ECE430.217 Data Structures

# The Tree Data Structure

Weiss Book Chapter 4.1

**Byoungyoung Lee**

**https://compsec.snu.ac.kr**

**byoungyoung@snu.ac.kr**

Lecture slides were prepared based on Douglas W. Harder's notes (**dwharder@alumni.uwaterloo.ca).**

# Outline

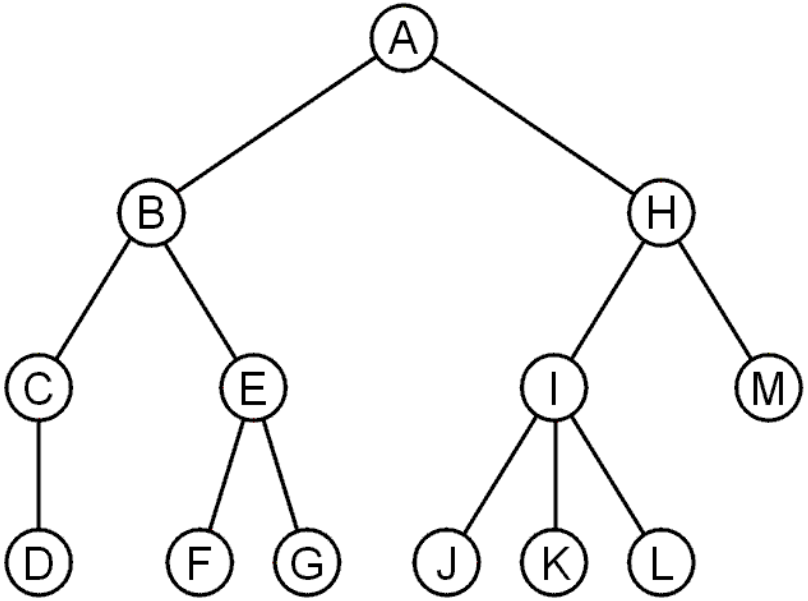
In this topic, we will cover:

- Definition of a tree data structure and its components
- Concepts of:
  - Root, internal, and leaf nodes
  - Parents, children, and siblings
  - Paths, path length, height, and depth
  - Ancestors and descendants
  - Ordered and unordered trees
  - Subtrees
- Examples
  - XHTML and CSS

# Trees

A rooted tree data structure stores information in *nodes*

- There is a first node, or *root*
- Each node has multiple references to successors



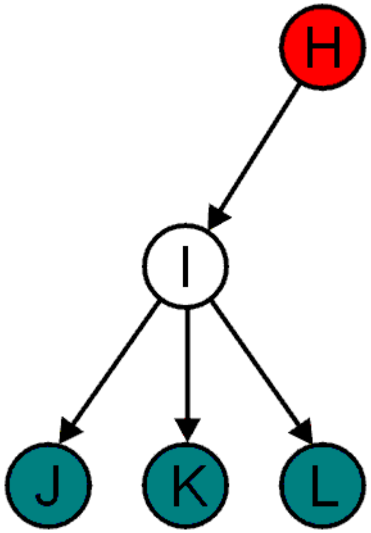
# Terminology

All nodes will have zero or more child nodes or ***children***

- I has three children:  J, K and L

For all nodes other than the root node, there is one **parent node**

- H is the parent I



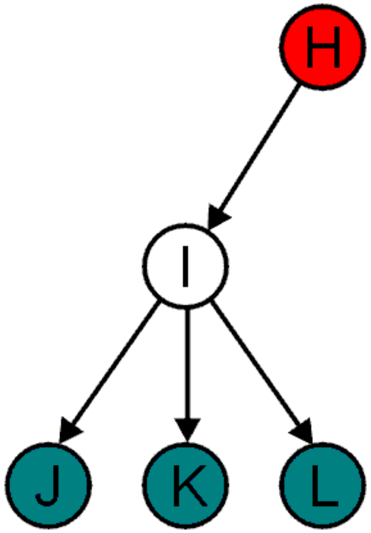
# Terminology: Degree

The ***degree*** of a node is defined as the number of its children:

$\deg(\mathrm{I}) = 3$

Nodes with the same parent are *siblings*

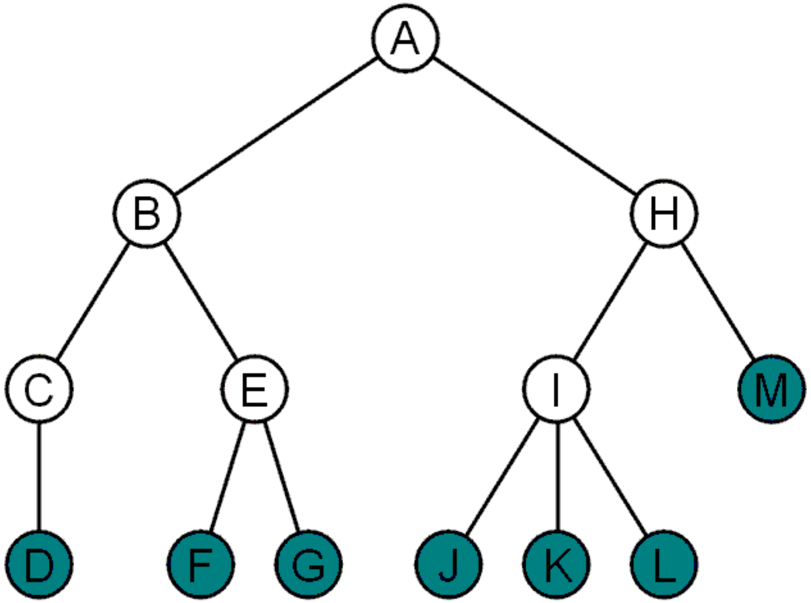
– J, K, and L are siblings



# Terminology: Nodes

Nodes with degree zero are also called ***leaf nodes***

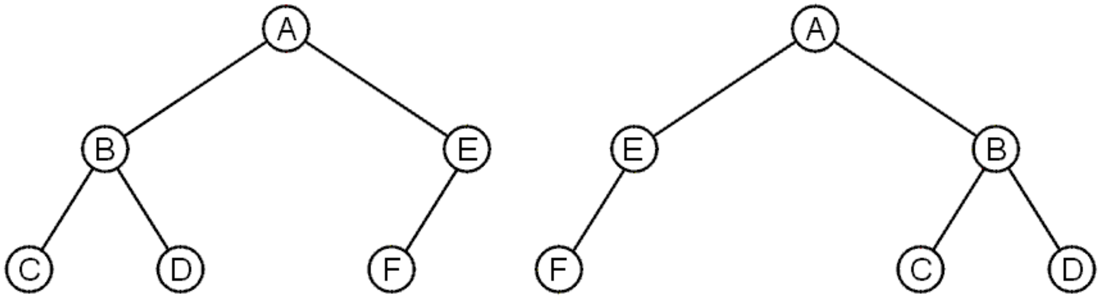
All other nodes are said to be ***internal nodes (or non-leaf nodes)***, that is, they are internal to the tree



# Terminology: Ordered Trees

These trees are equal if the order of the children is ignored

- ***unordered trees***

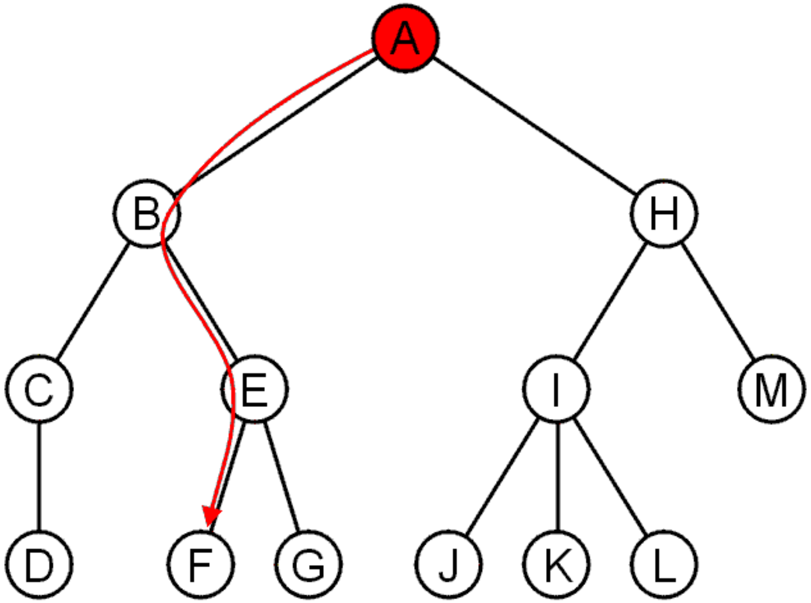


They are different if order is relevant (*ordered trees*)

- We will usually examine ordered trees (linear orders)

# Terminology: Root

The shape of a rooted tree gives a natural flow from the *root node*, or just *root*



