Four questions!

(1) How do you put a Giraffe into a refrigerator?

Four questions!

(1) How do you put a Giraffe into a refrigerator?

Answer:

- You open the refrigerator door
- You put the giraffe in
- You close the door

This question tests if you do simple things in an overly complicated way!

Four questions!

(2) How do you put an Elephant into the refrigerator?

Four questions!

(2) How do you put an Elephant into the refrigerator?

Answer:

- You open the door
- You remove the giraffe
- You put the elephant in
- You close the door

This question tests if you have the ability to think through the repercussions of previous decisions!

Four questions!

(3) The Lion King has called a meeting of all animals in the animal kingdom. Which animal cannot attend the meeting?

Four questions!

(3) The Lion King has called a meeting of all animals in the animal kingdom. Which animal cannot attend the meeting?

Answer:
The elephant!
Because you put him in the refrigerator!!

This question tests your memory!

Four questions!

(4) There is a river with crocodiles! There is no boat but you need to cross the river. How do you cross the river?

Four questions!

(4) There is a river with crocodiles! There is no boat but you need to cross the river. How do you cross the river?

Answer:
You swim!
All animals are at the meeting called by the Lion King; as are the crocodiles!!!

This question tests if you learn quickly from mistakes or not!!

Four questions!

PricewaterhouseCoopers used this test to test their senior managers.

They found that 90% of them could not get the four answers correct!

They took the test to a bunch of pre-schoolers, who got them all correct!!

Lesson:
We are all unique in the way we are. Don't lose yourself in "growing up" or trying to be how society wants you to be.

Don't lose what you have.
Stay true to who you are.

INFX 1606
INTRODUCTION TO
WEBSITE CREATION

DALHOUSIE
UNIVERSITY
FACULTY OF
COMPUTER SCIENCE

HTML Elements:
Text, Lists and Links

Raghav V. Sampangi

Instructor, Faculty of Computer Science, Dalhousie University

raghav@cs.dal.ca

Outline

HTML Elements

- Text
- Lists
- Links

HTML Elements (Recap): Top Hat

HTML Elements: Text

You have to consider two important distinctions in organizing and presenting text in HTML:

Structure. (i.e. structural markup)

This will help you organize text into headings and paragraphs.

Semantics. (i.e. semantic markup)

This will help you include extra information for the text, such as emphasizing a part of the sentence, including acronyms, etc.

Semantic elements ***define the nature of their content.***

HTML Text: Structural Markup

Headings.

There are six types or levels of headings:
<h1>, <h2>, <h3>, <h4>, <h5>, and <h6>

Each level of heading applies different font sizes to the heading text

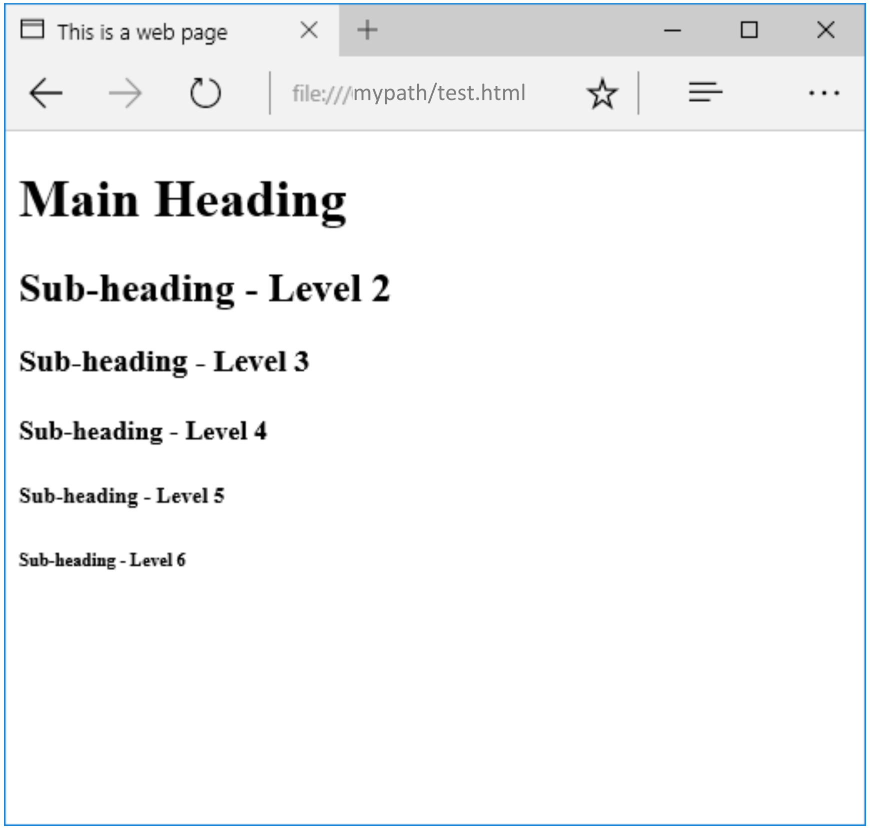
<h1> is the main heading;
<h2> is the first sub-heading, and all other levels are used for headings within the level 2 heading (<h2>)

HTML Text: Structural Markup

Headings.

Example:

```
<h1>Main Heading</h1>
<h2>Sub-heading - Level 2</h2>
<h3>Sub-heading - Level 3</h3>
<h4>Sub-heading - Level 4</h4>
<h5>Sub-heading - Level 5</h5>
<h6>Sub-heading - Level 6</h6>
```



Main Heading

Sub-heading - Level 2

Sub-heading - Level 3

Sub-heading - Level 4

Sub-heading - Level 5

Sub-heading - Level 6

Note. You can change the style / appearance of each type of heading using CSS.

HTML Text: Structural Markup

Paragraph.

You can create a paragraph in HTML by including the text between the opening and closing “p” tags:

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

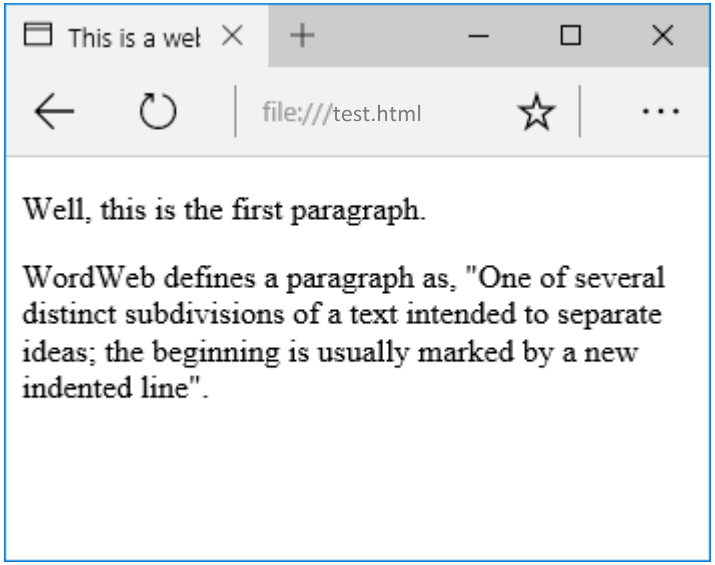
The browser adds a line space between paragraphs when it renders them

HTML Text: Structural Markup

Paragraph.

Example:

```
<p>Well, this is the first paragraph.</p>
<p>WordWeb defines a paragraph as, “One of
several distinct subdivisions of a text
intended to separate ideas; the beginning
is usually marked by a new indented line”.
</p>
```



Well, this is the first paragraph.

