



# Flexbox and Grid Layouts

Tessema Mengistu (Ph.D.)  
Department of Computer Science  
Virginia Tech  
[mengistu@vt.edu](mailto:mengistu@vt.edu)

# Outline

- HTML Layout
- Flexbox Layout
- Grid Layout

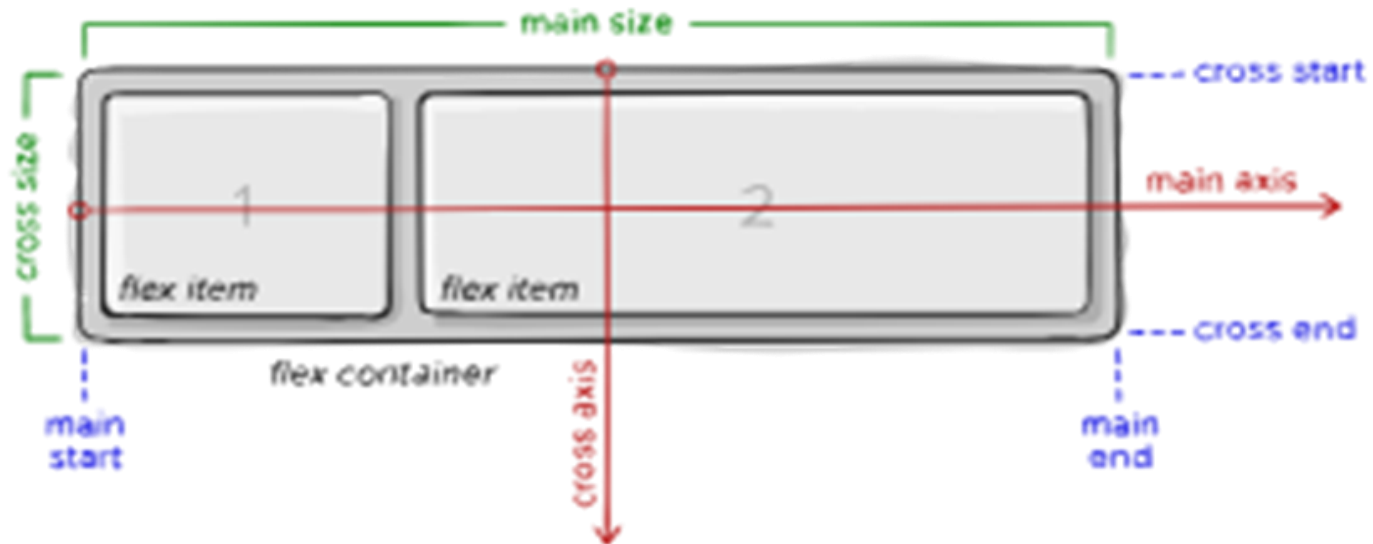
# HTML Layout

- By default:
  - Block elements flow from top to bottom
  - Inline elements flow from the left side of the block elements that contains them to the right side
- Using **float** you can take the element out of the default flow
  - Any content that follows a floated element fill in the space to the side of the element
- Absolute positioning
- Using table

# Flexbox Layout

- The Flexible Box Layout Module
  - Makes it easier to design flexible responsive layout structure without using float or positioning
- Supported by all modern browsers
- Elements are defined in a container
  - Flex container

# Flexbox Layout



Source: <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

# Flexbox Layout

- The flex container becomes flexible by setting the display property to flex.

```
<div class="fcontainer">  
  <div> . . . </div>  
  <div> . . .</div>  
  <div> . . . </div>  
</div>
```

```
.fcontainer{  
  display: flex;  
}
```

# Flexbox Layout

- The flex container properties are:
  - `flex-direction`
    - Controls the direction the items stack in the container
    - Values: `row`, `row-reverse`, `column`, `column-reverse`
  - `flex-wrap`
    - Specifies whether the flex items should wrap or not
    - Values: `wrap`, `nowrap`, `wrap-reverse`
  - `flex-flow`
    - A shorthand property for setting the above two
  - `justify-content`
    - Align the flex items
    - Values: `center`, `flex-start`, `flex-end`, `space-between`, `space-around`

# Flexbox Layout

- `align-items`

- Align the flex items
- Values: `center`, `flex-start`, `flex-end`, `stretch`, `baseline`

- `align-content`

- Align the flex lines
- Values: `center`, `flex-start`, `flex-end`, `stretch`, `space-between`, `space-around`



# Flexbox Layout

- The direct child elements of a flex container automatically becomes flexible (flex) items.
- The flex item properties are:
  - `order`
  - `flex-grow`
  - `flex-shrink`
  - `flex-basis`
  - `flex`
  - `align-self`

# Grid Layout

- The CSS Grid Layout Module offers a grid-based layout system
  - Rows and columns
- Supported in all modern browsers
- Elements are defined in a container
  - Grid container

