



HTML

ONE LOVE. ONE FUTURE.

1

Objectives

- ❑ Describe hypertext and HTML standards
- ❑ Understand HTML elements and markup tags
- ❑ Create the basic structure of an HTML file
- ❑ Insert an HTML comment
- ❑ Work with block-level elements
- ❑ Create lists, tables, hyperlinks and insert images
- ❑ Work with forms and inputs
- ❑ Learn HTML5 tags

2



Basic HTML

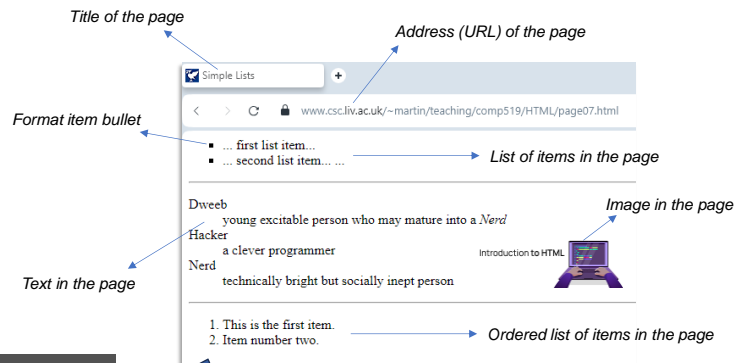
- Hypertext and HTML language
- Versions
- Language structure and elements
- Root elements in web page
- Others basic HTML tags
- HTML tags by category

3

3

Hypertext & HTML

- Hypertext refers to the fact that Web pages are more than just text
- Can contain multimedia, provide links for jumping within the same document & to other documents
- Markup refers to the fact that it works by augmenting text with special symbols (tags) that identify the document structure and content type
- HyperText Markup Language (HTML) is the language for specifying content of Web pages



```
<html>
<head>
  <title>Simple Lists</title>
</head>
<body>
  <ul style="list-style-type: square;">
    <li>... first list item... </li>
    <li>... second list item... </li>
  </ul>
  <hr>
  . . . . .
</body>
</html>
```

All is TEXT!!!



SOICT TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
School of Information and Communication Technology

4

4

HTML versions

- ❑ 1989 Tim Berners-Lee invented www
- ❑ 1991 Tim Berners-Lee invented HTML
- ❑ 1993 Dave Raggett drafted HTML+
- ❑ 1995 HTML Working Group defined HTML 2.0
- ❑ 1997 W3C Recommendation: HTML 3.2
- ❑ 1999 W3C Recommendation: HTML 4.01
- ❑ 2000 W3C Recommendation: XHTML 1.0
- ❑ 2008 WHATWG HTML5 First Public Draft
- ❑ 2012 WHATWG HTML5 Living Standard
- ❑ **2014 W3C Recommendation: HTML5**
- ❑ 2016 W3C Candidate Recommendation: HTML 5.1
- ❑ 2017 W3C Recommendation: HTML5.1 2nd Edition
- ❑ 2017 W3C Recommendation: HTML5.2

HTML Elements

- ❑ Web page = One or Set of HTML Elements
- ❑ HTML Element by Tag
 - ❖ Name & content (text)
 - ❖ Semantic *← language capability here!!!*
 - ❖ Attributes
- ❑ Nested Elements
 - ❖ Parent tag
 - ❖ Children

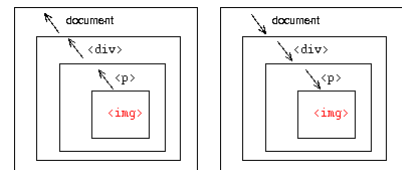
- Page layout
 - Text style (font, size, color...)
 - Image and multimedia embedded
 - Form to submit user input
 - Etc...
- ❑ Events
 - ❖ Occur by sort of interaction takes place in a web page
 - ❖ User clicks mouse on something, moving mouse over an element
 - ❖ Key pressing.
 - ❖ Something happens in browser (complete page loading, user scrolling or resizing the window, etc..).
 - ❖ Event building in nested elements
 - ❖ Event capturing

```
<table>
<tr>
<tr>
  <td>Name</td>
  <td>Department</td>
</tr>
<tr>
  <td>Hoang Pham</td>
  <td>SoICT / HUST</td>
</tr>
</table>
```

Opening tag Content Closing tag

`<p id="myId">This is a paragraph.</p>`

Attribute name Attribute value



More detail about Tags and Elements

- HTML specifies a set of *tags* that identify structure of the document and the content type
 - ❖ Not case sensitive
 - ❖ Tags are enclosed in `< >`
 - `` specifies an image
 - ❖ Most tags come in pairs, marking a beginning and ending
 - `<title>` and `</title>` enclose the title of a page
- An HTML *element* is an object enclosed by a pair (in most cases) of tags
 - ❖ `<title>My Home Page</title>` is a TITLE element
 - ❖ `This text appears bold.` is a BOLD element
 - ❖ `<p>Part of this text is bold. </p>` is a PARAGRAPH element that contains a BOLD element



