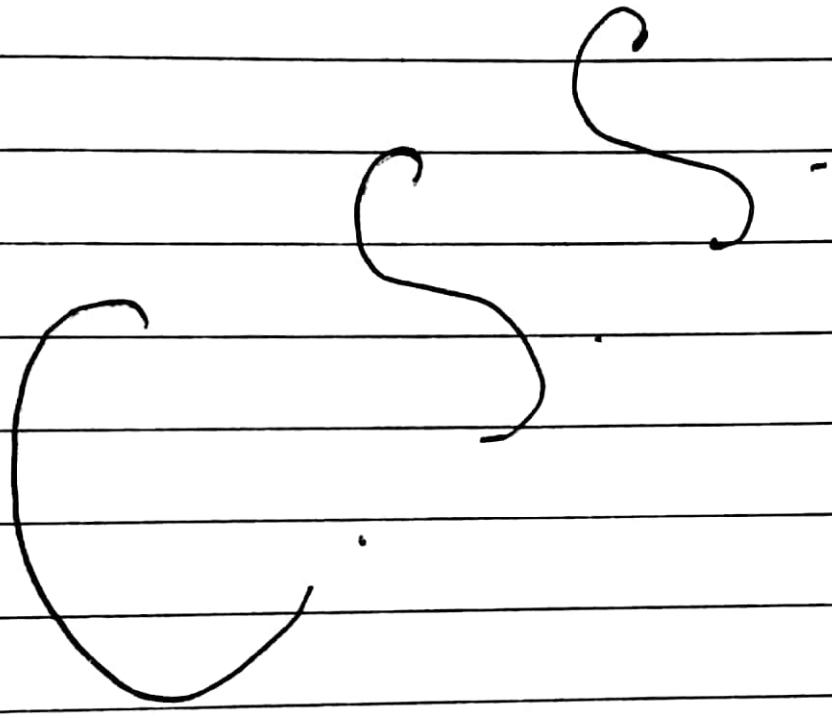


C.S.S

⇒ CSS stands for Cascading Style Sheets. CSS is used to control the style and layout of Web Pages.
for developing rich look web pages we are using CSS.

⇒ Three ways to Insert CSS.

1) Inline style sheet.

2) Internal style sheet.

3) External style sheet.

⇒ CSS Selectors.

1) class [.]

2) id [#]

3) tag [<>]

- Inline Style Sheet : For inline Style Sheet we are writing the CSS code directly inside the tag by using style attribute

Eg.

```
<HTML>
    <Head>
        <title> ———— </title>
    </Head>

<body bgcolor = "green" >

    <h1 Style = " color : white;
                  background-color : red;
                  text-align : center; >
        Good Morning India
    </h1>

    <P Style = " font-size : 26px;
                 color : white;
                 text-align : justify; >
        This is a Paragraph
    </P>

</body>
</HTML>
```

- Internal Style Sheet : For internal Style Sheet we are writing CSS code at the Head Part by using style tage.

Eg.

```
<HTML>
    <Head>
        <title> ———————— </title>
    <Style type = "text/css">
    p
    {
        color : white;
        font-size : 24px;
        background - color : purple;
        text - align : justify;
    }
    p.a
    {
        background - color : #000099;
    }
    p.b
    {
        background - color : #993300;
        Position : absolute;
        left : 350 px;
        top : 45 px;
    }
    </Style>  </Head>
```

<body>

</body>

</HTML>

- External Style Sheet

> For external style sheet we are writing CSS code in seprate Notepad.

> After that we are saving it with .CSS file format

> When calling that CSS file at the head part af af the HTML by using Link tag.

eg.

ex.HTML

```
<HTML>
<Head> <Title> ———— </Title>
<link href = " ex.css" rel = "stylesheet" >
</head>

<body>
 <p>
      ————
    ——— <span> Notepad </span>
 </p>

</body>
</HTML>
```

ex. CSS

```
p {
    font-size : 26 px;
    color : white;
    border-color : white;
    border-width : 3 px;
    border-style : dotted;
}

span {
    color : yellow;
    border-bottom : 2 px solid white;
}
```

- **CSS Syntax**: A CSS rule has two main part : a selector, and one or more declarations.