# Grid Layout

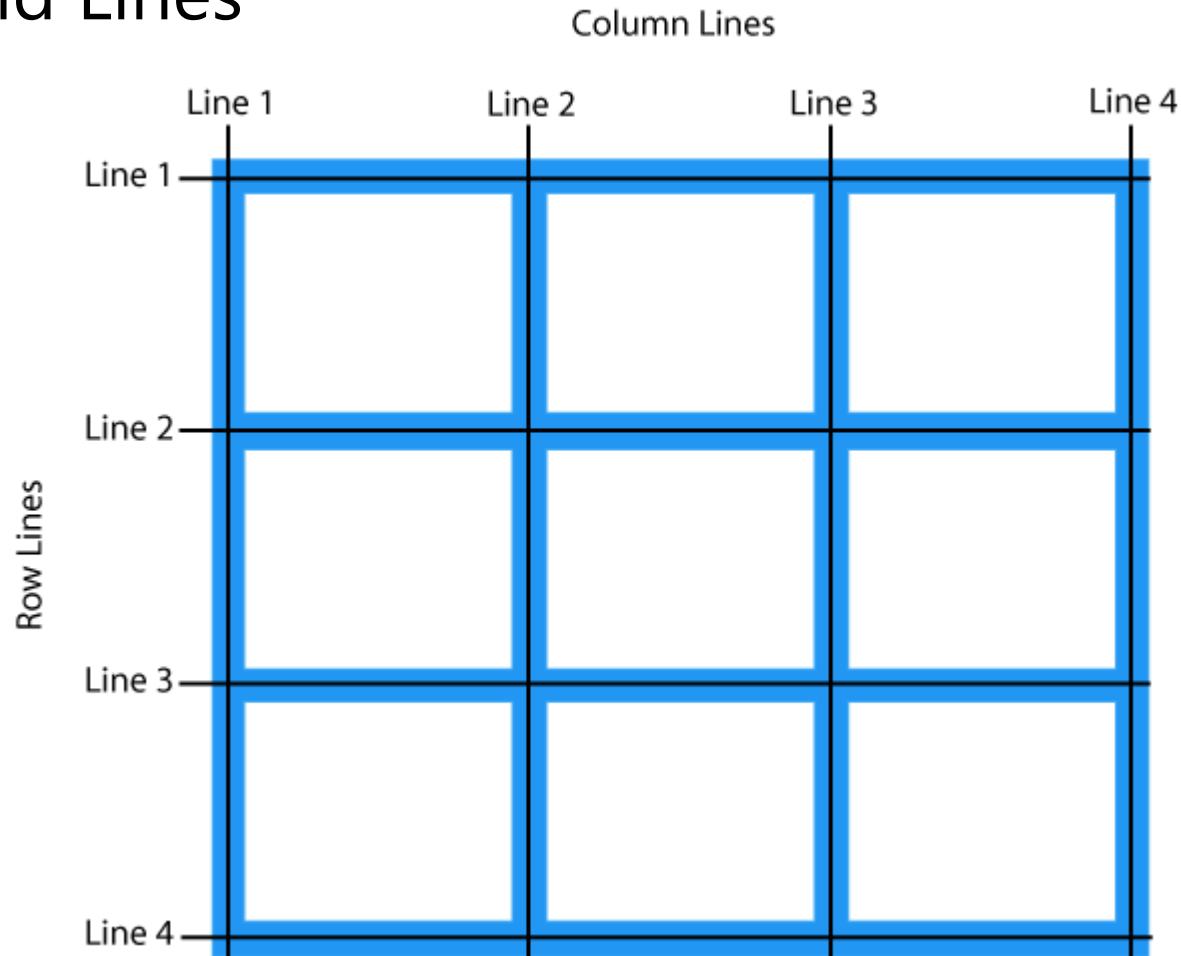
- The container becomes a grid container when its display property is set to **grid** or **inline-grid**

```
<div class="gcontainer">  
  <div> . . . </div>  
  <div> . . .</div>  
  <div> . . . </div>  
</div>
```

```
.gcontainer{  
  display: grid;  
}
```

# Grid Layout

- Grid Lines



# Grid Layout

- Grid Gaps
  - The spaces between each column/row are called gaps.
  - The gap size can be adjusted by using one of the following properties:
    - `column-gap`
    - `row-gap`
    - `gap`

# Grid Layout

- `grid-template-columns`

- Defines the number of columns in your grid layout and width of each column

```
.grid-container {  
  display: grid;  
  grid-template-columns: 200px 400px 150px  
  auto 400px;  
}
```

- `grid-template-rows`

- Defines the number of rows in your grid the height of each row

# Grid Layout

- justify-content

- Align the whole grid inside the container
- Values: center, start, space-evenly, end, stretch, space-between, space-around

- align-content

- Vertically align the whole grid inside the container
- Values: center, start, space-evenly, end, stretch, space-between, space-around

# Grid Layout

- Placing items
  - `grid-column-start`
  - `grid-column-end`
  - `grid-row-start`
  - `grid-row-end`
  - `grid-column`
  - `grid-row`



# References

- W3Schools.com
  - [https://www.w3schools.com/css/css3\\_flexbox.asp](https://www.w3schools.com/css/css3_flexbox.asp)
  - [https://www.w3schools.com/css/css\\_grid.asp](https://www.w3schools.com/css/css_grid.asp)
- <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>
- <https://css-tricks.com/snippets/css/complete-guide-grid/>