Root elements in Web page: <html>

- <html> element is the root of any HTML page
- an HTML document has two main structural elements
 - ❖ HEAD contains setup information for the browser & the Web page
 - ❖ BODY contains the actual content to be displayed in the Web page

```
<html>
<!-- First file---->
<head>
  <title>My first HTML document</title>
</head>
<body>
  <p> Hello world! </p>
</body>
</html>
```

view page

HTML documents : <html> and </html> tags

Comments: <!-- and -->

HEAD section: <head> and </head> tags

BODY section: <body> and </body>



Root elements in Web page(2): <head> and <body>

- <head> element
 - Title
 - Cascading Style sheet information
 - “Meta” data, such as who authored the page, keywords
 - JavaScript code
- The <body> element
 - Paragraphs
 - Tables and lists
 - Images
 - JavaScript code
 - PHP code

HTML tags by category

- Basic HTML
- Formatting
- Images
- Lists
- Tables
- Layout (frames)
- Links
- Forms and Input
- Styles and semantics
- HTML5



Format & rendering the contents

- ❑ Text formatting
- ❑ List
- ❑ Image
- ❑ Table
- ❑ Link

11

11

Text Layout

- ❑ For the most part, layout of the text is left to the browser
 - ❖ whitespace is interpreted as a single space
 - ❖ browser automatically wraps the text to fit the window size
- ❑ Can override some text layout
 - ❖ can specify a new paragraph (starts on a new line, preceded by a blank line) using `<p>...</p>`
 - ❖ can cause a line break using the `
`
 - ❖ can force a space character using ` `

[illegible][view page](#)

12

12

Separating Blocks of Text

- Can specify headings for paragraphs or blocks of text

- ❖ `<h1>...</h1>` tags produce a large, bold heading
- ❖ `<h2>...</h2>` tags produce a slightly smaller heading
- ❖ ...
- ❖ `<h6>...</h6>` tags produce a tiny heading

- Can insert a horizontal rule

- ❖ `<hr/>` draws line across window

- Block and inline elements

- ❖ Block: render in new line
- ❖ Inline: render in the same line

`<p>Text before div<div/> text after div</p>`
`<p>Text before span text after span</p>`

Text trước div
và text sau div
Text trước span và text sau span

```
<html>
<!-- CS443 page03.html 15/08/06 -->
<head>
  <title>Blocks of Text</title>
</head>

<body>
  <h1>Major heading 1</h1>
  <p>
    Here is some text.
  </p>

  <h2>Subheading</h2>
  <p>
    Here is some subtext.
  </p>

  <hr/>

  <h1>Major heading 2</h1>
  <p>
    Here is some more text.
  </p>
</body>
</html>
```

view page



SOICT TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
School of Information and Communication Technology

13

13

The Basic Web page – A Worked Example

```
<html>
<!-- CS443 page22.html 17.10.14 -->
<head>
  <title> Bill Smiggins Inc. </title>
</head>
<body>
  <h1>Bill Smiggins Inc.</h1>
  <h2>About our Company...</h2>
  <p>This Web site provides clients, customers,
    interested parties and our staff with all of
    the information that they could want on
    our products, services, success and failures.
  </p>
  <hr/>
  <h3> Products </h3>
  <p> We are probably the largest
    supplier of custom widgets, thingummybobs, and bits
    and pieces in North America. </p>
  <hr/>
</body>
</html>
```

view page



SOICT TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
School of Information and Communication Technology

14

14

Text Appearance

□ can specify styles for fonts

- ❖ `... ` specify bold
- ❖ `<i>... </i>` specify italics
- ❖ `<big>... </big>` increase the size
- ❖ `<small>... </small>` decrease the size
- ❖ `... ` put emphasis
- ❖ `... ` put more emphasis
- ❖ `_{...}` specify a subscript
- ❖ `^{...}` a superscript

```
<html>
<!-- CS443 page25.html 15.08.06 -->
<head>
  <title>Text Variations and Escape
  Sequences</title>
</head>
<body>
  <h1>Text Variations</h1>
  <p>We can use <b>simple</b> tags to
    <i>change</i> the appearance of
    <strong>text</strong> within
    <tt>Web pages</tt>.
    Even super<sup>script</sup>
    and sub<sub>scripts</sub> are
    <em>supported</em>.</p>

  <h1>Text Escape Sequences</h1>
  <p>
    &amp; &lt; &gt; &quot; &copy;
  </p>
  <h1>Preformatted text</h1>
  <pre>
    University of Liverpool
    Department of Computer Science
    Ashton Building, Ashton Street
    Liverpool, L69 3BX, UK
  </pre>
</body>
</html>
```

view page



SOICT TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
School of Information and Communication Technology