# Terminology: Path

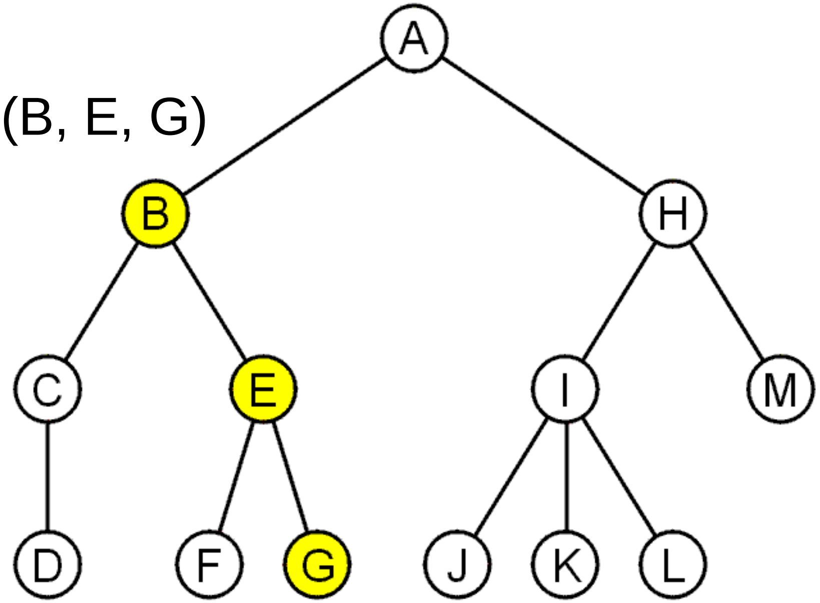
A **path** is a sequence of nodes

$(a_0, a_1, ..., a_n)$

where $a_{k+1}$ is a child of $a_k$ is

The length of this path is $n$

*E.g.*, the path (B, E, G)
has length 2

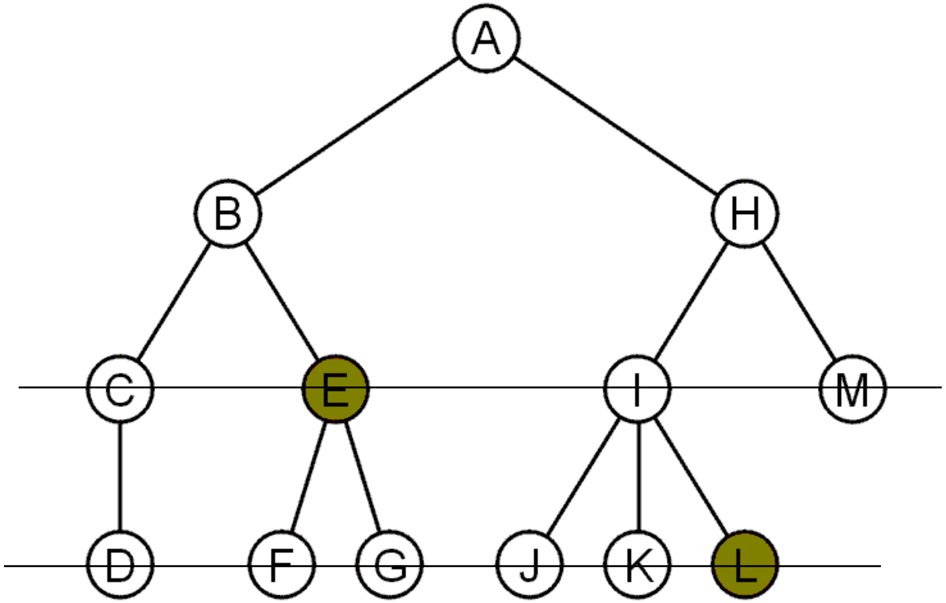


# Terminology: Depth

There exists a unique path from the root node to any non-root node

The length of this path is the ***depth*** of the node, *e*.*g*.,

- E has depth 2
- L has depth 3



# Terminology: Height

The ***height*** of a tree is defined as **the maximum depth** of any node within the tree

The height of a tree with one node is 0

- Just the root node

For convenience, we define the height of the empty tree to be $-1$

# Terminology: Ancestor/Descendent

If a path exists from node $a$ to node $b$:

- $a$ is an ***ancestor*** (***parent***) of $b$
- $b$ is a ***descendent*** (***child***) of $a$

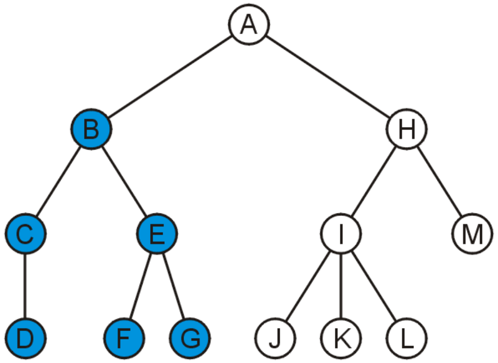
Thus, a node is both an ancestor and a descendant of itself

- We can add the adjective *strict* to exclude equality: $a$ is a *strict descendent* of $b$ if $a$ is a descendant of $b$ but $a \neq b$

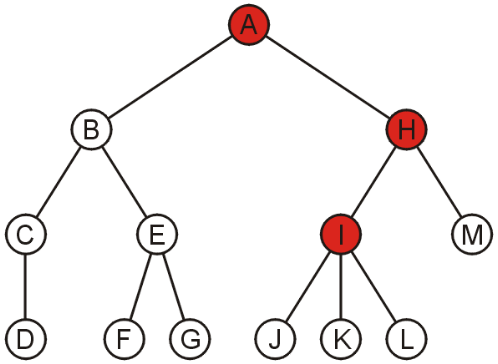
The root node is an ancestor of all nodes

# Terminology: Ancestor/Descendent

The descendants of node B are B, C, D, E, F, and G:

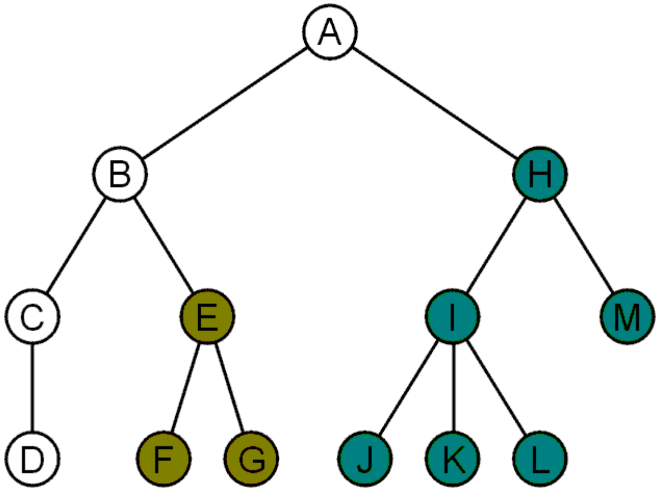


The ancestors of node I are I, H, and A:



# Terminology: Subtree

Given any node $a$ within a tree with root $r$, the collection of $a$ and all of its descendants is said to be a ***subtree*** *of the tree with root* $a$



# Example: XHTML and CSS

The XML of XHTML has a tree structure

Cascading Style Sheets (CSS) use the tree structure to modify the display of HTML