The selector is normally the HTML element you want to style. Each declaration consists of a property and a value.

The property is the style attribute you want to change. Each property has a value.

- CSS declaration always ends with a semicolon, and declaration groups are surrounded by curly brackets.

> p { color: red; text-align: center; }

To make the CSS more readable, you can put one declaration on each line, like this.

eg:

```
p
{
    color: red;
    text-align: center;
}
```

Date
Page

- **CSS Comments:** Comments are used to explain your code, and may help you when you edit the source code at a last date.

  Comments are ignored by browsers.

- Single line comment.

```
// This is single line comment
```

- Multi line comment

```
/* This is a multi
   line comment */
```

Date
Page

- CSS Id and class Selectors

In addition to setting a style for a HTML element, CSS allow you to specify your own selectors called 'id' and 'class'.

> The id Selector

The id Selector is used to specify a style for a single, unide element. The id Selector uses the id attribute of the HTML element, and is defined with a '#' The style rule below will be applied to the element with id = "ABC".

eg.

```
#ABC
{
    text-align : Center;
    color : red;
}
```

Note: Do Not Start and ID Name with a number it will not work in Mozilla / Firefox.

> The class Selector

- The class selector is used to specify a style for a group of elements. Unlike the id selector, the class selector is most often used on several elements.

- This allow you to set a particular style for any HTML elements with the same class.

- The class selector uses the HTML class attribute, and is defined with a '.'.

- In the example below, all HTML elements with class = 'center' will be center-allined.

Eg.

```
.center { text-align : center; }
```

You can Specify that only Specific HTML elements should be affected by a class In the example below, all p elements with class ='center' will be center-aligned.

eg.

```
p.center { text-align : center; }
```

Note: Do Not Start a class name with a number! This is only supported in IE.

Text related CSS

1) font-family : It deals with font face.

2) font-size : It is used to define font size.

3) font-weight : normal, bold
it is used to define thikness of the font.

4) font-style : It is used to choose italic font size.

5) font-Variant : normal, small, caps.

6) line-height : gap between two lines.

7) Text-transform : UPPERCASE, lowercase
Title case, Sentence case.

8) Color : It is used to choose color.

9) Text-decoration : underline, Overline
linethrough, blink,
none.

Eg.

```
.A {
    font-family : arial;
    font-size : 14px;
    color : gray;
    font-weight : bold;
    font-style : italic;
    text-transform : uppercase;
    line-height : 14px;
    word-spacing : 2px;
    vertical-align : center;
    text-align : justify;
    text-indent : 20 px;
}
```

3

Styling Backgrounds

1) background-color : it is used to control the color of the background.

2) background-image : it is used Place the background image.

3) background-repeat : repeat, repeat-x, repeat-y, no-repeat.

4) background-position : Xposition.

5) background-attachment : fixed, scroll

Eg.

```
• A {
    background-colour: red;
    background-image: url("a.jpg");

    background-repeat: no-repeat;

    background-position: 250px, 200px;

    background-attachment: fixed;
```

3

Block

These are used to control the Paragraph block of Text.

1) Word-spacing : gap between words.

2) letter-spacing : gap between letter.

3) Vertical-align : It controls how the text is Vertically aligned in the block.

4) Text-align : it controls how The Text is aligned in the block.

5) Text-indent : first line indent of Texe.

eg.

```
Word-spacing : 5px;

letter-spacing : 5px;

Vertical-align : 5px;

Text-align : justify;

Text-indent : 125px;
```

Padding

- The CSS Padding Properties define the space between the element border and the element content.