15

15

Lists

□ There are 3 different types of list elements

- ❖ `...` specifies an ordered list
 - `` identifies each list item
- ❖ `...` specifies unordered list (using a bullet for each)
 - `` identifies each list item
- ❖ `<dl>...</dl>` specifies a definition list
 - `<dt>` identifies each term
 - `<dd>` identifies its definition

```
<html>
<!-- CS443page07.html 23.09.08 -->
<head> <title>(Sort of) Simple Lists</title>
  <style type="text/css">
    .my_li:before { content: counter(list) ": ";
                  counter-increment: list; }
  </style> </head>
<body>

  <ul style="list-style-type: square;">
    <li> ... first list item... </li>
    <li> ... second list item... </li>
  </ul>
  <dl>
    <dt> Dweeb </dt>
    <dd> young excitable person who may
    mature into a <em>Nerd</em> </dd>
    <dt> Hacker </dt>
    <dd> a clever programmer </dd>
    <dt> Nerd </dt> <dd> technically bright but
    socially inept person </dd>
  </dl>
  <ol style="list-style-type: none;
    counter-reset: list 29;" >
    <li class="my_li">Makes first item number 30.</li>
    <li class="my_li">Next item continues to number
    31.</li>
  </ol>

</body>
</html>
```

view page



SOICT TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
School of Information and Communication Technology

16

16

Images

```
<html>
<!-- CS443 page10.html 18.09.13 -->
<head>
  <title>Image example</title>
</head>
<body>


<p>The Anglican Cathedral of Liverpool</p> </body>
</html>
```

```

```

[view page](#)


SOICT TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
School of Information and Communication Technology

17

17

Images (cont.)

- **src** - specifies the file name (and can include a URL)
- **width** and/or **height** - dimensions in pixel
- **title** - displayed when the mouse is “hovered” over the picture
- **alt** - text that is displayed when the image is missing, can’t be loaded



SOICT TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
School of Information and Communication Technology

18

18

Tables

`<table>...</table>` specify a table element

`<tr>...</tr>` specify a row in the table

`<td>...</td>` specify table data (i.e., each column entry in the table)

```
<html>
<!-- CS443 page11.html 17.10.14 -->
<head>
  <title>Tables</title>
</head>
<body>
<h2>A Simple Table</h2>
  <table>
    <tr>
      <td> Left Column </td>
      <td> Right Column </td>
    </tr>
    <tr>
      <td> Some data </td>
      <td> Some other data </td>
    </tr>
  </table>
</body>
</html>
```

[view page](#)


SOICT TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
School of Information and Communication Technology

19

19

Layout in a Table

□ Border on tables using the “style” attribute

`<table style= "border: 1px solid;">`

□ Horizontal & vertical layout within cells

`<td style= "text-align:center">`

`<td style= "vertical-align: bottom">`

□ Layout to an entire row

`<tr style="text-align: center">`

`<tr style="vertical-align: top">`

```
<html>
<!-- CS443 page12.html 17.10.14 -->
<head>
  <title>Table Layout</title>
</head>
<body>
  <table style="border: 1px solid;">
    <tr style="text-align: center;">
      <td style="border: 1px solid;">
        Left<br/>Column</td>
      <td style="border: 1px solid;
        vertical-align: top;">
        Right Column</td>
    </tr>
    <tr>
      <td style="border: 1px solid;">
        Some data</td>
      <td style="border: 1px solid;">
        Some data</td>
    </tr>
  </table>
</body>
</html>
```

[view page](#)


SOICT TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
School of Information and Communication Technology

20

20

Table Width

- by default, the table is sized to fit the data
- can override & specify the width of a table relative to the page

For example

```
<table style="width: 60%">
```

```
<html>
<!-- CS443 page13.html 17.10.14 -->
<head>
  <title>Table Width</title>
</head>

<body>
<table style="width: 100%;">
  <tr>
    <td>left-most </td>
    <td style="text-align: right;">
      right-most</td>
  </tr>
</table>
</body>
</html>
```

[view page](#)


SOICT TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
School of Information and Communication Technology

21

21

Other Table Attributes

- ❖ Can control the space between cells & margins within cells using "padding" attribute
- ❖ Can add headings
`<th>` is similar to `<td>` but displays heading centered in bold
- ❖ Can have data that spans more than one column
`<td colspan="2">`
- ❖ Can span more than one row
`<td rowspan="2">`

```
<html>
<!-- CS443 page14.html 17.10.14 -->
<head>
  <title>Table Formatting</title>

  <style type="text/css" media="screen">
    table { border: 1px solid; padding: 1px;}
    th, td { border: 1px solid; padding: 10px;
      text-align: center; }
  </style>
</head>

<body>
  <table>
    <tr>
      <th>HEAD1</th> <th>HEAD2</th>
      <th>HEAD3</th>
    </tr>
    <tr>
      <td>one</td> <td>two</td> <td>three</td>
    </tr>
    <tr>
      <td rowspan="2"> four </td>
      <td colspan="2"> five </td>
    </tr>
    <tr>
      <td> six </td> <td> seven </td>
    </tr>
  </table>
</body>
</html>
```

[view page](#)


SOICT TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
School of Information and Communication Technology

