# to keep the scroll bar free from space it just be flying not take any place use:

text-align: center;

position: fixed;

bottom: 80px;

right: 15px;

Calculation for line height

So finally: if  
font size = 14px  
line height = (14\***1.618**) = 23px  
width @ 50cpl = 50\*(14/**1.618**) = 433px

CSS Website Layout

[❮ Previous](https://www.w3schools.com/css/css_counters.asp)[Next ❯](https://www.w3schools.com/css/css_units.asp)

Website Layout

A website is often divided into headers, menus, content and a footer:

Header

Navigation Menu

Content

Main Content

Content

Footer

There are tons of different layout designs to choose from. However, the structure above, is one of the most common, and we will take a closer look at it in this tutorial.

Header

A header is usually located at the top of the website (or right below a top navigation menu). It often contains a logo or the website name:

Example

.header {  
    background-color: #F1F1F1;  
    text-align: center;  
    padding: 20px;  
}

Result

Header

[Try it Yourself »](https://www.w3schools.com/css/tryit.asp?filename=trycss_website_layout_header)

Navigation Bar

A navigation bar contains a list of links to help visitors navigating through your website:

Example

/\* The navbar container \*/  
.topnav {  
    overflow: hidden;  
    background-color: #333;  
}  
  
/\* Navbar links \*/  
.topnav a {  
    float: left;  
    display: block;  
    color: #f2f2f2;  
    text-align: center;  
    padding: 14px 16px;  
    text-decoration: none;  
}  
  
/\* Links - change color on hover \*/  
.topnav a:hover {  
    background-color: #ddd;  
    color: black;  
}

Result

[LinkLinkLink](javascript:void(0))

[Try it Yourself »](https://www.w3schools.com/css/tryit.asp?filename=trycss_website_layout_navbar)

Content

The layout in this section, often depends on the target users. The most common layout is one (or combining them) of the following:

* **1-column** (often used for mobile browsers)
* **2-column** (often used for tablets and laptops)
* **3-column layout** (only used for desktops)

1-column: 2-column:3-column:

We will create a 3-column layout, and change it to a 1-column layout on smaller screens:

Example

/\* Create three equal columns that floats next to each other \*/  
.column {  
    float: left;  
    width: 33.33%;  
}  
  
/\* Clear floats after the columns \*/  
.row:after {  
    content: "";  
    display: table;  
    clear: both;  
}  
  
/\* Responsive layout - makes the three columns stack on top of each other instead of next to each other on smaller screens (600px wide or less) \*/  
@media screen and (max-width: 600px) {  
    .column {  
        width: 100%;  
    }  
}

Result

Column

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas sit amet pretium urna. Vivamus venenatis velit nec neque ultricies, eget elementum magna tristique.

Column

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas sit amet pretium urna. Vivamus venenatis velit nec neque ultricies, eget elementum magna tristique.

Column

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas sit amet pretium urna. Vivamus venenatis velit nec neque ultricies, eget elementum magna tristique.

[Try it Yourself »](https://www.w3schools.com/css/tryit.asp?filename=trycss_website_layout_grid)

**Tip:** To create a 2-column layout, change the width to 50%. To create a 4-column layout, use 25%, etc.

**Tip:** Do you wonder how the @media rule works? [Read more about it in our CSS Media Queries chapter](https://www.w3schools.com/css/css3_mediaqueries.asp).

**Tip:** A more modern way of creating column layouts, is to use CSS Flexbox. However, it is not supported in Internet Explorer 10 and earlier versions. If you require IE6-10 support, use floats (as shown above).   
  
To learn more about the Flexible Box Layout Module, [read our CSS Flexbox chapter](https://www.w3schools.com/css/css3_flexbox.asp).

Unequal Columns

The main content is the biggest and the most important part of your site.

It is common with **unequal** column widths, so that most of the space is reserved for the main content. The side content (if any) is often used as an alternative navigation or to specify information relevant to the main content. Change the widths as you like, only remember that it should add up to 100% in total:

Example

.column {  
    float: left;  
}  
  
/\* Left and right column \*/  
.column.side {  
    width: 25%;  
}  
  
/\* Middle column \*/  
.column.middle {  
    width: 50%;  
}  
  
/\* Responsive layout - makes the three columns stack on top of each other instead of next to each other \*/  
@media screen and (max-width: 600px) {  
  .column.side, .column.middle {  
    width: 100%;  
  }  
}

Result

Side

Lorem ipsum dolor sit amet, consectetur adipiscing elit...

