

Web System & Technology

Lecture # 22

Ehtisham Rasheed

Department of Computer Science
University of Gurjat, Gujrat

UNIVERSITY OF GUJRAT



HTML Layout

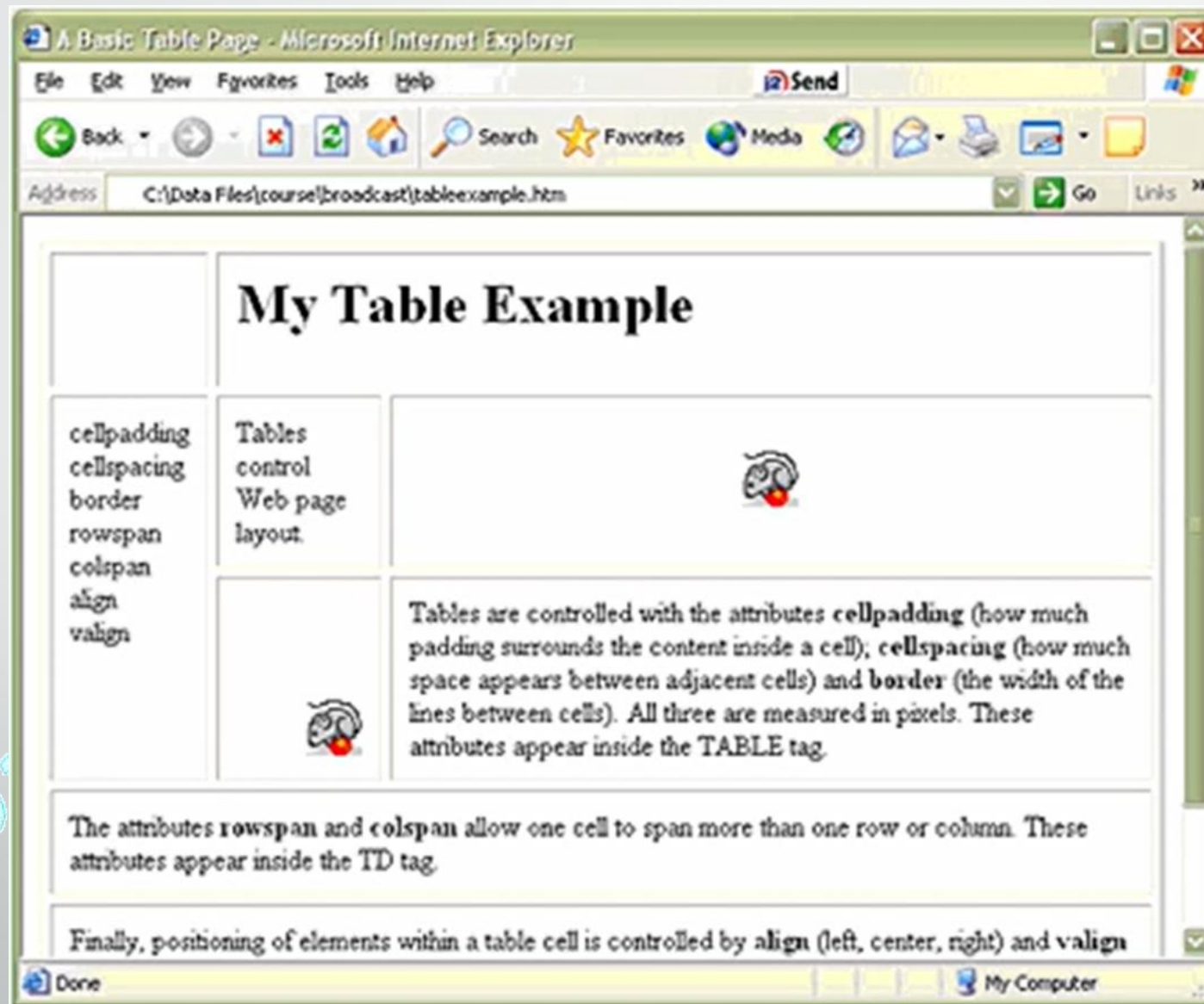
- The arrangements of elements (header, navigation, content, sidebar, footer) in a webpage
- Increase user experience and readability