Eg.

```
padding-top : 10 px;

padding-left : 10 px;

padding-bottom : 10 px;

padding-right : 10 px;
```

```
padding : 10 px  8 px  0 px  30 px;
          ↓      ↓     ↓     ↓
          /* top  Right bottom left */
```

```
padding : 10 px  40 px;
          ↓      ↳ left & right
          ↳ Top & bottom
```

```
padding : 25 px  50 px  75 px;
          ↓      ↓     ↓
          Top    right & left  bottom
```

```
padding : 25 px;

/* all four Paddings are 25 px */
```

Margin

- The CSS margin properties define the space between the one element boarder and the another element boarder.

Eg.

```
margin-top : 10px;

margin-right : 10px;

margin-bottom : 10px;

margin-left : 10px;
```

```
margin : 10px  5px  7px  8px;
          ↓     ↓     ↓     ↓
        /* Top  right  bottom  left */
```

```
margin : 10px  40 px;
          ↳ left & right
   ↳ Top & bottom
```

```
margin :  25 px   50px   75 px ;
            ↓       ↓       ↓
          Top    right & left   bottom
```

```
margin : 25 px;

/* all four margin are 25 px */
```

Positioning

- The CSS Positioning Properties allow you to position an element it can also place an element behind another, and specify what should happen when an element's content is too big.

- Different types of Positioning

> Position : static;

> Position : fixed;

> Position : relative;

> Position : Absolute;

> Position : initial;

> Position : inherit;

- Static Positioning : HTML elements are Positioned Static by default. A Static Positioned element is always Positioned according to the normal flow of the Page.

Static Positioned elements are not affected by the top, bottom, left, and right properties.

- Fixed Positioning : An element with fixed Position is Positioned relative to the browser window.
It will not move even if the window is scrolled.

eg.

```
P. Pos_fixed
{
    Position : fixed;
    top : 30px;
    right : 5px;
}
```

- Relative Positioning : A relative Positioned element is positioned relative to its normal position.

eg:

```
h2. pos_left
{
    position : relative;
    left : -20px;
}
h2. pos_right
{
    position : relative;
    left : 20px;
}
```

The content of relatively elements can be moved and overlap other elements, but the reserved space for the element is still preserved in the normal flow.

eg. h2. pos_top { position: relative; top: -50px; }

Relatively positioned elements are often used as container block for absolutely positioned element.

- Absolute Positioning : An absolute Position element is Positioned relative to the first Parent element that has a Position other than Static. if no such element is found, the containing block is <html>.

eg. h2 {
Position : absolute;
left : 100Px;
top : 150 Px;
}

→ Absolutely Positioned elements are removed from the normal flow. The Document and other elements behave like the absolutely Positioned element does not exist.

→ Absolutely Positioned elements Can overlap other elements.

- Overlapping elements

When elements are positioned outside the normal flow, they can overlap other elements.

The z-index property specifies the stack order of an element (which element should be placed in front of, or behind, the others).

An element can have a positive or negative stack order.

eg:

```
img
{   position: absolute;
    left: 0px;
    top: 0px;
    z-index: -1;
}
```

→ An element with greater stack order is always in front of an element with a lower stack order.

classmate
Date
Page

- Initial Positioning : Sets this Property to its default Value.

- Inherit Positioning : inherit this Property its Parent element.

Display Property

- The display Property specifies how an element is displayed.

eg.

```
display : none;

display : block;

display : inline;

display : inline-block;
```

- display : none;
  // its display nothing.
- display : block;
  // it's display element in block.
- display : inline;
  // it's display element in horizontal.
- display : inline-block;
  // its display element in horizontal block

eg.

```
<HTML>
 <Head>
  <Style>
    #a ul li
    {
        // display : none;
        // display : block;
        // display : inline;
        // display : inline-block;
    }
  </Style>
 </Head>

 <body>
    <div id = "a">
      <UL>
        <li> ABC </li>
        <li> XYZ </li>
        <li> ASD </li>
      </UL>
    </div>
 </body>
</HTML>
```

CSS Float

- With CSS float, an element can be pushed to the left or right, allowing other elements to wrap around it. Float is very often used for images, but it is also useful when working with layouts.

