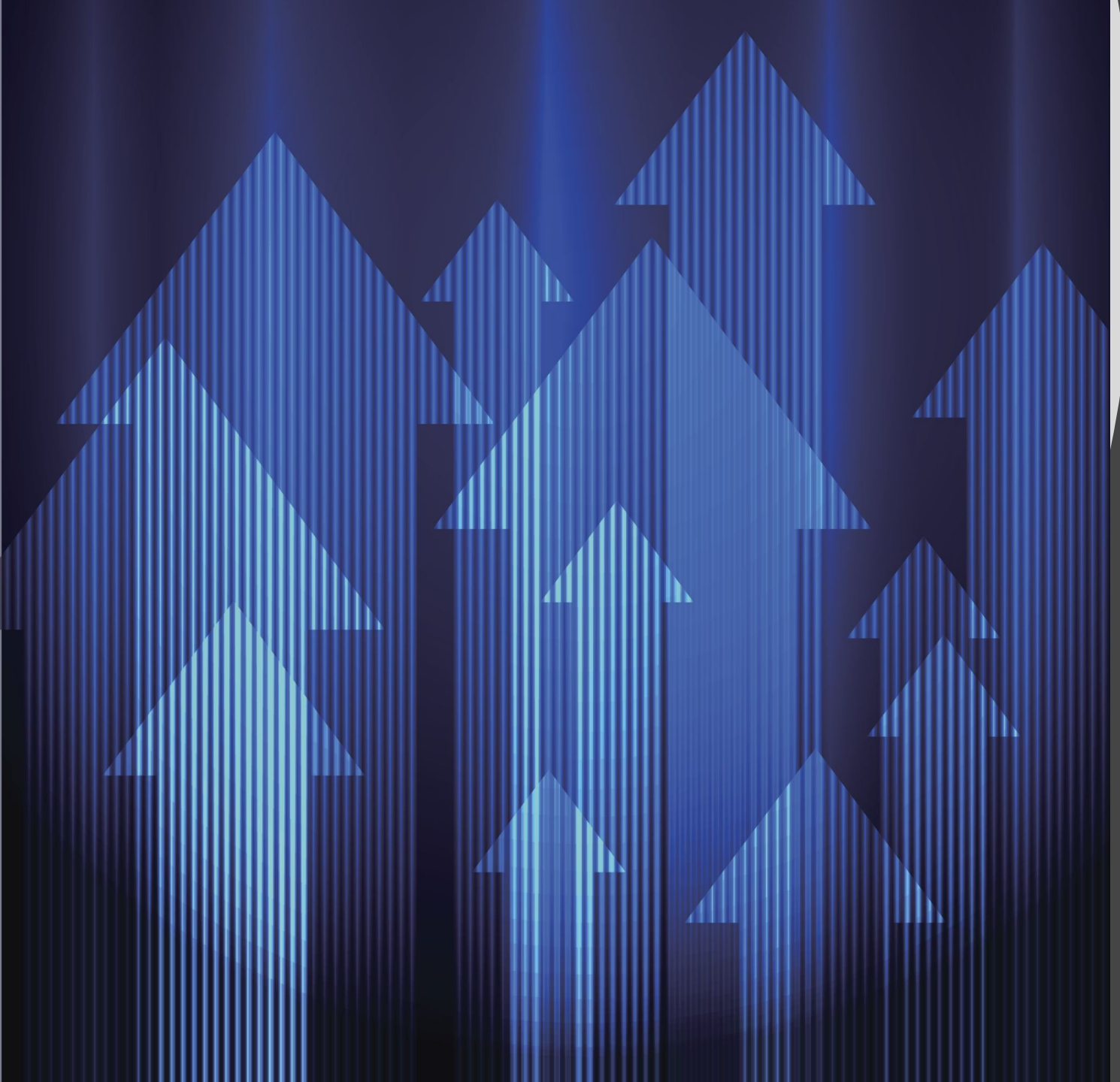
Web Programming

CS-406 Lecture # 02 HTML

TYPES - Static and dynamic

S.No	Static Website	Dynamic Website
1.	Code of the page is displayed to the user.	content is called in by the scripting language from other files or from a database depending on actions taken by the user.
2.	HTML , CSS	PHP, JavaScript, ASP,JSP etc
3.	Flexible and easy designing	Difficult to design
4.	Low Cost	High Cost
5.	Difficult to update the content	Easy to update the content
6.	Scalability is an issue	Easy to pull information from data bases
7.	Non responsive	User Interactive and Responsive



Web Development

Static Websites - HTML

Web Programming

- HTML is a markup language which is used by the browser to manipulate text, images and other content to display it in required format.
- Front End development
- HTML- Define content of the webpage
- CSS- Specify Layout of the webpage
- JavaScript- Program the behavior of webpages, user interactivity

Web Platform

The Web Browser creates the visual web page you see in the browser based on the HTML code

