



COSC 122

Computer Fluency

HTML

Dr. Firas Moosvi

Key Points

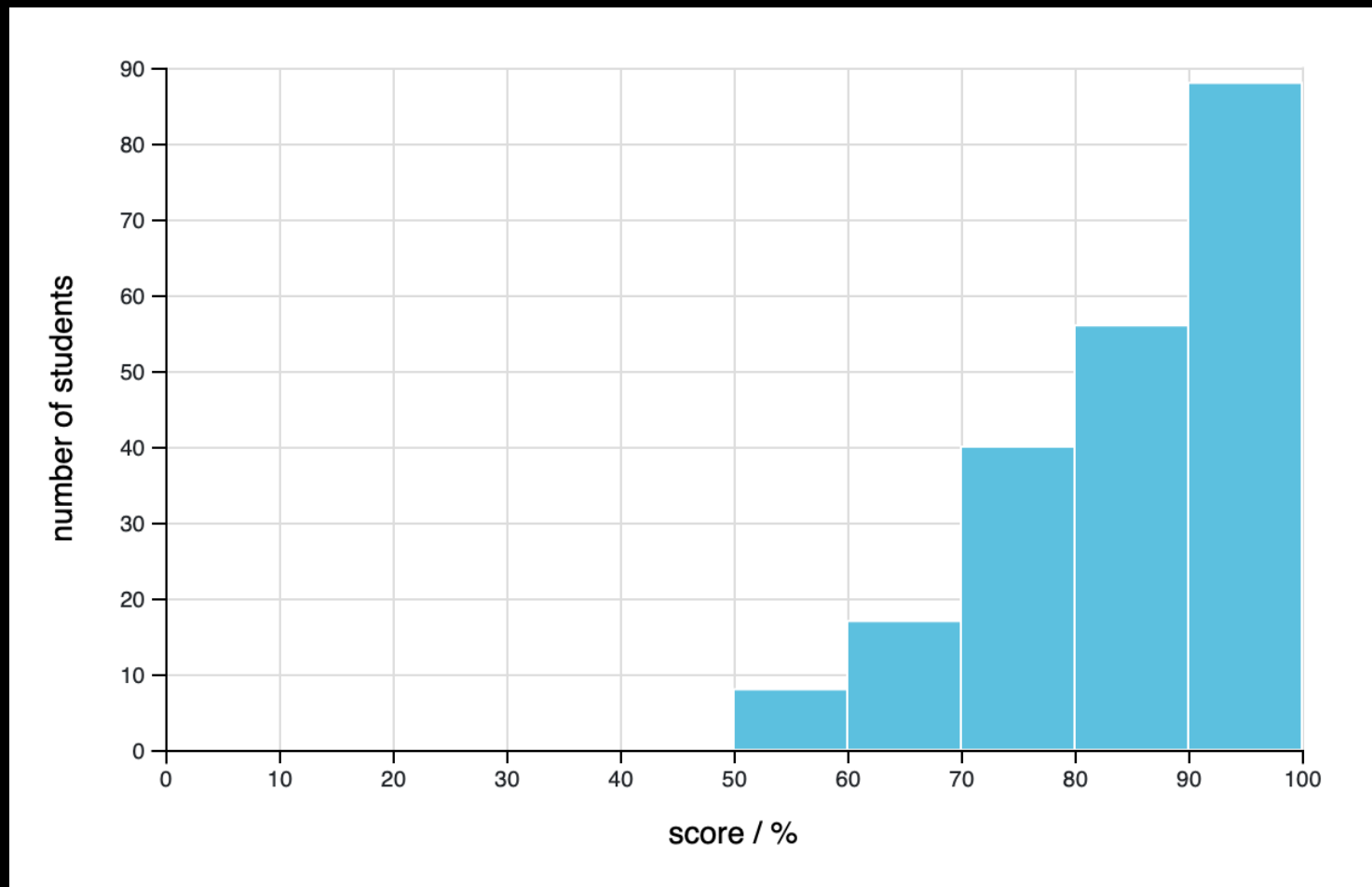
1) It is possible to add some HTML elements into Markdown files to “extend” Markdown.

2) Hypertext Markup Language (HTML) is the standard language for building complex web pages.

3) HTML is our second example of a language for communicating instructions to the computer (first was Markdown)

Announcements

1) Test 1 scores are now released – Average was 83%



Announcements

1) Test 1 scores are now released – Average was 83%

- Three “versions” of questions had a problem, and anyone who got those questions received 100%
- Not all “reports” of incorrect solutions were correct. Sometimes students arrived at the wrong answer incorrectly! Review your tests – ask questions on Ed Discussion or to your labs!

2) Bonus Test is going to be held on Friday during class time.

3) Lab 2 is released – start Early!

4) LL01 scores are released, LL02 will be graded shortly.



Hypertext Markup Language (HTML)

Hypertext Markup Language (HTML) is a language for describing how a web page appears in a web browser.

- ◆ HTML describes the layout of a document including fonts, text style, image placement, and hypertext links.

An HTML document looks like a regular text document except that it contains **tags** which are words or abbreviations enclosed in angle brackets: `<` and `>`.

- ◆ Each tag controls some appearance of the web page.
- ◆ In HTML 5, tags are not case-sensitive.
 - We will use lower case as convention.
- ◆ Tags **usually** come in pairs such as:

```
<p>Hello world!</p>
```

HTML "Hello World!" Example

Hello World using Markdown

Hello World



Markdown Syntax

```
1 # Hello World using Markdown
2
3 Hello World
```

HTML Source Code

A screenshot of a web browser window in "view-source" mode. The title bar says "Hello World using HTML" and "view-source:https://people.ok.ubc.ca/rlawrenc/teaching/122/Notes/01/HelloWorld.html". The address bar shows "view-source:https://people.ok.ubc.ca/rlawrenc/teaching/122/Notes/01/HelloWorld.html". The main content area displays the following HTML code:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Hello World using HTML</title>
5 </head>
6
7 <body>
8
9 <p>Hello world!</p>
10
11 </body>
12 </html>
```

HTML "Hello World!" Example (2)

```
<!DOCTYPE html>
```

```
<html> ← Start of document
```

```
<head>
```

```
<title>Hello World using HTML</title>
```

```
</head>
```

Beginning material
such as title in head

```
<body>
```

```
<p>Hello world!</p> ← Text formatted using  
paragraph <P> tags.
```

Content of document
goes in body