⇒ How elements float

- Elements are floated horizontally, this means that an element can only be floated left or right, not up or down.

- A floated element will move as far to the left or right as it can. Usually this means all the way to the left or right of the containing element.

- The elements after the floating element will flow around it.

- The elements before the floating element will not be affected.

- if an image is floated to the right, a following text flows around it, to the left.

eg.

```
img
{
    float: right;
}
```

⇒ Floating Element next to each other.

- if you place several floating elements after each other, they will float next to each other if there is room.

Here we have made an image gallery using the float property.

```
.thumbnail
{
    float: left;
    width: 110px;
    height: 90px;
    margin: 5px;
}
```

- Turning off float - using clear.

→ Element after the floating element will flow around it. To avoid this, use the clear property.

→ The clear property specifies which sides of an element other floating elements are not allowed.

→ Add a text line into the image gallary, using the clear properties.

eg:

```
.text_line
{
    clear: both;
}
```

- Height & Width : it controls the Div part of Heish and width.

eg. #div1
{
Height : 100 px;
width : 400 px;
}

- Placement : Top, Right, Bottom, left.
- Visibility : visible, hidden, Inherit.
- Z-index : it is used to control the layering of object.
  → z-index : 55;
- Overflow : visible, hidden, Scroll, auto.

// it is used to control the extra amount of Text in division area.

- clip : To clip the extra amount of the text.

- **Styling Links** : Links can be styled in different ways.

The four links states are:

1) a:link - a normal, unvisited link.
2) a:visited - a link the user has visited.
3) a:hover - a link when the user mouses over it.
4) a:active - a link the moment it is clicked.

eg.
```
a:link {
    color: blue;
    text-decoration: none;
}
a:visited {
    color: red;
    text-decoration: none;
}
a:hover {
    color: Black;
    text-decoration: underline;
}
a:active {
    color: green;
    text-decoration: underline;
}
```

- Image opacity :

```
<HTML>
  <Head>
    <Style>
      img {
            opacity : 0.4;
            filter : alpha (opacity =40);
          }
      img : hover {
            opacity : 1;
            filter : alpha (opacity =100);
          }
    </Style>
  </Head>

  <body>
      <img src = "a.jpg">
  </body>

</HTML>
```

CSS 3 Text Effects

CSS 3 contains several new text features.

- text-shadow
- word-wrap

⇒ Text-shadow : The Text shadow Property applies shadow to text.
You specify the horizontal shadow, the Vertical shadow, the blur distance and the color of the shadow.

eg.

```
text-shadow : 5px 5px 5px gray;
              ↓   ↓   ↓   ↓
              x   y   offset  color
```

Date
Page

⇒ word-wrap : if a word is too long to sit within an area, it expands outside. The word-wrap property allows you to force the text wrap - even if it means splitting it in the middle of a word.

eg.

Allow long word to be able to break and wrap onto the next line.

p { word-wrap : break-word; }

[CSS Gradients]

1) linear gradient()

2) radial gradient()

3) repeating-linear-gradient()

4) repeating-radial-gradient()

eg.
```
<Head>
  <Style>
    #grad {
        Position: relative;
        width: 300px;
        height: 300px;
        left: 400px;
        top: 400px;
        background: linear-gradient
            (red, blue);
    }
  </Style>
</Head>

<body>
  <div id = "grad">
  </div>
</body>
```

- Web browser code can support new CSS elements.
  - -webkit- // safari & chrome
  - -ms- // I.E.
  - -O- // Opera
  - -MOZ- // mozila

- background : -moz- linear-gradient (, red, blue);
  - To top, To right, To Left, To Bottom.
  - /* we can choose any one parameter */

- background : linear-gradient (180deg, rgba (255,0,0,0), rgba (255,0,0,1));

- background : repeating-linear-gradient (red, yellow 10%, green 20%);

CSS Pseudo class

⇒ This is used for creating special effect.

