

# Introduction

In this lesson, we will introduce the course, discuss its prerequisites, and talk about what we expect to learn.

## We'll cover the following



- Who is this course for?
- Learning outcomes for this course

This course covers the concepts of dynamic programming starting from basic recursion all the way to tabulation-based, bottom-up techniques. The concepts are explained with the help of visualizations and interactive code. There are many coding challenges including classics like the Traveling Salesman, Weighted Scheduling, and String Sub-Sequence problems. We compare multiple solutions for each of these challenges and explain each in depth.

## Who is this course for? #

This course is specifically designed for people who find dynamic programming challenging. If you are interested in learning the philosophy of dynamic programming or if you want to dive deep into different dynamic programming problems, this course is for you.

Although most of the concepts are explained in detail, some familiarity with data structures and algorithms is necessary. Moreover, knowledge of Python is expected, as we will be writing code in Python.

## Learning outcomes for this course #

After taking this course, you should be able to:

- Analyze and formulate a problem recursively.
- Use memoization to improve simple recursion.
- Write a bottom-up version of the recursive algorithms.
- Argue about the time and space complexity of various algorithms.

- Recognize recurring patterns in different problems to solve them using dynamic programming.
- 

Alright! Let's begin with an introduction to recursion in the next lesson.