Quick Review of Trees
Tree Data Structure

- Trees are useful for representing hierarchically ordered data
- The node at the top of the hierarchy is called the root node
- Nodes 1 level below the current node are children of the current node
- Nodes with no children are leaves
Tree Data Structure

Tree

0 root

1 intermediary nodes

3 leaves

Left Subtree
There are 4 common methods to traverse trees:

- **Pre-order**: 0 1 3 4 2 5 6
- **In-order**: 3 1 4 0 5 2 6
- **Post-order**: 3 4 1 5 6 2 0
- **Level-order**: 0 1 2 3 4 5 6
Since subtrees are also trees, recursion is a great technique in developing tree algorithms.

- **preOrder(node x)**
  - `print(x)`
  - `preOrder(x.left)`
  - `preOrder(x.right)`

- **inOrder(node x)**
  - `inOrder(x.left)`
  - `print(x)`
  - `inOrder(x.right)`

- **postOrder(node x)**
  - `postOrder(x.left)`
  - `postOrder(x.right)`
  - `print(x)`
Project 3
XML (eXtensible Markup Language)

- XML is designed to transport and store data, kind of like JSON (JavaScript Object Notation)

```xml
<?xml version="1.0" encoding="utf-8"?>
<MyBeerJournal>
  <Brewery name="Founders Brewing Company" location="Grand Rapids, MI">
    <Beer name="Centennial" description="IPA" rating="A+" dateSampled="01/02/2011">
      "What an excellent IPA. This is the most delicious beer I have ever tasted!"
    </Beer>
  </Brewery>
  <Brewery name="Brewery Vivant" location="Grand Rapids, MI">
    <Beer name="Farmhouse Ale" description="Belgian Ale" rating="B" dateSampled="02/07/2015">
      This beer is not so good... but I am not that big of a fan of english style ales.
    </Beer>
  </Brewery>
  <Brewery name="Bells Brewery" location="Kalamazoo, MI">
    <Beer name="Two Hearted Ale" description="IPA" rating="A" dateSampled="03/15/2012">
      Another excellent brew. Two Hearted gives Founders Centennial a run for it's money.
    </Beer>
  </Brewery>
</MyBeerJournal>
```
XML (eXtensible Markup Language)

- XML simplifies data transport by storing data in plain text format, which is software independent
- XML Tags are not predefined; you define your own tags
- XML is designed to be self-descriptive
- XML documents form a tree structure
<?xml version="1.0" encoding="utf-8"?>
<MyBeerJournal>
    <Brewery name="Founders Brewing Company" location="Grand Rapids, MI">
        <Beer name="Centennial" description="IPA" rating="A+" dateSampled="01/02/2011">
            "What an excellent IPA. This is the most delicious beer I have ever tasted!"
        </Beer>
    </Brewery>
    <Brewery name="Brewery Vivant" location="Grand Rapids, MI">
        <Beer name="Farmhouse Ale" description="Belgian Ale" rating="B" dateSampled="02/07/2015">
            This beer is not so good... but I am not that big of a fan of English style ales.
        </Beer>
    </Brewery>
    <Brewery name="Bells Brewery" location="Kalamazoo, MI">
        <Beer name="Two Hearted Ale" description="IPA" rating="A" dateSampled="03/15/2012">
            Another excellent brew. Two Hearted gives Founders Centennial a run for it's money.
        </Beer>
    </Brewery>
</MyBeerJournal>
XML (eXtensible Markup Language)

- Online Tutorial
  - [http://www.w3schools.com/xml/](http://www.w3schools.com/xml/)
Functions that you will use to build the tree

- `first_node()` // Go to child node
- `next_sibling()` // Go to sibling
- `value()` // Get value

The header file is very well documented, so take a look at it!
Questions?