WordWeb defines a paragraph as, "One of several distinct subdivisions of a text intended to separate ideas; the beginning is usually marked by a new indented line".

HTML Text: Structural Markup

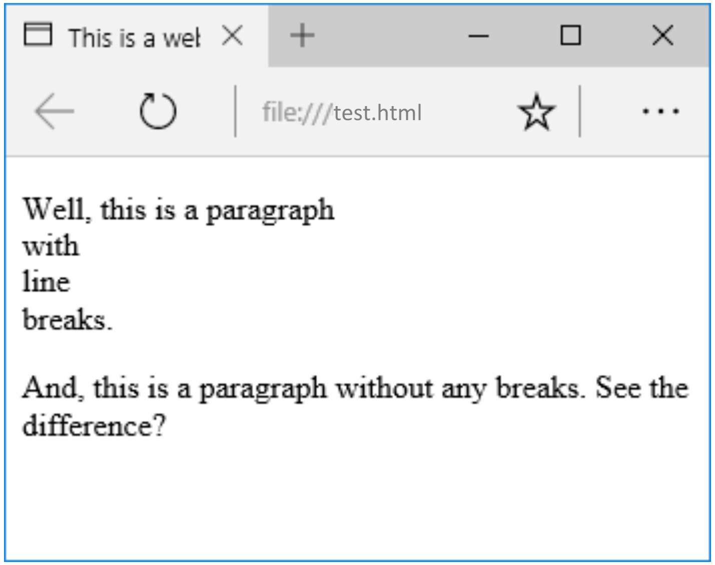
Paragraph: Line breaks.

Let’s suppose that you want to put some text in your paragraph in the next line – but you don’t want it in a new paragraph

For this, you would use **line breaks**, represented by:
<br /> or **<br/>** or **<br>**

Example:

```
<p>Well, this is a paragraph<br />
   with<br/>line<br>breaks.</p>
<p>And, this is a paragraph without
   any breaks. See the difference?</p>
```



Well, this is a paragraph
with
line
breaks.

And, this is a paragraph without any breaks. See the difference?

HTML Text: Structural Markup

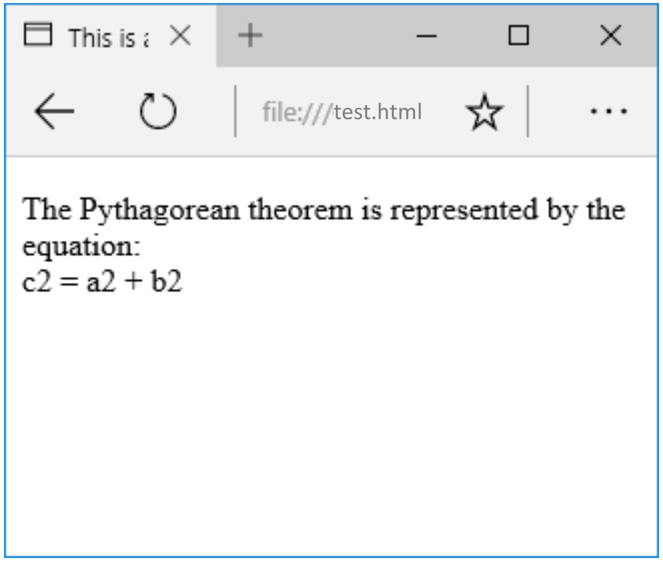
Paragraph: Special formatting.

You are required to display the Pythagorean theorem on your website. **What would you do to display the equation?**

Would this help?

```
<p>The Pythagorean theorem is represented by the equation:<br />
    c2 = a2 + b2</p>
```

Unfortunately, no!



The Pythagorean theorem is represented by the equation:
c2 = a2 + b2

HTML Text: Structural Markup

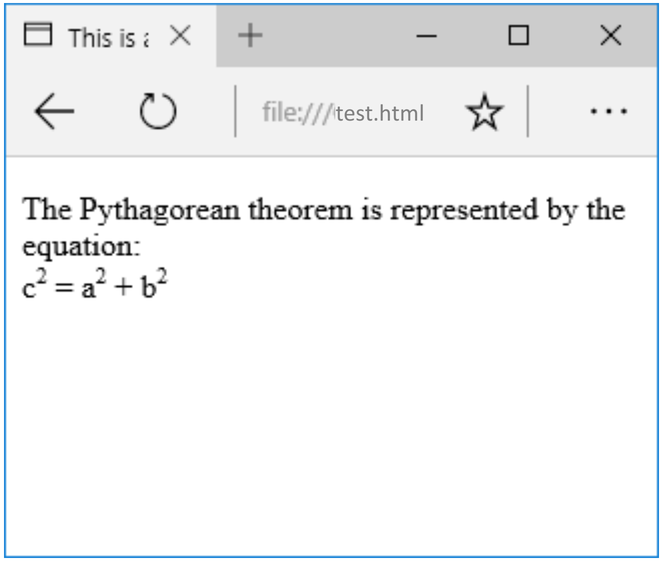
Paragraph: Special formatting.

This is a scenario where you would need to consider superscript tags, <sup></sup>

Superscript tag allows you to (positively) offset the vertical position of the text, i.e. display the text slightly above other text within the element

Here’s what you could use:

```
<p>The Pythagorean theorem is
   represented by the equation:<br />
   c<sup>2</sup> = a<sup>2</sup>
   + b<sup>2</sup>
</p>
```



The Pythagorean theorem is represented by the equation:
$c^2 = a^2 + b^2$

HTML Text: Structural Markup

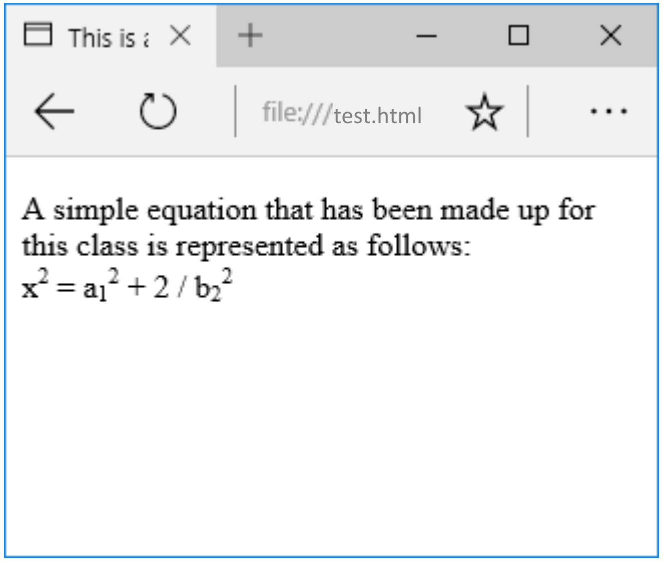
Paragraph: Special formatting.

