HTML —> CSS —> Bootstrap —> Javascript —> Flask …

Display:

There are 4 types of display:

1. **Block** : Consumes the entire width (of the screen), height depends on content.

You can modify the width of block element.

e.g. p, h1-h6, div, ol, ul, form, etc.

2. **Inline** : Width & height both depends on the content.

You can't change the width of inline element.

You can make a block element as switch and vice-versa.

e.g. span, img, a, etc.

3. **Inline-Block**

Best of both worlds - It occupies as much space as the content demands, but is flexible in terms of modifying the width.

e.g. img is kind of inline-block

4. **None**

Hides the element.

Hiding elements:

We can hide the elements in 2 ways:

1. display: none;
2. visibility : hidden;

Position of elements are determined by:

a. Box Model
b. Margin
c. Size of border
d. Width & Height of the element
e. Display block or inline
f. Allowed to set width or not.

Positions:

4 positions:

1. **Static**: All HTML elements are static in their position by default.

2. **Relative: (Push from …)**
    a. Top
    b. Bottom
    c. Left
    d. Right

HTML	CSS

<body>
  <div class = "red"> </div>
  <div class = "blue"> </div>
  <div class = "yellow"> </div>
</body>
	.red{
	  height:100px;
	  width:100px;
	  background-color: red;
	  display: inline-block;
	  position: relative;
	  left:200px;
	}
	.blue{
	  height:100px;
	  width:100px;
	  background-color: blue;
	  display: inline-block;
	  position: relative;
	  right:100px;
	}
	.yellow{
	  height:100px;
	  width:100px;
	  background-color: yellow;
	  display: inline-block;
	  position: relative;
	  right:100px;
	}

3. **Absolute**

HTML	**CSS**

<body>
  <div class = "red"> </div>
  <div class = "blue"> </div>
  <div class = "yellow"> </div>
</body>
	.red{
	  height:100px;
	  width:100px;
	  background-color: red;
	  position: absolute;
	  right: 20px;
	}
	.blue{
	  height:100px;
	  width:100px;
	  background-color: blue;
	
	}
	.yellow{
	  height:100px;
	  width:100px;
	  background-color: yellow;
	
	}

Challenge 1: Yellow, Blue, Red (Diagonally corner to corner)

HTML	CSS

<body>
  <div class="red"> </div>
  <div class="blue"> </div>
  <div class="yellow"> </div>
</body>
	body{
	  margin:0;
	}
	
	.red{
	  background-color: red;
	  height: 100px;
	  width: 100px;
	  position: absolute;
	  left:200px;
	
	}
	.blue{
	  background-color: blue;
	  height: 100px;
	  width: 100px;
	  position: absolute;
	  left:100px;
	}
	.yellow{
	  background-color: yellow;
	  height: 100px;
	  width: 100px;
	  position: absolute;
	}

Challenge 2: Yellow, Blue, Red (Side by side)

HTML	CSS (Method1)	CSS(Method 2)

<body>
<div class="red"> </div>
<div class="blue"> </div>
<div class="yellow"> </div>
</body>
	body{
	margin:0;
	}
	
	.red{
	background-color: red;
	height: 100px;
	width: 100px;
	position: absolute;
	margin:200px;
	
	}
	.blue{
	background-color: blue;
	height: 100px;
	width: 100px;
	position: absolute;
	margin:100px;
	}
	.yellow{
	background-color: yellow;
	height: 100px;
	width: 100px;
	position: absolute;
	}
		body{
		margin:0;
		}
		
		.red{
		background-color: red;
		height: 100px;
		width: 100px;
		position: relative;
		left:200px;
		}
		.blue{
		background-color: blue;
		height: 100px;
		width: 100px;
		position: relative;
		left:100px;
		bottom:100px;
		}
		.yellow{
		background-color: yellow;
		height: 100px;
		width: 100px;
		position: relative;
		bottom: 200px;
		}

Relative container with absolute divs within it

HTML	CSS

**<body>
  <div class="container">
    <div class="red"> </div>
    <div class="blue"> </div>
    <div class="yellow"> </div>
  </div>
</body>**
	body{
	  margin:0;
	}
	
	.container{
	  width: 500px;
	  height: 500px;
	  background-color: grey;
	  position: relative;
	}
	
	.red{
	  background-color: red;
	  height: 100px;
	  width: 100px;
	  position: absolute;
	  right: 20px;
	}
	.blue{
	  background-color: blue;
	  height: 100px;
	  width: 100px;
	  position: absolute;
	  left: 100px;
	}
	.yellow{
	  background-color: yellow;
	  height: 100px;
	  width: 100px;
	  position: absolute;
	}

4. Fixed

HTML	CSS

<body>
** <div class="container">**
**   <div class="red"> </div>**
**   <div class="blue"> </div>**
**   <div class="yellow"> </div>**
** </div>**
</body>
	body{
	  margin:0;
	}
	
	.container{
	  width: 500px;
	  height: 500px;
	  background-color: grey;
	  position: relative;
	}
	
	.red{
	  background-color: red;
	  height: 100px;
	  width: 100px;
	  position: absolute;
	  right: 0px;
	}
	.blue{
	  background-color: blue;
	  height: 100px;
	  width: 100px;
	  position: absolute;
	  left: 100px;
	  top:0;
	}
	.yellow{
	  background-color: yellow;
	  height: 100px;
	  width: 100px;
	  position: fixed;
	  top:0;
	}

CSS Static & Relative Positioning:

How things get rendered on the screen?

a. Content is everything

  a. Inline elements: ht & width both are content dependent

  b. Block elements : width 100%, height is dependent on content.

b. Ordering comes from the code

c. Children sit on parents (Introduction of z-axis)

HTML	CSS

<div>
  <h1> Sau<span class="rav">rav</span>
Bhattacharyya </h1>
</div>
	div{
	  height:200px;
	  width: 400px;
	  background-color: peachpuff;
	}
	
	h1{
	  background-color: lightseagreen;
	  padding: 30px;
	  margin: 10px;
	  border: solid 5px;
	}
	
	.rav{
	  background-color: yellow;
	  text-decoration: underline;
	}

CSS FONT STYLING:

2 major font types:

1. Serif (Default font) [Times New Roman is the default font for most browsers]
2. Sans-Serif [Default font is Arial]

Apart from the two major categories, 3 more categories are available:

1. Fantasy (Default font : Papyrus)
2. Monospace
3. Cursive

Good starting point to understand the basic fonts supported by most browsers:
https://www.w3schools.com/css/css_font.asp

Exhaustive list of browser compatible fonts:
https://www.cssfontstack.com/

Custom fonts:
fonts.google.com

1. Choose your desired font
2. Choose and add your desired style based on weight.
3. View Selected Families
4. Copy the pre-populated link in the head section of the html page.
5. Copy the font-family in css.

Custom images and animations:
https://www.flaticon.com/
https://giphy.com/search/code

Font Sizing:
Default font size is 16px.
You can change the font size by the following syntax: **font-size: 90px;**

Because a fixed pixel size is used for font-size, changing browser font settings won't impact our website content.
16px = 100% = 1em
90px = 562.5% = 5.625em

Note : If separate font-size is mentioned both in body as well as h1 in % or em, then the size of the font grows cumulatively. So be aware of where all you are putting the font-size.

Precaution: Use rem (Root em) instead of em;