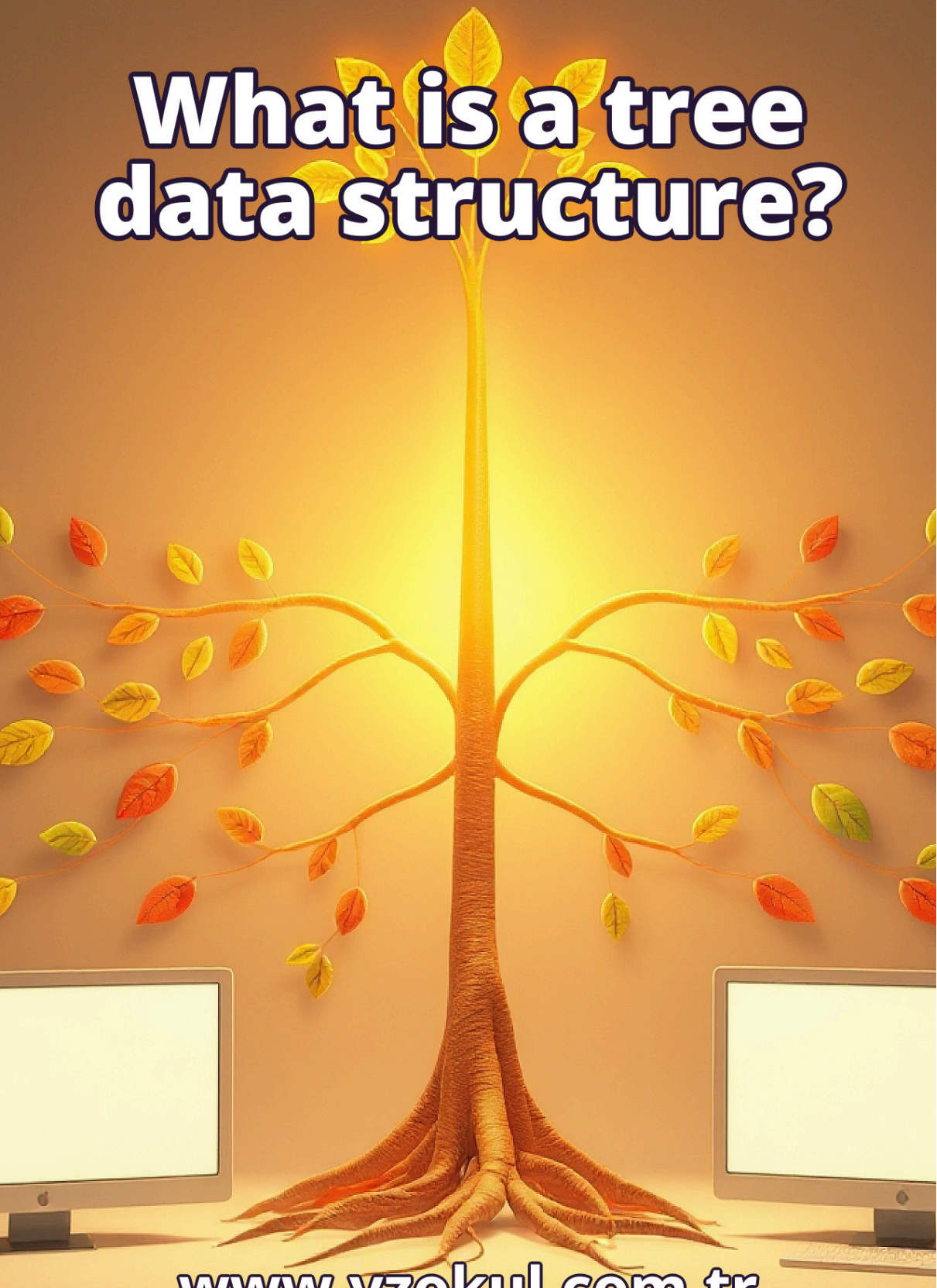
