CodeHS

Introduction to Python Programming 1 Course Syllabus
One Semester for High School, 60 Hours

Course Overview and Goals

Introduction to Python Programming 1 introduces students to the fundamentals of computer programming, with an emphasis on helping students develop logical thinking and problem-solving skills. Students will learn to design, code, and test their programs while applying mathematical concepts. Once students complete this course, they will have learned material equivalent to the first half of a semester college introductory course in computer science and be able to program basic Python 3 programs.

Learning Environment: The course utilizes a blended classroom approach. The content is fully web-based, with students writing and running code in the browser. Teachers utilize tools and resources provided by CodeHS to leverage time in the classroom and give focused 1-on-1 attention to students. Each unit of the course is broken down into lessons. Lessons consist of video tutorials, short quizzes, example programs to explore, and written programming exercises, adding up to over 60 hours of hands-on programming practice in total. Each unit ends with a comprehensive unit test that assesses students' mastery of the material from that unit as well as unit projects where students can display their understanding of the material.

Programming Environment: Students write and run Python programs in the browser using the CodeHS editor.

More information: Browse the content of this course at https://codehs.com/course/21075/overview

Prerequisites: The Introduction to Python Programming 1 course is designed for complete beginners with no previous background in computer science. The course is highly visual, dynamic, and interactive, making it engaging for new coders.

Course Breakdown

Unit 1: Karel in Python (3 weeks/15 hours)
Students learn the basics of programming by giving Karel the Dog commands in a grid world.

Browse the full content of this unit at https://codehs.com/course/21075/explore/module/29304

Objectives / Topics Covered	- Commands - Defining vs. Calling Functions - Designing Functions - Control Flow - Looping - Conditionals - Commenting Code - Top Down Design
Assignments / Labs	- Program-specific tasks for Karel the Dog - Example Exercise: Pyramid of Karel Write a program to have Karel build a pyramid. There should be three balls on the first row, two in the second row, and one in the third row.

- Teach Karel new commands like `turnRight()` or `makePancakes()`
  - Example Exercise: Pancakes
    Karel is the waiter. He needs to deliver a stack of pancakes to the guests on the 2nd, 4th, and 6th columns. Each stack of pancakes should have three pancakes. Create a function called `makePancakes()` to help Karel solve this problem. The world should end up exactly as shown here.

- Solve large Karel problems by breaking them down into smaller, more manageable problems using Top Down Design
  - Example Exercise: The Two Towers
    In this program, Karel should build two towers of tennis balls. Each tower should be 3 tennis balls high. In the end, Karel should end up on top of the second tower, facing East.

- Using control structures and conditionals to solve general problems
  - Example Exercise: Random Hurdles
    Write a program that has Karel run to the other side of the first row, jumping over all of the hurdles. Karel should only jump if there is a hurdle blocking the way. However, the hurdles can be in random locations. The world is fourteen columns long. You must write a function named `jumpHurdle()` as part of your solution.

Unit 2: Basic Python and Console Interaction (3 weeks/15 hours)

Students learn the basics of programming by writing programs that interact with users through the keyboard.

Browse the full content of this unit at https://codehs.com/course/21075/explore/module/29305

Objectives / Topics Covered	- Printing - Variables - Types - User Input - Converting Input Types - Arithmetic Expressions - String Operators - Comments
Example Assignments / Labs	- Printing - Print messages to the console - Variables - Create variables of different types, and print them to the console. - Types - Investigate the types of different variables - Convert between types - Arithmetic Expressions & Converting Input Types - Age in One Year - Ask the user how old they are, and tell them how old they will be in one year - Rectangle, part 1 - Make variables for length and width and compute area and perimeter - Rectangle, part 2 - Ask the user for length and width and compute area and perimeter

Unit 3: Project: Mad Libs (2 day/2 hours)

Students write a program that allows users to create their own Mad Lib stories using variables and user input.