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

<h1>My First Heading</h1>

<p>My first paragraph.</p>

</body>
</html>
```

My First Heading

My first paragraph.

HTML, CSS, JavaScript

Client-side

Web Browser

Web Page (HTML)

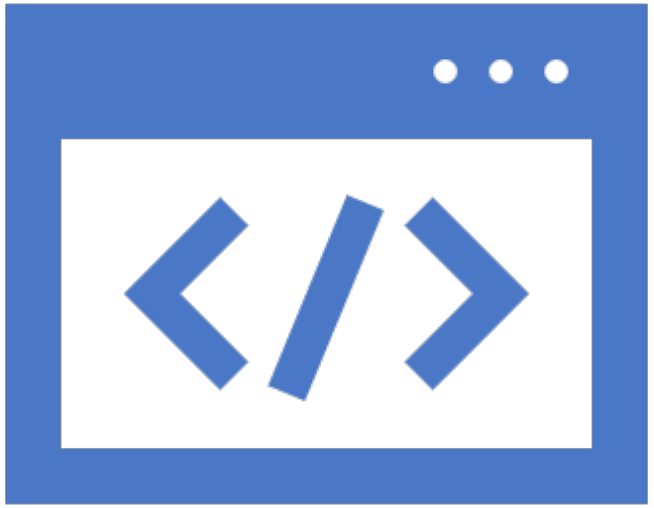
The code runs on the server and converted to HTML before sending to client (Web Browser)

Web Server

ASP.NET, PHP, …

Server-side

Internet Information Services (IIS), Apache, etc.



Hyper**T**ext**M**arkup **L**anguage (HTML)

- The Visual Appearnce of a Web Site
- “Web Browser Language”: All Web Browser understand HTML
- HTML 5 is the latest
- Maintained by W3C - World Wide Web Consortium
- Use tags
- <tagname>content</tagname>

HTML

- HTML is defined as a **markup language**.

- A markup language is simply a way of annotating a document in such a way to make the annotations distinct from the text being annotated.
- It consists of elements and tags that describes details of its structure.

Markup

- **Markup** is a way to indicate *information about the content*
- This “information about content” in HTML is implemented via elements.

Syntax & semantics

Syntax: It refers to the grammatical structure or rules defined for a language.

Semantics: It refers to the meaningfulness of the terms used. How well the name of the tag defines its purpose.

History

- HTML was created by Tim Berners-Lee in 1991.
- The first ever version of HTML was HTML 1.0 but the first standard version was HTML 2.0 which was published in 1999.

HTML VERSION	YEAR
HTML 1.0	1991
HTML 2.0	1995
HTML 3.2	1997
HTML 4.01	1999
XHTML	2000
HTML 5	2014

HTML Editors

- Notepad++
- Adobe Dreamweaver
- NetBeans
- Coffeecup etc

- Extension of the file: .htm

HTML Elements & Tags

<DOCTYPE! html>: This tag is used to tells the HTML version. This currently tells that the version is HTML 5.

<html>: This is called HTML root element and used to wrap all the code.

<head>: Head tag contains metadata, title, page CSS etc. All the HTML elements that can be used inside the <head> element are:

- <style>
- <title>
- <base>
- <noscript>
- <script>
- <meta>
- <title>

<body>: Body tag is used to enclosed all the data which a web page has from texts to links. All of the content that you see rendered in the browser is contained within this element.

Advantages

- It is easy to learn and easy to use.
- It is platform independent.
- Images, video and audio can be added to a web page.
- Hypertext can be added to text.
- It is a markup language.
- It can be integrated with other languages like CSS, JavaScript etc

Disadvantages

- HTML can create only static webpages
- Large amount of code has to be written to create a simple web page.
- Security features are not good.

Types of HTML

- HTML can be divided into three categories:
  - Transitional
  - Strict
  - Frameset
- These types apply to how HTML is used, not necessarily to the selection of tags

Transitional

- **Transitional** is the most common type of HTML.
- It has a flexible syntax, or grammar and spelling component.
