CSS 3

O. A. Marcus

SIMPLIFIED

CSS 3

O. A. Marcus

SIMPLIFIED

ISBN: 9798701049565

Cover design by: Niflare

TABLE OF CONTENT

- Visited Class

Playground (Text Formatting)

- Font Family
- Font Weight
- Text Shadow
- Text Transform
- Text Colour of Different Elements.
- Align the Text
- Decorate the Text
- Control the Letters in a Text
- Specify the Space between Characters
- Specify the Space between Lines
- Increase White space between Words
- Vertical Alignment of an Image inside Text
- An element with a total width of 250px
- Specify a black border for table, th, and td elements

Playground (Background)

- Background Color of a Page
- Backgrounds
- Background Attribute
- Background Color of different Elements
- Background Image only Horizontally
- Background Image
- Position a Background Image

Playground (List)

- Set an image as the list-item marker
- List Style
- Link Style Position

Playground (Box Model)

- Box Shadow
- Outline
- Float
- Set different borders on each sides
- Set the width of the four borders
- Set the color of the four borders

Set the style of the four borders

INTRODUCTION TO CSS

Cascading Style Sheets is a style sheet that decides how your HTML will look like.
Usefulness of CSS are:

- It add colors to the webpage.
- Providing webpage a rich and attractive look.
- Changes shapes, give borders to HTML Elements.
- Position your elements as you want(change the layout)
- Animate your HTML elements.

Syntax of CSS

The CSS syntax consists of **Selectors** and **Declarations.**
Selectors: are simply used to select the HTML element to be styled.
Declaration: consists of a set of instructions which tell the browser how to style the selected HTML element.
Selectors tell "where to do" and declarations tell the browser "what to do".

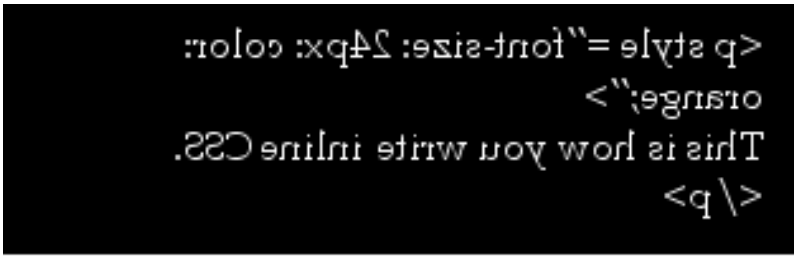
How to add CSS to your HTML file.

There are the three ways to add CSS to HTML:

- **Inline CSS** : write your CSS within the HTML tags.
- **Internal CSS** : write it inside your HTML file enclosed within the <style></style> tag.
- **External CSS** : write it in a separate file with a .css extension and include it in HTML using the <lnk> tag.

Inline CSS

Writing CSS inside the HTML tags using a style attribute.



It will change the color of text inside the paragraph to orange and font-size to 24px.

Internal CSS

When you write CSS enclosed in the <style> tag within the <head> tag in your HTML file, it is known as **internal CSS** .

```
<head>
<style>
p{
color: orange
font-size: 24px;
}
</style>
</head>
```

The above snippet will change the color of the text inside the paragraph to orange and font-size to 24px.

External CSS

When you write CSS in a separate file and include the file in the HTML using the <link> tag.

```
<head>
<link rel="stylesheet" type="text/ css"
href="style.css">
</ head>
```

“rel” attribute specifies the relationship that the linked file has with the source file.
“type” attribute specifies the internet media type and value “text/css” conveys that it is CSS.
“href” is used to provide the path of the file which we want to link.

```
//style.css
p{
font-size: 24px
color: orange
}
```

The style written inside .css file are not enclosed within any tags.
Note: if you include CSS in all the three ways having the same property and selector, then the inline CSS will have the highest precedence and it will override all the other CSS.

If you include internal CSS as well as external CSS then the precedence depends upon the order in which you specify them inside the <head> tag.

```
<head>
<link rel="stylesheet" type="text/ css"
href="style.css">
<style>
p{
color: red;
font-size:20px
}
</ style>
</ head>
```

Here the CSS written internally will override the external CSS, and the text color will change to red and its font size to 20px.
CSS is always read by browsers from top to bottom, so the one appearing below will override the above one if the selector and property are some.

SELECTING HTML ELEMENTS

An element can be selected using:

- Element selector
- Class selector
- Id selector
- Attribute selector

Element Selector

```
<p>here is some text written inside the
paragraph tag</p>
```

To simply select the above text, we can use element selector which is nothing but using the element name.
So, in order to select the above text and modify its style, we can write:

```
p{
/*property value pairs*/
}
```

The above snippet will select all the paragraph elements present on the webpage.
Note: /**/ is used to enclose comments in CSS. The things written inside/* and */ is not read by browsers.

Class Selector

```
<p class="select-me">here is a paragraph tag which also
has a class specified inside it.</p>
```

If an HTML element has class associated with it, then it can be referenced using class selector in CSS.
Writing a class selector involves writing. (dot) followed by class name.

```
.select-me{
/*property value pairs*/
}
```

The above snippet will select all the elements controlling the **select-me** class.

ID Selector

```
<P id="select-me">
Here is a paragraph tag which also has an id specified
inside it.
</p>
```

If an HTML element has an id associated with it, then it can be referenced using id selector in CSS.
Writing a class selector involves writing # (hash) followed by id name.

```
#select-me{
/*property value pairs*/
}
```

The above snippet will select the element containing **select-me** id.

```
<input type="text"
/>
```

Attribute Selector

Given above is an input tag containing **type** attribute with value text.
If an HTML element has an attribute associated with it, then it can be referenced using attribute selector in CSS.
Writing an **attribute** selector involves writing **element name** followed by [**attribute** =**value**] as shown below

```
Input[type=text]{
/*property value pairs*/
}
```

The above snippet will select all the input elements containing the **type** attributes with value text.

PLAYING WITH COLORS

As we know that colors are integral part of any web application.
Colors in CSS can be specified in following ways:

- Directly writing the name of colors.
- Using hex values of colors.
- Using RGB/RGBA values.

Using Color Names

The total number of color names supported is 140.
Black, blue, red, green, gray, aqua, brown, coral, indigo are few to name.

Hex value

Another way of specifying colors in CSS is by using hexadecimal aka Hex values.
Hexadecimal values are written in following format #rrggbb where rr, gg, and bb are values between 00 and ff.
rr- the amount of red
gg- the amount of green
bb- the amount of blue
Is lowest value and ff the highest.
#000000 forms black color, whereas #FFFFFF forms white.

RGB/RGBA Values

R- red, G-green, B- blue and A-alpha, alpha is a number between 0.0 (fully transparent) and 1.0 (fully opaque).
0 depicting the lowest intensity and 255 the highest.
For writing color in RGB format follow the syntax:

Rgb(R,G,B) where R, G and B can be any values between 0 and 255.
If there is a need to specify opacity: rgba (R G, B, A) is used and for A values between 0.0 -1.0 can be specified.
rgb(0.0.0)

MANIPULATING BACKGROUND

Changing the white background becomes a necessity when it comes to building for the web.
Either you can set the background color or add an image as background.
The property used to modify background color is background-color or background and values can be specified in any format from the ones we learnt.

Using color name:

```
body{
background-color: red;
}
```

Using Hex Value:

```
body{
background-color:#FF0000;
}
```

Using RGB Value:

```
body{
background-color: rgb(255,0,0);
}
```

Will change the background color of page to red.
*simply **background** can also be used instead of **background-color** .

Background Image

You can add image to your webpage in the background.
It's an easy step:
Background or **background-image** property is used to specify **background image** .

```
Background: url(*path to image");
```

Value consists of the url followed by the path to image in brackets. This path can be replaced by some url of image on the web or by the path of image present in your local machine.

Customizing background image

Background-size property enables us to set/ change the size of background image in CSS. It can take values of height followed by width, or contain or cover.
Contain makes sure that the height and width of background image are in accordance with the height and width of content area.
Cover makes sure that the height and width of the image spans the entire width and heights of the available area.
Valid background-size values are as follows

```
Background-size :100%;
Background-size: cover;
Background-size: contain;
```

*height and width can be specified in multiple ways which we will be learning ahead.

Background-repeat property enables us to control if the image will be repeated to fill the content area or not.
It can take either of repeat or no-repeat as values. Repeat specifies that the image is allowed to be repeated to fill the content area and no-repeat specifies it is not.

```
Background-repeat: no-repeat;
Background-repeat: repeat;
```

All these are valid declarations.

BEAUTIFYING TEXT USING CSS

Font-style, color, font-size, formatting of text is inevitable whether you are writing a novel or creating a web application.
Html cannot do these things alone. That's where CSS comes to rescue. Let's see how to modify properties of text.

Changing Colour

The text, when in contrast with the background, enhances the user readability.
To change color of text in CSS color property is used followed by value or name of color.

```
<p>Hey! Make me blue</p>
//style.css
p{
        color: blue;
}
```

The above CSS will change color of text inside every paragraph tag to blue.

Changing Alignment

Justified, center, right or left? How? Let's see how we can change alignment of text.

```
<p>Hey! I am here.</p>
//style.css
p{
        text-align: center;
}
```

The property used is text-align and values can be anyone of center, left, right and justify.
The above CSS will make the text inside paragraph tag center aligned.

Changing Font size

The size of the text plays a crucial role in writing, for prioritizing and emphasizing.
Font size in CSS can be changed using font-size property.
The value of font-size can be either set using pixels (px) or em.
1em = 16px;

```
<p>Hey! Please increase my font-size.</p>
//style.css
p{
        font-size: 32px;  /*can be specified
using em as 2em*/
}
```

The above will increase font size of text to 32px.

Changing Font weight

Font weight is used to change the weight (boldness) of the font.
The font-weight property is used to modify font-weight.
Values can be normal, bold, bolder, and lighter or any value from 100, 200, 300, 400, 500, 600, 700, 800 or 900. 100 being the lightest and 900 being the boldest.

```
<p>Hey! Please change my font-weight.</p>
//style.css
p{
        font-weight: 400;  /*can be anything
from above values*/
}
```

The command given above will change the font-weight of the text to 400.

Changing Font family

The font family or font can be changed using font-family property.

```
<p>Hey! Please change my font-family.</p>
//style.css
p{
        font-family:'Roboto', sans-serif;
}
```

The command given above will change the font-family to Roboto. If the roboto font is not found then sans-serif font is applied. It uses a fallback system.

HEIGHT WIDTH MARGIN AND MORE

CSS gives freedom to modify height, width, margin, apply border and padding to elements.
These properties come handy while making any web application visually appealing.
Let’s explore how we can use these properties.

Height and Width

To change the height of any element height property is used.
Similarly, to change width, width property is used.
Height and width can take values in pixels(px) and percentage.
Let’s see how it is done.

```
<img src="img.png" class="panda">Wow</div>
//style.css
.panda{
        height: 400px;
        width: 400px;
}
```

The above CSS will change the height and width of div to 400px.

```
.panda{
        height: 50%;
        width: 100%;
}
```

This will change the height to 50% of the available container height and width to 100% of the available container width.

Border

We have to admit that things do look better with borders!
To give **border** to any element we can use border-**width** , border-**style** and border-**color** properties.
Border-width can take any value in **pixels** , border-**style** provides variations like **dashed** , **solid** , **dotted** and border-**color** can take any **valid** color value.

```
<div class="rocket">The box has margin around
it.</ div>
// style.css
.rocket{
        border-width: 2px;
        border-style: solid;
        border-color: black;
}
```

The above css will apply a solid border to div with width 2px and blue color.
There is also a shorthand property to apply border which goes like:

```
.rocket{
        border: 2px solid black;
}
```

Where 2px is the border-width, solid is the border-style and the border-color is black.
If we need curved edges, then border-radius property can be used.

```
.rocket{
         border: 2px solid black;
        border-radius: 20px;
}
```

Making its value 50%, will transform it to a circle.

