

# Module 8 (advance CSS)

## Module 8.1 (CSS Display)

The `display` property specifies the display behavior (the type of rendering box) of an element.

There are 4 display property values :-

1. Block
2. Inline
3. Inline-block
4. none

### Display: Block

displays element as a block element (like p tag) it starts on new line and takes up the whole width of the screen

note : *Height and width of that element are changeable here.*

note : *no matter how much we change the width, the block element will never come to the next line.*

```
<head>
<style>

body {
  background-color:#E7E9EB;
}
#myDIV {
  height:300px;
  background-color:#FFFFFF;
}
#myDIV span {
  background:lightblue;
  display:block;
}
</style>

</head>
<body>

<h1>The direction property</h1>

<div id="myDIV">
<p>A demonstration of how to set the display value for an element:</p>

<p>Inside this paragraph, we have placed a <span>little blue SPAN</span>
```

### The direction property

A demonstration of how to set the display value for an element:

Inside this paragraph, we have placed a  
little blue SPAN  
element.

```
<head>
<style>

body {
  background-color:#E7E9EB;
}
#myDIV {
  height:300px;
  background-color:#FFFFFF;
}
#myDIV span {
  height:100px;
  width: 200px;
  background:lightblue;
  display:block;
}
</style>

</head>
<body>

<h1>The direction property</h1>

<div id="myDIV">
<p>A demonstration of how to set the display value for an element:</p>
```

### The direction property

A demonstration of how to set the display value for an element:

Inside this paragraph, we have placed a  
little blue SPAN  
element.

## Display: Inline

displays element as inline element (like span), and goes in line with the previous element.

note : *any height and width property will have no effect on the element.*

```
<head>
<style>

body {
  background-color:#E7E9EB;
}
#myDIV {
  height:300px;
  background-color:#FFFFFF;
}
#myDIV span {
  background:lightblue;
  display:inline;
}
</style>

</head>
<body>

<h1>The direction property</h1>

<div id="myDIV">
<p>A demonstration of how to set the display value for an element:</p>

<p>Inside this paragraph, we have placed a <span>little blue SPAN</span>
```

### The direction property

A demonstration of how to set the display value for an element:

Inside this paragraph, we have placed a little blue SPAN element.

```
<head>
<style>

body {
  background-color:#E7E9EB;
}
#myDIV {
  height:300px;
  background-color:#FFFFFF;
}
#myDIV span {
  height: 100px;
  width: 1000px;
  background:lightblue;
  display:inline;
}
</style>

</head>
<body>

<h1>The direction property</h1>

<div id="myDIV">
<p>A demonstration of how to set the display value for an element:</p>
```

*still no change* →

### The direction property

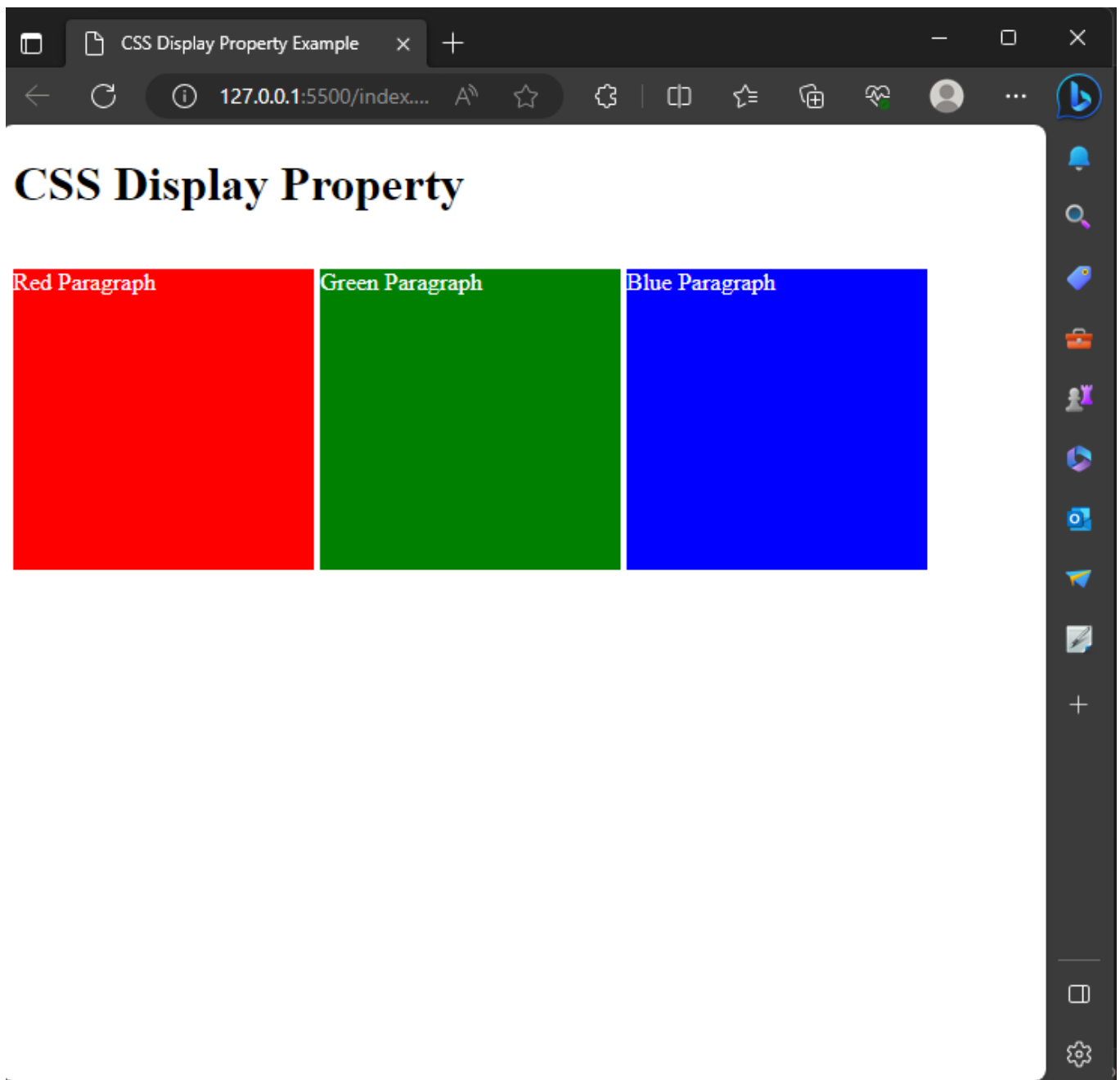
A demonstration of how to set the display value for an element:

