WEB SYSTEM AND TECHNOLOGIES

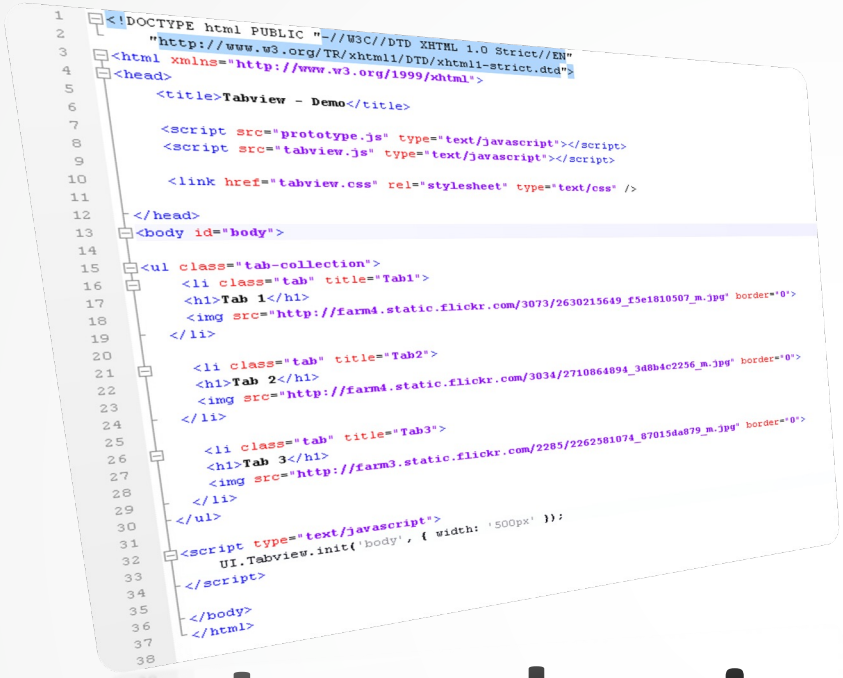
Chapter 2

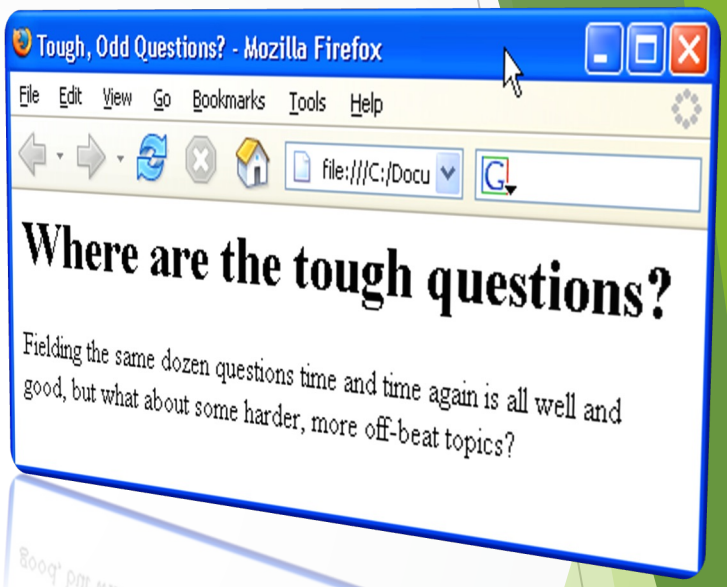
HTML

Hypertext Markup Language

Content

1. Introduction to HTML
2. HTML Structure
3. HTML Basics
4. HTML Table
5. HTML Form





Introduction to HTML

HTML DOCUMENT STRUCTURE IN DEPTH

Introduction to HTML

- HTML is a MARKUP language
- Using HTML tags and elements, we can:
  - Control the appearance of the page and the content
  - Publish online documents and retrieve online information using the links inserted in the HTML document
  - Create on-line forms. These forms can be used to collect information about the user, conduct transactions, and so on

The timeline of HTML

- **1991 -** Tim Berners-Lee invents HTML 1.0
- **1993 -** HTML 1.0 is released.
- **1995 -** HTML 2.0 is published
- **1997 -** HTML 3.0 was invented
- **1999 -** The widely-used HTML 4.0 comes out. It is very successful.
- **2014 -** HTML 5.0 is released and used worldwide. It is said to be the extended version of HTML 4.01 which was published in 2012.

What is Web page?

- Web pages are text files containing HTML
- HTML – Hyper Text Markup Language
  - A notation for describing
    - document structure (semantic markup)
    - formatting (presentation markup)
  - Looks (looked?) like:
    - A Microsoft Word document
- The markup tags provide information about the page content structure

Create HTML Pages

- An HTML file must have an .htm or .html file extension
- HTML files can be created with text editors:
  - NotePad, NotePad ++, …
- HTML editors (WYSIWYG Editors - **What You See Is What You Get**):
  - Visual Studio Code: https://code.visualstudio.com/download
  - Sublime Text

    https://www.sublimetext.com/download
  - IntelliJ IDEA
  - …

Create HTML Pages (cont.)

- Web Browser: *Chrome, Firefox, Safari, IE…*
- Website: create and run
  - Web Editor: sample.html
  - Right-click on the HTML file you wish to see and select "Open with" from the menu. You may see a long list of apps (Web browser) from which to choose to open your file.

Concepts in HTML

- Tags
  - Opening tag and closing tag
  - The smallest piece in HTML
- Attributes
  - Properties of the tag
  - Size, color, etc…
- Elements
  - Combination of opening, closing tag and attributes

HTML Structure

- HTML is comprised of “elements” and “tags”
  - Begins with **<html>** and ends with **</html>**
- Elements (tags) are nested one inside another:

```
<html>
    <head></head>
    <body></body>
</html>
```

- Tags have attributes:

```
<img src="logo.jpg" alt="logo" />
```

- HTML describes structure using two main sections: **<head>** and **<body>**

HTML Code Formatting

- The HTML source code should be formatted to increase readability and facilitate debugging.
  - Every block element should start on a new line.
  - Every nested (block) element should be indented.
  - Browsers ignore multiple whitespaces in the page source, so formatting is harmless.
- For performance reasons, formatting can be sacrificed.

First HTML Page

- File name: hello.html
- Content:

```
<!DOCTYPE html>
<html>
  <head>
    <title>My First HTML Page</title>
  </head>
  <body>
    <p>Hello</p>
    <p>This is some text...</p>
  </body>
</html>
```

First HTML Page: tags

```
<!DOCTYPE html>
<html>
  <head>
    <title>My First HTML Page</title>
  </head>
  <body>
     <p>Hello</p>
     <p>This is some text...</p>
  </body>
</html>
```

Opening tag

Closing tag

An HTML element consists of an opening tag, a closing tag and the content inside

First HTML Page: header

HTML header

```
<!DOCTYPE html>
<html>
  <head>
    <title>My First HTML Page</title>
  </head>
  <body>
    <p>Hello</p>
    <p>This is some text...</p>
  </body>
</html>
```

First HTML Page: body

```
<!DOCTYPE html>
<html>
  <head>
    <title>My First HTML Page</title>
  </head>
  <body>
    <p>Hello</p>
    <p>This is some text...</p>
  </body>
</html>
```

HTML body

Tag Attributes

Tags can have attributes

- Attributes specify properties and behavior
- Example:

Attribute `alt` with value "logo"

```
<img src="logo.gif" alt="logo" id="h1"/>
```

- Few attributes can apply to every element:
  - `id, style, class, title`
  - The `id` is unique in the document
  - Content of `title` attribute is displayed as hint when the element is hovered with the mouse
  - Some elements have obligatory attributes: `href, lang, alt, src, title, …`

The <!DOCTYPE> Declaration

- HTML documents must start with a document type definition (DTD)
  - It tells web browsers what type is the served code
  - Possible versions: HTML 4.01, XHTML 1.0 (Transitional or Strict), XHTML 1.1, HTML 5
- Example:

```
<!DOCTYPE html>
```

- See http://w3.org/QA/2002/04/valid-dtd-list.html for a list of possible doctypes

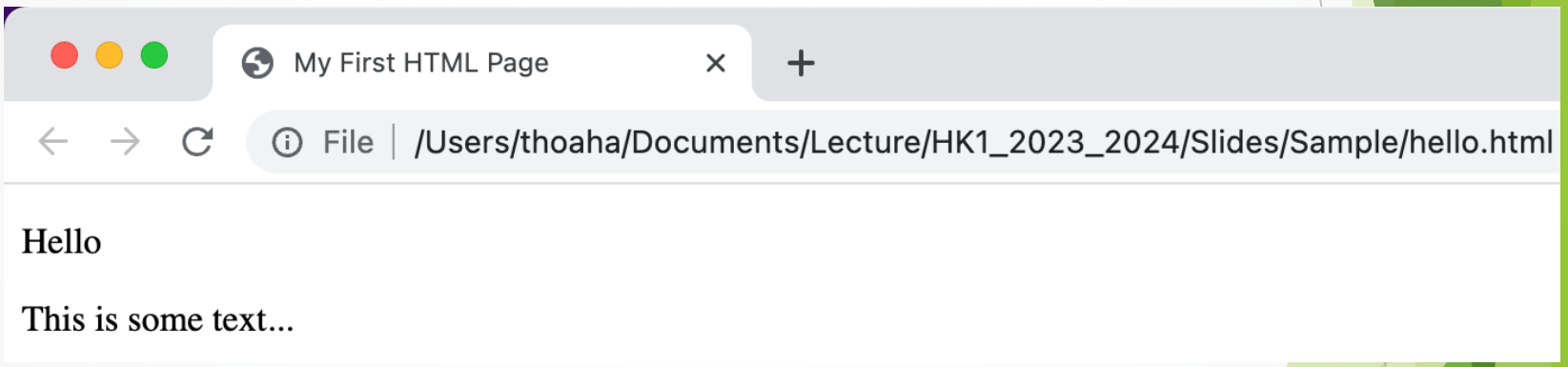
The <head> section

- Contains information that doesn't show directly on the viewable page
- Starts after the <!DOCTYPE> declaration
- Begins with <head> and ends with </head>
- Contains mandatory single <title> tag
- Can contain some other tags, e.g.
  - *<meta>*
  - *<script>*
  - *<style>*
  - *<!-- comments -->*

The <head> Section: <title> tag

- Title should be placed between <head> and </head> tags

```
<title>My First HTML Page </title>
```



- Used to specify a title in the window title bar
- Search engines and people rely on titles

The <head> section: <meta> tag

- Meta tags additionally describe the content contained within the page
- The name attribute specifies the name for the metadata.
- The name attribute specifies a name for the information/value of the content attribute.

```
<meta name="description" content="HTML tutorial" />
<meta name="keywords" content="html, web design, styles" />
<meta name="author" content="Chris Brewer" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
```