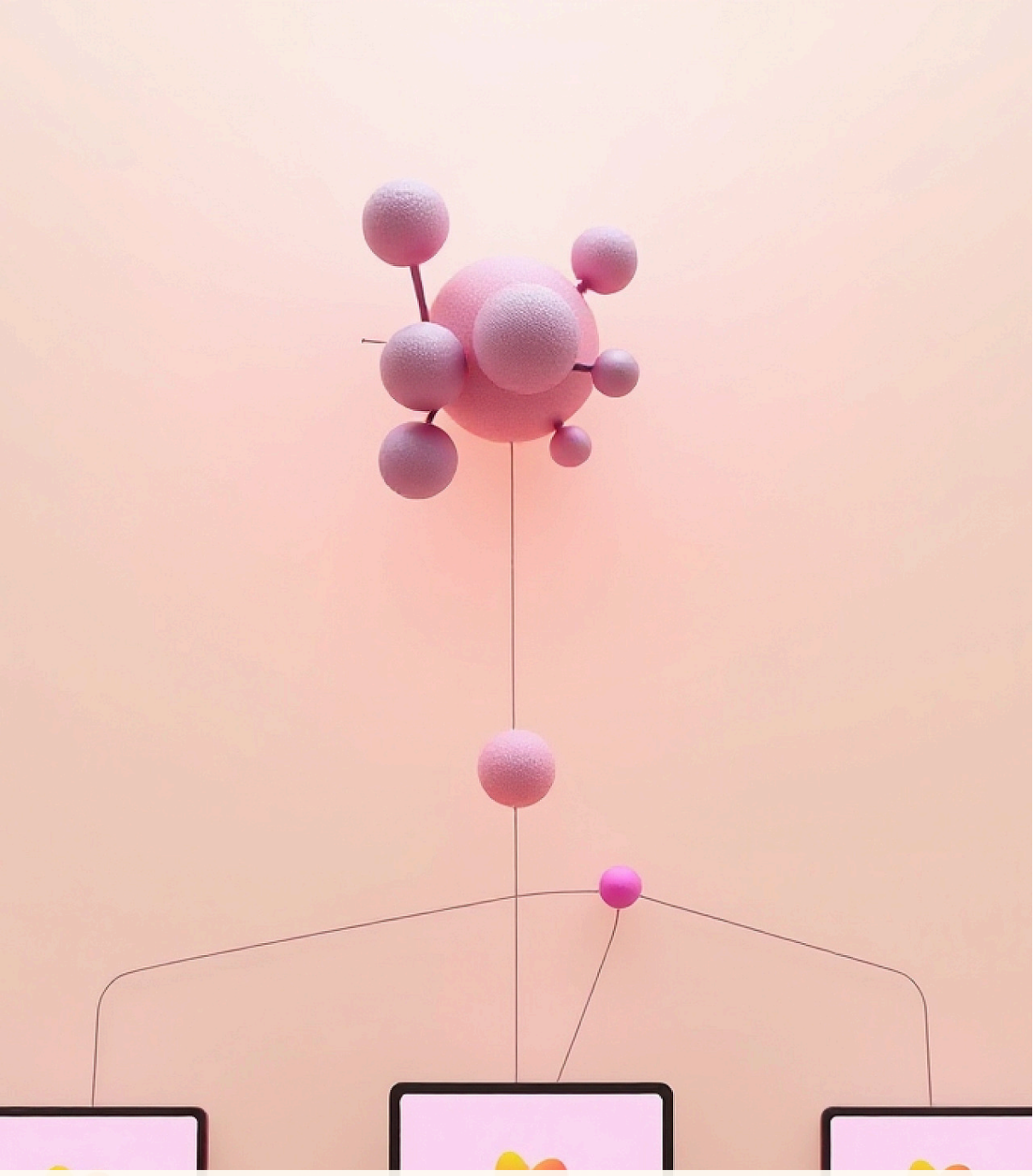
What is a tree data structure?