```
</body>
```

```
</html> ← End of document
```

Markdown Syntax - Headings

Headings

To create a heading, add number signs (#) in front of a word or phrase. The number of number signs you use should correspond to the heading level. For example, to create a heading level three (<h3>), use three number signs (e.g., `### My Header`).

Markdown	HTML	Rendered Output
<code># Heading level 1</code>	<code><h1>Heading level 1</h1></code>	Heading level 1
<code>## Heading level 2</code>	<code><h2>Heading level 2</h2></code>	Heading level 2
<code>### Heading level 3</code>	<code><h3>Heading level 3</h3></code>	Heading level 3
<code>#### Heading level 4</code>	<code><h4>Heading level 4</h4></code>	Heading level 4
<code>##### Heading level 5</code>	<code><h5>Heading level 5</h5></code>	Heading level 5
<code>##### Heading level 6</code>	<code><h6>Heading level 6</h6></code>	Heading level 6

Markdown Syntax – Bold Text

Bold

To bold text, add two asterisks or underscores before and after a word or phrase. To bold the middle of a word for emphasis, add two asterisks without spaces around the letters.

Markdown	HTML	Rendered Output
I just love <code>**bold text**</code> .	I just love <code>bold text</code> .	I just love bold text .
I just love <code>__bold text__</code> .	I just love <code>bold text</code> .	I just love bold text .
Love <code>**is**</code> bold	Love <code>is</code> bold	Love is bold

Markdown Syntax - Headings

Italic

To italicize text, add one asterisk or underscore before and after a word or phrase. To italicize the middle of a word for emphasis, add one asterisk without spaces around the letters.

Markdown	HTML	Rendered Output
Italicized text is the <code>*cat's meow*</code> .	Italicized text is the <code>cat's meow</code> .	Italicized text is the <i>cat's meow</i> .
Italicized text is the <code>_cat's meow_</code> .	Italicized text is the <code>cat's meow</code> .	Italicized text is the <i>cat's meow</i> .
<code>A*cat*meow</code>	<code>Acatmeow</code>	<i>Acatmeow</i>

Markdown Syntax – Ordered Lists

Ordered Lists

To create an ordered list, add line items with numbers followed by periods. The numbers don't have to be in numerical order, but the list should start with the number one.

Markdown	HTML	Rendered Output
<pre>1. First item 2. Second item 3. Third item 4. Fourth item</pre>	<pre> First item Second item Third item Fourth item </pre>	<pre>1. First item 2. Second item 3. Third item 4. Fourth item</pre>
<pre>1. First item 1. Second item 1. Third item 1. Fourth item</pre>	<pre> First item Second item Third item Fourth item </pre>	<pre>1. First item 2. Second item 3. Third item 4. Fourth item</pre>

Markdown Syntax – Ordered Lists

Ordered Lists

To create an ordered list, add line items with numbers followed by periods. The numbers don't have to be in numerical order, but the list should start with the number one.

Markdown	HTML	Rendered Output
<pre>1. First item 2. Second item 3. Third item 1. Indented item 2. Indented item 4. Fourth item</pre>	<pre> First item Second item Third item Indented item Indented item Fourth item </pre>	<pre>1. First item 2. Second item 3. Third item 1. Indented item 2. Indented item 4. Fourth item</pre>

Markdown Syntax – Unordered Lists

Unordered Lists

To create an unordered list, add dashes (-), asterisks (*), or plus signs (+) in front of line items. Indent one or more items to create a nested list.

Markdown	HTML	Rendered Output
<pre>- First item - Second item - Third item - Fourth item</pre>	<pre> First item Second item Third item Fourth item </pre>	<ul style="list-style-type: none">• First item• Second item• Third item• Fourth item
<pre>* First item * Second item * Third item * Fourth item</pre>	<pre> First item Second item Third item Fourth item </pre>	<ul style="list-style-type: none">• First item• Second item• Third item• Fourth item

Markdown Syntax – Unordered Lists

Unordered Lists

To create an unordered list, add dashes (-), asterisks (*), or plus signs (+) in front of line items. Indent one or more items to create a nested list.

Markdown	HTML	Rendered Output
<pre>- First item - Second item - Third item - Indented item - Indented item - Fourth item</pre>	<pre> First item Second item Third item Indented item Indented item Fourth item </pre>	<ul style="list-style-type: none">• First item• Second item• Third item<ul style="list-style-type: none">◦ Indented item◦ Indented item• Fourth item

Basic Formatting Tags

To put text in a *paragraph* use the `<p>` `</p>` tags:

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

To make text *bold* use the `` `` tags:

```
<b>This text is bold.</b>
```

To make text *italic* use the `<i>` `</i>` tags:

```
<i>This text is in italics.</i>
```

To identify important text use the `` `` tags:

```
<strong>This text is in strong format.</strong>
```

To emphasize text use the `` `` tags:

```
<em>This text is emphasized.</em>
```

HTML Formatting Example

```
<!DOCTYPE html>

<html>
<head>
<title>Basic Formatting using HTML</title>
</head>

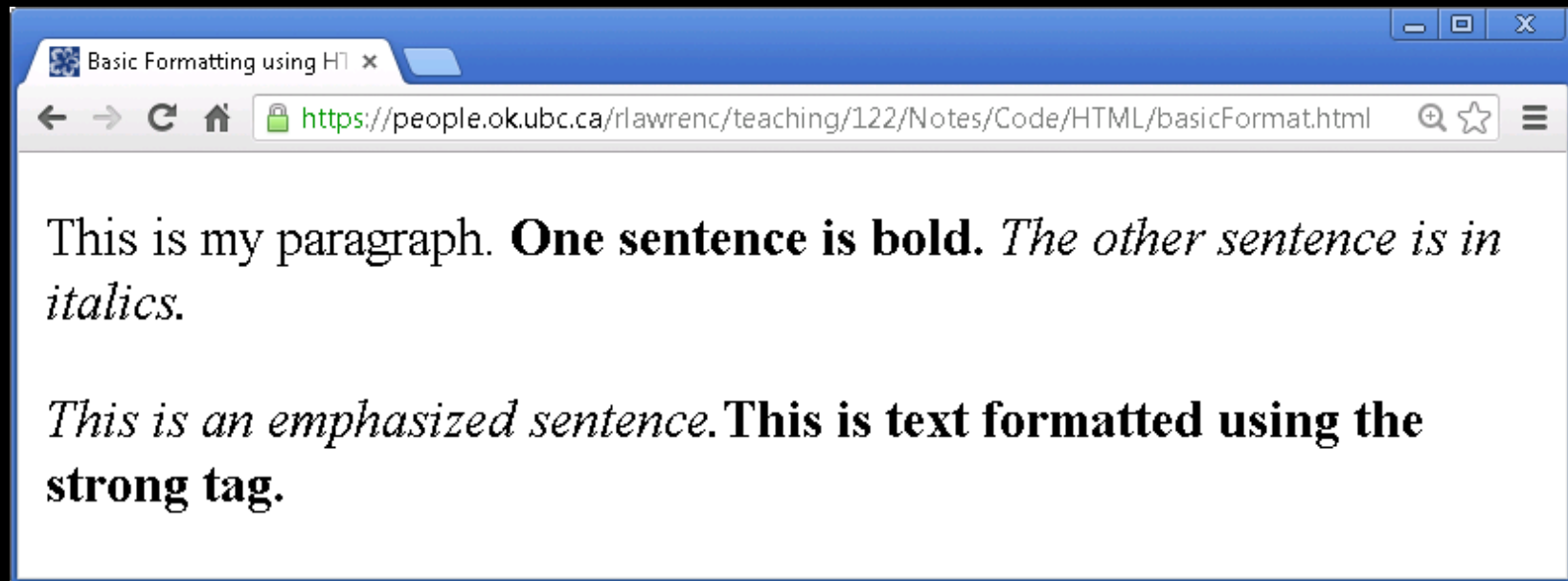
<body>

<p>This is my paragraph.  <b>One sentence is bold.</b>
<i>The other sentence is in italics.</i></p>

<p><em>This is an emphasized sentence.</em><strong>This is text
formatted using the strong tag.</strong></p>

</body>
</html>
```


HTML Formatting Example (2)



More Formatting Tags

You can apply more than one formatting at a time:

```
<p><b><i>This text is in bold and italics.</i></b></p>
```

There are 6 levels of heading defined `<h1>`, `<h2>`, .. `<h6>`.
Each heading creates a new line and displays in a large font.

```
<h1>Largest heading</h1>  
<h2>Next largest heading</h2>  
<h6>Lowest level heading</h6>
```

Use `<hr>` to put a horizontal line in the document.

Use `
` to put a line break.

◆ Note that these last two do not have a closing tag.

Display vs. HTML Format

When displaying an HTML document, a web browser **ignores** white space. **White space is considered spaces, tabs, and newlines.** Multiple spaces and newlines are replaced with a single space when displayed (unless the `<pre></pre>` tags are used).

Since white space is ignored, it is advisable to make your HTML document easier to read and edit by inserting spaces and blank lines.

Remember the web browser uses the tags to determine how to display the document, not what it looks like in your editor!

Special Symbols

Since the `<` and `>` are special (reserved) symbols in the HTML language, we need a way to use them in our documents.

The `&` (ampersand) is the escape symbol that tells HTML a special character is required. Terminate with a `;` (semi-colon).

Common characters:

<code>&lt;</code>	<code><</code>
<code>&gt;</code>	<code>></code>
<code>&amp;</code>	<code>&</code>
<code>&copy;</code>	<code>©</code>
<code>&eacute;</code>	<code>é</code>
<code>&ntilde;</code>	<code>ñ</code>
<code>&nbsp;</code>	<code>non-breaking space</code>

Text Alignment

You can left, right, or center a paragraph text by:

```
<p style="text-align:center">This text is centered.</p>
```

In general, **attributes** of tags are specified by providing their name and their value.

