Basic Construction of an HTML Page:

Tables in HTML

- How to create tables
- Table attributes
- Page lay-out using tables

Creating Html tables:

1. Tables display information in rows and columns. Tables are commonly used to display all manner of data that fits in a grid such as train schedules, television listings, financial reports etc.
2. <table> tag is used to start a table while </table> tag indicates the end of the table.

Table Structure

- ➔ <tr> tag starts a row of the table and </tr> ends the row
- ➔ <td> is used to create a cell inside the row while </td> ends the cell.
- ➔ The contents of the cell are written between <td> and </td> tags
- ➔ <th> tag is used to declare the cell of the heading row of the table

Basic Construction of an HTML Page:

Table attributes:

1. Table level attributes
2. Row level attributes
3. Cell level attributes

Table level attributes:

- The Border Attribute: Indicates the presence of the border around the table: <table border="1">
- The align Attribute: <table align= "center, right or left">
- The bgcolor Attribute: sets the background color of the table: <table bgcolor="gray">
- The background Attribute: sets the specified image at the background of the table: <table background="image-title">
- The height and width Attributes:
- The cellpadding Attribute: The cellpadding attribute is used to create a gap between the edges of a cell and its contents: <table cellpadding="50">

Row level attributes:

- The valign Attributes:
  <tr valign="center,top or bottom">
- The align Attribute:
  <tr align="center,right or left">
- The bgcolor Attribute:
  <tr bgcolor="gray">
- The background Attribute:
  <tr background="image-name">
- The height and width Attributes:
  <tr height="20" widht="20">

Basic Construction of an HTML Page:

Cell level attributes:

- **The align Attribute:**
  <td align="center,right or left">
- **The bgcolor Attribute:**
- **The height and width Attributes:**
- **The valign Attributes:**
- **a)The rowspan Attributes: used when a cell should span across more than one rows.**
- **b)The colspan Attribute: used when a cell should span across more than one column**

Adding caption to the table:

➔ **<caption> tag is used to add a caption of the table**

➔ **We usually add caption before the first row of the table**

Basic Construction of an HTML Page:

HTML FORMS

- **Forms provide a means of submitting information from the client to the server**
- **HTML supports tags for creating forms, however, it does not process the information**
- **Use server-side script to process form information**
- **Server-side script runs on the Web server and receives data from a form and uses it to perform a set of tasks**
- ***HTML FORMS elements***

  - ➔ **form> tag is used to start a form**
  - ➔ **</form> tag is used to end a form**
  - ➔ **Between <form> and </form>, form elements are placed**
  - ➔ **We can declare a form as: <form attributes>**
    form elements and layout tags: </form>
    - **A single page can include several different forms, but you cannot nest one form inside another**
    - **Common Form Attributes:**
      - **action - gives the URL of the application that is to receive and process the forms data**
      - **method - sets the HTTP method that the browser uses to send the form's data to the server for processing; Either POST or GET**
      - **name – name of the form**

Basic Construction of an HTML Page:

HTML Forms and their elements

1. Textbox: <INPUT TYPE="TEXT"> • NAME • SIZE • VALUE • MAXLENGTH **<INPUT TYPE="TEXT" NAME=NAME SIZE=30 VALUE="ALI" MAXLENGTH=15>**	*2. Password Field* <INPUT TYPE="PASSWORD"> • NAME • SIZE • VALUE • MAXLENGTH **<INPUT TYPE="PASSWORD" NAME=NAME SIZE=30 VALUE="ALI" MAXLENGTH=15>**	*3. Hidden Field* <INPUT TYPE="HIDDEN"> • NAME • VALUE **<INPUT TYPE="HIDDEN" NAME=NAME VALUE="ALI" >**	4. Checkbox <INPUT TYPE="CHECKBOX"> • CHECKED • NAME • VALUE **<INPUT TYPE="CHECKBOX" NAME="CHECK" VALUE=" CHECKED">**
5. Radio Button <INPUT TYPE="RADIO"> • CHECKED • NAME • VALUE **<INPUT TYPE="RADIO" NAME="RADIO" VALUE="" CHECKED>MALE** **<INPUT TYPE="RADIO" NAME="RADIO" VALUE="" >FEMALE**	6. Text Area Attributes: • Rows • Cols • Name **<TEXTAREA rows="5" cols="5"> Default text </TEXTAREA>**	7. File-field To input file from user: **<input type="file" name="file-name">**	8. Submit Button **<INPUT TYPE="SUBMIT" VALUE="SAVE">**

Basic Construction of an HTML Page:

HTML Forms and their elements

9. Select List

- Drop down list:
- Select
- Name, size
- Option

```
<SELECT name="name" >
    <OPTION value="lahore">LAHORE</option>
    <OPTION value="karachi">KARACH</option>
    <OPTION value="islamabad">ISLAMABAD</option>
</SELECT>
```

To create scrolling select list, we use the size attribute of the <select>
The value of the size attribute is the number of options you want to be visible at any one time.