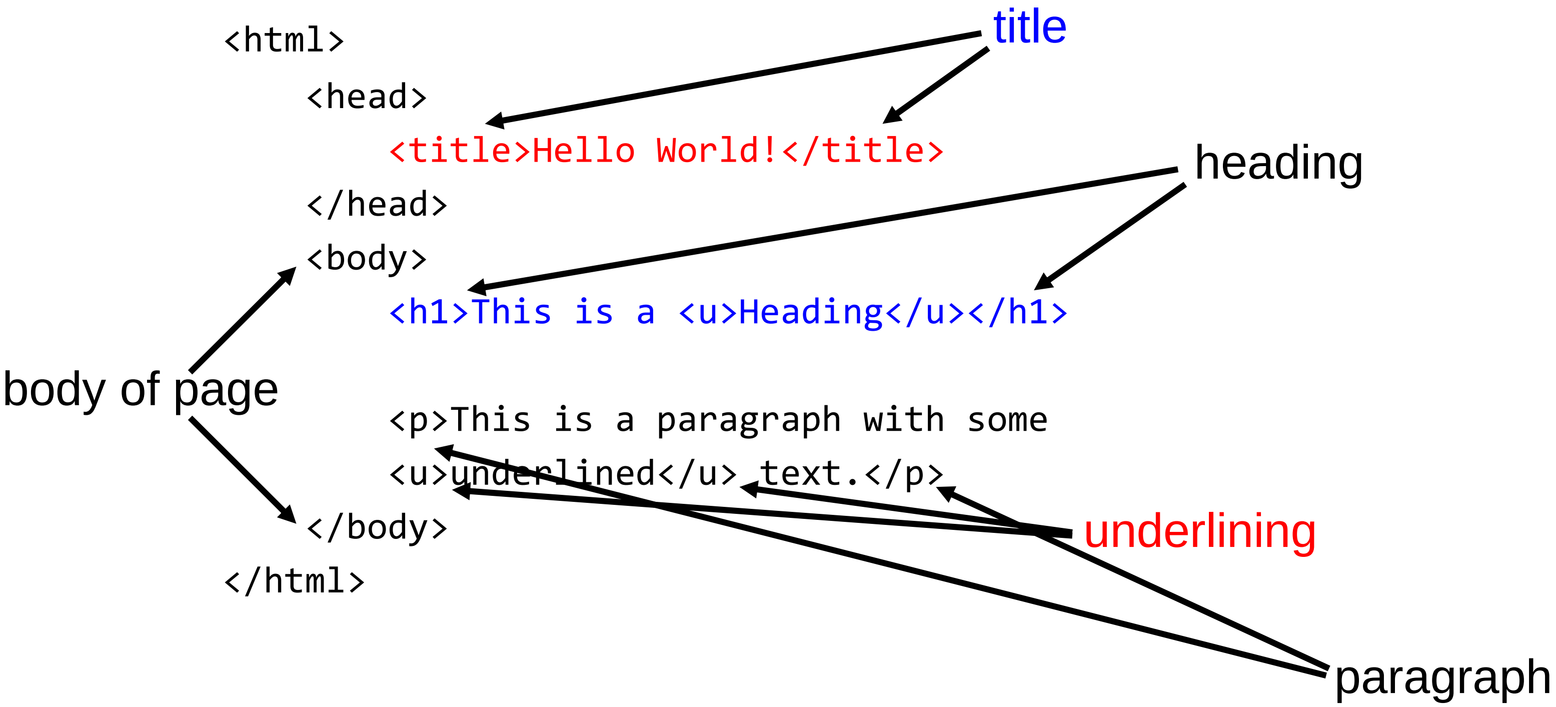
# Example: XHTML and CSS

Consider the following XHTML document

```
<html>
    <head>
        <title>Hello World!</title>
    </head>
    <body>
        <h1>This is a <u>Heading</u></h1>

        <p>This is a paragraph with some
        <u>underlined</u> text.</p>
    </body>
</html>
```

# Example: XHTML and CSS

Consider the following XHTML document



```
<html>
    <head>
        <title>Hello World!</title>
    </head>
    <body>
        <h1>This is a <u>Heading</u></h1>

        <p>This is a paragraph with some
        <u>underlined</u> text.</p>
    </body>
</html>
```