Main Content

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas sit amet pretium urna. Vivamus venenatis velit nec neque ultricies, eget elementum magna tristique. Quisque vehicula, risus eget aliquam placerat, purus leo tincidunt eros, eget luctus quam orci in velit. Praesent scelerisque tortor sed accumsan convallis.

Side

Lorem ipsum dolor sit amet, consectetur adipiscing elit...

[Try it Yourself »](https://www.w3schools.com/css/tryit.asp?filename=trycss_website_layout_grid2)

Footer

The footer is placed at the bottom of your page. It often contains information like copyright and contact info:

Example

.footer {  
    background-color: #F1F1F1;  
    text-align: center;  
    padding: 10px;  
}

Result

Footer

[Try it Yourself »](https://www.w3schools.com/css/tryit.asp?filename=trycss_website_layout_footer)

Responsive Website Layout

By using some of the CSS code above, we have created a responsive website layout, which varies between two columns and full-width columns depending on screen width:

[Try it Yourself »](https://www.w3schools.com/css/tryit.asp?filename=trycss_website_layout_blog)

1 To keep icon center

justify-content-between = to keep icon in between left & right

align-items-center = to keep icon in between top & bottom

Ex: <div class="d-flex flex-row justify-content-between align-items-center ">

<h1 class="display-4" >Alamgir Chowdhury</h1>

<div><i class="fa fa-facebook"></i> </div>

2 keep the word below the icon use= **d-block**

<div class="d-flex flex-row justify-content-between align-items-center ">

<h1 class="display-4" >Alamgir Chowdhury</h1>

<div><i class="fa fa-facebook d-block "></i> </div>

3. to make the Icon size double use **= fa-2x**

<div class="d-flex flex-row justify-content-between align-items-center ">

<h1 class="display-4" >Alamgir Chowdhury</h1>

<div><i class="fa fa-facebook fa-2x d-block"></i> </div>

4. the short form of a complete icon is

.p-4.bg-danger>i.fa.fa-graduation-cap.fa-2x.d-block

Means=

<div class="p-4 bg-danger"><i class="fa fa-graduation-cap fa-2x d-block"></i>Resume</div>

**CSS function for a website from Start:**

**Css markup for PSD to Html:**

1. Open the grid system & take the width of header
2. Menu format
3. .menu{
4. background: #F2F2F2;
5. border-radius: 5px;
6. box-shadow: 1px 1px 2px #AFAFAF;
7. margin-top: 20px;
8. padding: 2px 0;
9. }
10. .menu ul{
11. list-style: none;
12. }
13. .menu ul li{
14. float: left;
15. }
16. .menu ul li a{
17. display: block;
18. font-family: qlassik;
19. font-size: 19px;
20. font-weight: bold;
21. color: #525252;
22. }

**Part 1**

To connect with HTML just type : link at HTML

**Default css starter of any website:**

<link rel="stylesheet" href="css/style.css">

\*{margin: 0;padding: 0;outline: 0;}

body{

font-family: Arial, Helvetica, sans-serif;

font-size: 14px;

line-height: 18px;

background: #ccc; or

background:url(../img/background.jpg )repeat scroll 0 0 #fff;

}

template{

width: 960px;

margin: 0 auto;

}

.clear{

overflow: hidden;

}

\*{margin: 0;padding: 0;outline: 0;}

body{font-family: arial;font-size: 14px;line-height: 18px;color: #000;}

.template{width: 960px;background: #ddd;margin-bottom: 10px; margin: 0 auto;}

*(this class is must add at all divs at HTML like: <div class="headersection"> to <div class="headersection template">)*

.clear{overflow: hidden;}

(this class is must at all divs at html to keep the div sequentially otherwise one thing will mix with another:

*<div class="headersectin template"> to <div class="headersection template clear">)*

.headersection{}

.headersection h2{}

.navsection{}

. contentsection{}

.maincontent{}

.maincontent p{}

.sidebar{}

.sidebar p{}

.footersection{}

.footersection p{}

margin: 0 auto; (0 means top & Button 0, auto means left & right equal)

letter-spacing:0px; helps to reduce the space between letters

**part 2:**

**Header**

1.background & header

.headersection{height: 85px;background: #000;color: #ddd;}

Or

It will help to cancel the gap with other section

.header{

background: #333333;

height: 80px;}

Here , background will black but the color white means the text color white.

