In this repo, I will share my CSS knowledge as much as possible. I hope this documentation will help you.

master

Branches Tags

anisul-Islam flex layout is added ... on May 15 12

View code

README.md

My CSS Documentation

Total Chapters are following

1. Introduction
2. Selectors & Combinators
3. Typography
4. Box Model
5. Background
6. Layout Design
7. Responsive Web Design
8. Animation
9. How to create/desgin
10. CSS Architecture: BEM Methodology
11. Project

Chapter 1: Introduction

1.1 Introduction to CSS

What is CSS & Why CSS?

- CSS stands for Cascading Style Sheets
- It is used to style **html elements**
- Initial release on December 17, 1996

1.2 Inline CSS

- 3 main ways to add css with html: Inline CSS, Internal CSS, External CSS
- Inline CSS refers to style inside html element. Syntax: `<tagName style="property:value; property:value; ... ">`
- Inline CSS Example is given below:

```
<h1 style="background-color: green">Welcome to CSS</h1>
<p style="color: white; background-color: green">
  aperiam fugiat blanditiis voluptatibus quo!
</p>
<p style="color: white; background-color: green">
  aperiam fugiat blanditiis voluptatibus quo!
</p>
```

1.3 Internal CSS

- Inside `<head>` tag we can use internal css with the help of `<style>` tag
- Internal CSS Syntax:

```
<style>
  selector {
    property: value;
    property: value;
    ...;
  }
</style>
```

- Internal CSS Example is given below; In the example `<p>` tag is a selector:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Learn Internal CSS</title>
    <style>
      p {
```

```
        background-color: green;
        color: white;
      }
    </>
  </head>
  <body>
    <p>Welcome to CSS</p>
    <p>aperiam fugiat blanditiis voluptatibus quo!</p>
  </body>
</html>
```

1.4 Exetrnal CSS

- Inside `<head>` tag we can link the external css file with the help of `<link rel="stylesheet" href="cssFileNameOrAddressHere"/> tag`
- create a css file with an extension of .css as shown below: style.css

```
p {
  background-color: green;
  color: white;
}
```

- then add the css file inside html file as shown below:

```
<head>
  <title>Learn Internal CSS</title>
  <link rel="stylesheet" href="style.css" />
</head>
<body>
  <p>Welcome to CSS</p>
  <p>aperiam fugiat blanditiis voluptatibus quo!</p>
</body>
```

Chapter 2: Selectors & Combinators

2.1 Basic Selectors

- Basic Selectors: Element Selector, grouping selectors, nested selector, Universal Selector, ID selectors, class selectors,
- Element selector: select an element by using its name.
  Example:

```
<head>
  <style>
    h1 {
```

```
      background-color: green;
    }
  </style>
</head>
<body>
  <h1>Bangladesh</h1>
</body>
```

- Grouping selector: select multiple element by using their names separted with comma. Example:

```
<head>
  <style>
    h1,
    h2,
    p {
      background-color: green;
    }
  </style>
</head>
<body>
  <h1>Bangladesh</h1>
  <h2>Bangladesh</h2>
  <p>Bangladesh</p>
</body>
```

- Nested selector: select elements by nesting. ul li a {...}, div p {...} Example:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Learn Internal CSS</title>
    <style>
      ul li a {
        color: green;
      }
    </style>
  </head>
  <body>
    <ul>
      <li><a href="#">Google</a></li>
    </ul>
  </body>
</html>
```

- Universal selector can help to select all the html elements. It is denoted by * Example:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Learn Internal CSS</title>
```

```
    <style>
      * {
        background-color: salmon;
        color: white;
      }
    </style>
  </head>
  <body>
    <h1>Hello CSS</h1>
    <p>aperiam fugiat blanditiis voluptatibus quo!</p>
  </body>
</html>
```

- id selector: is unique inside html document. #id can help to select any element with a given id. use # notation for selecting an id. Example:

```
<head>
  <style>
    #title {
      background-color: green;
    }
  </style>
</head>
<body>
  <h1 id="title">Bangladesh</h1>
</body>
```

- class selector: .class can help to select any element with a given class. use dot notation for selecting a class. Example:

```
<head>
  <style>
    .title {
      background-color: green;
    }
  </style>
</head>
<body>
  <h1 class="title">Bangladesh</h1>
</body>
```

2.2 More on Class & ID Selectors

- we can use multiple class name for an html element such as `<h1 class="style1 style2" > this is something </h1>`

- selecting elements with class name, id name example is given below:

```
<!DOCTYPE html>
<html lang="en">
  <head>
```

```
    <title>Learn Internal CSS</title>
    <style>
      .heading h1 {
        background-color: salmon;
        color: white;
      }
      #heading2 p {
        background-color: green;
        color: white;
      }
    </style>
  </head>
  <body>
    <div class="heading">
      <h1>Hello CSS</h1>
      <p>aperiam fugiat blanditiis voluptatibus quo!</p>
    </div>
    <div id="heading2">
      <h1>Hello CSS</h1>
      <p>aperiam fugiat blanditiis voluptatibus quo!</p>
    </div>
  </body>
</html>
```

2.3 Selectors & Combinators

- Attribute selectors
  - References: https://developer.mozilla.org/en-US/docs/Web/CSS/Attribute_selectors
  - syntax for Attribute selectors

```
/*for attribute name attr.*/
element[attr] {
  property: value;
}

/*for attribute name attr with exactly same value.*/
element[attr='value'] {
  property: value;
}

/* element with "value" anywhere in the url.*/
element[attr*='value'] {
  property: value;
}

/* element with "value" anywhere in the url without case sensitivity.*/
element[attr*='value' i] {
  property: value;
}

/* element end with .value; mainly for link(a) tag.*/
element[attr$='.value'] {
```

```
  property: value;
}
```

- Pseudo class selectors
  - Link Pseudo classes: link, visited, hover, active
  - Input Pseudo classes: focus, enabled, disabled, checked, required, optional, valid, invalid
  - General Pseudo classes: first-child, last-child, first-of-type, last-of-type, nth-child(n), nth-last-child(n), nth-last-of-type(n), root, not
  - References: https://developer.mozilla.org/en-US/docs/Web/CSS/Pseudo-classes
  - syntax for Pseudo class selectors

```
selector:pseudo-class {
  property: value;
}
```

- Pseudo element selectors
  - Common Pseudo element: after, before, first-letter, first-line, placeholder, select
  - References: https://developer.mozilla.org/en-US/docs/Web/CSS/Pseudo-elements
  - syntax for Pseudo element selectors

```
selector::pseudo-element {
property: value;
}
-- Child selectors (div > p)
```

- descendent selectors (div p)
- adjacent selectors (div + p)
- general sibling selectors (div ~ p)

2.4 Pseudo class & Pseudo elements part-1

2.5 Pseudo class & Pseudo elements part-2

- class: hover, focus, nth-child(),
- elements: first-letter, first-line, after, before, selection

2.6 CSS Specificity

- References:
  - https://www.w3.org/TR/selectors-3/#specificity
  - specificity calculator: https://specificity.keegan.st/
- Universal selector (*) specificity - 0

- Count the number of Elements and Pesudo elements (c) - 1
- Count the number of Classes, attributes, Pseudo classes (b) - 10
- Count the number of IDs (a) - 100
- Inline CSS - 1000
- !important - 10000
- How to calculate specificity

```
/* specificity calculator */
/* a=number of id  */
/* b=number of class, pseduo classes, attributes  */
/* b=number of elements, pesudo elemnts, attributes  */

/* a=0 b=0 c=1 === 001 */
h1 {
  background-color: grey;
}

/* a=0 b=1 c=1  === 011 */
h1.heading {
  background-color: blue;
}

/* a=0 b=1 c=0  === 010 */
.heading {
  background-color: green;
}

/* a=1 b=1 c=0  === 100 */
#head {
  background-color: red;
}

/* a=1 b=0 c=1  === 101 */
h1#head {
  background-color: pink;
}

/* a=1 b=1 c=1  === 111 */
h1#head.heading {
  background-color: brown;
}
```

Chapter 3: Typography

3.1 Font Properties

- `font-size: value;` here value can be px/em/rem. 1rem=16px=100%
- `font-weight: value;` here value can be 100/thin, 200/extra light, 300/light, 400/normal, 500/medium, 600/semi-medium, 700/bold, 800/extra bold, 900/black
- `font-style: value;` here value can be italic/normal/oblique
- `font-family: value;` here value can be any valid font name. In the following example paragaph will have Times New Roman as its font; if Times New Roman is not available then Times will be applied and if Times is not available then serif font will be applied. This process is known as fallback.