22

22

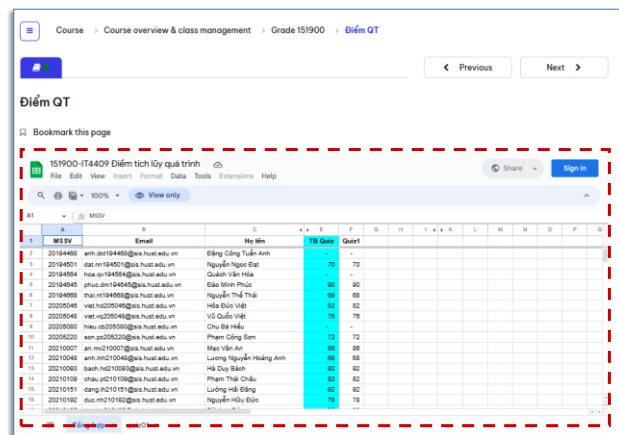
Page layout: HTML frames

- Typical layout: document by set of frames
 - ❖ Header
 - ❖ Navigator
 - ❖ Main
 - ❖ Footer
- Each frame is considered as a separated web page
- Cross-frame interaction and rendering
 - ❖ Click link in navigator → render in main frame
 - ❖ Header and footer do not change while loading main frame
- Old HTML versions:
 - ❖ <frame>
 - ❖ <frameset>
 - ❖ Not used anymore from HTML5



<iframe> tag

- An inline frame is used to embed another document within the current HTML document.
- Independent execution (scripting) and rendering between host document and embedded document
- Impossible to interfere to embedded document



<div> and Tags

- ❑ Problem: font properties apply to whole elements, which are often too large
 - ❖ Solution: a new tag to define an element in the content of a larger element - ****
 - ❖ is an inline element used to mark up a part of text

```
<p> Now is the <span> best time </span> ever! </p>
```

- ❑ Use to apply a document style sheet definition to its content
 - ❖ The tag is similar to other HTML tags, they can be nested and they have id and class attributes
- ❑ The <div> element is often used **as a container** to group sections of a web page together.
- ❑ The **** tag is an inline container used to mark up a part of a text, or a part of a document.

```
<style type = "text/css">
  .bigred {font-size: 24pt;
    font-family: Ariel; color: red}
</style>
...
<p> Now is the <span class="bigred">
  best time </span> ever!
</p>
```

[view page](#)


SOICT TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
School of Information and Communication Technology

25

25



Server interaction

- ❑ Hyperlink
- ❑ Link with user data
- ❑ Input data by HTML form
- ❑ Form customization by attributes
- ❑ Form data validation
- ❑ Form elements and input types

26

26

Hyperlinks

- Perhaps the most important HTML element is the hyperlink, or ANCHOR
- `...`
 - ❖ where URL is the Web address of the page
 - ❖ if the page is accessed over the Web, must start with `http://`
 - ❖ if not there, the browser will assume it is the name of a local file
- Target for rendering link contents
 - ❖ Causes the page to be loaded in a new Window or in the other frame of page layout
 - ❖ `...`
 - ❖ `"_self"`: Default. Opens the document in the same window/tab as it was clicked
 - ❖ `"_blank"`: Opens the document in a new window or tab
 - ❖ `"_parent"`: Opens the document in the parent frame
 - ❖ `"_top"`: Opens the document in the full body of the window

```
<html>
<!-- CS443page08.html 17.10.14 -->

<head>
  <title>Hyperlinks</title>
</head>

<body>
  <p>
    <a href="http://www.liv.ac.uk">
      The University of Liverpool</a>
    <br/>
    <a href="page07.html"
      target="_blank">
      Open page07 in a new window</a>
  </p>
</body>

</html>
```

[view page](#)

Hyperlinks (cont.)

- For long documents, you can even have links to other locations in that same document (internal link)
 - ❖ `<xxxx id="ident">...</xxxx>`
 - `ident` is a variable for identifying this location
 - `"xxxx"` can be any HTML element
 - ❖ `...`
 - will then jump to that location within the file
 - ❖ `...`
 - can jump into the middle of another file just as easily

```
<html>
<!-- CS443 page09.html 21.09.12 -->

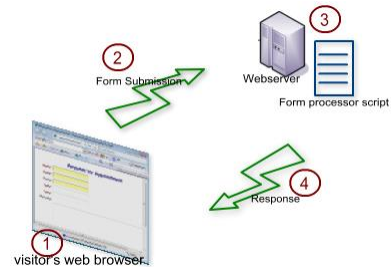
<head>
  <title>Internal Links in a Page</title>
</head>

<body>
  <p>
    [ <a href="#HTML">HTML</a> |
      <a href="#HTTP">HTTP</a> |
      <a href="#IP">IP</a> |
      <a href="#TCP">TCP</a> ]
  </p>
  <p>
    Computer acronyms:
    <dl>
      <dt id="HTML">HTML</dt>
      <dd>HyperText Markup Language
      <dt id="HTTP">HTTP</dt>
      <dd>HyperText Transfer Protocol...</dd>
      <dt id="IP">IP</dt>
      <dd>Internet Protocol...</dd>
      <dt id="TCP">TCP</dt>
      <dd>Transfer Control Protocol...</dd>
    </dl>
  </p>
</body>
</html>
```