```
<p>Regular font.<br></p>
```

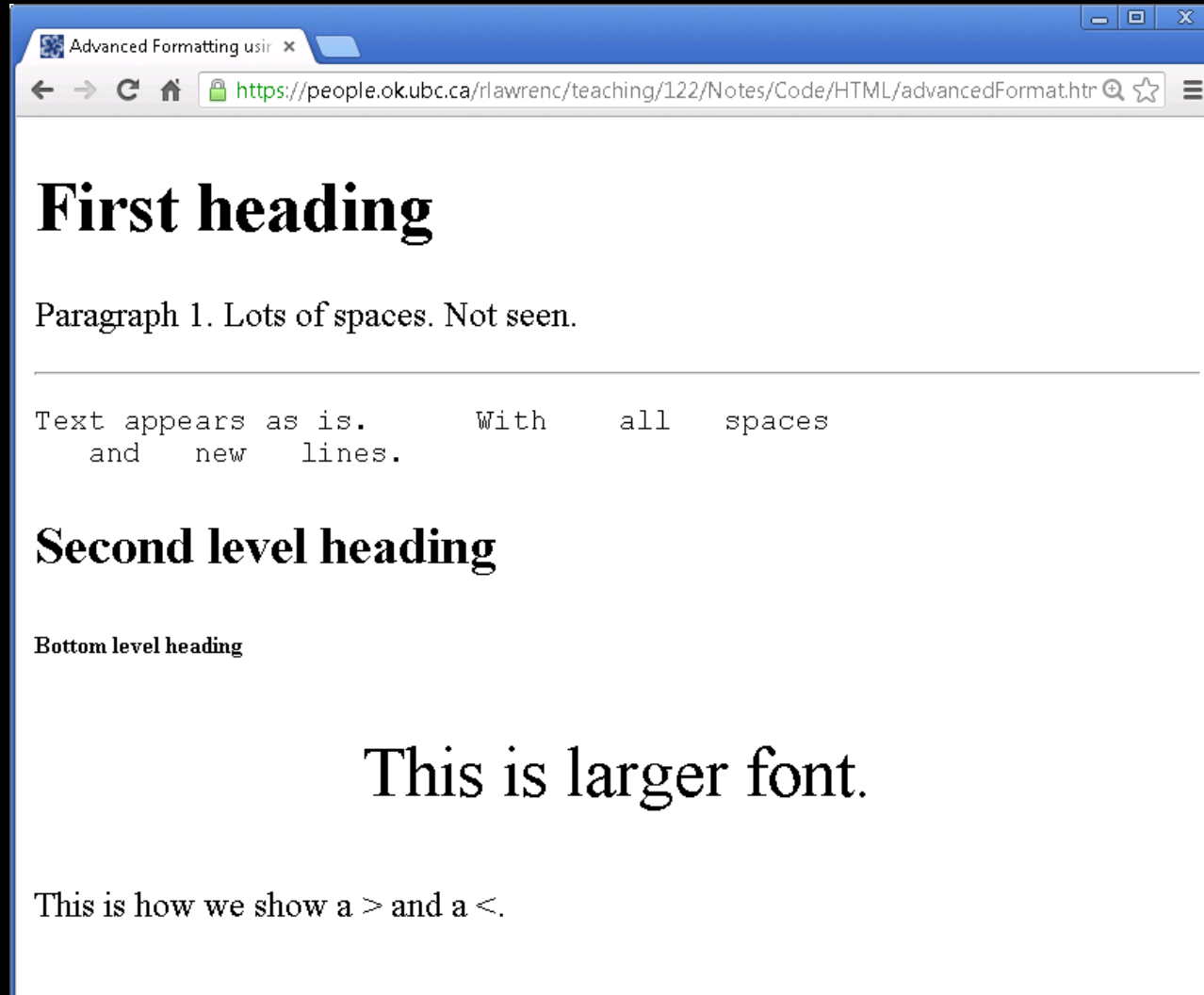
```
<p style="font-size:200%">Font twice as large!</p>
```

HTML Advanced Formatting Example

```
<!DOCTYPE html>
<html>
<head><title>Advanced Formatting using HTML</title></head>
<body>
<b>First heading</b>
<p>Paragraph 1.      Lots of spaces.      Not seen.</p>
<hr>
<pre>
Text appears as is.      With      all      spaces
    and      new      lines.
</pre>
<h2>Second level heading</h2>

<h6>Bottom level heading</h6>
<p style="text-align:center;font-size:200%">This is larger
font.</p>
<p>This is how we show a &gt; and a &lt;.</p>
</body>
</html>
```

HTML Advanced Formatting Example (2)



General Syntax Rules: Comments

Comments are used by the programmer to document and explain the code. Comments are ignored by the computer.

HTML comments use the syntax "**<!--**" for the start of the comment and "**-->**" for the end of the comment.

Example:

```
<!-- This is a HTML comment -->  
<!-- This is a HTML comment  
      that crosses  
      multiple lines -->
```


CQ 3.1- HTML Tags

Question: Select **one** of the tags that do not have a matching closing tag.

A) br

B) h1

C) hr

D) p

CQ 3.2- HTML Tags – No Formatting

Question: Which one of these tags will display text “as-is” in your document with line breaks, spaces, etc.?

A) p

B) hr

C) pre

D) b

CQ 3.3- HTML Tags – Reserved Symbols

Question: Which one of these symbols is **not** reserved?

A) <

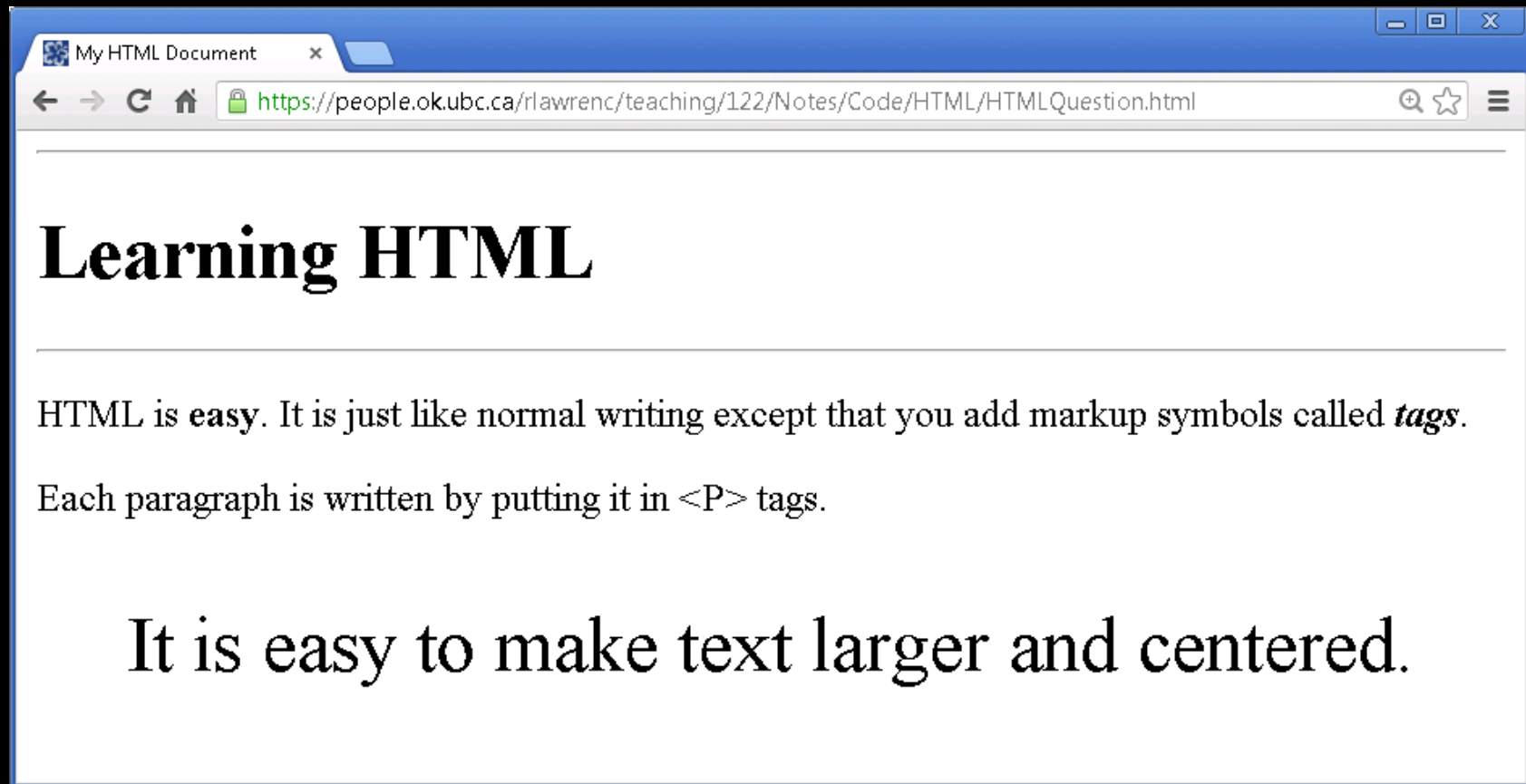
B) ,

C) >

D) &

Activity: Practice Question 1: Basics

Write the HTML document that looks like this:





Marking Links With Anchor Tags

Anchor tags are used to create hyperlinks in the document.

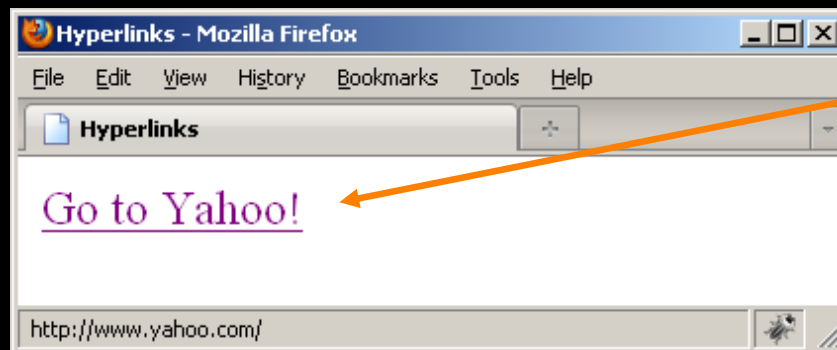
An anchor tag has two components:

- ◆ **Anchor text** – the text in the document that is highlighted
- ◆ **Hyperlink reference** – address of web page to link to

Example:

```
<a href="http://www.yahoo.com">Go to Yahoo!</a>
```

Hyperlink
(where to go to)



Text the user sees
to click on

Specifying A Hyperlink Location

Absolute and Relative Paths

The location where the user goes to when clicking on the link may be given as a complete **absolute** URL:

```
<a href="http://www.yahoo.com">Go to Yahoo!</a>
```

or **relative** to the current location:

```
<a href="mydir/helloWorld.html">Go to Hello World in mydir</a>
```

Use an absolute URL when the page is on a different server.

Use a relative path when the page is on the same machine. The path depends on the current page location.

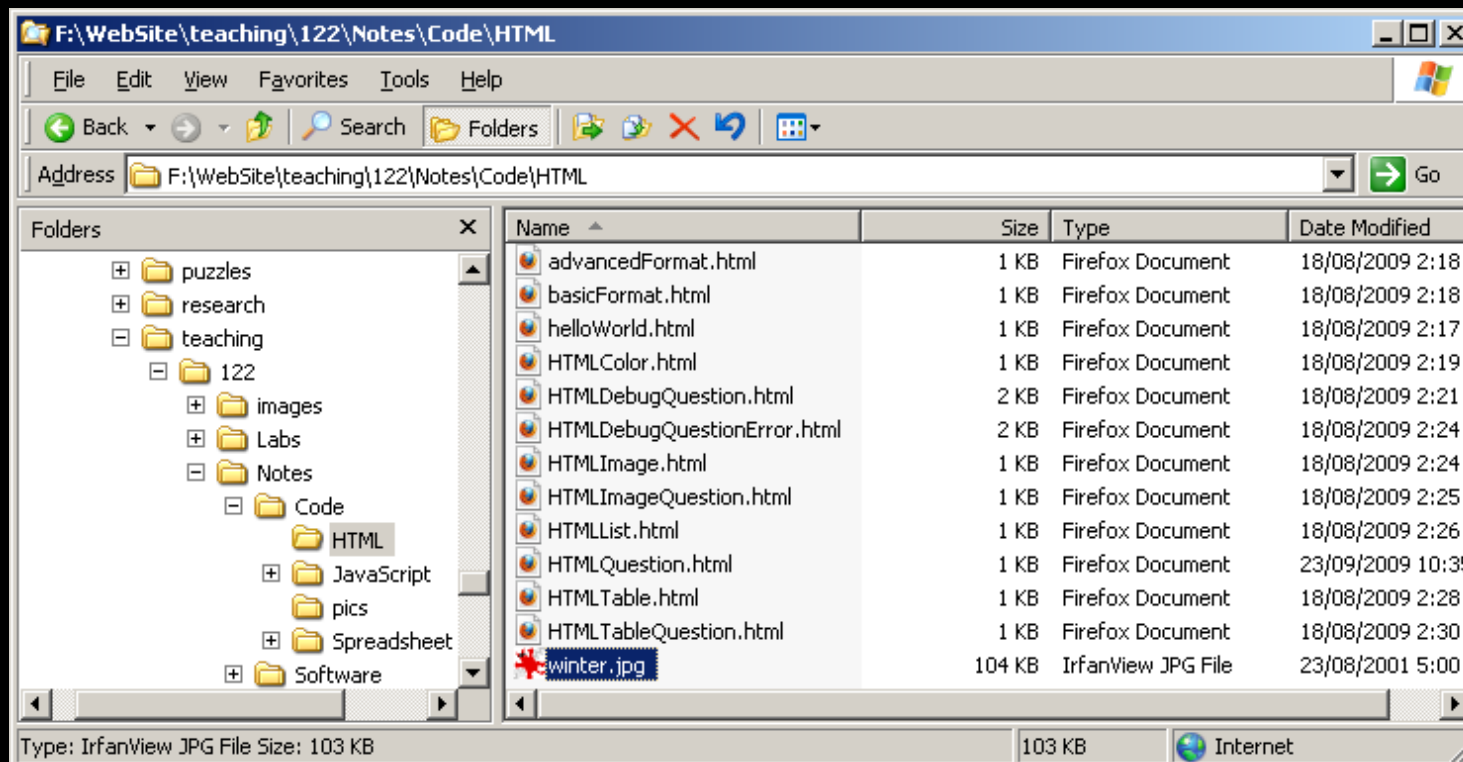
- ◆ Use ".." to navigate to the directory above your current location.

Analogy: If you give someone directions to the Science building, those directions will depend on where you start from!

Absolute and Relative Paths Question 2

Given these files, specify the location of the file `winter.jpg`:

- absolutely
- relative to directory `122` (directory `122` is the start point)
- relative to directory `pics`



Including Pictures With Image Tags

Images can be shown in a document with the image `` tag:

```

```

```

```

Notes:

- ◆ `src` stands for source and is either an absolute or relative path to a picture file.
- ◆ A picture may be in many different formats. Common formats:
 - GIF: Graphic Interchange Format
 - JPEG: Joint Photographic Experts Group
 - PNG: Portable Network Graphics
- ◆ The file extension (`.gif`, `.jpg`, `.png`) tells the browser which format the image is stored in.

Positioning the Image in the Document

By default, images are inserted in the page at the point where the tag is specified in the HTML, and the text lines up with the bottom of the image.

The align attribute can line up image with the top of the line of text or the bottom.

Align left or right attribute puts the image on the side of the browser window and the text flows around it.

To put image on separate line, enclose within paragraph tags.

Advanced: Images and Links Together

You can create a hyperlink on an image so when the user clicks on the image, they go to the desired location.

Example:

```
<a href="http://www.google.ca">  
  
</a>
```

- ◆ This example shows an image retrieved from Google's web site and will go to the web site when the image is clicked.
- ◆ Note that we could have sent the user to any site, not just the Google site where the image came from.

HTML Image and Link Example

```
<html>
<head><title>Images and Links in HTML</title></head>
<body>

<p>The image is placed 
in the text.</p>

<p>This is <a href="http://www.google.ca">

Google's Image</a>.</p>

<p>
We have wrapped some text around this wonderful winter scene and
resized it so that it is smaller than its original form.</p>

</body>
</html>
```

HTML Image and Link Example (2)



The image is placed  in the text.



This is [Google's Image.](#)

 We have wrapped some text around this wonderful winter scene and resized it so that it is smaller than its original form.

Handling Color

Color is used for both background and text. A color is specified either by name (red, yellow, orange) or by *hexadecimal* RGB color numbers.

- ◆ `background-color` defines the background color.
- ◆ `color` is text color.

Examples:

```
<body style="background-color:silver;color:yellow">  
<p style="color:red">Red font</p>  
<p style="color:#FF8E2A">Orange font</p>
```

To set link color (put this before the body tag):

