

# CS 191 L<sup>A</sup>T<sub>E</sub>X Notes

## Class 1

# Markup Language

- HTML stands for hypertext **markup language**
- a markup language is a system for embedding tags or codes into a plain text document
- these tags tell a renderer how to display the document's contents
- for example, the HTML on the left might be rendered by a browser to appear as seen on the right

```
<h1>A Heading</h1>
```

```
<p>This is a paragraph.</p>
```



# Markup Language

```
<h1>A Heading</h1>
```

```
<p>This is a paragraph.</p>
```

## A Heading

This is a paragraph.

- the tags tell the renderer **what kind** of text it is
- the tags do **not** tell the renderer **how to display** the text
- the renderer (with hints from CSS) makes its own decisions on how to display
  
- the author provides the content
- the renderer controls the appearance

# L<sup>A</sup>T<sub>E</sub>X

- HTML and CSS were invented for displaying documents in browsers
- they are the right tool for the job
- we need to use a tool for creating technical PDF documents in the computer science domain
- the right tool for that job is L<sup>A</sup>T<sub>E</sub>X
  
- L<sup>A</sup>T<sub>E</sub>X is pronounced with a long or short A in the first syllable, and the final syllable rhymes with “deck”
- the final letter is not the Roman character X, it is the Greek letter Chi, the equivalent of Roman K

# The WYSIWYG Problem

- WYSIWYG editors such as Word are menu systems
- the only commands you can give are the ones in the menus
- if there isn't a button for it, you can't do it
- a **complete** markup language, such as SGML or  $\text{\LaTeX}$ , allows the user complete control down to the individual pixels (if desired)

## Logical Design

- however, while you **can** micromanage your document down to the pixel
- you **should not** even think of doing that
- rather, you use  $\text{\LaTeX}$  to specify the **logical** design of your document
- then let  $\text{\LaTeX}$  make it look good
  
- by contrast Word makes it hard to specify the logical design
- and easy to mess around with local visual elements
- this results in a jumbled, disorganized document
- the reader thinks that the author's brain are similarly jumbled and disorganized
- $\text{\LaTeX}$  makes it easy to specify an organized, logical layout, freeing the author to concentrate on the content, not on the appearance

# A L<sup>A</sup>T<sub>E</sub>X Example

- use [overleaf.com](https://overleaf.com) for an example

# L<sup>A</sup>T<sub>E</sub>X Resources

- <https://www.latex-tutorial.com/tutorials/>
- read the sections in the left column below
- read the sections in the right column if you wish

<b>Required Sections</b>	<b>Optional Sections</b>
01 Your first document	00 Installation
03 Packages	02 Document structure
04 Math	13 Source code highlighting
05 Adding pictures	
09 Tables	
12 Drawing graphs (tikz)	