CST463 - WEB PROGRAMMING

Module 1: WWW & HTML

- Introduction to the Internet & WWW: Evolution of Internet & World Wide Web- Web Basics, URI’s & URL-MIME.

- Introduction to HTML5: Structuring & editing an HTML5 document, Fundamentals of HTML - Headings-Hyper Links- Images - Special Characters & Horizontal Rules-Lists- Tables -Forms - Internal Linking- Meta Elements-HTML5 Form input types -Input and Data List Elements and autocomplete attribute- Page Structure Elements -Multimedia-HTML5 Audio & video elements..

Evolution of the Internet and World Wide Web

Evolution of the Internet

- ARPAnet
  - late 1960s and early 1970s
  - US DoD developed large scale network,for communication
  - Network reliability ,robust network
  - DoD’s ARPA funded for the first project and named ARPAnet
  - ARPA(Advanced Research Project Agency)
  - Available to laboratories &universities conducted ARPA-funded research
- BITnet(Because It’s Time N/W), CSnet(Computer Science Network)
  - late 1970s & early 1980s
  - BITnet by City University of New York
  - email and file transfer for other institutions
  - CSnet by University of Delaware
  - not used as National Network

Evolution of the Internet

- Packet Switching
- One of the primary goals for ARPANET was to allow multiple users to send and receive information simultaneously over the same communications paths (e.g., phone lines).
- Then network operated with a technique called packet switching, in which digital data was sent in small bundles called packets.
- The packets contained address, error-control and sequencing information. The address information allowed packets to be routed to their destinations.
- The sequencing information helped in reassembling the packets.
- Packets from different senders were intermixed on the same lines to efficiently use the available bandwidth.
- This packet-switching technique greatly reduced transmission costs, as compared with the cost of dedicated communications lines.
- The network was designed to operate without centralized control. If a portion of the network failed, the remaining working portions would still route packets from senders to receivers over alternative paths for reliability.

Evolution of the Internet

- NSFnet(National Science Foundation) - 1986
  - Originally for non-DOD funded places
  - Initially connected five supercomputer centers
  - By 1990, it had replaced ARPAnet for non-military uses
  - Soon became the network for all (by the early 1990s)
- By 1992NSFnet connected more than 1 million computers , around the world
  - Small part of NSFnet returned to research network
  - NSFnet eventually became known as the Internet by 1995
  - Internet -A world-wide network of computer networks
- N/w of N/W's and different devices in the network communicates with each other based on the low level protocol TCP/IP(Transport layer).

Evolution of the Internet-TCP/IP

- TCP/IP(Transport layer)-TCP/IP standard allows a program on one computer to communicate with a program on another computer via internet
- Not every computer on internet directly connects to every other computer
- Individual computers in an organization can be connected each other in a local network.
- Internet Protocol (IP) Addresses -Every node has a unique numeric address
  - Form : IPv4 : 32-bit binary number
  - Four 8 bit numbers separated by periods x.x.x.x ranges 0 to 256
  - New standard, IPv6, has 128 bits (1998)
  - An IPv6 address is represented as eight groups of four hexadecimal digits, each group representing 16 bits . The groups are separated by colons (:). The hexadecimal digits are case-insensitive,
  - An example of an IPv6 address is:2001:0db8:85a3:0000:0000:8a2e:0370:7334

Evolution of the Internet

- IPv4 - IP address as a 32-bit value
- IPv6 addresses have a size of 128 bits.
- IPv6 has a vastly enlarged address space compared to IPv4.
- IP consists of two parts: N/W ID & Host ID
- Organization are assigned to machine with block of IP
- It will be assigned to machines that need internet
- ex)small organization(256 IP address)

  191.57.126.0 to 191.57.126.256
- Large organization(16 million IP)

  12.0.0.0 to 12.255.255.255

Evolution of the Internet-Domain names

- It is difficult to remember number to identify machines on net.so domains are used to identify the machine(in addition to IP) in a network.
- Domain begin with host name followed by one or more domain names

  Form: host-name.domain-names

  Ex) www.cars.maruthi.org

  - First domain is the smallest; last is the largest
  - Last domain specifies the type of organization
  - Fully qualified domain name - the host name and all of the domain names
- Domain names must be converted to IP address before the message can be transmitted over the internet to the destination. It is done using DNS.
- DNS servers - convert fully qualified domain names to IP's

Evolution of the Internet-World Wide Web, HTML, HTTP

- World Wide Web WWW allows computer users to locate and view multimedia based documents over internet In 1989,
- Tim Berners-Lee of CERN (the European Organization for Nuclear Research) began to develop a technology for sharing information via hyperlinked text documents named HyperText Markup Language (HTML).
- Berners-Lee proposed a new protocol for internet ,a system for document access .
  - Document form: Hypertext--- Pages? Documents? Resources?
  - Hypermedia – more than just text – images, sound, etc.
- He wrote the Hypertext Transfer Protocol (HTTP)—a communications protocol used to send information over the web.
- The URL (Uniform Resource Locator) specifies the address (i.e.,location) of the web page displayed in the browser window.
- Each web page on the Internet is associated with a unique URL. URLs usually begin with http://.

Evolution of the Internet-HTTPs

- URLs of websites that handle private information, such as credit card numbers, often begin with https://, (Hypertext Transfer Protocol Secure -HTTPS).
- HTTPS is the standard for transferring encrypted data on the web.
- It combines HTTP with the Secure Sockets Layer (SSL) and the more recent Transport Layer Security (TLS)
- cryptographic schemes for securing communications and identification information over the web.
- Although there are many benefits to using HTTPS, there are a few drawbacks,
- most notably some performance issues because encryption and decryption consume significant computer processing resources.

Web Basics

Web Basics

- web browser and a web server.
- Hyperlinks
- URL/URI’s
- HTTP request
- HTTP Response
- HTTP headers
- MIME
- HTTP request types
- Client-Side Caching

Web Basics

- Web Basics includes the fundamentals of web-based interactions between a client web browser and a web server.
- A web page is nothing more than an HTML (HyperText Markup Language) document (with the extension .html or .htm) that describes to a web browser the document's content and structure.
- Web browsers initiate network communications with servers by sending them URLs .
- A URL can specify one of two different things:
  - the address of a data file stored on the server that is to be sent to the client, or
  - a program stored on the server that the client wants executed, with the output of the program returned to the client.

Web Basics

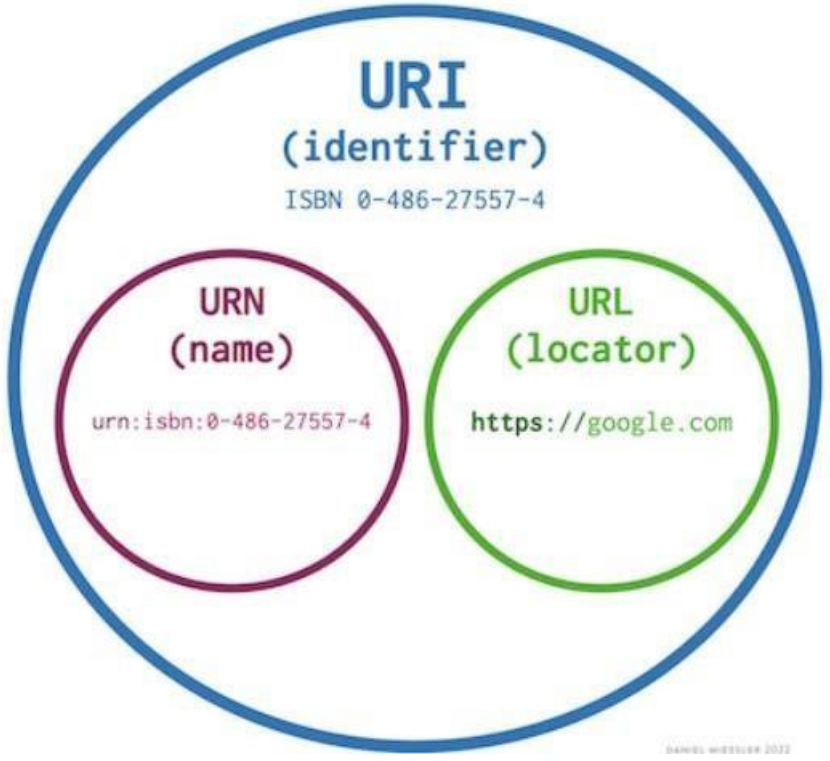
- All the communications between a Web client and a Web server use the standard Web protocol, Hypertext Transfer Protocol (HTTP)
- When a Web server begins execution, it informs the operating system under which it is running that it is now ready to accept incoming network connections through a specific port on the machine.
- A Web client, or browser, opens a network connection to a Web server, sends information requests and possibly data to the server, receives information from the server, and closes the connection.
- The primary task of a Web server is to monitor a communications port on its host machine, accept HTTP commands through that port, and perform the operations specified by the commands.

Hyper links

- A hyperlink is a piece of text that when clicked takes the user to a webpage.
- Hyperlinks can also link to email addresses; when clicked, these hyperlinks will open an email program (likely Microsoft Outlook) to send an email to that address.
- Hyperlinks are formatted with a different color (blue, by default) and an underline.
- Both images and text may be hyperlinked. When the mouse pointer hovers over a hyperlink, the default arrow pointer changes into a hand with the index finger pointing upward
- When the user clicks a hyperlink, a web server locates the requested web page and sends it to the user's web browser
- Hyperlinks can reference other web pages, e-mail addresses, files and more.

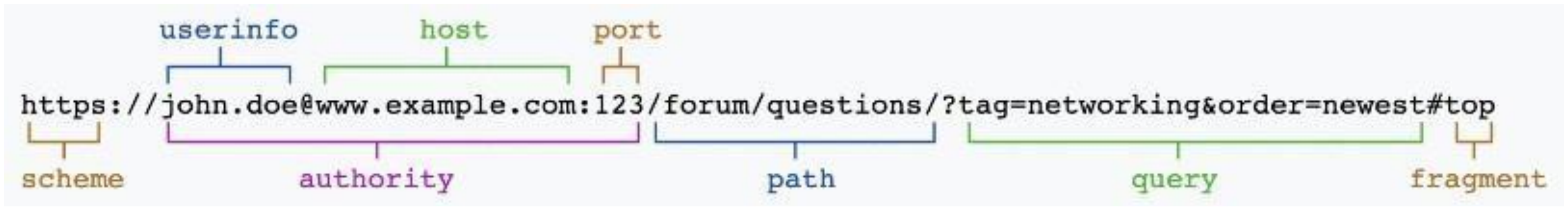
URI’s, URL’s and URN

- A Uniform Resource Identifier (URI) is a string of characters that uniquely identify a name or a resource on the internet.
- A URI identifies a resource by name, location, or both.
- URIs have two specializations known as
  - Uniform Resource Locator (URL), and
  - Uniform Resource Name (URN)
- A Uniform Resource Locator (URL) is a type of URI that specifies not only a resource, but how to reach it on the internet—like http://, ftp://, or mailto://.
- A Uniform Resource Name (URN) is a type of URI that uses the specific naming scheme of urn:—like urn:isbn:0-486-27557-4 or urn:isbn:0-395-36341-1



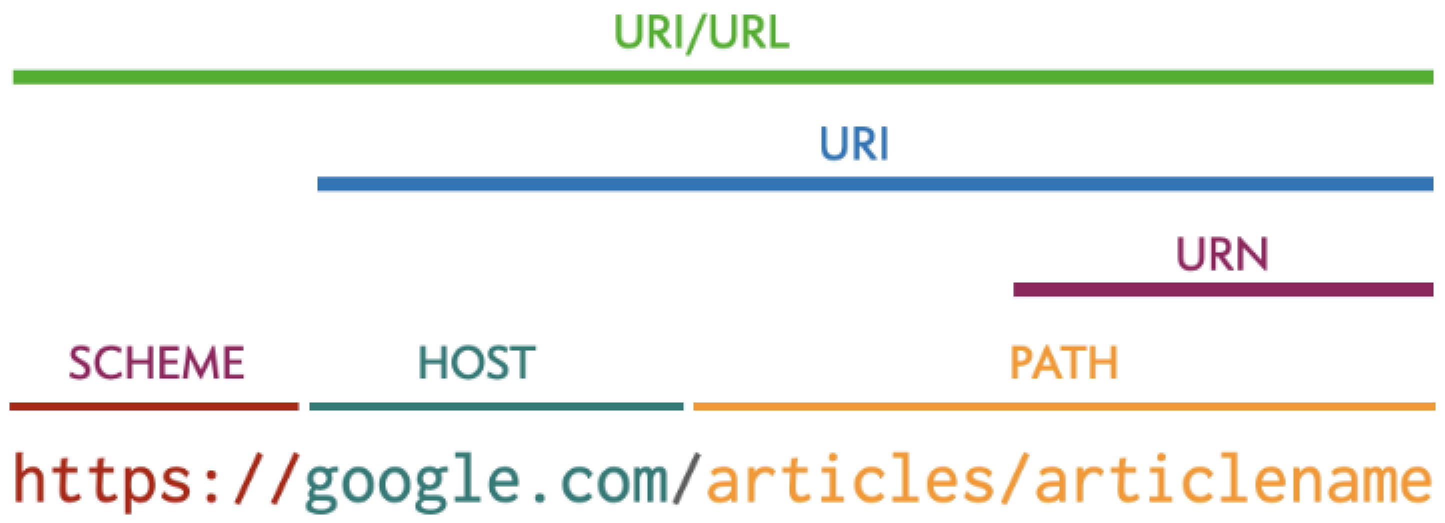
URL

- A URL contains information that directs a browser to the resource that the user wishes to access. Web servers make such resources available to web clients
- URIs (Uniform Resource Identifiers) identify resources on the Internet.
- URIs that start with http:// are called URLs (Uniform Resource Locators).
- Common URLs refer to files, directories or server-side code that performs tasks such as database lookups, Internet searches and business-application processing.


userinfo
host
port
https://john.doe@www.example.com:123/forum/questions/?tag=networking&order=newest#top
scheme
authority
path
query
fragment

URI’s, URL’s and URN -Structures

- Parts of a URL---- scheme:object-address
    - scheme : communications protocol, such as telnet , http, mailto ,ftp
    - object address used to request and send HTML docs
    - For the http protocol, the object-address is: fully qualified domain name/doc path
    - For the file protocol, only the doc path is needed



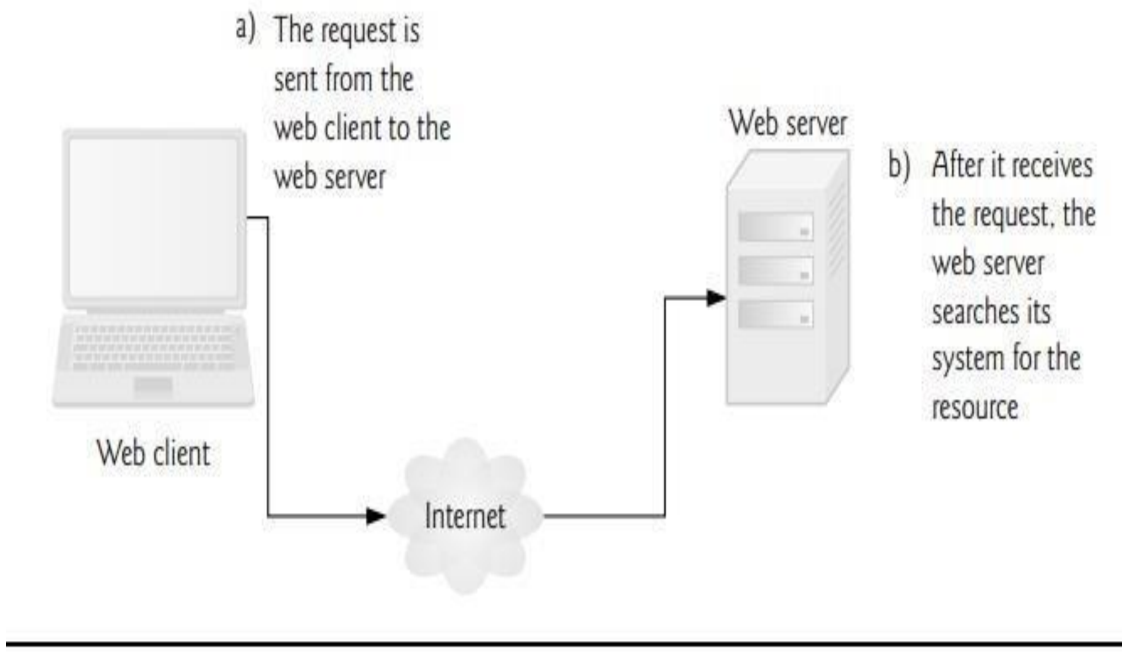
components of the URL

http:// www.deitel.com/ books/downloads.html

- The text http:// indicates that the HyperText Transfer Protocol (HTTP) should be used to obtain the resource.
- Next in the URL is the server's fully qualified hostname (for example, www.deitel.com)—the name of the web-server computer on which the resource resides.
- The hostname **www.deitel.com** is translated into an IP (Internet Protocol) address—a numerical value that uniquely identifies the server on the Internet.
- An Internet Domain Name System (DNS) server maintains a database of hostnames and their corresponding IP addresses and performs the translations automatically.
- The remainder of the URL (/books/downloads.html) specifies the resource's location (/books) and name (downloads.html) on the web server.
- The location could represent an actual directory on the web server's file system.
- For security reasons, however, the location is typically a virtual directory. The web server translates the virtual directory into a real location on the server, thus hiding the resource's true location

Making a Request and Receiving a Response

- When given a web page URL, a web browser uses HTTP to request the web page found at that address.
- The web browser sends an HTTP request to the server.
- The request (in its simplest form) is The word GET is an HTTP method indicating that the client wishes to obtain a resource from the server.
- The remainder of the request provides the path name of the resource (e.g., an HTML5 document) and the protocol's name and version number (HTTP/1.1).
- The client's request also contains some required and optional headers.
- Any server that understands HTTP (version 1.1) can translate this request and respond appropriately


a) The request is sent from the web client to the web server
Web server
b) After it receives the request, the web server searches its system for the resource
Web client
Internet

Client requesting a resource from a web server.

Making a Request and Receiving a Response

- The server first sends a line of text that indicates the HTTP version, followed by a numeric code and a phrase describing the status of the transaction.
- For example `HTTP/1.1 200 OK`
- indicates success, whereas `HTTP/1.1 404 Not found`
- informs the client that the web server could not locate the requested resource.
- A complete list of numeric codes indicating the status of an HTTP transaction can be found at www.w3.org/Protocols/rfc2616/rfc2616-sec10.html.
- Status code is a three-digit number; first digit specifies the general status

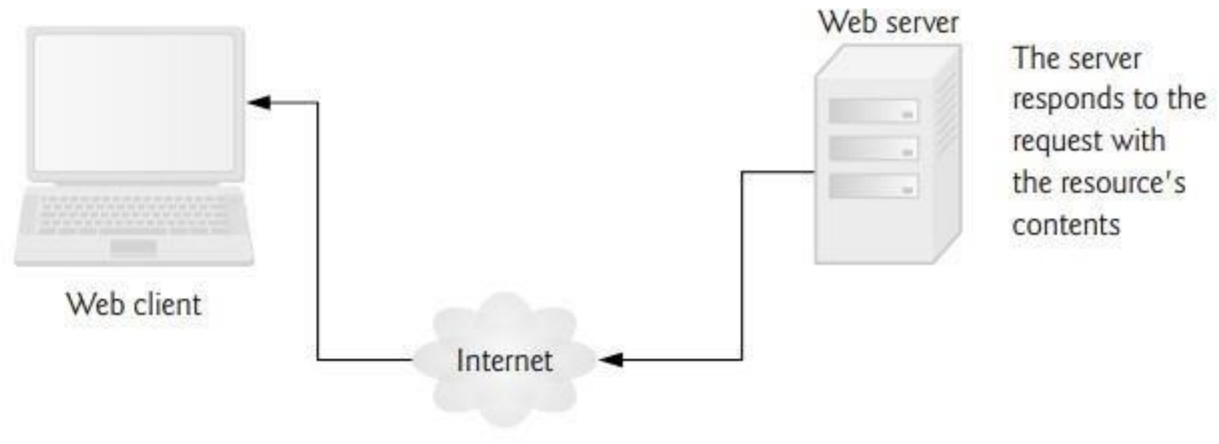
  1 => Informational

  2 => Success

  3 => Redirection

  4 => Client error

  5 => Server error


Web server
The server responds to the request with the resource's contents
Web client
Internet

Client receiving a response from the web server.

HTTP Headers

- HTTP is the application layer protocol used by all Web communications
- HTTP consists of two phases: (current version 1.1)
  - the request and the response.
- Each HTTP communication (request or response) between a browser and a Web server consists of two parts:
  - a header and a body.
  - The header contains information about the communication;
  - the body contains the data of the communication if there is any
- The server sends one or more HTTP headers, which provide additional information about the data that will be sent
- The server is sending an HTML5 text document, so one HTTP header for this example would read

```
Content-type: text/html
```

- The information provided in this header specifies the Multipurpose Internet Mail Extensions (MIME) type of the content that the server is transmitting to the browser

Multipurpose Internet Mail Extensions (MIME)

- MIME is an internet standard that specifies the data format of the content that the server is transmitting to the browser, so that programs can interpret the data correctly.
- Browser needs a format of the document it receives from a web server.
- MIME Specifies the format of different kinds of docs . Without knowing the form of the document ,browsers would not be able to render it .
- Web server attaches MIME format specification to the beginning of the document ,to provide to a browser.
- When browser receives the document ,it uses MIME format specification to determine what to do with the document.
  - Syntax : **type/subtype**
  - Common MIME types 1) text 2) images 3) video
  - SubTypes-text→plain,html, images→gif,jpeg, video→mpeg, quicktime
  - Examples: text/plain, text/html, image/gif,image/jpeg
  - MIME type text/plain indicates that the sent information is text that can be displayed directly.
  - Similarly, the MIME type image/jpeg indicates that the content is a JPEG image.