```
<style>  
a:link { color: orange }  
a:visited { color: green }  
a:active { color: orange }  
</style>
```

Advanced: Specifying Color by Number

When a color is specified by a hexadecimal number it consists of three numbers from 0 to 255 representing the intensity of red, green, and blue respectively.

- ◆ However, instead of using decimal numbers (base 10), hexadecimal numbers are used (base 16) which have the digits 0 to 9 plus A (10), B (11), C (12), D (13), E (14), and F (15).

Examples:

Color	RGB Intensity	Hexadecimal
Black	(0, 0, 0)	#000000
White	(255, 255, 255)	#FFFFFF
Red	(255, 0, 0)	#FF0000
Green	(0, 255, 0)	#00FF00
Blue	(0, 0, 255)	#0000FF
Orange	(255, 142, 42)	#FF8E2A
Purple	(147, 112, 219)	#9370DB
Yellow	(255, 255, 0)	#FFFF00

CQ 3.4- HTML Anchor Tag

Question: Which one of these anchor tags is correct?

- a) `http://www.yahoo.com`
- b) `My Home Page`
- c) `Yahoo`
- d) ``

CQ 3.5- HTML Image Tag

Question: Which one of these image tags is correct?

- a) ``
- b) `picture.gif`
- c) ``
- d) ``

HTML Practice Question 2: Images



- ◆ **The background is black. Image and link locations:**
 - greenball.gif – relative path 4 directories up then in images directory
 - CBC logo – <http://www.cbc.ca/logo.gif>
 - helloWorld link – [helloWorld.html](#) is in current directory
 - CBC link – <http://www.cbc.ca>

Lists in HTML

It is possible to create both bulleted and numbered lists.

For a bulleted list, use the tags `` and ``:

```
<ul>
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
</ul>
```

For a numbered list, use the tags `` and ``:

```
<ol>
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
</ol>
```

Advanced Lists

You can nest lists inside each other to produce sublists:

```
<ul>
  <li>Item 1
    <ol>
      <li>Subitem 1.1</li>
      <li>Subitem 1.2</li>
    </ol>
  </li>
</ul>
```

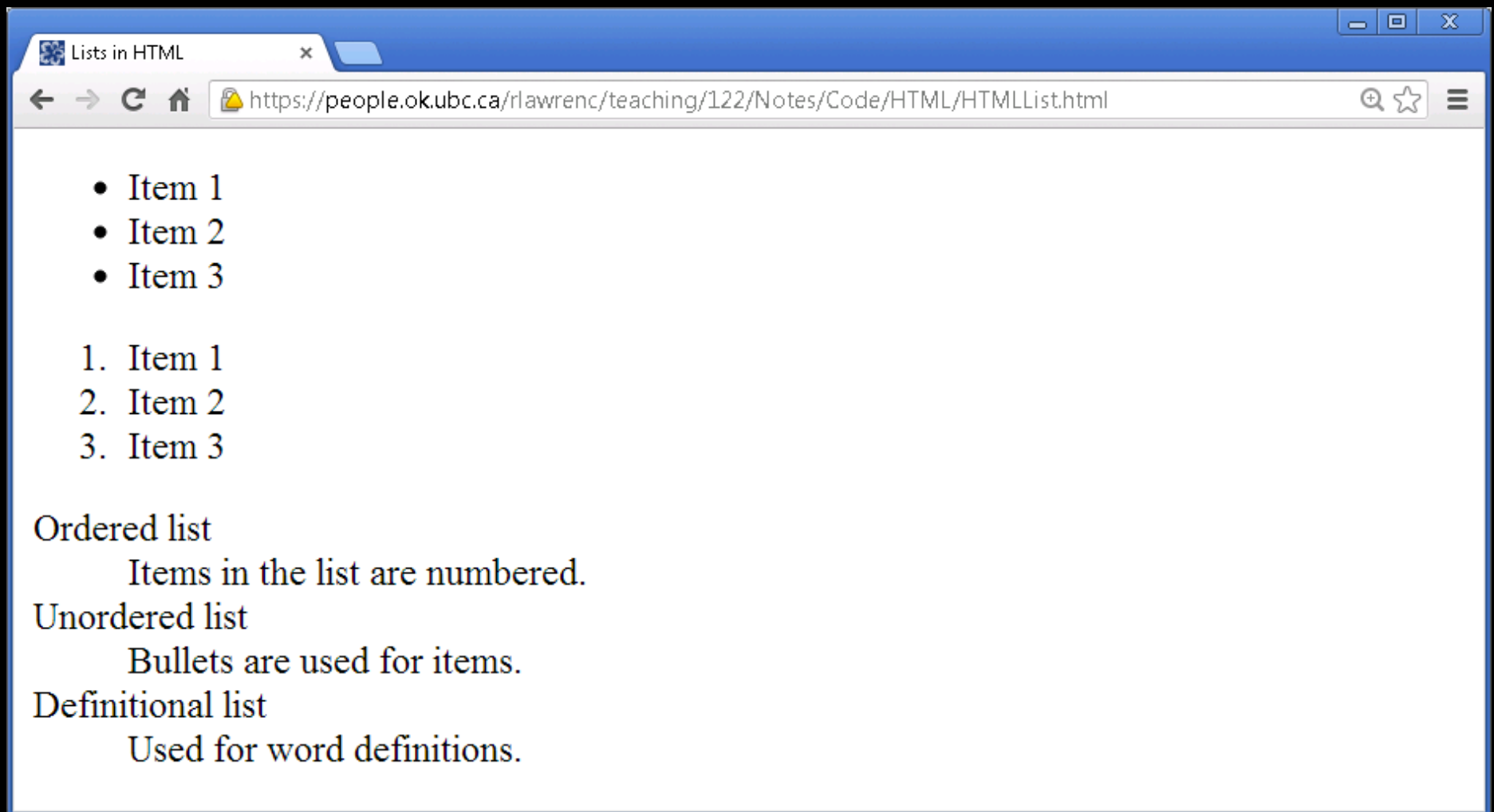
Another type of list is the **definitional list**:

- ◆ `<d1>` and `</d1>` tags begin and end the list
- ◆ `<dt>` and `</dt>` surround the terms to be defined
- ◆ `<dd>` and `</dd>` surround the definitions

HTML List Example

```
<html><head><title>Lists in HTML</title></head>
<body>
<ul>
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
</ul>
<ol>
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
</ol>
<dl>
  <dt>Ordered list</dt><dd>Items in the list are numbered.</dd>
  <dt>Unordered list</dt><dd>Bullets are used for items.</dd>
  <dt>Definitional list</dt><dd>Used for word definitions.</dd>
</dl>
</body>
</html>
```

HTML List Example (2)



The screenshot shows a web browser window with the title "Lists in HTML" and the URL "https://people.ok.ubc.ca/rlawrenc/teaching/122/Notes/Code/HTML/HTMList.html". The browser displays three types of lists:

- Unordered list:
 - Item 1
 - Item 2
 - Item 3
- Ordered list:
 - 1. Item 1
 - 2. Item 2
 - 3. Item 3
- Definitional list:
 - Ordered list
 - Items in the list are numbered.
 - Unordered list
 - Bullets are used for items.

Tables in HTML

Data can be displayed in tables using the `<table>/table>` tags. Rows are enclosed in table row `<tr>/tr>` tags, and each cell is denoted using table data `<td>/td>` tags.

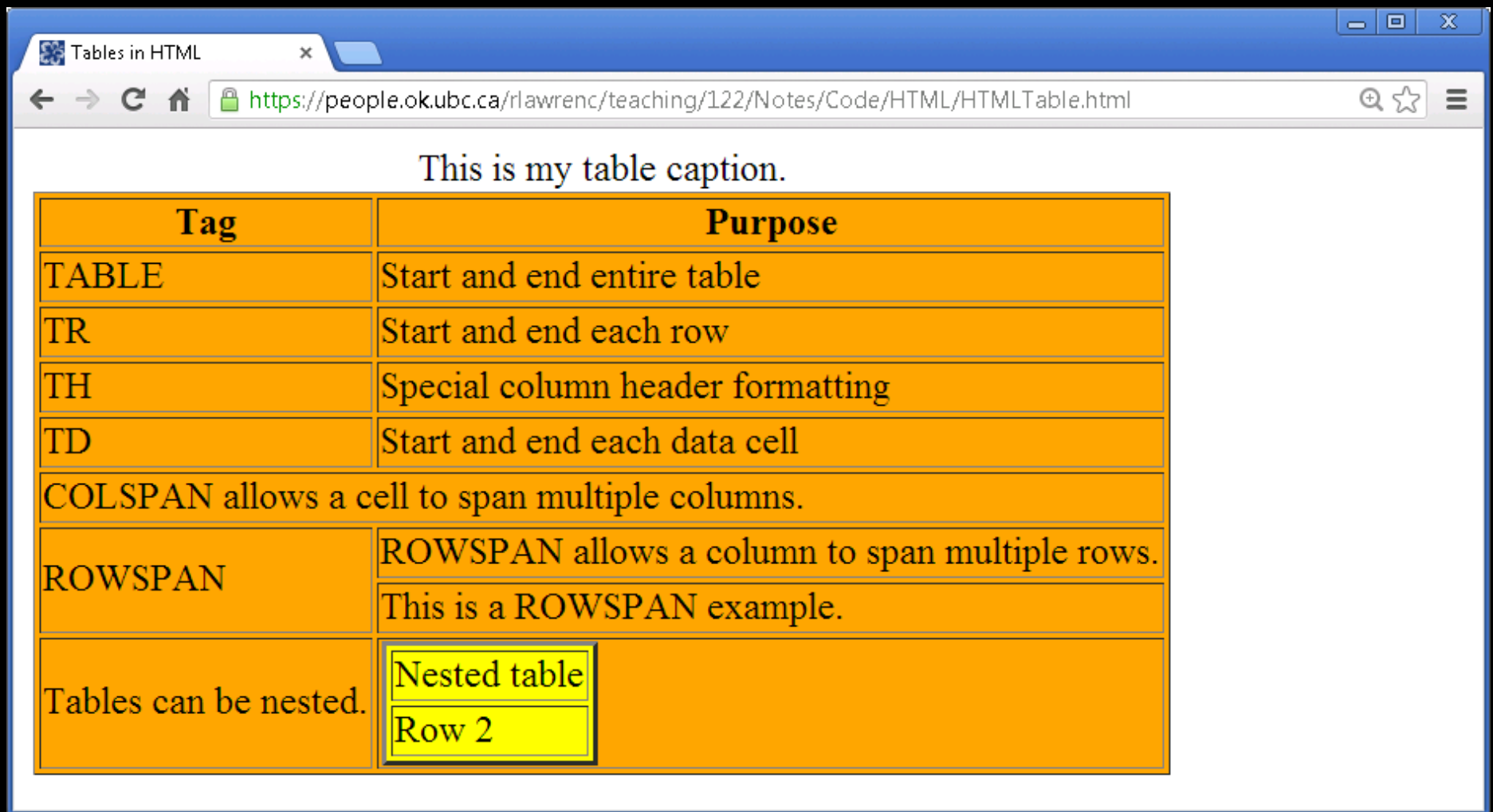
A table may have a caption centered at the top of the table by using the `<caption>/caption>` tags.

A header row may be created using the `<th>/th>` tags.

HTML Table Example

