



Applied Sciences Faculty

Management Information Systems

Dr. Yakup BAKIŞ

Marking Up Text

Learning Objectives



- Understand semantic text structure
- Use paragraphs and headings correctly
- Create ordered and unordered lists
- Apply div and span for grouping

What is HTML?



- **HTML = HyperText Markup Language**
 - The basic building block of every web page
 - Defines the **structure** of content on the web
 - Uses **tags** to describe elements (title, paragraph, image, link)
 - Works together with **CSS** (design) and **JavaScript** (interaction)
-
- **HTML is not a programming language — it is a structure language.**

Main Structure of an HTML Document



- `<html>`
 - `<head></head>`
 - `<body></body>`
 - `</html>`
-
- https://www.w3schools.com/html/tryit.asp?filename=tryhtml_default_default

Basic HTML Document






Element	Description
<code><!DOCTYPE html></code>	Declares the document type and HTML version (HTML5)
<code><html></code>	The root element of an HTML page
<code><head></code>	Contains metadata and resources for the page.
<code><title></code>	Specifies a title for the document
<code><body></code>	Contains the visible page content
<code><h1></code>	Defines a large heading
<code><p></code>	Used for blocks of text



HTML documents are defined by HTML elements. An HTML element is everything between the start tag and the end tag. The start tag is often called the opening tag. The end tag is often called the closing tag.

In HTML, tags usually come in pairs:

- **< >** is the opening tag.
 - It tells tag starts here.”
- **</ >** is the closing tag.
 - The slash (/) at the beginning means:
 - “This tag closes or ends the element.
- ”So the / symbol shows that:
-  the content is finished
-  the element is closing
-  the browser should stop applying that tag



--> Comments



- Comments

< ! - - insert single or multi-line comment here
information about the page - - >

Brace yourself for a shocker...

- Don't over-comment because of the download hit – comment only esoteric stuff.

Do use blank lines liberally to separate chunks of code, and use indentation to show nesting (but careful with nesting and server-side includes).

The <!DOCTYPE> Declaration



- ▪ The <!DOCTYPE> declaration represents the document type, and helps browsers to display web pages correctly.
- ▪ It must only appear once, at the top of the page before any HTML tags.
- ▪ The <!DOCTYPE> declaration is not case sensitive.
- ▪ The <!DOCTYPE> declaration for HTML5 is:

• <!DOCTYPE html>

HTML Versions



Version	Year
HTML 1.0	1993
HTML 2.0	1995
HTML3.2	1997
HTML 4.01	1999
XHTML 1.0	2000
HTML5	2014

<html> ... </html>



- `<html> ... </html> <!-- covers the whole page -->`
- the start of the entire document
- Indicates that it is an HTML document
- ↪ It is the basic root tag -->

- **<html> is the opening tag.**

- It tells the browser: "HTML starts here."

- **</html> is the closing tag.**

- The slash (/) at the beginning means:
- "This tag closes or ends the element."

- **Example**

```
<!DOCTYPE html>
```

- `<!-- This line tells the browser that this document uses HTML5.`

- `It is considered essential in modern web pages.-->`

- `<html lang="en">`

- `<!-- The <html> tag is the root element of the entire document. The "lang" attribute helps search engines and screen readers.-->`

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title>Penny's Page</title>
```

```
<meta name = "keywords" content = "penny" />
```

```
<meta name = "description" content = " this site is a page about penny" />
```