```
p {
  font-size: 2rem;
  font-weight: bold;
  font-style: italic;
  font-family: 'Times New Roman', Times, serif;
}
```

- Use google font: https://fonts.google.com/

3.2 Color

- `color: value;` here value can be any color names, hexadcimal colors value, RGB(Red, Green, Blue) color value, hsl (Hue, Saturation, Lightness) value
- Color Name: we can use color names directly as shown below:

```
p {
  color: green;
}
```

- RGB: we can use Red, Green, Blue values as shown below:

```
p {
  color: rgb(0, 255, 0);
}
```

- Hexadecimal color: It is a code consist of 6 characters where first 2 characters for Red, Next 2 for Green and last 2 characters for Blue. Example is given below:

```
p {
  color: #00ff00;
  /*we can write one value instead of two similar values*/
  color: #0f0;
}
```

- Important Tools:

- Canva color wheel: https://www.canva.com/colors/color-wheel/
- Color Picker: https://htmlcolorcodes.com/color-picker/
- Image color picker: https://imagecolorpicker.com/en
- How to use colorzilla plugin, how to use https://flatuicolors.com/

3.3 Text styling

- `text-align: value;` here value can be center / left / right / justify
- `text-transform: value;` here value can be uppercase / lowercase / capitalize
- `text-decoration: value;` here value can be underline / overline / line-through / none
- `text-shadow: value;` here value can be x axis, y axis, colorName
- `text-indent: value;`
- `letter-spacing: value;`
- `word-spacing: value;`
- `line-height: value;`
  Example

```
p {
  text-align: justify;
  text-decoration: underline;
  text-transform: uppercase;
  letter-spacing: 0.1rem;
  word-spacing: 0.2rem;
  line-height: 1rem;
  text-shadow: 0.1rem 0.1rem green;
}
```

3.3 Icon & emoji styling

- Get emoji from here: https://unicode-table.com/en/
- Get icon from here: https://www.iconfinder.com/

```
<style>
  span {
    color: red;
    font-size: 2rem;
  }
</style>

<p>I <span> ♥ </span> Bangladesh</p>
```

- How to use font awesome icons

- get font awesome icons here: https://fontawesome.com/
- get font awesome cdn from here: https://cdnjs.com/libraries/font-awesome
- add the font awesome cdn inside the html head tag and then you are ready to use font awesome icons
  Example

```
<i class="far fa-address-card"></i>
<i style="color: red;" class="far fa-address-card fa-2x"></i>
```

Chapter 4: Box Model

4.1 Margin

- margin-top, margin-right, margin-bottom, margin-left
- margin

4.2 Padding

- padding-top, padding-right, padding-bottom, padding-left
- padding

4.3 border

- border-top, border-right, border-bottom, border-left
- `border: borderWidth borderColor borderStyle;`
  - example: `border: 1px green solid;`
  - border-style
    - border-top-style
    - border-right-style
    - border-bottom-style
    - border-left-style
  - border-width
    - border-top-width
    - border-right-width
    - border-bottom-width
    - border-left-width
  - border-color
    - border-top-color
    - border-right-color
    - border-bottom-color
    - border-left-color

4.4 box model

- content, padding, border, margin

4.5 box sizing & opacity

- box-sizing: border-box
- opacity: value; value can be between 0-1

4.6 Inline, block element, width, max-width

- `display: inline/block`

4.7 overflow

- `overflow: value` ; here default value is visible, hidden, auto, scroll

Chapter 5: Background

5.1 How to set background image in webpage

- background-image, background-position, background-size, background-repeat, background-attachment
- shorthand: `background: bg-image position/bg-size bg-repeat bg-attachment bg-origin bg-clip`
- example

```
body {
  height: 80vh;
  background-image: url('./images/me.JPG');
  background-position: center center;
  background-size: cover;
  background-repeat: no-repeat;
  background-attachment: fixed;
  background-origin: padding-box;
  background-clip: border-box;
  background-color: #ccc;
}
```

5.2 gradient-linear/radial

- background: linear-gradient(direction, colors)
- example