Similarly, if you wished to use subscripts in your text, you would need to use subscript tags, <sub></sub>

Subscript tag allows you to (negatively) offset the vertical position of the text, i.e. display the text slightly below other text within the element

Here’s what you could use:

```
<p>A simple equation that has been
    made up for this class is
    represented as follows:<br />
    x<sup>2</sup> =
    a<sub>1</sub><sup>2</sup>
    + 2 / b<sub>2</sub><sup>2</sup>
</p>
```



A simple equation that has been made up for this class is represented as follows:
$x^2 = a_1^2 + 2 / b_2^2$

HTML Text: Structural Markup

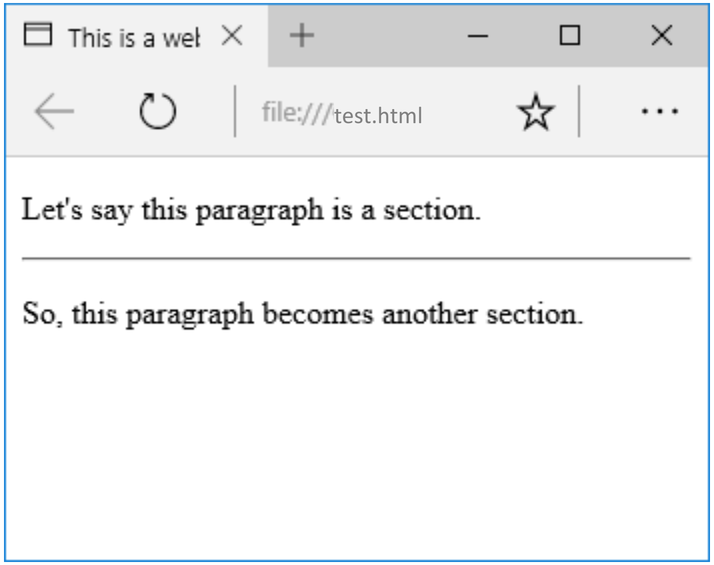
Horizontal rule.

We discussed horizontal rules, <hr> or <hr />, earlier

Horizontal rules help you in visually representing a change in sections, or let you include a "line of demarcation" between sections:

```
<p>Let's say this paragraph is a section.</p>

<hr/>
<p>So, this paragraph becomes another section.</p>
```



Let's say this paragraph is a section.

So, this paragraph becomes another section.

Semantic Markup

How would you define "semantics"?

The study of meaning.
Focus → the relationship between [words, phrases, etc.] and
what they denote or what they mean in a given context.

Why do we need to consider "semantics" in HTML?

- Previously, the web was being used ***only*** by human users;

- Now, there are automated web agents (e.g. search engines), and other user agents (e.g. screen readers) which also access web pages;

- While several tags were previously used to simply make content presentable to users, now, we must make sure our pages are meaningful to user agents or web agents as well.

- E.g. heading elements <h1>, <h2> … have semantic meaning as well; <h1> is considered to be more important than <h2>, and so on.

Semantic Markup

The purpose of semantic markup is to make description of your content more meaningful; not changing how your content appears.

Some examples of semantic markup:

<strong> → Used to indicate important textual content (typically, browsers represent such content in **bold**)

<em> → Used to emphasize content, or sometimes to indicate change in meaning of a part of sentence (typically, browsers represent such content *italicized*)

<blockquote> → Used to represent content “quoted” from another source (browsers indent blockquote text)

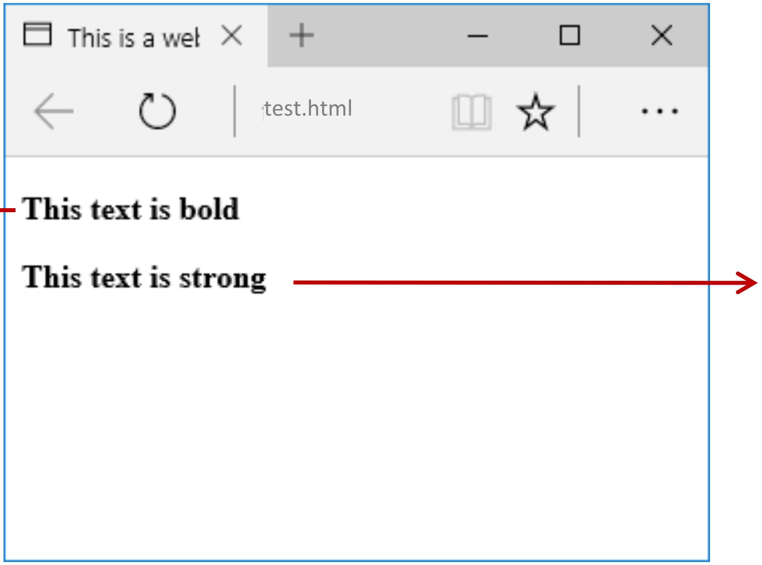
<cite> → Used to define the title of some piece of work (e.g. book title, pieces of art, movie name, etc.) (typically represented in *italic*)

And so on…

<b></b> versus <strong></strong>

What do you think is the difference between these two representations?

<p><b>This text is bold</b></p>



This text is bold

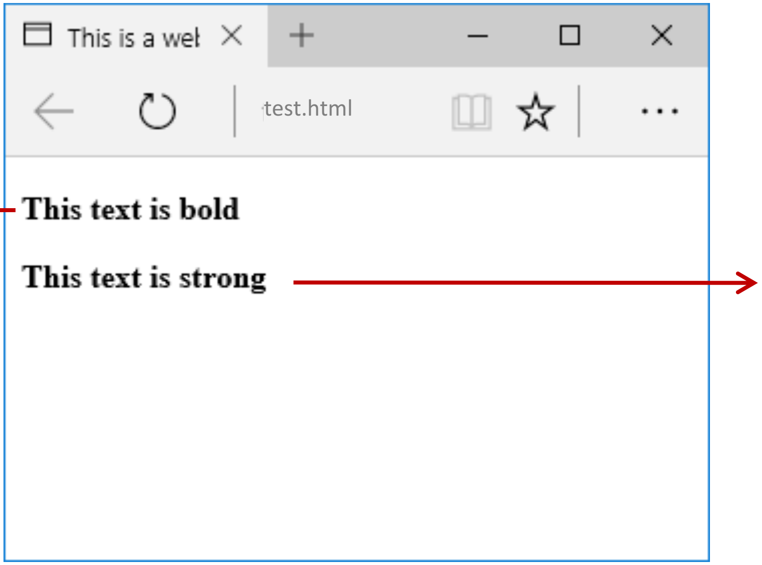
This text is strong

<p><strong>This text is strong</strong></p>

<b></b> versus <strong></strong>

What do you think is the difference between these two representations?

<p><b>This text is bold</b></p>



This text is bold

This text is strong

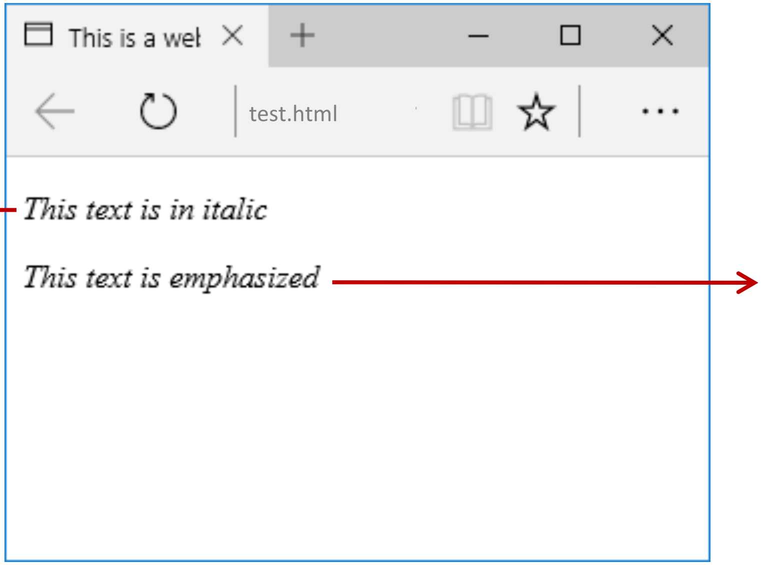
<p><strong>This text is strong</strong></p>

The **bold** element is only for presentation, where as the **strong** element indicates to the user/web agents that the content is important (and is typically represented as bold by browsers).