```
</head>
```

```
<body>
```

```
<h1>Header Stuff</h1>
```

```
<p>This is a paragraph of content of some sort, that goes on, and on,  
and on, and on, and on, and on....
```

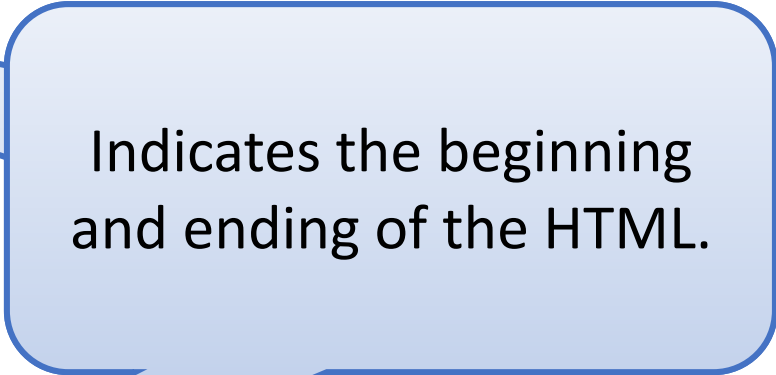
```
</p>
```

```
<p>This is another paragraph of some kind of content, that goes on,  
and on, and on, and on
```

```
</p>
```

```
</body>
```

```
</html>
```

A light blue callout box with a dark blue border and a pointer pointing to the opening and closing HTML tags. It contains the text: "Indicates the beginning and ending of the HTML."

Indicates the beginning and ending of the HTML.

<head> </head>



- **<head> </head>**
- **<!-- 2. <head>** (logically necessary) Title,
- meta information, CSS are here.
- The user does not see it, but it is important for the system -->

- **Example**

`<head>`

- `<!-- The <head> section contains information ABOUT the page, not the visible content of the page. -->`
- `<meta charset="UTF-8">`
- `<!-- UTF-8 ensures correct display of all characters, including special and international letters. -->`
- `<meta name="viewport" content="width=device-width, initial-scale=1.0">`
- `<!-- This makes the page responsive on mobile devices. -->`
- `<title>My First Web Page</title>`
- `<!-- The <title> appears in the browser tab. A page without a title is not professional. -->`
- `</head>`

```
<!DOCTYPE html>
<html>
<head>
  <title>Penny's Page</title>
  <meta name = "keyword" ...>
  <meta name = "description" ...>
</head>

<body>
<h1>Header Stuff</h1>
<p>This is a paragraph of content of some sort, that
  and on, and on, and on, and on....
</p>
<p>This is another paragraph of some kind of content
  and on, and on, and on...
</p>
</body>
</html>
```

Contains information about the document as a whole.

Generally not rendered on the page itself, merely informational.

Only one <head> per page.

```
<!DOCTYPE html>
<html>
<head>
  <title>Penny's Page</title>
  <meta name="keywords" content="phrase1, phrase2, ... ">
  <meta name="description" content="this site does ... ">
</head>
```

Displays on the colored bar across the top line of the browser window.

Displays when search engines hit on the site.

Is used by search engines to catalog the site.

Is the default name when a user adds the page to his or her favorites list.

Remember that users might link to this page directly, bypassing the homepage, so it really needs to be clear.

```
<!DOCTYPE html>
<html>
<head>
  <title>Penny's Page</title>
  <meta name = "keywords" content = "phrase1, p2, ... ">
  <meta name = "description" content = "this site does ... ">
</head>