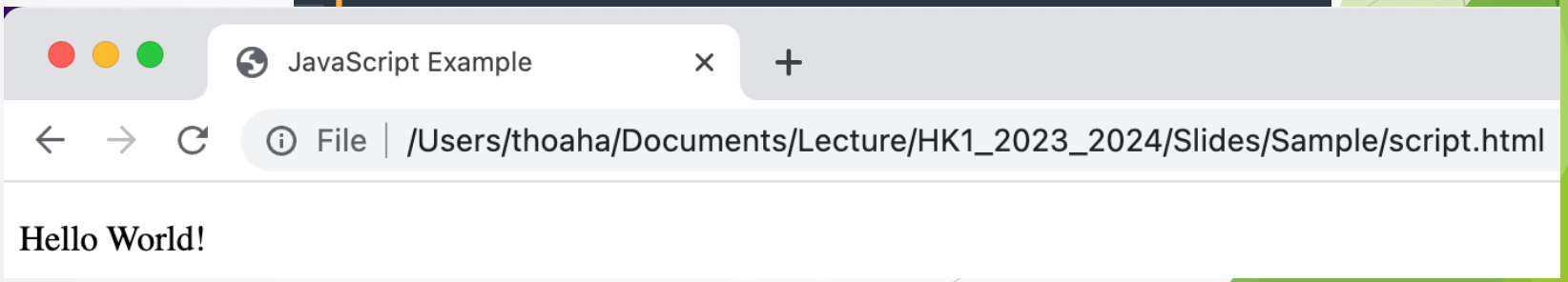
The <head> section: <script> tag

- The <script> element is used to embed scripts into an HTML document
  - Script are executed in the client's Web browser
  - Scripts can live in the <head> and in the <body> sections
- Supported client-side scripting languages:
  - JavaScript (it is not Java!)
  - VBScript
  - JScript

The <head> section: <script> tag

script.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>JavaScript Example</title>
    <script type="text/javascript">
      function sayHello() {
        document.write("<p>Hello World!<\/p>");
      }
    </script>
  </head>
  <body>
    <script type="text/javascript">
      sayHello();
    </script>
  </body>
</html>
```

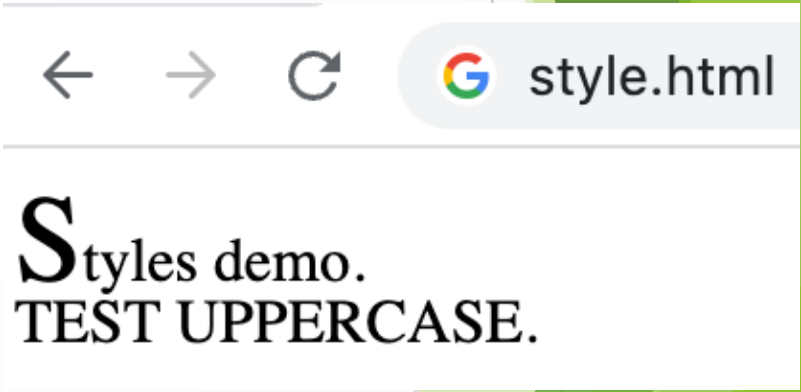


The <head> section: <style> tag

- The <style> element embeds formatting information (CSS styles) into an HTML page

style.html

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <style type="text/css">
5        p {
6          font-size: 12pt;
7          line-height: 12pt;
8        }
9        p:first-letter {
10         font-size: 200%;
11       }
12       span {
13         text-transform: uppercase;
14       }
15     </style>
16   </head>
17   <body>
18     <p>Styles demo.<br />
19       <span>Test uppercase</span>.
20     </p>
21   </body>
22 </html>
```



Comments: <!-- --> tag

- Comments can exist anywhere between the `<html></html>` tags
- Comments start with `<!--` and end with `-->`

```
<!-- Telerik Logo (a JPG file) -->
<img src="logo.jpg" alt="Telerik Logo">

<!-- Hyperlink to the web site -->
<a href="http://telerik.com/">Telerik</a>

<!-- Show the news table -->
<table class="newstable">
...
```

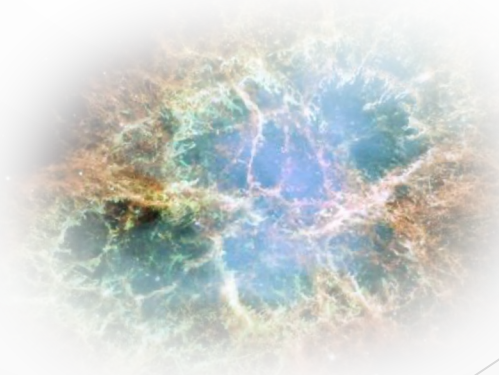
<body> section

- The <body> section describes the viewable portion of the page
- Starts after the <head> </head> section
- Begins with <body> and ends with </body>

```
<html>
  <head><title>Test page</title></head>
  <body>
    <!-- This is the Web page body -->
  </body>
</html>
```

HTML Basics

TEXT, IMAGES, TABLES, FORMS



Headings and Paragraphs

- Heading Tags: <h1></h1> … <h6></h6>

```
<h1>Heading 1</h1>
<h2>Sub heading 2</h2>
<h3>Sub heading 3</h3>
```

- Paragraph Tag:

```
<p>This is my first paragraph</p>
<p>This is my second paragraph</p>
```

- New line tag:

Text Formatting

- Text formatting tags modify the text between the opening tag and the closing tag

`<b></b>`	**bold**
`<i></i>`	*italicized*
`<u></u>`	underlined
`<sup></sup>`	Sample$^{\text{superscript}}$
`<sub></sub>`	Sample$_{\text{subscript}}$
`<strong></strong>`	**strong**
`<em></em>`	*emphasized*
`<pre></pre>`	`Preformatted text`
`<blockquote></blockquote>`	`Quoted text block`
`<del></del>`	Deleted text – ~~strike through~~

Text Formatting - Example

Notice

This is a *sample* Web page.

```
Next paragraph:
     preformatted.
```

More Info

Specifically, we're using XHMTL 1.0 transitional.
Next line.

Hyperlinks: <a> tag

- Link to a document called *form.html* on the same server in the same directory:

```
<a href="form.html">Fill Our Form</a>
```

- Link to a document called *parent.html* on the same server in the parent directory:

```
<a href="../parent.html">Parent</a>
```

- Link to a document called *cat.html* on the same server in the subdirectory stuff:

```
<a href="stuff/cat.html">Catalog</a>
```

Hyperlinks: <a> tag (cont.)

- Link to a document called *apply-now.html*
  - On the same server, in same directory
  - Using an image as a link button:

```
<a href="apply-now.html"><img
  src="apply-now-button.jpg" /></a>
```

- Link to a document called *index.html*
  - On the same server, in the subdirectory english of the parent directory:

```
<a
href="../english/index.html">Switch
to English version</a>
```

Hyperlinks - Example

Parent
Catalog
Google
Please contact here (by e-mail only)

```
<a href="../pa
<a href="stuff
<a href="https
<br />
<a href="mailt
<br />
<a href="apply
<a href="../er
```


```
>

by e-mail only)</a>

></a> <br />
/a> <br />
```

Switch to English version

Hyperlinks and sections

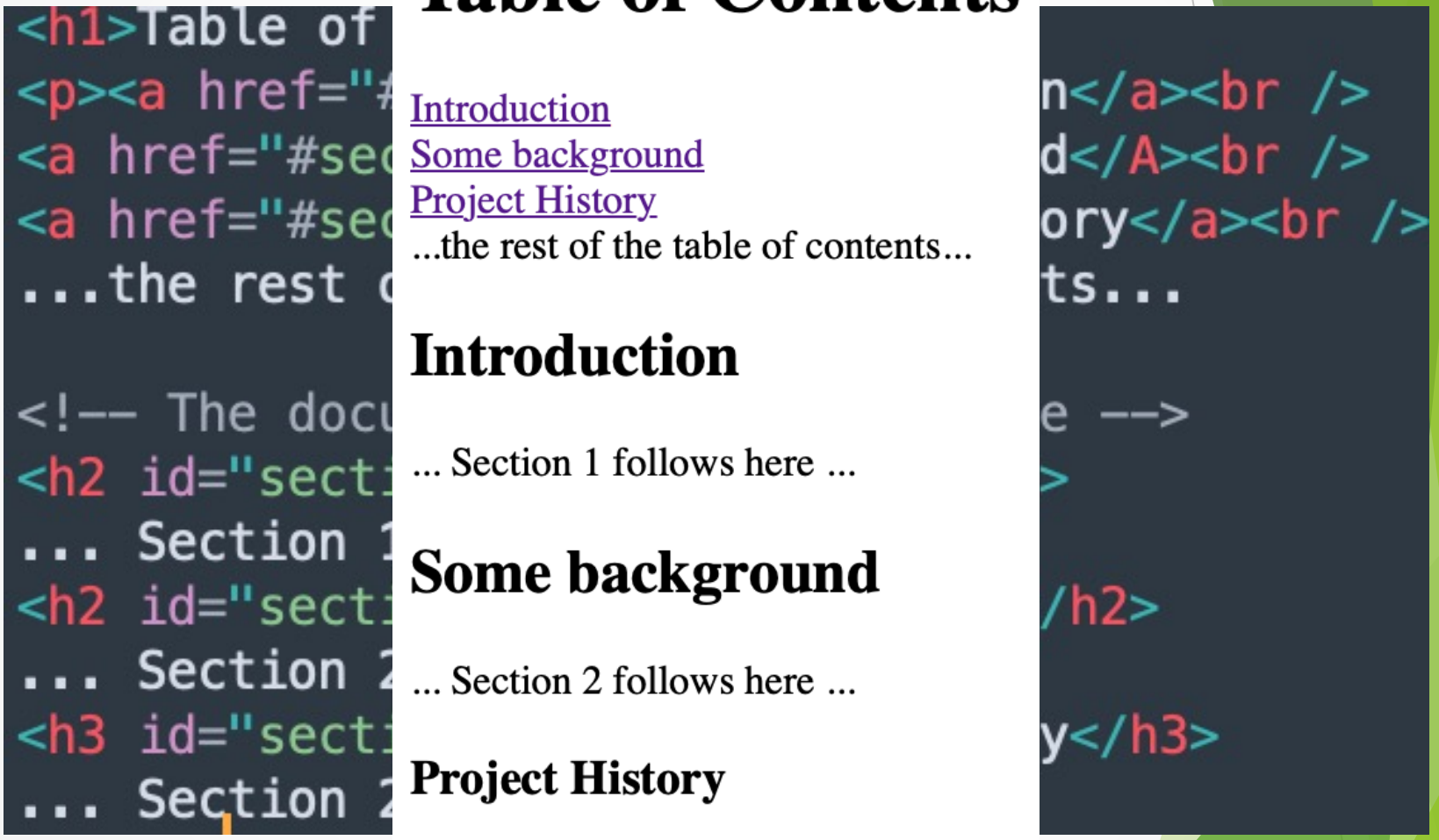
- Link to another location in the same document:

```
<a href="#section1">Go to Introduction</a>
...
<h2 id="section1">Introduction</h2>
```

- Link to a specific location in another document:

```
<a href="chapter3.html#section3.1.1">Go to
Section 3.1.1</a>
<!-- In chapter3.html -->
...
<div id="section3.1.1">
  <h3>3.1.1. Technical Background</h3>
</div>
```