```
.banner {
  width: 400px;
```

```
  height: 400px;
  background: linear-gradient(to right, green, orange);
}
```

- background: radial-gradient(style-type, colors)
- example

```
.banner {
  width: 400px;
  height: 400px;
  background: radial-gradient(circle, green, orange);
}
```

Chapter 6: Layout design

6.1 float

- float: left/right
- clear: left/right/both
- example: create 3 div in html and add div1, div2, div3 classes with them

```
.div1 {
  width: 50%;
  height: 10rem;
  background-color: orange;
  float: left;
}
.div2 {
  width: 50%;
  height: 10rem;
  background-color: plum;
  float: left;
}
.clear-div {
  clear: both;
}
.div3 {
  height: 10rem;
  background-color: burlywood;
}
```

6.2 Position

- `position: static(default)/absolute/relative/fixed/sticky`
- make sure to use top, right, bottom, left property with position property

6.3 z-index & css variables

- z-index helps us to maintain the order of stacked elements
- `z-index: value` ; value can be negative or positive
- to declare a varibale use the following syntax

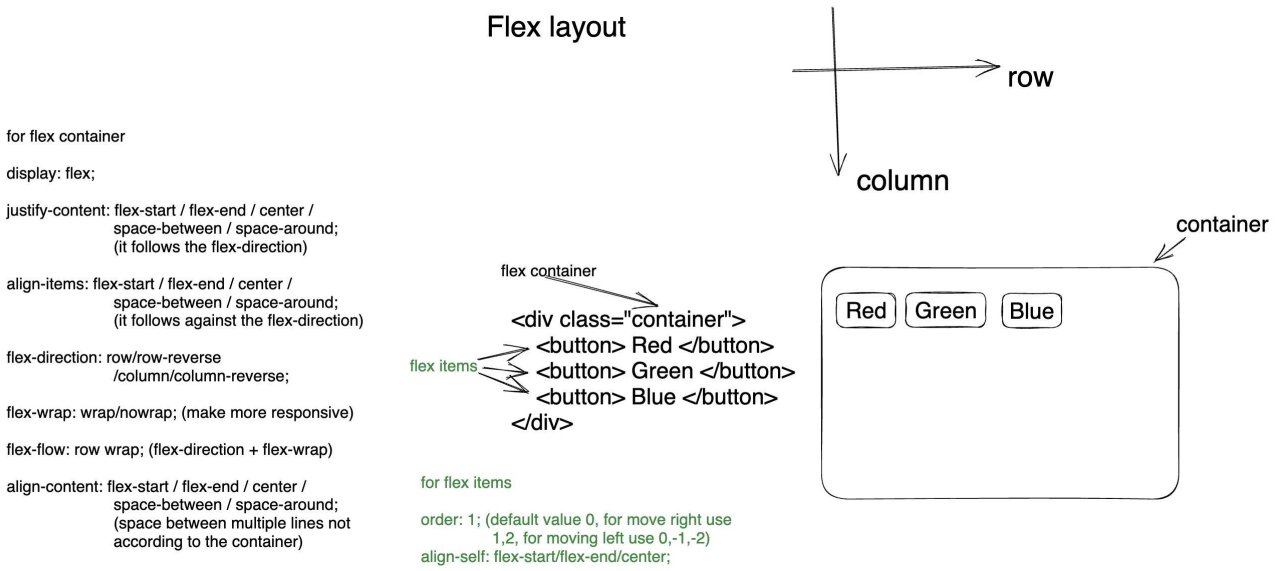
```
<!-- how to declare a valriable -- > :root {
  --variable-name: value;
}
<!-- how to use css valriable -- > selector {
  property: var(--variable-name, fallback-value);
}
```

- example of css variables: make sure to create 2 html div with class div1, div2

```
:root {
  --primary-color: black;
  --secondary-color: green;
}
.div1 {
  width: 10rem;
  height: 10rem;
  background-color: black;
  background-color: var(--primary-color);
  position: absolute;
  z-index: 1;
}

.div2 {
  width: 10rem;
  height: 10rem;
  background-color: green;
  background-color: var(--secondary-color);
  position: absolute;
  left: 5rem;
  top: 5rem;
}
```

6.4 flexbox layout



- flex layout learning game: https://flexboxfroggy.com/
- example