<body>
<h1>Header Stuff
<p>This is a paragraph of text, that goes on, and on.
```

Meta tags are used primarily by search engines.

name = "keywords" identifies the prime search words for search engines that use keywords (not all do).

name = "description" identifies the description that displays sometimes in search engine listings.

Only one `<body>` per page.

Contains the elements that display on the page.

Can include deprecated attributes for default text and link colors, and background color and images.

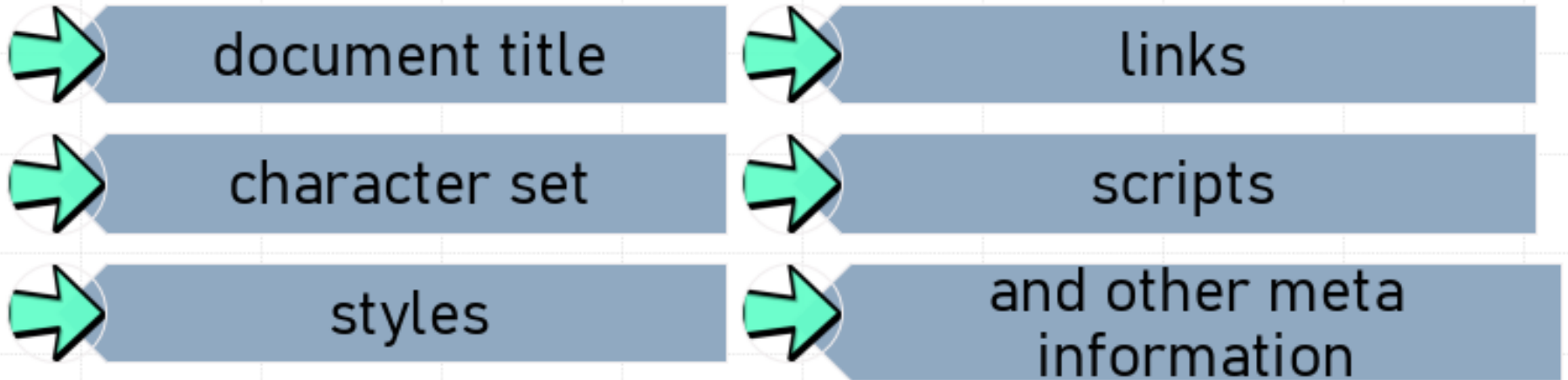
```
<!DOCTYPE html>
<html>
  <head>
    <title>
    <meta
    <meta name = "descri
  </head>

  <body >
    <h1>Header Stuff</h1>
    <p>This is a paragraph of content of some sort, that goes on, and on,
      and on, and on, and on, and on....
    </p>
    <p>This is another paragraph of some kind of content, that goes on,
      and on, and on, and on...
    </p>
  </body>
</html>
```

Metadata typically defines...



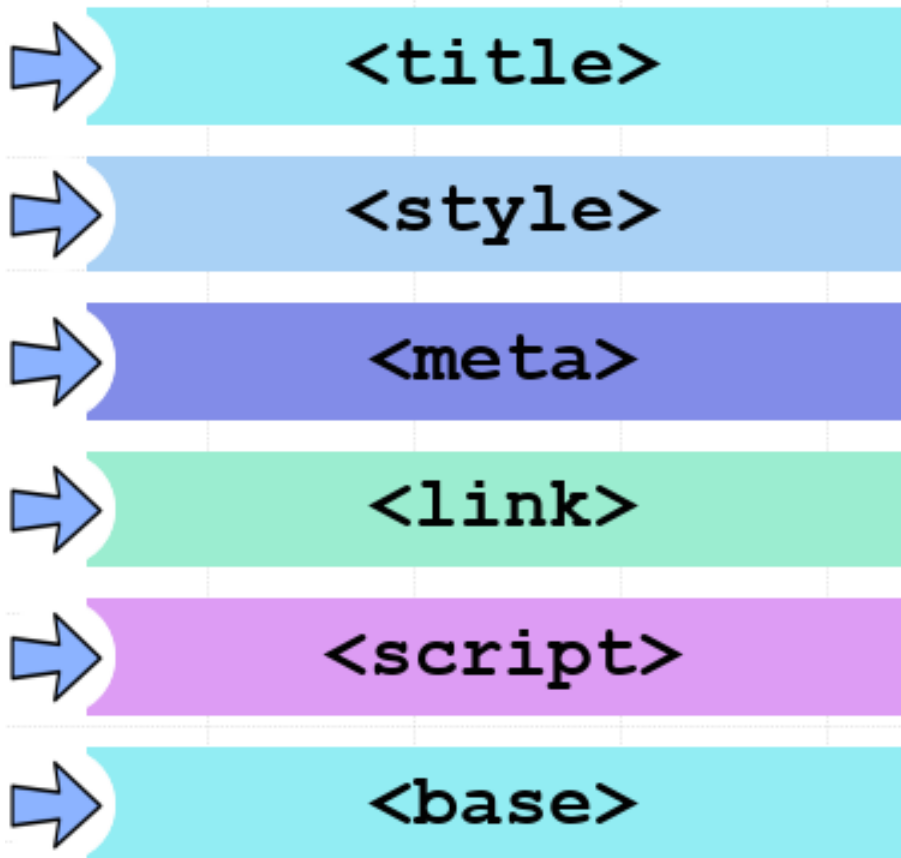
- define:



metadata



- The following tags describe metadata:



<title> Element



- ▪ HTML title element used to sets the title of the webpage, shown in the browser tab.
- ▪ Provides a title for the page when it is added to favourites.
- ▪ Displays a title for the page in search engine results.