```
<html><head><title>Tables in HTML</title></head><body>
<table border="1" style="background-color:orange" <— Table with
  <caption>This is my table caption.</caption> border
  <tr><th>Tag</th><th>Purpose</th></tr> <— Header row
  <tr><td>TABLE</td><td>Start and end entire table</td></tr>
  <tr><td>TR</td><td>Start and end each row</td></tr>
  <tr><td>TH</td><td>Special column header formatting</td></tr>
  <tr><td>td</td><td>Start and end each data cell</td></tr>
  <tr><td colspan="2">COLSPAN allows a cell to span multiple
columns.</td></tr>
  <tr><td rowspan="2">ROWSPAN</td><td>ROWSPAN allows a column
to span multiple rows.</td></tr>
  <tr><td>This is a ROWSPAN example.</td></tr>
  <tr><td>Tables can be nested.</td><td>
    <table border="2" style="background-color:yellow" <—
      <tr><td>Nested table</td></tr>
      <tr><td>Row 2</td></tr></table>
    </td></tr>
  </table></body></html>
```

HTML Table Example (2)



The screenshot shows a web browser window with the title "Tables in HTML" and the URL <https://people.ok.ubc.ca/rlawrenc/teaching/122/Notes/Code/HTML/HTMLTable.html>. The page content includes a caption "This is my table caption." followed by a table with two columns: "Tag" and "Purpose". The table lists HTML tags and their purposes, including examples of COLSPAN, ROWSPAN, and a nested table.

This is my table caption.

Tag	Purpose		
TABLE	Start and end entire table		
TR	Start and end each row		
TH	Special column header formatting		
TD	Start and end each data cell		
COLSPAN allows a cell to span multiple columns.			
ROWSPAN	ROWSPAN allows a column to span multiple rows.		
	This is a ROWSPAN example.		
Tables can be nested.	<table border="1"><tr><td>Nested table</td></tr><tr><td>Row 2</td></tr></table>	Nested table	Row 2
Nested table			
Row 2			

Where can I use HTML?

HTML is everywhere on the Internet: UBC Connect, Facebook, Yahoo, Google, etc.

All your most popular websites have HTML as a base (and maybe other languages on top of them.)

You can see the HTML source for any page in a browser by selecting `View->Page Source` in Firefox/Chrome or `View->Source` in Internet Explorer.

The idea of *markup* in HTML is common to many environments and languages.

Advanced: Style Attribute

The `style` attribute can be added to a tag to control its appearance. Different settings are separated by semi-colons.

Example:

```
<body style = "background-color:black; color: white;">
```

Some common style settings:

- ◆ `background-color` e.g. `background-color:yellow`
- ◆ `font-family` e.g. `font-family:"Times New Roman",Serif;`
- ◆ `font-style` e.g. `font-style:italic`
- ◆ `font-size` e.g. `font-size:100%`
- ◆ `color` e.g. `color:red;`
- ◆ `text-align` e.g. `text-align:center`

Advanced: Cascading Style Sheets

Cascading style sheets (CSS) is a language for controlling the appearance of web pages, especially color, layout, and fonts.

How it works:

- ◆ In a CSS source, you define the markup tag and its formatting.
- ◆ When that tag is used in your HTML page, the formatting is automatically applied. This makes changes easier!

Example:

```
<html><head><title>Using CSS</title></head>
<style type="text/css">
  body { font-family: "Times New Roman"; color: purple;}
  h1 { font-family: Helvetica; color: green;}
</style>
<body>
<h1>Formatted heading</h1>
<p>Regular text</p>
</body></html>
```

Advanced: Three Types of Selectors

By element - Apply to all instances of a particular element:

```
h1 { font-family: Helvetica; color: red;}
```

Use:

```
<h1>This will be red</h1>
```

By id - Apply to all content with a specific id:

```
#section {text-align: left; background-color: blue;}
```

Use:

```
<div id="section"><h1>Heading</h1><p>Text..</p></div>
```

By class - Apply to specified instances of any tag:

```
h1 { color: green;}
```

```
h1.red { color: red;}
```

Use:

```
<h1>This will be in green</h1> <h1 class="red">Red</h1>
```

CQ 3.6- HTML List Tag

Question: Which one of these HTML code fragments will produce a list like below?

```
1. Item 1  
2. Item 2
```

a)

```
<ol>  
  <li>Item 1</li>  
  <li>Item 2</li>  
</OL>
```

c)

```
<ol>  
  <il>Item 1</il>  
  <il>Item 2</il>  
</ol>
```

b)

```
<ul>  
  <li>Item 1</li>  
  <li>Item 2</li>  
</ul>
```

d)