<i></i> versus <em></em>

Similarly, *italic* element is only for presentation, where as *emphasis* indicates that the content has specific meaning, e.g. changing context, thought, etc.

<p><i>This text is in italic</i></p>



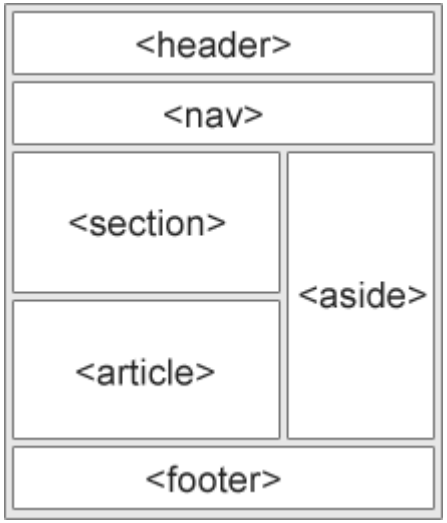
This text is in italic

This text is emphasized

<p><em>This text is emphasized</em></p>

Semantic Markup

Some examples of new semantic elements in HTML 5:



Source: w3schools.com

<header> → Used to indicate web page/section/article header

<nav> → Used to define navigation links

<section> → Used to indicate various sections in a page – sometimes, sections may have grouped content (***must have a heading***). E.g., a category of news items

<article> → Used to indicate independent / self-contained content. (***must have a heading***)
E.g., news article, blog post, etc.

<aside> → Used to indicate content other than main content
E.g. sidebar, containing announcements, quick links, etc.

<footer> → Used to indicate web page footer

And so on…

Semantic Markup

Using <article> and <section>

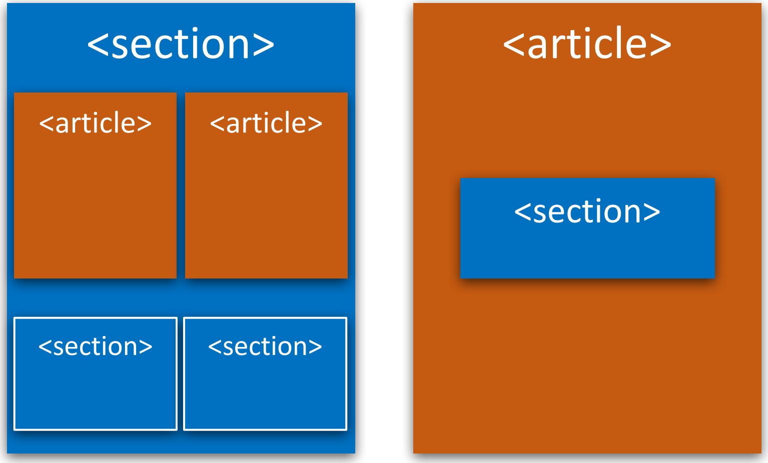
These elements are used depending on the content.

An <article> could be a standalone part on a website;
the same is true for a <section>

Furthermore, a <section> element can contain an <article>,
a <section> element can contain multiple <section> elements,
and an <article> can contain <section> elements.

Why is this important?

Because you are presenting content, and using these semantic elements to define the structure and nature of the content.



Semantic Markup

<header>

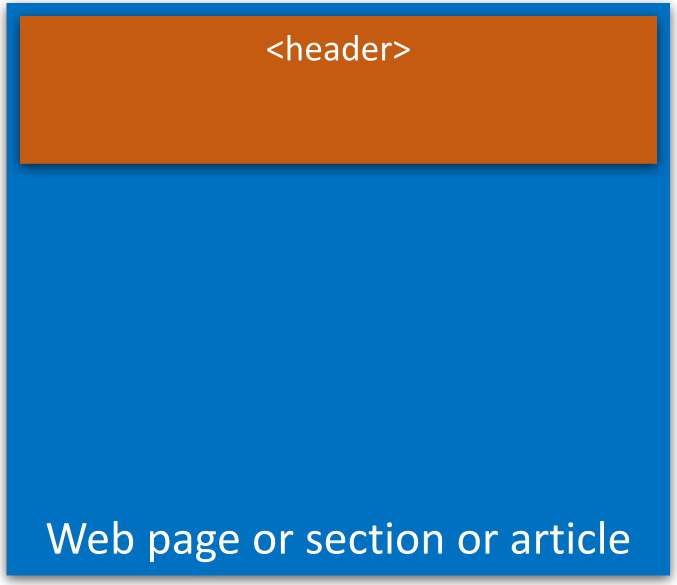
Indicates the header for a web page or a section or an article.

Useful if you are reusing styles or some type of content that needs to appear at the beginning of the element (e.g. introduction).

Where on the web page is this most commonly used?

To include document header content; Logos, navigation, etc.
These elements/styles are reused in every page, and hence, the page header styling can be set once and used many times.

Same with any article or section header styling.



What's the difference between <head> and <header>?

Semantic Markup

<footer>

Indicates the footer for a web page or a section or an article.

It should be used to include information about the element that “contains” the footer – i.e. the “parent” element.

Sidebar

Element nesting

HTML uses element nesting, or, including one element within another, and so on.

E.g.
<html> contains:
    <head> and <body> elements.
<head> contains:
    <title> and other elements.
<body> contains:
    <section>, <article>,
    and other elements.
<section> and <article> contain:
    <p>, <a>, etc.

and so on…

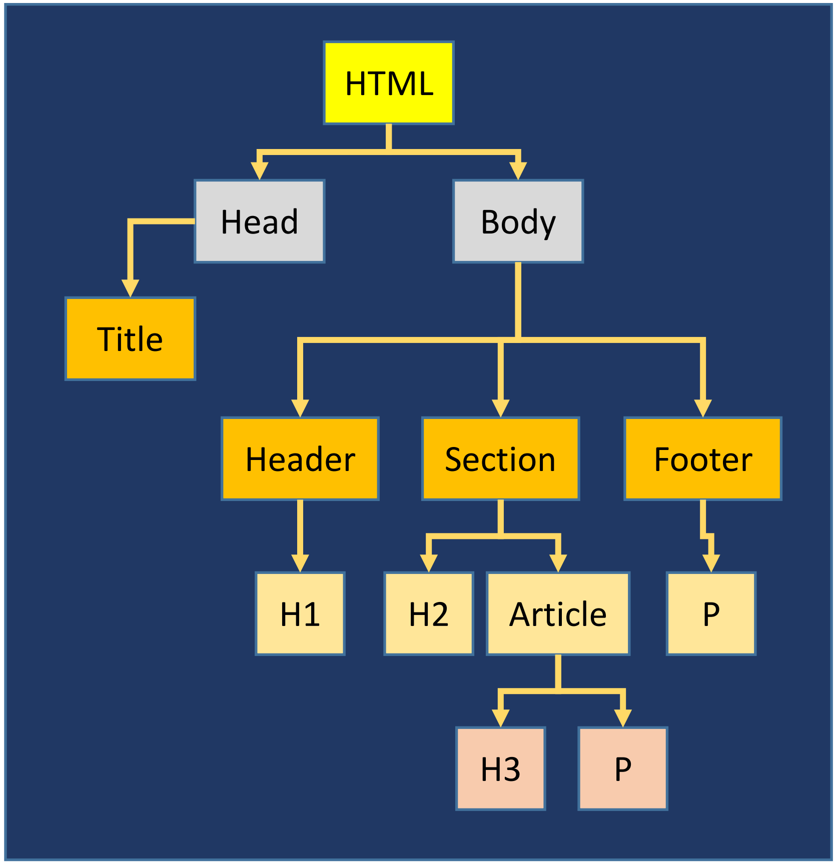
```
<html>
  <head>
    <title>A webpage!</title>
  </head>
  <body>
    <header>
      <h1>First web page!</h1>
    </header>
    <section>
      <h2>Here’s a Section</h2>
      <article>
        <h3>First Article</h3>
        <p>Here we go! This is going to be
          a fantastic year</p>
      </artcle>
    </section>
    <footer>
      Copyright 2016.
    </footer>
  </body>
</html>
```