Hyperlinks and sections: Example



Images: <img> tag

- Inserting an image with <img> tag:

```
<img src="apply-now-button.jpeg" alt="Apply now">
```

- Image attributes:

src	Location of image file (relative or absolute)
alt	Substitute text for display (e.g. in text mode)
height	Number of pixels of the height
width	Number of pixels of the width
border	Size of border, 0 for no border

Images map

- There are different areas that act as links in an image.
  - `map` element
  - `area` element

```
<map id="map_id">
 <area shape="{rect|circle|poly}"
 coords="x1,y1,r, x2,y2,…" href="…">
 …
</map>
```

Images map - Example

image_map.html

```
<!DOCTYPE html>
<html>
<body>

<h2>Image Maps</h2>
<p>Click on the computer, the phone, or the cup of coffee
to go to a new page and read more about the topic:</p>

<img src="workplace.jpg" alt="Workplace" usemap="#workmap" width="400" height="379">

<map name="workmap">
  <area shape="rect" coords="34,44,270,350" alt="Computer" href="computer.htm">
  <area shape="rect" coords="290,172,333,250" alt="Phone" href="phone.htm">
  <area shape="circle" coords="337,300,44" alt="Cup of coffee" href="coffee.htm">
</map>

</body>
</html>
```

Object element

- data: url of the resource
- width
- height
- name
- type: media type

Audio & Video

- Media Tags
  - `<audio>`
    - Attributes: autoplay, controls, loop, src
  - `<video>`
    - Attributes: autoplay, controls, loop, height, width, src

```
<audio width="360" height="240" controls loop>
  <source src="someSong.mp3" type="audio/mp3">
  </source>
  Audio tag is not supported
</audio>
```

Embed tag

- <embed>
  - Defines embedded content, such as a plug-in
  - Attributes: src="url", type="type"

```
<embed src="helloworld.swf" />
```

Miscellaneous tags

- `<hr>`: draws a horizontal rule (line)

```
<hr size="5" width="70%" />
```

- `<center></center>`: Deprecated!

```
<center>Hello World!</center>
```

- `<font></font>`: Deprecated!

```
<font size="3" color="blue">Font3</font>
<font size="+4"
color="blue">Font+4</font>
```

Miscellaneous tags - Example

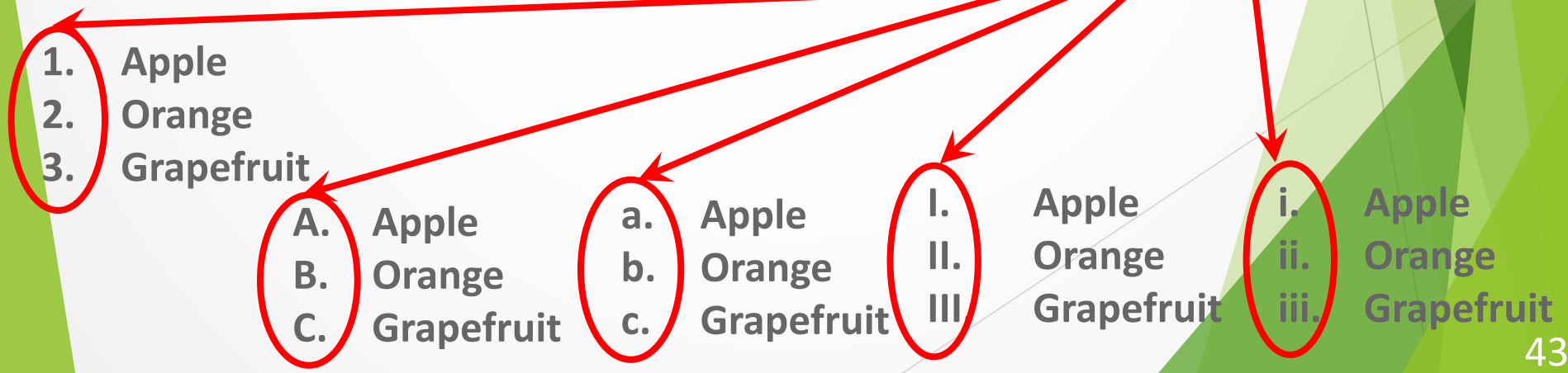
```
<html>
  <head>
  </body>
</html>
```

Ordered Lists: <ol> tag

- Create an **O**rdered **L**ist using :

```
<ol type="1">
  <li>Apple</li>
  <li>Orange</li>
  <li>Grapefruit</li>
</ol>
```

- Attribute values for `type` are `1`, `A`, `a`, `I`, or `i`



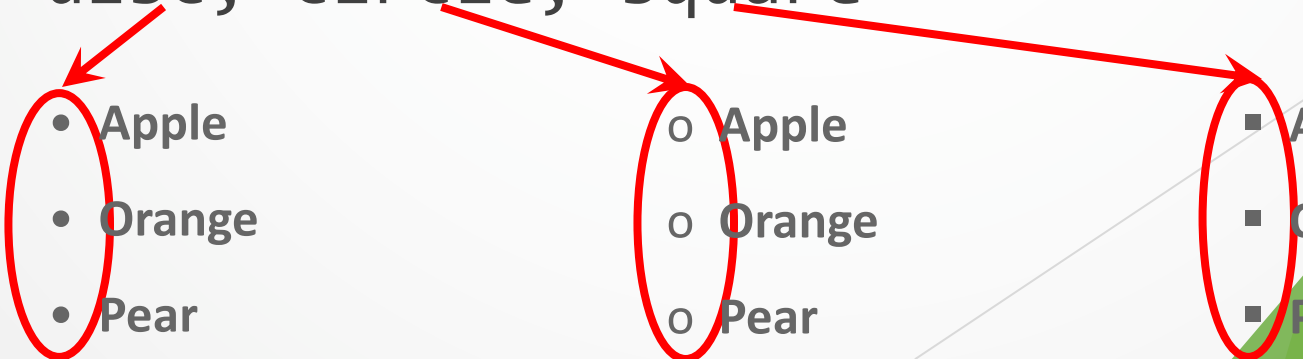
Unordered Lists: <ul> tag

- Create an **U**nordered **L**ist using :

```
<ul type="disc">
  <li>Apple</li>
  <li>Orange</li>
  <li>Grapefruit</li>
</ul>
```

- Attribute values for `type` are:

`disc, circle, square`



Definition Lists: <dl> tag

- Create definition lists using <dl>
  - Pairs of text and associated definition; text is in <dt> tag, definition in <dd> tag

```
<dl>
    <dt>HTML</dt>
    <dd>A markup language
    …</dd>
    <dt>CSS</dt>
    <dd>Language used to …</dd>
</dl>
```

HTML
A markup language …
CSS
Language used to …

  - Renders without bullets
  - Definition is indented

Exercise

I. Introduction HTML & Internet
- Intro
- Creating

II. HTML
a. Text
b. Paragraph
c. Font
d. Link
e. Image
f. Table
g. Form

III. CSS
1. Font
2. Text
3. Padding - Margin
4. Border

IV. Boostrap

V. jQuery

```
<ol type="I">
   <li>
   Introduce ...
      <ul>
      <li>Intro</li>
      <li>Creating</li>
       </ul>
   </li>
   <li>HTML</li  >
   <li>CSS</li>
   <li>Boostrap</li>
   <li>jQuery</li>
</ol>
```

HTML Special Characters

Symbol name	HTML Entity	Symbol
Copyright Sign	&copy;	©
Registered Trademark Sign	&reg;	®
Trademark Sign	&trade;	TM
Less Than	&lt;	<
Greater Than	&gt;	>
Ampersand	&amp;	&
Non-breaking space		
Em Dash	&mdash;	—
Quotation Mark	&quot;	"
Euro	&#8364;	€
British Pound	&pound;	£
Japanese Yen	&yen;	¥

HTML Special Characters - Example

```
<p>[&gt;&gt;  Welcome  &lt;&lt;]</p>
<p>&#9658;I prefer hard rock &#9835; music &#9835;</p>
<p>&copy; 2023 by ABC &amp; team</p>
```

[>> Welcome <<]

▶I prefer hard rock ♫ music ♫

© 2023 by ABC & team

Block and Inline Elements

- Block elements add a line break before and after them
  - is a block element
  - Other block elements are <table>, <hr>, headings, lists, <p> and etc.
- Inline elements don't break the text before and after them
  - is an inline element
  - Most HTML elements are inline, e.g. <a>

The tag

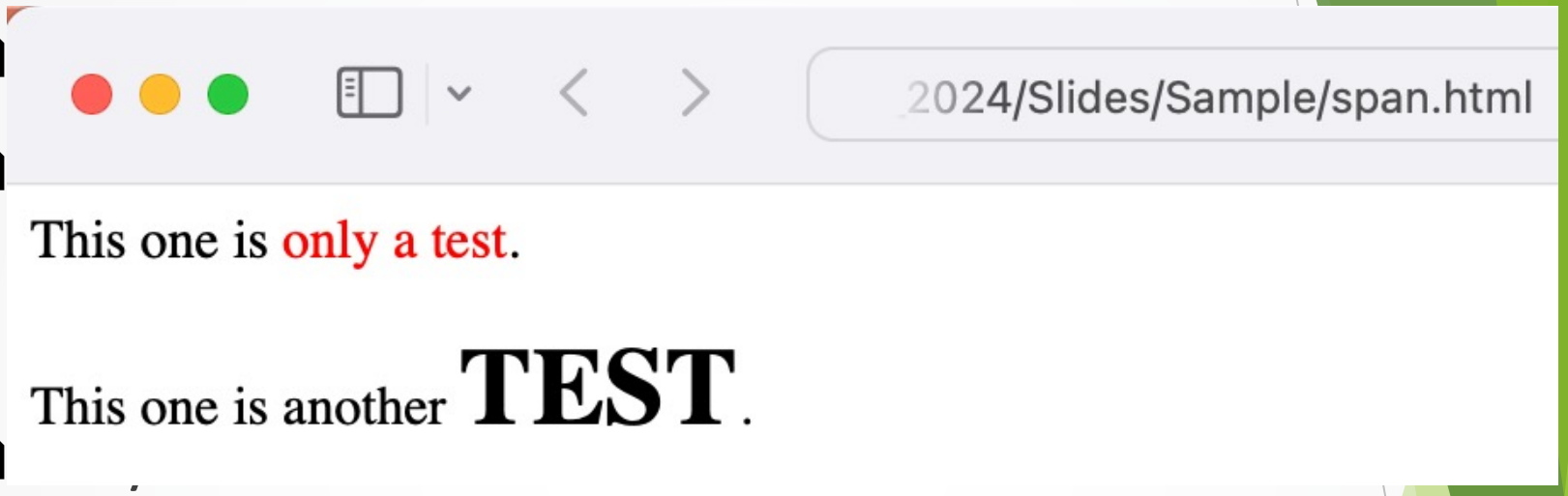
- creates logical divisions within a page
- Block style element
- Used with CSS

```
<div style="font-size:24px; color:red">DIV
example</div>
<p>This one is <span style="color:red; font-
weight:bold">only a
test</span>.</p>
```

DIV example

This one is **only a test**.