Margin

We create margins to separate the main content from the page's edge.
For a webpage, Margin is used to create space around elements outside the border.
We use margin-top, margin-right, margin-bottom and margin-left properties which in turn applies margin to top, right, bottom and left sides respectively.

```
<div class="starlord">Starlord, who?</ div>
.starlord{
        margin-top: 20px;
        margin-right: 30px;
        margin-bottom: 20px;
        margin-left: 30px;
}
```

The above command will change the default margin around div to 20px from the top, 30px from the right, 20px from the bottom and 30px from the left.
Is there any shorthand property for margin? Hell Yeah!

```
.starlord{
        margin: 20px 30px 20px 30px; /*In order
top, right, bottom, left*/
}
```

Also, we can write:

```
.starlord{
        margin: 20px 30px;
}
```

These values will repeat themselves for bottom and left and will end up giving the same result as the above.
Also,

```
.starlord{
        margin: 20px;
}
```

It will create a margin of 20px around all sides.

Padding

Padding is used to create space around elements, inside the border.
We use **padding-top** , padding-**right** , padding-**bottom** and padding-**left** properties which in turn applies padding to top, right, bottom and left sides respectively.

```
<div class="starlord">Starlord, who?</ div>
.starlord{
        padding-top: 20px;
        padding-right: 30px;
        padding-bottom: 20px;
        padding-left: 30px;
}
```

The above command will change the default padding around div to 20px from the top, 30px from the right, 20px from the bottom and 30px from the left.
Is there also a shorthand property for padding? Of course.

```
.starlord{
        padding: 20px 30px 20px 30px; /*In order
top, right, bottom, left*/
}
```

Also, we can write:

```
.starlord{
        padding: 20px 30px;
}
```

These values will repeat themselves for the bottom and left and will end up giving the same result as the above.
Also,

```
.starlord{
        padding: 20px;
}
```

It will create a padding of 20px around all sides.

POSITIONING HTML ELEMENTS

Better positioning of objects in a room gives it a neat and a managed look. Similarly, Positioning of elements is an integral part of building for web. The property used to change position of elements is position. Obviously! Want something sticking to the screen or make an element adjust its position relative to itself or place elements exactly where you want.

Static Position

For a room, the default position of a flower vase would be somewhere on the table, near the window.
For a webpage, an element which is positioned **static** is positioned according to the normal flow of the page.
By **default** the position of all HTML elements is set to static.

You can take the back seat, nothing to do here.

```
<div class="static-me">I am not special.</ div>
.static-me{
        position: static;
}
```

This is how you must give static position to elements, if ever required.

Relative Position

The rug we place in the room is near the door/bed/bathroom. So we keep changing its position relative to our need.
Relative positioning enables us to position the element relative to itself. That is, the element will change position relative to its default position.

```
<div class="relative-me">What does this even
mean? Look up.</ div>
.relative-me{
        position: relative;
        top: 10px;
}
```

This will move the div 10px towards bottom from where it would have been by default.
*top, left, right, bottom are used with position to specify distance from respective axes.

Fixed Position

We tend to fix the position of a painting in a room (till the time we get another one).
Fixed positioning enables us to position the element **relative to the screen.** If an element is given a fixed position it will occupy that position on the screen.

```
<div class="fixed-me">I get attached soon.</ div>
.fixed-me{
        position: fixed;
        top: 50px;
        left: 50px;
}
```

This will move the div 50px towards bottom and 50px towards right from its original position and will keep it fixed there.
Fixed elements stay at the same place on screen even if user scrolls.

Absolute Position

Absolute positioning enables us to position the element exactly where we want it. We can use top, right, bottom, let properties to place it wherever we want.

```
<div class="absolute-me">I am the best thing you have in
store.</ div>
.absolute-me{
        position: absolute;
        top: 50px;
        left: 50px;
}
```

This will place the div 50px towards bottom and 50px towards right from its original position.

MANIPULATING DISPLAY PROPERTIES

Remember, the excitement we get while playing tetris to change the default shape of the objects, just to win the game.
Display property in CSS is used to control the default rendering behavior of an element.
Different elements have different default rendering behavior.
For e.g.: <p> and are block level elements whereas <a> and are inline elements.

Display Block

An element which has display property set to block is displayed on a new line and spans across the whole width.
To change the display behavior of elements display property is used. And to make an element display as block, display property is set to value block.

```
<span class="block-me">I need space.</span>

//style.css
.block-me{
            display:block;
}
```

The above snippet will change the display behavior of to block.
* is an inline element by default.

Setting display behavior to inline-block enables us to set the height and width of the element and also occupy multiple elements in a single row.

```
<div class="inline-me">Hey, I wish we were
together.</ div>
<div class="inline-you">What's stopping
you?</ div>

// style.css
 .inline-me, inline-you{
        display: inline-block;
        height: 20px;
        width: 50%;
}
```

The above snippet will make both the divs inline where each will one take 50% of the container width and the output will be similar to the one shown in the above image.
*div is a block element by default.

None

Display value none is used to completely remove the element.

```
<div class="not-me">I didn't do anything.</ div>
//style.css
 .not-me{
        display: none;
}
```

The above snippet will remove the div from the view.

OVERFLOW

The **overflow** property in CSS, is used to specify what happens if content goes beyond the context area.
Two options: Either trim the content or find a genie.

Trim Overflowing Content

To trim the overflow content **hidden** value is passed to the overflow property. This will make the overflowing content invisible.

```
<div class="trimmed">I know I am going to get
slashed</div>

//style.css
 .trimmed{
       height: 20px;
       width: 400px;
       overflow:hidden;
}
```

The height and width of div is set to be 20px and 400px respectively, so the content overflows and since it is assigned a value of hidden, it get trimmed.

Add Scrollbars

The code is magical itself, no need of a genie.
To display the overflow content we can set the display property to scroll. This will add a horizontal as well as a vertical scrollbar to the div.

```
<div class="scroll">I knew you would save
me</div>

//style.css
 .scroll{
        height: 20px;
        width: 400px;
        overflow:scroll;
}
```

Scrollbar appear around the div and the content can be seen upon scroll.

Overflow Auto

Setting overflow to **auto** is an intelligent move. It automatically detects if the content is trimmed and adds scrollbar if necessary, and only along the axis where it is required.

```
<div class="auto-me">Niflare world</ div>
.auto-me{

        height:40px;
        width:200px;
        overflow:auto;
}
```

The scrollbar appears since the content is getting trimmed, also only vertical scrollbar appears.

Overflow Visible

This is not something you would want. But setting overflow to **visible** will make the content render even if it goes outside the container. Surprisingly, this is the default behavior.

```
<div class="visible-me">NIFLARE</ div>

//style
 .visible-me{
              height:40px;
              width:200px;
              overflow:visible;
             border: 1px solid #000;
}
```

WHAT ARE PSEUDO CLASSES?

The way we can twist and turn a piece of clay and change it into the shape/state we want. Similarly in HTML

An element can have multiple states.
These states can be anything like hovered, focused, visited, active depending upon the user's actions.
To change the properties of elements in these states we have **pseudo classes.**

Pseudo Classes are written in the following manner:

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

Let's explore the available pseudo classes.

Changing Element Properties on Hover

Have you ever noticed the text being focused when you hover over it?
For the same purpose we use pseudo class **hover** to target element when they are hovered.
Whatever property and value we give to hover pseudo class will apply only when the element is hovered.

```
<input class="hover-me type="button"
value="Hover" />

//style.css
 .hover-me:hover{ background-color:green}
```

Changing properties of an active link, active pseudo class is used.
A link is in active state when clicked.

```
<a herf="www.niflare.com">Click me</a>

//style.css
a:active{
    color: red;
}
```

Change Properties of Unvisited Links

To change the properties of links which are yet unvisited, **link** pseudo class is used.

```
<a herf=www.niflare.com/>Don't Click me</a>

//style.css
 a:link{
    color: red;
```

Change Properties of Visited Links

To change the properties of links which are already visited, visited pseudo class is used.
Visited a link is nothing but clicking on it.

```
<a herf="www.niflare.com">Click me</a>

//style.css
 a:visited{
    color: black;
```

WRITING CSS FOR MOBILE

We all are avid users of the mobiles and tablets. It becomes necessary to make sure that the things developed are compatible and properly visible across all devices.

Media queries comes to rescue.

Using media queries we can write styles based on screen sizes, specifically targeted to tablets, phones and desktops.

WRITING MEDIA QUERIES

To write **media** queries @**media** property is used.
It is written as follow:

```
@media screen and (condition) {
//Whatever CSS you want to write
}
```

Here condition consists of conditions involving height and width of screen. And inside the media block we can write whatever CSS we want specifically for that screen size.

To write CSS for screen sizes wider than or equal to 720px, we can write

```
@media screen and (min-width: 720px){
   p{
         Font-size: 16px;
    }
}
```

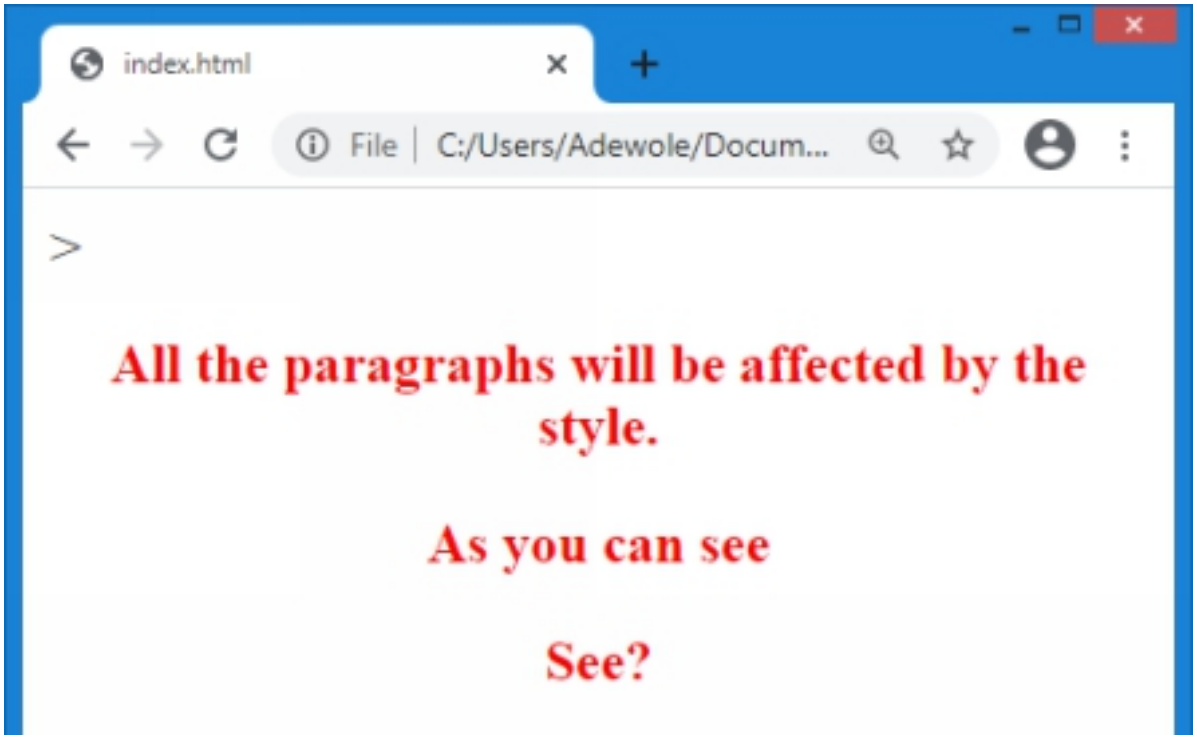
The above CSS will change font-size of every paragraph to 16px on devices with screen size 720px or more than 720px.