title

heading

body of page

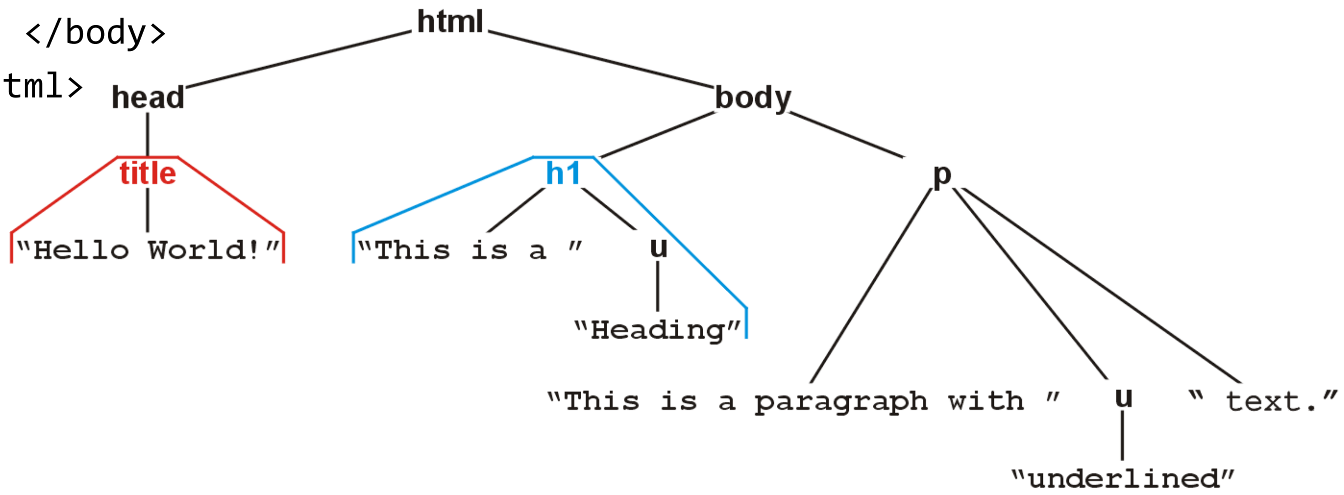
underlining

paragraph

# Example: XHTML and CSS

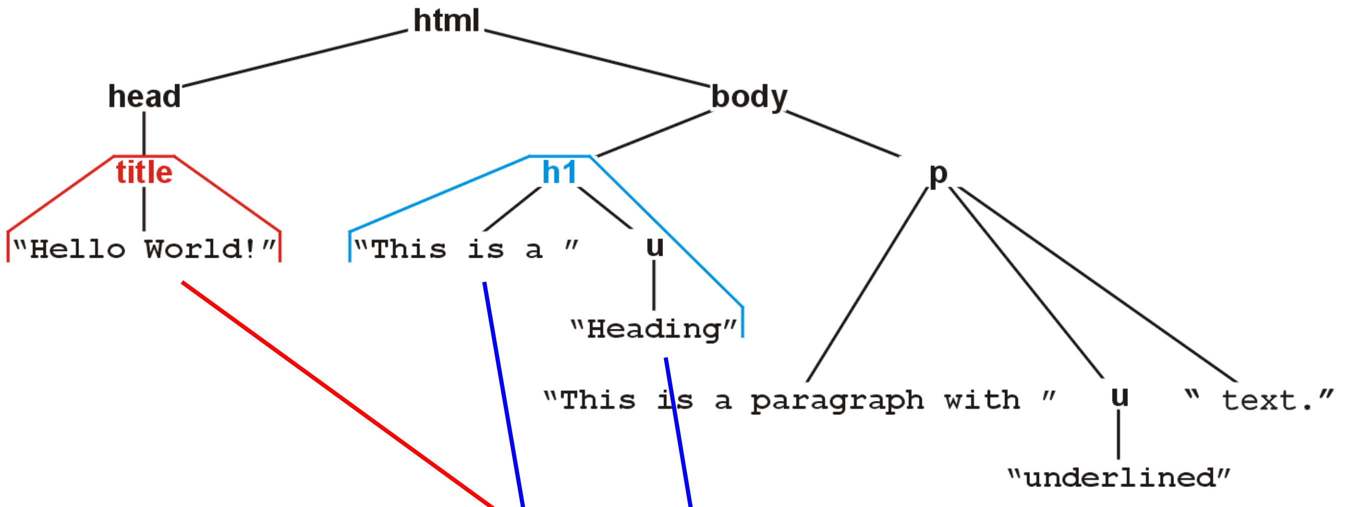
The nested tags define a tree rooted at the HTML tag

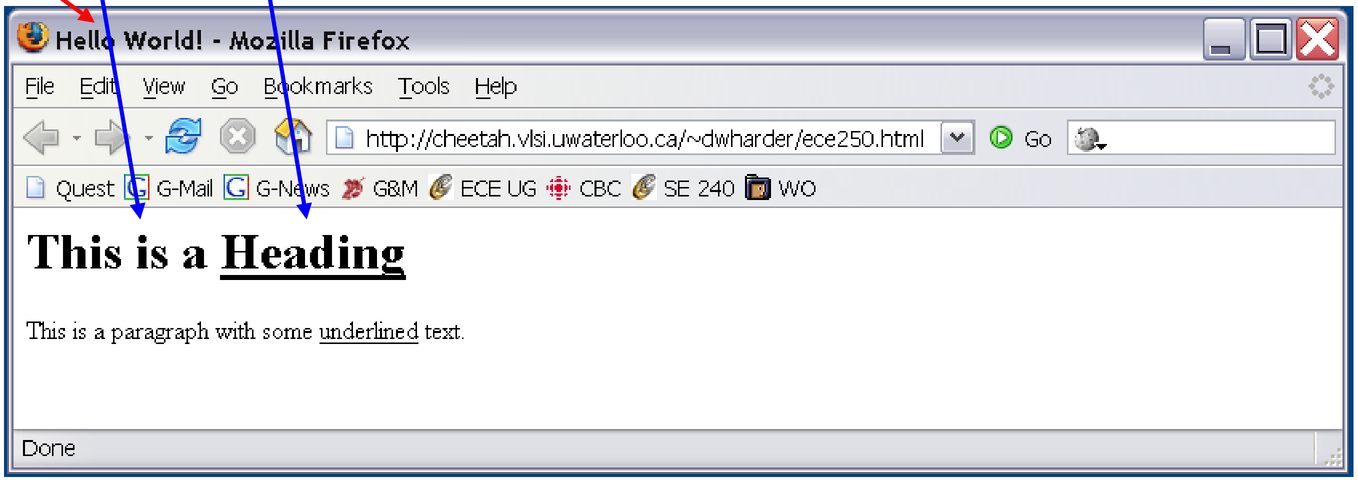
```
<html>
    <head>
        <title>Hello World!</title>
    </head>
    <body>
        <h1>This is a <u>Heading</u></h1>

        <p>This is a paragraph with some
        <u>underlined</u> text.</p>
    </body>
</html>
```