www.yzokul.com.tr

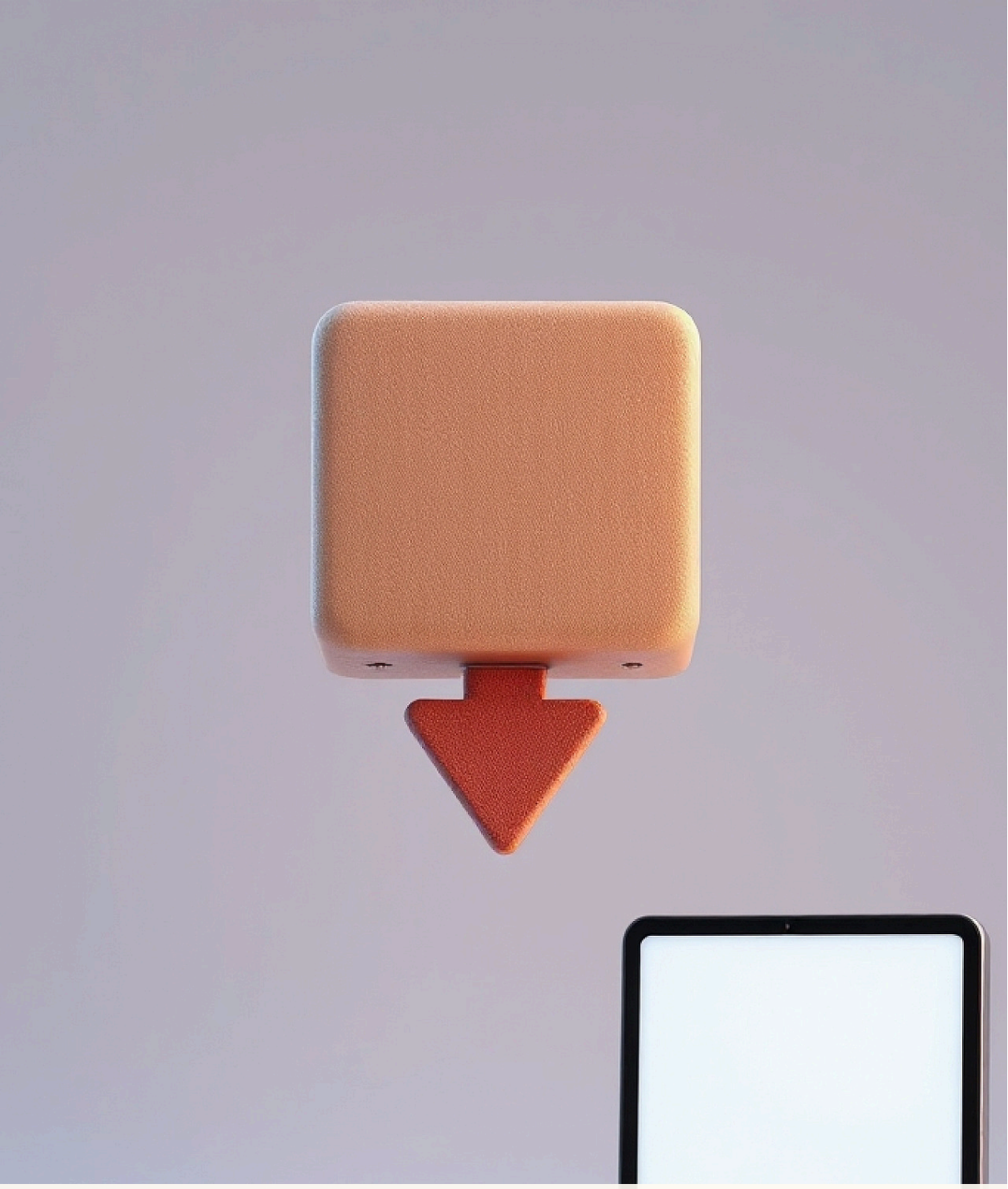
What is a tree data structure?