Similarly,
Max-width can be used to specifically for screen sizes equal to or less than specified.
On the same grounds **max-height** and **min-height** can also be used.

PLAYGROUND
(SELECTOR)

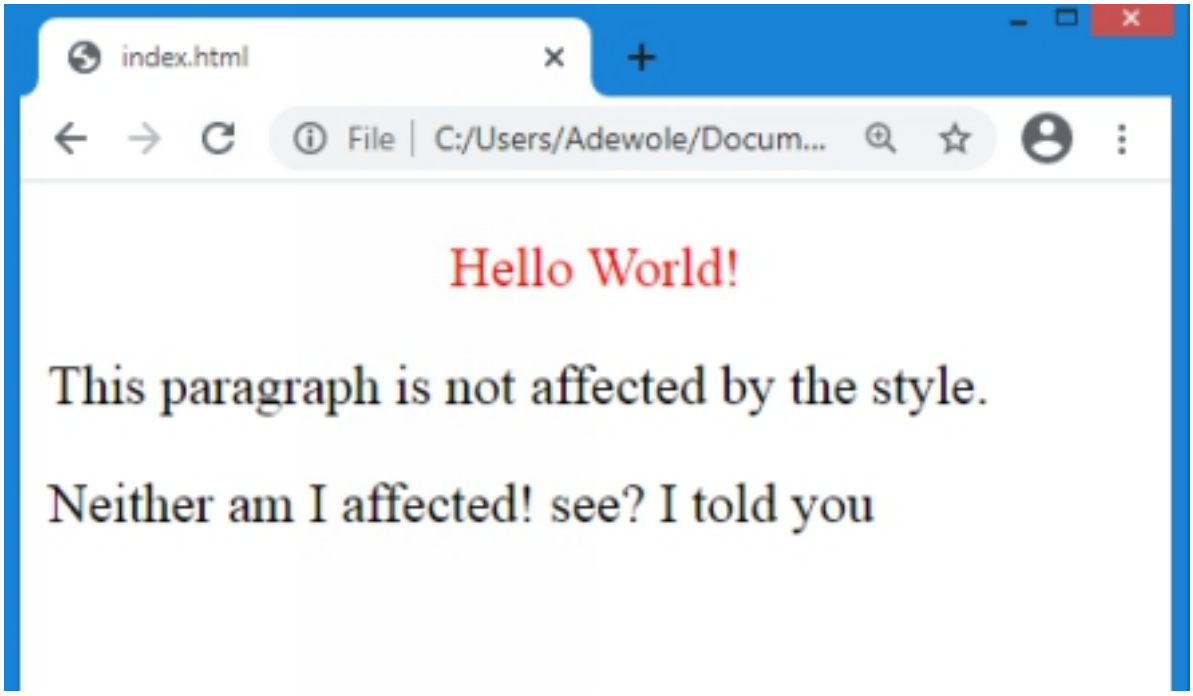
```
<!DOCTYPE HTML>
<html>

<head>

 <style>
   /* All the CSS properties are declared in the style tag in
HTML  */
     p {
        text-align: center;
        color: red;
        font-weight: bold;
     }
 </style>

</head>

<body>
 <p>All the paragraphs will be affected by the style.</p>
 <p id="para1">As you can see</p>
 <p>See?</p>
</body>

</html>
```

Element Selector


index.html
File | C:/Users/Adewole/Docum...
>
All the paragraphs will be affected by the style.
As you can see
See?

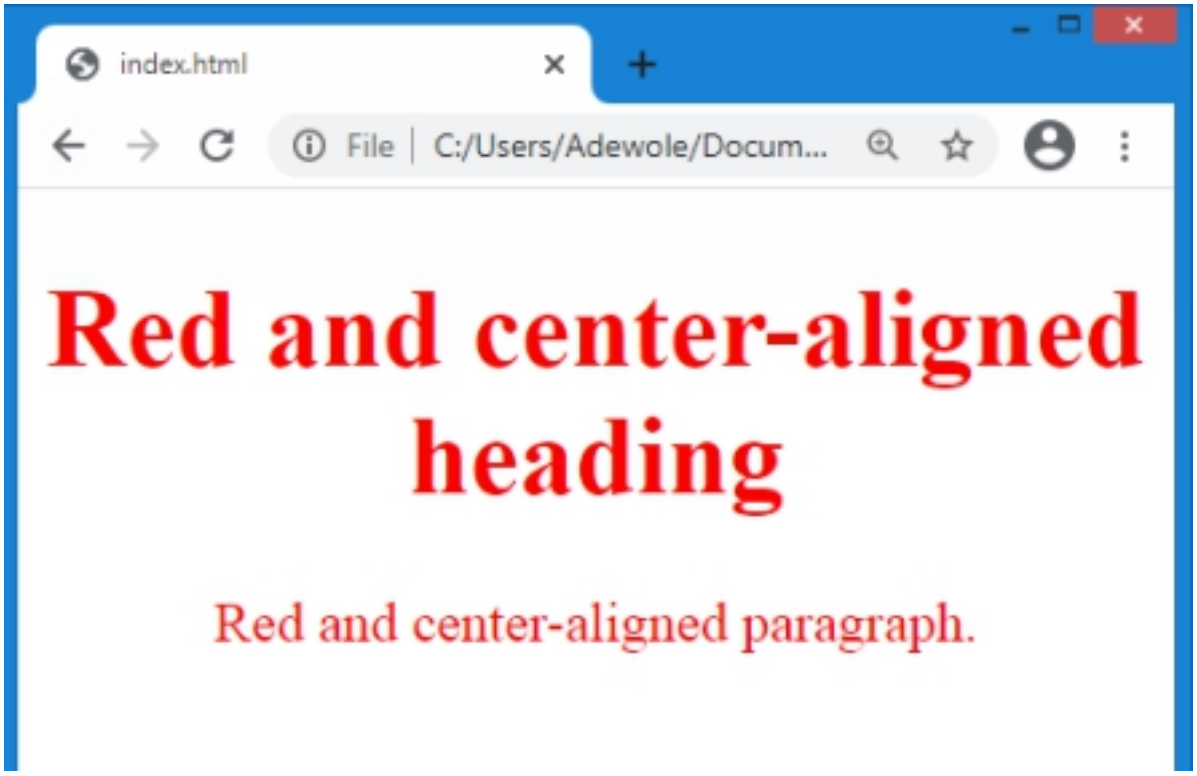
ID Selector

```
<!DOCTYPE html>
<html>

<head>

 <style>
  /* When you specify the ID of the tag, the style is
applied only on that perticular tag */
  #para1 {
     text-align: center;
     color: red;
  }
 </style>

</head>

<body>
 <p id="para1">Hello World!</p>
 <p>This paragraph is not affected by the style.</p>
 <p>Neither am I affected! see? I told you</p>
</body>

</html>
```


index.html
File | C:/Users/Adewole/Docum...
Hello World!
This paragraph is not affected by the style.
Neither am I affected! see? I told you

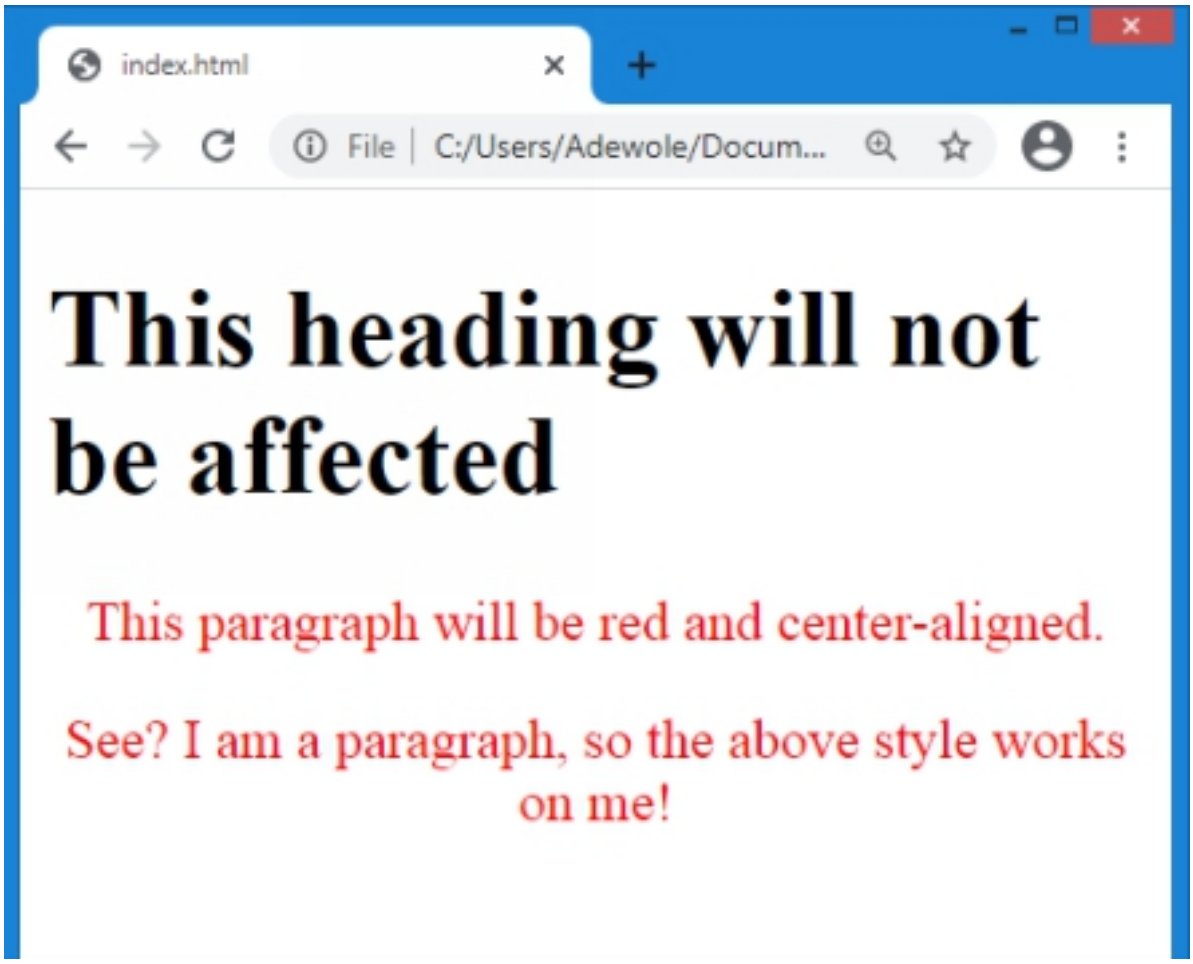
Class Selector

```
<!DOCTYPE html>
<html>

<head>

 <style>
  /* Here we have defined style by assigning it to a class
*/
  .center {
     text-align: center;
     color: red;
  }
 </style>

</head>

<body>
 <h1 class="center">Red and center-aligned
heading</h1>
 <p class="center">Red and center-aligned
paragraph.</p>
</body>
```


index.html
File | C:/Users/Adewole/Docum...
Red and center-aligned heading
Red and center-aligned paragraph.