[view page](#)

How does HTML Form work

1. User loads the form page in web browser
 - ❖ HTML page contains `<form>` tag
 - ❖ Form has several data fields with data type
 - ❖ User fills the data to form by correct data type
2. The form submission data is sent to the web server
 - ❖ `"action"` attribute to target the request
 - ❖ GET method is by default
 - ❖ Form fields data are conveyed by HTML request parameters
3. The web server processes the request
 - ❖ Extract parameters value
 - ❖ Generate HTML response message
4. A response is sent back to the browser
 - ❖ Check the status code
 - ❖ Render the response data



`http://.../action_page?user=Hoang&pass=abcdef`

Login name:

Password:

```

<form action="/action_page">
  Login name:<input type="text" name="user"><br>
  Password: <input type="password" name="pass"><br>
  <input type="submit" value="Click to Submit">
</form>
  
```



SOICT TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
School of Information and Communication Technology

29

29

Some important attributes of `<form>`

- ❑ `"action"`: to target the request
- ❑ `"method"`: method in HTTP request
 - ❖ GET (if omitted, by default)
 - ❖ POST, DELETE, etc...
- ❑ `"target"`: the same as in hyperlink
- ❑ `"autocomplete"`:
 - ❖ on/off
 - ❖ Accept browser suggests to autofill
- ❑ `"novalidate"`:
 - ❖ To turn off the data validation by browser
 - ❖ By default, browser validate user data input by field data type



SOICT TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
School of Information and Communication Technology

30

30

Input data type and auto validation

- Depend on browser capability
- Some simple data type validation before submitting
 - ❖ Email format not correct
 - ❖ URL format not correct
 - ❖ Password too simple
- Missing the feature to customize warning messages
- Turn off auto validation
 - ❖ "novalidate" attribute
 - ❖ Catch event (submit button click) to "manually" validate data input. Need JavaScript embedded.
 - ❖ Submit to server and validate at server side

Auto validate

By input type, and depend on browser capability:

Enter your email:

! Please include an '@' in the email address. '123' is missing an '@'.

```
<h1>Auto validate</h1>
<p>By input type, and depend on browser capability:</p>
<form action="/action_page" novalidate >
  Enter your email:
  <input type="email" name="email"
    onclick="validate_data_input()" >
  <input type="submit">
</form>
```



SOICT TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
School of Information and Communication Technology

31

31

Form elements and input types

- To define form fields with data type

□ Elements:

- ❖ <input>
- ❖ <label>
- ❖ <select>
- ❖ <textarea>
- ❖ <button>
- ❖ <fieldset>
- ❖ <legend>
- ❖ <datalist>
- ❖ <output>
- ❖ <option>
- ❖ <optgroup>

□ Input types

- ❖ <input type="button">
- ❖ <input type="checkbox">
- ❖ <input type="color">
- ❖ <input type="date">
- ❖ <input type="datetime-local">
- ❖ <input type="email">
- ❖ <input type="file">
- ❖ <input type="hidden">
- ❖ <input type="image">
- ❖ <input type="month">
- ❖ <input type="number">
- ❖ <input type="password">
- ❖ <input type="radio">
- ❖ <input type="range">
- ❖ <input type="reset">
- ❖ <input type="search">
- ❖ <input type="submit">
- ❖ <input type="tel">
- ❖ <input type="text">
- ❖ <input type="time">
- ❖ <input type="url">
- ❖ <input type="week">

Select month:

September 2024

2024

Jan	Feb	Mar	Apr
May	Jun	Jul	Aug
Sep	Oct	Nov	Dec

[Clear](#) [This month](#)

Select time:

02:42 AM

12	40	PM
01	41	AM
02	42	
03	43	
04	44	



SOICT TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
School of Information and Communication Technology