- Over the years, transitional HTML has been used without syntax restrictions, and browsers support a **'best effort**' approach to reading the tags.
- If tags are misspelled, the browsers do not correct web developers' errors, and they display the content anyway.
- Browsers do not report HTML errors - they simply display what they can. This is the 'best effort' concept.

Strict

- The **strict type** of HTML is meant to return rules into HTML and make it more reliable.
- For **example**, the strict type requires closing all tags for all opened tags.
- This style of HTML is important on phones, where the processing power may be limited.
- A clean and error-free code helps to load pages faster.

Frameset

- **Frameset** allows web developers to create a mosaic of HTML documents where multiple documents can be connected into a single screen.
- This technique is often used to create a menu system.
- One clicks on a menu item on the left side of the screen, and only the right side of the screen re-loads.
- The menu stays in place.

HTML Semantics

- A semantic element clearly describes its meaning to both the browser and the developer.
- Semantic HTML or semantic markup is HTML that introduces meaning to the web page rather than just presentation.
- Examples of **semantic** elements: <form>, <table>, <header>, <footer>

Presentational HTML

- Only concerned with the presentation of content in web browser.
- Divided into two groups:
  - "Deliberate" presentational elements: (b, i, ...) have no semantic meaning and were deliberately designed/created for presentational purposes.
  - "Abused" presentational elements (table, blockquote, ...) have a semantic meaning, but are abused because of their default appearance in browsers (the blockquote element could be used for indenting text, for example).

HTML 5

There are three main aims to HTML5:

- Specify unambiguously how browsers should deal with invalid markup.
- Provide an open, non-proprietary programming framework (via Javascript) for creating rich web applications.
- Be backwards compatible with the existing web.
- Not every browser support HTML but most of them does while others partially support it.

HTML 5

The most interesting new HTML5 elements are:

- New **semantic elements** like <Header>, <footer>, <article> & <section>.
- New **attributes of form** elements like number, date, time, calendar, and range.
- New **graphic elements**: <svg> and <canvas>.
- New **multimedia elements**: <audio> and <video>

HTML Syntax

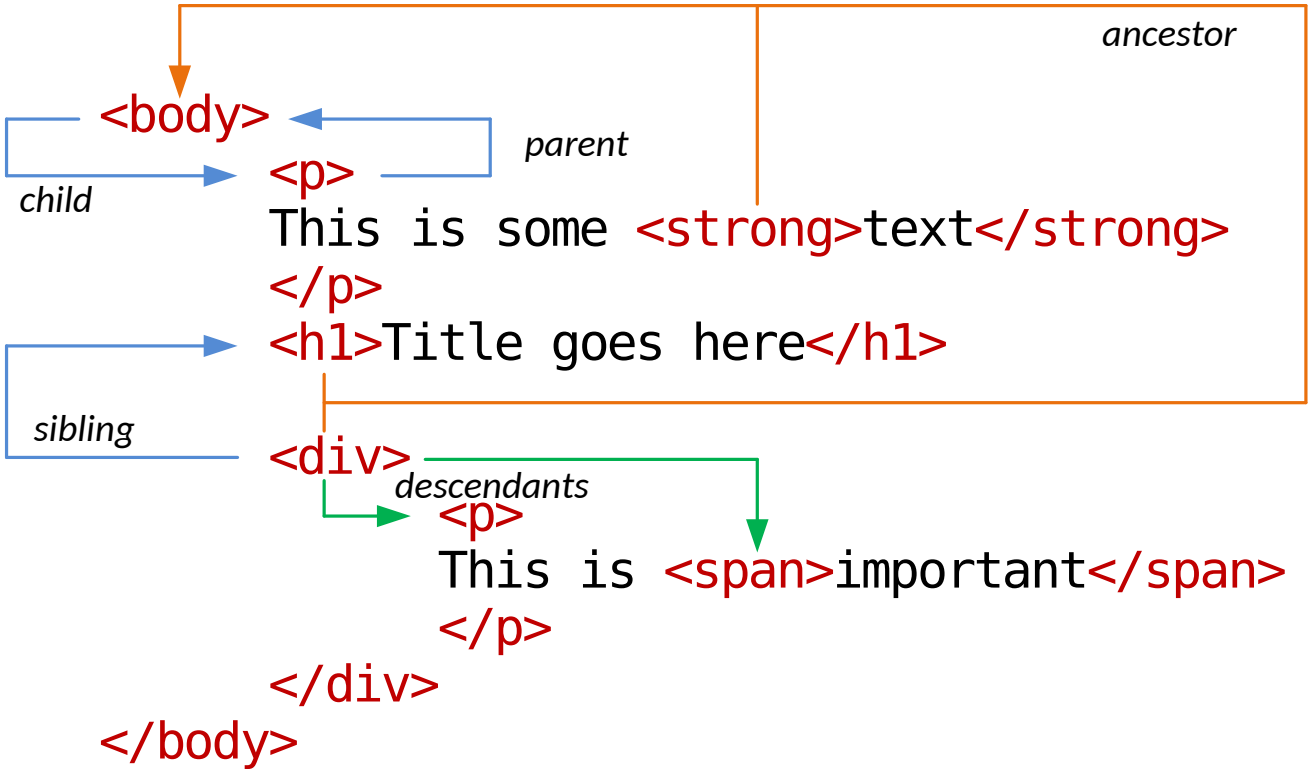
Elements, tags, attributes

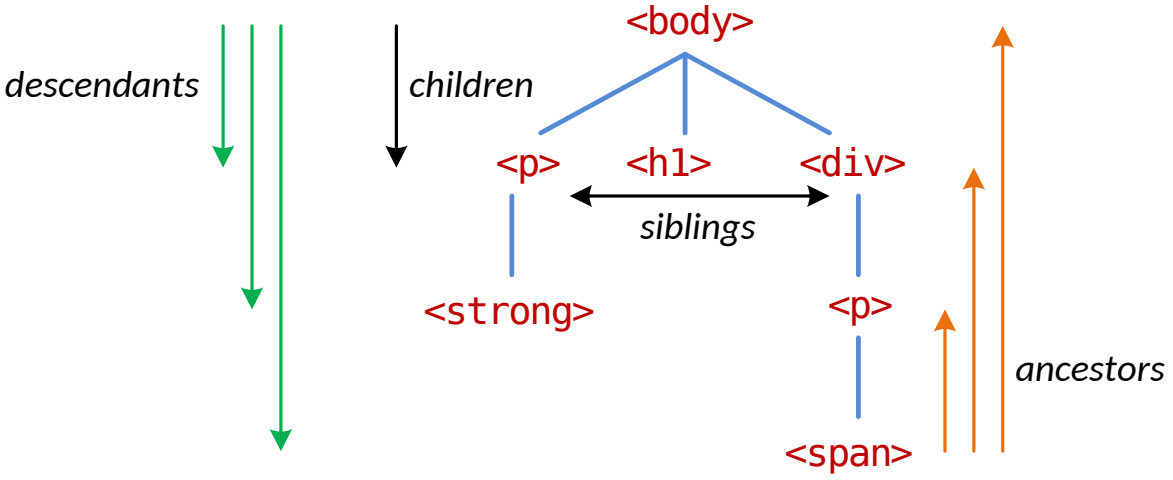
HTML elements

- **HTML documents** are composed of textual content and HTML elements.
- An **HTML element** can contain text, other elements, or be empty. It is identified in the HTML document by tags.
- In the HTML syntax, most elements are written with a start tag and an end tag, with the content in between.
- <tagname>Content goes here…</tagname>
- The HTML **element** is everything from the start tag to the end tag:
- My first paragraph.

- **<h1>...</h1>** is another HTML element.
