Unit 1 of 9

Ask Learn

✓ 100 XP

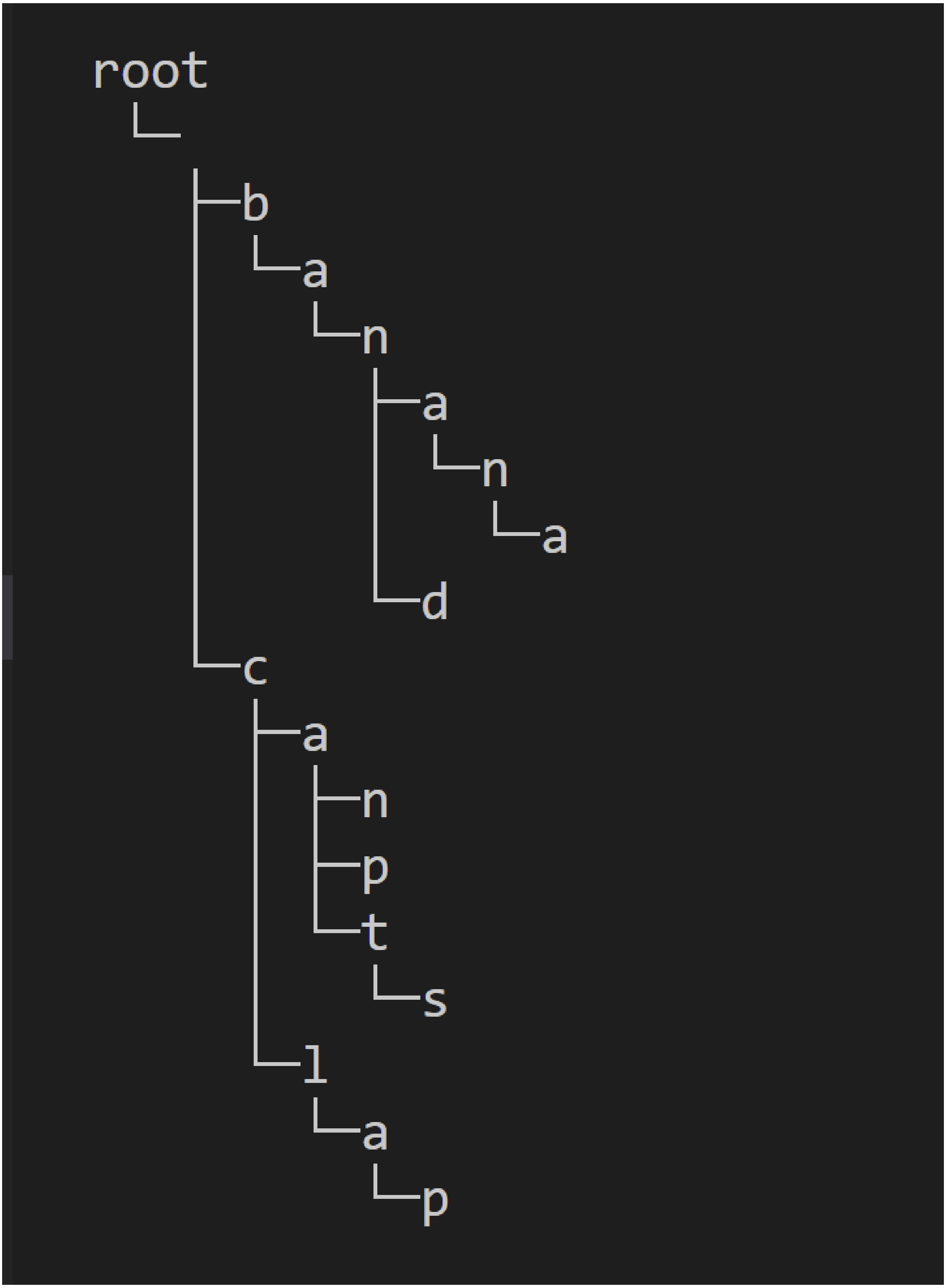
Introduction

3 minutes

> **Important**
>
> This module requires the GitHub Copilot extension which requires a subscription, as well as GitHub Copilot Chat, which is currently in public beta and subject to changes.

The Trie data structure is a tree-based data structure used for storing and searching a large collection of strings. It's useful for tasks that involve prefix matching and autocompletion. The name "Trie" comes from the word "retrieval," as it efficiently retrieves words that share a common prefix.

The key characteristic of a Trie is that each node in the tree represents a single character of a string. The root node represents an empty string, and each path from the root to a leaf node represents a complete word or string. Each node can have multiple children nodes, one for each possible character in the input alphabet. The following image is an example of a Trie containing the words `cat`, `clap`, `can`, `cats`, `cap`, `band`, and `banana`:



```
root
 └─
   ├─b
   │ └─a
   │   └─n
   │     ├─a
   │     │ └─n
   │     │   └─a
   │     └─d
   └─c
     ├─a
     │ ├─n
     │ ├─p
     │ └─t
     │   └─s
     └─l
       └─a
         └─p
```

This module guides you through an existing Trie project, and helps you learn to use GitHub Copilot to interpret, debug, and extend the project code. By the end of this module, your Trie project is able to insert, search, and delete words from the data structure.

> ⓘ Note
>
> If you’re unfamiliar with the Trie data structure, don’t worry! You can still complete this module by following the instructions.

> ⓘ Note
>
> This is a guided project module where you'll complete an end-to-end project by following step-by-step instructions.

> ⓘ Important
>
> This project relies on GitHub Copilot code completion. While GitHub Copilot can generate code, the results it produces may vary in accuracy and consistency. Please take this into consideration as you complete each exercise.

Learning objectives

In this module, you practice:

- Interpreting code with Copilot.
- Authoring code with Copilot.
- Debugging and testing with Copilot.

Prerequisites

- Visual Studio Code.
- GitHub Copilot extension.
- GitHub Copilot Chat extension.
- Basic understanding of the C# programming language.
- Familiarity with classes, methods, variables, and logic statements.

Next unit: Prepare

Next >