32

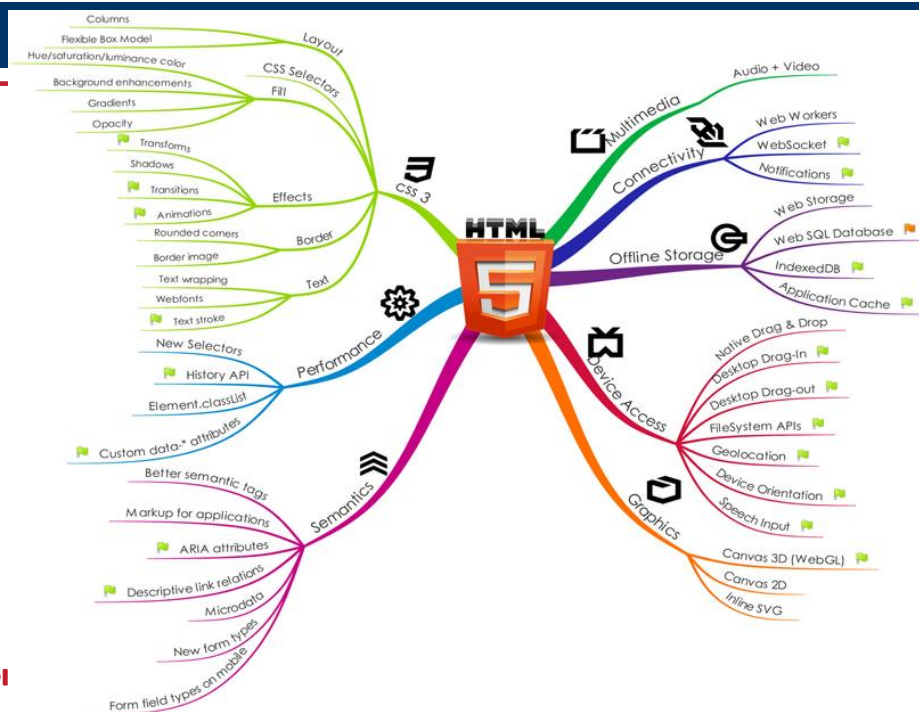
32



HTML5

33

33



34

34

HTML5 New Tags

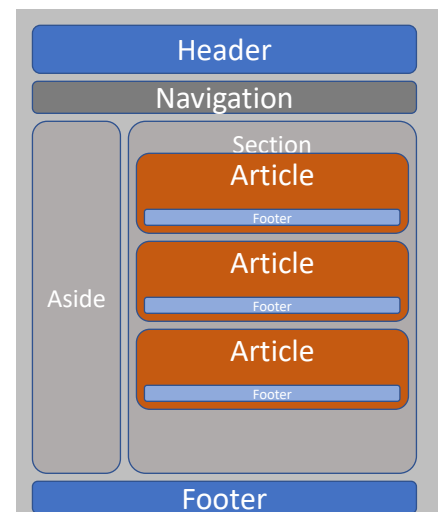
- HTML 5 specify DOCTYPE as follows: `<!DOCTYPE html>`
- specify Character Encoding as follows: `<meta charset="UTF-8">`
- New tags introduced in HTML5 for better structure
 - ❖ **header** – This tag represents the header of a section.
 - ❖ **footer** – This tag represents a footer for a section and can contain information about the author, copyright information, etc.
 - ❖ **nav** – This tag represents a section of the document intended for navigation.
 - ❖ **dialog** – This tag can be used to mark up a conversation.
 - ❖ **figure** – This tag can be used to associate a caption together with some embedded content, such as a graphic or video.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>...</title>
  </head>
  <body>
    <header>...</header>
    <nav>...</nav>
    <article>
      <section>...</section>
    </article>
    <aside>...</aside>
    <figure>...</figure>
    <footer>...</footer>
  </body>
</html>
```



HTML5 New Tags

- **section** – This tag represents a generic document or application section. It can be used together with h1-h6 to indicate the document structure.
- **article** – This tag represents an independent piece of content of a document, such as a blog entry or newspaper article.
- **aside** – This tag represents a piece of content that is only slightly related to the rest of the page.



HTML5 New Tags

□ New input elements:

button
checkbox
color
date
datetime
datetime-local
email
file
hidden
image

month
number
password
radio
range
reset
search
submit
tel
text
time
url
week

• Form

Number:

Range:

Month:

DateTime: UTC

DateTime-Local:

Date:

Calendar view: May 2010

Week	Mon	Tue	Wed	Thu	Fri	Sat	Sun
17	26	27	28	29	30	1	2
18	3	4	5	6	7	8	9
19	10	11	12	13	14	15	16
20	17	18	19	20	21	22	23
21	24	25	26	27	28	29	30
22	31	1	2	3	4	5	6

Today None



SOICT TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
School of Information and Communication Technology

37

37

HTML5 New Tags

□ HTML5 offers new elements for media content:

Tag	Description
<audio>	Defines sound content
<video>	Defines a video or movie
<source>	Defines multiple media resources for <video> and <audio>
<embed>	Defines a container for an external application or interactive content (a plug-in)
<track>	Defines text tracks for <video> and <audio>

```
<audio controls="true">
  <source src="audiodemo.ogg" />
  <source src=" audiodemo.mp3" />
  <source src=" audiodemo.wav" />
  Not supported.
</audio>
```



```
<video src="video.ogv" controls poster="poster.jpg" width="320" height="240">
  <a href="video.ogv">Download movie</a>
</video>
```



SOICT TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
School of Information and Communication Technology

38

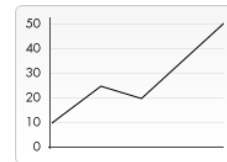
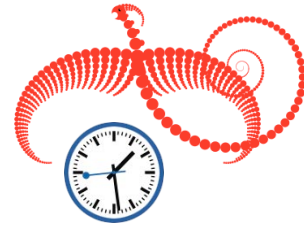
38

HTML5 New Tags

□ <canvas> element:

Tag	Description
<canvas>	Used to draw graphics, on the fly, via scripting (usually JavaScript)