Important MIME types for Web developers

- application/octet-stream : Default for binary files.
- text/plain: Default for textual files.
- text/css : CSS files used to style a Web page
- text/html : HTML content should be served
- text/javascript :JavaScript content should always be served
- image/apng : Animated Portable Network Graphics (APNG)
- image/avif : AV1 Image File Format (AVIF)
- image/gif: Graphics Interchange Format (GIF)
- image/jpeg: Joint Photographic Expert Group image (JPEG)
- image/png : Portable Network Graphics (PNG)
- image/svg+xml : Scalable Vector Graphics (SVG)
- image/webp : Web Picture format (WEBP)
- audio/wave,audio/wav audio/x-wav audio/x-pn-wav : An audio file in the WAVE container format.
- audio/webm : An audio file in the WebM container format.
- video/webm : A video file, possibly with audio, in the WebM container format.
- audio/ogg : An audio file in the Ogg container format.
- video/ogg : A video file, possibly with audio, in the Ogg container format

HTTP headers

- The header or set of headers is followed by a blank line, which indicates to the client browser that the server is finished sending HTTP headers.
- Finally, the server sends the contents of the requested document (downloads.html).
- The client-side browser then renders (or displays) the document, which may involve additional HTTP requests to obtain associated CSS and images.
- An example of the first line of a request:

```
GET  /ktu.edu/degrees.html  HTTP/1.1
```

- Most commonly used methods:
  - GET - Fetch a document and returns the content of specification
  - POST - Execute the document, using the enclosed data in body
  - HEAD - Fetch just the header of the document
  - PUT – replace the specific doc/Store a new document on the server
  - DELETE - Remove a document from the server

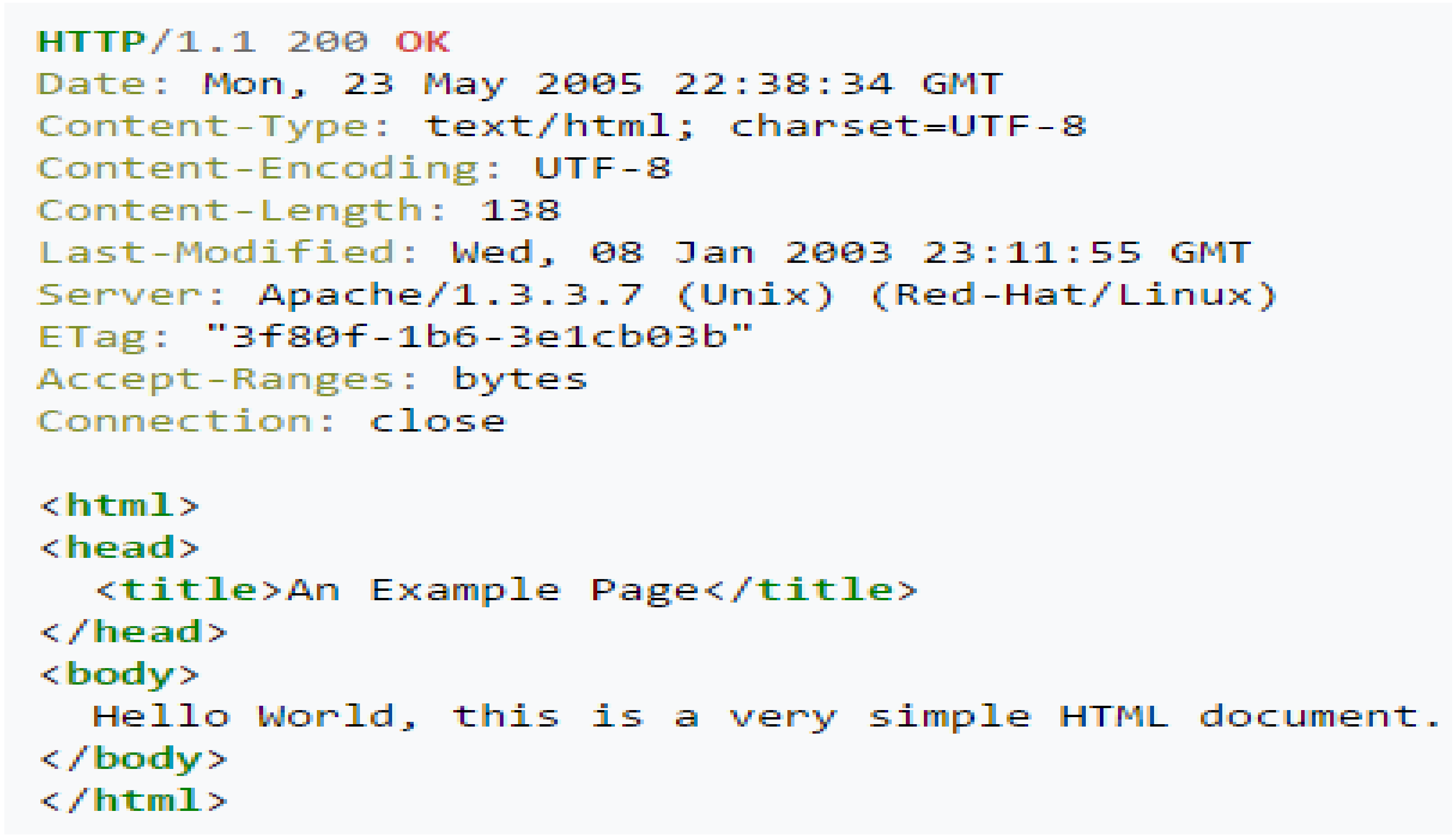
HTTP get and post Requests

- The two most common HTTP request types (also known as request methods) are get and post.
- A get request typically gets (or retrieves) information from a server, such as an HTML document, an image or search results based on a user-submitted search term.
- A post request typically posts (or sends) data to a server. Common uses of post requests are to send form data or documents to a server.
- An HTTP request often posts data to a server-side form handler that processes the data.
  - A get request appends data to the URL, e.g., www.google.com/search?q=deitel.
  - In this case search is the name of Google's server-side form handler, q is the name of a variable in Google's search form and deitel is the search term
  - The ? in the preceding URL separates the query string from the rest of the URL in a request.
  - A name/value pair is passed to the server with the name and the value separated by an equals sign (=).
  - If more than one name/ value pair is submitted, each pair is separated by an ampersand (&).
  - The server uses data passed in a query string to retrieve an appropriate resource from the server and then sends a response to the client.
  - A get request may be initiated by submitting an HTML form whose method attribute is set to "get", or by typing the URL (possibly containing a query string) directly into the browser's address bar

HTTP get and post Requests

- A post request sends form data as part of the HTTP message, not as part of the URL.
- A get request typically limits the query string (i.e., everything to the right of the ?) to a specific number of characters, so it’s often necessary to send large amounts of information using the post method.
- The post method is also sometimes preferred because it hides the submitted data from the user by embedding it in an HTTP message.
- If a form submits several hidden input values along with user-submitted data, the post method might generate a URL like www.searchengine.com/search.
- The form data still reaches the server and is processed in a similar fashion to a get request, but the user does not see the exact information sent

HTTP RESPONSE EXAMPLE



```
HTTP/1.1 200 OK
Date: Mon, 23 May 2005 22:38:34 GMT
Content-Type: text/html; charset=UTF-8
Content-Encoding: UTF-8
Content-Length: 138
Last-Modified: Wed, 08 Jan 2003 23:11:55 GMT
Server: Apache/1.3.3.7 (Unix) (Red-Hat/Linux)
ETag: "3f80f-1b6-3e1cb03b"
Accept-Ranges: bytes
Connection: close

<html>
<head>
  <title>An Example Page</title>
</head>
<body>
  Hello World, this is a very simple HTML document.
</body>
</html>
```

Response fields

- ETag (entity tag) header field is used to determine if a cached version of the requested resource is identical to the current version of the resource on the server.
- *Content-Type* specifies the Internet media type of the data conveyed by the HTTP message
- *Content-Length* indicates its length in bytes.
- *Accept-Ranges: bytes:* HTTP/1.1 webserver publishes its ability to respond to requests for certain byte ranges of the document by setting the field *Accept-Ranges: bytes*.
- *Connection: close* is sent, it means that the web server will close the TCP connection immediately after the transfer of this response.
- Most of the header lines are optional.
- A *Content-Encoding* can be used to **compress the transmitted data**.

Client-Side Caching

- Browsers often cache (save on disk) recently viewed web pages for quick reloading.
- If there are no changes between the version stored in the cache and the current version on the web, this speeds up your browsing experience.
- An HTTP response can indicate the length of time for which the content remains “fresh.”
- If this amount of time has not been reached, the browser can avoid another request to the server.
- If not, the browser loads the document from the cache. Similarly, there’s also the “not modified” HTTP response, indicating that the file content has not changed since it was last requested (which is information that’s send in the request).
- Browsers typically do not cache the server’s response to a post request, because the next post might not return the same result.

Web Basics

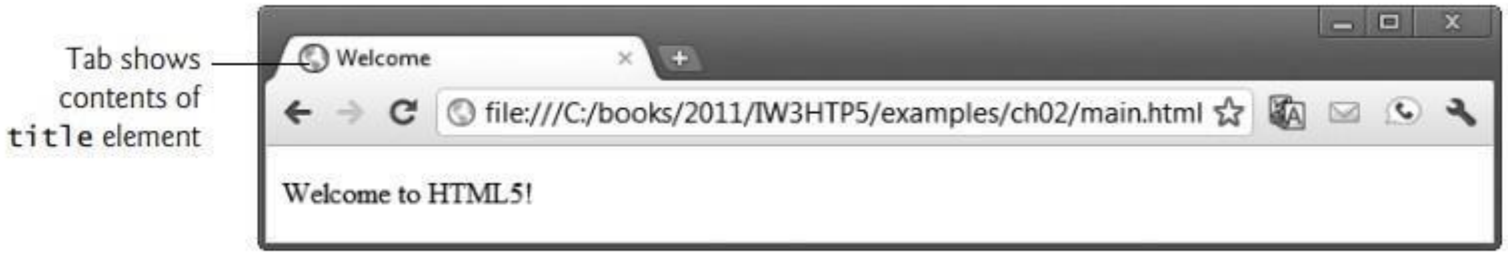
Introduction to HTML5

- HTML5 is a markup language that specifies the structure and content of documents that are displayed in web browsers.
- HTML5 is a markup language used for structuring and presenting content on the World Wide Web.
- It is the fifth and final major HTML version that is a World Wide Web Consortium (W3C) recommendation.
- The current specification is known as the HTML Living Standard.
- It is maintained by the Web Hypertext Application Technology Working Group (WHATWG), a consortium of the major browser vendors (Apple, Google, Mozilla, and Microsoft).
- HTML5 was published as a Working Draft by the W3C in January 2008.
- In January 2011, the WHATWG renamed its "HTML5" living standard to "HTML".
- In December 2012, W3C designated HTML5 as a Candidate Recommendation
- On 28 October 2014, HTML5 was released as a stable W3C Recommendation, meaning the specification process is complete.

First HTML5 Example

- Tag format:
  - Opening tag: <p>
  - Closing tag: </p>
- The opening tag and its closing tag together specify a container for the content they enclose
- Not all tags have content
- If a tag has no content, its form is <br/>
- Comment form: <!-- … -->
- Browsers ignore comments, unrecognizable tags, line breaks, multiple spaces, and tabs

```
<!DOCTYPE html>


<!-- First HTML5 example. -->
<html>
   <head>
      <meta charset = "utf-8">
      <title>Welcome</title>
   </head>

   <body>
      <p>Welcome to HTML5!</p>
   </body>
</html>
```


Tab shows contents of title element
Welcome
file:///C:/books/2011/IW3HTP5/examples/ch02/main.html

Welcome to HTML5!

First HTML5 example.

First HTML5 Example

- Every XHTML document must begin with: **<!DOCTYPE html>**
- The <!DOCTYPE> declaration is not an HTML tag; it is an instruction to the web browser about what version of HTML the page is written in.

- <html>, <head>, <title>, and <body> are required in every document

  - The whole document must have <html> as its root. A document consists of a head and a body

- The **<title>** tag is used to give the document a title, which is normally displayed in the browser's window title bar (at the top of the display).This is called a nested element, because it's enclosed in the head element's start and end tags
- The meta tag is used to provide the character set used   <meta charset = "utf-8" />
- UTF-8 Unicode Transformation Format(8 bit blocks to represent a character)
- Unicode-international encoding standard. Charset for all languages.it assigns a unique number for every character called code point,.0-127 chars identical to ASCII Unicode can be implemented by different character encodings.
- The Unicode standard defines UTF-8, UTF-16, and UTF-32, and several other encodings are in use

W3C HTML5 Validation Service

- The World Wide Web Consortium (W3C) provides a validation service (at validator.w3.org) for checking a document's syntax
- XHTML documents
  - Text editor (e.g. Notepad, Wordpad, sublime, etc.) or
  - Use software like Expression Web or Dreamweaver/Frontpage
  - `.html` or `.htm` file-name extension
  - To use validator.w3.org/#validate-by-upload, click the Choose File button to select a file from your computer to validate. Next, click More Options.
  - In the Document Type drop-down list, select HTML5 (experimental). Select the Verbose Output checkbox, then click the Check button to validate your document.
  - If it contains syntax errors, the validation service displays error messages describing the errors.

Headings

Headings

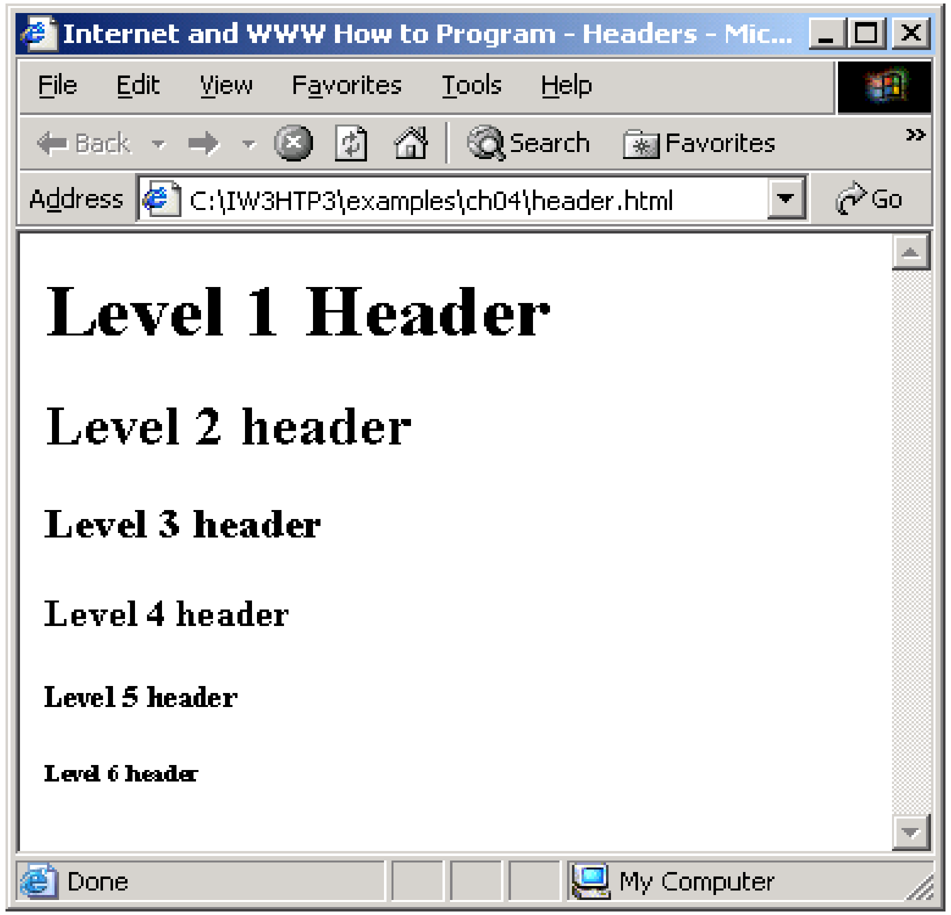
- HTML headings are titles or subtitles that you want to display on a webpage.
- The <h1> to <h6> tags are used to define HTML headings.
- <h1> defines the most important heading. <h6> defines the least important heading.
- Browsers automatically add some white space (a margin) before and after a heading.
- <h1> headings should be used for main headings, followed by <h2> headings, then the less important <h3>, and so on.
- Each HTML heading has a default size. However, you can specify the size for any heading with the style attribute, using the CSS font-size property:
- Example

```
<h1 style="font-size:60px;">Heading 1</h1>
```

Headings

- Six sizes, 1 - 6, specified with `<h1>` to `<h6>`
- 1, 2, and 3 use font sizes that are larger than the default font size
- 4 uses the default size
- 5 and 6 use smaller font sizes

```
//header.html
<!DOCTYPE html>
<html>
  <head>
    <title> Headings </title>
    <meta charset = "utf-8" />
  </head>
  <body>
    <h1> Level 1 header </h1>
    <h2> Level 2 header</h2>
    <h3> Level 3 header </h3>
    <h4> Level 4 header </h4>
    <h5> Level 5 header </h5>
    <h6> Level 6 header </h6>
  </body>
</html>
```


Internet and WWW How to Program - Headers - Mic...

Address: C:\IW3HTP3\examples\ch04\header.html

Level 1 Header

Level 2 header

Level 3 header

Level 4 header

Level 5 header

Level 6 header

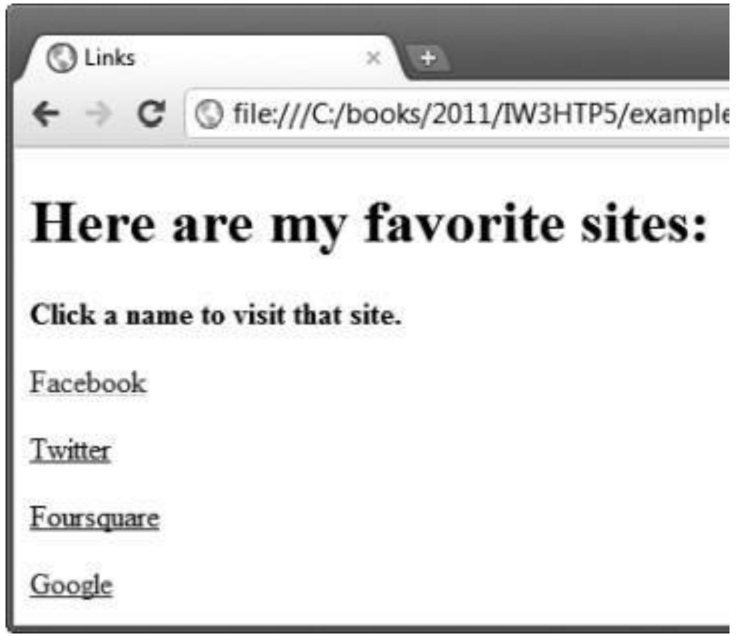
Hyper Links

Linking

- One of the most important HTML5 features is the hyperlink, which references (or links to) other resources, such as HTML5 documents and images.
- When a user clicks a hyperlink, the browser tries to execute an action associated with it
- Any displayed element can act as a hyperlink. Web browsers typically underline text hyperlinks
- Hypertext is the essence of the Web!
- A link is specified with the href (hypertext reference) attribute of <a> (the anchor tag)
- The content of <a> is the visual link in the document
- Both text and images can be the content of hyperlinks
- The target is the document specified in the link

Linking

```
<!DOCTYPE html>

<!-- Fig. 2.3: links.html -->
<!-- Linking to other web pages. -->
<html>
   <head>
      <meta charset = "utf-8">
      <title>Links</title>
   </head>

   <body>
      <h1>Here are my favorite sites:</h1>
      <p><strong>Click a name to visit that site.</strong></p>

      <!-- create four text hyperlinks -->
      <p><a href = "http://www.facebook.com">Facebook</a></p>
      <p><a href = "http://www.twitter.com">Twitter</a></p>
      <p><a href = "http://www.foursquare.com">Foursquare</a></p>
      <p><a href = "http://www.google.com">Google</a></p>
   </body>
</html>
```



Here are my favorite sites:

Click a name to visit that site.

Facebook

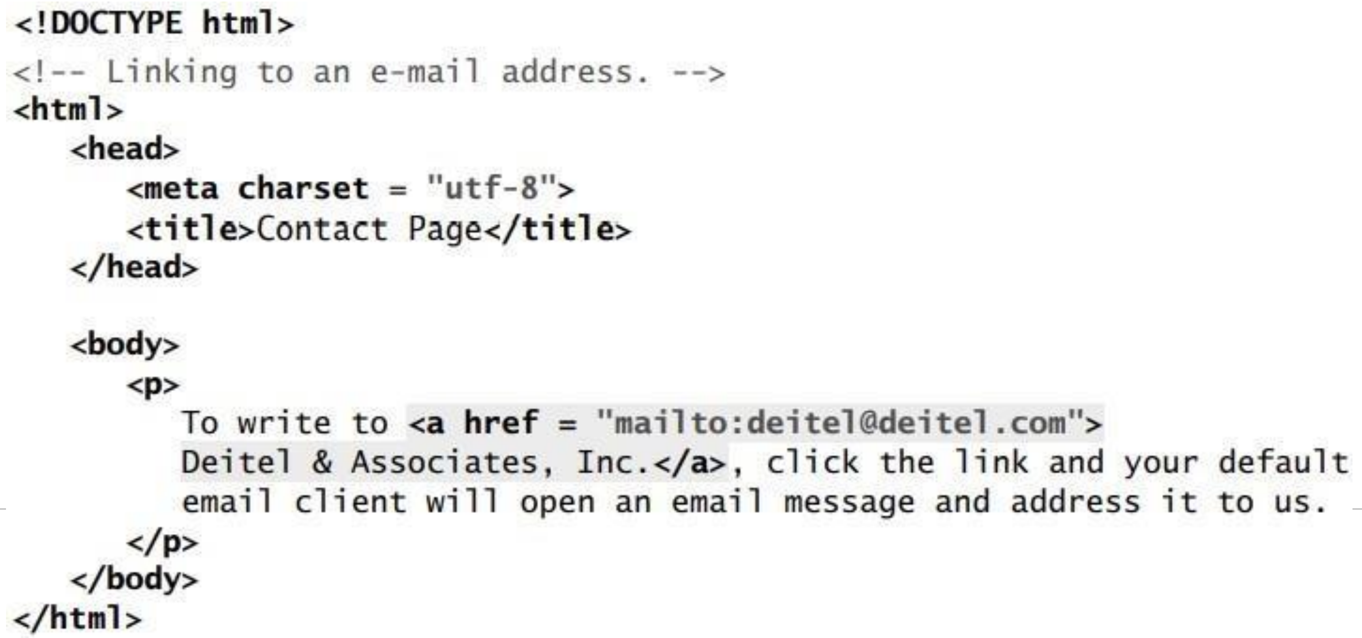
Twitter

Foursquare

Google

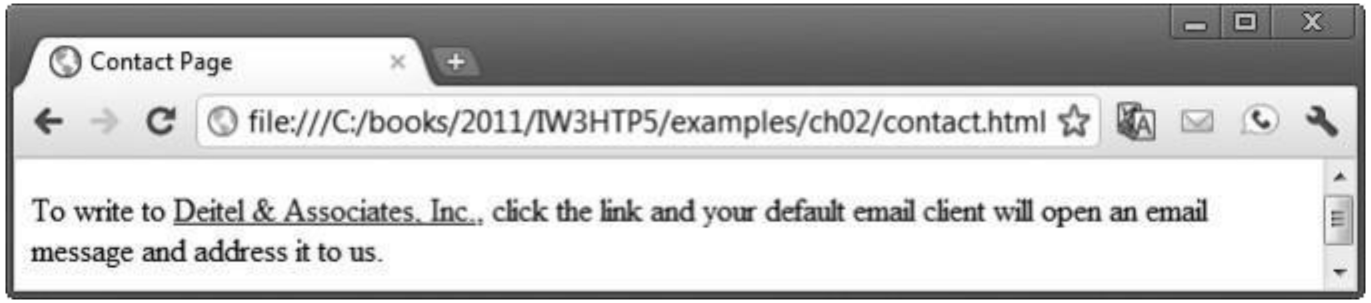
Hyperlinking to an E-Mail Address

- Anchors can link to e-mail addresses using a mailto: URL.
- When the user clicks this type of anchored link, most browsers launch the user's default e-mail program (for example, Mozilla Thunderbird, Microsoft Outlook or Apple Mail) to enable the user to write an e-mail message to the linked address.
- The form of an e-mail anchor is <a href = "mailto:emailAddress">...</a>. In this case, we link to the e-mail address deitel@deitel.com.