Class Selector (for only particular elements)

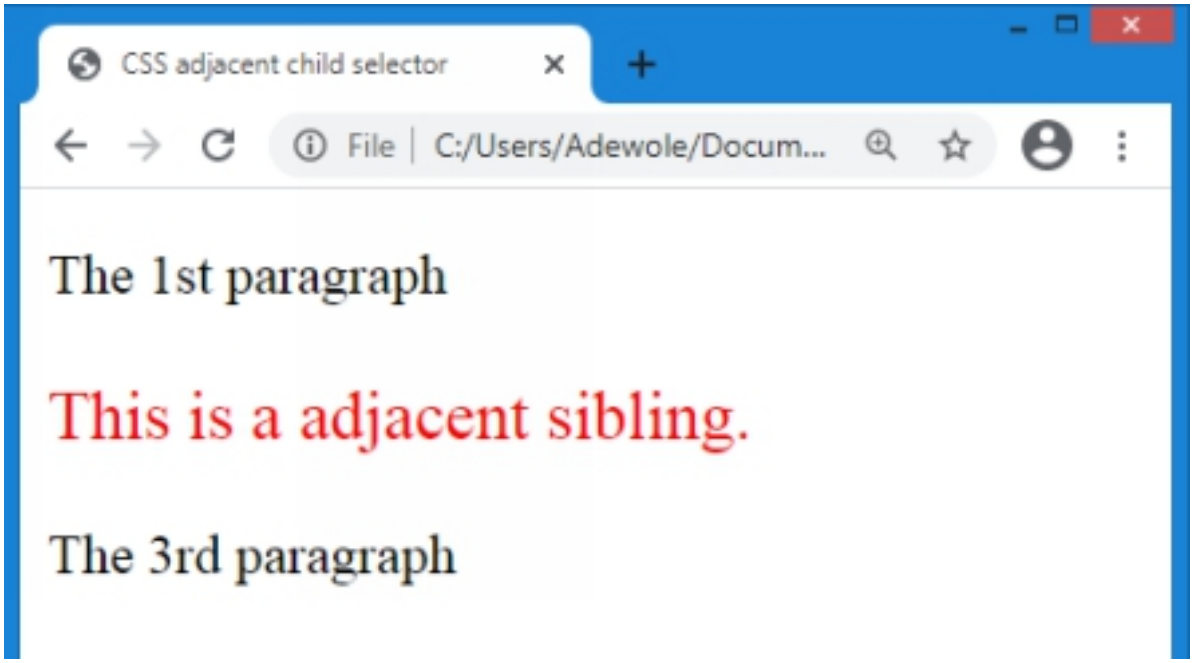
```
<!DOCTYPE html>
<html>

<head>

 <style>
  /* Here we are defining which element can use the style
class */
   p.center {
      text-align: center;
      color: red;
   }
 </style>

</head>

<body>
 <h1 class="center">This heading will not be
affected</h1>
 <p class="center">This paragraph will be red and center-
aligned.</p>
 <p class="center">See? I am a paragraph, so the above
style works on me!</p>
</body>
```


index.html
File | C:/Users/Adewole/Docum...
This heading will not be affected
This paragraph will be red and center-aligned.
See? I am a paragraph, so the above style works on me!

Adjacent Sibling Selector

The adjacent sibling CSS selector is used to select HTML element that are adjacent siblings of some other HTML element.

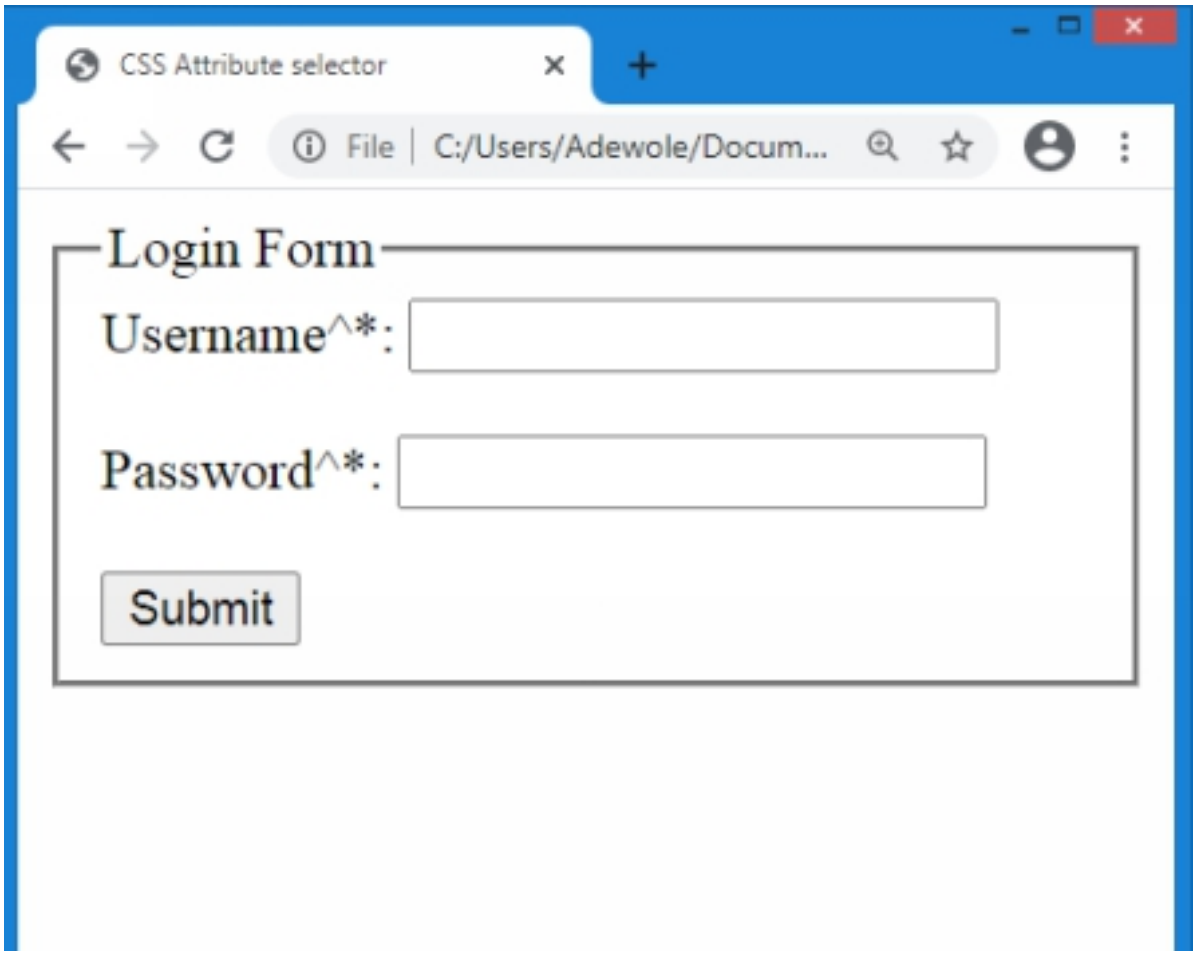
```
<!DOCTYPE html>
<html>

<head>
   <title>CSS adjacent child selector</title>

  <style type="text/css">
        p+div
        {font-size:20px;
        color:red;}

   </style>
  </head>
   <body>
      <p>The 1st paragraph</p>
      <div>
       <p>This is a adjacent sibling.</p>
      </div>
      <p>The 3rd paragraph</p>
   </body>
   </html>
```


CSS adjacent child selector
File | C:/Users/Adewole/Docum...
The 1st paragraph
This is a adjacent sibling.
The 3rd paragraph

Attribute Selector

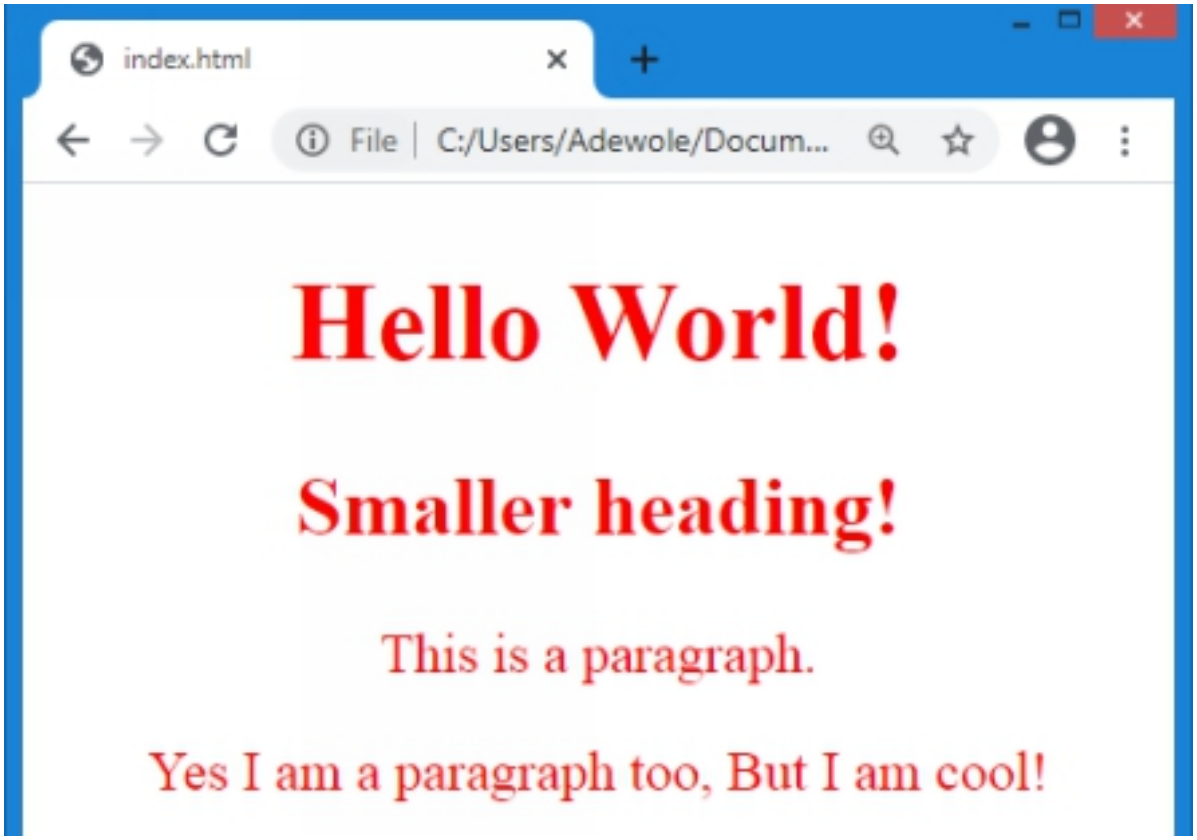
This is used to apply styles to HTML elements with particular attributes.

```
<!DOCTYPE html>
<html>

<head>
   <title>CSS Attribute selector</title>

  <style>
   /^*This code will change background color to yellow
of 'label' tag having 'for' attribute^*/
      labal[for]
      {background:yellow}

   </style>
  </head>

   <body>
    <form>
       <fieldset>
        <legend>Login Form</legend>
        <label for="user"> Username^*: </label><input
type="text" id="user"/><br><br>
        <label for="pass"> Password^*: </label><input
type="password" id="pass"/><br><br>
        <input type="submit" value="Submit"/>
        </fieldset>
   </form>
   </body>
```


CSS Attribute selector
File | C:/Users/Adewole/Docum...
Login Form
Username^*:
Password^*:
Submit