eg.
```
<HTML>
  <Head>
    <link rel="stylesheet" href="main.css">
  </Head>
  <body>
    <p> You </p>
    <p> only </p>
    <p> ABC </p>
    <p> XYZ </p>
  </body>
</HTML>
```

main.css

```
p:first-letter {
    color: blue;
    font-size: 20px;
}

/* This code is used for changing
   first letter. */
```

```
<HTML>
    <Head>
        <link rel = "stylesheet"
              href = "main.css" >
    </Head>

    <body>
        <p>  ————————
             ————————
             ————————
        </p>
    </body>
</HTML>
```

main.css

```
p:first-line {
        color: blue;
        font-size: 20px;
    }

/* This code is used for changing
or creating effect for first
line. */
```

```
p:first-child {
    color: cyan;
    font-size: 20px;
}

p:before {
    content: url("a.png");
}

p:after {
    content: url("b.png");
}
```

2D effects

This is used for creating 2D effect in our document.

```
<HTML>
  <Head>
    <Style>
      div {
          Position: relative;
          width: 300px;
          height: 250px;
          background-color: green;
          left: 300px;
          top: 250px;
          transform: rotate (30deg);
          transform: scale (2,4);
          transform: translate (200px, 400px);
          transform: skew (30deg, 20deg);
          transform: matrix (0.866, -0.5,
                             -0.5, 0.866, 0, 0);

          transform: rotatx (120 deg);
          transform: rotaryC 120 deg);
```

3

Date
Page

eg.

```
div {
    Position : relative;
    width : 280px;
    height : 300 px;
    background-color : green;

    -webkit- transition : width 2s;

div : hover {
        width : 800px;
    }

div {
        width : 100px;
        height : 100px;
        background : red;
    -webkit- transition : width 2s,
                          height 2s;
    }
div : hover {
        width : 200 px;
        height : 200 px;

    -webkit- transform : rotate (180 deg);
```

- Some task related to HTML & CSS.

> Creating a HORIZENTAL MENU

HTML CODE

```
<HTML>
<Head>
  < link href = "menu.css" rel="stylesheet" >
</Head>
< body>
  < div class = 'menu' >
    <UL>
      <li> <a href = "#"> Home </a> </li>
      <li> <a href = "#"> Aboutus </a> </li>

      <li> <a href = "#"> Product </a> </li>

      <li> <a href = "#"> Contact </a> </li>

    </UL>

  </div>

</body>
</HTML>
```

CSS Code

menu.css

```
body {
    margin : 0px;
}
div.menu
{
    height : 50px;
    width : 1024 px;
    background-color : green;
    margin-left : auto;
    margin-right : auto;
}
div.menu ul
{
    font-size : 20px;
    color : white;
    margin : 0px;
    padding : 0px;
}
div.menu ul li
{
    list-style-type : none;
    float : left;
    padding-left : 30px;
    padding-right : 12px;
    height : 50 px;
    line-height : 50 px;
}
```

```
div.menu ul li a
{
    color : white;
    text-decoration : none;
}
div.menu ul li a : hover
{
    color : Pink;
    text-decoration : none;
}
div.menu ul li : hover
{
    background-color : blue;
}
```

[Output in Browser]

Home AboutUs Product Contact

• CREATING A POP-UP MENU

[HTML code]

```
<body>
  <div Id="menu">
  <ul>
    <li><a href="#">Home</a></li>
    <li><a href="#">About us</a></li>
    <li><a href="#">Product</a>
      <ul>
        <li><a href="#">Product1</a></li>
        <li><a href="#">Product2</a></li>
        <li><a href="#">Product3</a></li>
      </ul>
    </li>

    <li><a href="#">Contact</a></li>

  </ul>
  </div>
</body>
```