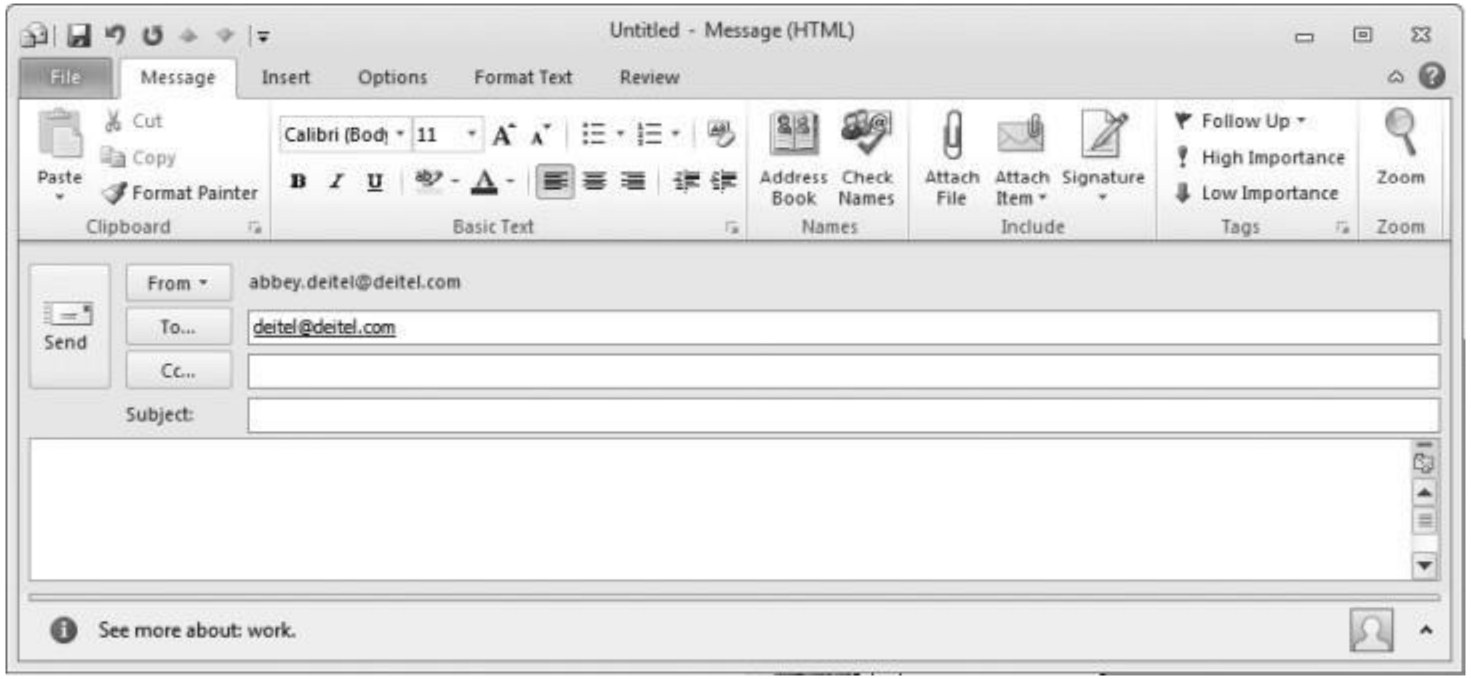


```
<!DOCTYPE html>
<!-- Linking to an e-mail address. -->
<html>
   <head>
      <meta charset = "utf-8">
      <title>Contact Page</title>
   </head>

   <body>
      <p>
         To write to <a href = "mailto:deitel@deitel.com">
         Deitel & Associates, Inc.</a>, click the link and your default
         email client will open an email message and address it to us.
      </p>
   </body>
</html>
```

Linking an email address



To write to Deitel & Associates, Inc., click the link and your default email client will open an email message and address it to us.



From: abbey.deitel@deitel.com
To...: deitel@deitel.com

Internal Linking

- Internal linking—a mechanism that enables the user to jump between locations in the same document.
- Internal linking is useful for long documents that contain many sections.
- Clicking an internal link enables the user to find a section without scrolling through the entire document.
- A tag with the id attribute can create an internal hyperlink.
- To link to a tag with this attribute inside the same web page, the href attribute of an anchor element includes the id attribute value, preceded by a pound sign (as in #features)
- A hyperlink can also reference an internal link in another document by specifying the document name followed by a pound sign and the id value

```
href = "filename.html#id"
```

```
href = "URL/filename.html#id"
```

Internal Linking

- If the target is not at the beginning of the document, the target spot must be marked
- Target labels can be defined in many different tags with the id attribute, as in

```
<h1 id = “www"> World Wide Web</h1>
```

- The link to an id must be preceded by a pound sign (#); If the id is in the same document, this target could be

```
<a href = "#www"> What is www? </a>
```

- If the target is in a different document, the document reference must be included

```
<a href = “faq.html#internet”> Internet   </a>
```

Images

Images

- Images are inserted into a document with the <img /> tag with the src attribute
- The <img> tag has 30 different attributes, including width and height (in pixels)

```
<img src = “sjc.jpg“ alt = "Picture of SJCET" />
```

  - src means source of the image..
  - The alt attribute means ”Alternate Text” which is required by XHTML where the image cannot be loaded by the browser.
  - A browser may not be able to render an image for several reasons
  - It may not support images—as is the case with text-only browsers—or the client may have disabled image viewing to reduce download time.
  - Every img element in an HTML5 document must have an alt attribute.
  - If a browser cannot render an image, the browser displays the alt attribute’s value.

Image Formats

- **GIF (Graphic Interchange Format)**
  - 8-bit color
  - 256 different colors
- **JPEG (Joint Photographic Experts Group)**
  - 24-bit color
  - 16 million different colors
  - Small file size
  - 10:1 compression ratio.(10 MB images will end up as 1 MB after compression.)
  - Both use compression, but JPEG compression is better
- **Portable Network Graphics (PNG)**
  - Relatively new, replaces both gif and jpeg
  - Developed in 1996
  - Compression algorithm does not sacrifice picture quality ie it Supports loseless data compression.
  - Supports palette-based images(24 bit RGB or 32 bit RGBA or gray scale)
  - Size of the image is greater compare to JPEG

Images

```
<!DOCTYPE html>
<!-- Including images in HTML5 files. -->
<html>
   <head>
      <meta charset = "utf-8">
      <title>Images</title>
   </head>

   <body>
      <p>
         <img src = "cpphtp.png" width = "92" height = "120"
            alt = "C++ How to Program book cover">
         <img src = "jhtp.png" width = "92" height = "120"
            alt = "Java How to Program book cover">
      </p>
   </body>
</html>
```

Internet Explorer 9 showing an image and the `alt` text for a missing image

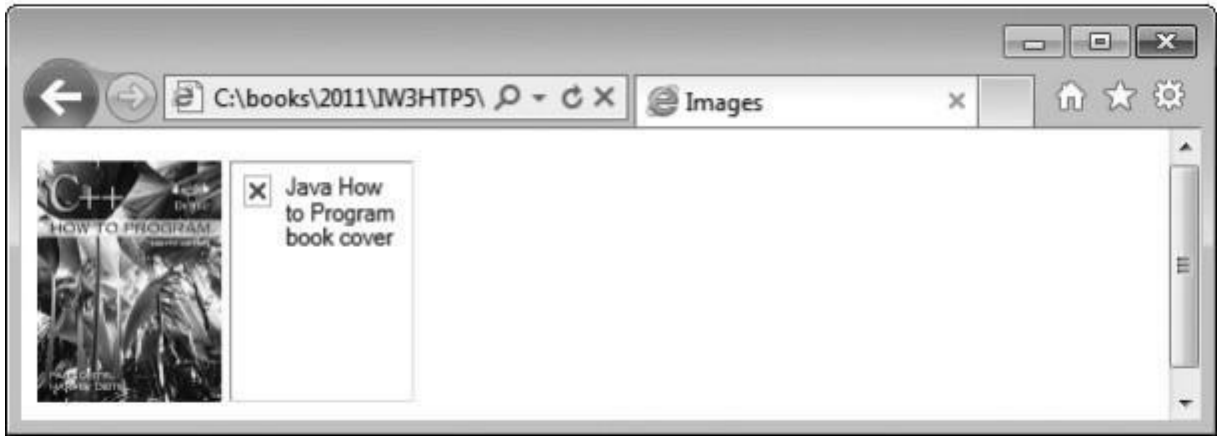


Image tag-void element

- Some HTML5 elements (called void elements) contain only attributes and do not mark up text (i.e., text is not placed between a start and an end tag).
- you can terminate void elements (such as the img element) by using the forward slash character (/) inside the closing right angle bracket (>) of the start tag.

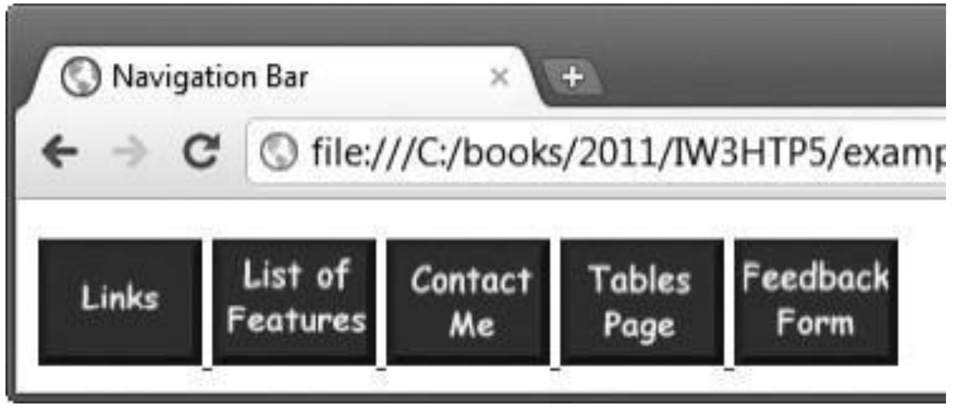
```
<img src = "jhtp.png" width = "92" height = "120"
   alt = "Java How to Program book cover" />
```

- Optional attributes width and height specify the image's dimensions.
- You can scale an image by increasing or decreasing the values of the image width and height attributes. If these attributes are omitted, the browser uses the image's actual width and height.
- Images are measured in pixels ("picture elements"), which represent dots of color on the screen.

Using Images as Hyperlinks

- By using images as hyperlinks, you can create graphical web pages that link to other resources

```
<a href = "links.html">
   <img src = "buttons/links.jpg" width = "65"
      height = "50" alt = "Links">
</a>

<a href = "list.html">
   <img src = "buttons/list.jpg" width = "65"
      height = "50" alt = "List of Features">
</a>

<a href = "contact.html">
   <img src = "buttons/contact.jpg" width = "65"
      height = "50" alt = "Contact Me">
</a>

<a href = "table1.html">
   <img src = "buttons/table.jpg" width = "65"
      height = "50" alt = "Tables Page">
</a>

<a href = "form.html">
   <img src = "buttons/form.jpg" width = "65"
      height = "50" alt = "Feedback Form">
</a>
```



Special Characters and Horizontal Rules

Special Characters and Horizontal Rules

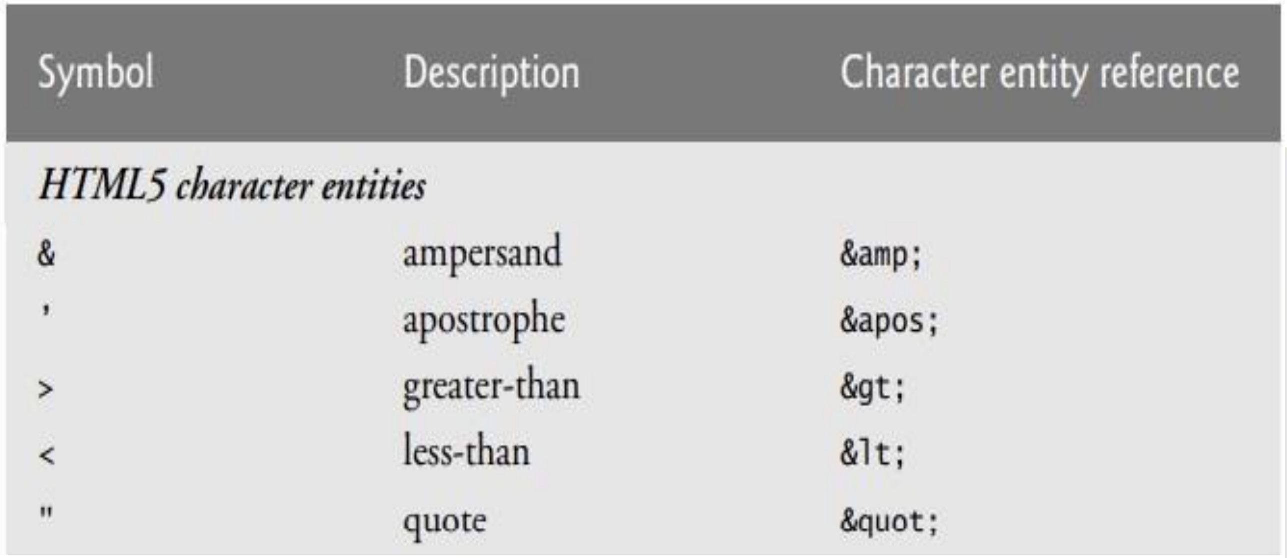
- When marking up text, certain characters or symbols may be difficult to embed directly into an HTML5 document. Some keyboards do not provide these symbols (such as ©), or their presence in the markup may cause syntax errors (as with <). For example,

```
<p>if x < 10 then increment x by 1</p>
```

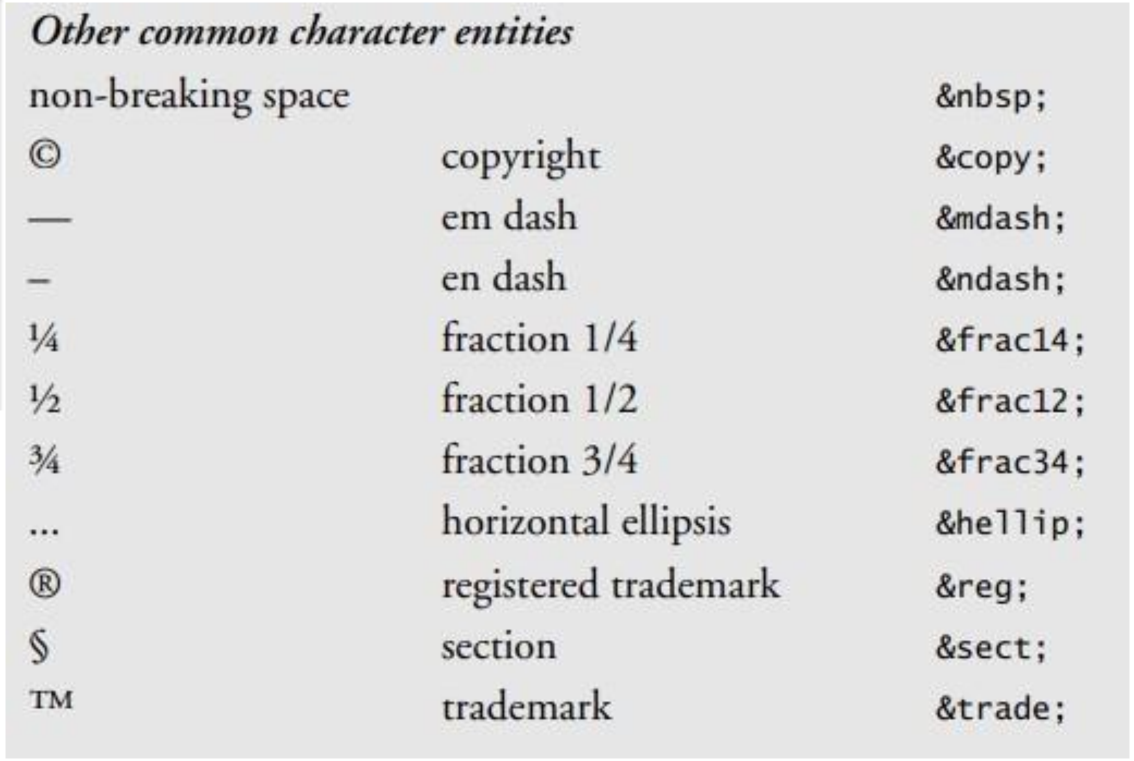
- the markup results in a syntax error because it uses the less-than character (<), which is reserved for start tags and end tags such as <p> and </p>.
- HTML5 provides character entity references (in the form &code;) for representing special characters which uses the character entity reference &lt; for the less-than symbol (<).

```
<p>if x &lt; 10 then increment x by 1</p>
```

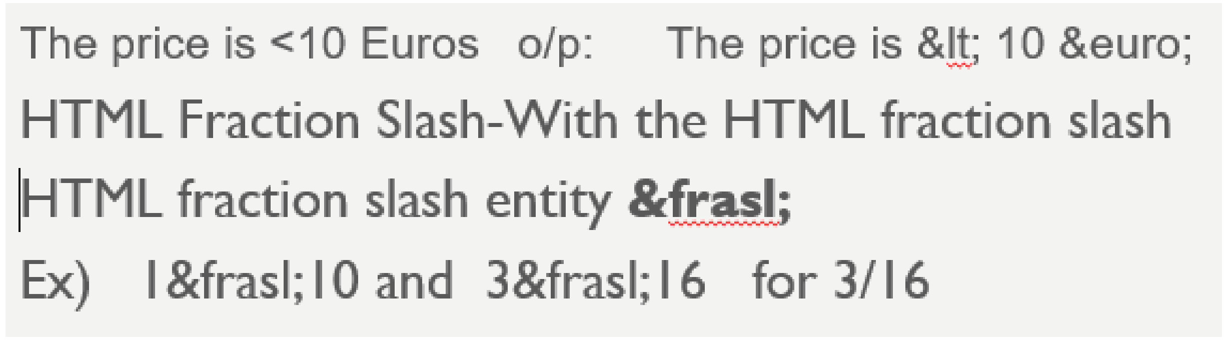
Special Characters and Horizontal Rules



Symbol	Description	Character entity reference
HTML5 character entities		
&	ampersand	`&amp;`
'	apostrophe	`&apos;`
>	greater-than	`&gt;`
<	less-than	`&lt;`
"	quote	`&quot;`



Other common character entities		
non-breaking space		`&nbsp;`
©	copyright	`&copy;`
—	em dash	`&mdash;`
–	en dash	`&ndash;`
¼	fraction 1/4	`&frac14;`
½	fraction 1/2	`&frac12;`
¾	fraction 3/4	`&frac34;`
...	horizontal ellipsis	`&hellip;`
®	registered trademark	`&reg;`
§	section	`&sect;`
™	trademark	`&trade;`



The price is <10 Euros o/p: The price is &lt; 10 &euro;

HTML Fraction Slash-With the HTML fraction slash

HTML fraction slash entity **&frasl;**

Ex) 1&frasl;10 and 3&frasl;16 for 3/16

Special Characters and Horizontal Rules

Horizontal rules

- `<hr />` draws a line across the display, after a line break
- Most browsers render a horizontal rule as a horizontal line with extra space above and below it
- The horizontal rule element should be considered a legacy element and you should avoid using it.

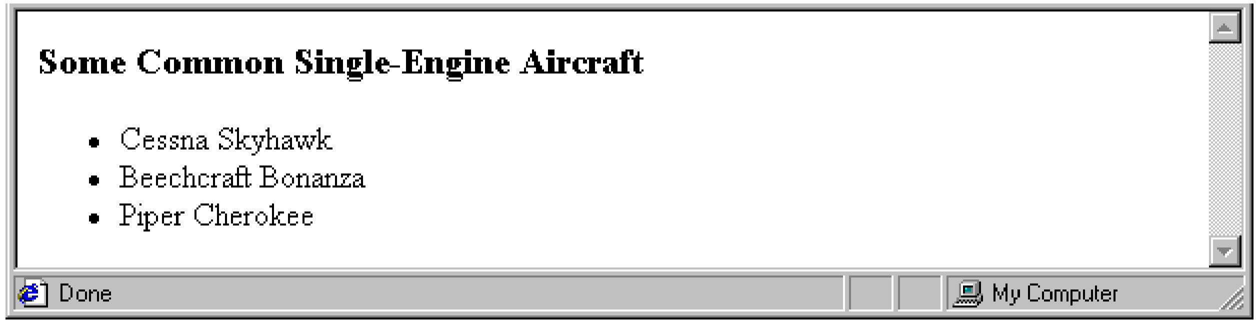
- The <hr> tag in HTML stands for horizontal rule and is used to insert a horizontal rule or a thematic break in an HTML page to divide or separate document sections.
- The <hr> tag is an empty tag, and it does not require an end tag.
- **Tag Attributes are not supported in HTML5:**

Lists

Lists

- lists in a web page to organize content that similar in nature
- ***Unordered lists***
- the `<ul>` tag creates a block tag , creates an unordered list
- List elements are the content of the `<li>` tag
- When displayed , each list item is implicitly preceded by a bullet.

```
<h3> Some Common Single-Engine Aircraft </h3>
 <ul>
   <li> Cessna Skyhawk </li>
   <li> Beechcraft Bonanza </li>
   <li> Piper Cherokee </li>
 </ul>
```

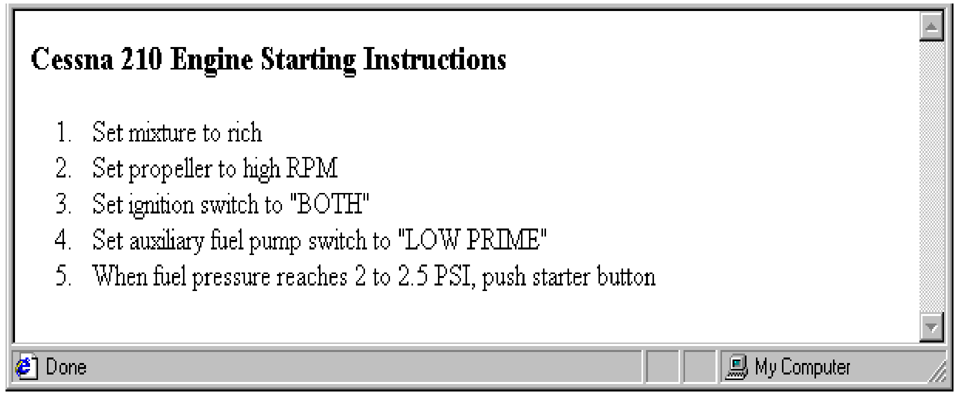


Some Common Single-Engine Aircraft

- Cessna Skyhawk
- Beechcraft Bonanza
- Piper Cherokee

Lists

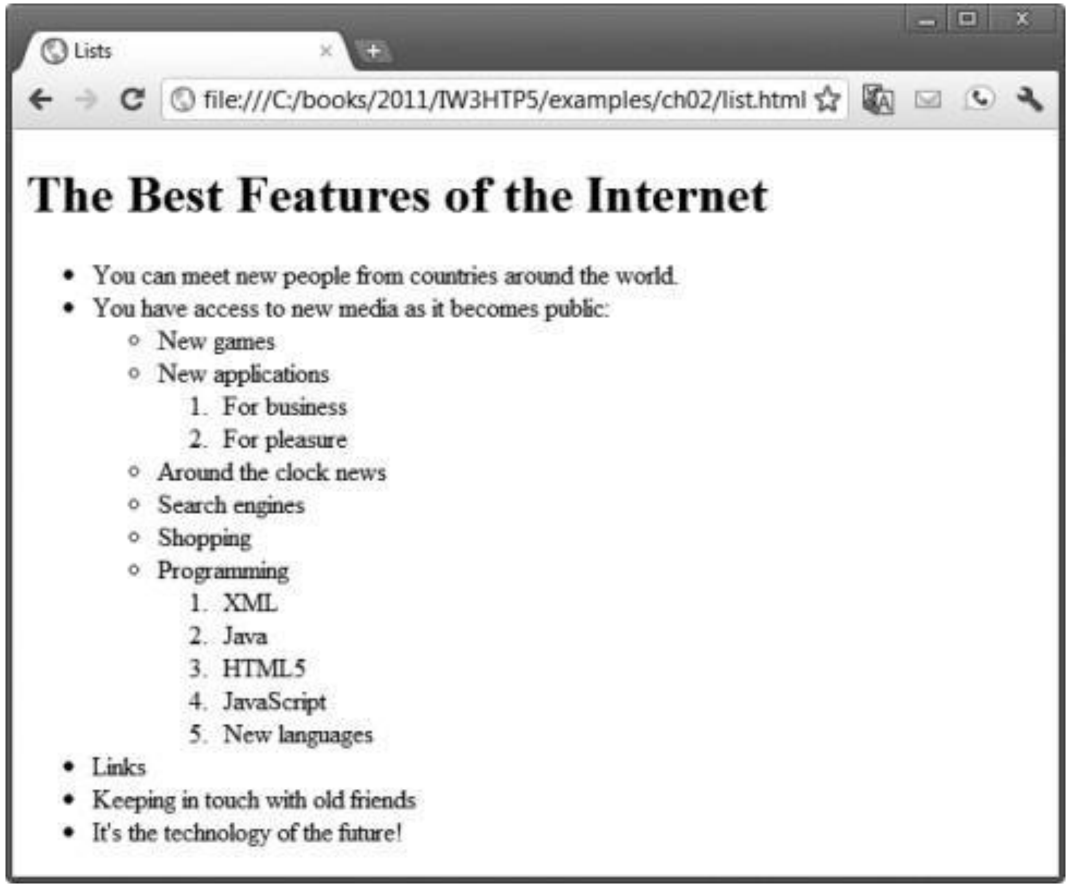
- ***Ordered lists***
  - The list is the content of the `<ol>` tag
  - Implicit attachment of a sequential value to the beginning of each item.
  - Each item in the display is preceded by a sequence value
  - Default sequential values are Arabic Numerals beginning with 1.
- ***Nested lists***
  - Any type list can be nested inside any type list
  - The nested list must be in a list item
  - Lists may be nested to represent hierarchical relationships, as in a multilevel outline.