```
<head>  
<title>Page Title</title>  
</head>
```

<style> Element



- The <style> element is used to define internal style information for a single HTML page:

```
<style>
  body {
    background-color: lightgray;
  }
  h1 {
    color: green;
  }
</style>
```

<link> Element



- The <link> element is used to link to external style sheets for examples CSS and JavaScript files.
- Example:

```
<link rel="stylesheet" href="mystyle.css">  
<script src="script.js" defer></script>
```

<meta> Element



- ▪ Charset Meta Tag: Define the character set used:

```
<meta charset="UTF-8">
```

- ▪ Description Meta Tag: Define a description of your web page:

```
<meta name="description"  
content="Free Web tutorials">
```

- Keywords Meta Tag: Define keywords for search engines:

```
<meta name="keywords" content="HTML, CSS, XML, JavaScript">
```

- ▪ Author Meta Tag: Define the author of a page:

```
<meta name="author" content="John Doe">
```

- ▪ Refresh Meta Tag: Refresh document every 30 seconds:

```
<meta http-equiv="refresh" content="30">
```

Example of <meta> tags:



- `<meta charset="UTF-8">`
- `<meta name="description" content="Free`
- `Web tutorials">`
- `<meta name="keywords" content="HTML,CSS,X`
- `ML,JavaScript">`
- `<meta name="author" content="John Doe">`

Viewport Meta Tag: Setting The Viewport



- HTML5 introduced a method to let web designers take control over the viewport, through the <meta> tag.
- • The viewport is the user's visible area of a web page.
- • It varies with the device, and will be smaller on a mobile phone than on a computer screen.
- • To include viewport element in all web pages:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Viewport Meta Tag: Setting The Viewport



- A `<meta>` viewport element gives the browser instructions on how to control the page's dimensions and scaling.
- ▪ **width=device-width**
- ▪ Sets the width of the page to follow the screen-width of the device (which will vary depending on the device).
- ▪ **initial-scale=1.0**
- ▪ Sets the initial zoom level when the page is first loaded by the browser.

<body> </body>



- **<body> </body>**
- <!-- content appearing on the screen -->
- <!-- 3. <body> Everything that appears on the page happens here
- The <body> element contains all the contents of an HTML document, such as text, hyperlinks, images, tables, lists, etc.
- ☞ There is no page without it -->
- **Example**
<body>
 - <!-- The <body> contains EVERYTHING visible on the screen.
 - Without the body, nothing can be displayed. -->

Headings

`<h1>`

`</h1>`

to `<h6>`

`</h6>`



- `<html>`
- `<body>`

- `<h1>`This is Heading 1`</h1>`
- `<h2>`Heading 2 is Smaller`</h2>`
- `<h3>`Heading 3 is Smaller Still`</h3>`

- `</body>`
- `</html>`

- **HTML headings are defined with the `<h1>` to `<h6>` tags The lower the number, the larger the heading size**

- `<h1>` is most important
- Used for document outline
- Essential for accessibility
- Do not skip levels
- Use logical nesting
- One main `<h1>` per page
- Supports screen readers

HTML paragraphs

`<p>` `</p>`



- HTML paragraphs are defined with the `<p>` tag. Most browsers automatically put a line break and space after a `</p>` tag.

- `<html>`
- `<body>`

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

- `</body>`
- `</html>`

- Defines a block of text
- Automatically adds spacing
- Used for readable content blocks
- Should not contain headings or lists
- Browsers render spacing automatically
- Multiple paragraphs create structure

- All of the text placed within the paragraph tags appears as a single paragraph. The text wraps to fill the available line length within the browser.
- There is a blank line both before and after the paragraph display.