A tree data structure is a way to organize items so that each item is linked to others as a parent or a child. The short answer: items are connected by parent-child links, starting from one top item called the root. This gives you the whole idea before the details. Think of it as a family of items with clear connections.



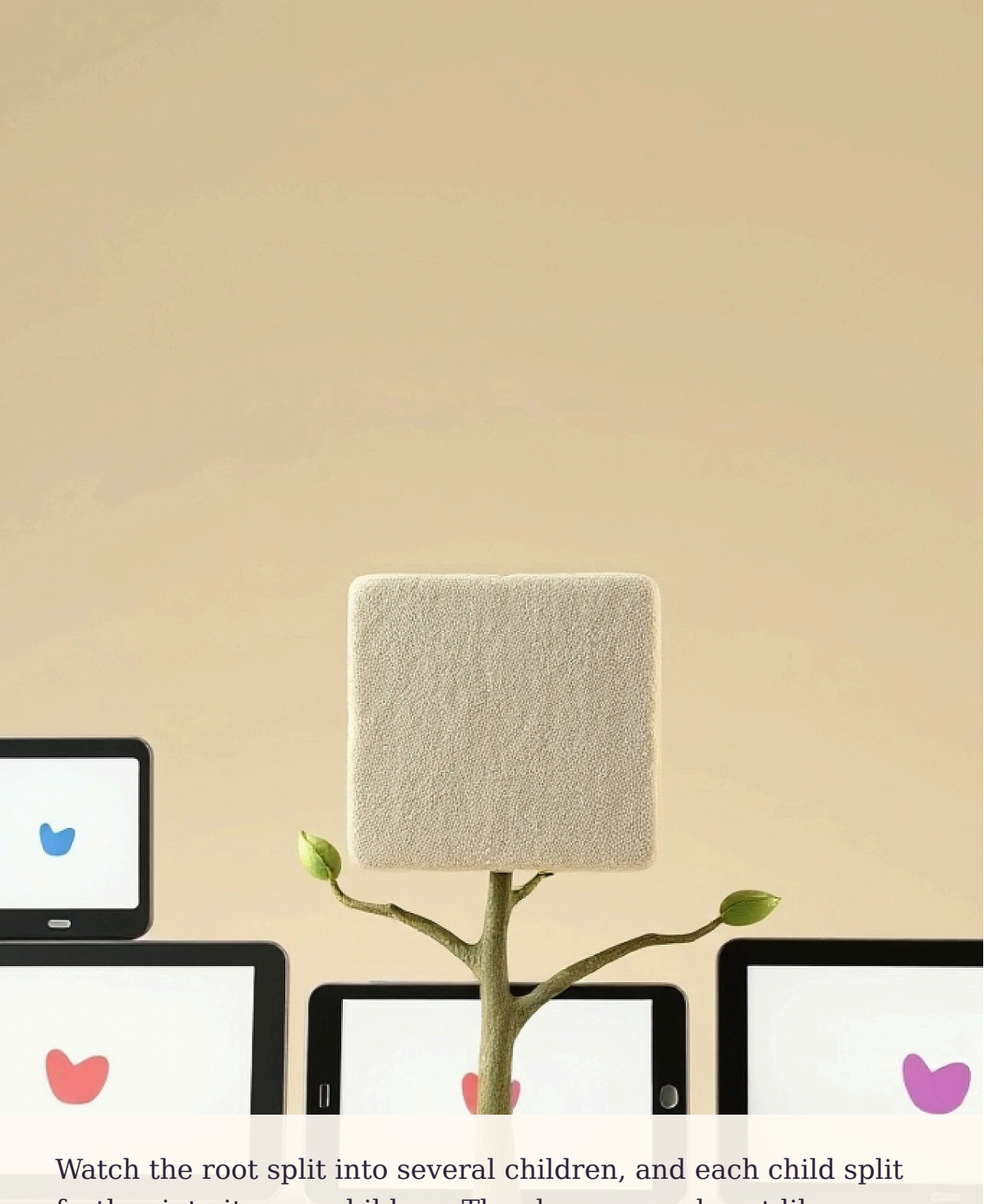
Picture a few items joined by lines, where each line connects one item to another. A tree is about how items are connected, not just a list of them. The connections form a shape, and that shape matters. For example, item A connects to item B and item C, and B connects to item D. The links are the structure.