2.headersection h2{}

.headersection h2{

font-size: 40px;

text-align: center;

padding-top: 30px;

line-height:60px;

}

this is an ideal condition for a header. Here line height ensure the font height into the header.

Common properties needed to keep perfect of any thing

padding-left: 50px;

padding-top: 5px;

font-size: 20px;

font-weight: bold;

color: white;

For search option:

<div class="headright clear">

<input type="text"placeholder="Search Our Website" >

<input type="submit" value="Go">

</div>

.headright input[type="text"] {

width: 245px;

background: #D5E5E8;

border: 1pxrgb(85, 195, 214);

padding: 5px;

margin-right: 3px;

}

.headright input[type="submit"] {

width: 30px;

background: #d5e5e8 none repeat scroll 0 0;

border: 1px solid #a1bcc2;

cursor: pointer;

padding: 5px;

}

**Footer**

.footersection{

height: 90px;

background: #000;

color: #ddd;

}

.footersection p{

font-size: 40px;

text-align: center;

padding-top: 30px;

}

.description{

font-size: 20px;

margin: 28px

margin: 28px 50px 40px;

line-height: 28px;

border-bottom: 2px dotted #ddd;

padding-bottom: 14px;

padding: 9px 10px 36px}

here padding 9px means the content will give space 9px from top 10px means content will give space form left & 36 px means give space at bottom but it ensure space inside the paragarph

but margin 20 px means it ensure 20px margin on left, right & top and for bottom we need to give padding bottom

and margin 28px means it ensure 20px margin on top, 50px on left and 49 px on bottom but it ensure the space form the hole para out side

basically margin ensure space for the whole content outside the box but padding ensure space inside the box.

**Content:**

.contentsection{

background: #FFFFFF;

}

.maincontent{

width: 650px;

background: beige;

float: left;

margin: 15px 15px 15px 0px;

padding: 10px

}

.maincontent p{

font-size: 16px;

line-height: 22px;

margin-bottom: 15px;

line-height: to ensure gape between two line

margin-bottom: to ensure gap between two paragaraph

Float: ensure where the paragraph will locate left or right

**Sidebar**

.sidebar{

width: 239px;

background: burlywood;

float: right;

padding: 10px;

margin-top: 15px

margin-bottom: 15px;

**navsection/Menubar:**

**min-height is used to show the menus into the menu bar** & also to ensure the cancellation of overflow hidden.

.navsection{background:#E6AF4B;min-height: 40px;}

to cancel the list or vertical menu style:

.navsection ul{

margin: 0;padding: 0;list-style: none;

z-index: 999; (it is used to show the submenu on the sider)

}

to, show the menu Horigentally:use float, position relative is used if the menu have submenu & its use to relate with the submenu

.navsection ul li{

border-right: 1px solid#B7801C;

border-left: 1px solid #B78012;

display: block;

float: left;

position: relative;

In this submenu position absolute is use to connect with the main menu it but it will disapper.

z-index: 999; will ensure the hover effect on the slider that means it will show the submenu onto the slider when we on the home page, otherwise not

.navsection ul li ul{

position: absolute;left:-99999px;

z-index: 999;

}

To ensure the hover effect in submenu we need to use ul after hover

.navsection ul li:hover ul{left:0px;}

To style the submenu

Here both right & left border 0px ensure that the submenu will not contain any border which exist in the main menu

.navsection ul li ul li {

background: #E6AF4B;

width: 200px;

border-bottom: 1px solid yellow;

float: none;

border-right: 0px solid#B7801C;

border-left: 0px solid #B78012;}

0px will ensure that the last submenu will not contain any border

.navsection ul li ul li:last-child{

border-bottom: 0px solid yellow;

}

Here,

font-weight: bold; means the alphabet will be bold

margin: 5px; means the font will get 5px extra space top & bottom.

padding: 10px 30px; 10px means top & bottom will get extra space inside the box,30px means left & right will get extra space inside the box.