```
function draw() {
  var ctx = document.getElementById('canvas').getContext('2d');
  var img = new Image();
  img.onload = function(){
    ctx.drawImage(img,0,0);
    ctx.beginPath();
    ctx.moveTo(30,96);
    ctx.lineTo(70,66);
    ctx.lineTo(103,76);
    ctx.lineTo(170,15);
    ctx.stroke();
  }
  img.src = 'images/backdrop.png';
}
```



SOICT TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
School of Information and Communication Technology

39

39

HTML Contents & Presentation

- Data & Data Structure
- CSS introduction



hust.edu.vn



fb.com/dhbkhn

40

40

Content vs. Presentation

- Most HTML tags define content type, independent of presentation.
 - ❖ exceptions? (e.g. ` ` for bold text and `<i> </i>` for italicized text)
- Style sheets associate presentation formats with HTML elements.
 - ❖ CSS1: developed in 1996
 - ❖ CSS2: released in 1998
 - ❖ CSS3: introduced in 1999, latest version (as of 2022)
- The trend has been towards an increasing separation of the content of webpages from the presentation of them.
- Style sheets allow us to maintain this separation, which allows for easier maintenance of webpages, and for a consistent look across a collection of webpages.



Content vs. Presentation (cont.)

- Style sheets can be used to specify how tables should be rendered, how lists should be presented, what colors should be used on the webpage, what fonts should be used and how big/small they are, etc.
- HTML style sheets are known as *Cascading Style Sheets*, since can be defined at three different levels
 1. *inline* style sheets apply to the content of a single HTML element
 2. *document* style sheets apply to the whole BODY of a document
 3. *external* style sheets can be linked and applied to numerous documents, might also specify how things should be presented on screen or in print *lower-level style sheets can override higher-level style sheets*
- User-defined style sheets can also be used to override the specifications of the webpage designer. These might be used, say, to make text larger (e.g. for visually-impaired users).



Same rendering, but different meaning

□ Same rendering, but **different meaning**

- ❖ Same rendering, but ``different meaning``
- ❖ Same rendering, but ``different meaning``

Only text style,
ambiguous meaning!!

□ Same rendering, but *different meaning*

- ❖ Same rendering, but `<i>`different meaning`</i>`
- ❖ Same rendering, but ``different meaning``

Standardize both text style
and semantic by tagging

➔ Web 3.0 (semantic web)

Inline Style Sheets

- Using the `style` attribute, you can specify presentation style for a single HTML element
- within tag, list sequence of `property:value` pairs separated by semi-colons

font-family:Courier,monospace
font-style:italic
font-weight:bold
font-size:12pt font-size:large font-size:larger

color:red color:#000080
background-color:white

text-decoration:underline
text-decoration:none
text-align:left text-align:center
text-align:right text-align:justify
vertical-align:top vertical-align:middle
vertical-align:bottom

text-indent:5em text-indent:0.2in

```
<html>
<!-- CS443 page17.html 17.10.14 -->

<head>
  <title>Inline Style Sheets</title>
</head>

<body>
  <p style="font-family:Arial,sans-
    serif; text-align:right">This is a
    right-justified paragraph in a sans
    serif font (preferably Arial), with some
    <span style="color:green">green
  </span>.
  </p>

  <p>And <a style="color:red;
    text-decoration:none;
    font-size:larger;"
    href="page01.html">here</a>
    is a formatted link.
  </p>
</body>
</html>
```

view page

Inline Style Sheets (cont.)

```
<html>
<!-- CS443 page18.html 17.09.09 -->

<head>
  <title>Inline Style Sheets</title>
</head>

<body>
  <p>Here is an image
    
    embedded in text.
  </p>

  <ol style="list-style-type:upper-alpha">
    <li> one thing</li>
    <li> or another</li>
    <ul style="list-style-type:square;
              whitespace:pre">
      <li> with this</li>
      <li> or that</li>
    </ul>
  </ol>
</body>
</html>
```

[view page](#)

•more style properties & values

margin-left:0.1in margin-right:5%
margin:3em
padding-top:0.1in padding-bottom:5%
padding:3em

border-width:thin border-width:thick
border-width:5
border-color:red
border-style:dashed border-style:dotted
border-style:double border-style:none

whitespace:pre

list-style-type:square
list-style-type:decimal
list-style-type:lower-alpha
list-style-type:upper-roman

45

45

Inline Style Sheets (cont.)

□ style sheets can be applied to tables for interesting effects