```
.flex-container {
  display: flex;
  flex-direction: column/column-reverse/row/row-reverse;
  flex-wrap: wrap/no-wrap;
  justify-content: flex-start/flex-end/center/space-between/space-around;
  align-items: flex-start/flex-end/center/space-between/space-around;
}
.flex-item1 {
  order: 2;
  flex-basis: 30%;
  flex: 1;
}
.flex-item2 {
  order: 1;
  flex-basis: 70%;
  flex: 2;
}
```

6.5 text-shadow and box-shadow

- `text-shadow: x-value y-value blur-value color`
- `box-shadow: x-value y-value color`
- `box-shadow: x-value y-value blur-radius color`
- `box-shadow: inset x-value y-value color`

6.6 How to design a card

6.7 Grid Layout part-1

- example

```
.grid-container {
  display: grid;
  grid-template-columns: auto auto auto;
  <!-- grid-template-rows: 120px 110px 40px; -->
  <!-- grid-column-gap: 10px;
  grid-row-gap: 10px; -->
  grid-gap: 10px;

}
.grid-item1{
  grid-column-start: 1;
  grid-column-end: 3;
  grid-column: 1 / 3;
  grid-column: 1 / span 3;
  grid-row-start: 1;
  grid-row-end: 3;
  grid-row: 1 / 3;
  grid-row: 1 / span 3;
}
```

6.8 Grid Layout part-2

- example 1

```
<head>
  <style>
    .grid-container {
      display: grid;
      grid-template-columns: auto auto auto auto auto auto;
    }
    header {
      background-color: chocolate;
      grid-column: 1/7;
    }

    nav {
      background-color: cornflowerblue;
      grid-column: 1/2;
    }

    main {
      background-color: cornsilk;
      grid-column: 2/5;
    }
    aside {
      background-color: aqua;
      grid-column: 5/7;
    }
    footer {
      background-color: burlywood;
      grid-column: 1/7;
    }
  </style>
</head>
```

```
<body>
  <div class="grid-container">
    <header>
      <p>Header</p>
    </header>
    <nav>
      <p>Menu</p>
    </nav>
    <main>
      <p>Main</p>
    </main>
    <aside>
      <p>Aside</p>
    </aside>
    <footer>
      <p>footer</p>
    </footer>
  </div>
</body>
```

- example 2

```
<head>
  <style>
    .grid-container {
      display: grid;
      grid-template-areas:
        'header header header header header header'
        'nav main main main aside aside'
        'footer footer footer footer footer footer';
    }
    header {
      background-color: chocolate;
      grid-area: header;
    }

    nav {
      background-color: cornflowerblue;
      grid-area: nav;
    }

    main {
      background-color: cornsilk;
      grid-area: main;
    }
    aside {
      background-color: aqua;
      grid-area: aside;
    }
    footer {
      background-color: burlywood;
      grid-area: footer;
    }
  </style>
</head>
```

```
<body>
  <div class="grid-container">
    <header>
      <p>Header</p>
    </header>
    <nav>
      <p>Menu</p>
    </nav>
    <main>
      <p>Main</p>
    </main>
    <aside>
      <p>Aside</p>
    </aside>
    <footer>
      <p>footer</p>
    </footer>
  </div>
</body>
```

6.9 Grid Layout part-3

Chapter 7: Responsive web design (RWD)

7.1 Introduction to RWD

- Use box-sizing `box-sizing: border-box`
- Use media query
- Use media end points

7.2 Responsive navigation menu

7.3 Responsive column design

7.4 Responsive web design using grid view part-1

7.5 Responsive web design using grid view part-1

Chapter 8: Animation

8.1

8.2

html and css basic setup

```
<header class="center">
  <div class="header__circle center">Hello guys good morning!</div>
</header>
```

```
.center {
  display: flex;
  justify-content: center;
  align-items: center;
}
header {
  height: 100vh; /* display: flex; justify-content: center; align-items: center;
*/
  background-color: green;
}
.header__circle {
  width: 20rem;
  height: 20rem;
  background-color: brown;
  color: white; /* display: flex; justify-content:
center; align-items: center; */ /* making circle */
  border-radius: 50%; /*
triangel shape width: 0; height: 0; border-left: 50px solid transparent;
border-right: 50px solid transparent; border-bottom: 100px solid #32557f; */
}
```

transition property

- transition properties
  - transition-property
  - transition-duration
  - transition-delay
  - transition-timing-function
  - transition [some animations can be attractive however sometime they can cause accessibility issues and also cause migraine]