Group Selector

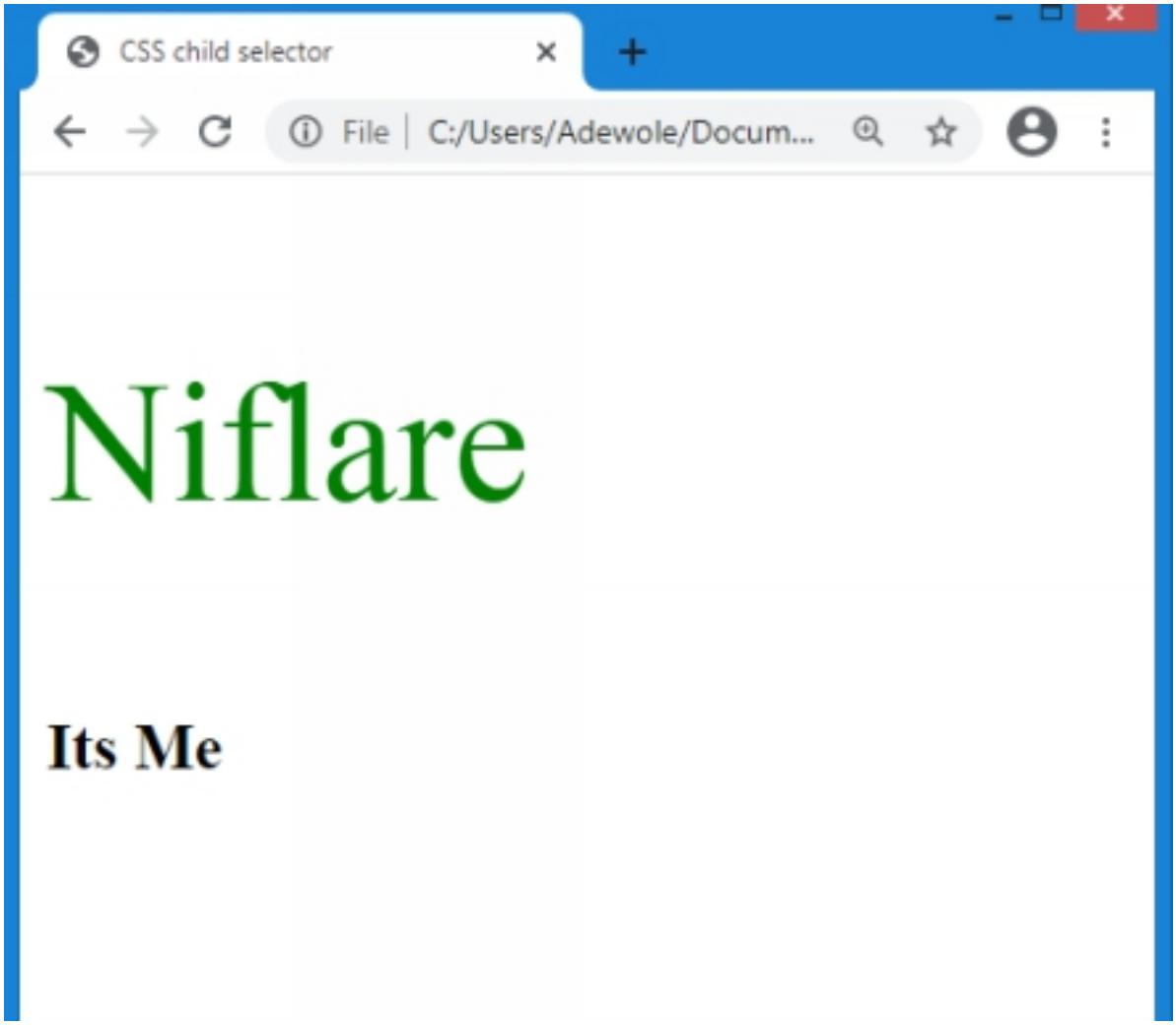
```
<!DOCTYPE html>
<html>

<head>

 <style>
  /* Here we are grouping together elements that require
CSS style */
  h1, h2, p {
      text-align: center;
      color: red;
  }
 </style>

</head>

<body>
 <h1>Hello World!</h1>
 <h2>Smaller heading!</h2>
 <p>This is a paragraph.</p>
 <p>YesI am a paragraph too, ButI am cool!</p>
</body>

</html>
```


index.html
File | C:/Users/Adewole/Docum...
Hello World!
Smaller heading!
This is a paragraph.
Yes I am a paragraph too, But I am cool!

Child or Direct Selector

This is very similar to descendants but have different functionality. This rule will render all the paragraphs in red and size will be 20px if they are direct child of <body> element. Other paragraph put inside other element like would not have any effect of this rule.

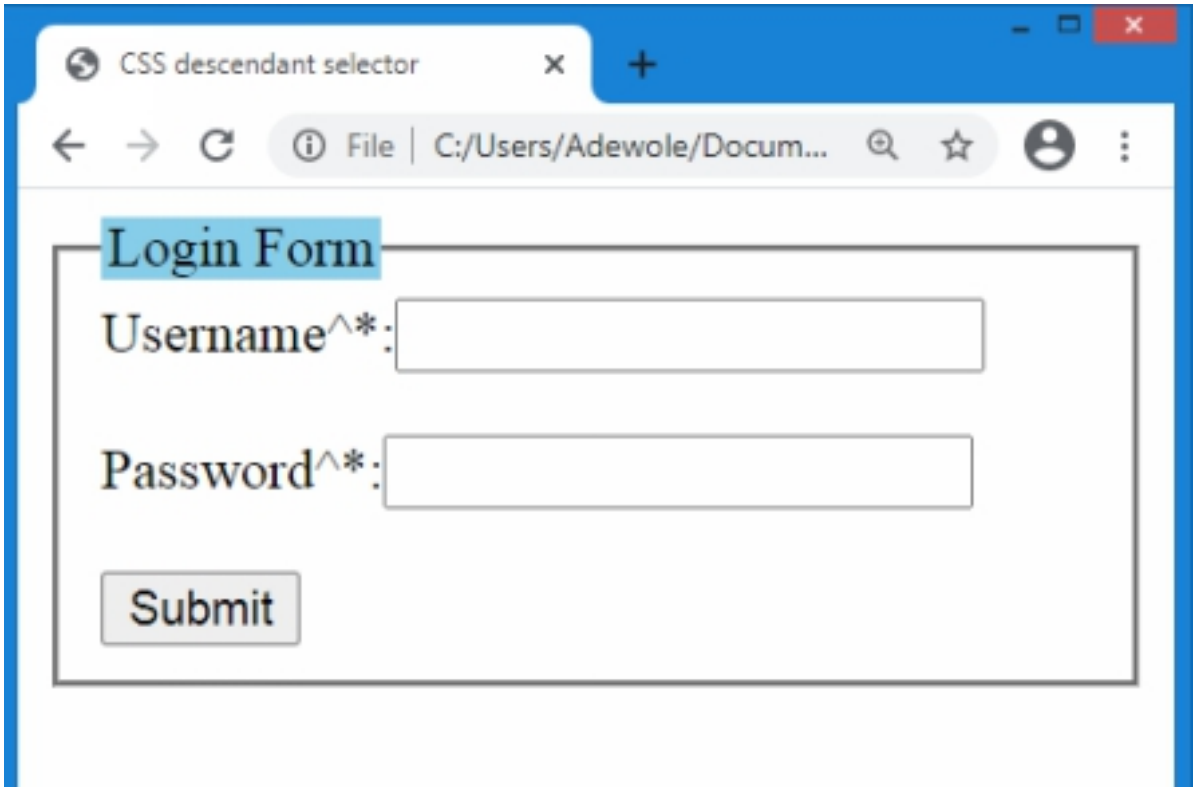
```
<!DOCTYPE html>
<html>

<head>
  <title>CSS child selector</title>


 <style type="text/css">
   div>p{font-size: 50px;
   color: green;}

  </style>
 </head>
  <body>
    <div>
      <p>Niflare
      <h3>Its Me</h3>
      </p>
    </div>
  </body>
  </html>
```


CSS child selector
File | C:/Users/Adewole/Docum...
Niflare
Its Me

Descendant Selector

This selector selects all elements that are descendants of a specified element.
A descendant of an element could be a child, grandchild, great-grandchild, etc. of that element.

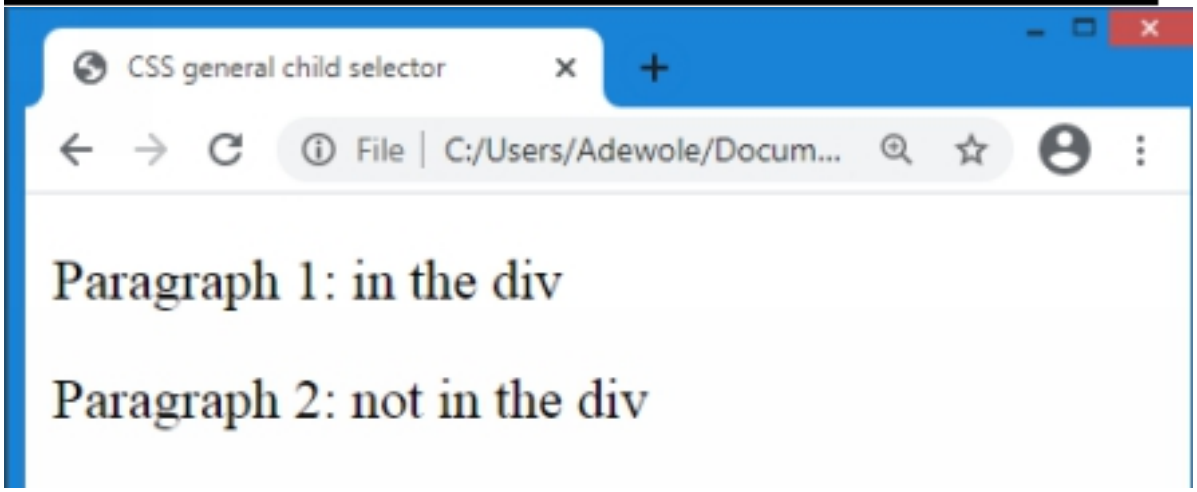
```
<!DOCTYPE html>
<html>

<head>
  <title>CSS descendant selector</title>


 <style>
   body legend{background: skyblue;}
  </style>
 </head>
  <body>
    <form>
     <fieldset>
       <legend>Login Form</legend>
       <label for="user">Username^*:</label><input
type="text" id="user"/><br><br>
       <label for="pass">Password^*:</label><input
type="password" id="pass"/><br><br>
       <input type="submit" value="Submit"/>
    </fieldset>
    </form>
  </body>
  </html>
```


CSS descendant selector
File | C:/Users/Adewole/Docum...
Login Form
Username^*:
Password^*:
Submit

General Sibling Selector

The CSS general sibling selector is used to select HTML elements which have the same parent.

```
<!DOCTYPE html>
<html>

<head>
  <title>CSS general child selector</title>
 <style type="text/css">
    div~p
      {background: lightgray}
  </style>
 </head>
  <body>
    <div>
    <p>Paragraph 1: in the div</p></div>
    <div>
    <p>Paragraph 2: not in the div</p></div>
  </body>
  </html>
```

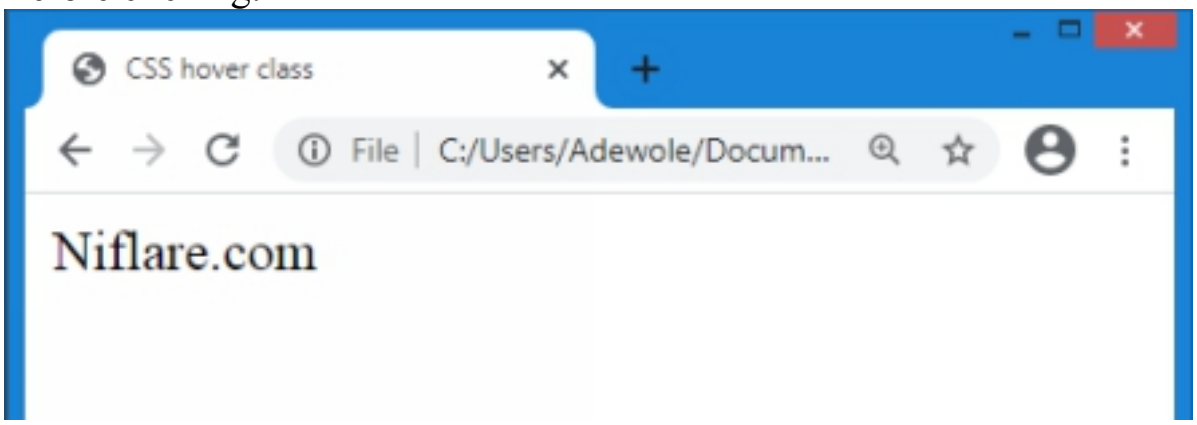


PLAYGROUND (PSEUDO CLASS)