**text-decoration: none; means the text underline will remove**

transition: it will ensure the timing effect of hover on different types of search engine

**display:block** = keep the 2nd line below the first line & keep the first line bit below the top.

.navsection ul li a{

display: block;

background: #f1bd79;

color: #050b0b;

font-size: 20px;

font-weight: bold;

margin: 5px;

padding: 10px 30px;

text-decoration: none;

-webkit-transition: .5s;

-moz-transition: .5s;

-o-transition: .5s;

}

**Hover is css property it is used to change the existing background & color when the mouse over an element/menu**.

.navsection ul li a:hover{background:#2F2D2E;color:#fff;

transition: .5s;}

Child property is used to cancel the border both left & right side with adding the 0px solid..

.navsection ul li:first-child{

border-left: 0px solid #B78012;}

.navsection ul li:last-child{

border-right: 0px solid#B7801C;}

**Image css property**

.samepost img{

width: 200px;

background: #fff none repeat scroll 0 0;

float: left;

margin-right: 10px;

padding: 5px;

border: 1px solid #9b999a;}

here, **text-align: justify means the text will be at equal line**, line-height always bigger than font-size.

.samepost p{

font-size: 16px;

line-height: 22px;

text-align: justify;}

.readmore{

background: #0076A4;

text-align: center;

text-transform: uppercase;

font-size: 16px;

font-weight: bold;

padding: 10px;}

When padding & margin is not working then use **display:block** and then use **padding & margin**

Border-radius is use to make the box corner smooth rather than sharp.

.readmore a{

text-decoration: none;

background: #ffff;

display: block;

padding: 3px 8px;

font-size: 18px;

border-radius: 3px;}

Border is used for keep the hole website into a square border

border: 1px solid;

For Logo (total space)

.logo {

float: left;

width: 430px;}

For Logo Image

.logo img {

float: left;

margin-right: 15px;

padding-bottom: 10px;

padding-left: 20px;

padding-top: 10px;

width: 70px;

For Logo headline

.logo h2 {

color: #fff;

font-size: 32px;

margin-bottom: 11px;

margin-top: 24px;

text-shadow: 2px 2px 0 #666;}

Social Media;

.social {

float: right;

padding-top: 16px;

width: 255px;}

Social media Img size

.social img {

width: 60px;}

SliderSection: to give shadow out side the slider

.slidersection{

margin-bottom: 8px;

box-shadow:0px 7px 7px #B7801C;}

for top & bottom same margin & others

.mainbottom{

margin-left: 50px;

margin-right: 50px;

margin-top: 30px;

border-top: 2px dotted #cccc;

padding-top: 38px;

border-bottom: 2px dotted #cccc;

margin-bottom: 30px;

padding-bottom: 30px;

}

Footer:

It will keep space top & bottom

.footermenu{

margin-top: 15px;

margin-bottom: 10px;

it need to done to keep the thing clear and swquentially

.footermenu ul{padding: 0;margin: 0;list-style: none;text-align: center;}

It will ensure the menus in proper line

.footermenu ul li{

display: inline-block;}

Margin-right ensure the gap between the menus

.footermenu ul li a{

color: honeydew;

font-size: 18px;

margin-right: 5px;}

Contact Section for Address bar:

input[type="text"],input[type="email"]{

width: 300px;

padding: 5px;

margin-bottom: 5px;

border: 1px solid #ddd;

border-radius: 3px;}

input[type="submit" ]{

background: #B7801C;

border: 1px solid#cfc5b6;

padding: 5px 10px;

font-size: 18px;

color: wheat;

cursor: pointer}

input[type="submit" ]:hover {

background: #FFA500;

color: #fff;

textarea{

width: 289px;

height: 110px;

margin-bottom: 10px;

padding: 10px;

border: 1px solid#ffff;

Related articles

.relatedpost h2{

background: #B7801C;

color: black !important;

!important is use when color is not work normally

.relatedpost img{

width: 200px;

height: 100px;

margin-bottom: 10px;

border-bottom: 2px solid #000 !important;

border-bottom: 2px dotted #DDC;

}

.relatedpost img:hover {opacity: 0.4;}

Border-bottom ensure the border like dotted,dashed, solid,double

Here, opacity ensure that when the cursor on the image it will give a hover effect onto the image.