# Example: XHTML and CSS

Web browsers render this tree as a web page




Hello World! - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://cheetah.vlsi.uwaterloo.ca/~dwharder/ece250.html
Go
Quest G-Mail G-News G&M ECE UG CBC SE 240 WO
Done

This is a Heading

This is a paragraph with some underlined text.

# Example: XHTML and CSS

XML tags `<tag>...</tag>` must be nested

For example, to get the following effect:

1 2 3 **4 5 6 7 8 9**

you may use

```
<u>1 2 3 <b>4 5 6</b></u><b> 7 8 9</b>
```

You may not use:

```
<u>1 2 3 <b>4 5 6</u> 7 8 9</b>
```

# Example: XHTML and CSS

Cascading Style Sheets (CSS) make use of this tree structure to describe how HTML should be displayed

- For example:

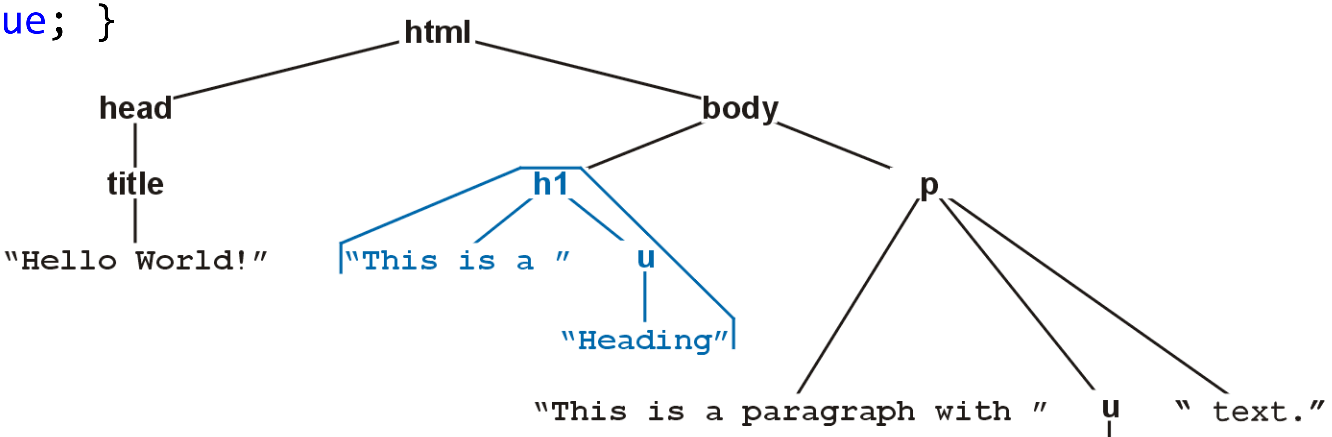
```
<style type="text/css">
   h1 { color:blue; }
</style>
```

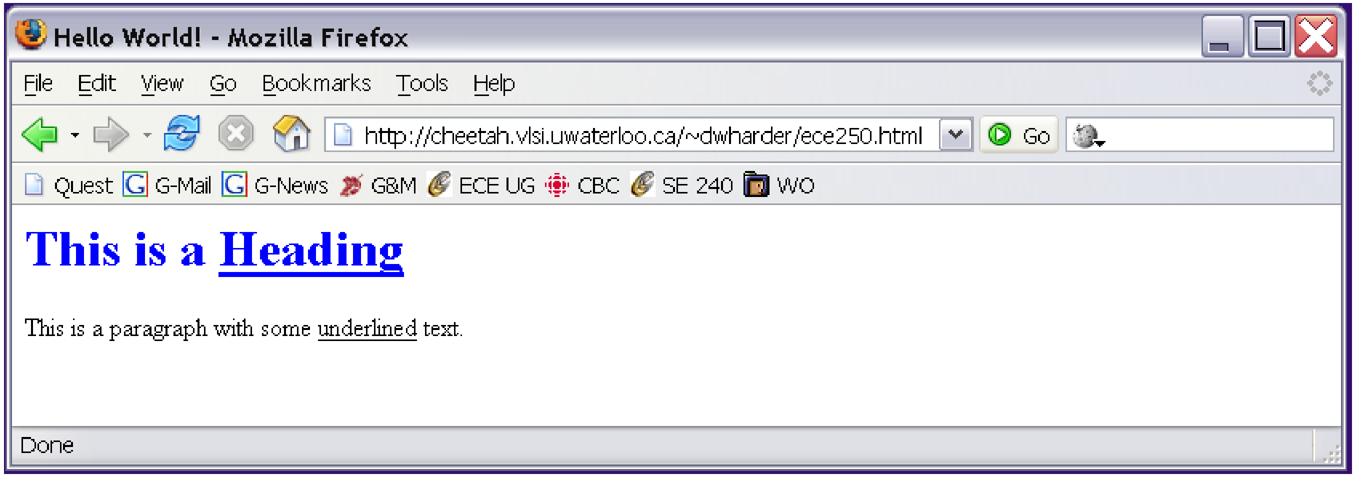
indicates all text/decorations descendant from an `h1` header should be blue