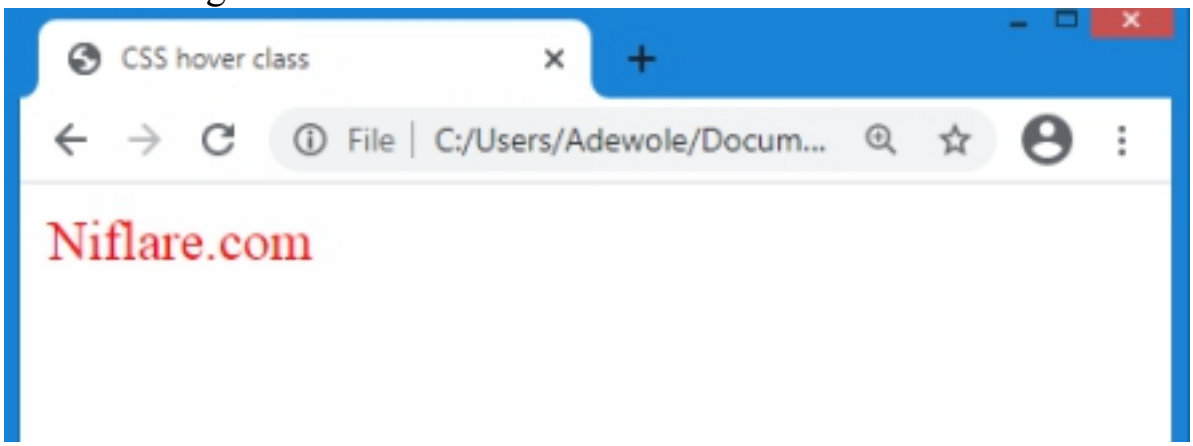
Hover Class

```
<!DOCTYPE html>
<html>

<head>
   <title>CSS general child selector</ title>
 <style type="text/ css">
     div~p
       {background: lightgray}
  </ style>
 </ head>
  <body>
     <div>
     <p>Paragraph 1: in the div</ p></ div>
     <div>
     <p>Paragraph 2: not in the div</ p></ div>
  </ body>
  </ html>
```

Before clicking:

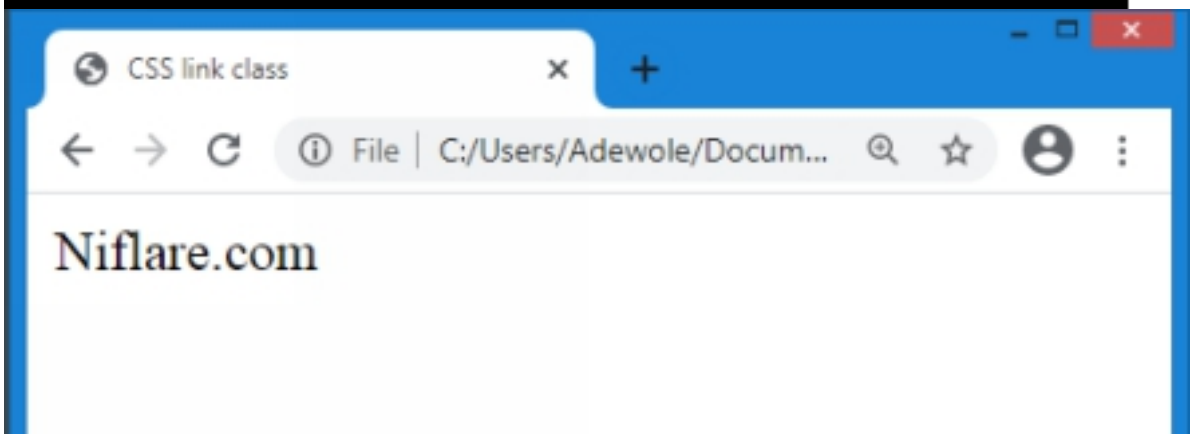


After clicking:



Link Class

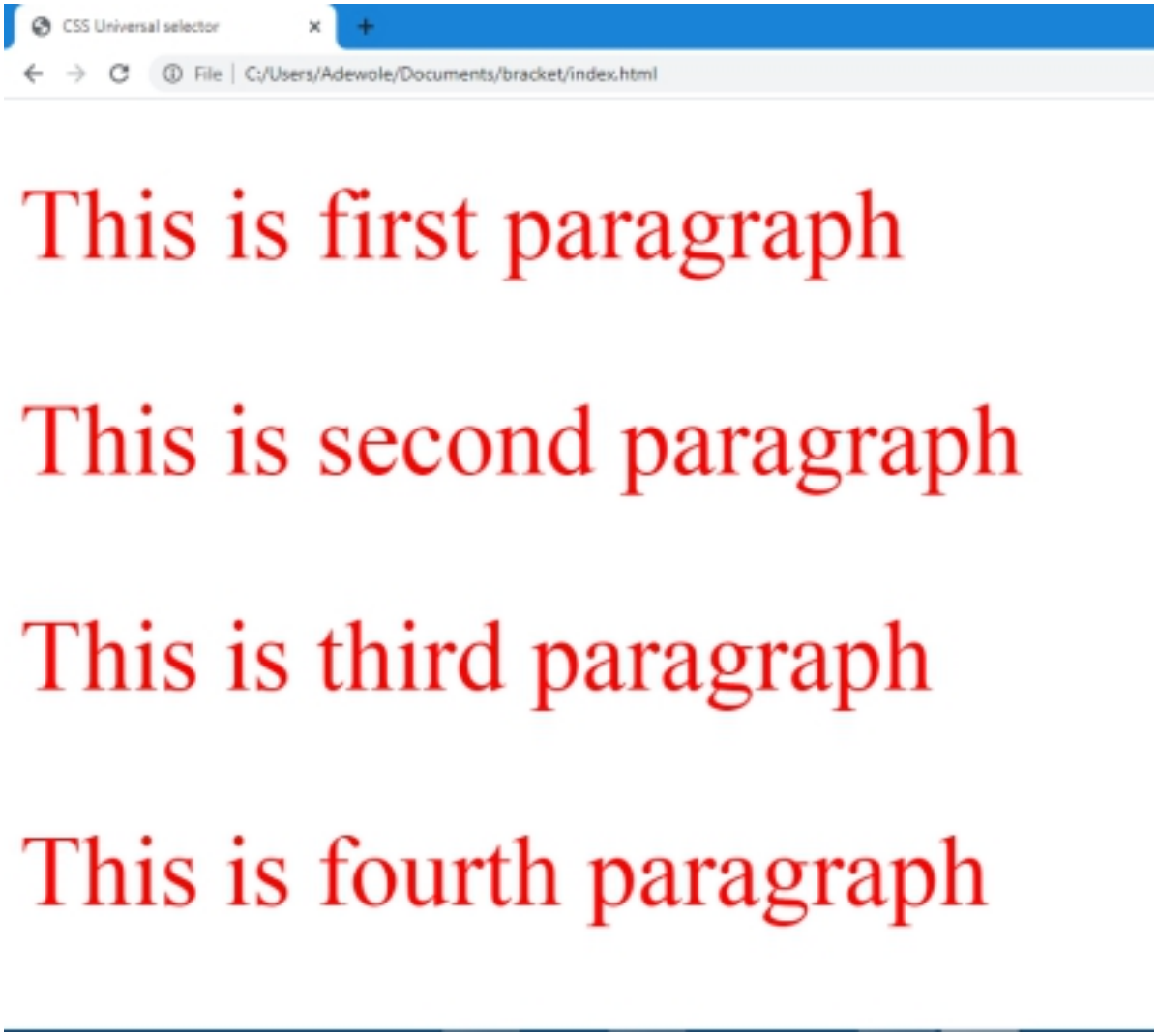
```
<!DOCTYPE html>
<html>

<head>
   <title>CSS link class</ title>
 <style type="text/ css">

    a:link{color: red;}
  </ style>
 </ head>
  <body>
     <a herf="#">Niflare.com</ a>
  </ body>
  </ html>
```



Universal Selector

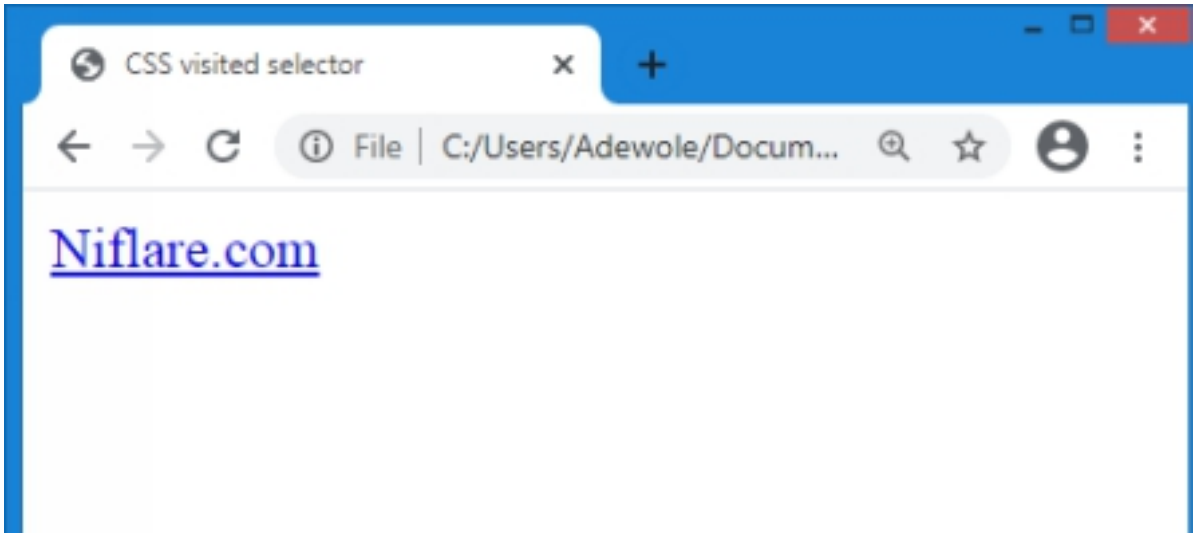
This selector will target every single element on the page.

```
<!DOCTYPE html>
<html>

<head>
 <title>CSS Universal selector</ title>
 <!-- css code is written in style tag .-->
 <style type="text/ css">
  *{
    font-size:50px;
    color:red;
  }
 </ style>

</ head>

<body>
 <p>This is first paragraph</ p>
 <p>This is second paragraph</ p>
 <p>This is third paragraph</ p>
 <p>This is fourth paragraph</ p>
</ body>

</ html>
```


CSS Universal selector
File | C:/Users/Adewole/Documents/bracket/index.html
This is first paragraph
This is second paragraph
This is third paragraph
This is fourth paragraph

Visited Class

```
<!DOCTYPE html>
<html>

<head>

 <title>CSS visited selector</ title>
 <style type="text/ css">
   /* Use this class to add special style to a visited link.
*/
   a:visited
   {
     color:gray;
   }
 </ style>
</ head>

<body>
 <a href="#"> Niflare.com </ a>
</ body>

</ html>
```


CSS visited selector
File | C:/Users/Adewole/Docum...
Niflare.com

PLAYGROUND (TEXT FORMATTING)