To keep a image center

.items{

width: 400px;

height: 250px;

margin: 200px auto;

}

.items img{

width: 400px;

height: 250px;

}

.maincontent {

background: #FFFFFF;

width: 920px;

height: 350px;

margin: -40px auto 0 40px;

margin: 0 auto; it is used for give any h1 tag in middle from left to middle.

position: absolute;

**Position: absolute** = ensure that the main content will appear on top of other background or other sections .and **position;relative** should done with the content

**About Margin**:

CSS Margins

The CSS margin properties are used to create space around elements, outside of any defined borders.

With CSS, you have full control over the margins. There are properties for setting the margin for each side of an element (top, right, bottom, and left).

Margin - Individual Sides

CSS has properties for specifying the margin for each side of an element:

* margin-top
* margin-right
* margin-bottom
* margin-left

All the margin properties can have the following values:

* auto - the browser calculates the margin
* *length* - specifies a margin in px, pt, cm, etc.
* *%* - specifies a margin in % of the width of the containing element
* inherit - specifies that the margin should be inherited from the parent element

**Tip:** Negative values are allowed.

The following example sets different margins for all four sides of a <p> element:

Example

p {  
    margin-top: 100px;  
    margin-bottom: 100px;  
    margin-right: 150px;  
    margin-left: 80px;  
}

## Margin - Shorthand Property

To shorten the code, it is possible to specify all the margin properties in one property.

The margin property is a shorthand property for the following individual margin properties:

* margin-top
* margin-right
* margin-bottom
* margin-left

So, here is how it works:

If the margin property has four values:

* **margin: 25px 50px 75px 100px;**
  + top margin is 25px
  + right margin is 50px
  + bottom margin is 75px
  + left margin is 100px

### Example

p {  
    margin: 25px 50px 75px 100px;  
}

CSS Padding

The CSS padding properties are used to generate space around an element's content, inside of any defined borders.

With CSS, you have full control over the padding. There are properties for setting the padding for each side of an element (top, right, bottom, and left).

Padding - Individual Sides

CSS has properties for specifying the padding for each side of an element:

* padding-top
* padding-right
* padding-bottom
* padding-left

All the padding properties can have the following values:

* *length* - specifies a padding in px, pt, cm, etc.
* *%* - specifies a padding in % of the width of the containing element
* inherit - specifies that the padding should be inherited from the parent element

**Note:** Negative values are not allowed.

The following example sets different padding for all four sides of a <div> element:

Example

div {  
    padding-top: 50px;  
    padding-right: 30px;  
    padding-bottom: 50px;  
    padding-left: 80px;  
}

## Padding - Shorthand Property

To shorten the code, it is possible to specify all the padding properties in one property.

The padding property is a shorthand property for the following individual padding properties:

* padding-top
* padding-right
* padding-bottom
* padding-left

So, here is how it works:

If the padding property has four values:

* **padding: 25px 50px 75px 100px;**
  + top padding is 25px
  + right padding is 50px
  + bottom padding is 75px
  + left padding is 100px

### Example

div {  
    padding: 25px 50px 75px 100px;  
}

If the padding property has three values:

* **padding: 25px 50px 75px;**
  + top padding is 25px
  + right and left paddings are 50px
  + bottom padding is 75px

### Example

div {  
    padding: 25px 50px 75px;  
}

If the padding property has two values:

* **padding: 25px 50px;**
  + top and bottom paddings are 25px
  + right and left paddings are 50px

### Example

div {  
    padding: 25px 50px;  
}

If the padding property has one value:

* **padding: 25px;**
  + all four paddings are 25px

### Example

div {  
    padding: 25px;  
}

## Padding and Element Width

The CSS width property specifies the width of the element's content area. The content area is the portion inside the padding, border, and margin of an element ([the box model](https://www.w3schools.com/css/css_boxmodel.asp)).

So, if an element has a specified width, the padding added to that element will be added to the total width of the element. This is often an undesirable result.

In the following example, the <div> element is given a width of 300px. However, the actual rendered width of the <div> element will be 350px (300px + 25px of left padding + 25px of right padding):

### Example

div {  
    width: 300px;  
    padding: 25px;  
}

To keep the width at 300px, no matter the amount of padding, you can use the box-sizing property. This causes the element to maintain its width; if you increase the padding, the available content space will decrease. Here is an example:

### Example

div {  
    width: 300px;  
    padding: 25px;  
    box-sizing: border-box;  
}

CSS Borders

The CSS border properties allow you to specify the style, width, and color of an element's border.