- There are some HTML elements which don't need to be closed, such as **<img.../>**, **<hr />** and **
** elements. These are known as **void elements** or **empty elements**.
- HTML elements can be **nested** (elements can contain elements).
- All HTML documents consist of nested HTML elements.

Nested Elements





Html tags

- Tags are used to mark up the start and end of an HTML element.
- <tagname>
- <img>,<audio>, <p>, <head>, <body>

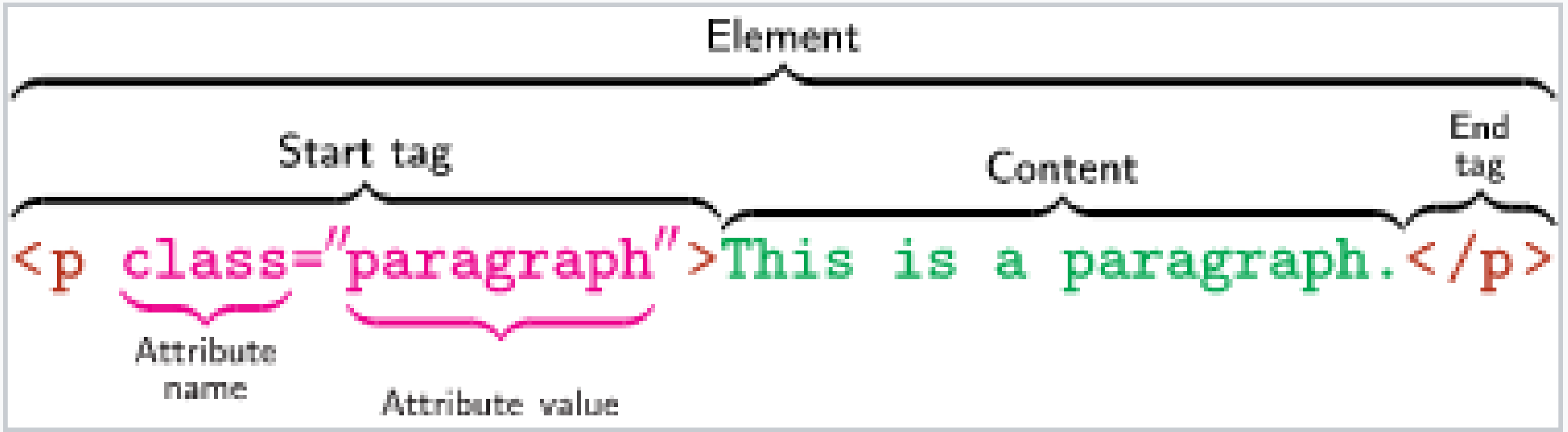
Html attributes

- An attribute defines a property for an element
- Consists of an attribute/value pair
- Appears within the element’s start tag.
- An element’s start tag may contain any number of space separated attribute/value pairs.
- <element attribute=“value”> element content </element>
- <img src=“abc.jpg”, alt=“Image not availabe”>

Attributes

HTML attributes are generally classed as **required attributes**, **optional attributes**, **standard attributes**, and **event attributes**.

- *Required attributes*, needed by a particular element type for that element type to function correctly;
- *Optional attributes*, used to modify the default functionality of an element type;
- *Standard attributes*, supported by many element types; and
- *Event attributes*, used to cause element types to specify scripts to be run under specific circumstances.



Parts of an HTML container element


Opening Tag
Closing Tag
Central Park
Element Name
Attribute
Content
May be text or other HTML elements
Trailing Slash
Example empty element

Element Name

Basic Tags

- <!DOCTYPE HTML>
- <html></html>
- <head></head>
- <meta>
- <title></title>
- <body</body>

- <h1……. h6> </h1……..h6> //headings
-    //paragraph
- <img>   // image
-    //bold
- </br>   //break rule
- <i> //italic
- <body style="background-color:blue;">

Style

- The ***property*** is a CSS property.
- The ***value*** is a CSS value.
- Syntax: <*tagname* style="*property*:*value*;">

- <body style="background-color:blue;">
- <h1 style="font-family:verdana;">abc </h1>

Formatting

HTML uses elements like <b> and <i> for formatting output, like bold or *italic* text.
Formatting elements were designed to display special types of text:

- <b> - Bold text
- - Important text
- <i> - Italic text
- - Emphasized text
- <mark> - Marked text
- <small> - Small text
- <del> - Deleted text
- <ins> - Inserted text
- <sub> - Subscript text
- <sup> - Superscript text

Quotation

- The HTML <q> element defines a short quotation.
- Browsers usually insert quotation marks around the <q> element.

The HTML <blockquote> element defines a section that is quoted from another source.
Browsers usually indent <blockquote> elements.

Color

- In HTML, a color can be specified by using a color name, hex value, rgb value and hsl value.
- Color can be specified for background, text or border etc.
- <h1 style="background-color:DodgerBlue;">Hello World</h1>
- <h1 style="color:rgb(255, 99, 71);">Hello World</h1>
- <h1 style="border:2px solid #ff6347;">Hello World</h1>

Links

- HTML links are hyperlinks.
- You can click on a link and jump to another document.
- ***link text***