The tag




```
<p>This one is <span style="color:red; font-
weight:bold">only a test</span>.</p>
<p>This one is another <span style="font-
size:32px;
font-weight:bold">TEST</span>.</p>
```

DIV with the structure of a Web page

The “HTML 4 and Before” Way

- Using divs with IDs
  - The IDs are needed for styling

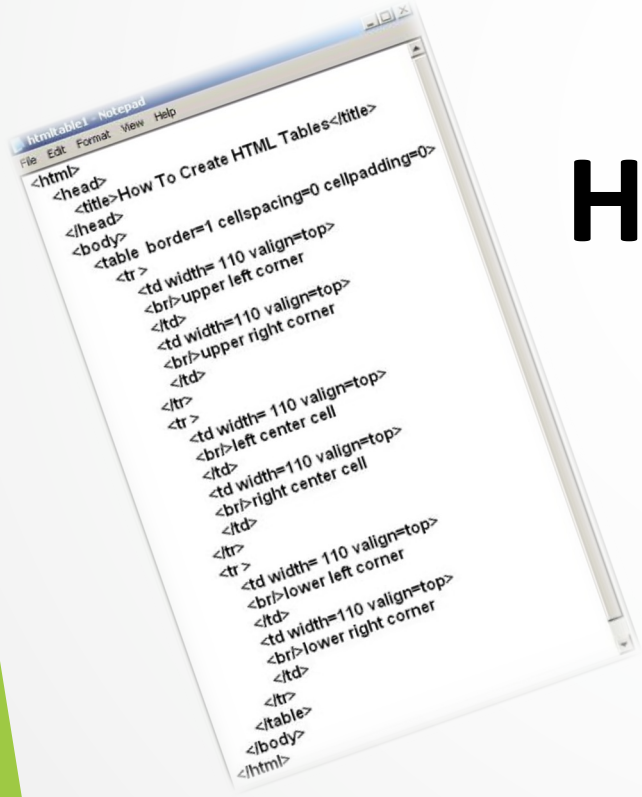
```
<html>
<head> … </head>
<body>
    <div id="header"> … </div>
    <div id="navigation"> … </div>
    <div id="sidebar"> … </div>
    <div id="content"> … </div>
    <div id="footer"> … </div>
</body>
</html>
```

The “HTML 5” Way

- In HTML 5 there are semantic tags for layout
  - <nav>, <header>, <footer>, <section>, <aside>, <article>

```
<html>
<head> … </head>
<body>
    <header> … </header>
    <nav> … </nav>
    <aside> … </aside>
    <section> … </section>
    <footer> … </footer>
</body>
</html>
```

US time	European date (D/M/Y) & time	Y-M-D date & time	Dollar	Chinese money	IP addresses	Names	Numbers
	29/10/1965	83-03-24		YMB 4	98.176.35.80		26.32 E +03
Fri Mar 22 21:48:49 UTC+0200 1957		1967-08-22 06:07:16 PM		YMB -81.38	162.117.253.34	dyse chidi	
Thu, 14 Feb 2002 04:24:20 UTC	06/07/99 06:46:01 AM	81-02-04 09:09:54 AM		YMB -108.83	122.205.50.6	bochai dychai	-191.45E-05
Monday, May 30, 1994 4:47:31 PM	06/09/05 05:11:16 AM			YMB 33.16		dydy balie	-131.20E+01
09/28/2000	24/11/1957		$-38.77	YMB 112.42	15.192.151.209		
		97-08-13 00:01:33 AM	$14.5	YMB -1.75	99.93.147.150	dychai tonchai	-187.28E-05
Mon, 29 Oct 1979 00:44:03 UTC		87-10-16	$14.66	YMB 61.14		chite malie	- 125.19 E -03
Sat, 9 Jan 1982 05:45:06 UTC	04/06/68	74-10-20	$20.47		121.169.225.22	dyma bama	138.11E+02
04/05/75		2000-03-20	$68.84	YMB 88.19	239.133.227.68	made liete	195.44 E +03
Monday, July 15, 2002 1:05:02 AM	01/02/1961 09:40:16 AM		$97.9	YMB 44.28	223.66.228.116	mava sete	-107
this is footer	row	number	ONE!	adsf	adsf	adsf	adsf



HTML Tables

Title	Title	Title	Title	Title	Title
Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data
Data	Data	Data	Data	Data	Data

HTML tables

- Tables represent tabular data
  - A table consists of one or several rows
  - Each row has one or more columns
- Tables comprised of several core tags:
  - table></table>: begin / end the table
  - <th></th>: a header cell in table
  - <tr></tr>: row in table
  - <td></td>: cell in table

HTML tables (cont.)

- <thead></thead>: groups the header content in a table
- <tbody></tbody>: groups the body content in a table
- <tfoot></tfoot>: groups the footer content in a table
- <colgroup> and <col> define columns (most often used to set column widths)

- Tables should not be used for layout. Use CSS floats and positioning styles instead

```
<table>
  <thead>
       <tr>
            <td> </td>
            <td> </td>
       </tr>
  </thead>
  <tbody>
       <tr>
            <td> </td>
            <td> </td>
       </tr>
  </tbody>
  <tfoot>
       <tr>
            <td> </td>
            <td> </td>
       </tr>
  </tfoot>
</table>
```

HTML tables (cont.)

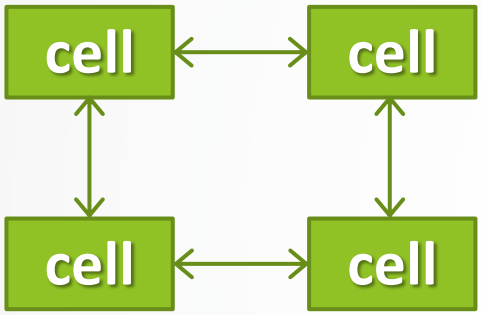
Thứ 2	Thứ 3	Thứ 4
Anh Văn	Văn	Toán
Anh Văn	Văn	Toán
Nghỉ		SHCN

Table/row/cell properties

- border
- width
- height
- align: horizontal [left, right]
- valign: vertical [top, bottom, middle]
- background-color
- Cell spacing: distance of cells
- Cell padding: distance from cells to data

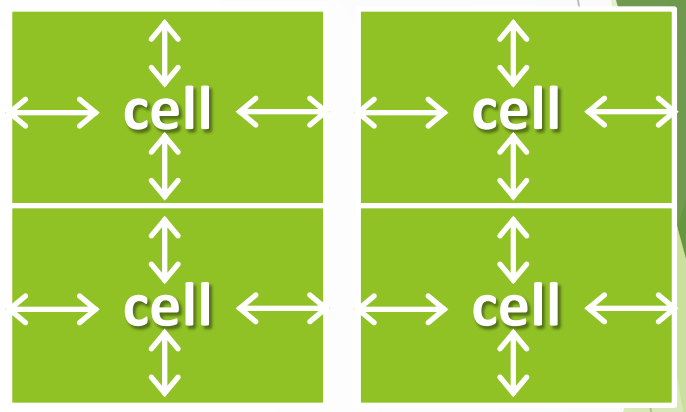
Cell spacing and padding: <td>

- cellspacing



- Defines the empty space between cells

- cellpadding

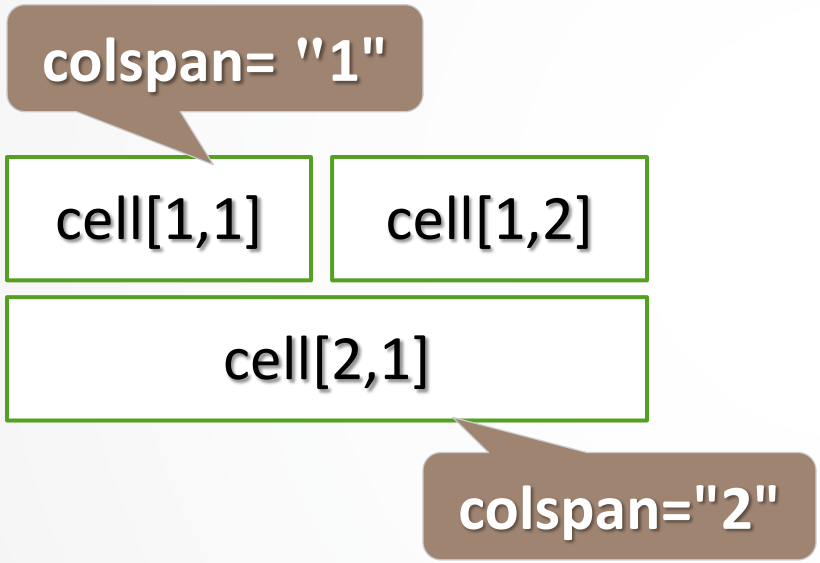


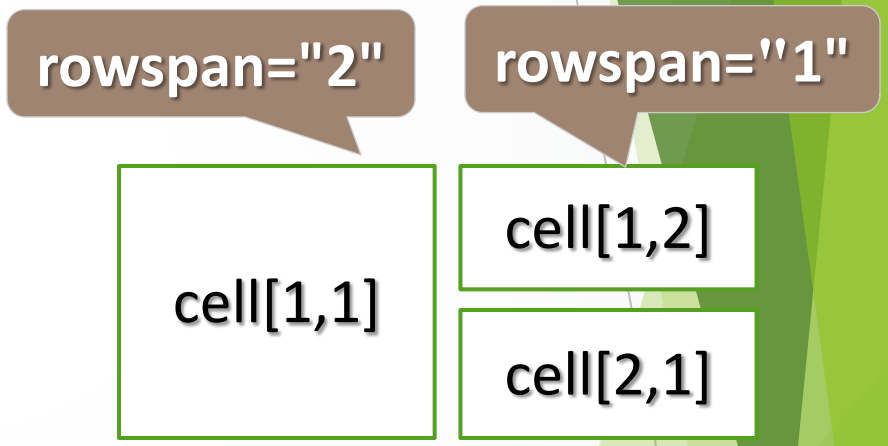
- Defines the empty space around the cell content

Column and row span

Tables have two another important attributes:

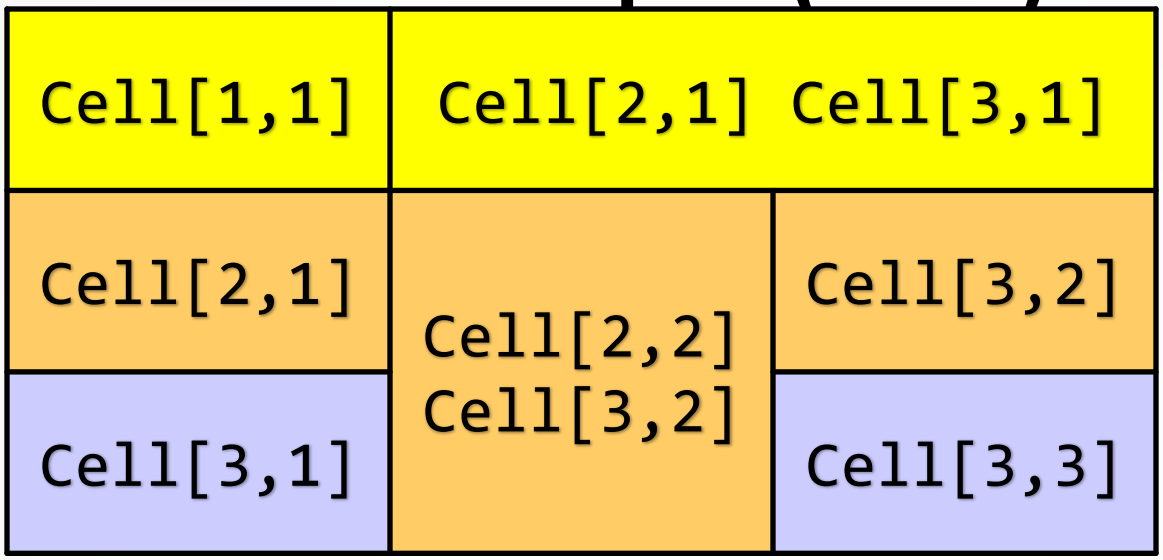
- colspan
- rowspan





- Defines how many columns the cell occupies
- Defines how many rows the cell occupies

Column and row span (cont.)




```
<table cellspacing="0">
    <tr class="1"><td>Cell[1,1]</td>
        <td colspan="2">Cell[2,1] Cell[3,1]</td></tr>
    <tr class="2"><td>Cell[2,1]</td>
        <td rowspan="2">Cell[2,2] <br> Cell[3,2]</td>
        <td>Cell[3,2]</td></tr>
    <tr class="3"><td>Cell[3,1]</td>
        <td>Cell[3,3]</td></tr>
</table>
```

HTML Forms

Entering User Data from a Web Page

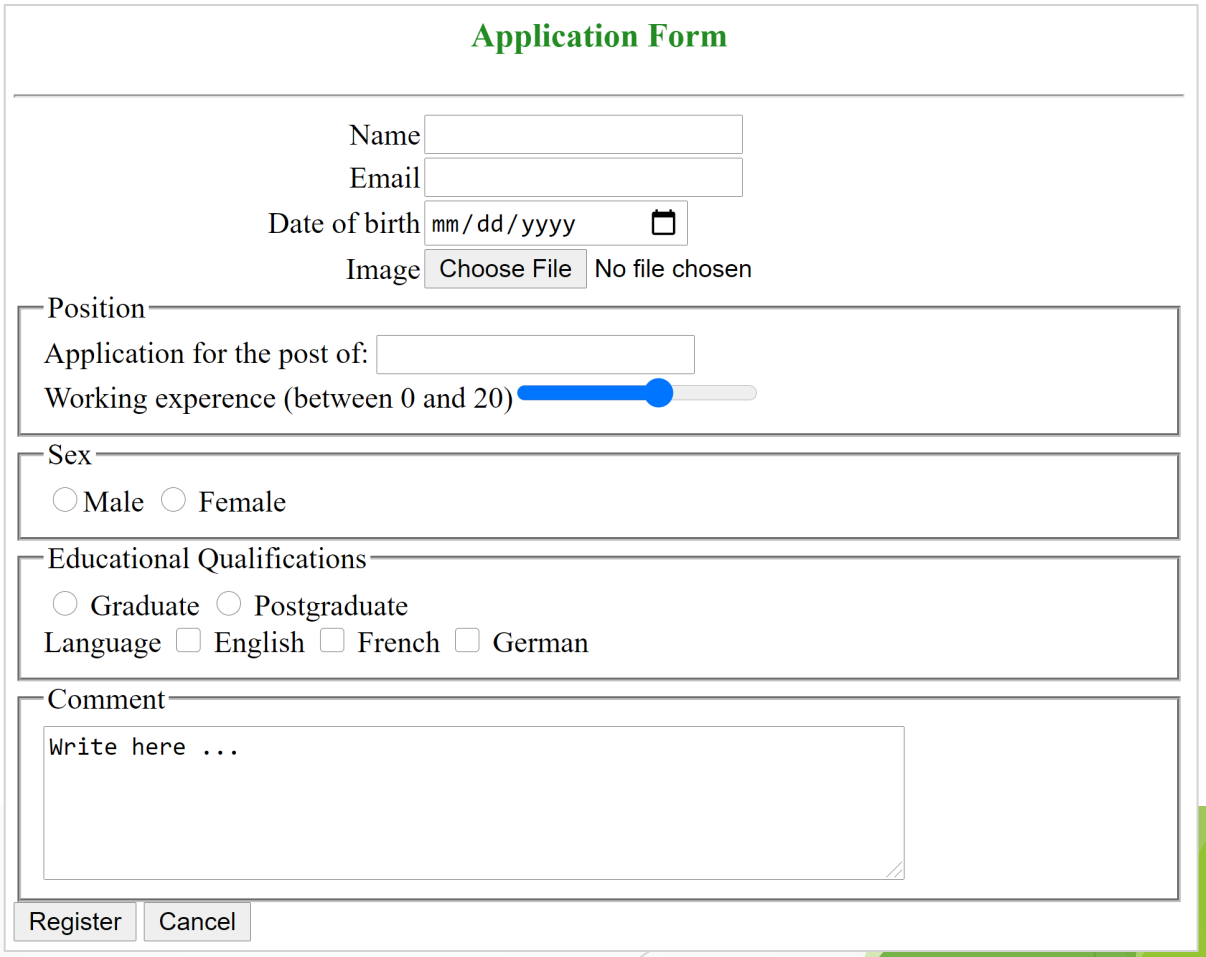
What are HTML Forms?

- The primary method for gathering data from site visitors
- HTML Forms can contains:
  - Text fields for the user to type
  - Buttons for interactions like "Register", "Login", "Search"
  - Menu, Slider, etc…
- Check Google
  - Google search field is a simple Text field

What are HTML Forms?

The HTML `<form>` element can contain one or more of the following form elements:

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



How to create form?

- Create a form block with `<form></form>`
- Example

```
<form name="myForm" method="POST"
action="path/to/some-script.php">
  ...
</form>
```

- The "action" attribute tells where the form data should be sent
- The "method" attribute tells how the form data should be sent – via GET or POST request

HTML Form Method

- The form data can be sent into a server side script in two methods: GET & POST
- POST
  - Data is sent in the HTTP message body of a POST request
  - For secure data: do not remain in the browser history
  - For large data: have no restrictions on data length

```
POST /register.php HTTP/1.1
Host: w3schools.com
Payload: {“name1”:”value1”,”name2”:”value2”}
```

HTML Form Method (cont.)

- GET
  - Data is as part of the request URL
  - Requests remain in the browser history
  - For non-secure data
  - Limitation in size of data
  - Is the default method

```
/register.php?name1=value1&name2=value2
```

Text fields

- Single-line text input field

```
<input type="text" name="firstName"
value="This is a text field" size=25/>
```

- Multi-line text input fields (textarea):

```
<textarea name="comments" cols=40
rows=10>This is a multi-line text
field</textarea>
```

- Password input – a text field which masks the entered text with * signs

```
<input type="password" name="pass" />
```

Buttons

- Reset button – brings the form to its initial state

```
<input type="reset" name="resetBtn"
value="Reset the form" />
```

- Submit button:

```
<input type="submit" value="Apply Now" />
```

- Image button – acts like submit but image is displayed and click coordinates are sent

```
<input type="image" src="submit.gif"
name="submitBtn" alt="Submit" />
```

- Ordinary button – no default action, used with JS

```
<input type="button" value="click me" />
```

Checkboxes and Radio buttons

- Checkboxes:

```
<input type="checkbox" name="fruit"
value="apple" />
```

- Radio buttons:

```
<input type="radio" name="title"
value="Mr." />
```

- Radio buttons can be grouped, allowing only one to be selected from a group:

```
<input type="radio" name="city" value="Lom"
/>
<input type="radio" name="city"
value="Ruse" />
```

Hidden fields

- Hidden fields contain invisible data

```
<input type="hidden" name="account" value="This
is a hidden text field" />
```

  - Not shown to the user

File input

- File input – a field used for uploading files

```
<input type="file" name="photo" />
```

  - When used, it requires the form element to have a specific attribute: enctype attribute specifies how the form-data should be encoded when submitting it to the server.

```
<form enctype="multipart/form-data"
method="post">
...
  <input type="file" name="photo" />
...
</form>
```

Select fields

- Dropdown:

```
<select name="gender">
  <option value="Value 1"
    selected="selected">Male</option>
  <option value="Value 2">Female</option>
  <option value="Value 3">Other</option>
</select>
```

- Multiple choice:

```
<select name="products" multiple="multiple">
  <option value="Value 1"
    selected="selected">keyboard</option>
  <option value="Value 2">mouse</option>
</select>
```

Labels

- Labels are used to associate an explanatory text to a form field using the field's ID.

```
<label for=”firstName">First Name</label>
<input type="text" id="firstName" />
```

- Clicking on a label focuses its associated field (checkboxes are toggled, radio buttons are checked)
- Labels are both a usability and accessibility feature and are required in order to pass accessibility validation.

Fieldsets

- Fieldsets are used to enclose a group of related form fields:

```
<form method=”POST" action="form.aspx">
   <fieldset>
      <legend>Client Details</legend>
      <input type="text" id=”name" />
      <input type="text" id=”phone" />
   </fieldset>
   <fieldset>
      <legend>Order Details</legend>
      <input type="text" id=”quantity" />
      <textarea cols="40" rows="10"
         id=”remarks"></textarea>
   </fieldset>
</form>
```

