**Introduction: Interview Skills**

The goal of this unit is to review some key technical interview topics as well as soft skills which employers are looking for in potential candidates. You will practice common JavaScript algorithms and prepare yourself for the whiteboarding interview. You will also learn what soft skills will set you apart and leave a lasting impression on employers. Finally, you will watch several walkthroughs of coding problems often used by tech industry leaders such as Google, Facebook, and Amazon.

After this unit, you will be able to:

* Think on the fly and show your thought processes in solving a problem
* Work your way through several common coding questions
* Respond comfortably to open-ended interview questions
* Set yourself apart and leave an impression on employers

**Clarifying the Problem**

Whiteboarding interviews begin with a problem from the interviewer. The interviewee must be confident they understand the dimensions of the problem!

Software development is full of ambiguity. Programming requires concrete deliverables, but company needs can be murky. Even when the need is clear, a feature could have dozens of possible implementations. The ability to clearly define a problem is an important skill to demonstrate.

When the interviewer presents their technical question, repeat the question back to the interviewer in your own words. This gives you a moment to think and will resolve any glaring misunderstandings.

Once you’ve repeated the question, ask every clarifying question that comes to mind.

Assumptions must be communicated to the interviewer so there is agreement on the scope of the problem.

For example, if asked:

**Write a function that returns duplicate characters in string.**

Here are some questions which may come to mind:

* What is the desired return value?
* True|False, a list of characters, or …?
* Do punctuation and spaces count as “characters”?
* Should case be considered?
* are "a" and "A" duplicates?
* Should we be checking for [Unicode](https://en.wikipedia.org/wiki/Unicode) characters?
* Can we assume it’s a 26 character alphabet?

**Instructions**

We’ll apply these steps to a single problem through the rest of the lesson.

Given a list of numbers, return whether the list contains Pythagorean Triplets.

Rephrase this problem in your own words and write that down.

Then, write down every question or assumption you have.

Watch the video to see how we clarified this question.

**Producing Inputs and Finding Edge Cases**

When the question is clear, we then produce concrete inputs and outputs. These inputs guide a solution for the remainder of the interview so write them on the board!

You may still be unclear how to solve the problem in code, but it’s certain that given an input, X, your function will produce an output, Y.

Make one input the happy path: input that reflects a common scenario.

For example, you’re asked to write a function which capitalizes the first letter of an input string.

A good input could be "apple", which returns "Apple" because this demonstrates the function’s purpose.

If the input were "Apple", it would return "Apple". That’s correct but less informative.

Also think about edge cases, or inputs which do not reflect a common scenario and may cause problems.

For the capitalization function, what should you return if given None as input? Or a number?

Instructions

Write inputs and outputs for the Pythagorean Triplet problem.

Try to give yourself a few different cases.

Watch the video to see how we made inputs/outputs.

<https://www.youtube.com/watch?v=xzYgM0eIauA&t=161s>

<https://www.youtube.com/watch?v=tccfJmGM0XI&t=25s>

[Cheatsheet](https://www.codecademy.com/learn/paths/back-end-engineer-career-path/tracks/becp-22-interview-skills/modules/wdcp-22-technical-interviews-whiteboarding/cheatsheet)