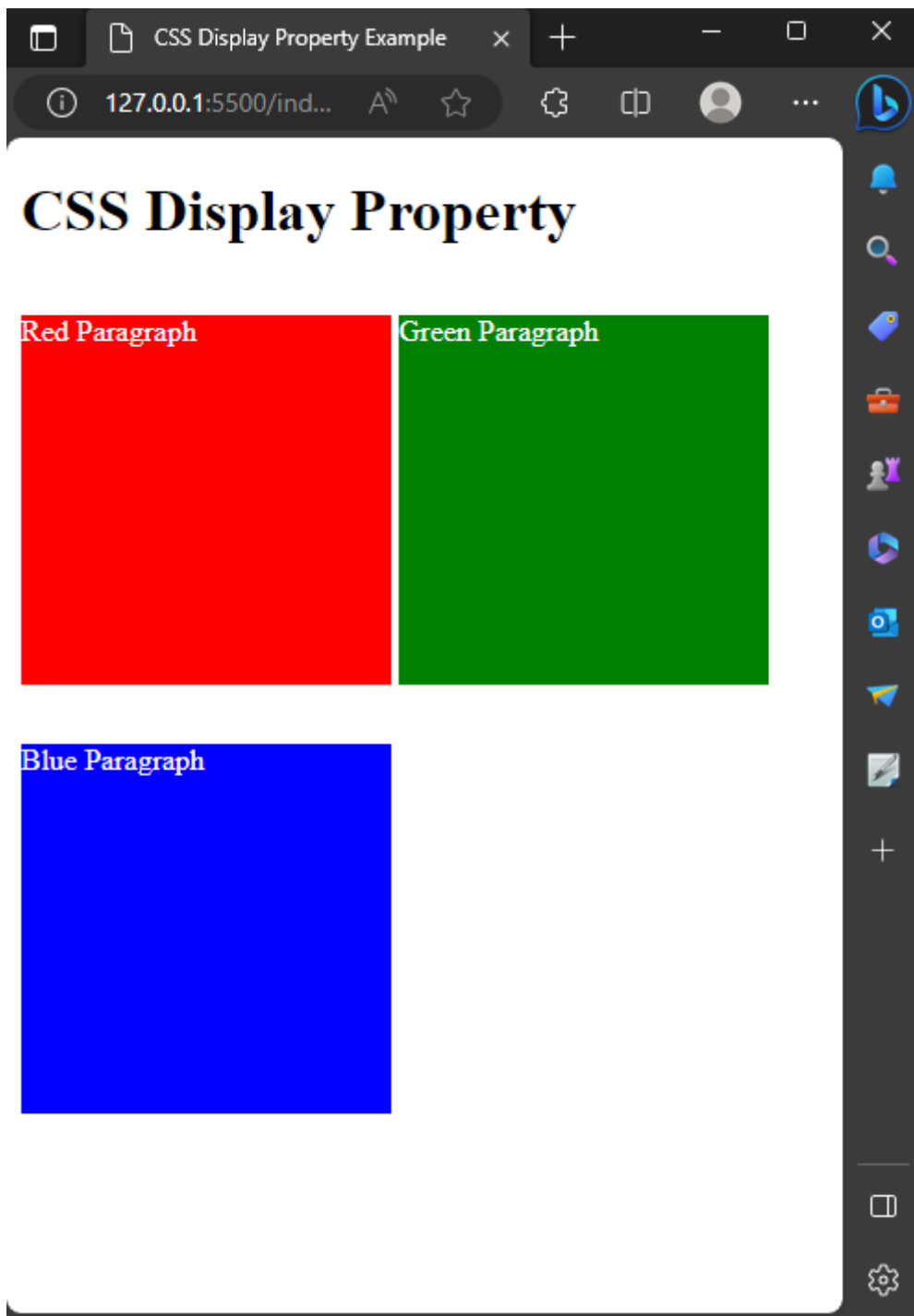
Inside this paragraph, we have placed a little blue SPAN element.