Sidebar

Element nesting

HTML uses element nesting, or, including one element within another, and so on.

```
<html>
  <head>
    <title>A webpage!</title>
  </head>
  <body>
    <header>
      <h1>First web page!</h1>
    </header>
    <section>
      <h2>Here’s a Section</h2>
      <article>
        <h3>First Article</h3>
        <p>Here we go! This is going to be
          a fantastic year</p>
      </artcle>
    </section>
    <footer>
      <p>Copyright 2016.</p>
    </footer>
  </body>
</html>
```



Semantic Markup

<footer>

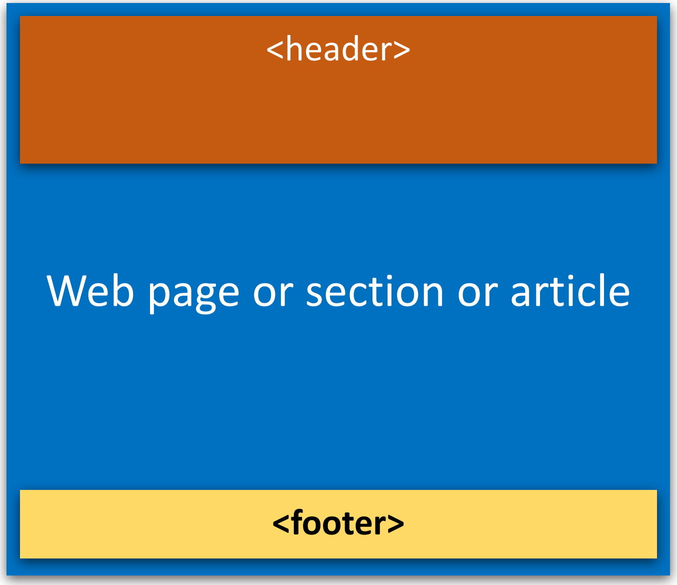
Indicates the footer for a web page or a section or an article.

It should be used to include information about the element that “contains” the footer – i.e. the “parent” element.

Where on the web page is this most commonly used?

To include document footer content; where you would specify some standard information such as copyright, contact etc.

If you want to include it in articles or sections, you could use it to specify author information, contact information, etc.



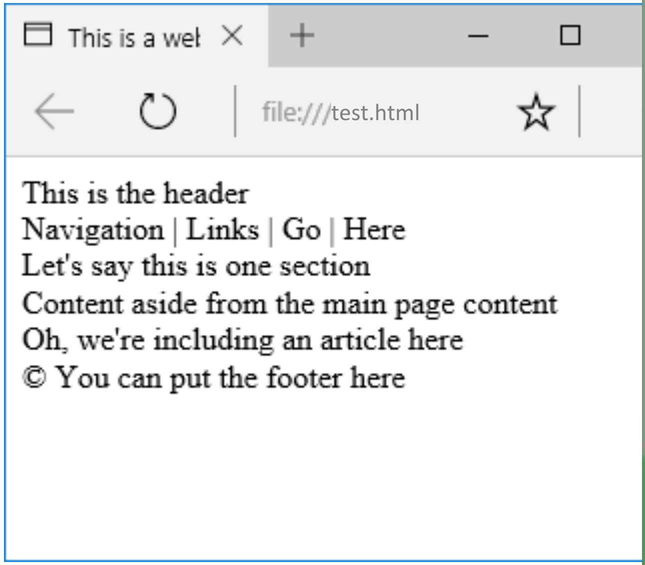
Semantic Markup: Example

```html
<header>This is the header</header>
<nav>Navigation | Links | Go | Here</nav>
<section>Let's say this is one section</section>
<aside>Content aside from the main page content</aside>
<article>Oh, we're including an article here</article>
<footer>&copy; You can put the footer here</footer>
```

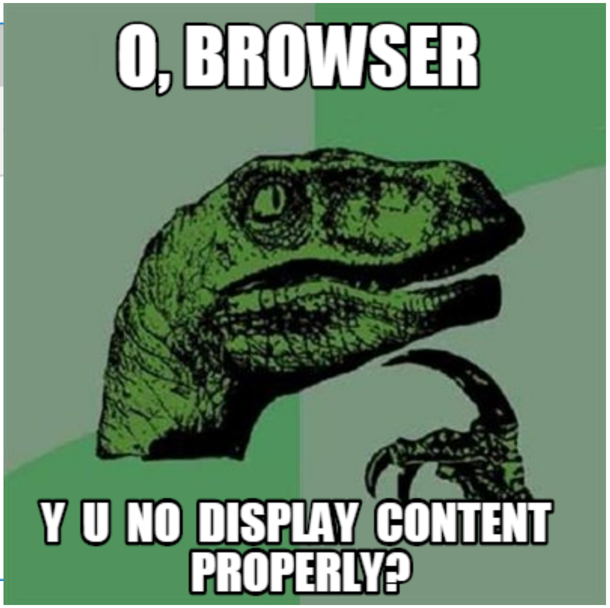
Semantic Markup: Example

```
<header>This is the header</header>
<nav>Navigation | Links | Go | Here</nav>
<section>Let's say this is one section</section>
<aside>Content aside from the main page content</aside>
<article>Oh, we're including an article here</article>
<footer>&copy; You can put the footer here</footer>
```

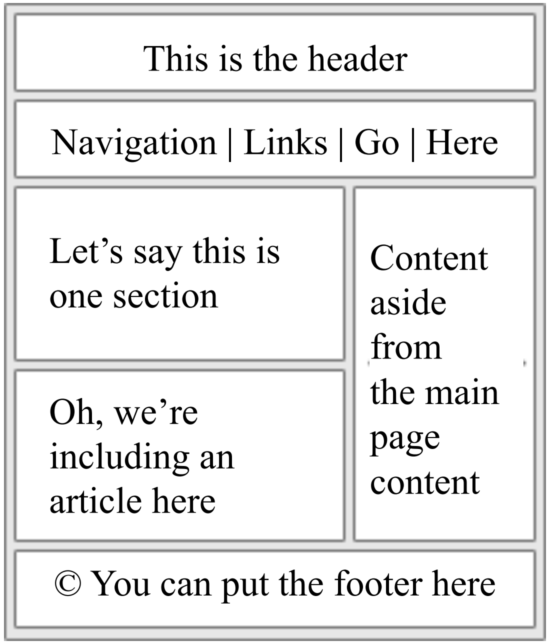
is rendered as



This is the header
Navigation | Links | Go | Here
Let's say this is one section
Content aside from the main page content
Oh, we're including an article here
© You can put the footer here



why not as…



This is the header

Navigation | Links | Go | Here

Let’s say this is one section

Content aside from the main page content

Oh, we’re including an article here

© You can put the footer here

Why is it not organized in the way it is expected to be?