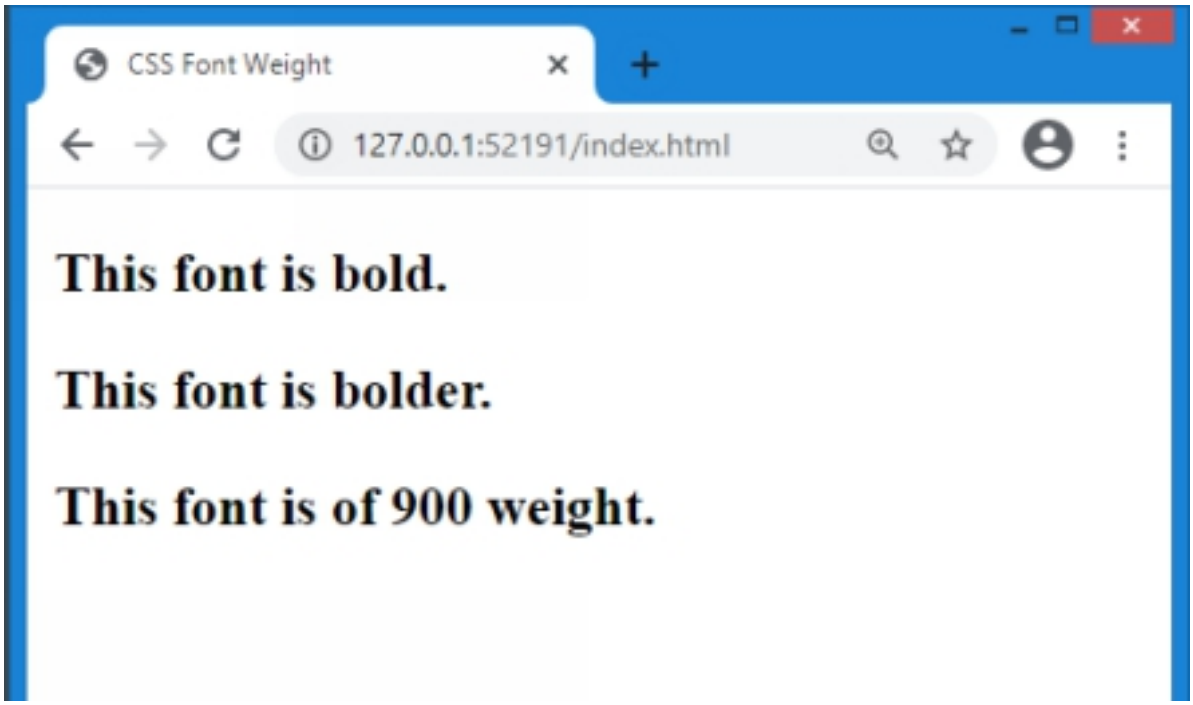
```
<!DOCTYPE html>
<html>

<head>
 <title> CSS Font Family </ title>

 <style>
  h2
  {
   font-family:Comic sans MS;
  }
 </ style>
</ head>

<body>
 <h2> This is NIFLARE! I am in comic sans MS. </ h2>
</ body>
</ html>
```

Font Family


CSS Font Family
127.0.0.1:52191/index.html
This is NIFLARE! I am in comic sans MS.

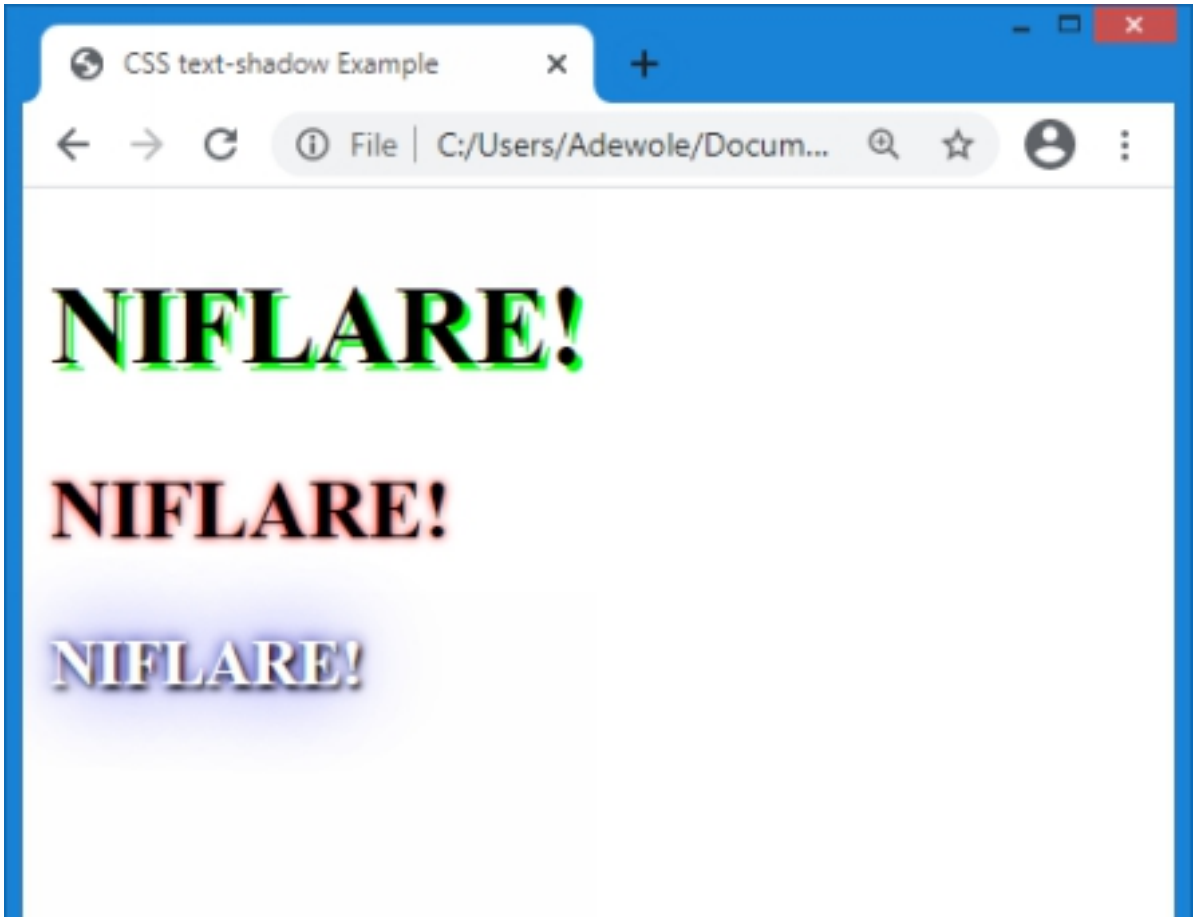
```
<!DOCTYPE html>
<html>
        <head>
 <title> CSS Font Weight </ title>

 <style>
  .bold{
    font-weight:bold;
   }
   .bolder{
    font-weight:bolder;
   }

   .weight{
    font-weight:900;
    }
 </ style>
        </ head>

<body>
 <p class="bold">
  This font is bold.
 </ p>

 <p class="bolder">
  This font is bolder.
 </ p>

 <p class="weight">
  This font is of 900 weight.
 </ p>
</ body>
</ html>
```

Font Weight


CSS Font Weight
127.0.0.1:52191/index.html
This font is bold.
This font is bolder.
This font is of 900 weight.

```
<!DOCTYPE html>
<html>
        <head>
         <title>CSS text-shadow Example </title>
 <style>
  h1
  {
   text-shadow: 2px 2px #00ff00;
  }

  h2
  {
   text-shadow: 0 0 3px #FF0000;
  }

  h3
  {
   color: white;
   text-shadow: 1px 1px 2px black, 0 0 25px blue, 0 0 5px
darkblue;
  }
 </style>

</head>
        <body>

         <h1>NIFLARE!</h1>
        <h2>NIFLARE!</h2>
         <h3>NIFLARE!</h3>

        </body>
</html>
```

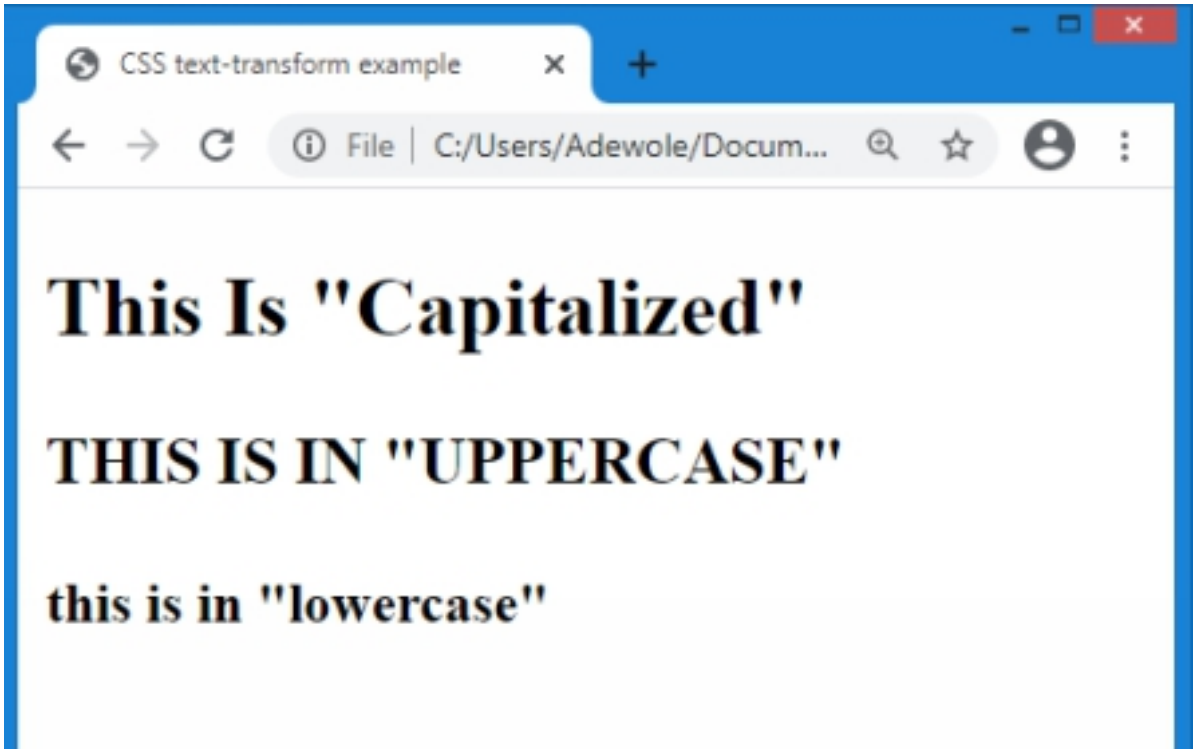
Text Shadow


CSS text-shadow Example
File | C:/Users/Adewole/Docum...
NIFLARE!
NIFLARE!
NIFLARE!