```
<ul>  
  <ol>Item 1</ol>  
  <ol>Item 2</ol>  
</ul>
```

CQ 3.7- HTML Table Tag

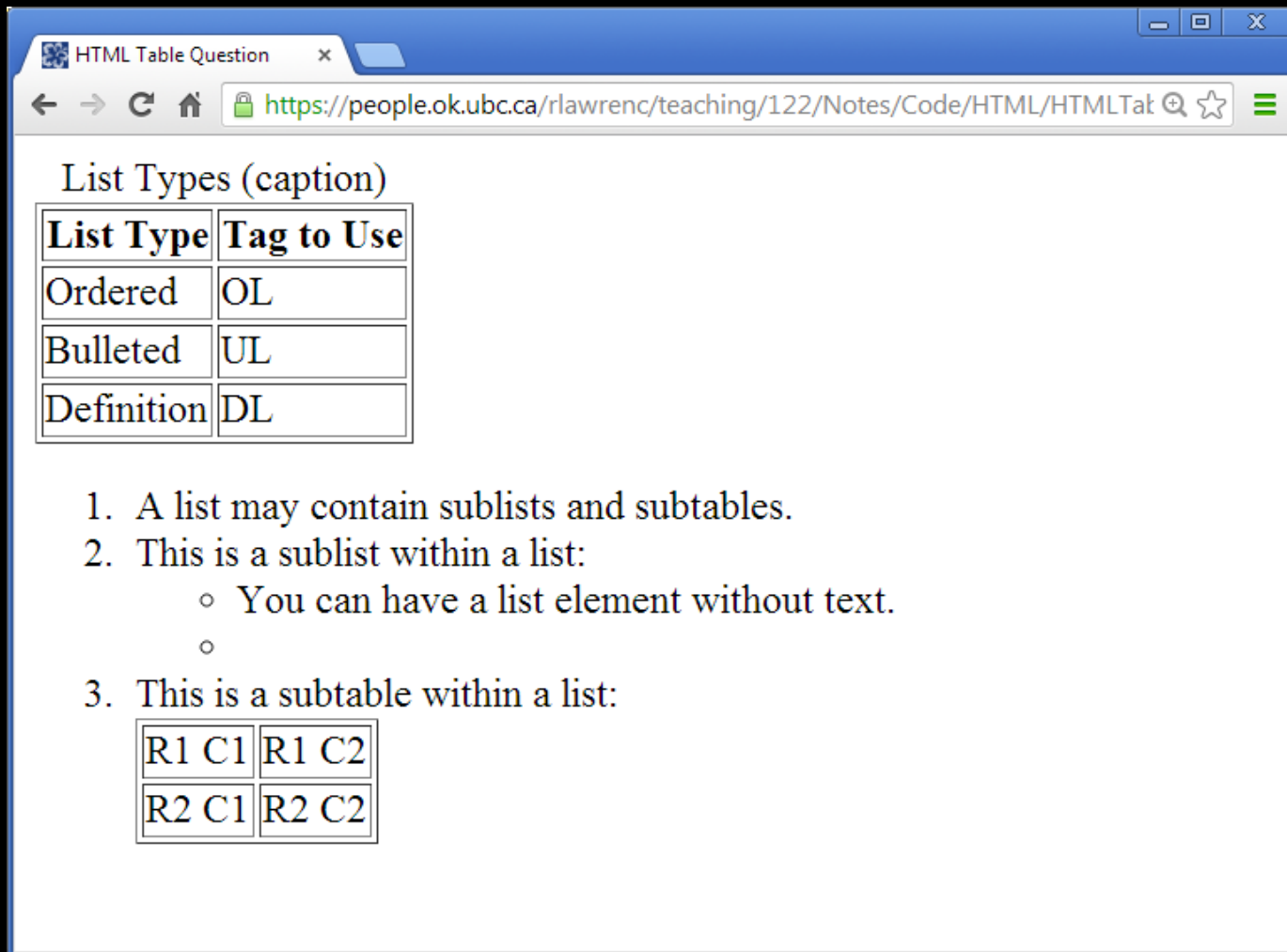
Question: Which one of these HTML code fragments will produce a table like this?

Heading 1	Heading 2
Val1	Val2

- a) `<table border="1">`
`<tr><th>Heading 1</th><th>Heading 2</th></tr>`
`<tr><td>Val1</td><td>Val2</td></tr>`
`</table>`
- b) `<table border="2">`
`<tr><td>Heading 1</td><td>Heading 2</td></tr>`
`<tr><dt>Val1</dt><dt>Val2</dt></tr>`
`</table>`

HTML Practice Question 3: Lists & Tables

Write the HTML document that looks like this:



The screenshot shows a web browser window with the title "HTML Table Question". The address bar contains the URL <https://people.ok.ubc.ca/rlawrenc/teaching/122/Notes/Code/HTML/HTMLTat>. The main content of the page is as follows:

List Types (caption)

List Type	Tag to Use
Ordered	OL
Bulleted	UL
Definition	DL

1. A list may contain sublists and subtables.
2. This is a sublist within a list:
 - o You can have a list element without text.
 - o
3. This is a subtable within a list:

R1 C1	R1 C2
R2 C1	R2 C2

Conclusion

Hypertext Markup Language (HTML) is a language for describing how a web page should appear in a web browser.

We have seen how we can use markup tags of HTML to:

- ◆ change fonts and formatting
- ◆ add images and hyperlinks
- ◆ create lists and tables

HTML is our first example of a language to instruct the computer on what to do.

Objectives

- ◆ Define: HTML, tag

Remember the HTML syntax for:

- ◆ Formatting tags: `<p>`, `<i>`, ``, `<h1>` to `<h6>`, `<pre>`
- ◆ Image and link tags: `<a>` and ``
- ◆ Special symbols using the escape character `&` (ampersand)
- ◆ Changing colors and text alignment
- ◆ Lists and tables



Be able to create HTML pages that have a given appearance.
Be able to draw what a HTML document will look like when displayed in a web browser.

Special Symbols

Since the `<` and `>` are special (reserved) symbols in the HTML language, we need a way to use them in our documents.

The `&` (ampersand) is the escape symbol that tells HTML a special character is required. Terminate with a `;` (semi-colon).

Common characters:

<code>&lt;</code>	<code><</code>
<code>&gt;</code>	<code>></code>
<code>&amp;</code>	<code>&</code>
<code>&copy;</code>	<code>©</code>
<code>&eacute;</code>	<code>é</code>
<code>&ntilde;</code>	<code>ñ</code>
<code>&nbsp;</code>	<code>non-breaking space</code>



COSC 122
Computer Fluency

Debugging

Dr. Firas Moosvi

Key Points

- 1) Debugging is the act of **finding** and **correcting errors** in a system.
- 2) All users need to know the general **debugging steps** due to the complexity of computer systems.
- 3) A common **reason for computer errors** is our **lack of precision** in specifying instructions to the computer.

Computers are Dumb... so We Must be Precise

Computers have no knowledge or intelligence **unless they are programmed with it.**

When talking with people, we **assume knowledge** and the **ability to reason** out errors or missing details when communicating.

Computers hate imprecision and cannot handle it by default.

- ◆ **Programmers often write applications to detect simple, common imprecise statements and fix them (but not always).**

Entering Data into Forms

Data is typically entered into a computer using a form.

A programmer can **restrict** the types and number of symbols that can go into a form field.

- ◆ e.g. only allow numbers in a phone number field

Many errors occur when users either enter data that does not follow these restrictions, or they enter incorrect data that is accepted by the computer **because it is not properly checked.**

Question: Have you ever entered false data into a form?

Debugging: What's the Problem?

Debugging is the process of determining why a **system** does not work properly.

We perform debugging all the time in daily life, usually to fix problems with other systems and tools we interact with (cars, lights, appliances, electronics, our own bodies, etc.).


Debugging is a little different with computers and information technology because **usually** it is not a **component failure** that is the source of the problem. More commonly, it is our **interaction and limited understanding** of how the computer works.

Example

HTML Debug Question

Let us try find some bugs in HTML. It is *easy* to find bugs; **if you know where to look**. However, sometimes it is just not obvious.

Here is a list of things I watch for:

1. Non-matching tags
2. Incorrect tag names
 - Bold (b) and italics (i).
 - Make sure you use a not ahref.
3. Missing or non-matched quotes
4. Be careful with [references](#).
5. Make sure images have correct path and name 

Tables are also **tricky**.

Common table errors

Error Type	Description
No closing <TD> or <TR> tag	Table appearance gets messed up
COLSPAN/ROWSPAN	Hard to track down
Nested tables:	Use colors to help solve nested table issues.

Desired Output

HTML Debug Question

Let us try find some bugs in HTML. It is *easy to find bugs*; **if you know where to look**. However, sometimes it is just not obvious.

Here is a list of things I watch for:

1. Non-matching tags
2. Incorrect tag names
3. Bold (b) and italics (i).
4. Make sure you use a not ahref.
5. Missing or non-matched quotes
6. Be careful with

Tables are also tricky.

tag

Common table errors

Error Type	Description
No closing <TD> or	
Table appearance gets messed up	
COLSPAN/ROWSPAN	Hard to track down
Nested tables	Use colors to solve nested table issues.

Actual Output

Debugging: Solving a Mystery

Debugging is very similar to solving a mystery.

To discover and solve the problem we ask questions like:

- ◆ Do I need more clues?
- ◆ Are my clues reliable?
- ◆ What is a theory to explain the problem?
- ◆ How can I test if my theory is correct?

Like solving mysteries, the only way to **get good at debugging** is **practice** and **gaining experience** about common problems and solutions.



The Four Key Steps in Debugging

1) Check that the error is reproducible.

- ◆ **Computers are deterministic. Make sure you *know exactly how to reproduce the error. Identify the error!***

2) Do not jump to conclusions.

- ◆ **Make sure you know what the problem is.**

3) Check all the "obvious" sources of error.

- ◆ **You would be surprised how often a cable is not plugged in...**

4) Isolate the problem

- ◆ **The goal is to make good assumptions and divisions of parts that you know are working and others that need investigation.**
- ◆ **Be careful! It is often parts (including yourself) that you assume are working that really are not.**

- **Make sure assumptions are backed up by tests.**

Debugging HTML Web pages

How to debug HTML web pages according to the 4 steps:

- ◆ 1) **Reproduce errors** - This is easy. Every time you reload or refresh the page, you should see the same errors.
- ◆ 2) **Do not jump to conclusions** - Although there are bugs in web browsers, it is vastly more likely that the **HTML document** contains errors. **Focus your attention there.**
- ◆ 3) **Obvious sources of errors** - One "obvious" source of errors is **non-matching open and closed tags**.
 - As you gain experience, more errors become obvious.
- ◆ 4) **Isolate the problem** - An HTML document is processed starting at the beginning, so try to fix errors at the start of the document first then work down.

Common HTML Errors

Some common HTML errors:

- ◆ Open with no matching close tag

- ` link <p> paragraph </p></html>`

- ◆ Non-matching quotes

- `` (open with ", close with ')
 - `` (HTML does not like smart quotes)

- ◆ Missing attribute or incorrect attribute name.

- ◆ Incorrect tag name (which may result in non-matching tags).

- ◆ Incorrect file name or hyperlink address.

- ◆ Forgot required tags like `<html>`, `<head>`, `<body>`.

- ◆ Forget to stop escape sequence with a semi-colon

- e.g. `<` (missing semi-colon should be `<`)

Aside: The Cost of Debugging

When developing a computer system or application, the process of **testing and debugging is extremely costly.**

Most software requires **40% of the total time, cost, and effort** to debug and fix problems in the system.

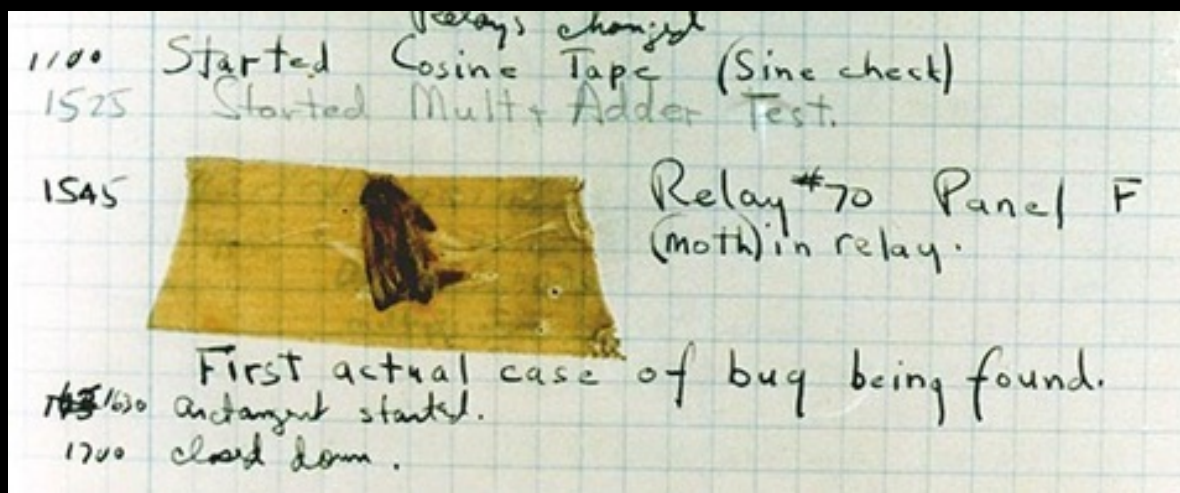
- ◆ **Even so, many errors go unnoticed until the system is used.**

To make software development more efficient and less costly, **software engineering** principles and techniques are followed.

- ◆ **Although building software is harder than building a bridge due to its complexity, software engineers continually strive to make software development better.**

Aside: The First Bug

The first "bug" in a computer system was actually a moth found in the *Harvard Mark II*, an **electromechanical** computer system, in 1947 by Rear Admiral **Grace Hopper**.



Source: U.S. Naval Historical Center Online Library Photograph NH 96566-KN



Debugging Follows the Scientific Method

Determining what a program does and **finding any errors (i.e., *testing*)** follows the scientific method.

- 1) **Model** – create a hypothesis on what the program does
- 2) **Predict** - for inputs not yet tried / simulated
- 3) **Experiment** – run the program to check your prediction
- 4) **Refine** – modify your hypothesis based on experimental results and repeat.

Understanding software is in many ways similar to understanding how complex real-world processes work.


HTML Debugging Question

Desired Output

HTML Debug Question

Let us try find some bugs in HTML. It is *easy* to find bugs; **if you know where to look**. However, sometimes it is just not obvious.

Here is a list of things I watch for:

1. Non-matching tags
2. Incorrect tag names
 - Bold (b) and italics (i).
 - Make sure you use a not ahref.
3. Missing or non-matched quotes
4. Be careful with [references](#).
5. Make sure images have correct path and name 

Tables are also **tricky**.

Common table errors

Error Type	Description
No closing <TD> or <TR> tag	Table appearance gets messed up
COLSPAN/ROWSPAN	Hard to track down
Nested tables:	Use colors to help solve nested table issues.

HTML Debugging Question

Actual Output

The screenshot shows a web browser window with the address bar containing the URL `https://people.ok.ubc.ca/rlawrenc/teaching/122/Notes/Code/HTML/HTMLDebugQuestionError.html`. The page title is "HTML Debug Question". The main content includes an introductory paragraph, a list of six items to watch for, a note about tables being tricky, a table of common table errors, and a final note about nested tables.

HTML Debug Question

Let us try find some bugs in HTML. It is *easy to find bugs; if you know where to look. However, sometimes it is just not obvious.*

Here is a list of things I watch for:

- 1. Non-matching tags*
- 2. Incorrect tag names*
- 3. Bold (b) and italics (i).*
- 4. Make sure you use a not ahref.*
- 5. Missing or non-matched quotes*
- 6. Be careful with*

Tables are also **tricky**.

tag

Common table errors

<i>Error Type</i>	<i>Description</i>
<i>No closing <TD> or</i>	
<i>Table appearance gets messed up</i>	
<i>COLSPAN/ROWSPAN</i>	<i>Hard to track down</i>

Nested tables Use colors to solve nested table issues.

HTML Debugging Question

HTML Document