Cessna 210 Engine Starting Instructions

1. Set mixture to rich
2. Set propeller to high RPM
3. Set ignition switch to "BOTH"
4. Set auxiliary fuel pump switch to "LOW PRIME"
5. When fuel pressure reaches 2 to 2.5 PSI, push starter button

```
<!-- create an unordered list -->
<ul>
   <li>You can meet new people from countries around
      the world.</li>
   <li>
      You have access to new media as it becomes public:

      <!-- this starts a nested unordered list, which uses a -->
      <!-- different bullet. The list ends when you -->
      <!-- close the <ul> tag. -->
      <ul>
         <li>New games</li>
         <li>New applications

            <!-- nested ordered list -->
            <ol>
               <li>For business</li>
               <li>For pleasure</li>
            </ol>
         </li> <!-- ends line 27 new applications li-->

         <li>Around the clock news</li>
         <li>Search engines</li>
         <li>Shopping</li>
         <li>Programming

            <!-- another nested ordered list -->
            <ol>
               <li>XML</li>
               <li>Java</li>
               <li>HTML5</li>
               <li>JavaScript</li>
               <li>New languages</li>
            </ol>
         </li> <!-- ends programming li of line 38 -->
      </ul> <!-- ends the nested list of line 24 -->
   </li>

   <li>Links</li>
   <li>Keeping in touch with old friends</li>
   <li>It's the technology of the future!</li>
</ul> <!-- ends the unordered list of line 15 -->
```



The Best Features of the Internet

- You can meet new people from countries around the world.
- You have access to new media as it becomes public:
  - New games
  - New applications
    1. For business
    2. For pleasure
  - Around the clock news
  - Search engines
  - Shopping
  - Programming
    1. XML
    2. Java
    3. HTML5
    4. JavaScript
    5. New languages
- Links
- Keeping in touch with old friends
- It's the technology of the future!

Tables

Tables

- A table is a matrix of cells, each possibly having content
- The cells can include almost any element
- Some cells have row or column labels and some have data
- A table is specified as the content of a <table> tag
- A border attribute in the <table> tag specifies a border between the cells
- If border is set to "border", the browser's default width border is used
- The border attribute can be set to a number, which will be the border width
- Without the border attribute, the table will have no lines!
- Tables are given titles with the <caption> tag, which can immediately follow <table>
- In HTML5, tables do not have lines between the rows or between the columns
- The caption element specifies a table's title. Text in this element is typically rendered above the table

```
<table border = "1">
    <caption><strong>Table of Fruits (1st column) and
       Their Prices (2nd column)</strong></caption>
    <thead>
       <tr> <!-- <tr> inserts a table row -->
          <th>Fruit</th> <!-- insert a heading cell -->
          <th>Price</th>
       </tr>
    </thead>
    <tfoot>
       <tr>
          <th>Total</th>
          <th>$3.75</th>
       </tr>
    </tfoot>
   <tbody>
      <tr>
         <td>Apple</td> <!-- insert a data cell -->
         <td>$0.25</td>
      </tr>
      <tr>
         <td>Orange</td>
         <td>$0.50</td>
      </tr>
      <tr>
         <td>Banana</td>
         <td>$1.00</td>
      </tr>
      <tr>
         <td>Pineapple</td>
         <td>$2.00</td>
      </tr>
   </tbody>
</table>
```

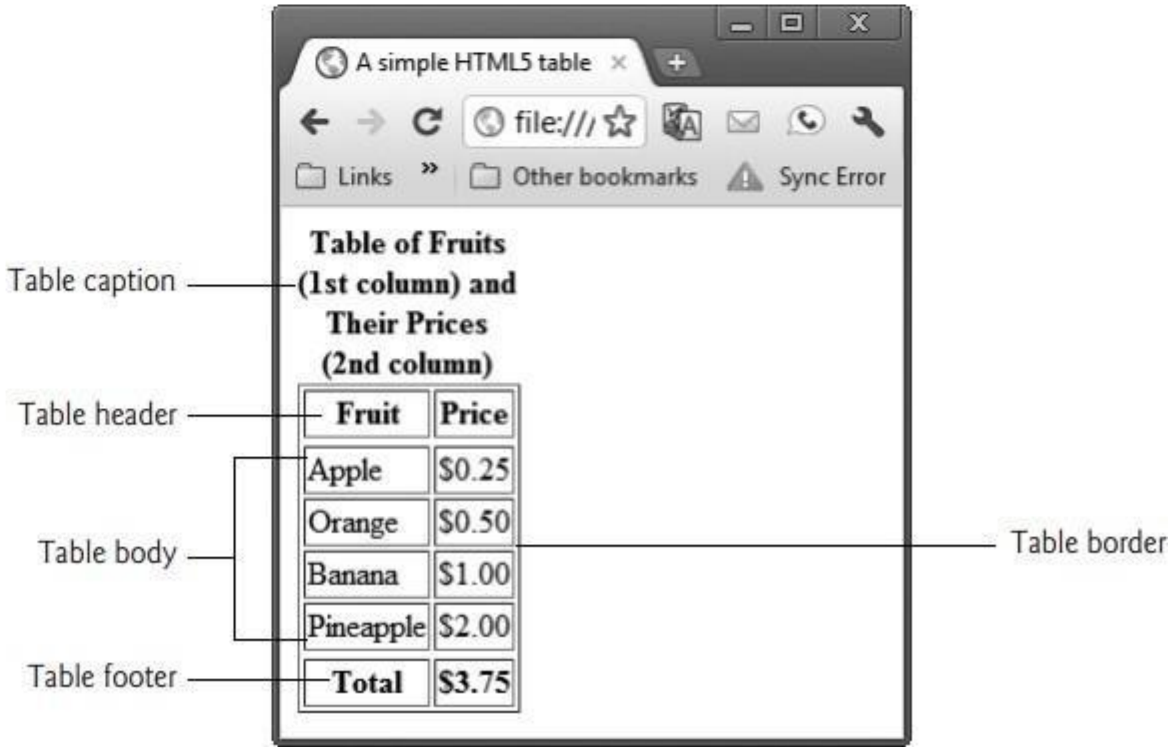


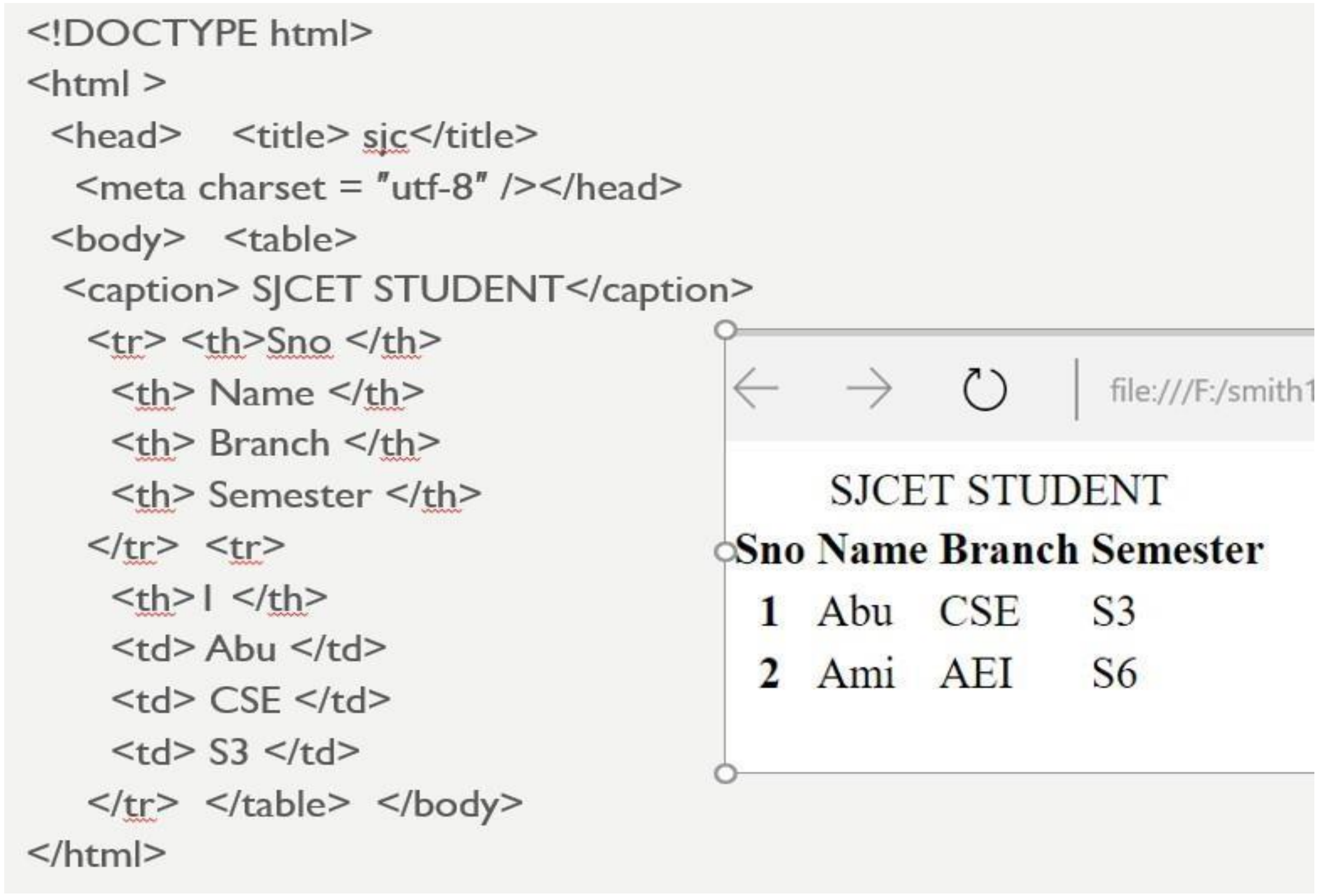
Table of Fruits (1st column) and Their Prices (2nd column)

Fruit	Price
Apple	$0.25
Orange	$0.50
Banana	$1.00
Pineapple	$2.00
Total	**$3.75**

Tables

- A table has three distinct sections—**head, body and foot.**
- The head section (or header cell) is defined with a thead element which contains header information such as column names.
- Each tr element defines an individual table row.
- The columns in the thead section are defined with th elements. Most browsers center text formatted by th (table header column) elements and display them in bold.
- Table header elements are nested inside table row elements.
- The body section, or table body, contains the table's primary data.
- The table body is defined in a tbody element.
- In the table body, each tr element specifies one row.
- Data cells contain individual pieces of data and are defined with td (table data) elements in each row.
- The tfoot section is defined with a tfoot (table foot) element. The text placed in the footer commonly

Tables



```
<!DOCTYPE html>
<html >
 <head>   <title> sjc</title>
  <meta charset = "utf-8" /></head>
 <body>   <table>
  <caption> SJCET STUDENT</caption>
    <tr> <th>Sno </th>
      <th> Name </th>
      <th> Branch </th>
      <th> Semester </th>
    </tr>  <tr>
      <th>1 </th>
      <td> Abu </td>
      <td> CSE </td>
      <td> S3 </td>
    </tr>  </table>  </body>
</html>
```

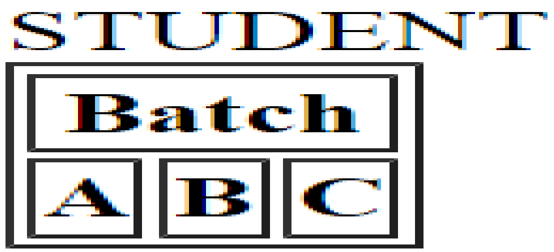
file:///F:/smith1

```
     SJCET STUDENT
Sno Name Branch Semester
 1  Abu   CSE     S3
 2  Ami   AEI     S6
```

Tables

- A table can have two levels of column labels
  - If so, the `colspan` attribute must be set in the `<th>` tag to specify that the label must span some number of columns

```
<tr>
  <th colspan = "3"> Batch </th>
</tr>
<tr>
  <th> A </th>
  <th> B </th>
  <th> C </th>
</tr>
```

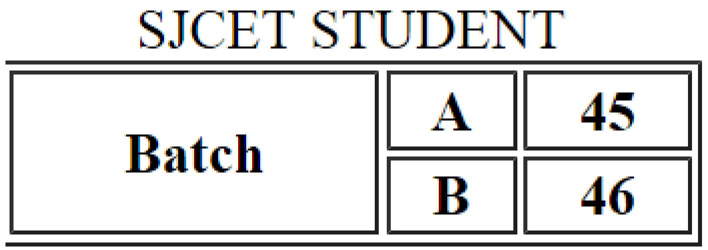


STUDENT

Batch		
A	B	C

- If the rows have labels and there is a spanning column label, the upper left corner must be made larger, using `rowspan`

```
<table border="border" width="200">
   <caption> SJCET STUDENT</caption>
 <tr> <th rowspan = "2"> Batch </th>
 <th> A </th> <th>45 </th></tr>
<tr>  <th> B </th>  <th> 46 </th></tr>
   </table>
```



SJCET STUDENT

Batch	A	45
	B	46

Tables-rowspan and colspan

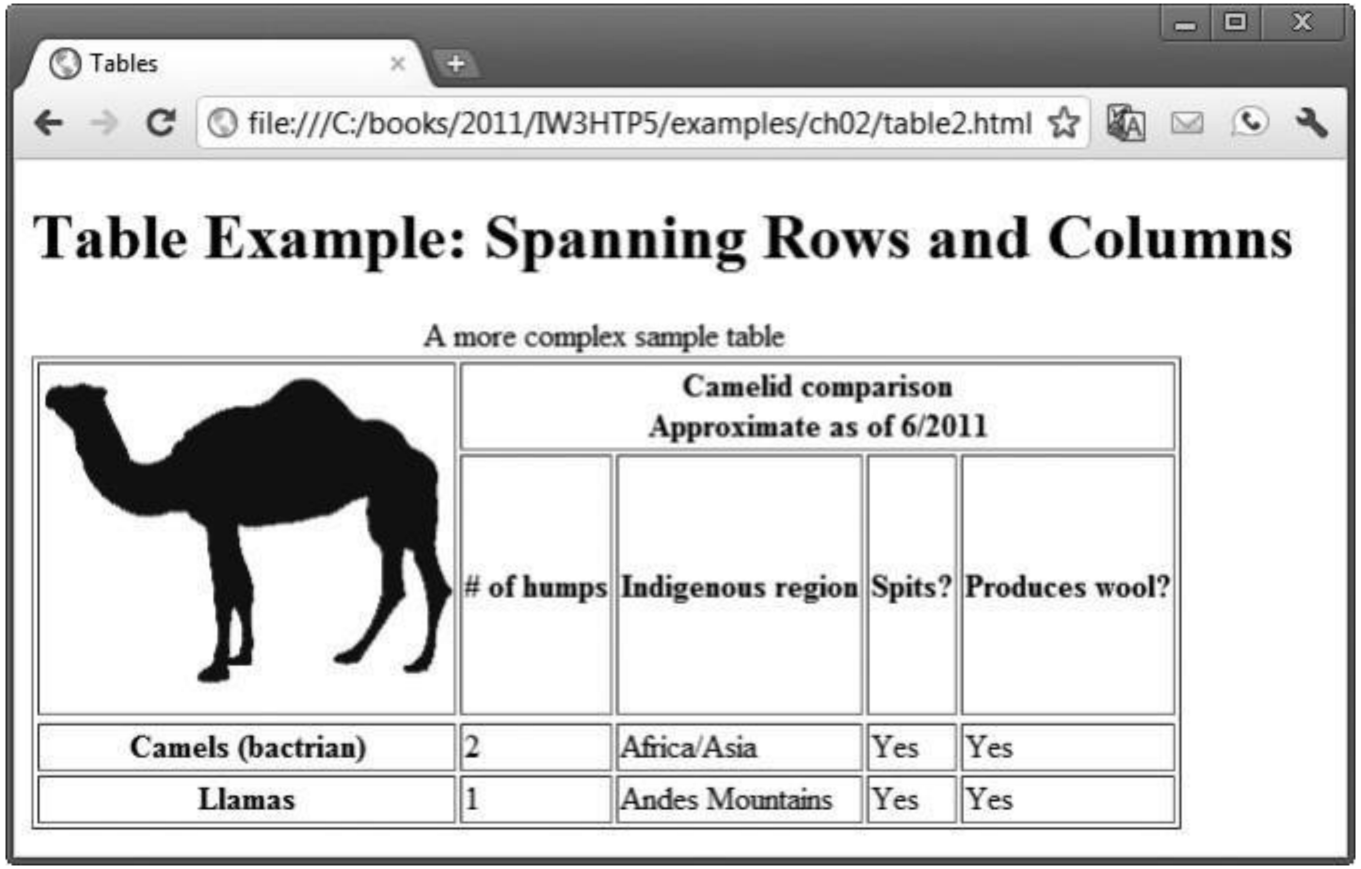
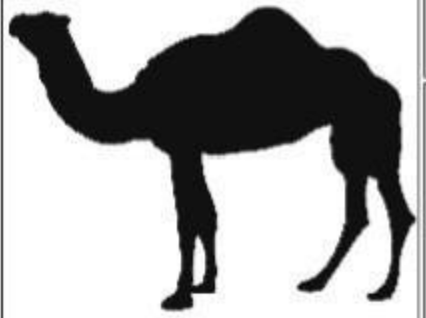


Table Example: Spanning Rows and Columns

A more complex sample table

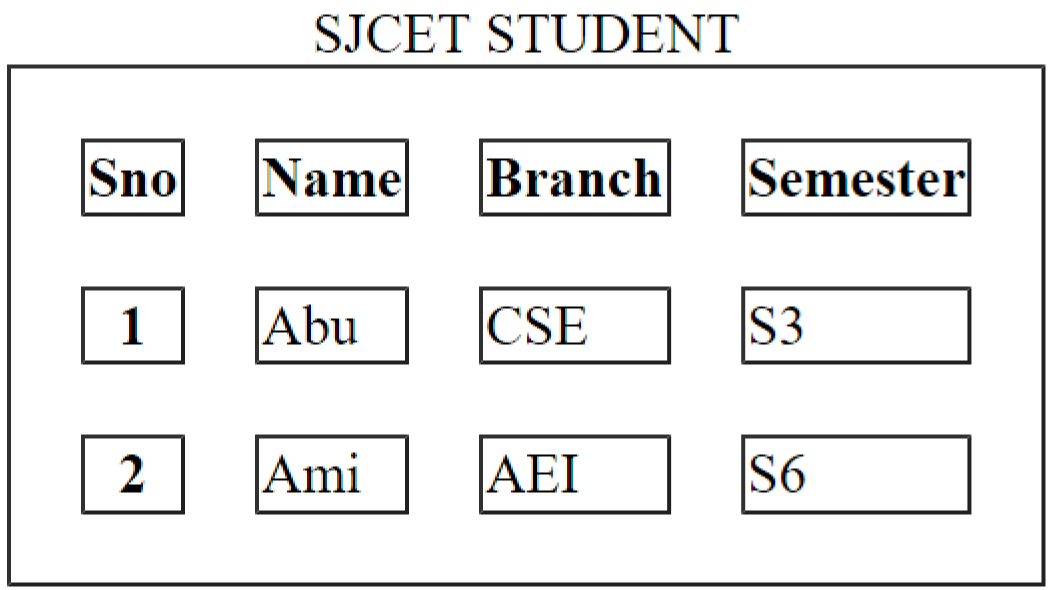
	**Camelid comparison** **Approximate as of 6/2011**			
	# of humps	**Indigenous region**	**Spits?**	**Produces wool?**
Camels (bactrian)	2	Africa/Asia	Yes	Yes
Llamas	1	Andes Mountains	Yes	Yes

Tables

- The `align` attribute controls the horizontal placement of the contents in a table cell
    - Values are `left`, `right`, and `center` (default)
    - `align` is an attribute of `<tr>`, `<th>`, and `<td>` elements
- The `valign` attribute controls the vertical placement of the contents of a table cell
    - Values are `top`, `bottom`, and `center` (default)
    - `valign` is an attribute of `<th>` and `<td>` elements

  → SHOW `cell_align.html` and display it
- The `cellspacing` attribute of `<table>` is used to specify the distance between cells in a table
- The `cellpadding` attribute of `<table>` is used to specify the spacing between the content of a cell and the inner walls of the cell

Tables-cellspacing=20



SJCET STUDENT

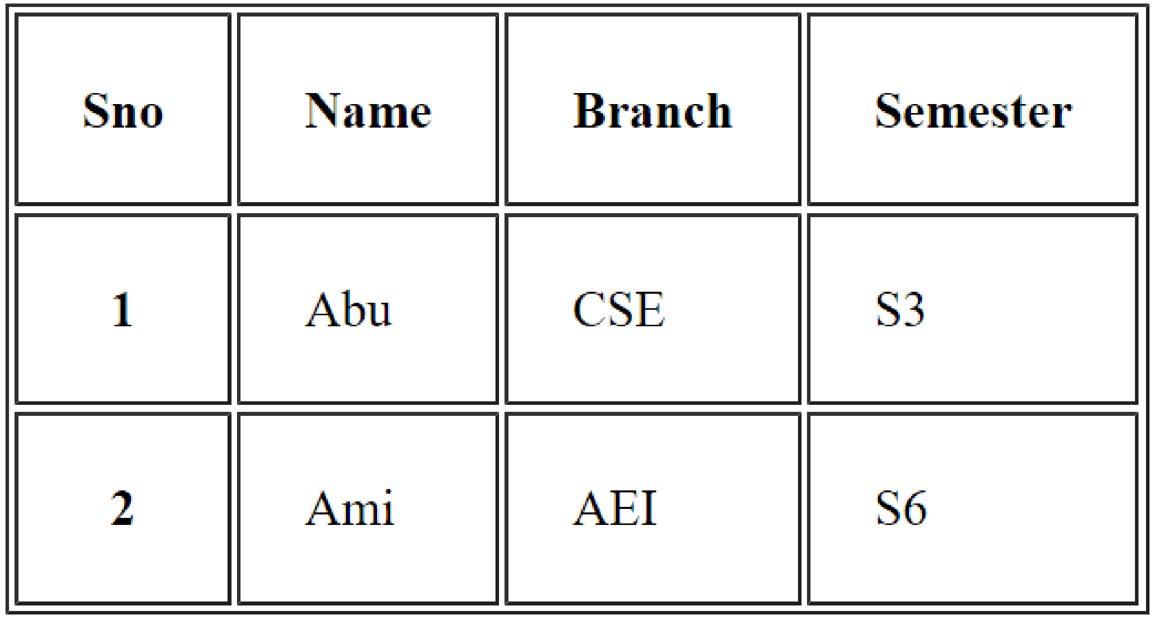
Sno	Name	Branch	Semester
1	Abu	CSE	S3
2	Ami	AEI	S6

specify the distance between cells in a table

- *Table Sections*
  - Header, body, and footer, which are the elements: `thead`, `tbody`, and `tfoot`

Tables-cellpadding=20

SJCET STUDENT



Sno	Name	Branch	Semester
1	Abu	CSE	S3
2	Ami	AEI	S6

- the spacing between the content of a cell and the inner walls of the cell

Forms

Forms

- A form is the usual way information is gotten from a browser to a server
- The HTML <form> element defines a form that is used to collect user input:

```
<form action="url">
        …
        form elements
        …
</form>
```

