Example tree
Definition of a tree

• A tree is either:
  – empty, or
  – a root value connected to other trees, called subtrees

• Notice that the definition is recursive.
Terminology

• **Size**: Number of nodes in the tree.
• **Leaf**: A node with no subtrees.
• **Internal node**: A node with one or more subtrees.
• **Height**: Length of the longest path from the root to a leaf (counting items on the path).

• **Child, Parent**
• **Descendant, Ancestor**

Every node (except the root) has *one* parent.
Example tree
Uses for trees

• Representing anything with a hierarchical structure.
• The inheritance hierarchy in a program
• Relationships in an arithmetic expression, e.g.,
  \[(3 - (4 \times (6 - 8))) + (25 / 2)\]
• The structure of files and folders on your computer
Recall: Recipe for writing recursive functions

1. Identify the recursive structure of the data.
2. Identify case(s) where the problem is so simple, it can be solved directly (base cases).
3. Pick a concrete example that is larger.
4. Write down the relevant recursive calls and what they should return.
5. **Assume** each one does do what it should.
6. Figure out how to combine their results to produce the overall result.
A common pattern with trees

The data’s recursive structure again suggests the code’s structure.

def f(self):
    if self.isempty():
        _______
    elif len(self._subtrees) == 0:
        _______
    else:
        for subtree in self.subtrees:
            subtree._____(_____)

The data’s recursive structure again suggests the code’s structure.