## Display: Inline-block

Displays an element as an inline-level block container. The element itself is formatted as an inline element, but you can apply height and width values

note : suppose 3 div's have a display property set to `inline-block` , then they will go straight in a horizontal line, but if we reduce the size of screen then the divs will go into vertical order





note : When we apply height and width to the element, then it will be changed, and suppose width is set to be more than the screen size, then in that case the element's on right of the `inline-block`

element will go to the next line

```
<head>
<style>

body {
  background-color:#E7E9EB;
}
#myDIV {
  height:300px;
  background-color:#FFFFFF;
}
#myDIV span {
  background:lightblue;
  display:inline-block;
}
</style>

</head>
<body>

<h1>The direction property</h1>

<div id="myDIV">
<p>A demonstration of how to set the display value for an element:</p>

<p>Inside this paragraph, we have placed a <span>little blue SPAN</span>
element.</p>
</div>
</body>
```

## The direction property

A demonstration of how to set the display value for an element:

Inside this paragraph, we have placed a little blue SPAN element.

```
<head>
<style>

body {
  background-color:#E7E9EB;
}
#myDIV {
  height:300px;
  background-color:#FFFFFF;
}
#myDIV span {
  height: 100px;
  width: 200px;
  background:lightblue;
  display:inline-block;
}
</style>

</head>
<body>

<h1>The direction property</h1>

<div id="myDIV">
<p>A demonstration of how to set the display value for an element:</p>
```

## The direction property

A demonstration of how to set the display value for an element:

Inside this paragraph, we have placed a little blue SPAN element.

```
<head>
<style>

body {
  background-color:#E7E9EB;
}
#myDIV {
  height:300px;
  background-color:#FFFFFF;
}
#myDIV span {
  height: 100px;
  width: 2000px;
  background:lightblue;
  display:inline-block;
}
</style>

</head>
<body>

<h1>The direction property</h1>

<div id="myDIV">
<p>A demonstration of how to set the display value for an element:</p>
```

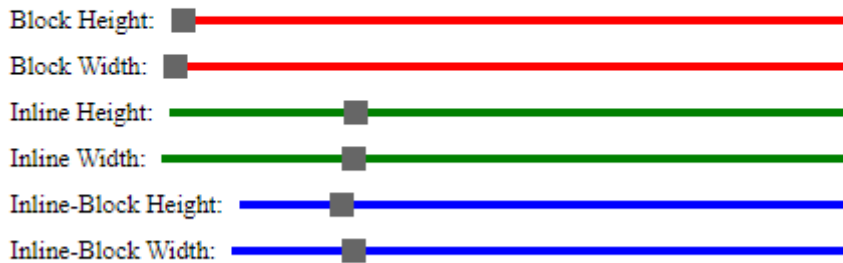
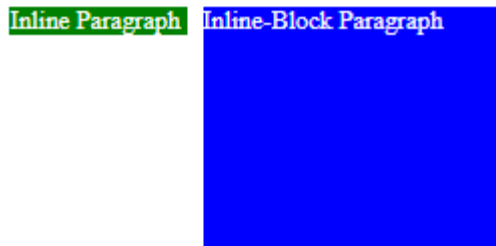
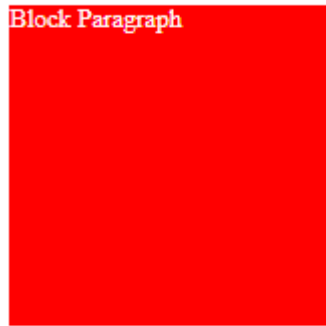
## The direction property

A demonstration of how to set the display value for an element:

Inside this paragraph, we have placed a little blue SPAN element.

note : if width of inline-block element is more then the screen then this element will come to the next line

# CSS Display Property



## Display: None

In this case the element will hide and not be visible.

**Que - Suppose we have a para inside div, now how can we place it in absolute center of the div ?**

```
<div>  
    <p> Some Text </p>  
</div>
```



Solution :-

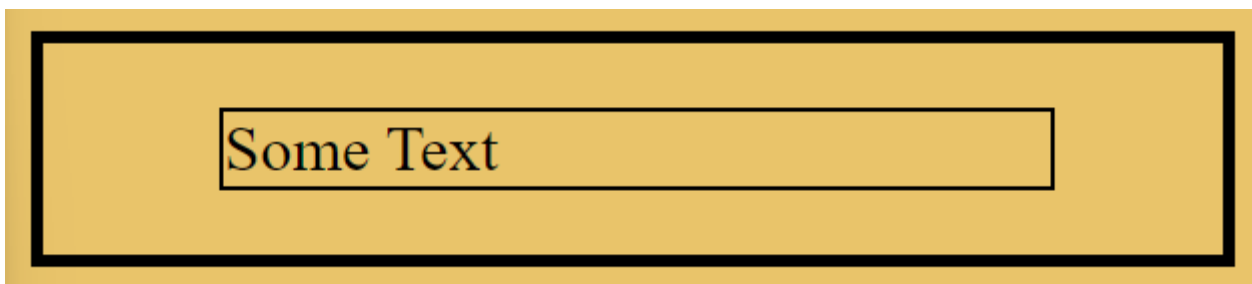
Way-1: using `margin` & `width` (not the best way)

set p's width to any percent eg. 70% of div

and now the left space is 30%, so if we give p's `margin-left` and `margin-right` to 15%(of div), then p will be at center of div.

```
/* method-1 (using margin)*/
div{

}
p{
  width:70%;
  margin-left: 15%;
  margin-right: 15%;
}
/*
to center the text set p's text-align: center
*/
```