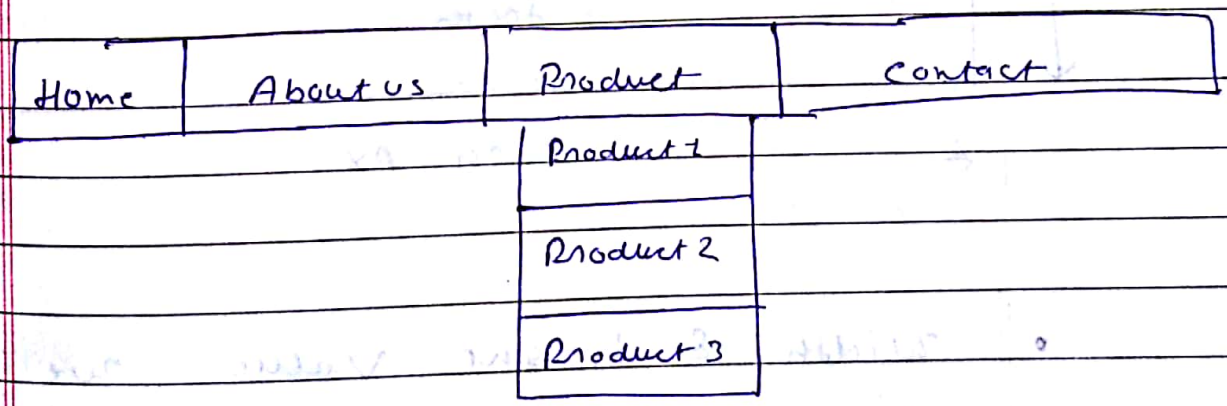
CSS code

```
body
{
    margin : 0 px;
}
#menu
{
    height : 50px;
    width : 1024 px;
    background-color : green;
    margin-left : auto;
    margin-right : auto;
}

#menu ul
{
    font-size : 20 px;
    color : white;
    margin : 0 px;
    Padding : 0 px;
}
#menu ul li
{
    list-style-type : none;
    float : left;
    Padding-left : 30 px;
    Padding-right : 12 px;
    height : 50 px;
}   line-height : 50 px;
```

