



# Sriju & the Talking TREES of CodeLand!



Understanding Tree Data Structures

# Sriju & the Talking Trees of CodeLand!

## WHAT IS A TREE?

Sriju! Welcome to  
Tree Town! 🌳



A Tree is a way to store data in  
a hierarchical form.

So it's like a  
family tree?



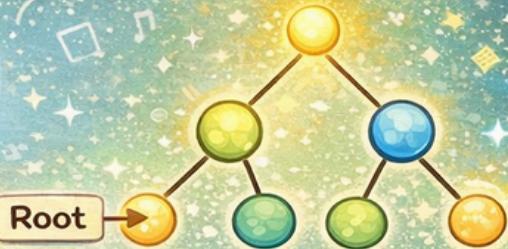
A Tree is a way to store data  
in a hierarchical form

# Sriju & the Talking Trees of CodeLand!

## TREE PARTS

Let's break it down!

Each circle is called  
a **Node**.



Root

★ Root → Starting point

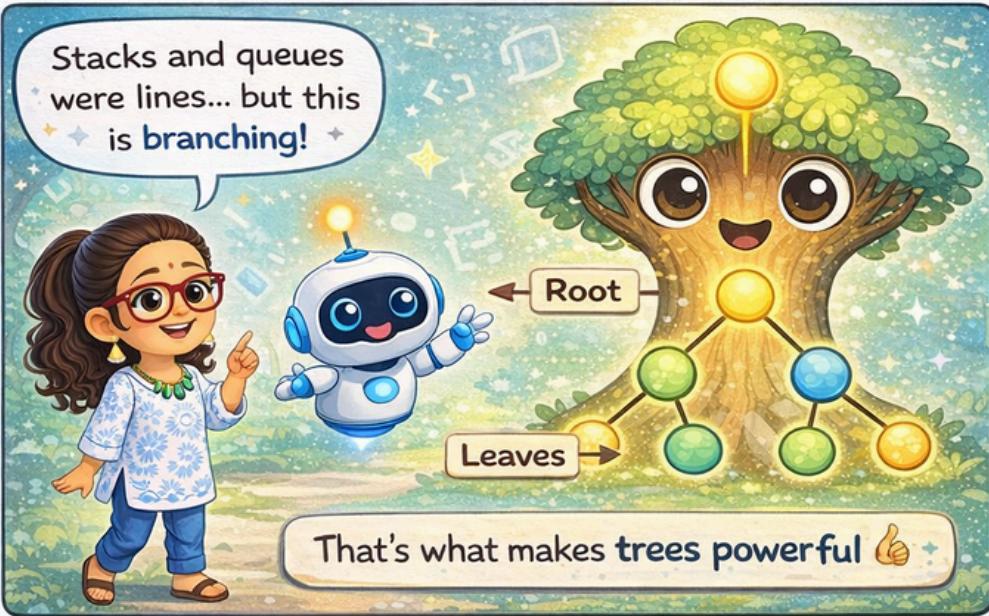
└ Children → Connected nodes

└ Leaves → Nodes with  
no children

Stacks and queues  
were lines... but this  
is branching!

← Root

Leaves →



# Sriju & the Talking Trees of CodeLand!

## WHEN TO USE TREES?

Use Trees when data has  
levels or categories.



Use Trees when data has  
levels or categories.

Oh! Everywhere data  
needs order!



File System



Family Tree



Website Menus

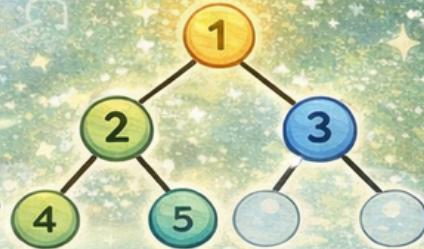


Search Results

# Sriju & the Talking Trees of CodeLand!

## HOW DO TREES WORK?

This is a **Binary Tree**.



Each node can have at most **two children**.

## How we step through the tree:



Inorder → left, root, right

## Preorder → root, left, right

**Postorder** → left, right, root

## 2 How we step through the tree:



# Sriju & the Talking Trees of CodeLand!

## WHEN NOT TO USE TREES?

Are there times when trees  
are a bad idea?



Trees are not good if data  
is totally random.

Use trees only when  
structure matters.



Use trees only when structure matters

# Sriju & the Talking Trees of CodeLand!

## WHEN NOT TO USE TREES?

Trees help data grow  
in the right direction!



### TREE SUMMARY

**What:** Hierarchical data structure

**When:** Structured & categorized data

**How:** Nodes connected like branches



See you in the next  
adventure, Sriju!

**A Sriju Comic**



**Curiosity Lives Here!**

**Stories that smile back at you**