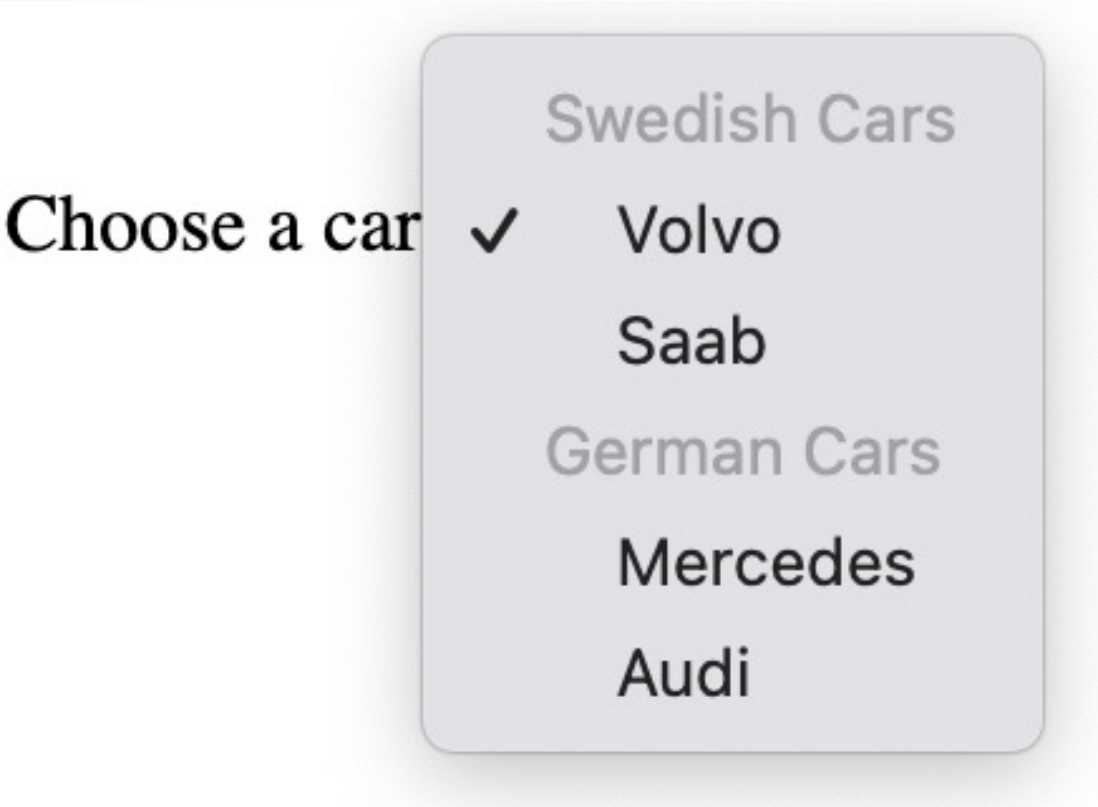
- The <legend> tag is used to set the fieldset's title.

optgroup

- The <optgroup> tag is used to group related options in a <select> element (drop-down list).

```
<label for="cars">Choose a car:</label>
<select name="cars" id="cars">
  <optgroup label="Swedish Cars">
    <option value="volvo">Volvo</option>
    <option value="saab">Saab</option>
  </optgroup>
  <optgroup label="German Cars">
    <option value="mercedes">Mercedes</option>
    <option value="audi">Audi</option>
  </optgroup>
</select>
```

optgroup - Example



FORM SAMPLE

Academic information

Degree Master of Business Administration

Student ID

Classes attended
- Geography
- English
- Mathematics

Personal Details

First Name

Last Name

Gender: Male Female

Email

TERMS AND CONDITIONS...

Send Clear

Attributes from HTML5

- To display an HTML page correctly, a web browser must know the character set used in the page
  - This is specified in the <meta> tag:

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

- Autocomplete
  - The browser stores the previously typed values
  - Brings them back on a later visit on the same page
- Autofocus
  - The field becomes on focus on page load
- Required
  - The field is required to be filled/selected

HTML5 Form: New Input Controls

- search
- email
- url
- tel
- number
- range
- date
- month
- week
- time
- datetime
- datetime-local
- color

HTML5 Form: New Input Controls (cont.)

- Input type: search

```
# In HTML4.1
 <input type="text" name=”search"/>

# In HTML5
 <input type=”search" name=”search"/>
```

- Input type: email

```
# In HTML4.1
 <input type="text" name=”email"/>

# In HTML5
 <input type=”email" name=”email"/>
```

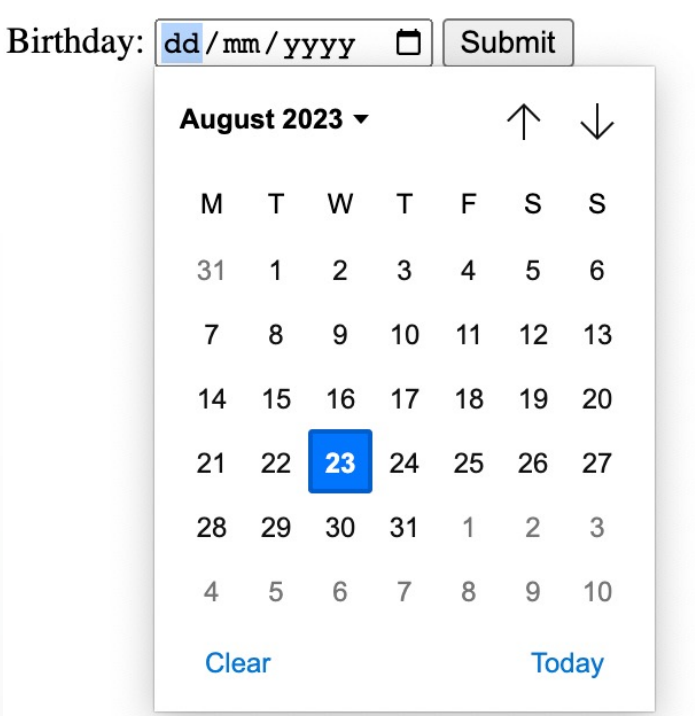
HTML5 Form: New Input Controls (cont.)

- Input type: tel differs from email and url in that not particular syntax is enforced

```
<input type="tel" id="phone" name="phone"
placeholder="123-45-678" pattern="[0-9]{3}-[0-
9]{2}-[0-9]{3}">
```

- Input type: date

```
<input
type="date"
id="birthday"
name="birthday">
```



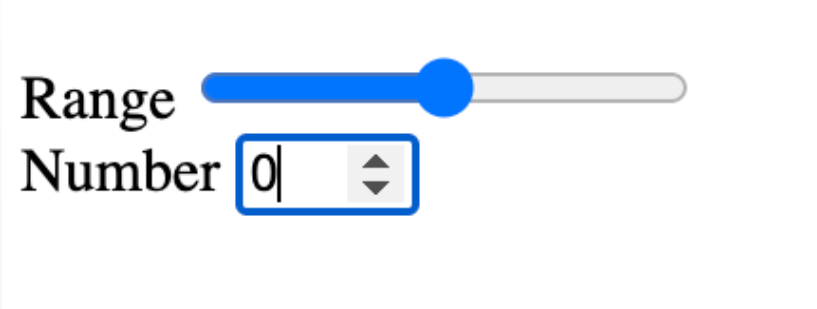
HTML5 Form: New Input Controls (cont.)

- Restricts users to enter only numbers
  - Additional attributes min, max and step and value
  - Can become Spinbox or Slider, depending on the input type

```
<input type="range" min="0" max="100" />
<input type="number" min="0" max="100" />
```

  - Have some differences on different browsers
  - Sliders and Spinboxes do not work on Firefox

    Shown as regular textboxes



Input fields with validation

- Email – provides a simple validation for email
  - Can be passed a pattern for validation
  - On a mobile device brings the email keyboard

```
<input type="email" required="true"
pattern="[^ @]*@[^ @].[^ @]"/>
```

- URL – has validation for url
  - On a mobile device brings the url keyboard

```
<input type="url" required="true" />
```

- Telephone
  - Brings the numbers keyboard

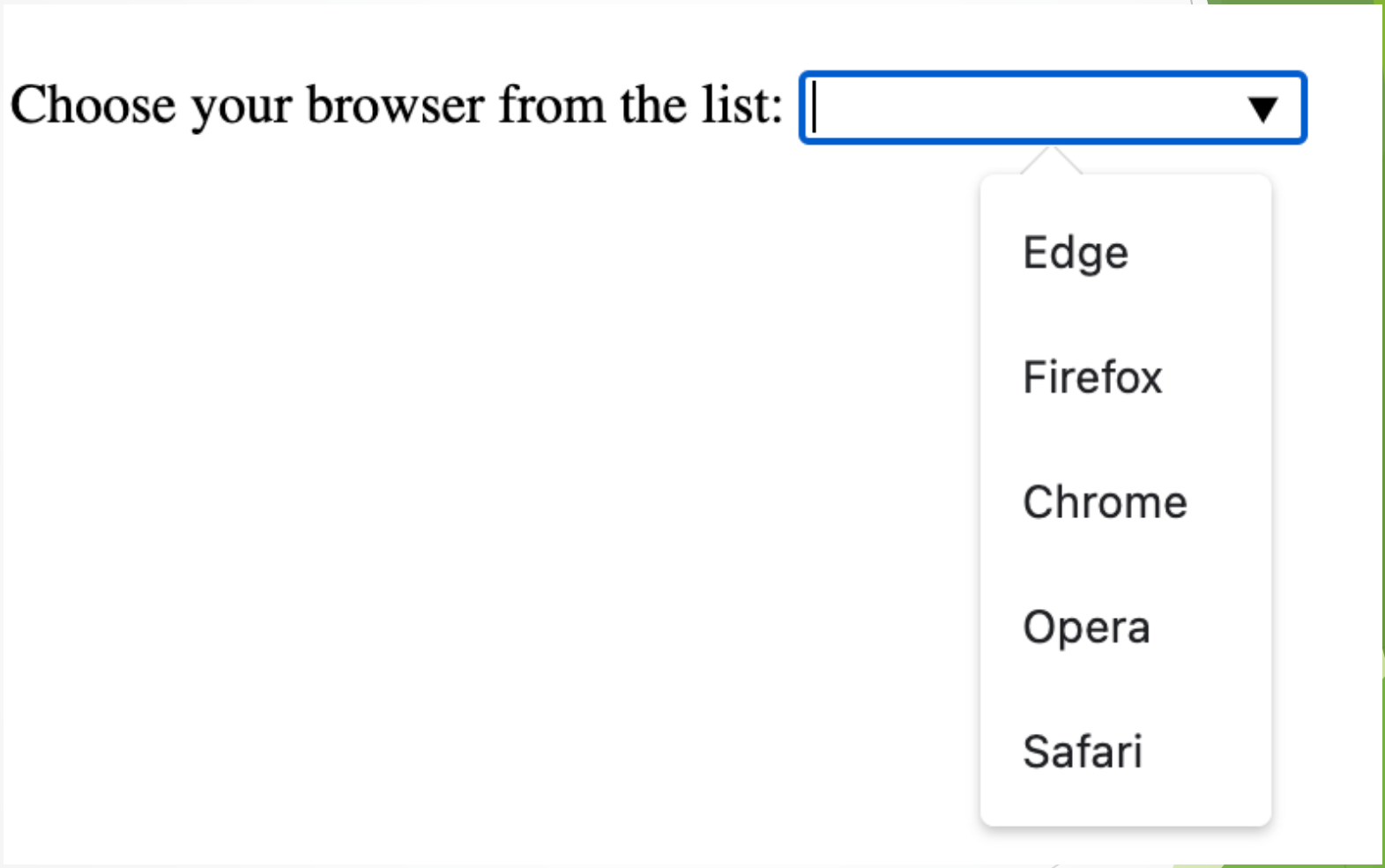
```
<input type="tel" required="true" />
```

Datalist

- The <datalist> tag specifies a list of pre-defined options for an <input> element.
- The <datalist> tag is used to provide an "autocomplete" feature for <input> elements. Users will see a drop-down list of pre-defined options as they input data.
- The <datalist> element's id attribute must be equal to the <input> element's list attribute (this binds them together).

```
<label for="browser">Choose your browser from the list:</label>
<input list="browsers" name="browser" id="browser">
<datalist id="browsers">
  <option value="Edge">
  <option value="Firefox">
  <option value="Chrome">
  <option value="Opera">
  <option value="Safari">
</datalist>
```