```
/* transition-property: background-color;
transition-duration: 1s;
transition-timing-function: linear;
transition-delay: 2s;
*/

<!-- shrothand  -->
transition: background-color 1s;
```

```
transition: background-color 1s linear;
transition: background-color 1s linear 0.5s;

/* default value; slow down at the end */
transition-timing-function: ease;

/* starts of slowly but then transition speed get fast */
transition-timing-function: ease-in;

/* starts of fast but then transition speed gets slow */
transition-timing-function: ease-out;

/* transition at an even speed */
transition-timing-function: linear;

/* A Cubic Bezier curve is defined by four points P0, P1, P2, and P3. */
/* P1 and P3 are the start and the end of the curve */
/* p1 and p3 values must be in the range of 0 to 1. */
/* it can be used with transition and animation  */
/* https://cubic-bezier.com/#.17,.67,.83,.67 */
transition-timing-function: cubic-bezier(p1, p2, p3, p4);
```

transform property

- transform property has 4 differnt values
  - transform: scale(number)
  - transform: rotate(degree)
  - transform: translate(x,y)
  - transform: skew(degree) / skewX(degree) / skewY(degree)

[we can also use multiple transform property together like: transform: translate() rotate()]

```
.header__circle {
  width: 20rem;
  height: 20rem;
  background-color: brown;
  color: white;

  /* making circle  */
  border-radius: 50%;

  /* making beautiful style  */
  border-radius: 220px 220px 40px 50px;

  transition: all 0.3s linear;
}

/* lets add some transform properties here  */
.header__circle:hover {
  background-color: orange;
  /* transform: scale(1.2); */
```

```
  /* transform: translate(0px, -340px); */
  transform: rotate(25deg);
}
```

- skew example

```
.center {
  display: flex;
  justify-content: center;
  align-items: center;
}

header {
  height: 100vh;
}

.header__circle {
  width: 20rem;
  height: 20rem;
  background-color: brown;
  color: white;

  transform: skew(-5deg);
  transition: all 0.3s linear;
  border-radius: 0.6rem;
}

/* lets add some transform properties here  */
.header__circle:hover {
  background-color: orange;
  /* transform: scale(1.2); */
  /* transform: translate(0px, -340px); */
}
```

animation property

- example

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Document</title>
    <style>
      .container {
        min-height: 100vh;
        display: grid;
        place-items: center;
      }
      .circle-div {
        width: 100px;
        height: 100px;
        background-color: chocolate;
        border-radius: 50%;
```

```
        animation-name: circle-anim;
        animation-duration: 2s;
        animation-fill-mode: forwards;
        animation-iteration-count: infinite;
        animation-timing-function: linear;
        position: relative;
      }

      @keyframes circle-anim {
        0% {
          background-color: chocolate;
          top: 0;
          left: 0;
        }
        25% {
          background-color: chocolate;
          top: -100px;
          left: 0;
        }
        50% {
          background-color: chocolate;
          top: 0px;
          left: 0;
        }
        75% {
          background-color: chocolate;
          top: 100px;
          left: 0;
        }
        100% {
          background-color: rgb(30, 210, 60);
          top: 0;
        }
      }
    </style>
  </head>
  <body>
    <div class="container">
      <div class="circle-div"></div>
    </div>
  </body>
</html>
```

8.3 transition and transform

- Important transition properties: `transition-property, transition-duration, transition-delay, transition-timing-function, transition`

8.4 Animated progress bar

Chapter 9: How to create/design

9.1 How to design a navigation menu

- example

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Document</title>
  </head>
  <body>
    <div id="nav-menu">
      <ul>
        <li><a href="#">Home</a></li>
        <li><a href="#">Tutorials</a></li>
        <li><a href="#">About</a></li>
        <li><a href="#">Contact</a></li>
      </ul>
    </div>
  </body>
</html>
```

9.2 How to center elements

- using flex

