

Trees as areas of knowledge

A **tree** is a scope of view over a connected group of skills — an *area of knowledge* that forms a logical part of a larger whole.

Take a front-end developer. Their full role covers a lot of ground, but the ground has natural regions: **Testing**, **Frameworks**, **JavaScript/TypeScript**, **Styling/CSS**. Each of those regions becomes a tree. Inside each tree sit the skills of that area, connected by their prerequisites — and because most skills in an area depend on other skills in the same area, the whole thing naturally reads as a tree shape in a diagram: broad foundations at the bottom, specialised skills branching above.

Three things to know about trees:

- **The slicing is yours.** How a role divides into areas is up to the team. There's no canonical set of trees for "front-end developer" — there's the set that matches how *your* team thinks about its work. If the names feel obvious to the team, the slicing is right.
- **Every tree gets a clear name.** The name describes the area it covers — a person glancing at the forest should know what lives in each tree without opening it.
- **A skill is defined only once, anywhere.** Trees show skills; they don't own them. The same skill is never defined in two trees. (It may still *appear* under another tree as a root — that's the subject of the roots page.)

A tree is deliberately human-sized: big enough to represent a real area of expertise, small enough to take in at a glance and reason about in a conversation.



Revision #3

Created 2026-07-08 16:08:46 UTC by Admin

Updated 2026-07-08 16:37:48 UTC by Admin