# Example: XHTML and CSS

For example, this style renders as follows:

```
<style type="text/css">
   h1 { color:blue; }
</style>
```




Hello World! - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://cheetah.vlsi.uwaterloo.ca/~dwharder/ece250.html
Go
Quest G-Mail G-News G&M ECE UG CBC SE 240 WO
Done

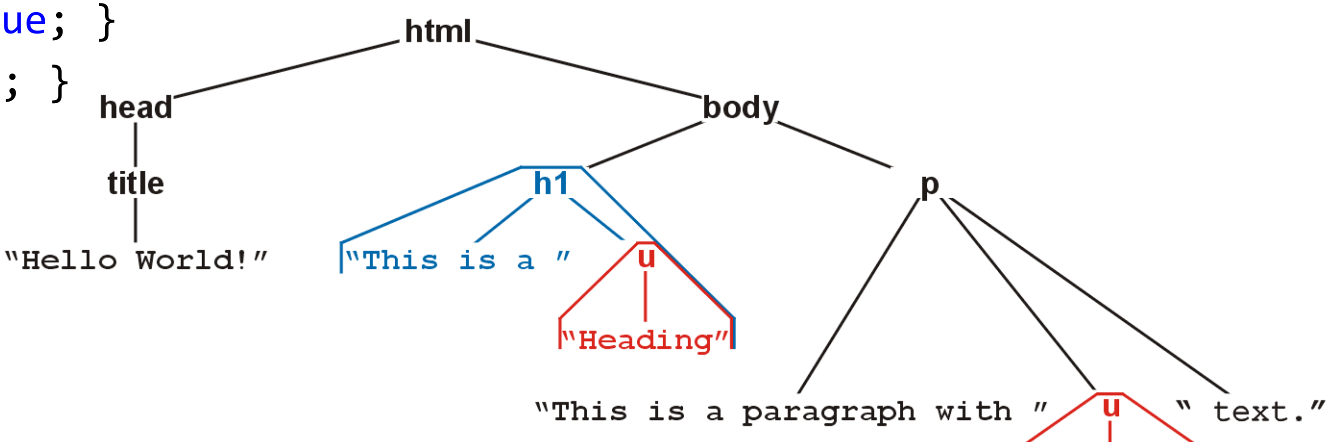
This is a Heading

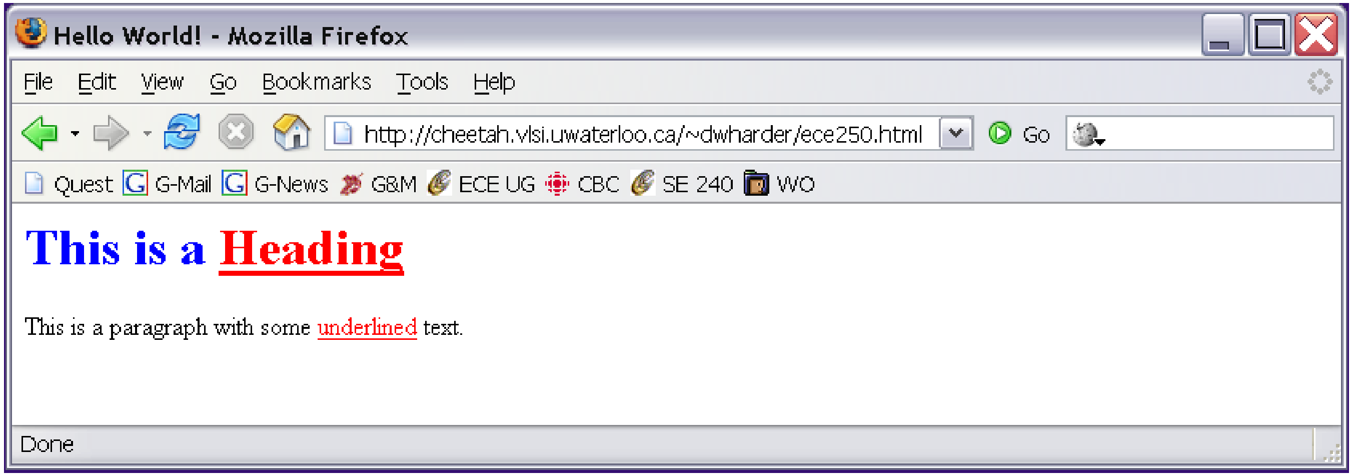
This is a paragraph with some underlined text.

# Example: XHTML and CSS

For example, this style renders as follows:

```
<style type="text/css">
   h1 { color:blue; }
   u { color:red; }
</style>
```




Hello World! - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://cheetah.vlsi.uwaterloo.ca/~dwharder/ece250.html
Go
Quest G-Mail G-News G&M ECE UG CBC SE 240 WO

# This is a Heading

This is a paragraph with some underlined text.