```
<html>
<!-- CS443 page19.html 17.10.14 -->

<head>
  <title> Inline Style Sheets </title>
</head>

<body>
  <table style="font-family:Arial,sans-serif">
    <caption style="color:red;
                  font-style:italic;
                  text-decoration:underline">
      Student data. </caption>
    <tr style="background-color:red">
      <th> name </th> <th> age </th>
    </tr>
    <tr>
      <td> Chris Smith </td> <td> 19 </td>
    </tr>
    <tr>
      <td> Pat Jones </td> <td> 20 </td>
    </tr>
    <tr>
      <td> Doogie Howser </td> <td> 9 </td>
    </tr>
  </table>
</body>
</html>
```

[view page](#)

46



46

Document Style Sheets

- Inline style sheets apply to individual elements in the page.
 - ❖ using inline style directives can lead to inconsistencies, as similar elements are formatted differently
 - e.g., we might like for all `<h1>` elements to be centered
 - ❖ inline definitions mix content & presentation
 - ➔ violates the general philosophy of HTML
- As a general rule, inline style sheet directives should be used as sparingly as possible
- Alternatively, document style sheets allow for a cleaner separation of content and presentation.
 - ❖ style definitions are placed in the `<head>` of the page (within `STYLE` tags)
 - ❖ can apply to all elements, or a subclass of elements, throughout the page



Document Style Sheets

- document style sheets ensure that similar elements are formatted similarly
- can even define subclasses of elements and specify formatting
 - `p.indented` defines subclass of paragraphs
 - inherits all defaults of `<p>`
 - adds new features
- to specify this newly defined class, place `class="ID"` attribute in tag

- note how "clean" the `<body>` element is

```
<html>
<!-- CS443 page20.html 17.10.14 -->

<head>
  <title>Document Style Sheets</title>
  <style type="text/css">
    h1 {color:blue;
        text-align:center}
    p.indented {text-indent:0.2in}
  </style>
</head>

<body>
  <h1> Centered Title </h1>

  <p class="indented">This paragraph
will have the first line indented, but
subsequent lines will be flush. </p>

  <p>This paragraph will not be
indented.
</p>

  <h1> The End </h1>

</body>
</html>
```

[view page](#)


Document Style Sheets (cont.)

- document style sheets are especially useful in formatting tables
- effectively separates content from presentation
- what if you wanted to right-justify the column of numbers?
- what if you changed your mind?

```
<html>
<!-- CS443 page21.html 17.10.14 -->

<head>
  <title> Inline Style Sheets </title>
  <style type="text/css">
    table {font-family:Arial,sans-serif}
    caption {color:red;
              font-style:italic;
              text-decoration:underline}
    th {background-color:red}
  </style>
</head>

<body>
  <table>
    <caption> Student data. </caption>
    <tr><th> name </th>          <th> age</th></tr>
    <tr><td> Chris Smith </td>    <td> 19 </td></tr>
    <tr><td> Pat Jones </td>      <td> 20 </td></tr>
    <tr><td> Doogie Howser </td>  <td> 9 </td></tr>
  </table>
</body>
</html>
```

[view page](#)


SOICT TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
School of Information and Communication Technology

49

49

Pseudo-Elements

- pseudo-elements are used to address sub-parts of elements
 - ♦ can specify appearance of link in various states
 - :visited :active :hover
 - ♦ can specify format of first line in page or paragraph
 - :first-line
 - ♦ can specify format of first letter in page or paragraph
 - :first-letter

□ **Danger** : changing the look of familiar elements is confusing

```
<html>
<!-- CS443 page23.html 17.10.14 -->

<head>
  <title>Title for Page</title>
  <style type="text/css">
    a {color : red;
       text-decoration : none;
       font-size : larger}
    a:visited {color : black}
    a:active {color : orange}
    a:hover {color : blue}
    p:first-letter {font-size : large;
                    color : white;
                    background-color : darkblue}
  </style>
</head>

<body>
  <p> Welcome to my Web page. I am so
    happy you are here.
  </p>
  <p> Be sure to visit
    <a href="http://www.cnn.com">CNN</a>
    for late-breaking news.
  </p>
</body>
</html>
```

[view page](#)


SOICT TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
School of Information and Communication Technology

50

50

External Style Sheets

- modularity is key to the development and reuse of software
 - ❖ design/implement/test useful routines and classes
 - ❖ package and make available for reuse
 - ❖ saves in development cost & time
 - ❖ central libraries make it possible to make a single change and propagate the changes
- external style sheets place the style definitions in separate files
 - ❖ multiple pages can link to the same style sheet, consistent look across a site
 - ❖ possible to make a single change and propagate automatically
 - ❖ represents the ultimate in content/representation separation



Modularity & Style Sheets

- Ideally, the developer(s) of a Web site would place all formatting options in an external style sheet.
- All Web pages link to that same style sheet for a uniform look.
 - ❖ simplifies Web pages since only need to specify structure/content tags
 - ❖ Note: no <style> tags are used in the external style sheet

```
/* myStyle.css CS443 02.09.05 */
h1 {color : blue; text-align : center}
p.indented {text-indent:0.2in}
```

```
<html>
<!-- CS443 page26.html 17.10.14 -->

<head>
  <title>Title for Page</title>
  <link rel="stylesheet"
        type="text/css"
        href="myStyle.css"
        title="myStyle" />
</head>

<body>
  <h1>Centered Title</h1>

  <p class="indented">This paragraph
  will have the first line indented, but
  subsequent lines will be flush.</p>

  <p>This paragraph will not be
  indented.
  </p>

  <h1>The End</h1>

</body>
</html>
```

[view page](#)