EXAMPLE:

```
<SELECT name="name" size="size">
<OPTION value="lahore">LAHORE</option>
<OPTION value="karachi">KARACH</option>
<OPTION value="islamabad">ISLAMABAD</option>
</SELECT>
```

10. Reset Button

```
<INPUT  TYPE="RESET"  VALUE="RESET">
```

Basic Construction of an HTML Page:

Enhancements in forms

HTML5 enhances the forms in two ways

Adding new attributes to exiting elements

Required attribute:	***Placeholder attribute:***	***Pattern attribute:***	***Disabled attribute:***
• tells the browser to only submit the form if the field in question is filled out • ensures that all necessary information is provided by the user • <input type="text" name="name" required>	• Allows a short hint to be displayed inside the form element • tell the user what data should be entered in that field • <input type="text" name="name" placeh older="Only upper case letters" >	• enables you to provide a regular expression that the user's input must match in order to be considered valid • <input type="text" pattern="Regular Expression"> • <input type="text" pattern="[a-z]{1,20}">	• have the content grayed out in the browser • prohibit the user from focusing on a form control that has the disabled • <input type="text" disabled>

readonly attribute:	***The autocomplete attribute:***
• it makes it impossible for the user to edit the form field • the field can receive focus • <input type="text" readonly>	• this will be a drop-down list that appears when the user begins typing • <input type="text" autocomplete>

Basic Construction of an HTML Page:

HTML new elements:			
Email element	***Date element***	***Number element***	***Color element***
• It ensures that user enters a valid email address <input type="email" name="email">	• The date element: • It shows a calendar to user to select a date <input type="date" name="DoB">	• It ensures that user enters only a numeric value <input type="number" name="number">	• It facilitates the user to choose a color <input type="color" name="color">

Write accessible HTML using Semantic HTML tags

- Semantic within HTML is the practice of giving content on the page meaning and structure by using the proper element. Semantic code describes the value of content on a page, regardless of the style or appearance of that content. There are several benefits to using semantic elements, including enabling computers, screen readers, search engines, and other devices to adequately read and understand the content on a web page. Additionally, semantic HTML is easier to manage and work with, as it shows clearly what each piece of content is about.

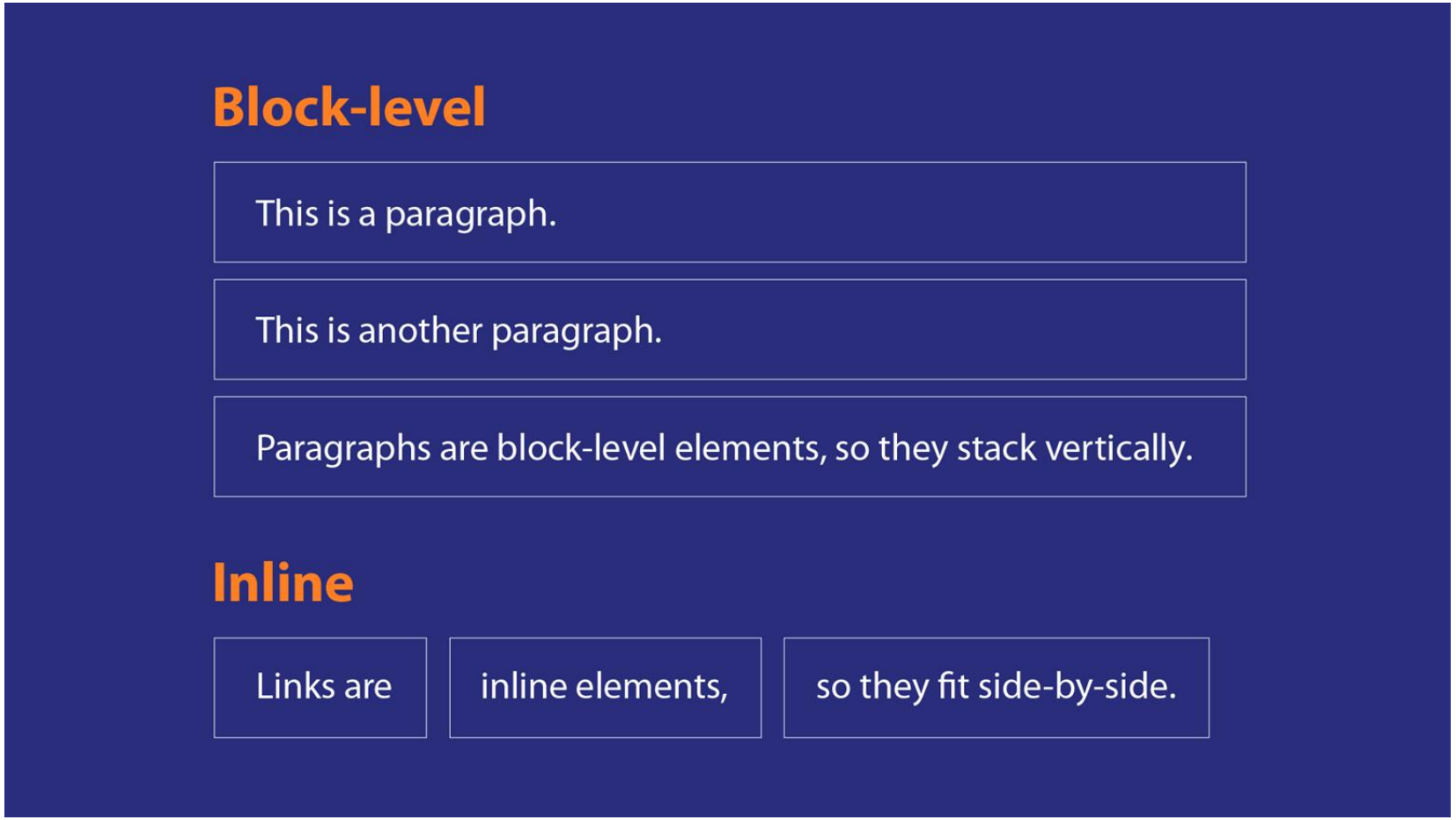
- Before going into semantic elements, let's look at two elements s and s that don't hold any semantic value. They exist for styling purposes only. s, and s are HTML elements that act as containers solely for styling purposes.

