* A way of structuring data in a hierarchical way

Examples:

* + Family tree
  + Folders and files
  + Java Class hierarchy
  + Parse trees
  + Expression trees
* A tree T is a finite set of nodes connected by edges such that
  + If the tree is not empty, one node ^ is designated as the root
  + Each node v which is not the root has a unique parent node w, to which it is connected by an edge
* A tree T is a finit set of nodes such that, either
  1. The three is empty (having no nodes)
  2. Not empty
     1. One of the nodes is designated as the root, and
     2. The remaining nodes are partitioned into zero or more subtrees
* Terminology
  + Sibling – have the same parent
  + Parent
  + Child
  + Root
  + Degree – number of children
  + Leaf- external node
  + Internal node
  + Ancestor
  + Descendant
* Path – a sequence of nodes {n1, n2, …, nx} such that (Ni, Ni+1) is an edge (parent, child)
* Level – the level of a node v is the length of the path from the root to v
* Height – the of a tree is a the length of the longest path from the root to a leaf
* Ordered tree- a tree in which the order of the children is significant
* Binary tree – each node has at most 2 children
* Tree traversals – a process which visits every node exactly once
  + Binary trees
    - Pre Order: Root – left – right
    - Post Order: left – right – root
    - In Order: left – root – right
* Using a tree
  + Each node contains a data item
* Implementing trees
  + Usually a linked structure
  + Either a recursive or using tree nodes