- The target attribute specifies where to open the linked document.
- The target attribute can have one of the following values:
- **_blank** - Opens the linked document in a new window or tab
- **_self** - Opens the linked document in the same window/tab as it was clicked (this is default)
- **_parent** - Opens the linked document in the parent frame
- **_top** - Opens the linked document in the full body of the window
- **framename** - Opens the linked document in a named frame
- Visit Google!

- create a bookmark with the id attribute:
- <h2 id="C4">Chapter 4</h2>

- Then, add a link to the bookmark ("Jump to Chapter 4"), from within the same page:
- Jump to Chapter 4

- Add a link to a bookmark on another page:
- Jump to Chapter 4

Images

- Enhance the appearance of the website
- <img src="*url*">
- The alt attribute provides an alternate text for an image, if the user for some reason cannot view it.
- Can specify height and width of the elements.
- <img src=“abc.jpg" alt=“jpg pic” style="width:500px;height:600px;>
- <img src="img_girl.jpg" alt="Girl in a jacket" width="500" height="600">

-

Lists

- Unordered list -
- Ordered list -

- List item <li>
- Style type (Disc, circle, square, none) for UL - <ul style="list-style-type:square;">
- Type(Roman, alphabets, numbers) Attribute for OL- <ol type="1">

- Nested Lists

- <ul style="list-style-type:square;">
- Coffee
- Tea
- Milk
-
  - <ol type="A">
    A. Coffee
    B. Tea
    C. Milk

-
-

Character entities

- These are special characters for symbols for which there is either no easy way to type in via a keyboard (such as the copyright symbol or accented characters) or which have a reserved meaning in HTML (for instance the “<” or “>” symbols).
- They can be used in an HTML document by using the entity name or the entity number.
- e.g.,   and ©

Block Elements

- A block-level element always starts on a new line and takes up the full width available (stretches out to the left and right as far as it can).
- The element is a block-level element.

- An inline element does not start on a new line and only takes up as much width as necessary.
- This is an inline element inside a paragraph.

Divisions

- This ** ** tag is also a container element and is used to create a logical grouping of content
- The element has no intrinsic presentation.
- he tag is an empty container, which defines a division or a section. It does not affect the content or layout and is used to group HTML elements to be styled with CSS or manipulated with scripts.

HTML Forms

Forms

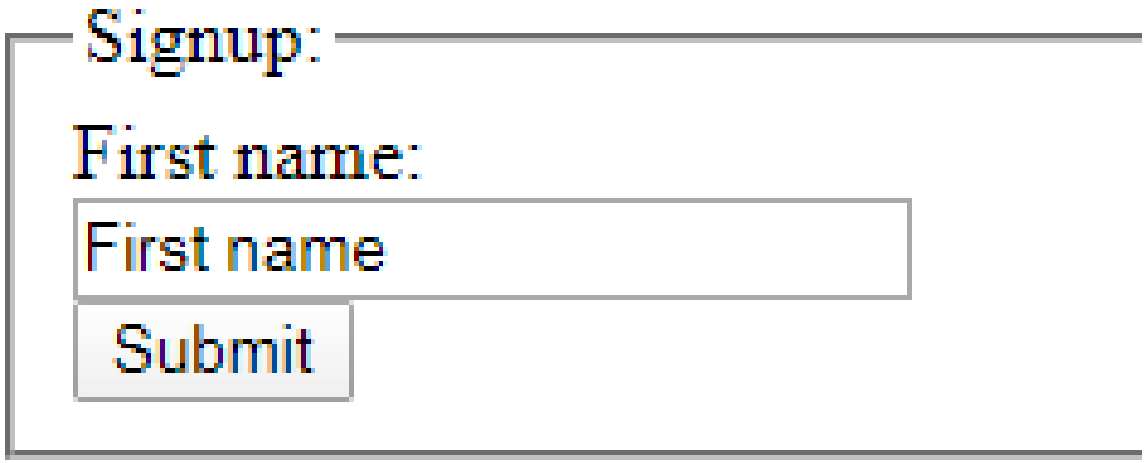
- **Forms** provide the user with an alternative way to interact with a web server.
- Provide rich mechanisms like:
    - Text input
    - Password input
    - Options Lists
    - Radio and check boxes

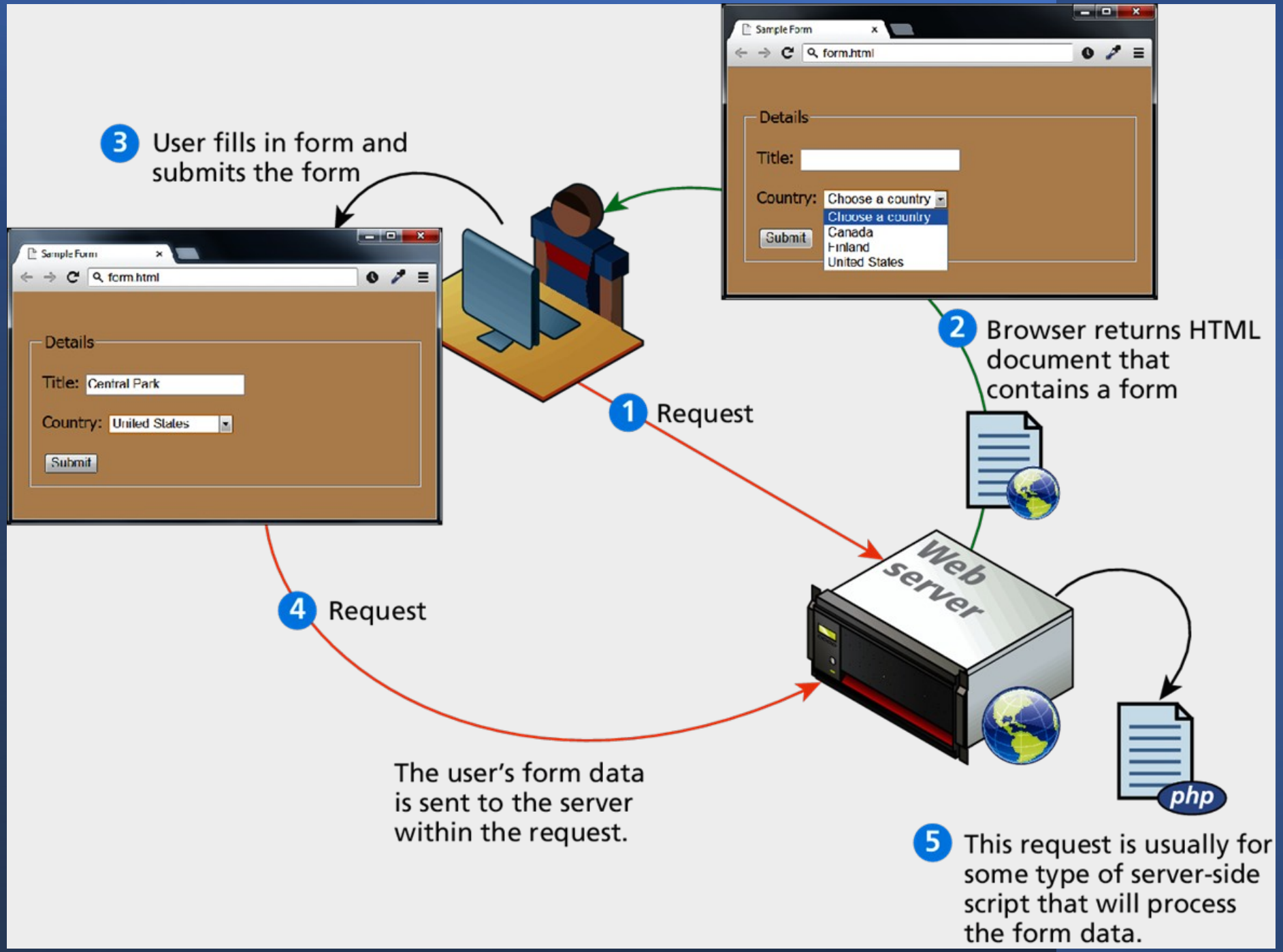
- The HTML <form> element defines a form that is used to collect user input
- An HTML form contains form elements.
- Form elements are different types of input elements, like text fields, checkboxes, radio buttons, submit buttons etc.

- <form>
  .
  form elements
  .
  </form>

Form structure