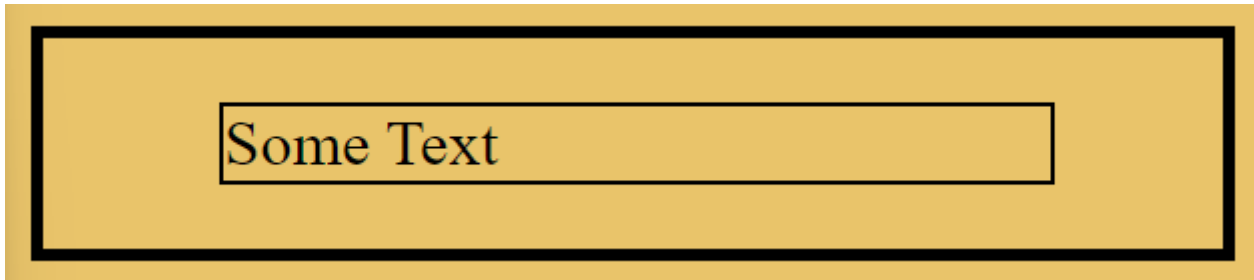


★ Way-2): using `position: absolute` (`left`, `right`, `top`, `bottom`) properties

similarly instead of padding, we could also use `position: absolute` (for p) and `position: relative` (for div) then push the p a 15%(of div) from the left and right.

this is the right way.

```
div{  
  position: relative;  
}  
p{  
  width: 70%; /*rest : 30% of div*/  
  position: absolute;  
  left: 15%;  
  right: 15%;  
}
```



---

## Module 8.2 (CSS Float & Clear property)

**Que - How to make text wrap around our image ?**

lets say we have a `img` and a `p` tag so we want our text to wrap around our img, then if we use `display: inline` on both of them then the text will just go at the right end of the img, and will not wrap around, so we can use the `float: left` here to let the image float on the left of its parent element (html body in this case).


values of float : `float: left`, `float: right`



eg. if we used the display: inline then :-

```
try.html > html
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1">
6   <title>Document</title>
7   <style>
8     img{
9       display: inline;
10    }
11    p{
12      display: inline;
13    }
14  </style>
15 </head>
16 <body>
17   
18   <p>
19     Lorem ipsum dolor sit amet consectetur adipiscing elit.
20   </p>
21 </body>
22 </html>
```

WebDev - Bookmark | img to link generat... | www.grabon.in | Whiteboard for Onl...




consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat erat quisque. Pellentesque dapibus efficitur laoreet. Nam risus ante, dapibus a molestie consequat, ultrices ac magna. Fusce dui lectus, congue vel laoreet ac, dictum quis, velit. Suspendisse elit. Moris euismod in nunc malesuada ultricies. Donec nunc elit, tincidunt ac, conubia non, nisi, nunc. Nullam diam orci, tempor eu, euismod in, tempus eu, nisi. Donec nunc elit, tincidunt ac, conubia non, nisi, nunc. Nullam diam orci, tempor eu, euismod in, tempus eu, nisi. Donec nunc elit, tincidunt ac, conubia non, nisi, nunc. Nullam diam orci, tempor eu, euismod in, tempus eu, nisi.

if we used the float : left property.

```
try.html > html > head > style > img
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1">
6   <title>Document</title>
7   <style>
8     img{
9       float: left;
10      margin: 10px;
11    }
12  </style>
13 </head>
14 <body>
15   
16   <p>
17     Lorem ipsum dolor sit amet consectetur adipiscing elit.
18   </p>
19 </body>
20 </html>
```

WebDev - Bookmark | img to link generat... | www.grabon.in | Whiteboard for Onl...



consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat erat quisque. Pellentesque dapibus efficitur laoreet. Nam risus ante, dapibus a molestie consequat, ultrices ac magna. Fusce dui lectus, congue vel laoreet ac, dictum quis, velit. Suspendisse elit. Moris euismod in nunc malesuada ultricies. Donec nunc elit, tincidunt ac, conubia non, nisi, nunc. Nullam diam orci, tempor eu, euismod in, tempus eu, nisi. Donec nunc elit, tincidunt ac, conubia non, nisi, nunc. Nullam diam orci, tempor eu, euismod in, tempus eu, nisi. Donec nunc elit, tincidunt ac, conubia non, nisi, nunc. Nullam diam orci, tempor eu, euismod in, tempus eu, nisi.

more examples :



Another scenario :- let say we have 3 elements, `img`, `p`, `footer` now here we have our image float : left, now our result will look like this :-

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <style>
    img{
      float: left;
    }
  </style>
</head>
<body>

  
  <p>
    Lorem ipsum dolor sit amet consectetur adipisicing
  </p>
  <footer style="background-color: #248; color: white;">
    © 2023 document.org
  </footer>