- HTML has tags to create a collection of objects that implement this information gathering.
- When the Submit button of a form is clicked, the form's values are sent to the server
- All of the widgets, or components of a form are defined in the content of a <form> tag
- The only required attribute of <form> is action, which specifies the URL of the application
  <form action ="reg.php">……. </form>

Forms

- The method attribute specifies how to send form-data
- the form-data is sent to the page specified in the action attribute
- The form-data can be sent as URL variables (with method="get") or as HTTP post transaction (with method="post").
- GET:
  - Appends form-data into the URL in name/value pairs
  - The length of a URL is limited (about 3000 characters)
  - Never use GET to send sensitive data! (will be visible in the URL)
  - Useful for form submissions where a user want to bookmark the result
  - GET is better for non-secure data, like query strings in Google
- POST:
  - Appends form-data inside the body of the HTTP request
  - data is not shown is in URL
  - Has no size limitations
  - Form submissions with POST cannot be bookmarked

Forms

- The method attribute of <form> specifies one of the two possible techniques of transferring the form data to the server, get and post
- <form method="post" action="login.php">
- If the form has no action, the value of action is the empty string. If the action attribute is omitted, the action is set to the current page.
- The action attribute defines the action to be performed when the form is submitted.Normally, the form data is sent to a web page on the server when the user clicks on the submit button.
- Many widgets are created with the <input> tag
- The type attribute of <input> specifies the kind of widget being created

Forms

<input type="text">

- Creates a horizontal box for text input
- Default size is 20; it can be changed with the `size` attribute
- If more characters are entered than will fit, the box is scrolled (shifted) left

<input type="text" name="fname" size="25" maxlength="35"/>

- Users can type data in text fields.
- The input element's size attribute specifies the number of characters visible in the text field.
- Optional attribute maxlength limits the number of characters input into the text field—in this case, the user is not permitted to type more than 30 characters.
- If you don't want to allow the user to type more characters than will fit, set `maxlength`, which causes excess input to be ignored

```
<input type = "text" name = "Phone" size = "12" />
```

```
<input type = "hidden" name = "recipient" value = "deitel@deitel.com">
```

Forms

Checkboxes - to collect multiple choice input

- Every checkbox requires a `value` attribute, which is the widget's value in the form data when the checkbox is 'checked'
  - A checkbox that is not 'checked' contributes no value to the form data
- By default, no checkbox is initially 'checked'
- To initialize a checkbox to 'checked', the `checked` attribute must be set to `"checked"`

```
Grocery Checklist
<form action = "">
<label><input type = "checkbox"  name ="groceries"
         value = "milk"  checked = "checked" /> Milk
    </label>
    <label><input type = "checkbox"  name ="groceries"
         value = "bread" />Bread
    </label>
    <label><input type = "checkbox"  name = "groceries"
         value= "eggs" /> Eggs
    </label
</form>
```

Forms

- Radio Buttons - collections of checkboxes in which only one button can be ‘checked’ at a time
- Every button in a radio button group MUST have the same name
- If no button in a radio button group is ‘pressed’, the browser often ‘presses’ the first one

```
<form action = "">
  <label><input type = "radio"  name = "age"  value = "under20" checked =
  "checked‟ /> 0-19 </label>
   <label><input type = "radio"  name = "age" value = "20-35‟ /> 20-35</label>
   <label><input type = "radio"  name = "age"  value = "36-50‟ /> 36-50 </label>
   <label><input type = "radio"  name = "age" value = "over50 /"> Over 50
  </label>
</form>
```

Forms <select>

- Menus - created with <select> tags
- Menus that behave like checkboxes are specified by including the multiple attribute, which must be set to "multiple"
- The name attribute of <select> is required
- The size attribute of <select> can be included to specify the number of menu items to be displayed (the default is 1)
- If size is set to > 1 or if multiple is specified, the menu is displayed as a pop-up menu
- Each item of a menu is specified with an <option> tag, whose pure text content (no tags) is the value of the item
- An <option> tag can include the selected attribute, which when assigned "selected" specifies that the item is preselected

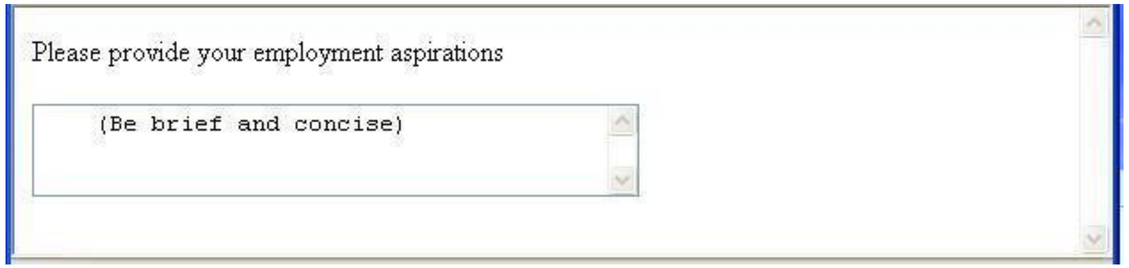
```
Grocery Menu - milk, bread, eggs, cheese
<form action = "">
  <p>
    <label>With size = 1 (the default)
      <select name = "groceries">
         <option> milk </option>
         <option> bread </option>
         <option> eggs </option>
         <option> cheese </option>
      </select>
    </label>
  </p>
</form>
```

Forms

Text areas - created with `<textarea>`

- Usually include the `rows` and `cols` attributes to specify the size of the text area
- Default text can be included as the content of `<textarea>`
- Scrolling is implicit if the area is overfilled

```
Please provide your employment aspirations
<form action = "">
   <p>
     <textarea name = "aspirations"  rows = "3”
             cols = "40">
     (Be brief and concise)
     </textarea>
   </p>
</form>
```



Please provide your employment aspirations

(Be brief and concise)

Forms

Reset and Submit buttons

- Both are created with `<input>`

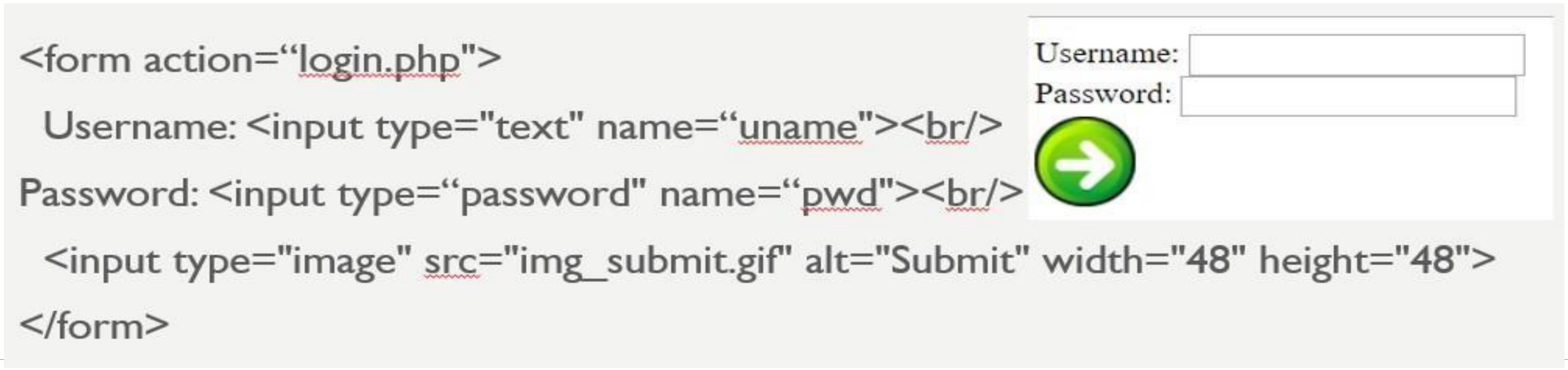
```
<input type = "reset"  value = "Reset Form‶ />
<input type = "submit″  value = "Submit Form‶ />
```

- Submit has two actions:
  1. Encode the data of the form
  2. Request the server to execute the server-resident program specified as the value of the `action` attribute of `<form>`
- A Submit button is required in every form

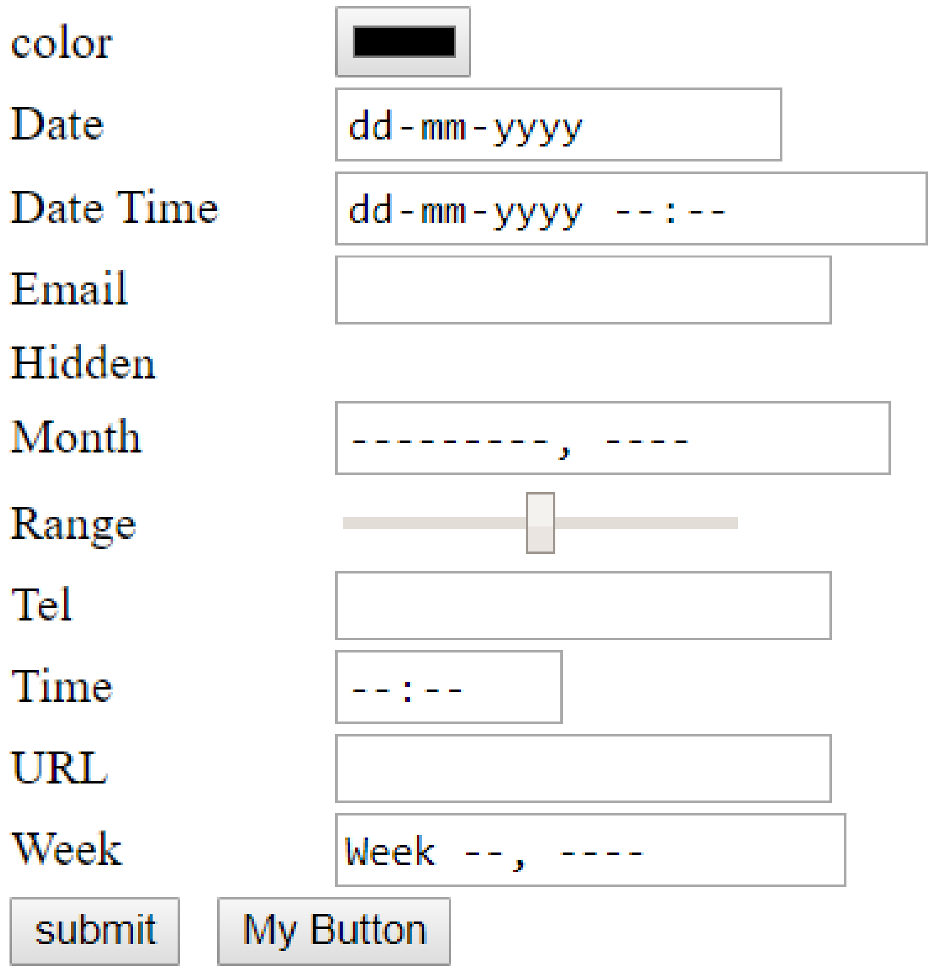
other HTML input types

- <input type="button">
- <input type="color">
- <input type="date">
- <input type="datetime-local">
- <input type="email">
- <input type="file">
- <input type="hidden">
- <input type="tel">
- <input type="time">
- <input type="url">
- <input type="week">
- <input type="month">
- <input type="password">
- <input type="range">

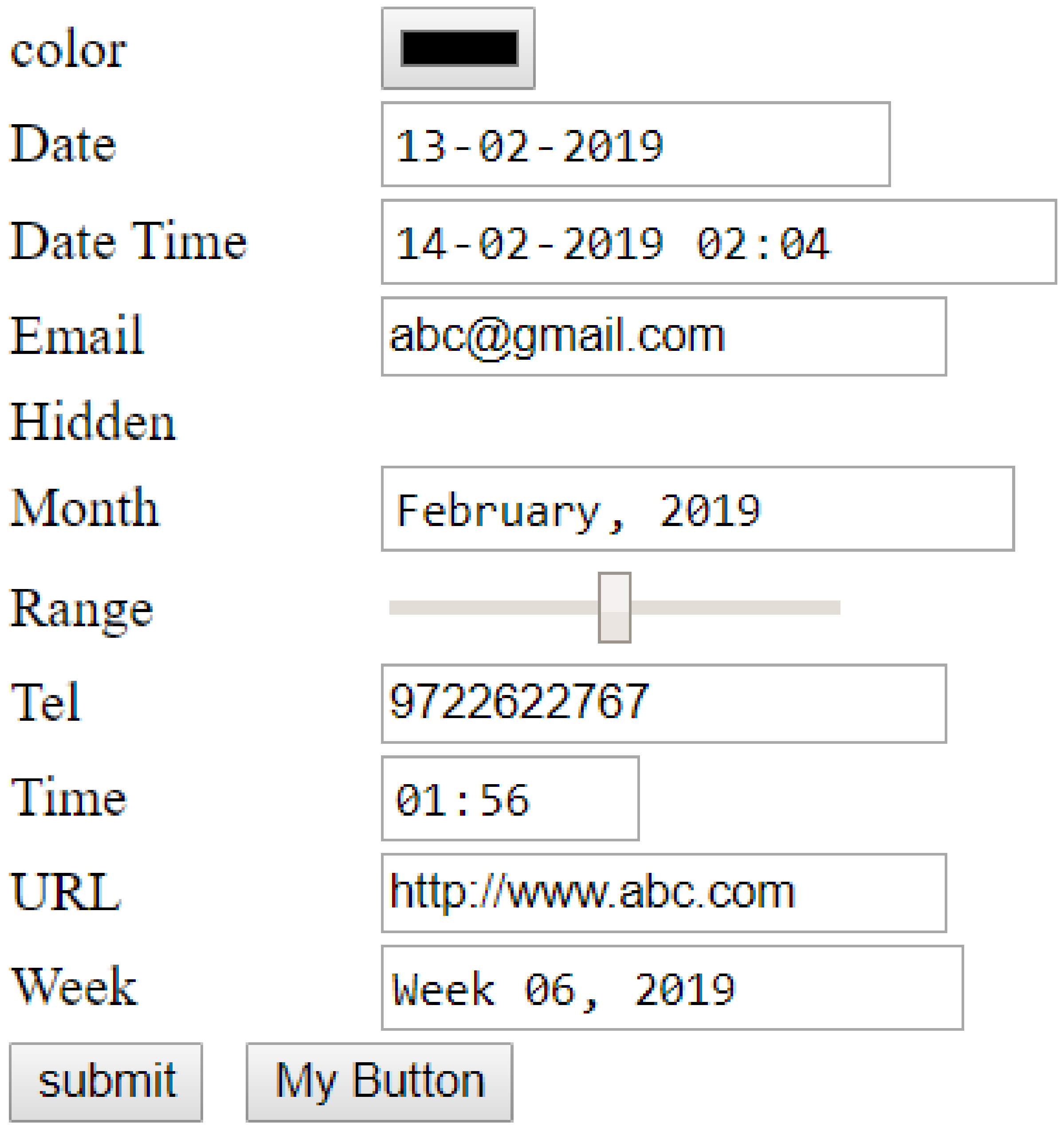
<input type="image">



```
<form action="login.php">
  Username: <input type="text" name="uname"><br/>
Password: <input type="password" name="pwd"><br/>
  <input type="image" src="img_submit.gif" alt="Submit" width="48" height="48">
</form>
```



color	■
Date	dd-mm-yyyy
Date Time	dd-mm-yyyy --:--
Email	
Hidden	
Month	---------, ----
Range	
Tel	
Time	--:--
URL	
Week	Week --, ----



color	■
Date	13-02-2019
Date Time	14-02-2019 02:04
Email	abc@gmail.com
Hidden	
Month	February, 2019
Range	
Tel	9722622767
Time	01:56
URL	http://www.abc.com
Week	Week 06, 2019

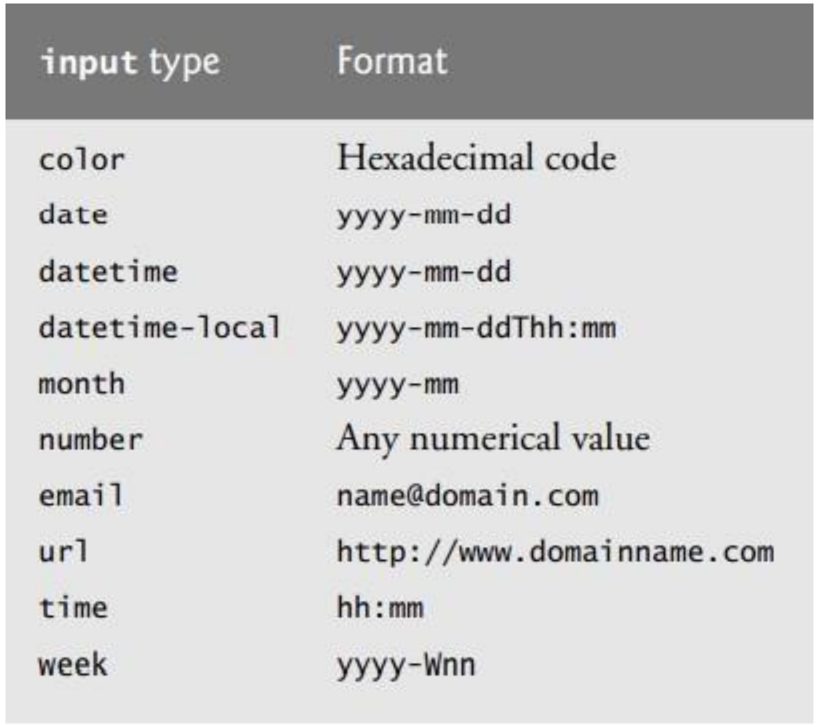
```
<html><body><form action="">
<table align="center" width="400">
	<tr><td align="left">color</td><td><input type="color"> </td></tr>
	<tr><td align="left">Date</td><td> <input type="date"></td></tr>
	<tr><td align="left">Date Time</td><td><input type="datetime-
local"></td></tr>
	<tr><td align="left">Email</td><td> <input type="email" multiple></td></tr>
	<tr><td align="left">Hidden</td><td><input type="hidden"></td></tr>
	<tr><td align="left">Month</td><td> <input type="month"></td></tr>
	<tr><td align="left">Range</td><td> <input type="range"></td></tr>
	<tr><td align="left">Tel</td><td> <input type="tel"></td></tr>
	<tr><td align="left">Time</td><td> <input type="time"></td></tr>
	<tr><td align="left">URL</td><td> <input type="url"></td></tr>
	<tr><td align="left">Week</td><td> <input type="week"></td></tr>
	<tr><td colspan="2"> <input type="submit" value="submit"> &nbsp;&nbsp;
			<input type="button" value="My Button">
	</td></tr>
</table>
</form></body></html>
```

HTML input types-autofocus Attribute

- The autofocus attribute an optional attribute that can be used in only one input element on a form—automatically gives the focus to the input element,
- allowing the user to begin typing in that element immediately.
- You do not need to include autofocus in your forms.

HTML input types-Validation

- The new HTML 5 input types are self validating on the client side, eliminating the need to add complicated JavaScript code to your web pages to validate user input, reducing the amount of invalid data submitted and consequently reducing Internet traffic between the server and the client to correct invalid input.
- The server should still validate all user input.
- When a user enters data into a form then submits the form, the browser immediately checks the self-validating elements to ensure that the data is correct.



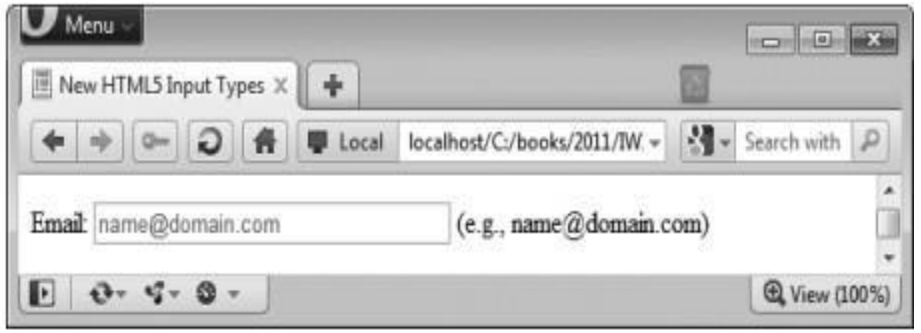
input type	Format
color	Hexadecimal code
date	yyyy-mm-dd
datetime	yyyy-mm-dd
datetime-local	yyyy-mm-ddThh:mm
month	yyyy-mm
number	Any numerical value
email	name@domain.com
url	http://www.domainname.com
time	hh:mm
week	yyyy-Wnn

Self-validating input types.

HTML input -placeholder attribute

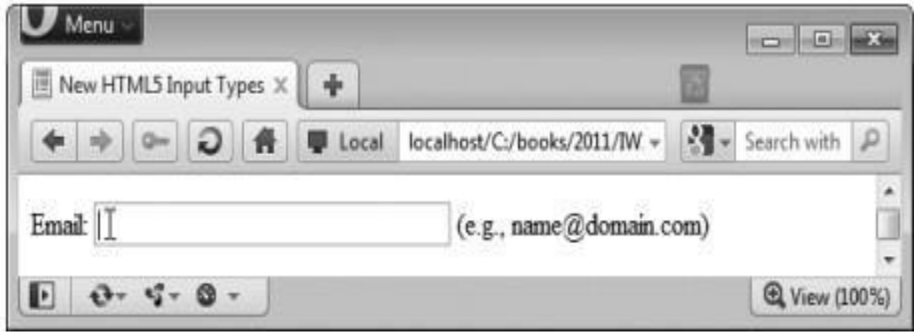
- The placeholder attribute allows you to place temporary text in a text field.
- placeholder text is light gray and provides an example of the text and/or text format the user should enter
- When the focus is placed in the text field (i.e., the cursor is in the text field), the placeholder text disappears—it's not "submitted" when the user clicks the Submit button (unless the user types the same text

a) Text field with gray `placeholder` text



Email: name@domain.com (e.g., name@domain.com)

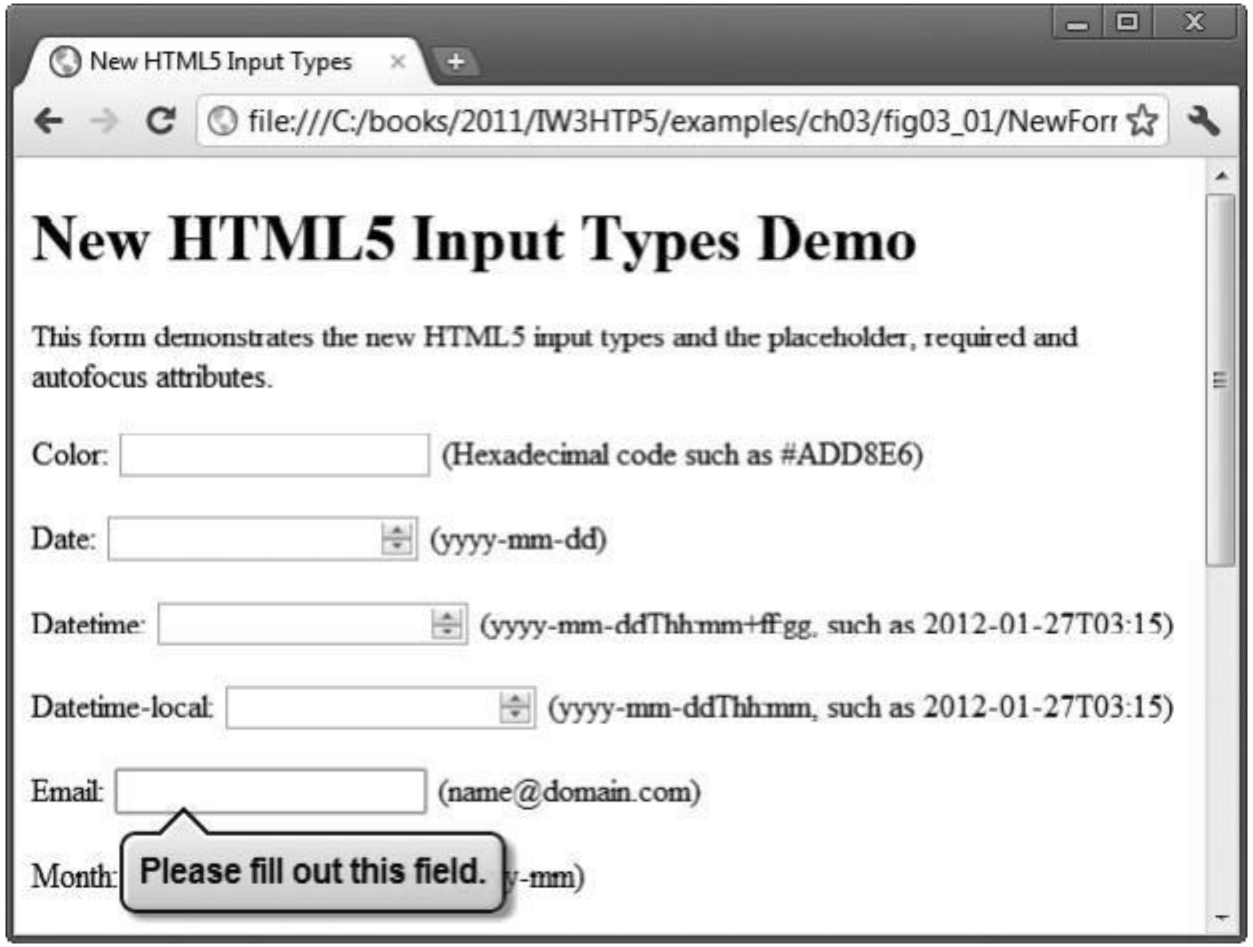
b) `placeholder` text disappears when the text field gets the focus



Email: (e.g., name@domain.com)

HTML input -required attribute

- required Attribute forces the user to enter a value before submitting the form.
- You can add required to any of the input types.
- For example, if the user fails to enter an e-mail address and clicks the Submit button, a callout pointing to the empty element appears, asking the user to enter the information


New HTML5 Input Types
file:///C:/books/2011/IW3HTP5/examples/ch03/fig03_01/NewForr

New HTML5 Input Types Demo

This form demonstrates the new HTML5 input types and the placeholder, required and autofocus attributes.