```
<html>
<head><title>HTML Debug Question</title></head> <body>
<p>Let us try find some bugs in HTML. It is <i>easy to find </i>
bugs; <b>if you know where to look</b>. However, sometimes it is
just not obvious.</p>
```

```
<p>...</p>
Here is a list of things I watch for:
```

```
<ol>
<li>Non-matching tags</li>
<li>Incorrect tag names <li>
<ul>
<li>Bold (B) and italics (I).</li>
<li>Make sure you use A not AHREF.</li>
</ul>
<li>Missing or non-matched quotes</li>
<li>Be careful with <a href="HelloWorld.html">references</a>.
</li>
<li>Make sure images have correct path and name

</ol>
```

HTML Debugging Question

HTML Document (2)

```
<p>Tables are also <b style="font-size:150%;  
color:red">tricky</b>.</p>
```

```
<table border=2 style="background-color:yellow">  
  <caption>Common table errors</caption>  
  <tr><th>Error Type</th><td>Description</td></tr>  
  <tr><td>No closing &lt;TD&gt; or <TR> tag</td>  
    <td>Table appearance gets messed up</td></tr>  
  <tr><td>COLSPAN/ROWSPAN</td><td>Hard to track down</td></tr>  
  <tr><table style="background-color:orange">  
    <tr><td colspan=2><td>Nested tables</td>  
    <td>Use colors to help solve nested table issues.</td></tr>  
  </table></td></tr></table>  
</table>  
  
</body>  
</html>
```

Conclusion

Debugging is a systematic approach to discover and fix errors in a system.

- ◆ Debugging a computer system requires working with the computer to diagnose the problem with the realization that we are often the cause of the problem.

The four key steps of debugging are:

- ◆ 1) Check that the error is reproducible.
- ◆ 2) Make sure you know what the problem is.
- ◆ 3) Check all the "obvious" sources of error.
- ◆ 4) Isolate the problem

As users, we can resolve many errors with a little practice, experience, and patience without requiring help from IT service technicians.

Objectives

- ◆ Give some examples of imprecise communication.
- ◆ Explain why precision is important for a computer.
- ◆ Define: debugging
- ◆ List and explain the 4 key steps of debugging.
- ◆ List (and remember) some common HTML errors.



Be prepared to debug HTML documents both on the computer and on paper.