```
<form action="Test.htm" target="_blank“
method=“GET”>
<fieldset>
    <legend>Signup:</legend>
    First name:<br>
    <input type="text" name="firstname"
value="First name">
    <br>
     <br><br>
     <input type="submit" value="Submit">
  </fieldset>
</form>
```




Sample Form
form.html
Details
Title:
Country:
Choose a country
Choose a country
Canada
Finland
United States
Submit
2 Browser returns HTML document that contains a form
3 User fills in form and submits the form
Sample Form
form.html
Details
Title: Central Park
Country: United States
Submit
1 Request
Web server
4 Request
The user's form data is sent to the server within the request.
php
5 This request is usually for some type of server-side script that will process the form data.

Form Element

- There are two main attributes of <form> element.
- **Action** : Specifies the URL of the server-side resource that will process the form data.
- **Method** : Specifies how the query string data will be transmitted from the browser to the server.

Form - action

- 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.
- <form **action="/action_page.htm** ">
- The form data will be sent to a page on the server called "/action_page.htm". This page contains a server-side script that handles the form data.
- If the action attribute is omitted, the action is set to the current page.

Form - Target

- The target attribute specifies if the submitted result will open in a new browser tab, a frame, or in the current window.
- The default value is "_self" which means the form will be submitted in the current window.
- To make the form result open in a new browser tab, use the value "_blank”
- Other legal values are "_parent", "_top", or a name representing the name of an iframe.

- <form action="/action_page.htm" **target="_blank"**>

Form - method

- The method attribute specifies the HTTP method (GET or POST) to be used when submitting the form data.

- <form action="/action_page.htm" **method="get"**>
- GET method
- POST method

Form – get method

- The default method when submitting form data is GET.
- However, when GET is used, the submitted form data will be visible in the page address field.
- /action_page.php?
  firstname=Mickey&lastname=Mouse

- Appends form-data into the URL in name/value pairs
- The length of a URL is limited (2048 characters)
- Never use GET to send sensitive data! (will be visible in the URL)
- Useful for form submissions where a user wants to bookmark the result
- GET is better for non-secure data, like query strings in Google

forms – post method

- Always use POST if the form data contains sensitive or personal information.
- The POST method does not display the submitted form data in the page address field.
- POST has no size limitations, and can be used to send large amounts of data.
- Form submissions with POST cannot be bookmarked

Form Control Elements

Type	Description
<button>	Defines a clickable button.
<datalist>	An HTML5 element form defines lists to be used with other form elements.
<fieldset>	Groups related elements in a form together.
<form>	Defines the form container.
<input>	Defines an input field. HTML5 defines over 20 different types of input.
<label>	Defines a label for a form input element.
<legend>	Defines the label for a fieldset group.
<option>	Defines an option in a multi-item list.
<optgroup>	Defines a group of related options in a multi-item list.
<select>	Defines a multi-item list.
<textarea>	Defines a multiline text entry box.

Form - Input

- The <input> element is the most important form element.
- The <input> element can be displayed in several ways, depending on the type attribute.

- <input type="text"> Defines a one-line text input field
- <input type="radio"> Defines a radio button (for selecting one of many choices)
- <input type="submit"> Defines a submit button (for submitting the form)

Type	Description
text	Creates a single line text entry box. **<input type="text" name="title" />**
textarea	Creates a multiline text entry box. **<textarea rows="3" ... />**
password	Creates a single line text entry box for a password **<input type="password" ... />**
search	Creates a single-line text entry box suitable for a search string. This is an HTML5 element. **<input type="search" ... />**
email	Creates a single-line text entry box suitable for entering an email address. This is an HTML5 element. **<input type="email" ... />**
tel	Creates a single-line text entry box suitable for entering a telephone. This is an HTML5 element. **<input type="tel" ... />**
url	Creates a single-line text entry box suitable for entering a URL. This is an HTML5 element. **<input type="url" ... />**

- <input type="text"> defines a one-line input field for text input.
- <form>
  First name:
  <input type="text" name="firstname">
  </form>

- <input type="submit"> defines a button for submitting the form data to a form-handler.
- The form-handler is typically a server page with a script for processing input data.
- The form-handler is specified in the form's action attribute.
- <form action="/action_page.php">
  First name:
  <input type="text" name="firstname" value="Mickey">
  Last name:
  <input type="text" name="lastname" value="Mouse">

  <input type="submit" value="Submit">
  </form>

Form – Name

- Each input field must have a name attribute to be submitted.
- If the name attribute is omitted, the data of that input field will not be sent at all.