Color: (Hexadecimal code such as #ADD8E6)

Date: (yyyy-mm-dd)

Datetime: (yyyy-mm-ddThh:mm+ff:gg, such as 2012-01-27T03:15)

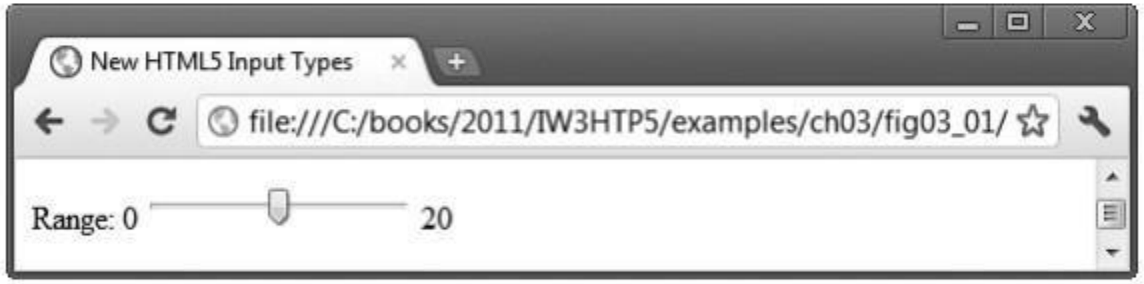
Datetime-local: (yyyy-mm-ddThh:mm, such as 2012-01-27T03:15)

Email: (name@domain.com)

Please fill out this field.

Month: y-mm)

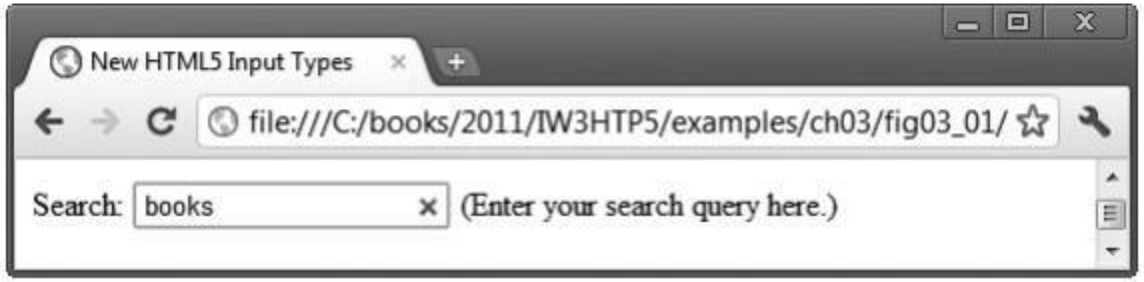
HTML input


New HTML5 Input Types

file:///C:/books/2011/IW3HTP5/examples/ch03/fig03_01/

Range: 0 20

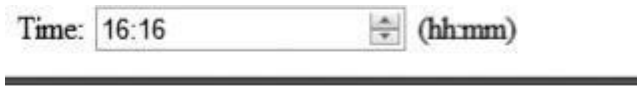
| range slider with a value attribute of 10 as rendered in Chrome.


New HTML5 Input Types

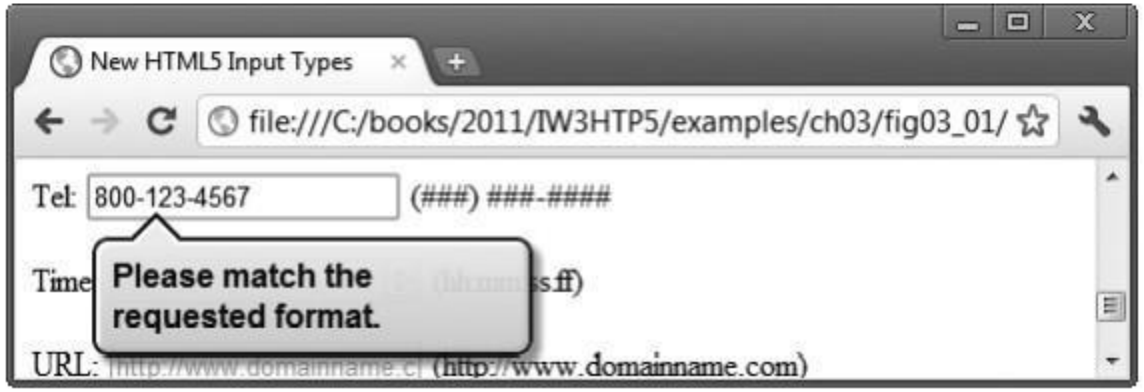
file:///C:/books/2011/IW3HTP5/examples/ch03/fig03_01/

Search: books (Enter your search query here.)

| Entering a search query in Chrome.


Time: 16:16 (hh:mm)

| time input as rendered in Chrome.


New HTML5 Input Types

file:///C:/books/2011/IW3HTP5/examples/ch03/fig03_01/

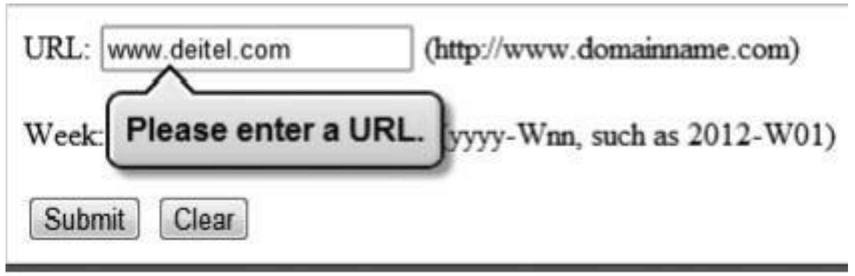
Tel: 800-123-4567 (###) ###-####

Time (hh:mm:ss.ff)

Please match the requested format.

URL: http://www.domainname.c (http://www.domainname.com)

| Validating a phone number using the pattern attribute in the tel input type.


URL: www.deitel.com (http://www.domainname.com)

Please enter a URL.

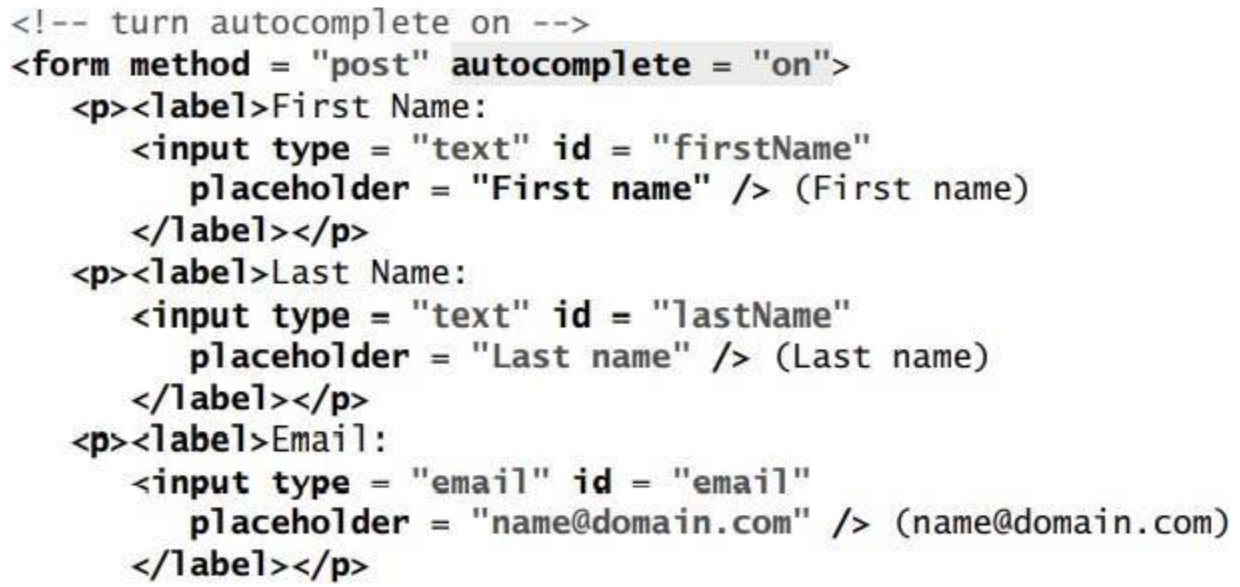
Week: (yyyy-Wnn, such as 2012-W01)

Submit Clear

Validating a URL in Chrome.

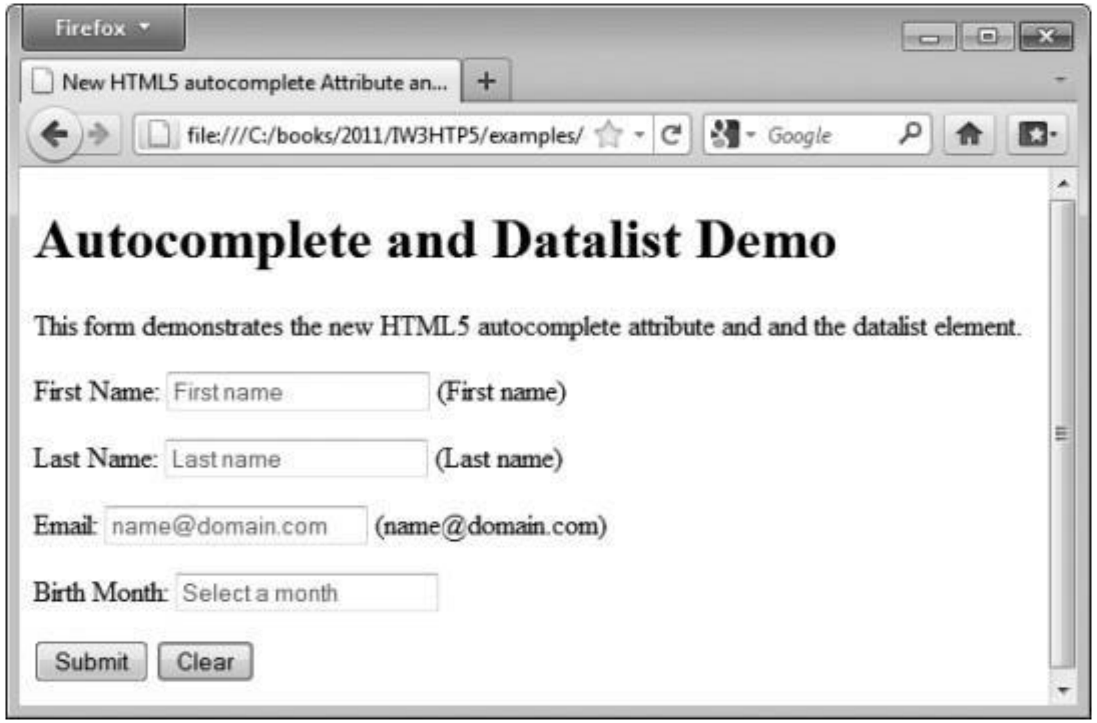
input and data list Elements and autocomplete Attribute

- The autocomplete attribute can be used on input types to automatically fill in the user's information based on previous input—such as name, address or e-mail.
- You can enable autocomplete for an entire form or just for specific elements.
- For example, an online order form might set automcomplete = "on" for the name and address inputs and set autocomplete = "off" for the credit card and password inputs for security purposes.



```
<!-- turn autocomplete on -->
<form method = "post" autocomplete = "on">
   <p><label>First Name:
      <input type = "text" id = "firstName"
         placeholder = "First name" /> (First name)
      </label></p>
   <p><label>Last Name:
      <input type = "text" id = "lastName"
         placeholder = "Last name" /> (Last name)
      </label></p>
   <p><label>Email:
      <input type = "email" id = "email"
         placeholder = "name@domain.com" /> (name@domain.com)
      </label></p>
```

input and datalist Elements and autocomplete Attribute



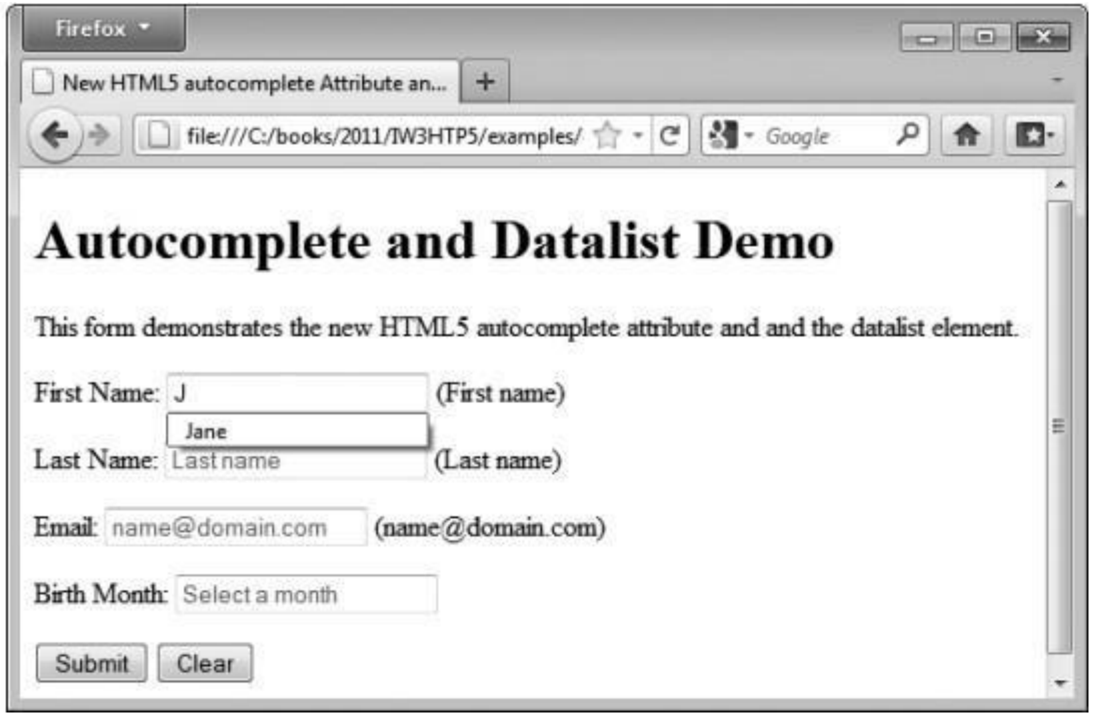
Firefox
New HTML5 autocomplete Attribute an...
file:///C:/books/2011/IW3HTP5/examples/

Autocomplete and Datalist Demo

This form demonstrates the new HTML5 autocomplete attribute and and the datalist element.

First Name:	First name	(First name)
Last Name:	Last name	(Last name)
Email:	name@domain.com	(name@domain.com)
Birth Month:	Select a month	

b) `autocomplete` automatically fills in the data when the user returns to a form submitted previously and begins typing in the **First Name** `input` element; clicking `Jane` inserts that value in the `input`



Firefox
New HTML5 autocomplete Attribute an...
file:///C:/books/2011/IW3HTP5/examples/

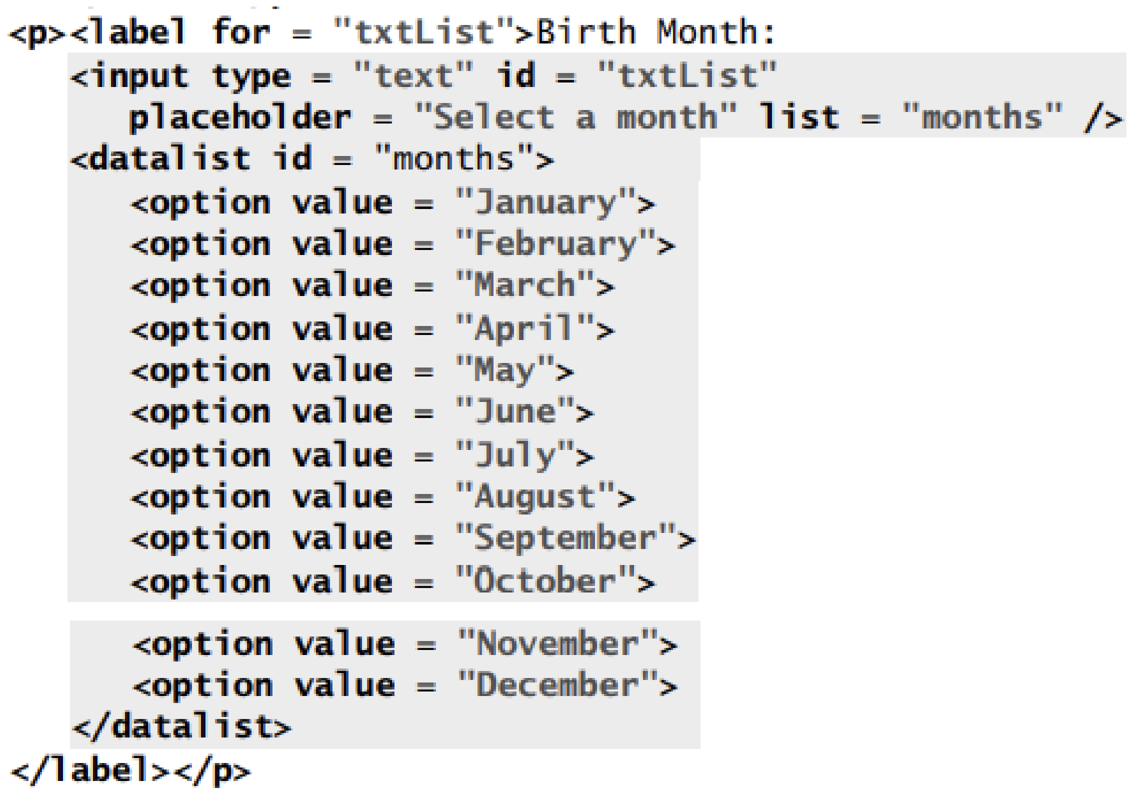
Autocomplete and Datalist Demo

This form demonstrates the new HTML5 autocomplete attribute and and the datalist element.

First Name:	J	(First name)
	Jane	
Last Name:	Last name	(Last name)
Email:	name@domain.com	(name@domain.com)
Birth Month:	Select a month	

datalist Elements

- The datalist element provides input options for a text input element.
- Using Opera, when the user clicks in the text field, a drop-down list of the months of the year appears.
- If the user types "M" in the text field, the list on months is narrowed to March and May. When using Firefox, the drop-down list of months appears only after the user begins typing in the text field

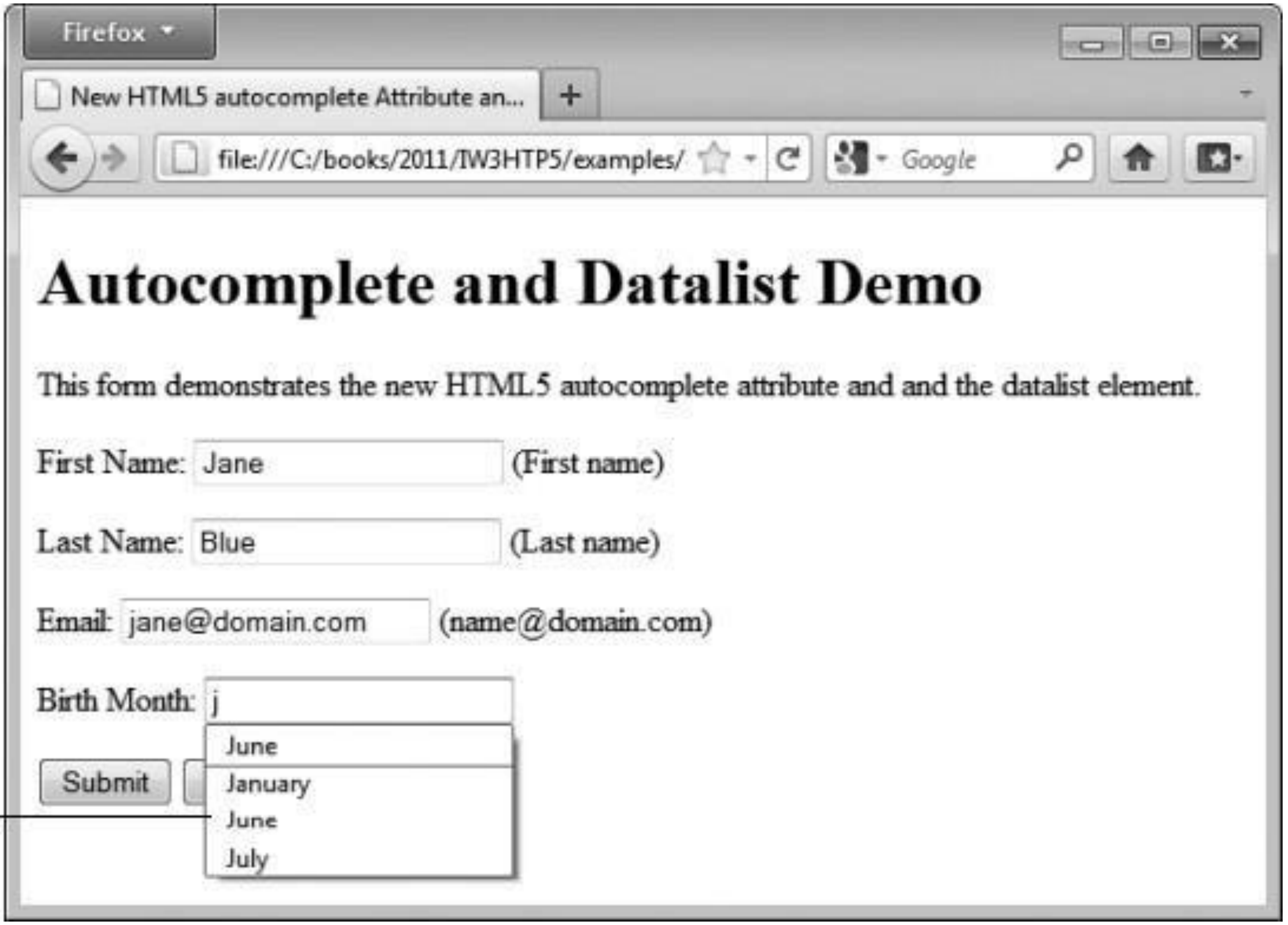


```
<p><label for = "txtList">Birth Month:
   <input type = "text" id = "txtList"
      placeholder = "Select a month" list = "months" />
   <datalist id = "months">
      <option value = "January">
      <option value = "February">
      <option value = "March">
      <option value = "April">
      <option value = "May">
      <option value = "June">
      <option value = "July">
      <option value = "August">
      <option value = "September">
      <option value = "October">
      <option value = "November">
      <option value = "December">
   </datalist>
</label></p>
```

datalist Elements

c) `autocomplete` with a `datalist` showing the previously entered value (`June`) followed by all items that match what the user has typed so far; clicking an item in the `autocomplete` list inserts that value in the `input`

`datalist` values filtered by what's been typed so far


Firefox
New HTML5 autocomplete Attribute an...
file:///C:/books/2011/IW3HTP5/examples/
Google

Autocomplete and Datalist Demo

This form demonstrates the new HTML5 autocomplete attribute and and the datalist element.