Prepared BY: Ehtisham Rasheed

Common Layout Types

- Single Column Layout
 - Good for mobile devices
- Two Column Layout
 - Sidebar + Main Content
- Three Column Layout
 - Sidebar + Main Content + Sidebar
- Header-Content-Footer Layout
 - Classic blog or article site

Table (1995-2005)



Float (2005-2017)



Float (2005-2017)

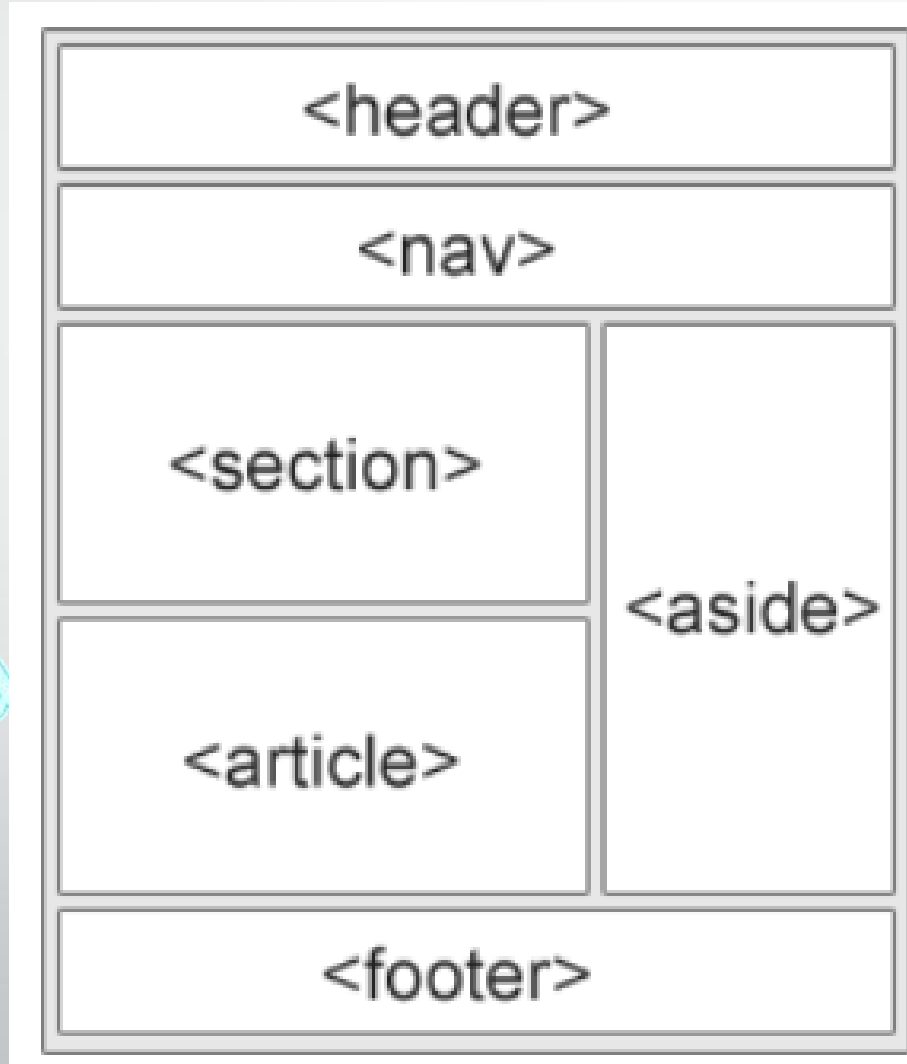


Flex & Grid (2017-now)



In 2017 Bootstrap 4 was in beta and they had replaced their float based grid system with a flexbox based system.