To understand the difference between and span<>, we must know the difference between inline and block elements

Block-level element: Block-level elements begin on a new line, stacking one on top of the other, and occupy any available width. Block-level elements may be nested inside one another and may wrap inline-level elements. We'll most commonly see block-level elements used for larger pieces of content, such as paragraphs.

Inline-level elements: Inline-level elements do not begin on a new line. They fall into the normal flow of a document, lining up one after the other, and only maintain the width of their content. Inline-level elements may be nested inside one another; however, they cannot wrap block-level elements. We'll usually see inline-level elements with smaller pieces of content, such as a few words.

Write accessible HTML using Semantic HTML tags



Basic Construction of an HTML Page:

- A is a block-level element that is commonly used to identify large groupings of content, and which helps to build a web page's layout and design.

- A : It is an inline-level element commonly used to identify smaller groupings of text within a block-level element.

For example, if we have a with an orange background that contains social media links, our first thought might be to give the a class value of orange. What happens if that orange background is later changed to blue? Having a class value of orange no longer makes sense. A more sensible choice for a class value would be social, as it pertains to the contents of the , not the style.

```
<!-- Division -->
<div class="social">
  <p>I may be found on...</p>
  <p>Additionally, I have a profile on...</p>
</div>

<!-- Span -->
<p>Soon we'll be <span class="tooltip">writing HTML</span> with the best of them.</p>
```

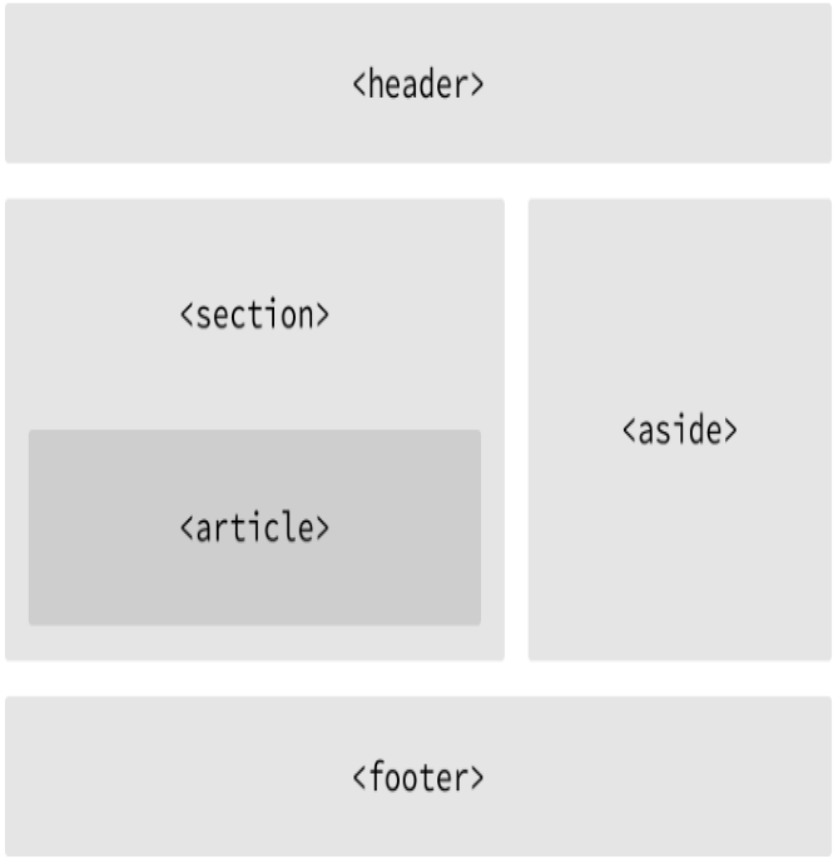
Write accessible HTML using Semantic HTML tags

Building Structure:

For the longest time the structure of a web page was built using divisions. The problem was that divisions provide no semantic value, and it was fairly difficult to determine the intention of these divisions.