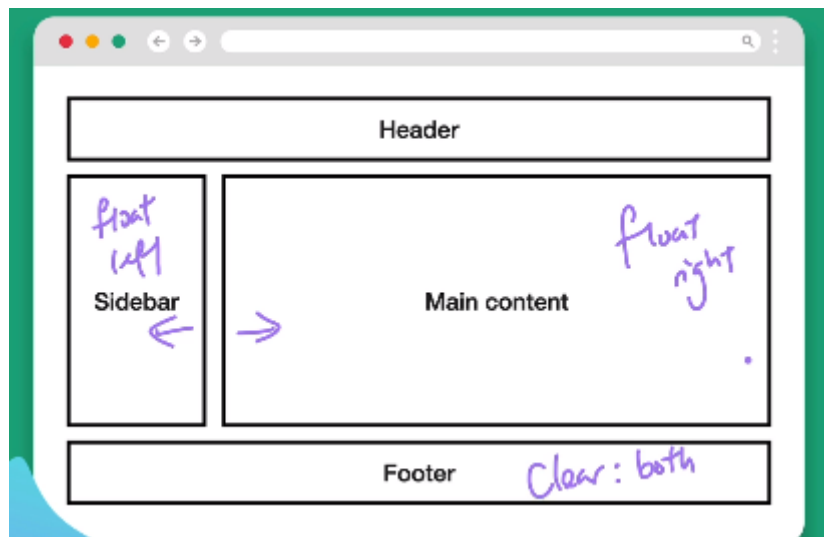
</body>
</html>
```

Lorem ipsum dolor sit amet consectetur adipisicing elit. Aperiam laboriosam cumque aut dignissimos, similique molestiae expedita at, vitae iste eius consectetur. Officiis perspicatis tenetur iusto eaque ipsa, earum natus consequuntur ipsum recusandae? Sit voluptates sed quaeat nam omnis corrupti laudantium, maiores explicabo iste consequatur placeat adipisci, labore ipsam illo cum perspicatis. Quas mollitia officiis incidunt repellendus sapiente, dicta dignissimos saepe porro quis. Error voluptatibus similique, voluptatum beatae nihil maxime molestiae dignissimos fuga veniam sequi officiis delectus omnis rerum necessitatibus exercitationem sapiente modi, quibusdam corporis. Fuga aspernatur unde accusamus at odio explicabo atque architecto, vel iusto quisquam illo aliquid nisi facilis.

© 2023 document.org

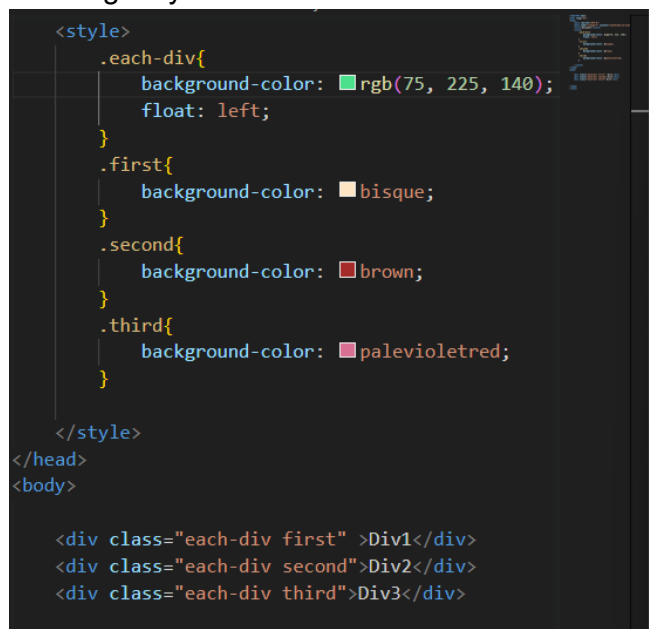
because the `float left` is applied to `img`, now every other element will wrap around it, but we do not want our footer to wrap, because footer always comes at the end (capturing all the width of screen) so in this case we can use the `clear: left, right, both` property which will allow the 'footer' to escape the mentioned float and will not wrap around.

like this :-



**Que - Difference between float & display:inline on multiple divs ?**

When we set all 3 div's to float left then they will come in line with each other, but note that they have no margin by default.



Div1Div2Div3

But instead of the float, if we used the `display: inline` then all 3 became inline element but with some default margin :-

```
<style>
  .each-div{
    background-color: rgb(75, 225, 140);
    display: inline;
  }
  .first{
    background-color: bisque;
  }
  .second{
    background-color: brown;
  }
  .third{
    background-color: palevioletred;
  }
</style>
</head>
<body>