Look at one connection between two items. The upper item is called the parent, and the lower item is called the child. Every link goes one way, from parent down to child. For example, if folder A contains folder B, then A is the parent and B is the child. You never go back up the same link.



Find the single item at the very top that has no parent above it. This top item is called the root node, and every tree starts from it. The root is the entry point for the whole structure. For example, in a computer, the main drive folder can act as the root. All other items can be reached from there.



Watch the root split into several children, and each child split further into its own children. The shape spreads out like branches, which is why it is called a tree. Each branch is a path of parent-child links. For example, the root has two children, and each of those has two more, making a small branching tree.



Follow the branches to the items at the very bottom that have no children. These end items are called leaves, and every path stops there. Leaves are the final items in a branch. For example, in a folder tree, the files inside the deepest folders are leaves because they contain nothing further. Every tree has at least one leaf.



Compare the tree to folders on a computer: one main folder holds subfolders, which hold more subfolders. You recognize the same parent-child shape you just learned. The main folder is like the root, and subfolders are like child nodes. For example, a Documents folder holds a Work folder, which holds a Reports folder. This is a familiar tree.



Open a folder, then a subfolder inside it, then one inside that. You move down one level at a time, just like following a branch. Each open shows you the children of the current folder. For example, open Documents, then Work, then Reports. You have followed a path from parent to child to child.