Fortunately HTML5 introduced new structurally based elements, including the <header>, <nav>, <article>, <section>, <aside>, and <footer> elements. All of these new elements are intended to give meaning to the organization of our pages and improve our structural semantics.

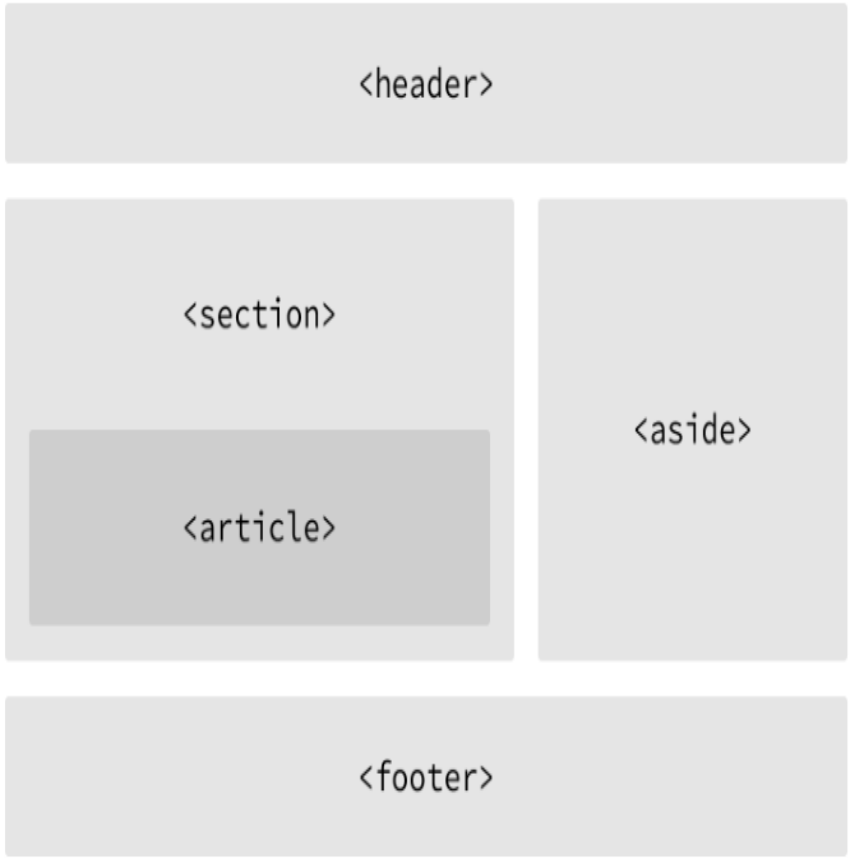
They are all block-level elements and do not have any implied position or style. Additionally, all of these elements may be used multiple times per page, so long as each use reflects the proper semantic meaning.



Write accessible HTML using Semantic HTML tags

Building Structure:

- Main: The main tag defines the primary section in the document related to the central content of a document with a <main> tag.
- Section: It is used to define specific sections in a document such as a chapter, header, footer, or any other section, and is specified with the <section> tag.
- Header: The header tag defines the title or heading of a document or its section. It is specified with the <header> tag.
- Footer: The footer tag defines the section of a document that contains information such as copyright or author's information. It is designated with the <footer> tag.
- Article: The article tag represents an independent or self-contained part of the content of a document with the tag <article>.



Example of using some semantic tags:

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

<section>
  <h1>WWF</h1>
  <p>The World Wide Fund for Nature (WWF) is an international organization working on issues regarding the
conservation, research and restoration of the environment, formerly named the World Wildlife Fund. WWF was
founded in 1961.</p>
</section>

<section>
  <h1>WWF's Panda symbol</h1>
  <p>The Panda has become the symbol of WWF. The well-known panda logo of WWF originated from a panda named
Chi Chi that was transferred from the Beijing Zoo to the London Zoo in the same year of the establishment of
WWF.</p>
</section>

</body>
```

Making an html page using Semantics tags:

Open the notepad++ or VScode editor, and use some other semantic html tags on the above mentioned coding example: i.e. add <header>, <footer>, <aside> etc.

Questions ?