

Data Structures and Algorithms in Python - Full Course for Beginners

This is a practical beginner friendly and coding focused online course that will help you improve your programming skills. The course runs over six weeks with two hour video lectures every week with live interactive coding using the python programming language. You will get a chance to practice and improve your coding skills with weekly programming assignments consisting of real interview questions. This course will help you prepare for coding interviews and you can earn a verified certificate of accomplishment. This course assumes very little background in programming and mathematics but you still do need a little bit for instance you do need to know basic programming with python things like variables data types loops and functions and do n't worry if you don't know them already you can click through and follow these links. This course takes a coding focused approach towards learning and in each notebook or each tutorial we will focus on solving one problem and then learn the techniques algorithms and data structures to device. The course is set up a machine for you on the cloud using software called binder it 's an open source software.

The strategy and technique that we're learning here will be widely applicable and we will soon use it to solve harder problems now before while you think about the problem and before we start solving it. I just want to talk about why you should learn data structures and algorithms. Whether you're pursuing a career in software development or data science it 's almost certain that you will be asked programming problems like reversing a linked list or balancing a binary tree in a technical interview. A systematic strategy that you should apply in interviews or in coding problems on encoding assessments or in general whenever you are faced with a problem like this so here 's the strategy that we will apply step one state the problem clearly identify the input and output formats. Step two come up with some example inputs and outputs and try to cover all the edge cases step three comes up with a correct solution for the problem it can be as simple as possible and state it in plain english step four and this is optional sometimes implement the solution and test it using example inputs. The first step is to state the problem clearly and precisely in abstract terms because computers do n't understand people. Computer science can represent the sequence of cards as a list of numbers. We need to write a program to find the position of a given number in a list arranged in decreasing order.