- ```
  <form action="/action_page.php">
    Last name:<br>
    <input type="text" name="lastname" value="Mouse"><br><br>
    <input type="submit" value="Submit">
  </form>
  ```
  ```

HTML 5 form Elements – datalist

- The <datalist> element specifies a list of pre-defined options for an <input> element.
- Users will see a drop-down list of the pre-defined options as they input data.

- The list attribute of the <input> element, must refer to the id attribute of the <datalist> element
- <form action="/action_page.php">
  <input list="browsers">
  <datalist id="browsers">
    <option value="Internet Explorer">
    <option value="Firefox">
    <option value="Chrome">
  </datalist>
</form>

HTML5 Attributes - Pattern

- Explore yourself

Form – fieldset

- The <fieldset> element is used to group related data in a form.
- The <legend> element defines a caption for the <fieldset> element.

- ```
  <form action="/action_page.htm">
    <fieldset>
      <legend>Personal information:</legend>
      First name:<br>
      <input type="text" name="firstname" value="Mickey"><br>
      Last name:<br>
      <input type="text" name="lastname" value="Mouse"><br><br>
      <input type="submit" value="Submit">
    </fieldset>
  </form>
  ```
  ```

Form – select

- The <select> element defines a drop-down list.
- The <option> elements defines an option that can be selected.
- By default, the first item in the drop-down list is selected.
- To define a pre-selected option, add the **selected** attribute to the option
- Option items can be grouped together via the **<optgroup>** element.
- The **value** attribute of the <option> element is used to specify what value will be sent back to the server.
- The value attribute is optional; if it is not specified, then the text within the container is sent instead

Form – select

- Use the **size** attribute to specify the number of visible values

- Use the **multiple** attribute to allow the user to select more than one value.
- <select name="cars" size="4" **multiple**>
  <option value="volvo“>Volvo</option>
  <option value=“saab” selected>Saab</option>
  <option value="fiat">Fiat</option>
  <option value="audi">Audi</option>
  </select>

Form – Textarea

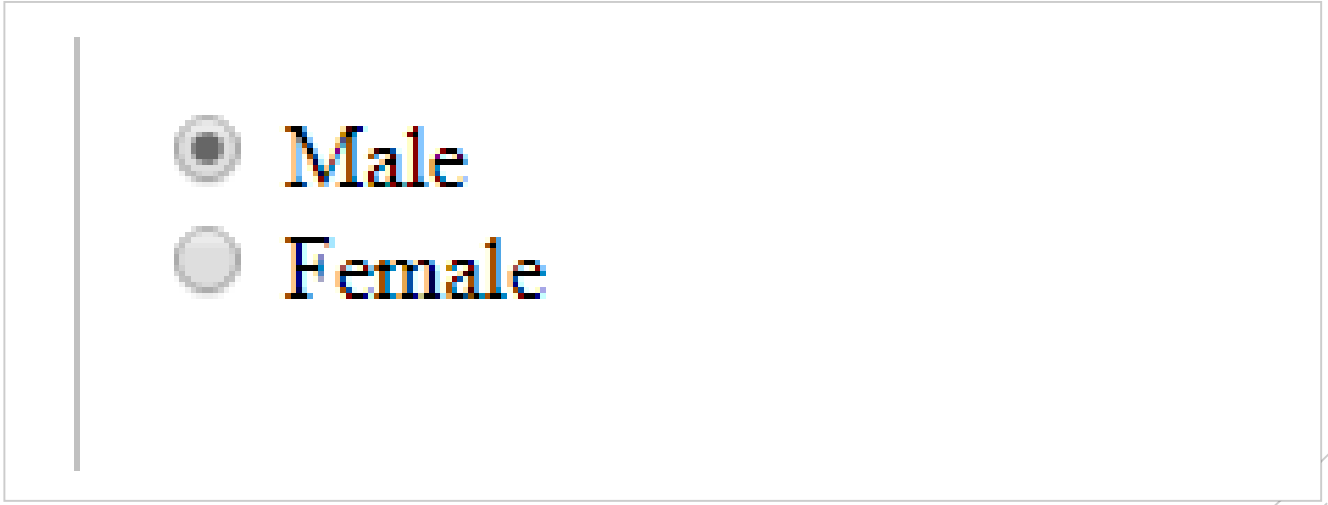
- The <textarea> element defines a multi-line input field (a text area)
- <textarea name="message" rows="10" cols="30">
  The cat was playing in the garden.
  </textarea>

Radio buttons & checkboxes

- <input type="radio"> defines a radio button.
- **Radio buttons** are useful when you want the user to select a single item from a small list of choices and you want all the choices to be visible
- Radio buttons let a user select ONE of a limited number of choices
- **Checkboxes** are used for getting yes/no or on/off responses from the user.
- checkboxes are added via **the <input type="checkbox”>** element
- You can also group checkboxes together by having them share the same name attribute
- Each checked checkbox will have its value sent to the server
- Like with radio buttons, the checked attribute can be used to set the default value of a checkbox

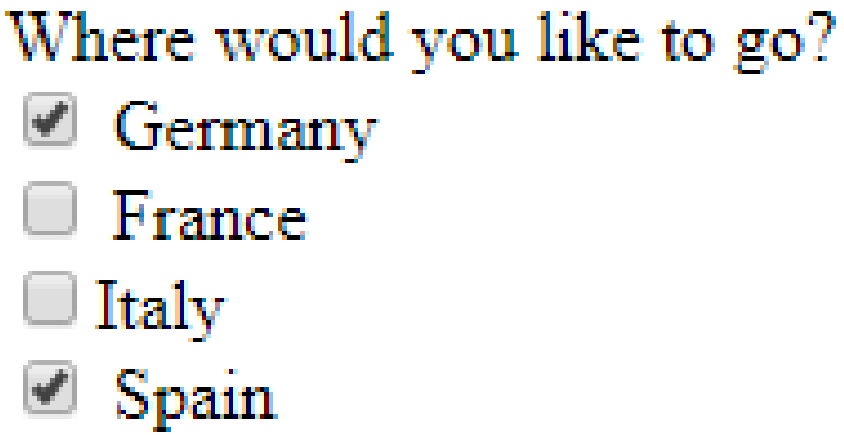
Radio Buttons

- <form>
  <input type="**radio**" name="gender" value="
male" checked> Male
  <input type="radio" name="gender" value="f
emale"> Female
</form>



CheckBoxes

- Where would you like to go?
- <input type="**checkbox**" name="Country" value="Germany" checked> Germany
- <input type="checkbox" name="Country" value="France"> France
- <input type="checkbox" name="Country" value="Italy">Italy