Datalist: Example



output

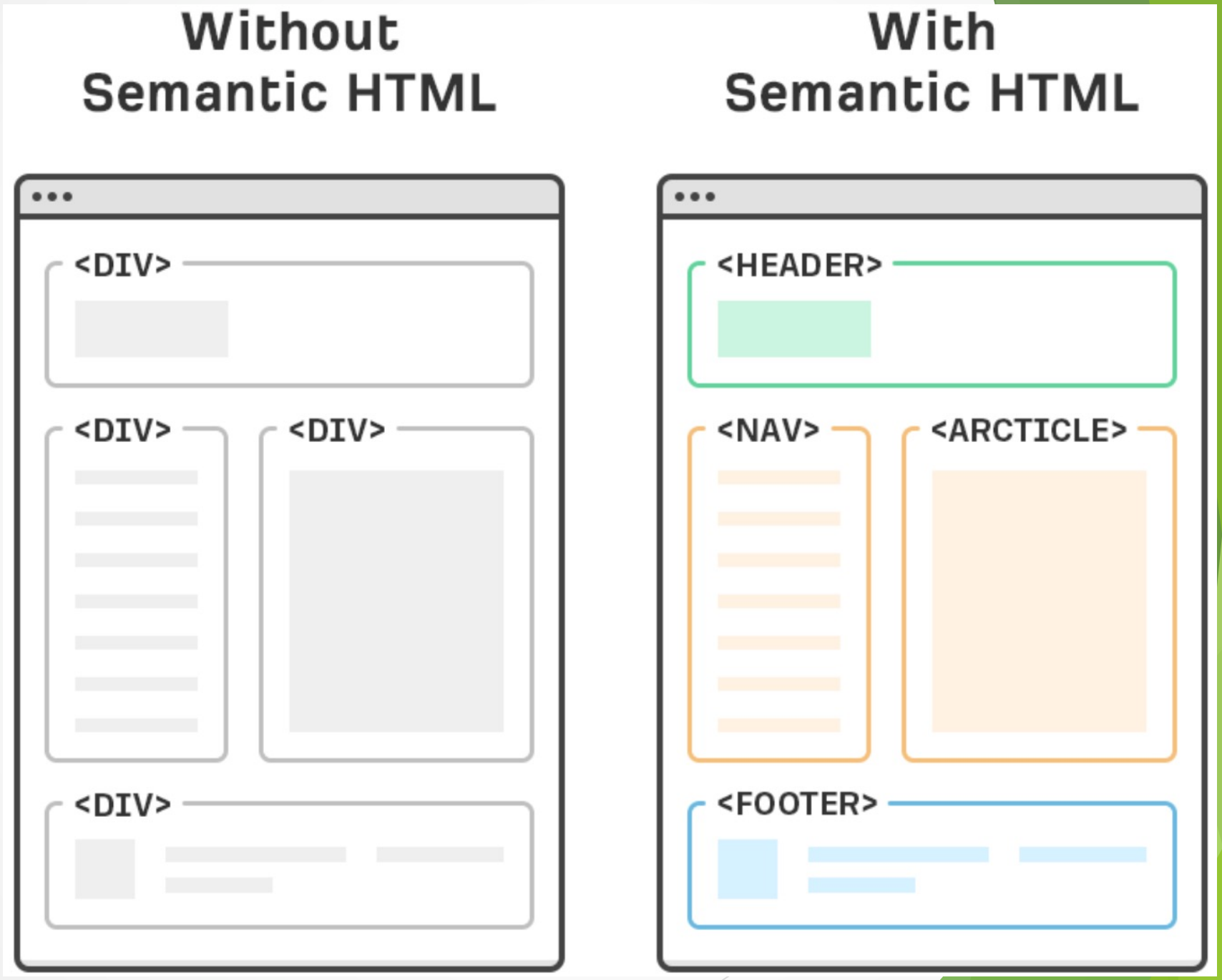
- The <output> tag is used to represent the result of a calculation (like one performed by a script).

```
<form oninput="x.value=parseInt(a.value)
+parseInt(b.value)">
  <input type="range" id="a" value="50">
  +<input type="number" id="b" value="25
">
  =<output name="x" for="a b"></output>
</form>
```

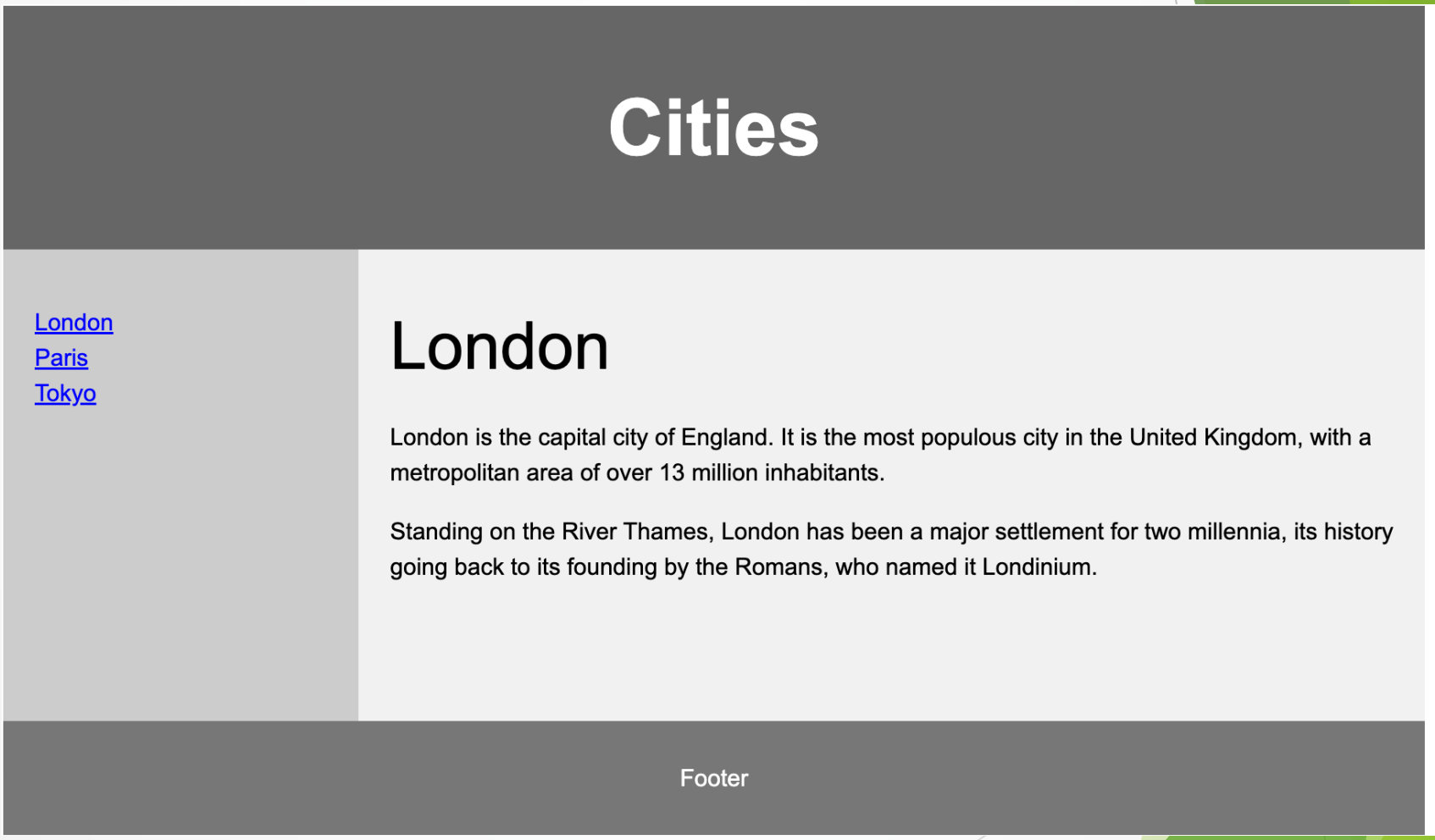
Tab-index

- The tabindex HTML attribute controls the order in which form fields and hyperlinks are focused when repeatedly pressing the TAB key
  - tabindex="0" (zero) - "natural" order
  - If X < Y, then elements with tabindex="X" are iterated before elements with tabindex="Y"
  - Elements with negative tabindex are skipped, however, this is not defined in the standard

```
<input type="text" tabindex="10" />
```


Without
Semantic HTML
<DIV>
<DIV>
<DIV>
<DIV>
With
Semantic HTML
<HEADER>
<NAV>
<ARCTICLE>
<FOOTER>

Webpage layout – Example



Homework

1. Làm bài tập **Homework 1 -> Homework 4**

2. Nộp bài:

    2.1 Link: https://drive.google.com/drive/folders/1ry4nUMtMzl6Wqn9rMh6hCogiW6llP-gZ

    2.2 Thời hạn: **27/8/2023 -> 5/9/2023**

    2.3 Tên Folder: **MSSV_HOTEN**

Homework 1

Create Web Pages like the following using tables:

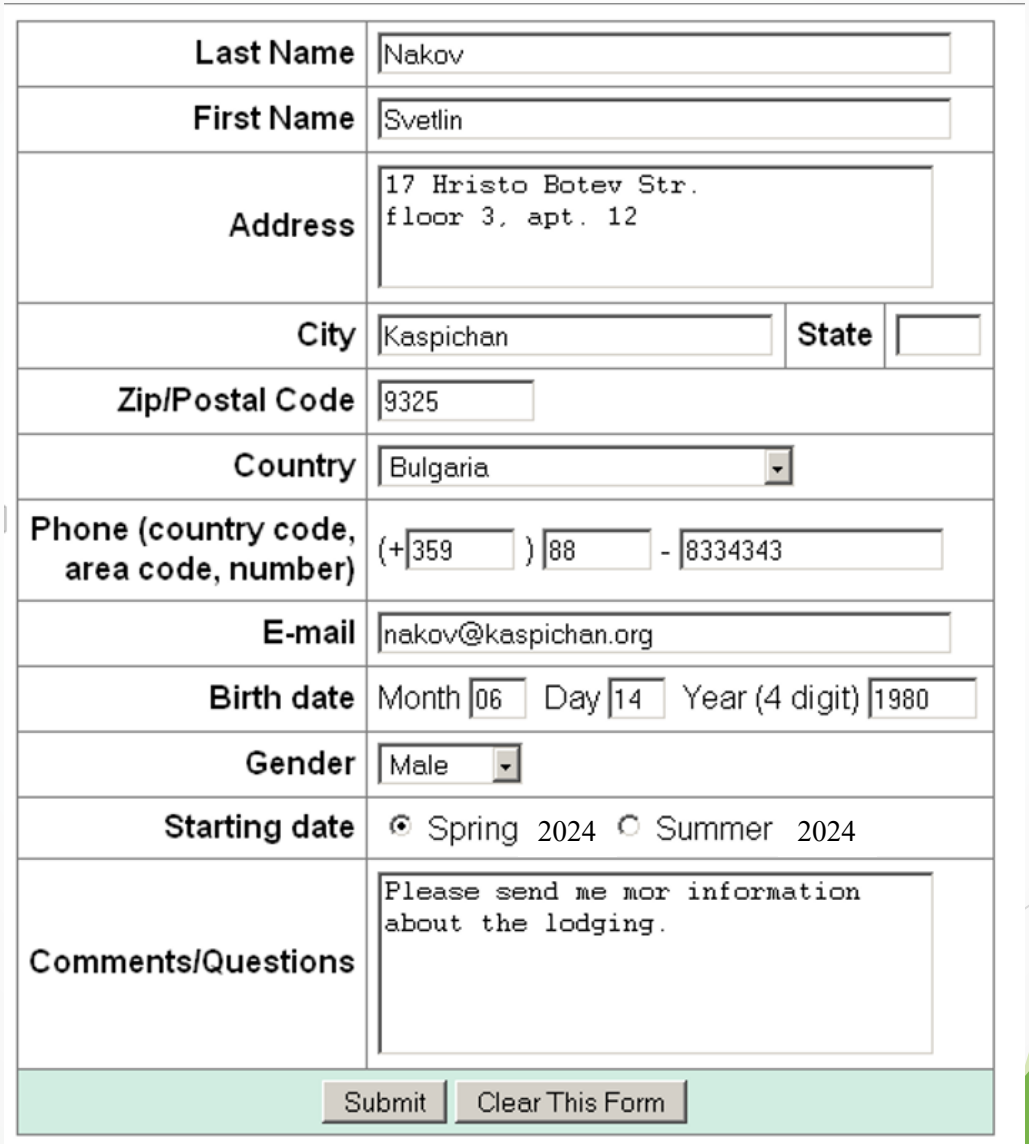
Fresh Fruits

It has long been known that a diet that includes at least a few servings of fresh fruit every day will help keep you healthy, fit and trim.

<table>
<tr><td colspan="3">Title goes here</td><td>A</td><td>B</td></tr>
<tr><td rowspan="3">C</td><td>D</td><td>E</td><td>F</td><td>G</td></tr>
<tr><td>H</td><td colspan="2">I</td><td rowspan="2">J</td></tr>
<tr><td>K</td><td>L</td><td>M</td></tr>
<tr><td>N</td><td colspan="4">O</td></tr>
</table>

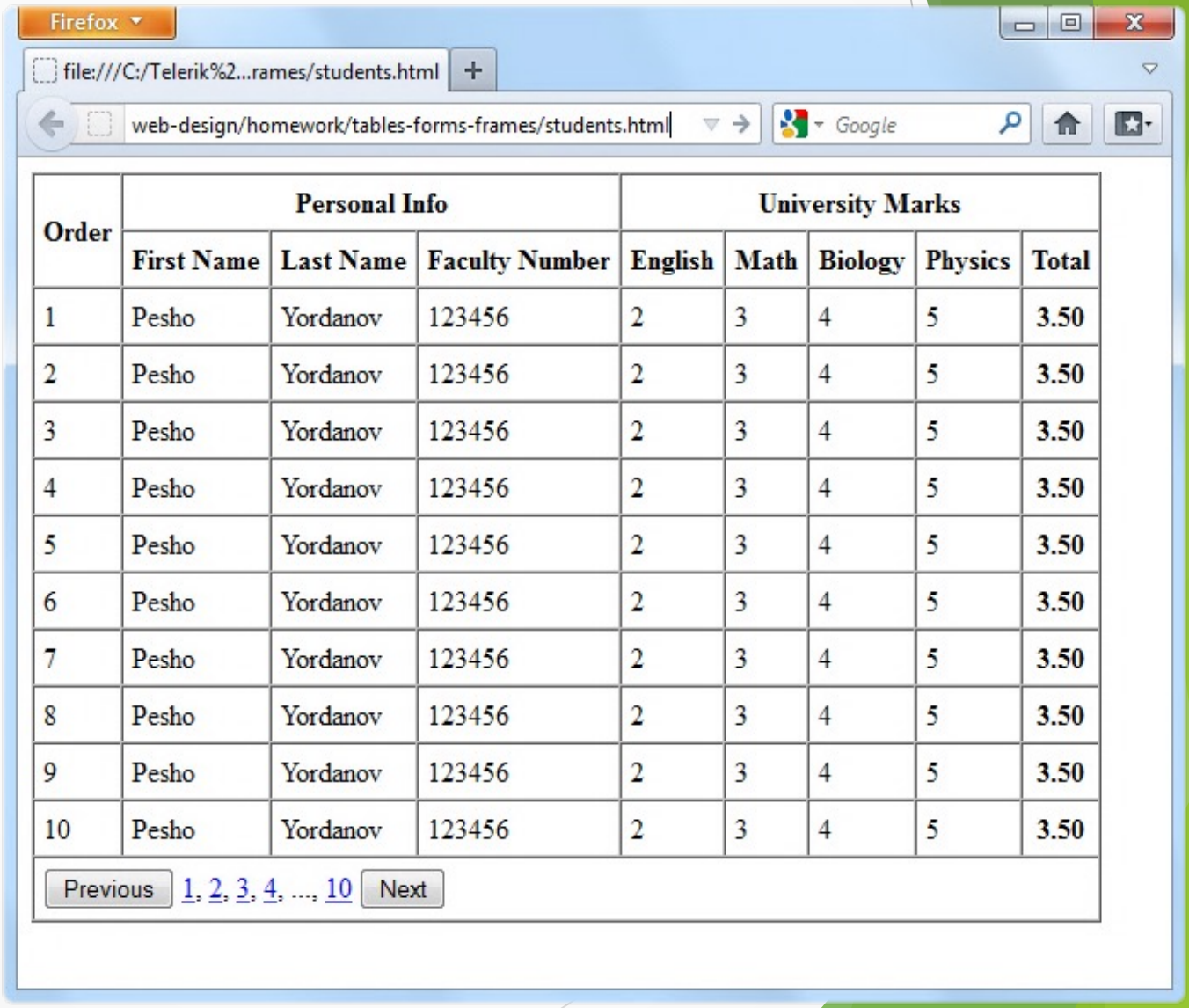
Homework 2

Create the following using form:



Homework 3

Create the following using table:



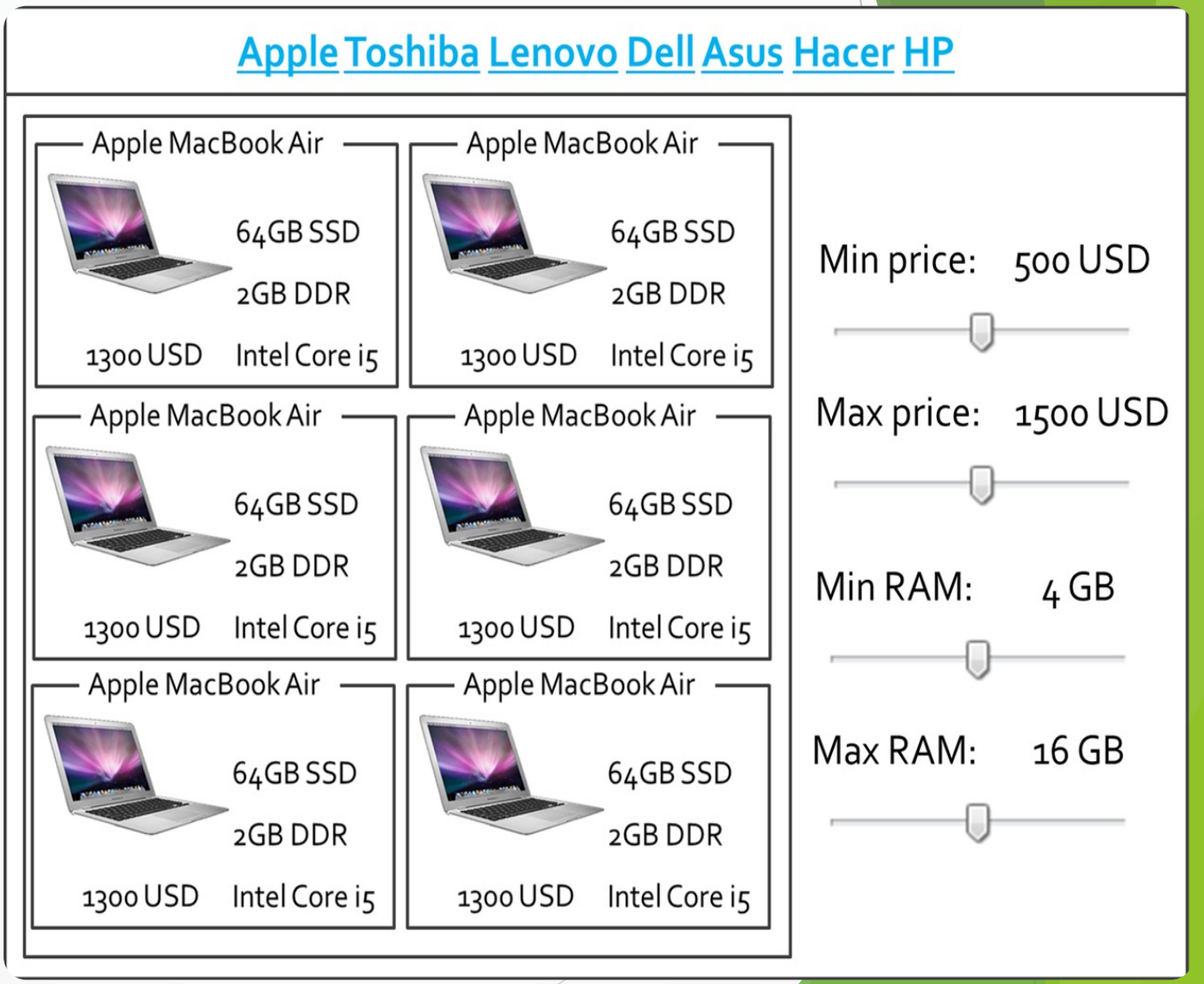
Order	Personal Info			University Marks				
	First Name	Last Name	Faculty Number	English	Math	Biology	Physics	Total
1	Pesho	Yordanov	123456	2	3	4	5	**3.50**
2	Pesho	Yordanov	123456	2	3	4	5	**3.50**
3	Pesho	Yordanov	123456	2	3	4	5	**3.50**
4	Pesho	Yordanov	123456	2	3	4	5	**3.50**
5	Pesho	Yordanov	123456	2	3	4	5	**3.50**
6	Pesho	Yordanov	123456	2	3	4	5	**3.50**
7	Pesho	Yordanov	123456	2	3	4	5	**3.50**
8	Pesho	Yordanov	123456	2	3	4	5	**3.50**
9	Pesho	Yordanov	123456	2	3	4	5	**3.50**
10	Pesho	Yordanov	123456	2	3	4	5	**3.50**

Previous 1, 2, 3, 4, ..., 10 Next

Homework 4

Create the following HTML 5 Page

Hint: Use Fieldsets and Nested tables



Q&A