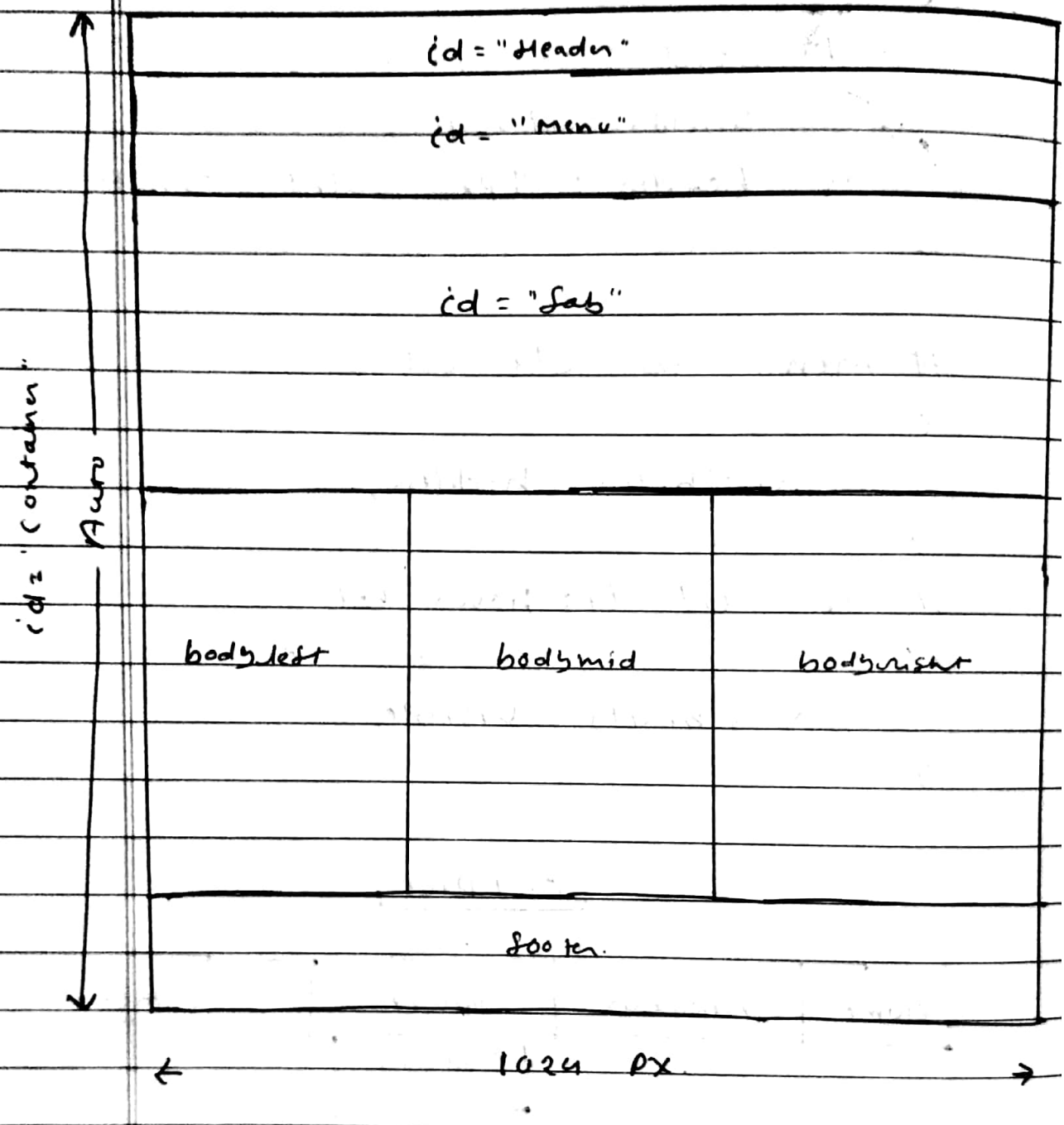
```
#menu ul li a
{
    color : white;
    text-decoration : none;
}
#menu ul li a:hover
{
    color : pink;
    text-decoration : none;
}
#menu ul li:hover
{
    background-color : blue;
}

#menu ul li
{
    position : relative;
}

#menu ul li ul
{
    position : absolute;
    background-color : green;
    left : 0px;

}
```

```
#menu ul li ul li
{
    font-size : 16px;
    width : 75 px;
    boarder : 1 px solid white;
}

#menu ul li ul
{
    visibility : hidden;
}
#menu ul li : hover ul
{
    visibility : visible;
}
```

Output



- Design a layout using Div tag.



- Width & height value we can give px value or % value.

HTML code

```
<body>
    <div id = "container">
        <div id = "header">
        </div>

        <div id = "menu">
        </div>

        <div id = "fab">
        </div>

        <div id = "body">
            <div id = "bodyleft">
            </div>

            <div id = "bodymid">
            </div>

            <div id = "bodyright">
            </div>
        </div>

        <div id = "footer">
        </div>
    </div>
</body>
```

CSS Code

```
body
{
    margin : 0px;
}
#Container
{
    width : 1024px;
    height : auto;
    margin-left : auto;
    margin-right : auto;
}
# header
{
    height : 110 px;
    width : 1024 px;
    background-color : green;
}

#menu
{
    height : 60 px;
    width : 1024 px;
    background-color : blue;
    float: left;

}
```

```
#Sab
{
    height : 300px;
    width : 1024 px;
    background-color : red;
    float : left;
}
#body
{
    height : 200px;
    width : 1024px;
    background-color : purple;
    float : left;
}

# body left
{
    height : 200px;
    width : 341 px;
    background-color : purple;
    float : left;
}
#bodymid
{
    height : 200px;
    width : 341 px;
    background-color : yellow;
    float : left;
}
```

#bodyright

{

height : 200px;

width : 347px;

background-color : red;

float : left;

}

#footer

{

height : 100px;

width : 1024 px;

float : left;

background-color : red;

}