Done

# Example: XHTML and CSS

Suppose you don’t want underlined items in headers (`h1`) to be red

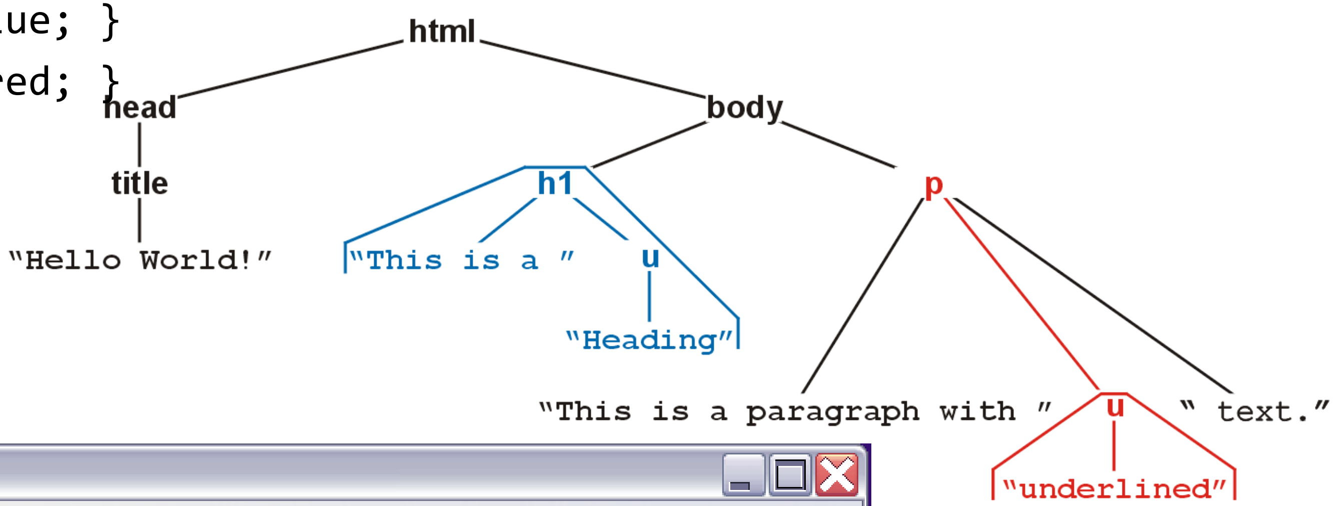
- More specifically, suppose you want any underlined text within paragraphs to be red

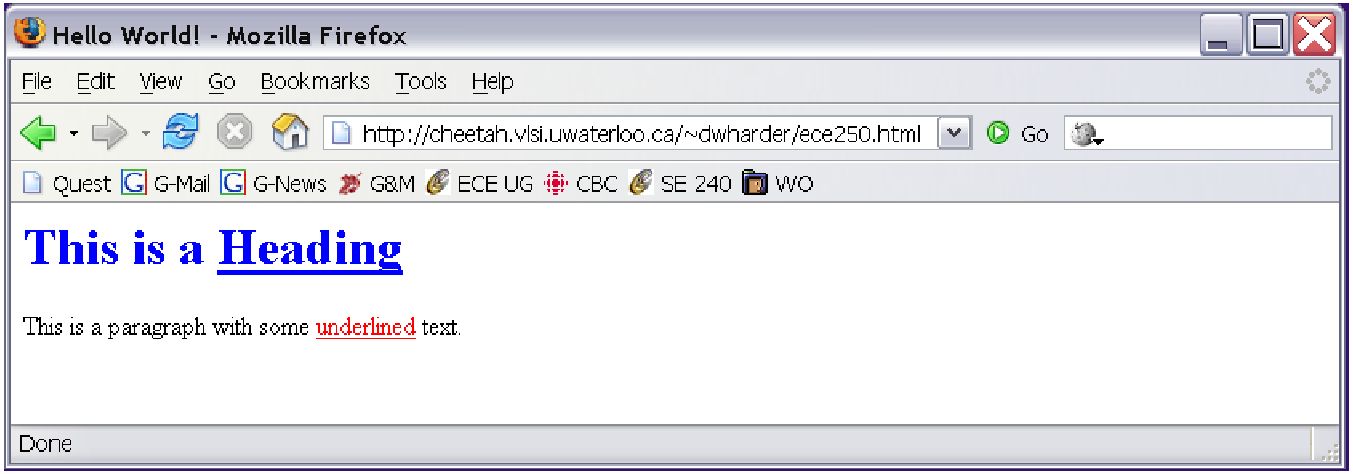
That is, you only want text marked as `<u>text</u>` to be underlined if it is a descendant of a `<p>` tag

# Example: XHTML and CSS

For example, this style renders as follows:

```
<style type="text/css">
   h1 { color:blue; }
  p u { color:red; }
</style>
```




Hello World! - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://cheetah.vlsi.uwaterloo.ca/~dwharder/ece250.html
Go
Quest G-Mail G-News G&M ECE UG CBC SE 240 WO
Done

This is a Heading

This is a paragraph with some underlined text.

# Example: XHTML and CSS

You can read the second style

```
<style type="text/css">
   h1 { color:blue; }
   p u { color:red; }
</style>
```

as saying “text/decorations descendant from the underlining tag (`<u>`) which itself is a descendant of a paragraph tag should be coloured red”

# Summary

In this topic, we have:

- Introduced the terminology used for the tree data structure
- Discussed various terms which may be used to describe the properties of a tree, including:
  - root node, leaf node
  - parent node, children, and siblings
  - ordered trees
  - paths, depth, and height
  - ancestors, descendants, and subtrees
- We looked at XHTML and CSS

# References

[1] Donald E. Knuth, *The Art of Computer Programming, Volume 1:  Fundamental Algorithms*, 3rd Ed., Addison Wesley, 1997, §2.2.1, p.238.