First Name: Jane (First name)

Last Name: Blue (Last name)

Email: jane@domain.com (name@domain.com)

Birth Month: j

June

January

June

July

Submit

Page-Structure Elements

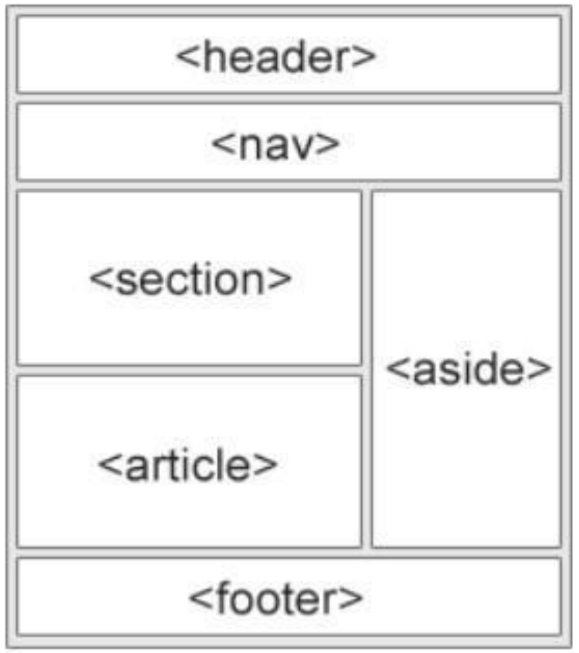
Page-Structure Elements

- HTML5 introduces several new page-structure elements that meaningfully identify areas of the page as headers, footers, articles, navigation areas, asides, figures.
- A semantic element clearly describes its meaning to both the browser and the developer.
- Examples of non-semantic elements: <div> and <span> - Tells nothing about its content.
- Examples of semantic elements: <form>, <table>, and <article> - Clearly defines its content.
- Many web sites contain HTML code like: <div id="nav"> <div class="header"> <div id="footer"> to indicate navigation, header, and footer.
- In HTML there are some semantic elements that can be used to define different parts of a web page

Page-Structure Elements

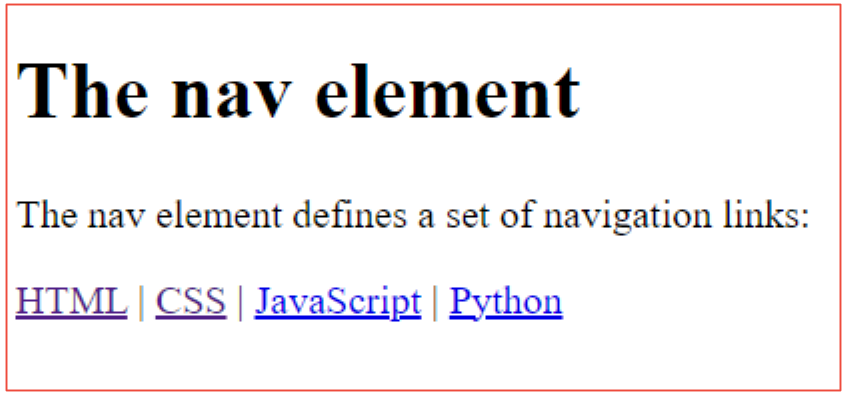
- header Element :creates a header for this page that contains both text and graphics. The header element can be used multiple times on a page and can include
- time Element:The time element which does not need to be enclosed in a header, enables you to identify a date, a time or both.
- nav Element:The nav element groups navigation links.

- <article>
- <aside>
- <details>
- <figcaption>
- <figure>
- <footer>
- <header>
- <main>
- <mark>
- <nav>
- <section>
- <summary>
- <time>



Page-Structure Elements

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

<h1>The nav element</h1>

<p>The nav element defines a set of navigation links:</p>

<nav>
<a href="/html/">HTML</a> |
<a href="/css/">CSS</a> |
<a href="/js/">JavaScript</a> |
<a href="/python/">Python</a>
</nav>

</body>
</html>
```

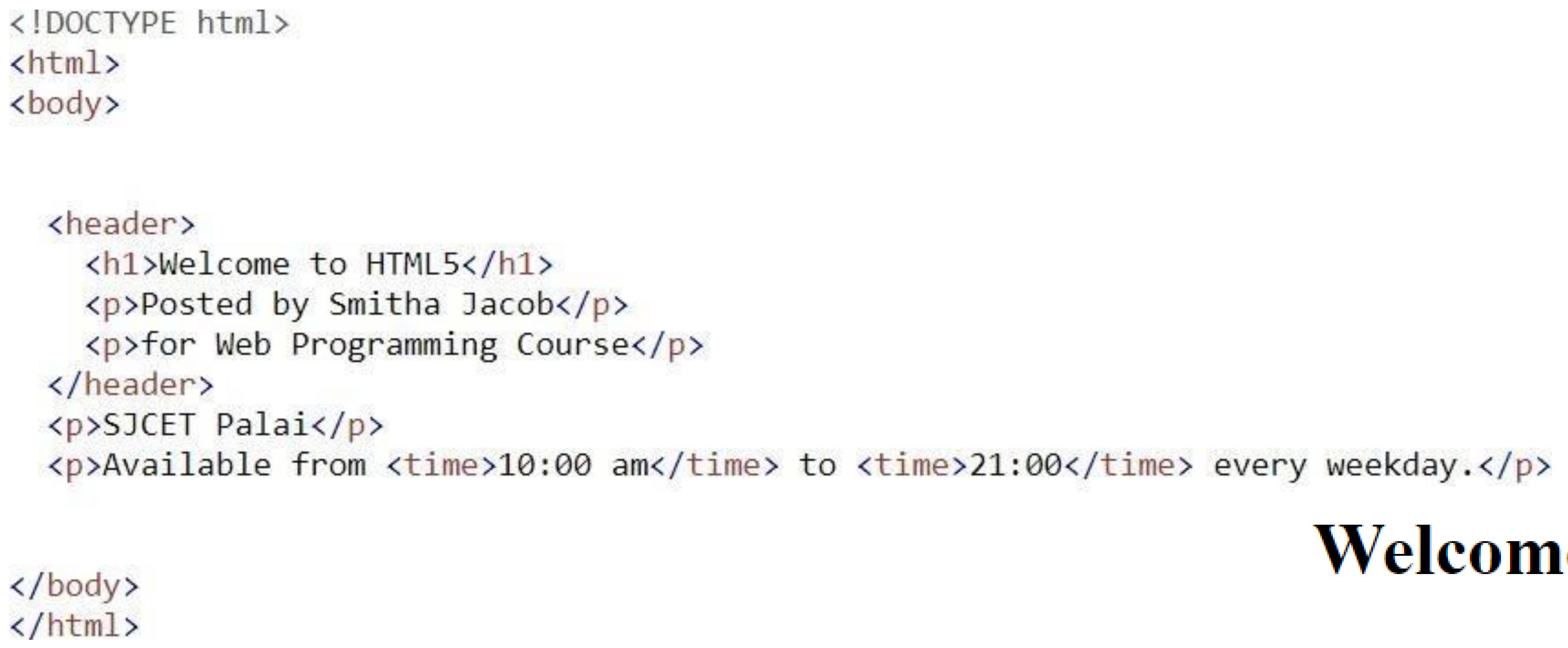


The nav element

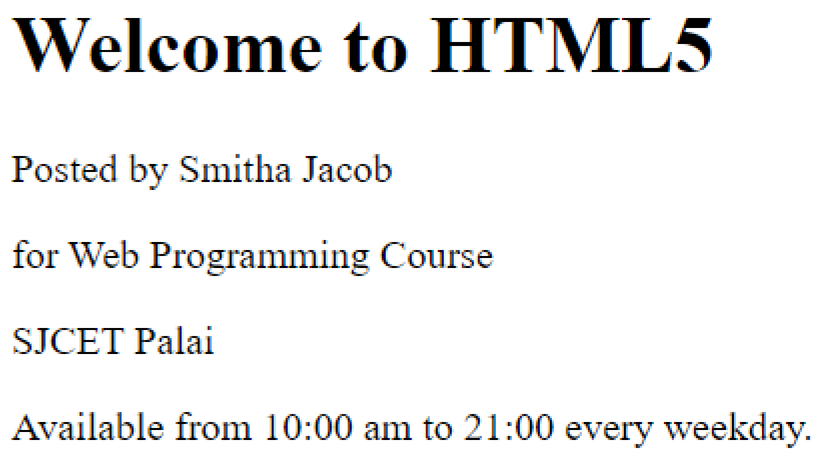
The nav element defines a set of navigation links:

HTML | CSS | JavaScript | Python

Page-Structure Elements



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


  <header>
    <h1>Welcome to HTML5</h1>
    <p>Posted by Smitha Jacob</p>
    <p>for Web Programming Course</p>
  </header>
  <p>SJCET Palai</p>
  <p>Available from <time>10:00 am</time> to <time>21:00</time> every weekday.</p>


</body>
</html>
```



```
Welcome to HTML5

Posted by Smitha Jacob

for Web Programming Course

SJCET Palai

Available from 10:00 am to 21:00 every weekday.
```

Page-Structure Elements

- figure Element and figcaption Element
- The figure element describes a figure (such as an image, chart or table) in the document so that it could be moved to the side of the page or to another page .The figure element does not include any styling, but you can style the element using CSS.
- The figcaption element provides a caption for the image in the figure element.
- The article element - describes standalone content that could potentially be used or distributed elsewhere, such as a news article, forum post or blog entry. You can nest article elements.
- summary Element and details Element-The summary element displays a right-pointing arrow next to a summary or caption when the document is rendered in a browser . When clicked, the arrow points downward and reveals the content in the details element .

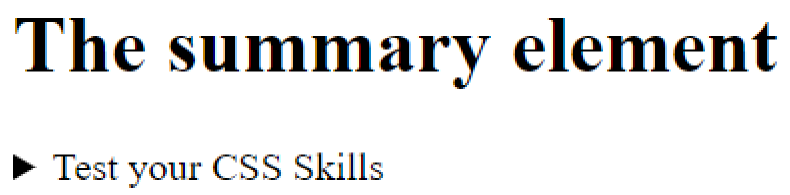
Page-Structure Elements

```
<figure>
  <img src="/macaque.jpg" alt="Macaque in the trees">
  <figcaption>A cheeky macaque, Lower Kintaganban
River, Borneo. Original by <a
href="http://www.flickr.com/photos/rclark/">Richard
Clark</a></figcaption>
</figure>
```

A cheeky macaque, Lower Kintaganban River, Borneo. Original by Richard Clark

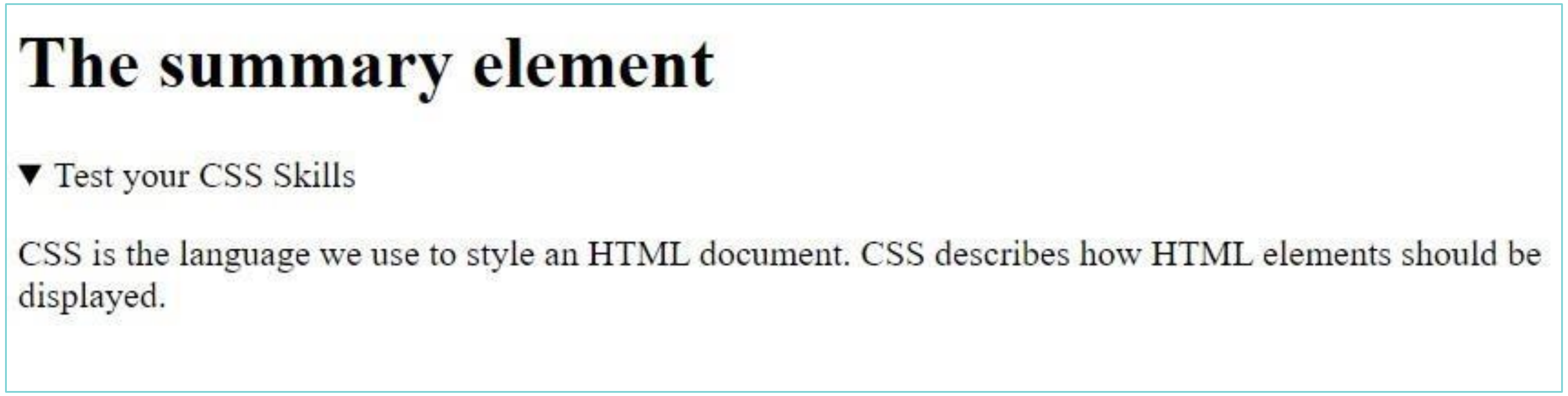
Page-Structure Elements

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

<h1>The summary element</h1>

<details>
  <summary>Test your CSS Skills</summary>
  <p>CSS is the language we use to style an HTML document.

CSS describes how HTML elements should be displayed.</p>
</details>

</body>
</html>
```



The summary element

▶ Test your CSS Skills



The summary element

▼ Test your CSS Skills

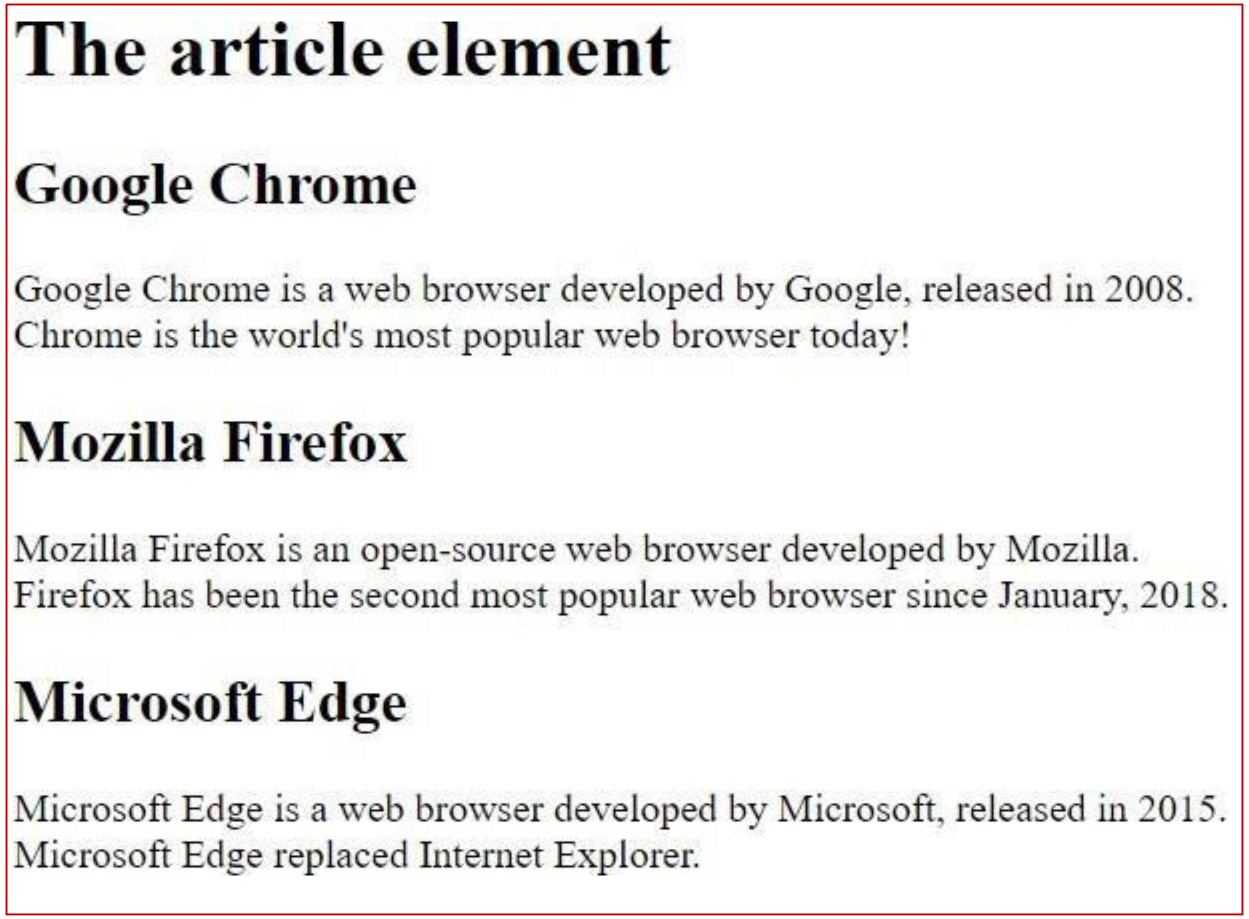
CSS is the language we use to style an HTML document. CSS describes how HTML elements should be displayed.

Page-Structure Elements

- The section element describes a section of a document, usually with a heading for each section—these elements can be nested. For example, you could have a section element for a book, then nested sections for each chapter name in the book.
- aside Element-The aside element describes content that’s related to the surrounding content (such as an article) but is somewhat separate from the flow of the text
- footer Element-The footer element describes a footer—content that usually appears at the bottom of the content or section element.
- Article element-The <article> tag specifies independent, self-contained content.An article should make sense on its own and it should be possible to distribute it independently from the rest of the site.

Page-Structure Elements

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

<h1>The article element</h1>

<article>
  <h2>Google Chrome</h2>
  <p>Google Chrome is a web browser developed by Google,
released in 2008.<br/> Chrome is the world's most popular
web browser today!</p>
</article>

<article>
  <h2>Mozilla Firefox</h2>
  <p>Mozilla Firefox is an open-source web browser
developed by Mozilla.<br/> Firefox has been the second
most popular web browser since January, 2018.</p>
</article>

<article>
  <h2>Microsoft Edge</h2>
  <p>Microsoft Edge is a web browser developed by
Microsoft, released in 2015.<br/> Microsoft Edge replaced
Internet Explorer.</p>
</article>

</body>
</html>
```



The article element

Google Chrome

Google Chrome is a web browser developed by Google, released in 2008.
Chrome is the world's most popular web browser today!

Mozilla Firefox

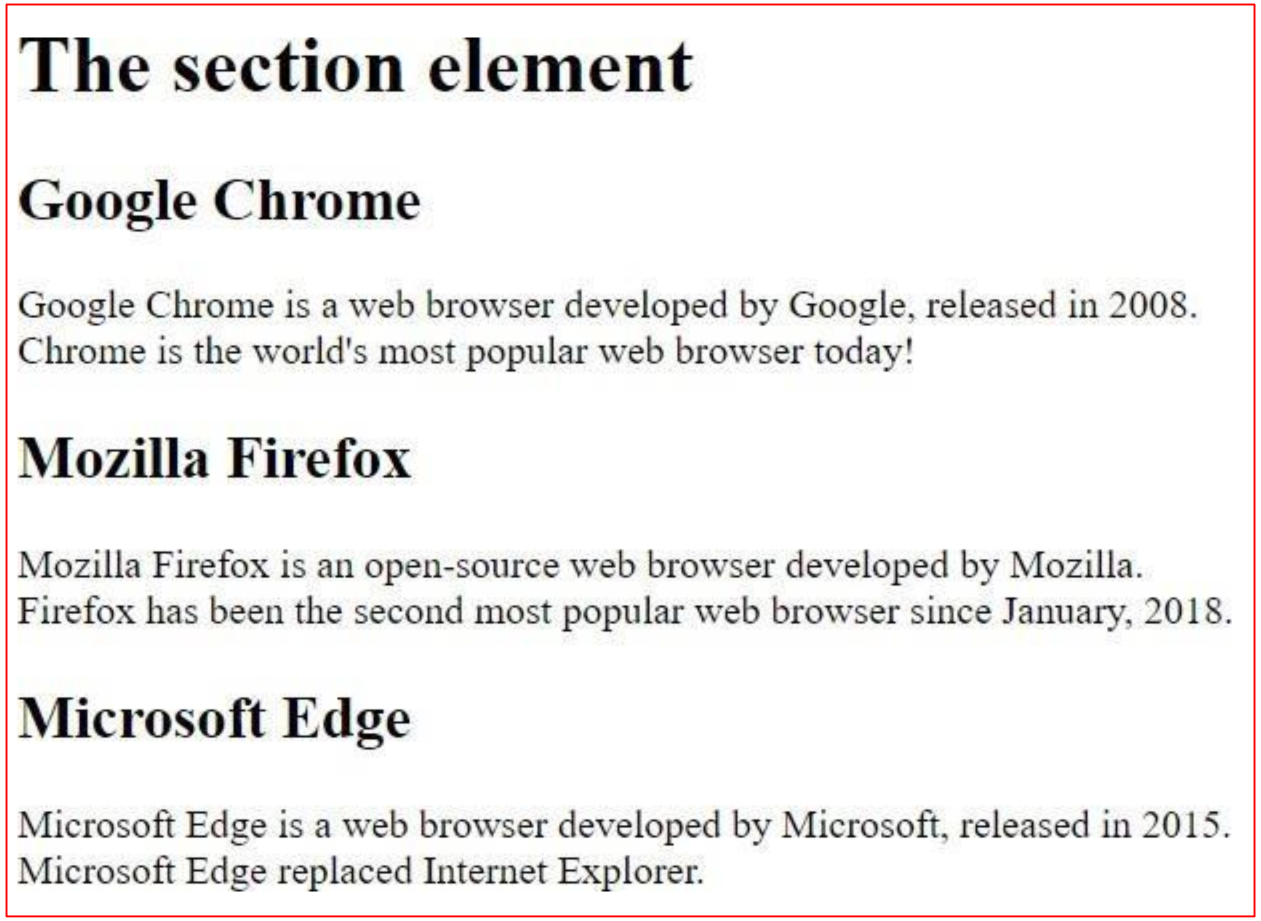
Mozilla Firefox is an open-source web browser developed by Mozilla.
Firefox has been the second most popular web browser since January, 2018.

Microsoft Edge

Microsoft Edge is a web browser developed by Microsoft, released in 2015.
Microsoft Edge replaced Internet Explorer.

Page-Structure Elements

```
<!DOCTYPE html>
<html>
<body>
<h1>The section element</h1>
<section>
  <h2>Google Chrome</h2>
  <p>Google Chrome is a web browser developed by
Google, released in 2008.<br/> Chrome is the world's
most popular web browser today!</p>
</section>
<section>
  <h2>Mozilla Firefox</h2>
  <p>Mozilla Firefox is an open-source web browser
developed by Mozilla.<br/> Firefox has been the
second most popular web browser since January, 2018.
</p>
</section>
<section>
  <h2>Microsoft Edge</h2>
  <p>Microsoft Edge is a web browser developed by
Microsoft, released in 2015.<br/> Microsoft Edge
replaced Internet Explorer.</p>
</section>
</body></html>
```



The section element

Google Chrome

Google Chrome is a web browser developed by Google, released in 2008.
Chrome is the world's most popular web browser today!