- `</p>` is required by newer standards even though browsers will accept paragraphs without ending tags.

- `<body>`

- `<h1>Headline</h1>`

- **`<p>`This is a paragraph of content of some sort, that goes on, and on, and on, and on, and on, and on....**

- **`</p>`**

- **`<p>`This is another paragraph of some kind of content, that goes on, and on, and on, and on...**

- **`</p>`**

- `</body>`

- `</html>`

Paragraph Tag



- Blank lines (created with carriage returns) or extra spaces within the text are ignored.
- In other words, you must do formatting with HTML, not by formatting within the text.

Paragraph Tag

- `<p>`
- Fourscore and seven years ago, our
- fathers brought forth on this continent
- a new nation, ...
- `</p>`
- Display depends upon screen width:

Fourscore and seven years ago, our fathers brought forth on this continent a new nation,...

- or
- Fourscore and seven years ago,
our fathers brought forth on this
continent a new nation, ...

Paragraph Tag



- `<p>`Penny McIntire
Department of Computer Science
Northern Illinois University
DeKalb, IL 60115`</p>`
- displays

Penny McIntire Department of Computer Science Northern Illinois
University DeKalb, IL 60115

- because spaces and line breaks in the text are ignored.

Paragraph Tag



- On the other hand, using `<p>...</p>` on every line can insert blank lines where you don't want them...

Paragraph Tag

- `<p>Penny McIntire</p>`
- `<p>Department of Computer Science</p>`
- `<p>Northern Illinois University</p>`
- `<p>DeKalb, IL 60115</p>`

• displays

- Penny McIntire
- Department of Computer Science
- Northern Illinois University
- DeKalb, IL 60115

Using `<p>` tags on each line.

Break Tag



- **
** Forces a line break, with no extra blank line, like a carriage return.
- Was deprecated and **
** for XHTML, but **
** is fine for HTML5.
- Inserted within text within other tags like **<p>...</p>**.
- Useful for times when the additional spacing caused by **<p>** would be irritating...

Break Tag



`<p>`

Penny McIntire`
`

Department of Computer Science`
`

Northern Illinois University`
`

DeKalb, IL 60115

`</p>`

displays

Penny McIntire

Department of Computer Science

Northern Illinois University

DeKalb, IL 60115

- **Other HTML tags**

Horizontal Rule

```
<hr      size = "n"           height in pixels  
        width = "m" or "m%"  
        align = "left" or "center" or "right"  
        color = "#hhhhhh"  
        noshade >
```

- Horizontal *rule* (dividing line) n pixels high, and m pixels or m percent of container width.
- align is deprecated in favor of CSS.
- color works only in IE, but degrades gracefully to the default color in other browsers. Deprecated in favor of CSS's color (which also works only in IE).
- noshade is optional – shades the line with a 3-D speed bump effect if omitted and line is tall enough.

Center Tag



`<center>...</center>`

- Causes the items within this *container* (a tag with both start and end tags) to be centered horizontally on the page.
- Deprecated in favor of CSS's

`text-align`

or

`margin-left:auto; margin-right:auto;`

Center Tag



- Example:

<center>

`<h1>Level 1 header </h1>`

`<h2>Level 2 header </h2>`

`<h3>Level 3 header </h3>`

</center>

`<h4>Level 4 header </h4>`

`<h5>Level 5 header </h5>`

`<h6>Level 4 header </h6>`

Center Tag



- Displays:

Level 1 header

Level 2 header

Level 3 header

Level 4 header

Level 5 header

Level 6 header

Align Attribute



`align = "center"/"left"/"right"/"justify"`

- An *attribute*, not a *tag*, so embedded within another tag, such as an `<h1>` tag.
- Used to align a single element such as a header or a paragraph.
- `align = "center"` works the same as `<center>...</center>` except `<center>` is used to enclose multiple tags all at once.
- Deprecated in favor of CSS.

Align Attribute



- Example:

```
<h1 align = "center">This is a level 4 header </h1>
```

```
<h1 align = "left">This is a level 4 header </h1>
```

```
<h1 align = "right">This is a level 4 header </h1>
```

displays

This is a level 4 header

This is a level 4 header

This is a level 4 header

Text Styles



- `<u>...</u>` underline
 - Avoid, because underline is generally reserved for links.
 - Deprecated in favor of CSS's text-decoration.
- `...` boldface
- `...`
 - `` is deprecated.
 - CSS's font-weight a better option than ``? (I disagree.)

Text Styles



- **...** italics (“emphasized”)
 - **<i>...</i>** (deprecated)
 - CSS’s font-style a better option? (I disagree.)
- **<tt>...</tt>** fixed width font
 - CSS’s typeface options are preferred.
- **...** strike-through
 - CSS’s text-decoration a better option? (Maybe).
- **^{...}** superscript
 - CSS’s vertical-align a better option? (I disagree).
- **_{...}** subscript
 - CSS’s vertical-align a better option? (I disagree).

Text Styles



- Example:

```
<h1 align = "center">
```

```
<u> This is a</u><em> level 1<strong> header</strong> </em> </h1>
```

displays

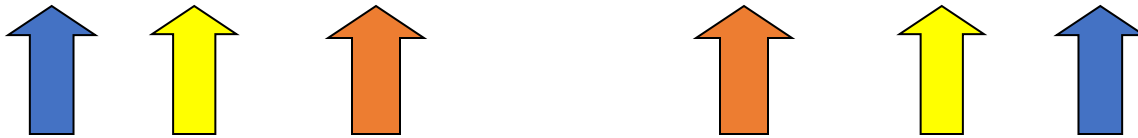
This is a *level 1 header*

Text Styles



- Well-formed HTML closes tags in the reverse order in which they are opened.
- Correct:

`<u> Hi! </u>`



- Incorrect because overlapping:

`<u> Hi!</u> `



Font Tag



```
<font color = "#hhhhh"  
  "-n"          face = "font name">  
  ...paragraph text here...
```

```
size = "n" or "+n" or
```

```
</font>
```

- Overrides the defaults in the <body> element.
- **Deprecated (viewed as the ultimate evil)!** in favor of CSS's font properties.

Font Tag



- size attribute
 - Valid size range is 1-7, with 1 the smallest and 7 the largest.
 - size = 3 is the browser default.
 - +2 (relative sizing) gives a font size of 5, if applied to the base font set in the browser.
 - If the font size has already been increased, say to a 4, then +2 results in a size 6.
 - Best to use relative sizes, for accessibility reasons.

Font Tag



- face attribute changes the typeface.
 - The typeface must exist on the user's browser.
 - You can list multiple typefaces as attributes for face – the browser checks the user's system for the first one, then the second one, etc. The last one on the list should be a standard font like Times or Ariel, or serif or san-serif.
face = "Calligrapher, Georgia, Times, serif"

Font Tag



- If the browser doesn't find any of them, it uses the default.
- Nothing to lose here if the font isn't found; it “degrades gracefully.”
- If you need a fancy font for something special, create a text graphic, save it as an image, and use the image.
- This comes with a download price and is harder to maintain in the future, so use sparingly.
- Newer browsers provide the ability to save fonts with the page.
- A huge download hit (font files can be above 100K).
- In theory, the fonts could contain viruses.

Preformatted Text



- `<pre width="n">...</pre>`
 - Preformatted text preserves the formatting in the original, including spaces, line breaks, and tabs.
 - Particularly useful for displaying code listings because the text won't wrap to fit the available space in the browser window.
 - The width attribute specifies the maximum number of characters in a line.
 - Don't use as a way of avoiding HTML/CSS formatting!