```
.container {
  width: 30rem;
  height: 30rem;
  background-color: chocolate;
  display: flex;
  justify-content: center;
  align-items: center;
}
.child {
  width: 50px;
  height: 50px;
  background-color: burlywood;
}
```

- using grid

```
.container {
  width: 30rem;
  height: 30rem;
  background-color: chocolate;
  display: grid;
  place-items: center;
}
.child {
```

```
  width: 50px;
  height: 50px;
  background-color: burlywood;
}
```

- using position

```
.parent-div {
  width: 30rem;
  height: 30rem;
  background-color: chocolate;
  position: relative;
}
.child-div {
  width: 50px;
  height: 50px;
  background-color: burlywood;
  position: absolute;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
}
```

9.3 How to create linable icon button

9.4 How to create drop down menu

9.5 How to design a table

- create a basic table first and then start styling
- Example:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <title>Document</title>
    <style>
      table {
        border-collapse: collapse;
        height: 300px;
        width: 300px;
      }
      td,
      th {
        border: 1px solid black;
        padding: 5px;
        text-align: center;
        vertical-align: middle;
      }
```

```
      th {
        background-color: darkgreen;
        color: white;
        height: 30px;
        font-size: 18px;
      }
      tr:nth-child(odd) {
        background-color: gray;
      }
      tr:nth-child(even) {
        background-color: sandybrown;
      }
      tr:hover {
        background-color: tomato;
      }
    </>
  </head>
  <body>
    <table>
      <caption>
        Student details
      </caption>
      <thead>
        <tr>
          <th scope="col">ID</th>
          <th scope="col">Name</th>
          <th scope="col">GPA</th>
        </tr>
      </thead>
      <tbody>
        <tr>
          <td>101</td>
          <td>Anis</td>
          <td>3.92</td>
        </tr>
        <tr>
          <td>102</td>
          <td>Rasel</td>
          <td>3.44</td>
        </tr>
        <tr>
          <td>103</td>
          <td>Kolpona</td>
          <td>2.44</td>
        </tr>
      </tbody>
    </table>
  </body>
</html>
```

9.6 How to design a form part-1

9.7 How to design a form part-2

- form elements styling example

```css
input[type='text'] {
  box-sizing: border-box;
  width: 50%;
  padding: 0.5rem 1rem;
  font-size: 1rem;
  margin: 1rem 0;
  border: 0.3rem solid orange;
  border-radius: 0.5rem;
}

button {
  background-color: sandybrown;
  border: none;
  border-radius: 0.5rem;
  color: white;
  cursor: pointer;
  font-size: 1.5rem;
  padding: 2rem 1rem;
  width: 10rem;
}

select {
  background-color: sandybrown;
  padding: 1rem;
  border: none;
  border-radius: 0.5rem;
}

textarea {
  resize: none;
  width: 50rem;
  padding: 1rem;
  border: 0.3rem solid black;
  border-radius: 0.5rem;
}
```

Chapter 10: CSS Architecture: BEM Methodology - https://github.com/anisul-Islam/bem-methodology

Chapter 11: Project

Project 1 - CV Project

- Project-1 - CV Project part-1
- Project-1 - CV Project part-2
- Publish a website on github

Project 2 - Calculator Project

- Project-2- Calculator Project part-1
- Project-2- Calculator Project part-2

Project 3 - Portfolio Project

- Project-3- Portfolio Project part-1
- Project-3- Portfolio Project part-2
- Project-3- Portfolio Project part-3
- Project-3- Portfolio Project part-4
- Project-3- Portfolio Project part-5
- Project-3- Portfolio Project part-6
- Project-3- Portfolio Project part-7
- Project-3- Portfolio Project part-8
- Project-3- Portfolio Project part-9

Project 4 - Restaurant Project

- Project-4- Restaurant Project part-1
- Project-4- Restaurant Project part-2
- Project-4- Restaurant Project part-3

Project 5 - Blog website Project

- Project-5 Blog website Project part-1
- Project-5 Blog website Project part-2
- Project-5 Blog website Project part-3
- Project-5 Blog website Project part-4
- Project-5 Blog website Project part-5
- Project-5 Blog website Project part-6
- Project-5 Blog website Project part-7
- Project-5 Blog website Project part-8
- Project-5 Blog website Project part-9

Releases

No releases published

Packages

No packages published