Mozilla Firefox

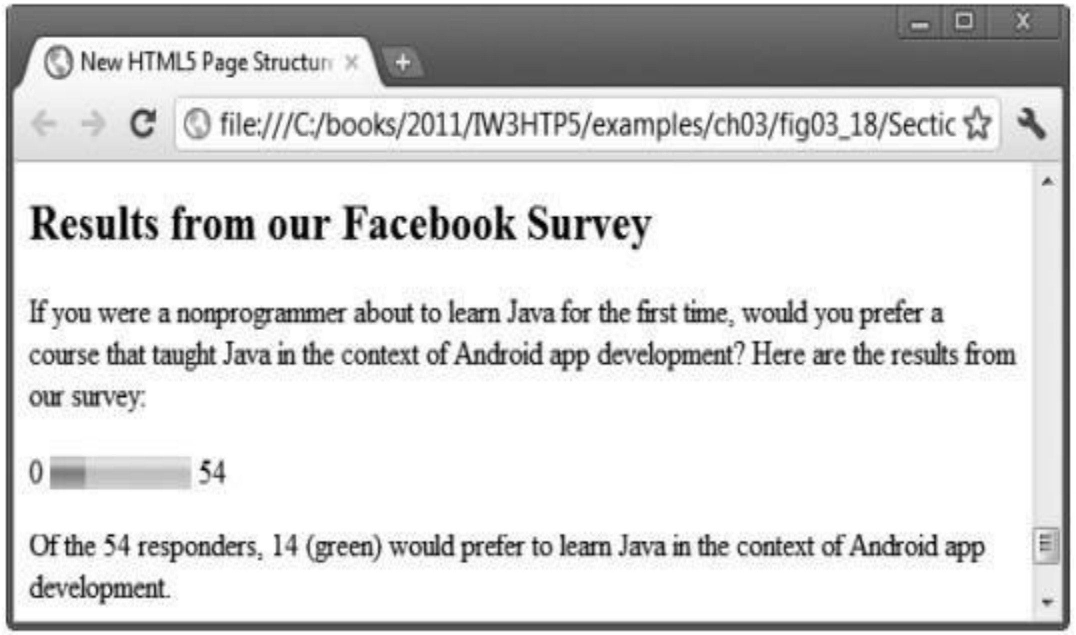
Mozilla Firefox is an open-source web browser developed by Mozilla.
Firefox has been the second most popular web browser since January, 2018.

Microsoft Edge

Microsoft Edge is a web browser developed by Microsoft, released in 2015.
Microsoft Edge replaced Internet Explorer.

Page-Structure Elements

- meter Element-The meter element renders a visual representation of a measure within a range.
- In this example, The min attribute is "0" and a max attribute is "54" —indicating the total number of responses to our survey. The value attribute is "14", representing the total number of people who responded “yes” to our survey question.



Results from our Facebook Survey

If you were a nonprogrammer about to learn Java for the first time, would you prefer a course that taught Java in the context of Android app development? Here are the results from our survey:

0 54

Of the 54 responders, 14 (green) would prefer to learn Java in the context of Android app development.

Chrome rendering the `meter` element.

Page-Structure Elements

- Text-Level Semantics: mark Element and wbr Element
- The mark element highlights the text that's enclosed in the element.
- The wbr element- **Word Break Opportunity** indicates the appropriate place to break a word when the text wraps to multiple lines.
- You might use wbr to prevent a word from breaking in an awkward place.
- When a word is too long, the browser might break it at the wrong place. You can use the <wbr> element to add word break opportunities.

```
<p>To learn AJAX, you must be familiar with the XML<wbr>Http<wbr>Request Object.</p>
```

Multimedia-Audio and video tags

Multimedia elements

- The <audio> tag defines sound, such as music or other audio streams.
- audio element, is used to embed audio into a web page. We specify an id for the element, so that we can programmatically control when the audio clip plays
- The <video> tag specifies video, such as a movie clip or other video streams.
- Not all browsers support the same audio file formats
- 3 commonly supported file formats for the <audio> element: MP3, WAV, and OGG:
- 3 commonly supported video formats for the <video> element: MP4, WebM, and Ogg:
- MP4 = MPEG 4 files with H264 video codec and AAC audio codec
- WebM = WebM files with VP8 video codec and Vorbis audio codec
- Ogg = Ogg files with Theora video codec and Vorbis audio codec

Multimedia elements-audio tag

- Prior to HTML5, a plug-in was required to play sound while a document was being displayed
- Audio information is coded into digital streams with encoding algorithms called audio codecs
- – e.g., MP3, Vorbis,wav
- A *codec* (a blend word derived from "***co***der-***dec***oder") is a program, algorithm, or device that encodes or decodes a data stream.
- A given codec knows how to handle a specific encoding or compression technology.
- Coded audio data is stored in containers.
- Audio container may contain MP3 / Vorbis / Wav codec.
- 3 audio container types
- e.g Ogg, MP3, and Wav (same as extension of file names)

- Vorbis codec is stored in Ogg containers

- MP3 codec is stored in MP3 containers

- Wav codec is stored in Wav containers

Multimedia elements-audio tag

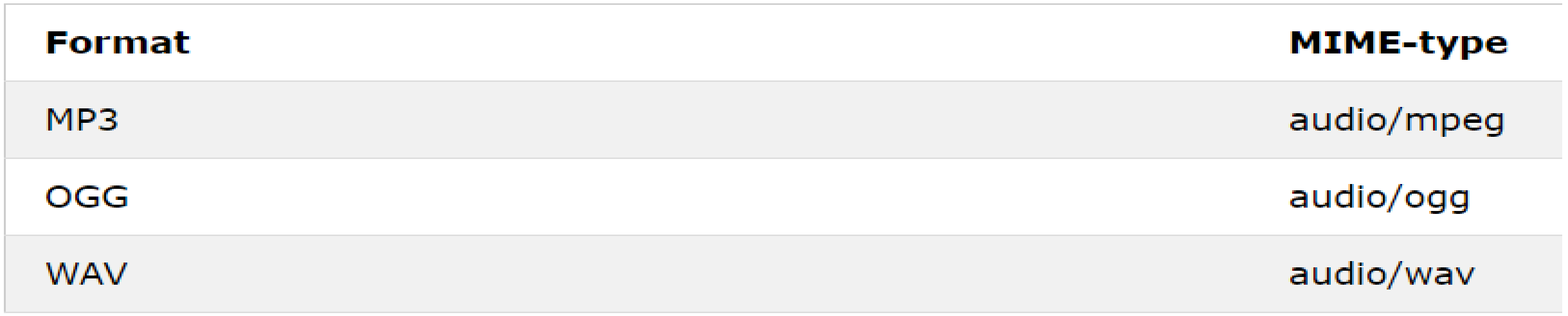
- Ogg→free & open digital container format maintained by Xiph.org foundation.

  →container for Vorbis audio coding format
- wav→wave form audio file format used in windows systems for raw and uncompressed audio

  →Audio file format by Microsoft & IBM for storing audio bit streams ,digital audio on system.
- Mp3→Standard technology audio format for compressing a sound sequence into a very small file.

  →common audio coding format for digital audio.

MIME Types for Audio Formats



Format	MIME-type
MP3	audio/mpeg
OGG	audio/ogg
WAV	audio/wav

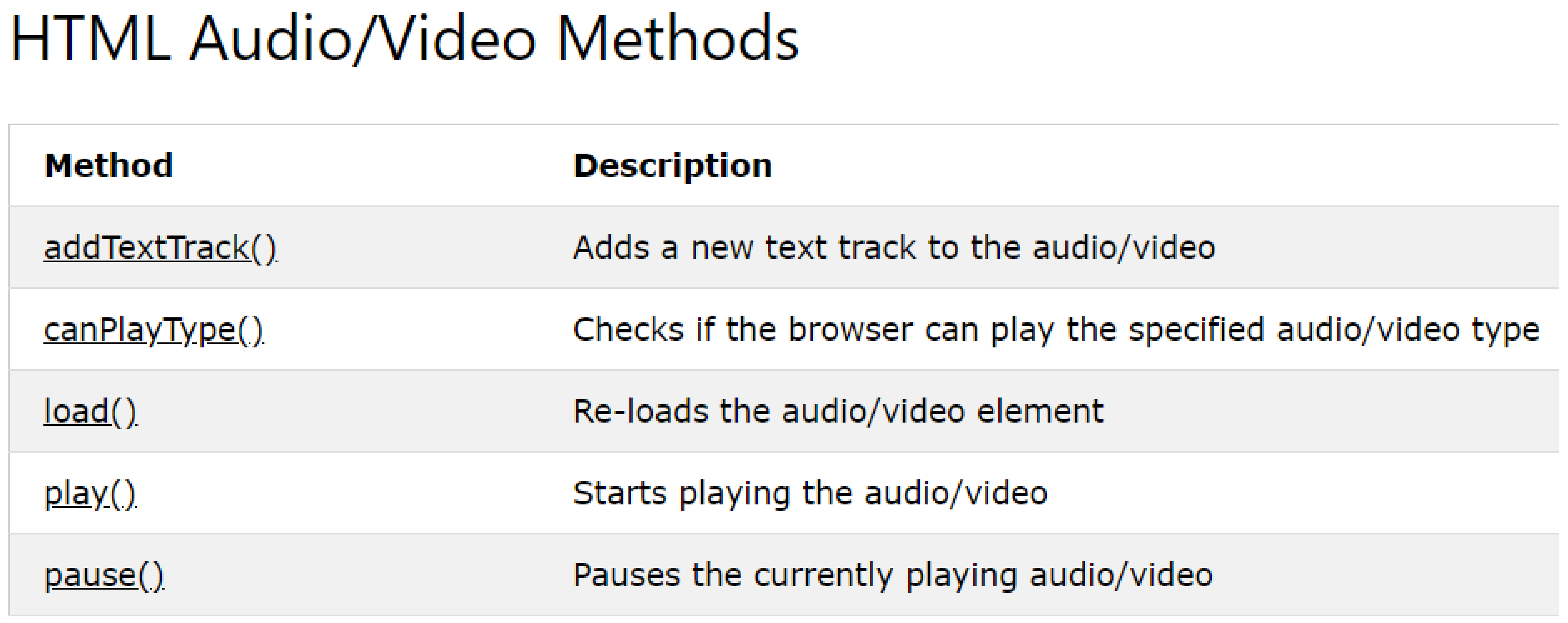
Multimedia elements-audio tag

- General syntax:

```
<audio attributes>
  <source src = "filename1" />
  …
  <source src = "filenamen" />
Your browser does not support the audio   element
</audio>
```

- Browser chooses the first audio file it can and skips the content; if none, it displays content
- Different browsers have different audio capabilities .Browser chooses the first audio file
- The controls attribute, which is set to controls", creates a start/stop button, a clock, a progress slider, total time of the file, and a volume slider
  - Firefox 3.5+→ogg/vorbis & wav/wav container/codec audio files
  - Chrome 3.0→ogg/vorbis &mp3/mp3 container/codec audio files
  - IE9→wav/wav container/codec audio files
  - Safari→wav/wav container/codec audio files

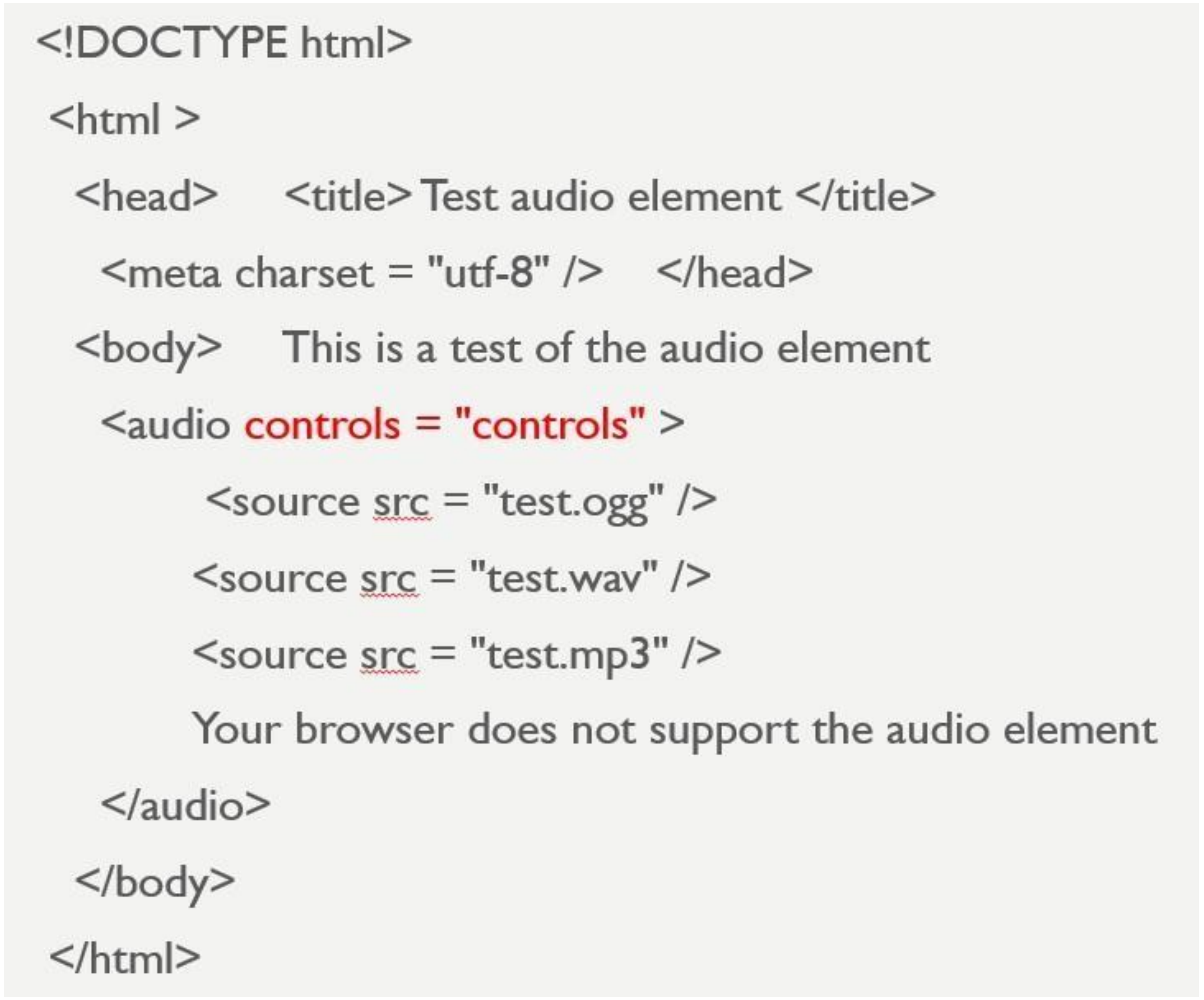
Multimedia elements-audio tag



HTML Audio/Video Methods

Method	Description
addTextTrack()	Adds a new text track to the audio/video
canPlayType()	Checks if the browser can play the specified audio/video type
load()	Re-loads the audio/video element
play()	Starts playing the audio/video
pause()	Pauses the currently playing audio/video

Multimedia elements-audio tag

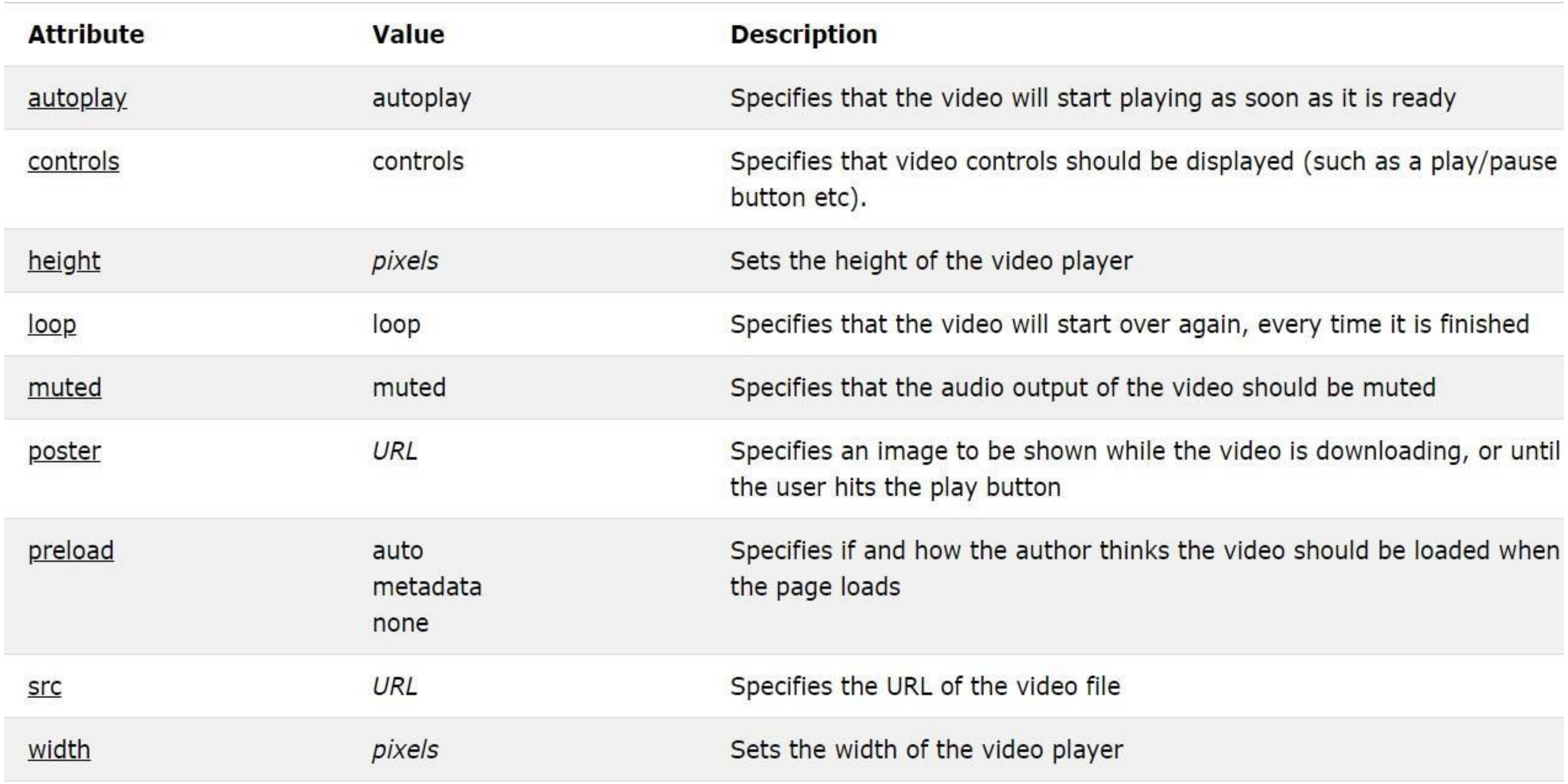


```
<!DOCTYPE html>
 <html >
   <head>     <title> Test audio element </title>
     <meta charset = "utf-8" />     </head>
   <body>     This is a test of the audio element
     <audio controls = "controls" >
              <source src = "test.ogg" />
             <source src = "test.wav" />
             <source src = "test.mp3" />
             Your browser does not support the audio element
     </audio>
   </body>
 </html>
```

Multimedia elements-video tag

- Prior to HTML5, there was no standard way to play video clips while a document was being displayed
- Video codecs are stored in containers
- Video codecs:
  - H.264 (MPEG-4 AVC) – can be stored in MPEG-4
  - Theora – can be stored in any container
  - VP8 can be stored in any container
- Different browsers support different codecs

Multimedia elements-video tag



Attribute	Value	Description
autoplay	autoplay	Specifies that the video will start playing as soon as it is ready
controls	controls	Specifies that video controls should be displayed (such as a play/pause button etc).
height	*pixels*	Sets the height of the video player
loop	loop	Specifies that the video will start over again, every time it is finished
muted	muted	Specifies that the audio output of the video should be muted
poster	*URL*	Specifies an image to be shown while the video is downloading, or until the user hits the play button
preload	auto metadata none	Specifies if and how the author thinks the video should be loaded when the page loads
src	*URL*	Specifies the URL of the video file
width	*pixels*	Sets the width of the video player

Multimedia elements-video tag

```
<!DOCTYPE html>
<!-- testvideo.html test the video element -->
<html>
  <head>
    <meta charset = "UTF-8" />
    <title> test video element </title>
  </head>
  <body>
    This is a test of the video element.....
    <video width = "600" height = "500"
           autoplay = "autoplay"
           controls = "controls"
           preload = "preload">
      <source src="v1.mp4" type="video/mp4"/>
Your browser does not support the video
      element
    </video>
  </body>
</html>
```

University questions

(a) What is the difference between radio buttons and checkboxes when implemented using HTML? Write HTML code to implement a form which has the following elements:
i. A textbox which can accept a maximum of 25 characters
ii. Three radio buttons with valid Label, Names and values
iii. Three check boxes buttons with valid Label, Names and values
iv. A selection list containing four items, two which are always visible v. A submit button clicking on which will prompt the browser to send the form data to the server "http://www..mysite.com/reg.php" using "POST" method and reset button to clear its contents. You can use any text of your choice to label the form elements

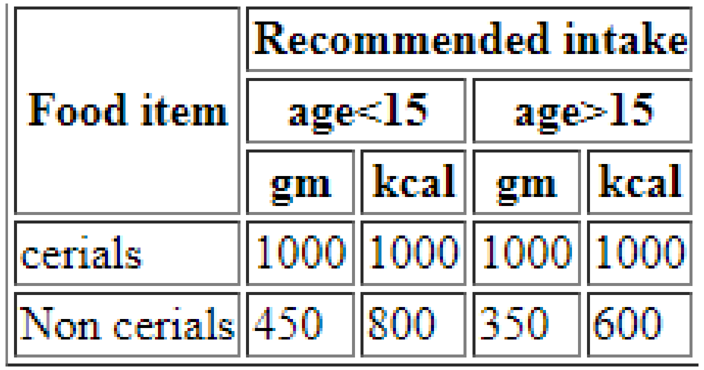
(b)Define WWW. List any two examples of web server & web browser.(web server- Apache,IIS, Sun java Systems)

Differentiate between URL and a domain? Write the syntax of the URL? Rewrite the default URL of your university website by adding a subdomain named 'Research' and a web page named 'FAQ.html'. Also link this URL through the logo of 'kturesearch.png' placed in a web page. The FAQ page should be opened in a new window.

Design a webpage that displays the following table

```
<table border ="border">
  <tr>
    <th rowspan="3">Food item</th>
    <th colspan="4" >Recommended intake</th>
  </tr>
  <tr>
    <th colspan="2">age<15</th>
    <th colspan="2">age>15</th>
  </tr>
  <tr>
    <th>gm</th>
    <th>kcal</th>
    <th>gm</th>
    <th>kcal</th>
  </tr>
```

```
 <tr>

<td>cerials</td>
<td> 1000</td>
<td> 1000</td>
<td> 1000</td>
<td> 1000</td>
  </tr>
  <tr>

<td> Non cerials</td>
<td> 450</td>
<td> 800</td>
<td> 350</td>
<td> 600</td>
  </tr>
</table>
```



Food item	Recommended intake			
	age<15		age>15	
	gm	kcal	gm	kcal
cerials	1000	1000	1000	1000
Non cerials	450	800	350	600

. (a) Write the equivalent HTML code to implement the following in a web page: (i) An image titled “birds.jpg” with a height of 100 pixels and width of 200 pixels. If the image cannot be accessed, a message “No image available” should be displayed (ii) A hyperlink to the URL “www.mysite.com/birds.jpg”. The hyperlink should have the label “Click Here”

Create a static HTML document for your portfolio, which includes the following contents: your name, address, Mobile Number and email address. Also add the details about your college, university, your major and the batch of study. Include a picture of yourself and at least one other image (friend/pet/role model) to the document with a short description about that. Add three paragraphs about your personal history, with links to your social media profile. Also create an ordered list for describing your Skill Set & an unordered list showing your Strengths & Weaknesses.

Code for RESEt & Submit Button

```
<input type="reset" value="Reset">
  <input type="submit" value="Submit">
```

Thank You