  <div class="each-div first" >Div1</div>
  <div class="each-div second">Div2</div>
  <div class="each-div third">Div3</div>
```



---

## Module 8.3 (How to make Website Responsive)

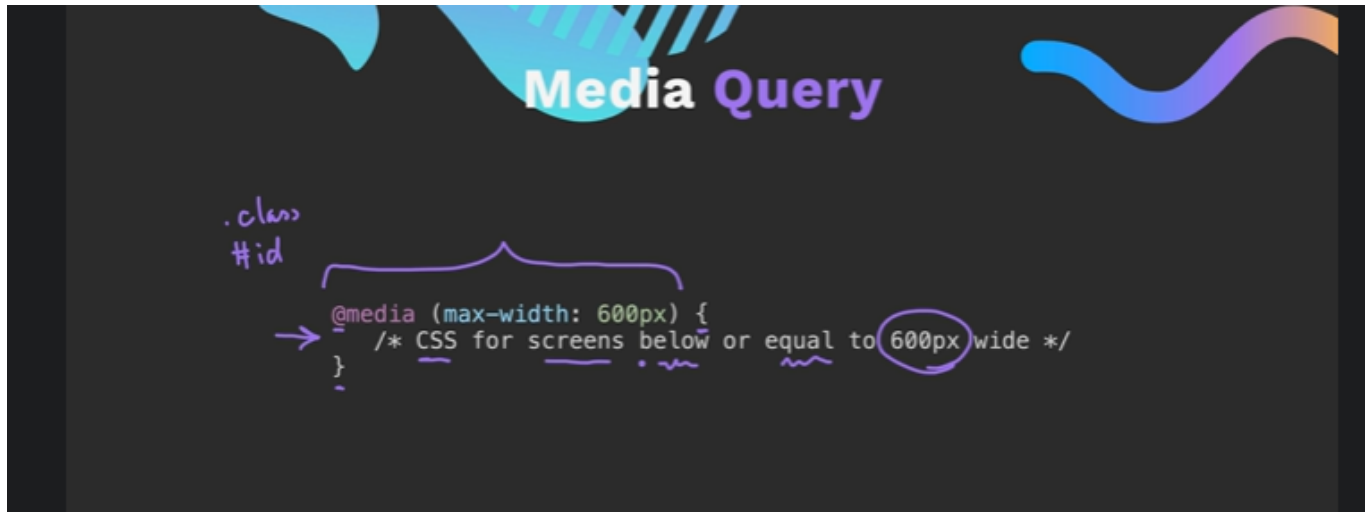
Responsiveness of a website means, that on different devices of different screen sizes our website should also adapt.

There are 4 ways to do that :-

1. Media Queries
2. CSS Grid
3. CSS Flexbox
4. External Framework (eg. bootstrap)

### Media Queries

In this we specify different CSS for different screen sizes



CSS Grid

It is used for 2d designs in a webpage

```
.card{
  background-color: #993333;
}

.grid-container{
  display: grid; /*for 2d*/
  grid-template-columns: 1fr 1fr 1fr; /* 3 cols*/
  grid-template-rows: 100px 100px 100px; /* 2 rows*/
  gap: 10px;
}

.first{
  grid-column: span 2;
}

</style>
</head>

<body>
  <h1>CSS Grid</h1>
  <div class="grid-container">
    <div class="first card"></div>
    <div class="card"></div>
    <div class="card"></div>
    <div class="card"></div>
    <div class="card"></div>
    <div class="card"></div>
  </div>
```

## CSS Grid



miro

## CSS Flexbox

It is used for 1d design (either horizontal or vertical)

## CSS Flexbox

```
<div class="flex-container">
  <div class="first card"></div>
  <div class="second card"></div>
  <div class="card"></div>
  <div class="card"></div>
</div>
```

```
.flex-container {
  display: flex;
}

.card {
  background: blue;
  border: 30px solid white;
  height: 100px;
  flex: 1;
}

.first {
  flex: 2;
}

.second {
  flex: 0.5;
}
```




## Bootstrap

it is called framework because it is external, not present in html or css itself.

Bootstrap has a 12 division flexbox system i.e entire screen is divided into 12 columns of equal size.

## Bootstrap Framework



```
<div class="container">
  <div class="row">
    <div class="card col-6">
      Card
    </div>
    <div class="card col-2">
      Card
    </div>
    <div class="card col-4">
      Card
    </div>
  </div>
</div>
```

## Module 8.4 (Media Queries)

The `@media` rule, introduced in CSS2, made it possible to define different style rules for different media types.

To target screens from 0px to 100px width (inclusive)

```
@media (max-width: 100px){  
  body{  
    color: seablue;  
  }  
}
```

To target screens from 100px to 300px width (inclusive)

```
@media (min-width: 100px) and (max-width: 300px){  
  body{  
    color: seablue;  
  }  
}
```

To target screens from 0 to 300px and 400px to max width (inclusive)

```
@media (max-width: 300px) and (min-width: 400px){  
  body{  
    color: seablue;  
  }  
}
```

---

**Que - How to create a card like a newspaper?**

Solution :-

---

**Que - How to create a attractive navigation bar, and center its anchor and division tags ?**

Solution :-

example1 :-

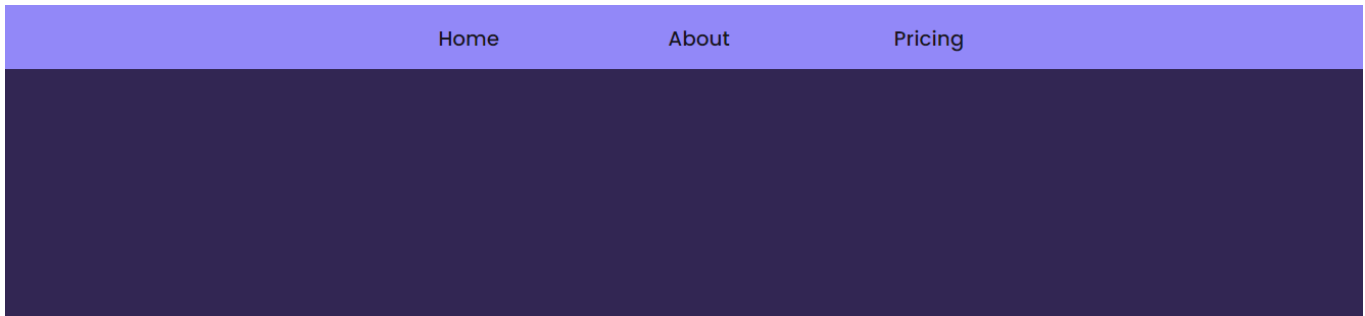


breakdown:





example 2:-



Breakdown :-



note : the red box is the anchor tags and blue one is div container.

To start this first, choose a dark background for body, and a lighter color for the navbar

- step1: now create a `div` (the `class="navbar-outermost-div"` )
- step2 : inside this create another `div` (the `class="navbar-container-div"` ) (this will contain smaller divs and will help to center the links)
- step3 : create 3-4 `divs` (as per the number of links) inside the container-div give these same ( `class="icon-divs"` )
- inside these divs create anchor tags `<a>` and give them same `class="icon-anchor-links"` .

html :-