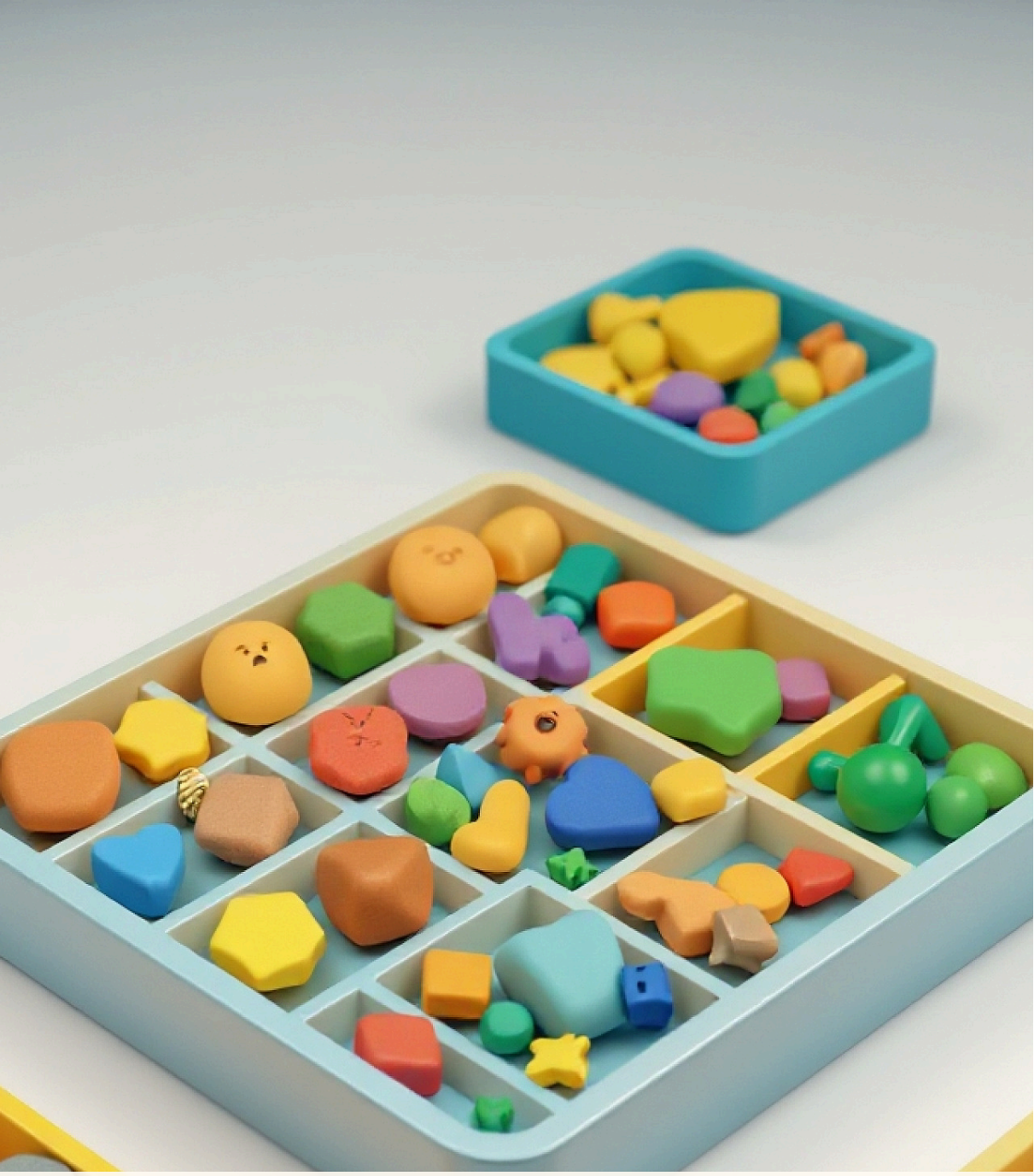
Line up the items by how deep they sit: the root at level one, its children at level two, and so on. A tree stores information in levels, and this arrangement is called a hierarchy. Each level shows how far an item is from the root. For example, the root is level one, its children are level two, and their children are level three.



See examples like a family tree or a company chart where some things sit above others. Trees are a natural way to store hierarchical information because they show who reports to whom or who comes from whom. For example, a company chart has a boss at the top, managers below, and staff below them. The levels make the structure clear.



Start at the root and check one branch at a time to find an item. A tree helps searching because you can skip whole branches you do not need. If the item is not in one branch, you move to the next. For example, to find a file, you open the likely folder first and ignore unrelated folders, saving time.



Place items into groups, then subgroups inside them, like sorting animals by kind. Trees are used for classification, putting similar things together. Each group is a parent, and its members are children. For example, animals split into mammals and birds, and mammals split into dogs and cats. This grouping forms a tree.



