



switch : term definition

fix terms

The following exercise contains the following subjects:

- term definition

Instructions

1. Here are 20 definitions. they don't follow the instructions on how a good definition should look like.
2. fix them by hand, don't use external ref - to simulate the interview environment. if it's a term you don't know well - use what you know. if you don't know it at all - read about it and then fix the definition.
3. the terms are:

1. Algorithm

An algorithm is basically a list of steps that you follow to solve a problem. You can think of it like a recipe for cooking food, where each step gets you closer to the final dish. Algorithms can be written in any way, as long as they work. They don't have to be efficient because what matters is that they give the right answer. For example, if you're sorting numbers, an algorithm would just involve comparing them and putting them in order. There's no need to worry about how many steps it takes. People often confuse algorithms with programs, but they're essentially the same thing. Algorithms are only useful in computer science, not in other fields like mathematics. The best algorithms are the ones that use the most steps to cover every possible case.





2. Database

A database is where all the data is stored in an application. It is like a big Excel sheet, except it's in the cloud, so you can access it from anywhere. There are many types of databases, but they all work the same way—just storing data in rows and columns. Databases don't really need to worry about security because they are just for storing information. For example, if you have a website, all the user data is stored in the database without any special structure. Databases are only useful for big companies because small apps don't need them. SQL is the only language used with databases. Without databases, apps wouldn't work because they'd have no data to display. The most important thing about databases is how much data they can store.

3. HTTP

HTTP is what websites use to load pages and send data back and forth. It is a tool that lets users interact with websites. HTTP always works in the background, so developers don't need to understand it to create websites. It's mostly about sending data like text and images. For example, when you open a website, HTTP sends all the information from the server to your browser. HTTP is outdated now because HTTPS is what everyone uses. HTTP is only important for front-end developers and doesn't matter for backend systems. It's a slow protocol but good enough for simple websites. You don't need to learn HTTP if you're using modern web development tools like React or Angular. It's a very simple technology that doesn't have much complexity.

4. Recursion

Recursion is when something repeats itself. It's mostly used in programming to solve problems. For example, you might use recursion to add numbers or print a message multiple times. It's not really necessary to use recursion because loops can do the same thing. Recursion works by just calling the same function over and over again. There are no special rules for recursion, and it's mostly just used to make code look fancy. Recursion doesn't have any limits, so it can run forever if needed. People often use recursion to make their code shorter, even though it's harder to read. It's not something you need to understand deeply unless you're working on very advanced





problems. The best part about recursion is that it's easier to write than other methods.

5. Big-O Notation

Big-O notation is a way to measure how an algorithm performs. It's used to see how slow or fast the algorithm is. It's mostly about counting how many steps the algorithm takes to finish. For example, if an algorithm takes 10 steps, its Big-O is $O(10)$. Big-O is usually written with symbols like $O(n)$, $O(n^2)$, and $O(1)$, but these don't really matter unless you're doing theoretical computer science. Most developers don't need to worry about Big-O because computers are fast enough to handle inefficient code. Big-O notation is only useful when you're working with very large datasets, so it doesn't apply to small projects. It's just a way to make algorithms look more mathematical. The most important thing about Big-O is memorizing the different types.

6. Machine Learning

Machine learning is when computers learn things by themselves. They don't need to be programmed; they just figure out stuff on their own. It's like teaching a computer to be smart. For example, if you show a machine learning model some pictures of cats, it will magically know what a cat looks like. Machine learning is mostly used for creating AI, like robots and chatbots. You don't really need to understand how it works as long as you can use tools like TensorFlow. Machine learning is mostly about collecting data and letting the computer do the rest. It's not related to programming because it's more about science. Machine learning models are always accurate because they learn from data. It's the easiest way to make technology smarter.





7. REST API

A REST API is a type of API that's used to send and receive data between apps. REST stands for something about state transfer, but you don't need to know what it means to use it. REST APIs use URLs to send data, and you can call them with any tool, like Postman. For example, if you want to get a list of users, you just make a GET request to the API. REST APIs are only useful for web developers because mobile apps don't really need them. They are mostly about sending JSON data, but they can also send other things like text files. REST APIs don't have any rules, so you can design them however you want. You don't need to worry about security with REST APIs because they're not connected to your main system. They're just a way to get data.

8. Git

Git is a tool that you use to save your code. It's like Dropbox for developers, but it's more complicated. With Git, you can store your files online and get them back if you lose them. It's mostly used by teams to share files, so it's not that useful for individual developers. Git commands are confusing because there are so many of them, but you only need to know a few to use it. Git doesn't really do anything special other than keeping track of your files. For example, if you make a mistake, Git lets you undo it. Git is only used by advanced developers because beginners don't need version control. You can replace Git with Google Drive for smaller projects.