```
<div class="navbar-main-div">
  <div class="navbar-icon-container">

    <div class="navbar-icon-div">
      <a href="#" class="icon-anchor-link">Home</a>
    </div>

    <div class="navbar-icon-div">
      <a href="#" class="icon-anchor-link">About</a>
    </div>

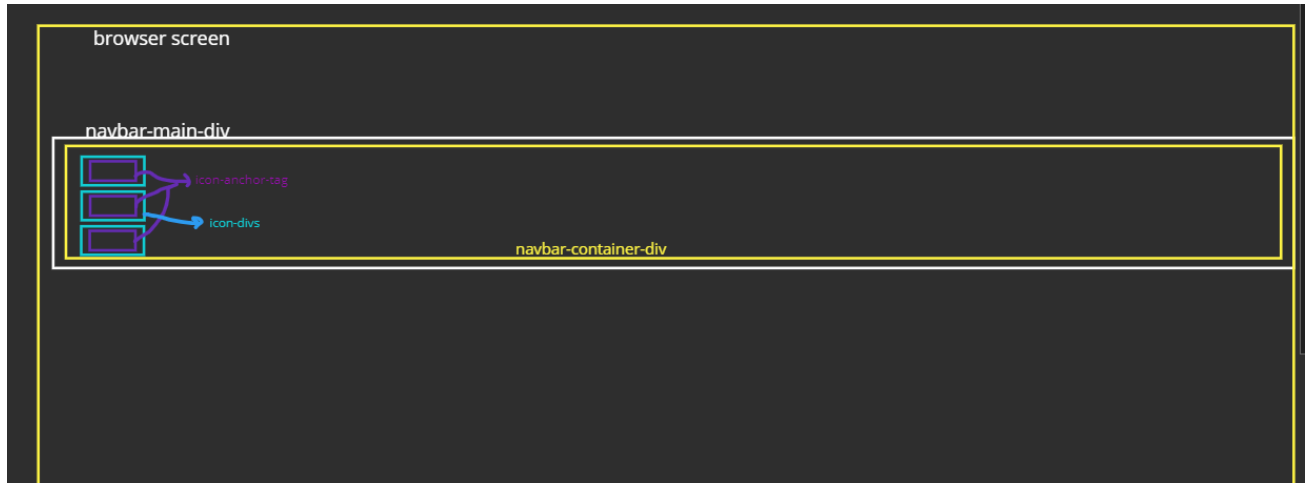
    <div class="navbar-icon-div">
      <a href="#" class="icon-anchor-link">Pricing</a>
    </div>