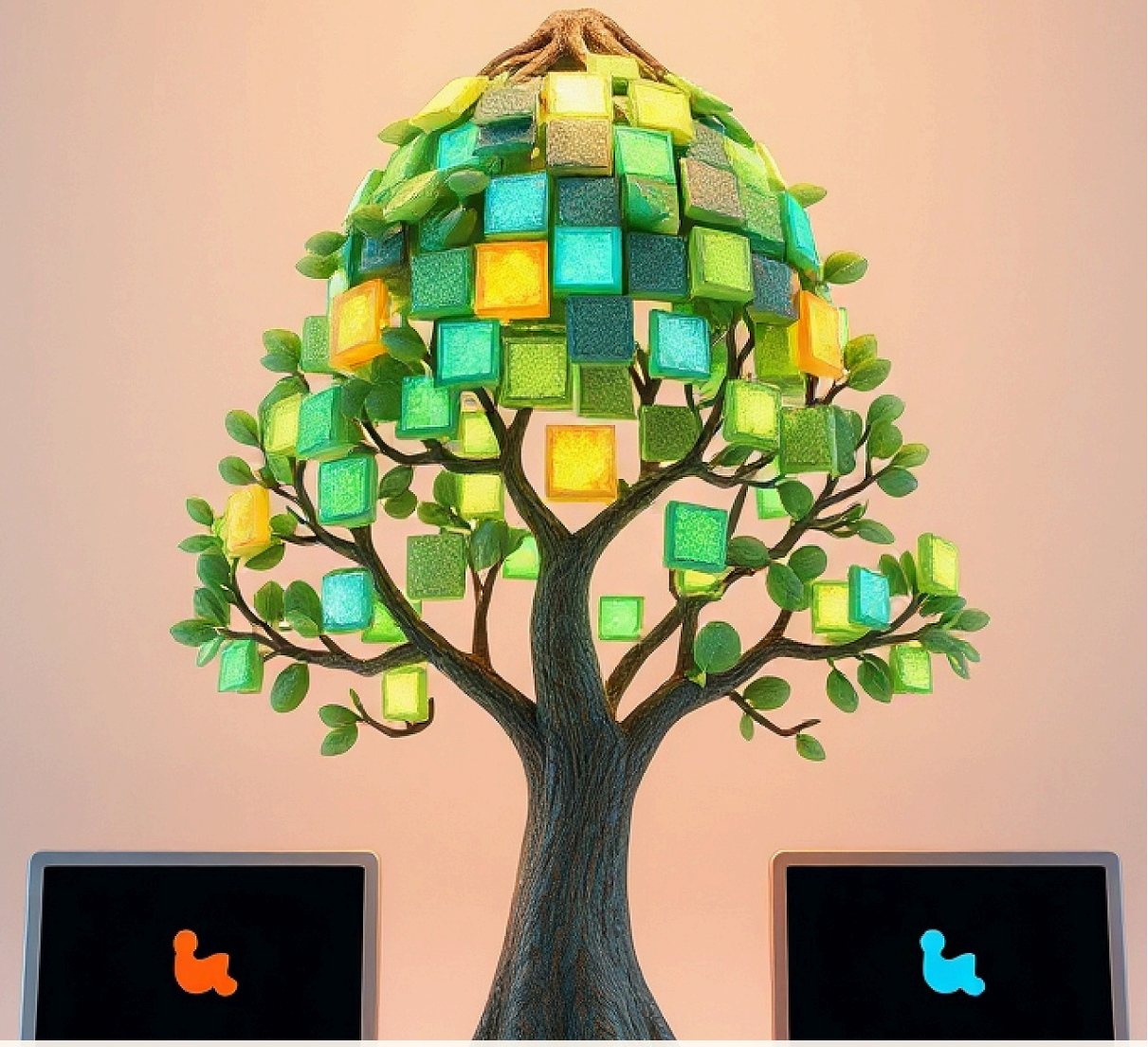
Compare two trees and notice one keeps children in order while another does not. Each type of tree has its own access rules, so behavior can differ. Some trees allow fast searching, while others focus on grouping. For example, a sorted tree keeps left children smaller and right children larger, but an unordered tree does not care about order.



See someone treat a tree like a straight line of items. Correct this: in a tree, one item can have many children, so it branches instead of staying in a row. A list has one item after another, but a tree spreads out. For example, a root with three children is not a list; it is a branching structure.



See a drawing where an item has two parents above it, which is not a tree. Learn that each item in a tree has exactly one parent, except the root, which has none. Two parents would make a network, not a tree. For example, if folder B is inside both folder A and folder C, that is not a tree.



Sum up the answer: a tree links items as parents and children from one root, branching into levels. Remember folders on a computer as the example, and that rules vary by tree type. The root is the top, leaves are the ends, and every other item has one parent. This structure stores hierarchical information and helps with searching and classification.