Semantic Markup

Semantic markup is mainly used to tell browsers and web agents that the content has special meaning. These elements define the nature of the content.

By default, some of these markup may have specific styling

However, we have to use CSS to apply styles on the content to make it appear the way we want, i.e. to customize its presentation

So, structural markup only indicates content, semantic markup identifies the meaning that is depicted by the content, and styles help make the content “presentable”

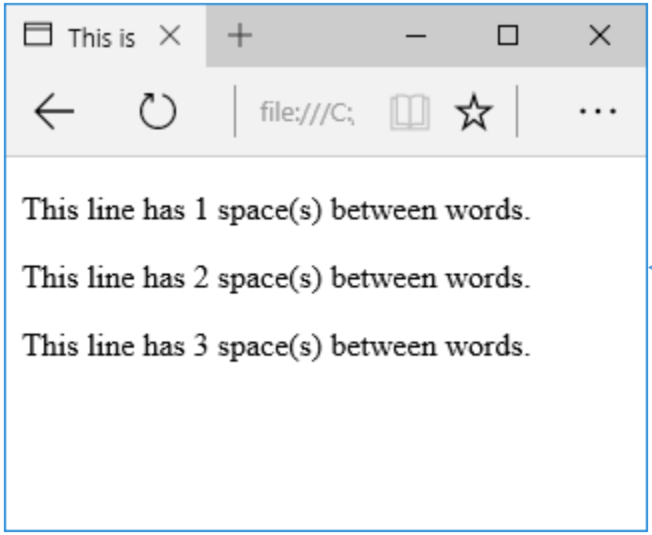
Top Hat

White “space” in markup

White space helps differentiate between various elements on a page

A browser “collapses” multiple white spaces when it displays the web page content → ***white space collapsing***

```
<p>This line has 1 space(s) between words.</p>
<p>This  line  has  2  space(s)  between  words.</p>
<p>This   line   has   3   space(s)   between   words.</p>
```



This line has 1 space(s) between words.

This line has 2 space(s) between words.

This line has 3 space(s) between words.

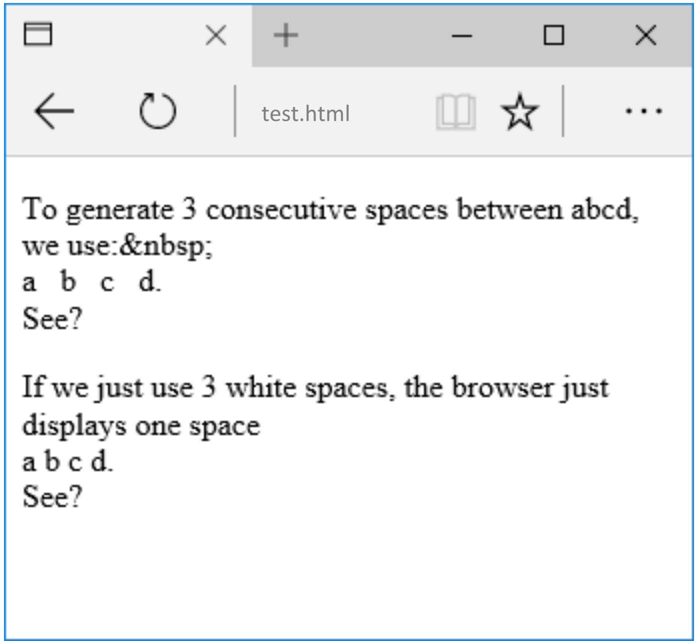
White “space” in markup

White space helps differentiate between various elements on a page

A browser “collapses” multiple white spaces when it displays the web page content → ***white space collapsing***

But, what if I want to use more than one space?

You can generate multiple spaces if you really want to.
You will have to use an ***HTML entity*** → ** **



To generate 3 consecutive spaces between abcd, we use:
a   b   c   d.
See?

If we just use 3 white spaces, the browser just displays one space
a b c d.
See?

HTML Entities

Tags include < and > symbols

What if you wanted to use less-than or greater-than symbols in your content?

You have to explicitly tell the browser that the symbols are not tags, but a part of the content

For this purpose, we use ***HTML entities***

Syntax:

&entity_name; OR **&#entity_number;**

HTML Entities

Some examples of HTML entities:

Entity name	Entity number	Description
&nbsp;	**&#160;**	Non-breaking space ( )
&lt;	**&#60;**	Less-than (**<**)
&gt;	**&#62;**	Greater-than (**>**)
&amp;	**&#38;**	Ampersand (**&**)
&copy;	**&#169;**	Copyright (©)
&reg;	**&#174;**	Registered trademark (®)

HTML Lists

HTML Elements: Lists

List. An ordered way to display content or a set of items in content

Three main types of lists in HTML:

Unordered lists. Use symbols (e.g. bullet points) to indicate order.

Ordered lists. Use characters or numbers to indicate order.

Description or definition lists. Used to indicate descriptions or definitions of terms.

You don’t need to put these lists inside paragraphs, i.e. <p></p>
They can exist by themselves, as independent content elements.

HTML Elements: Lists

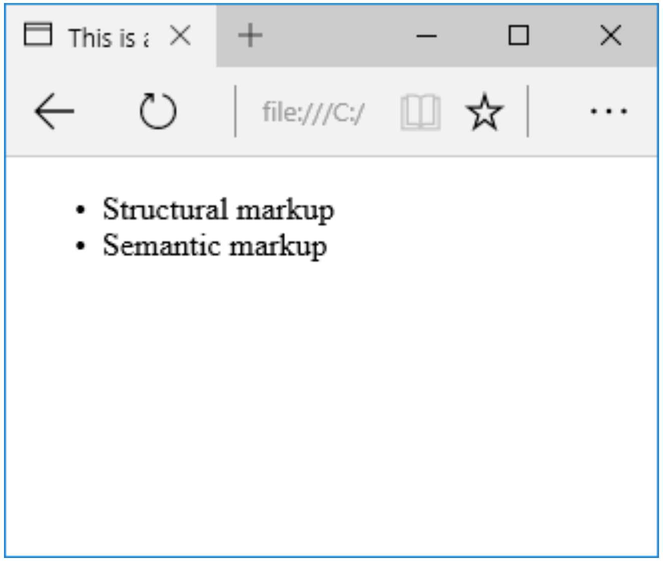
Unordered lists. Use symbols (e.g. bullet points) to indicate order.