## Border Style

The border-style property specifies what kind of border to display.

The following values are allowed:

* dotted - Defines a dotted border
* dashed - Defines a dashed border
* solid - Defines a solid border
* double - Defines a double border
* groove - Defines a 3D grooved border. The effect depends on the border-color value
* ridge - Defines a 3D ridged border. The effect depends on the border-color value
* inset - Defines a 3D inset border. The effect depends on the border-color value
* outset - Defines a 3D outset border. The effect depends on the border-color value
* none - Defines no border
* hidden - Defines a hidden border

The border-style property can have from one to four values (for the top border, right border, bottom border, and the left border).

### Example

p.dotted {border-style: dotted;}  
p.dashed {border-style: dashed;}  
p.solid {border-style: solid;}  
p.double {border-style: double;}  
p.groove {border-style: groove;}  
p.ridge {border-style: ridge;}  
p.inset {border-style: inset;}  
p.outset {border-style: outset;}  
p.none {border-style: none;}  
p.hidden {border-style: hidden;}  
p.mix {border-style: dotted dashed solid double;}

Result:

A dotted border.

A dashed border.

A solid border.

A double border.

A groove border. The effect depends on the border-color value.

A ridge border. The effect depends on the border-color value.

An inset border. The effect depends on the border-color value.

An outset border. The effect depends on the border-color value.

No border.

A hidden border.

A mixed border.

## Border - Shorthand Property

As you can see from the examples above, there are many properties to consider when dealing with borders.

To shorten the code, it is also possible to specify all the individual border properties in one property.

The border property is a shorthand property for the following individual border properties:

* border-width
* border-style (required)
* border-color

### Example

p {  
    border: 5px solid red;  
}

Result:

Some text

You can also specify all the individual border properties for just one side:

Left Border

p {  
    border-left: 6px solid red;  
    background-color: lightgrey;  
}

Result:

Some text

Bottom Border

p {  
    border-bottom: 6px solid red;  
    background-color: lightgrey;  
}

Result:

Some text

Rounded Borders

The border-radius property is used to add rounded borders to an element:

Normal border

Round border

Rounder border

Roundest border

Example

p {  
    border: 2px solid red;  
    border-radius: 5px;  
}

The display

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

**inline** Displays an element as an inline element (like <span>). Any height and width properties will have no effect

**block** Displays an element as a block element (like <p>). It starts on a new line, and takes up the whole width

**contents** Makes the container disappear, making the child elements children of the element the next level up in the DOM

**flex** Displays an element as a block-level flex container

**none** The element is completely removed

Example

A demonstration of how to use the contents property value. In the following example the .a container will disappear, and making the child elements (.b) children of the element the next level up in the DOM:

.a {  
    display: contents;  
    border: 2px solid red;  
    background-color: #ccc;  
    padding: 10px;  
    width: 200px;  
}  
  
.b {  
    border: 2px solid blue;  
    background-color: lightblue;  
    padding: 10px;  
}

* 1. **margin: 0 auto;** (means the website will be into the middle with the mentioned width)
  2. **list-style: none;** to cancel the dot dot from the menu
  3. **display:block**; = **it will block the full width of the line**. b 4 using padding it needs/keep the 2nd line below the first line & keep the first line bit below the top
  4. **display: inline-block**; = it will only block the word which need to block like into a box
  5. **Padding is preferable if margin is create problem. Margin is preferable to create gap between different section.**
  6. **Float:left & right =means exactly it will be on the wall.**
  7. **Overflow:hidden= its always necessary for all type of problem specially side bar. And its necessary to reduce the width of a box of any thing when u use overflow:hidden.**
  8. In many cases we can use **height 100% width 100%** when **Cover** is not work properly

  background-position: center center;

   background-size: 100% 100%; or Cover;

* 1. In many cases we found padding is not working because the serial of the padding the below serial ensure the proper padding but if the padding-left position is b4 padding then it will not work

padding: 10px 0px;

padding-left: 30px;

* .single\_sidebar p{
* font-family: lato;
* font-size: 16px;
* color: #737373;
* margin-left: 15px;
* width: 85.3%;

for side bar it is always create problem % always help to keep space right side or left side perfectly.