HTML Layout Elements



HTML Layout Techniques

- There are four different techniques to create multicolumn layouts. Each technique has its pros and cons:
 - CSS Framework
 - CSS Float Property
 - CSS Flexbox
 - CSS Grid

Prepared BY: Ehtisham Rasheed

CSS Framework

- CSS Frameworks are pre-written CSS and JavaScript components to create responsive layouts quickly
- Popular examples include:
 - Bootstrap, Tailwind CSS, Foundation etc
- **How It Works**
 - Frameworks offer pre-defined classes for grid systems, buttons, navigation bars, and other UI components
 - Developers apply these classes to HTML elements to achieve a consistent design and responsive layout

CSS Framework (Pros & Cons)

- **Pros**

- **Rapid Development:** Speeds up the development process with ready-to-use components.
- **Consistent Design:** Ensures uniform styling across all pages.
- **Responsive Design:** Built-in responsiveness for multiple screen sizes.
- **Cross-Browser Compatibility:** Handles browser inconsistencies.

- **Cons**

- **Bloat and Overhead:** Unused classes and styles increase page size.
- **Lack of Uniqueness:** Websites may look similar due to common design patterns.
- **Learning Curve:** Requires learning the framework's class naming conventions.
- **Customization Limitations:** Difficult to override pre-defined styles without deep knowledge

CSS Float Layout

- We can design entire layout using **float** property
- You just need to remember how **float** and **clear** property works
- The **float** property was historically used to create multi-column layouts before Flexbox and Grid were introduced
- **How It Works**
 - Elements are floated to the left or right, allowing other content to wrap around them
 - Clear property is used to overlapping elements

CSS Float Layout (Pros & Cons)

- **Pros**
 - **Simple to implement:** Easy to understand and implement for basic layouts
 - **Browser support:** Supported by all browsers, even older versions
- **Cons**
 - **Layout Issues:** Floats can cause clearfix issues and unexpected behavior
 - **Not Flexible:** Difficult to maintain in responsive designs
 - **No Vertical Alignment:** Requires additional tricks for vertical alignment
 - **Outdated:** Considered outdated with the introduction of Flexbox and Grid

CSS Flexbox Layout

- **Flexbox (Flexible Box Layout)** is a one-dimensional layout model that arranges items in a row or column
- **How It Works**
 - A container is set to `display: flex;` making its children flex items
 - Flex items are aligned and distributed within the container using properties like `justify-content` and `align-items`

CSS Flexbox Layout (Pros & Cons)

- **Pros**

- **Flexible and Responsive:** Easily adjusts to different screen sizes
- **One-Dimensional Layout:** Ideal for creating layouts in a single row or column
- **Alignment Control:** Precise control over horizontal and vertical alignment
- **Order Property:** Allows reordering of elements without changing HTML structure

- **Cons**

- **Complex for Complex Layouts:** Not suitable for two-dimensional layouts
- **Browser Compatibility:** Requires fallbacks for very old browsers (e.g., IE 9)

CSS Grid Layout

- **CSS Grid** is a two-dimensional layout system that allows the creation of complex layouts using rows and columns
- **How It Works**
 - A container is set to **display: grid;** and **grid items** are arranged using rows and columns defined by **grid-template-rows** and **grid-template-columns**
 - Items can be positioned precisely using **grid-row** and **grid-column**

CSS Grid Layout (Pros & Cons)

- **Pros**

- **Two-Dimensional Layouts:** Handles both rows and columns simultaneously
- **Complex Layouts Made Easy:** Ideal for complex, magazine-style layouts
- **Alignment Control:** Precise control over alignment and spacing
- **Responsive Design:** Media queries make it easy to adjust the grid structure

- **Cons**

- **Learning Curve:** More complex to learn compared to Flexbox
- **Overkill for Simple Layouts:** Not necessary for one-dimensional layouts
- **Browser Support:** Limited support in very old browsers (e.g., IE 10 and below)

Comparison of Layout Techniques

Technique	Pros	Cons	Best Use Case
CSS Framework	Rapid development, consistent design	Bloat, learning curve, less uniqueness	Quick prototyping, admin dashboards
CSS Float	Simple to use, good browser support	Layout issues, outdated, no alignment	Wrapping text around images
CSS Flexbox	Flexible, responsive, alignment control	Complex for 2D layouts	One-dimensional layouts (navbars, cards)
CSS Grid	2D layouts, precise control, complex designs	Learning curve, overkill for simple layouts	Two-dimensional layouts (galleries)