  </div>
</div>
```

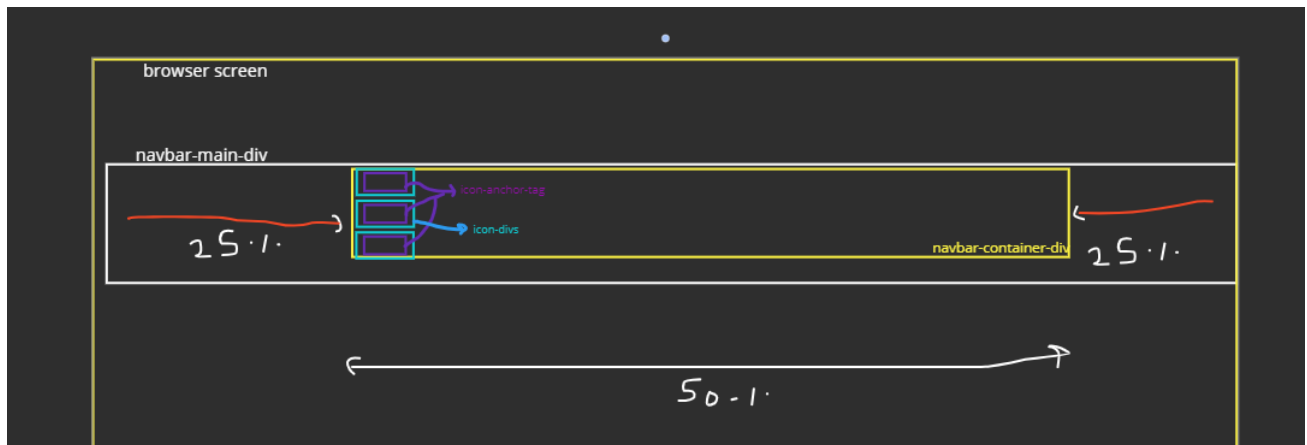
```
</div>  
</div>
```

now lets set them at their right positions (using css):-

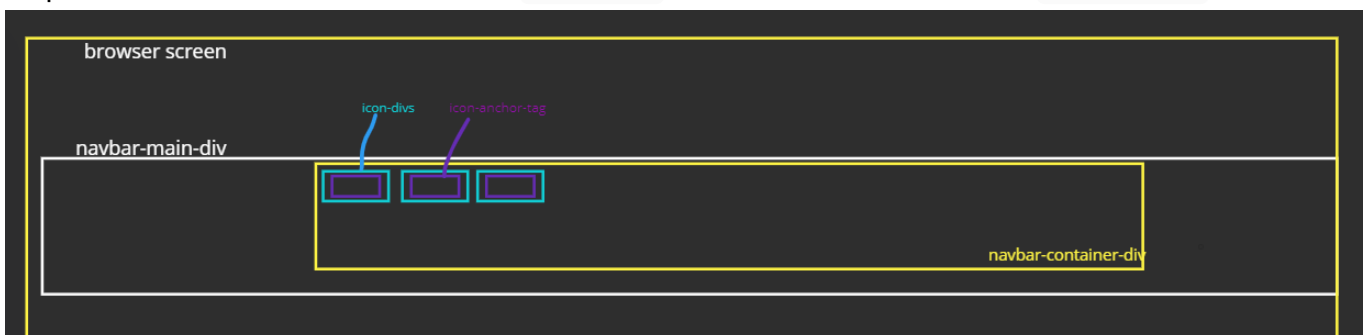
- step1 : 1st set the width of the div (navbar-main-div) to 100%.



- step2 : now center the inner div (navbar-icon-container) inside the outer main div, by setting its width to lets say any number eg. 50%. now set its `position: absolute` and set outer main div's `position: relative`, now we know the remaining space is 50%, so now for the `navbar-icon-container` div push `left: 25%` and `right: 25%`. and here out `navbar-icon-container` div is in center.

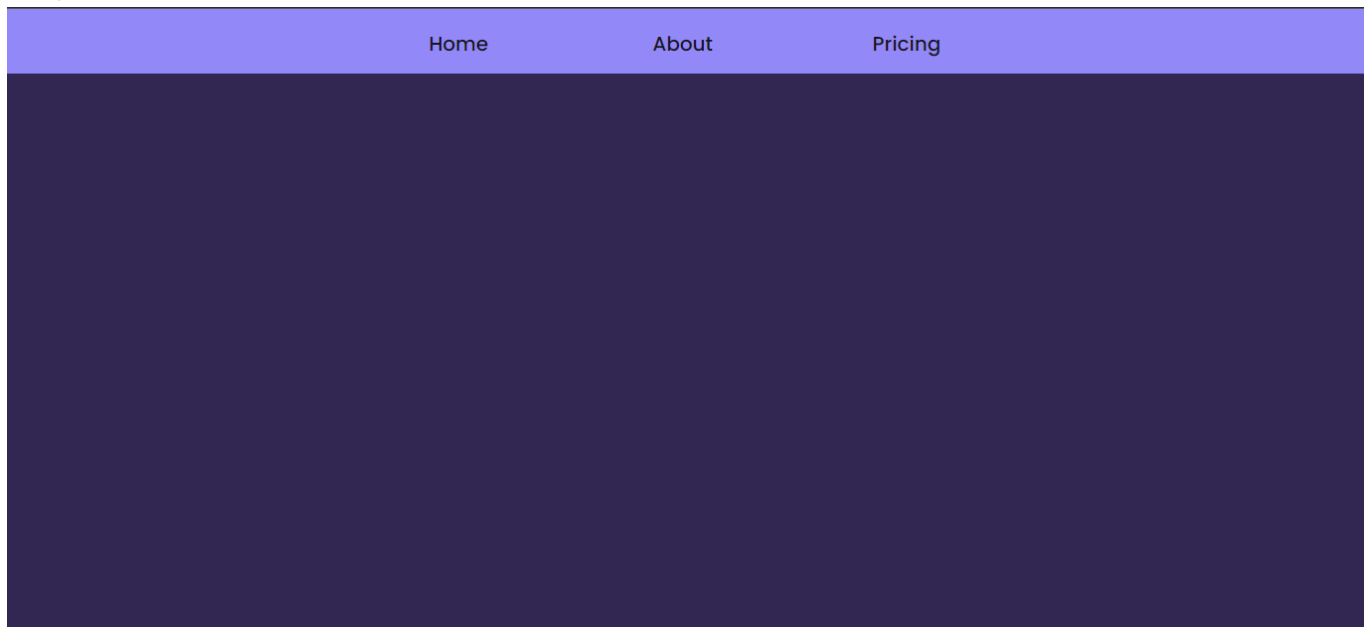


step3: now to set these smaller 3 div's `icon-divs` into horizontal line set their `float: left` .





Output :



```
body,html{
    margin:0;
}

body{
    background-color: #322653;
}

.navbar-main-div{
    background-color: #9288F8;
    height: 65px;
    position: relative;
}

.navbar-icon-container{
    position: absolute;
    /* border: 1px solid red; */
    height: 70%;
    width: 50%;
    left: 25%;
    right: 25%; /* i set the ht to 70% then pushed the container 30% so that to center
the icons*/
    top:30% ;
}

.navbar-icon-div{
    float: left;
    /* border: 1px solid black; */
    height: 100%;
    width: 33%; /*note: their is little space on right end of the navbar-container-div
because 33+33+33-> 99, remaining=1*/
```

```
    text-align: center;
}

.icon-anchor-link{
    font-size: 20px;
    color: #17151c;
    font-family: 'Poppins', sans-serif;
    text-decoration: none;
}

.icon-anchor-link:hover{
    font-size: 20px;
    color: #d21f1f;
    font-family: 'Poppins', sans-serif;
    text-decoration: none;
}
```

---