- <input type="checkbox" name="Country" value="Spain"> Spain
-



Forms- Buttons

- The <button> element defines a clickable button.
- <button type="**button**" onclick="alert('Hello World!')">Click Me!</button>

- **Note:** Always specify the **type** attribute for the button element. Different browsers may use different default types for the button element.

Type	Description
<input type="submit">	Creates a button that submits the form data to the server.
<input type="reset">	Creates a button that clears any of the user's already entered form data.
<input type="button">	Creates a custom button. This button may require Javascript for it to actually perform any action.
<input type="image">	Creates a custom submit button that uses an image for its display.
<button>	Creates a custom button. The <button> element differs from <input type="button"> in that you can completely customize what appears in the button; using it, you can, for instance, include both images and text, or skip server-side processing entirely by using hyperlinks. You can turn the button into a submit button by using the type="submit" attribute.

Other inputs

<input type=“hidden”>
<input type=“file”>
<input type=“color”>
<input type=“date”>
<input type=“time”>
<input type=“datetime”>
<input type=“datetime-local”>
<input type=“month”>
<input type=“week”>

Type	Description
date	Creates a general date input control. The format for the date is "yyyy-mm-dd".
time	Creates a time input control. The format for the time is "HH:MM:SS", for hours:minutes:seconds.
datetime	Creates a control in which the user can enter a date and time.
datetime-local	Creates a control in which the user can enter a date and time without specifying a time zone.
month	Creates a control in which the user can enter a month in a year. The format is "yyyy-mm".
week	Creates a control in which the user can specify a week in a year. The format is "yyyy-W##".

Other Control elements

- The <progress> and <meter> elements can be used to provide feedback to users,
  - but requires JavaScript to function dynamically.
- The <output> element can be used to hold the output from a calculation.
- The <keygen> element can be used to hold a private key for public-key encryption

Multimedia

- Browser support
- Formats
- Plugins

- <video> element
- <audio> element

Multimedia- Video

- <video width="320" height="240" controls>
  <source src="movie.mp4" type="video/mp4“ >
  <source src="movie.ogg" type="video/ogg">
  Your browser does not support the video tag.
  </video>
- Controls add control functionality
- Autoplay plays the video automatically when page loads

Multimedia- audio

- <audio controls>
  <source src="horse.ogg" type="audio/ogg">
  <source src="horse.mp3" type="audio/mpeg">
  Your browser does not support the audio element.
  </audio>

Youtube videos

- <iframe width="420" height="315" src="https://www.youtube.com/embed/tgbNymZ7vqY"> </iframe>

- <iframe width="420" height="315" src="https://www.youtube.com/embed/tgbNymZ7vqY?**autoplay=1**"> </iframe>

- <iframe width="420" height="315" src="https://www.youtube.com/embed/tgbNymZ7vqY?playlist=tgbNymZ7vqY&**loop=1**"> </iframe>

Assignment # 01

- Web development(frontend) – Tools, Pros &Cons and Languages (1 page, table)
- Specify the tool/language you like and why?
- Make a 1 page website utilizing all these concepts.
- Include all tags studied.
- Visually appealing.
- Set images as background and as links.
- Website based on some idea.
- Specify the type of website i.e. personal, commercial etc
- Due date: Sunday

References

- W3Schools.com
- https://en.wikipedia.org/wiki/Website