Unordered lists are identified by **<ul></ul>**

List items are identified by **<li></li>**

Example:

```
<ul>
    <li>Structural markup</li>
    <li>Semantic markup</li>
</ul>
```



- Structural markup
- Semantic markup

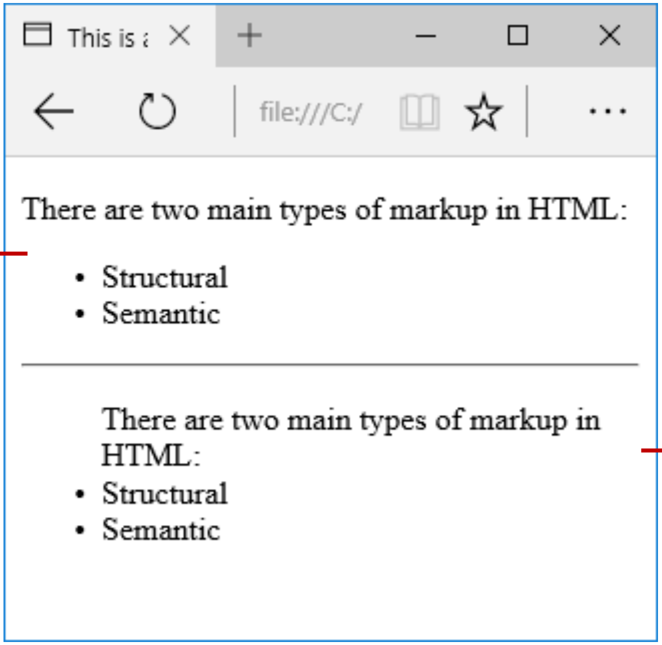
HTML Elements: Lists

Unordered lists. Use symbols (e.g. bullet points) to indicate order.

If you wish to have an association between some text and the items in the list, you must include a ***paragraph element before the list***, rather than including the text when the list begins.

The browser will render the markup in the second case, but it is wrong!!

```
<p>There are two main
types of markup in
HTML:</p>
<ul>
<li>Structural</li>
<li>Semantic</li>
</ul>
```



There are two main types of markup in HTML:

- Structural
- Semantic

There are two main types of markup in HTML:

- Structural
- Semantic

Text not allowed between <ul> and <li>

```
<ul>There are two main
types of markup in
HTML:
<li>Structural</li>
<li>Semantic</li>
</ul>
```

HTML Elements: Lists

Unordered lists. Use symbols (e.g. bullet points) to indicate order.

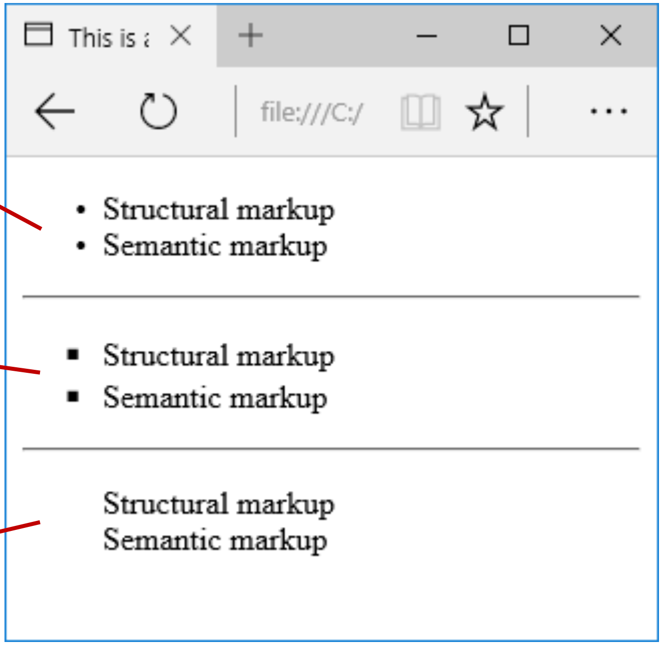
There is a default set of symbols used to render unordered lists. However, if you wish to change the symbol, you have to use the “style” attribute of the lists

Example:

```
<ul>
      <li>Structural markup</li>
      <li>Semantic markup</li>
</ul>
```

```
<ul style="list-style-type:square">
      <li>Structural markup</li>
      <li>Semantic markup</li>
</ul>
```

```
<ul style="list-style-type:none">
      <li>Structural markup</li>
      <li>Semantic markup</li>
</ul>
```



```
  • Structural markup
  • Semantic markup

  ▪ Structural markup
  ▪ Semantic markup

    Structural markup
    Semantic markup
```

HTML Elements: Lists

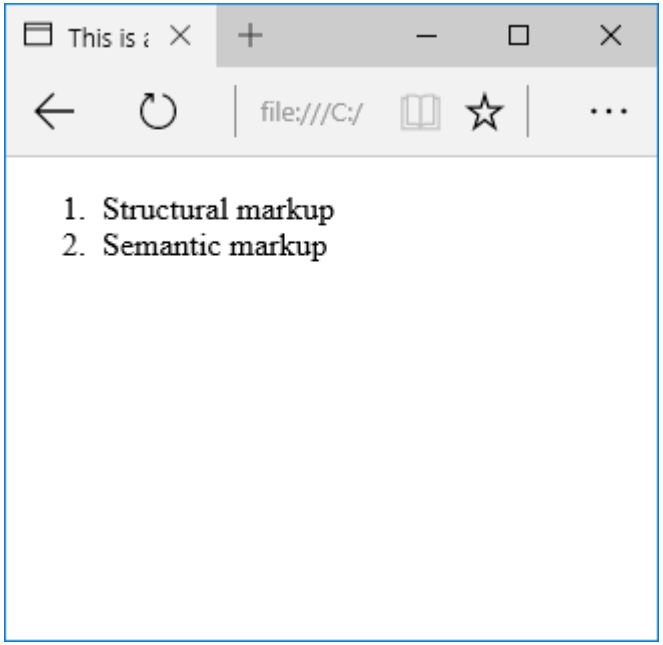
Ordered lists. Use characters or numbers to indicate order.

Ordered lists are identified by **<ol></ol>**

List items are, once again, identified by **<li></li>**

Example:

```
<ol>
    <li>Structural markup</li>
    <li>Semantic markup</li>
</ol>
```



1. Structural markup
2. Semantic markup

HTML Elements: Lists

Ordered lists. Use characters or numbers to indicate order.

There is a default set of characters used to render ordered lists. However, if you wish to change the characters, you have to use the "type" attribute of ordered lists

Type	Description
"1"	Numbers (default)
"A"	Uppercase letters
"a"	Lowercase letters
"I"	Uppercase Roman numbers
"i"	Lowercase Roman numbers

HTML Elements: Lists

Ordered lists: Examples of using the “type” attribute:

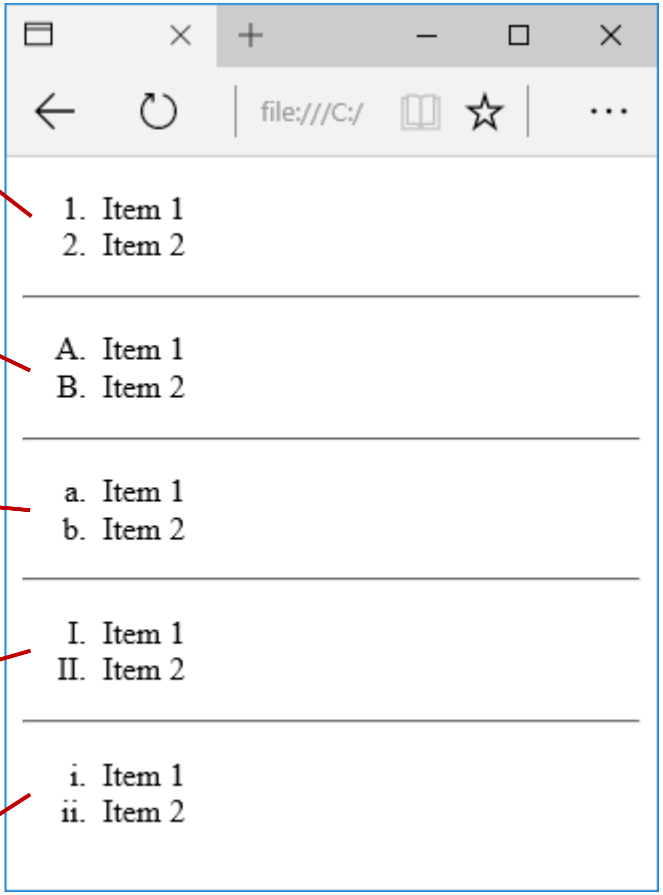
```
<ol type="1">
      <li>Item 1</li>
      <li>Item 2</li>
</ol>
```

```
<ol type="A">
      <li>Item 1</li>
      <li>Item 2</li>
</ol>
```

```
<ol type="a">
      <li>Item 1</li>
      <li>Item 2</li>
</ol>
```

```
<ol type="I">
      <li>Item 1</li>
      <li>Item 2</li>
</ol>
```

```
<ol type="i">
      <li>Item 1</li>
      <li>Item 2</li>
</ol>
```



```
 1. Item 1
 2. Item 2

 A. Item 1
 B. Item 2

 a. Item 1
 b. Item 2

  I. Item 1
 II. Item 2

  i. Item 1
 ii. Item 2
```

HTML Elements: Lists

Description or definition lists. Used to indicate descriptions or definitions of terms.

Let’s say that you are defining some key words / terms used in an article. Description / definition lists are very useful in such contexts.

Example:

<p>

Defines a paragraph

<ol>

Defines an ordered list

<ul>

Defines an unordered list

HTML Elements: Lists

Description or definition lists. Used to indicate descriptions or definitions of terms.

Elements of a description list:

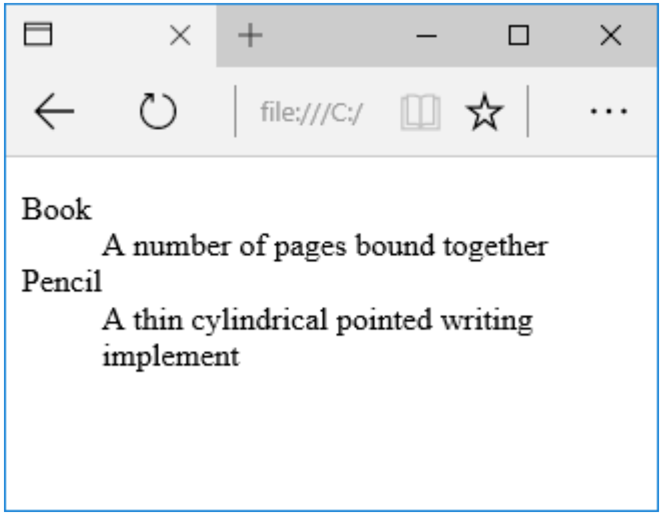
<dl></dl> defines a description list element

<dt></dt> defines a description term (or the name)

<dd></dd> defines the description of the term

Example:

```
<dl>
    <dt>Book</dt>
    <dd>A number of pages bound together</dd>
    <dt>Pencil</dt>
    <dd>A thin cylindrical pointed writing implement</dd>
</dl>
```



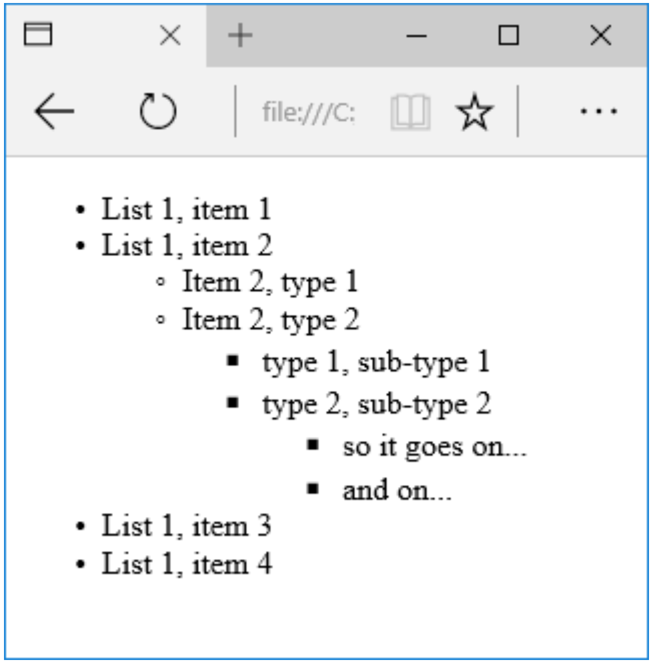
Book
    A number of pages bound together
Pencil
    A thin cylindrical pointed writing implement

HTML Elements: Lists

Nested lists. You may also include lists within lists.

Example:

```
<ul>
   <li>List 1, item 1</li>
   <li>List 1, item 2
      <ul>
         <li>Item 2, type 1</li>
         <li>Item 2, type 2
            <ul>
               <li>type 1, sub-type 1</li>
               <li>type 2, sub-type 2
                  <ul>
                     <li>so it goes on...</li>
                     <li>and on...</li>
                  </ul>
               </li>
            </ul>
         </li>
      </ul>
   </li>
   <li>List 1, item 3</li>
   <li>List 1, item 4 </li>
</ul>
```



- List 1, item 1
- List 1, item 2
    - Item 2, type 1
    - Item 2, type 2
        - type 1, sub-type 1
        - type 2, sub-type 2
            - so it goes on...
            - and on...
- List 1, item 3
- List 1, item 4

HTML Elements: Lists

Nested lists. You may also include lists within lists.

This is wrong, but will be rendered by the browser.

Why is this wrong?

```
<ul>
  <li>List 1, item 1</li>
  <li>List 1, item 2</li>
  <ul>
    <li>Item 2, type 1</li>
    <li>Item 2, type 2</li>
  </ul>
  <li>List 1, item 3</li>
  <li>List 1, item 4 </li>
</ul>
```

HTML Elements: Lists

Nested lists. You may also include lists within lists.

This is wrong, but will be rendered by the browser.

Why is this wrong?

```
<ul>
  <li>List 1, item 1</li>
  <li>List 1, item 2</li>  ← This is wrong!
  <ul>
    <li>Item 2, type 1</li>
    <li>Item 2, type 2</li>
  </ul>
  <li>List 1, item 3</li>
  <li>List 1, item 4 </li>
</ul>
```

This is the correct way to create nested lists, and will be validated by W3C Validator.

```
   <ul>
     <li>List 1, item 1</li>
     <li>List 1, item 2
       <ul>
         <li>Item 2, type 1</li>
         <li>Item 2, type 2</li>
       </ul>
→    </li>
     <li>List 1, item 3</li>
     <li>List 1, item 4 </li>
   </ul>
```

<ul> elements are **not valid child elements of another <ul>**;
they are valid child elements of **<li>**

HTML Elements: Lists

Summary.

Browsers indent list items by default

Three types of lists – unordered, ordered and description/definition

If you wish to change the appearance of [the bullet or type of numbers used in] the lists, you can use CSS to set such properties.