 random 모듈 활용하기 • 랜덤한 숫자, 위치, RGB 색상 생성하기 • 랜덤 기능을 활용하여 다양한 그림과 패턴 만들기	한글날을 기념하여 Python Turtle 로 한글 글자 그리기
8	조건문 • 조건에 따라 판단하고 실행하는 방법 이해하기 • if, elif, else 와 비교 연산자 활용하기 • 조건에 따라 Turtle 의 행동을 다르게 설정하기	조건에 따라 Turtle 이 다르게 반응하는 프로그램 만들기
9	함수 • 함수의 개념을 이해하고 def 를 사용하여 함수 만들기 • 함수를 호출하고 매개변수 활용하기 • 함수를 활용하여 코드를 정리하고 재사용하기	draw_square(), draw_tree()와 같은 재사용 가능한 함수 만들기

10	이벤트 I – 키보드 입력 - 이벤트와 사용자 상호작용 이해하기 - onkey()와 listen()을 활용하여 키보드 입력 처리하기 - 키보드 입력과 함수를 연결하기	스페이스바를 누르면 Turtle 이 움직이는 프로그램 만들기
11	이벤트 II – 방향키 조작 - 방향키를 활용하여 Turtle 조작하기 - 키보드 이벤트와 함수 결합하기 - 방향키로 Turtle 을 움직이는 간단한 프로그램 만들기	방향키를 활용하여 Turtle 조작하기
12	게임 요소와 게임 만들기 - 변수, 반복문, 조건문, 함수, 이벤트를 활용하여 게임 만들기 - Game Over 조건을 추가한 게임 구성하기	간단한 Turtle 게임 만들기
13	최종 프로젝트 기획 12 주 수업에서 만든 프로그램에 새로운 Shape, 색상, 움직임, 게임 요소 등을 추가할 내용 계획하기 - 점수, 목숨(Lives), 충돌 감지, 게임 규칙 추가하기 - 프로젝트 개발 시작하기	최종 프로젝트 계획을 세우고 코딩 시작하기
14	최종 프로젝트 개발 13 주차에 기획한 내용들을 수업에서 배운 내용을 활용하여 구현하기	Python Turtle 을 활용하여 최종 프로젝트를 개발하고 완성하기
15	최종 프로젝트 발표 - 프로젝트를 완성하고 최종 테스트하기 - 프로젝트를 시연하기	자신만의 Python Turtle 프로젝트를 발표하고 직접 시연하기