Browse the full content of this unit at https://codehs.com/course/21075/explore/module/29306

Objectives / Topics Covered	- Variables - User Input - Print Statements
Example Assignments / Labs	Create a Mad Libs game where users can input different types of words to complete a story. The program should prompt the user for various types of words (e.g., noun, adjective, verb) and then construct a story using the provided words.

Unit 4: Conditionals (2 weeks/10 hours)

Students teach their programs to make decisions based on the information it receives.

Browse the full content of this unit at https://codehs.com/course/21075/explore/module/29307

Objectives / Topics Covered	- If Statements - Boolean Values - Logical Operators - Comparison Operators - Floating Point Numbers and "Equality"
Example Assignments / Labs	- If statements and boolean values - Is it raining? - Write a program that uses a boolean variable to determine whether or not it is raining - Boolean operators, and expressions - Boolean variable - Take a variable and use it in an if statement - Legally allowed to vote - User reports age and the program tells them whether or not they can vote in the US - Transaction - The user reports balance and deposit/withdrawal, and the program prints a new balance or error - Recipe - Ask the user for ingredients, amounts per serving, and number of servings, and report the total amount of each ingredient needed

Unit 5: Project: Quiz Game (2 days/2 hours)

Students write a program that quizzes users on a series of multiple-choice questions and checks their answers.

Browse the full content of this unit at https://codehs.com/course/21075/explore/module/29308

Objectives / Topics Covered	- Variables - Boolean Logic - Arithmetic Operators - Conditionals - Comparison Operators
Project Description	Create a simple multiple-choice quiz game using Python. The program should ask the user a series of questions and check their answers. After all the questions have been answered, the program should display the final score.

Unit 6: Looping (2 weeks/10 hours)

Students learn how to write more efficient code by using loops as shortcuts.

Browse the full content of this unit at https://codehs.com/course/21075/explore/module/29309

Objectives / Topics Covered	- While Loops - For Loops - Break and Continue - Nested Control Structures
Example Assignments / Labs	- While Loops - Divisibility - Ask the user to enter a numerator and denominator, and re-prompt until the denominator is non-zero - For Loops - Average test score - Compute the average of several test scores - Break and Continue - Higher/ Lower - Ask the user to guess a particular number between 1 and 100. If the user's guess was too high or too low, they should be notified - Nested Control Structures - Rolling Dice - Print out all combinations that can be made when 2 dice are rolled

Unit 7: Project: Password Authenticator (2 day/2 hours)

Students write a program to provide feedback on whether the entered password is correct or incorrect.

Browse the full content of this unit at https://codehs.com/course/21075/explore/module/29310

Objectives / Topics Covered	- Variables - User Input - Boolean Operations - Loops - Control Structures
Project Description	Create a password authentication program that prompts the user to enter a password. The program should compare the entered password with a predefined correct password and provide feedback on whether the entered password is correct or incorrect. The program should allow the user to try again a limited number of times before locking them out.

Unit 8: Roles in a Software Development Team (1 week/5-6 hours)

Students learn the key roles and responsibilities of members of a software development team.

Browse the full content of this unit at https://codehs.com/course/21075/explore/module/29311

Objectives / Topics Covered	- Software Engineers - Quality Assurance Engineers - Designers - Project Managers
Example Assignments / Labs	- Create a Mood Board - In this assignment, you will act as a designer and create a mood board for a store of your choosing. To visually represent the brand

and theme of the store, your mood board must include the following:
- ■ 1. A color palette that best represents the store's brand
- ■ 2. One or two fonts that align with the store's identity
- ■ 3. Images related to the store's products or target audience

- Create a Task Board
    - Imagine that you are a Project Manager. Before assigning work to members of the software development team, you need to create a list of tasks needed to create an application for the store you created a mood board for in the previous lesson.

Introduction to Python Programming 1 Supplemental Materials

Supplementary Units	Prerequisite/Recommended Unit(s)	Amount of Material
Assessment 1	Complete all units in main course	1 quiz (offline materials found in resources)
Intro to Python with Turtle Graphics	Can be used in place of *Karel in Python* module	21 lessons