Text Transform

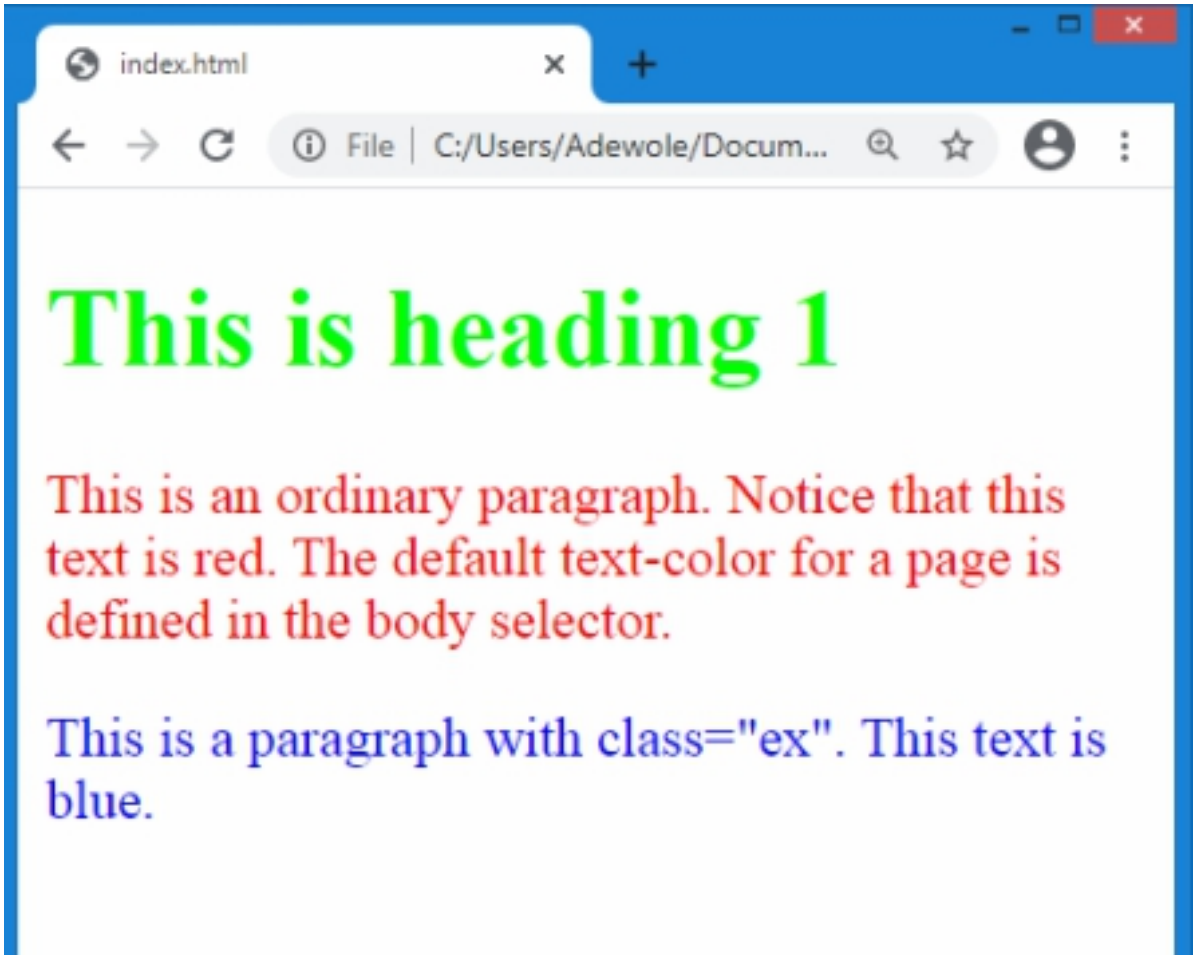
The text-transform property changes the capitalization of text within an element.

```
<!DOCTYPE html>
<html>
	<head>
		<title>CSS text-transform example</title>
	<style>
   h2
   {
     text-transform:capitalize;
   }

   h3
   {
	text-transform:uppercase;
   }

   h4
   {
	text-transform:lowercase;
   }
  </style>
</head>

<body>
  <h2>This is "capitalized"</h2>
  <h3>This is in "uppercase"</h3>
  <h4>This is in "lowercase"</h4>
</body>

</html>
```


CSS text-transform example
File | C:/Users/Adewole/Docum...
This Is "Capitalized"
THIS IS IN "UPPERCASE"
this is in "lowercase"

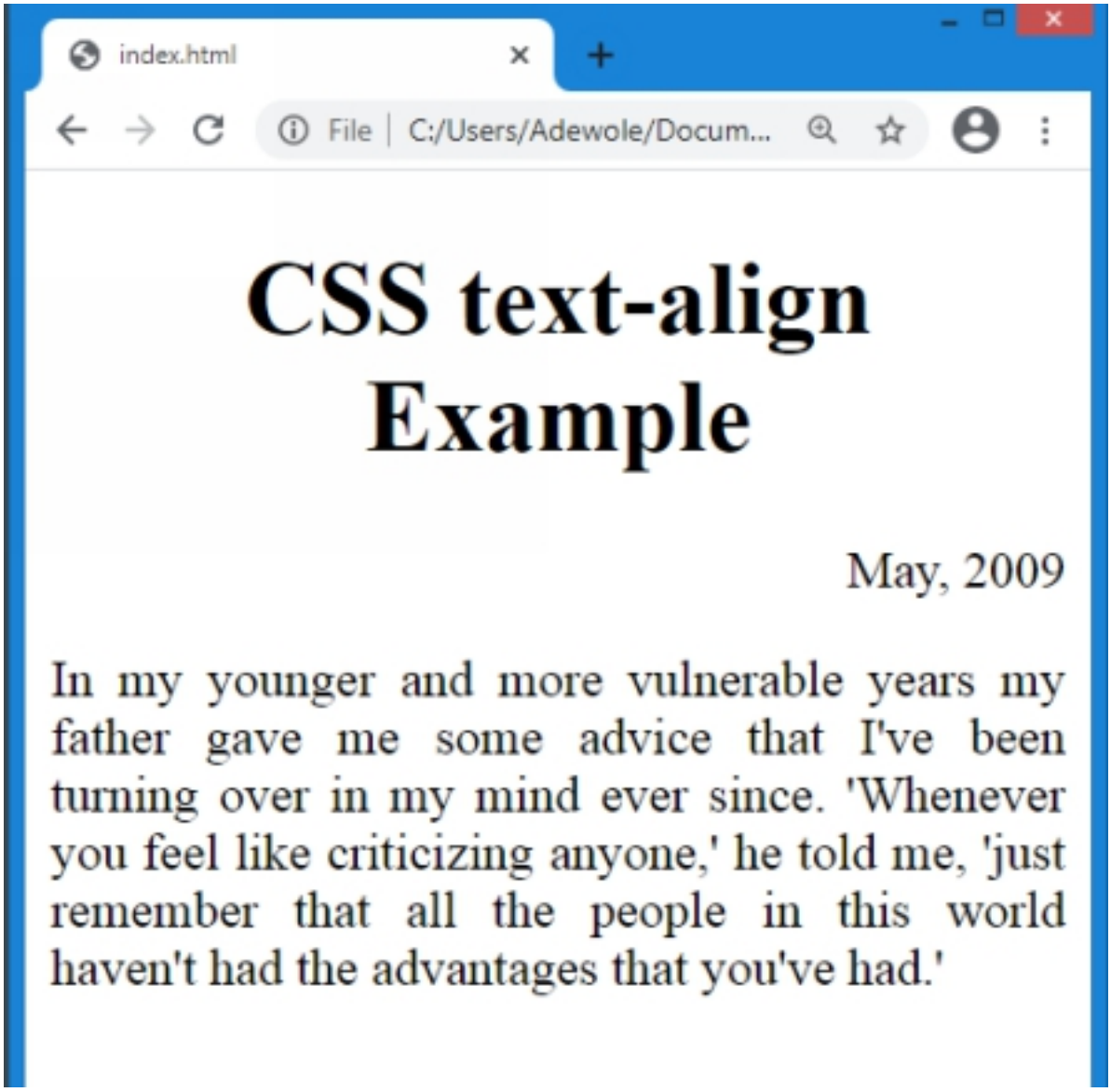
```
<!DOCTYPE html>
        <html>
<head>
 <style>
         body {
         color: red;
          }
   h1 {
      color: #00ff00;
   }

   p.ex {
      color: rgb(0,0,255);
   }
 </style>
</head>

<body>

  <h1>This is heading 1</h1>
  <p>This is an ordinary paragraph. Notice that this text
is red. The default text-color for a page is defined in the
body selector.</p>
  <p class="ex">This is a paragraph with class="ex". This
text is blue.</p>

</body>
</html>
```

Text Colour of Different Elements.


index.html
File | C:/Users/Adewole/Docum...
This is heading 1
This is an ordinary paragraph. Notice that this text is red. The default text-color for a page is defined in the body selector.
This is a paragraph with class="ex". This text is blue.

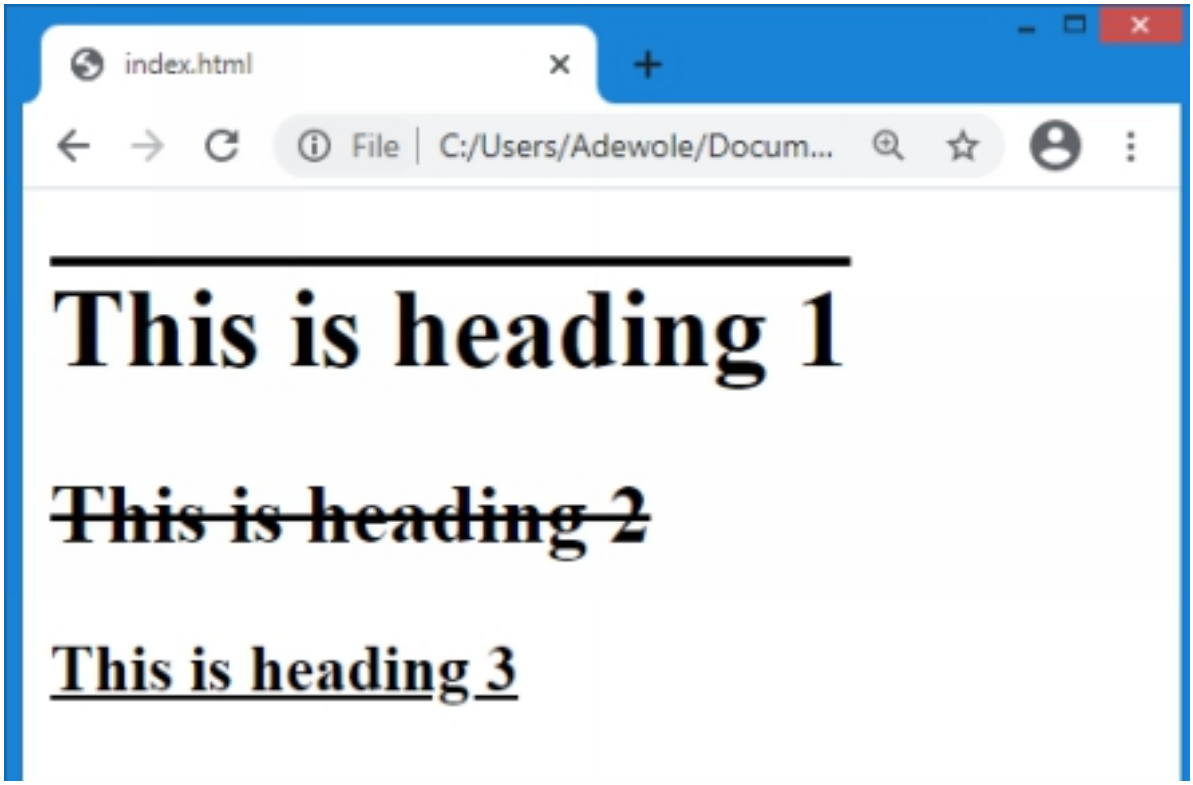
```
<!DOCTYPE html>
<html>

<head>
 <style>
  h1 {
     text-align: center;
  }

  p.date {
     text-align: right;
  }

  p.main {
     text-align: justify;
  }
 </style>
</head>

<body>

 <h1>CSS text-align Example</h1>
 <p class="date">May, 2009</p>
 <p class="main">In my younger and more vulnerable
years my father gave me some advice that I've been
turning over in my mind ever since. 'Whenever you
feel like criticizing anyone,' he told me, 'just remember
that all the people in this world haven't had the
advantages
   that you've had.'</p>

</body>
</html>
```

Align the Text


index.html
File | C:/Users/Adewole/Docum...
CSS text-align Example
May, 2009
In my younger and more vulnerable years my father gave me some advice that I've been turning over in my mind ever since. 'Whenever you feel like criticizing anyone,' he told me, 'just remember that all the people in this world haven't had the advantages that you've had.'

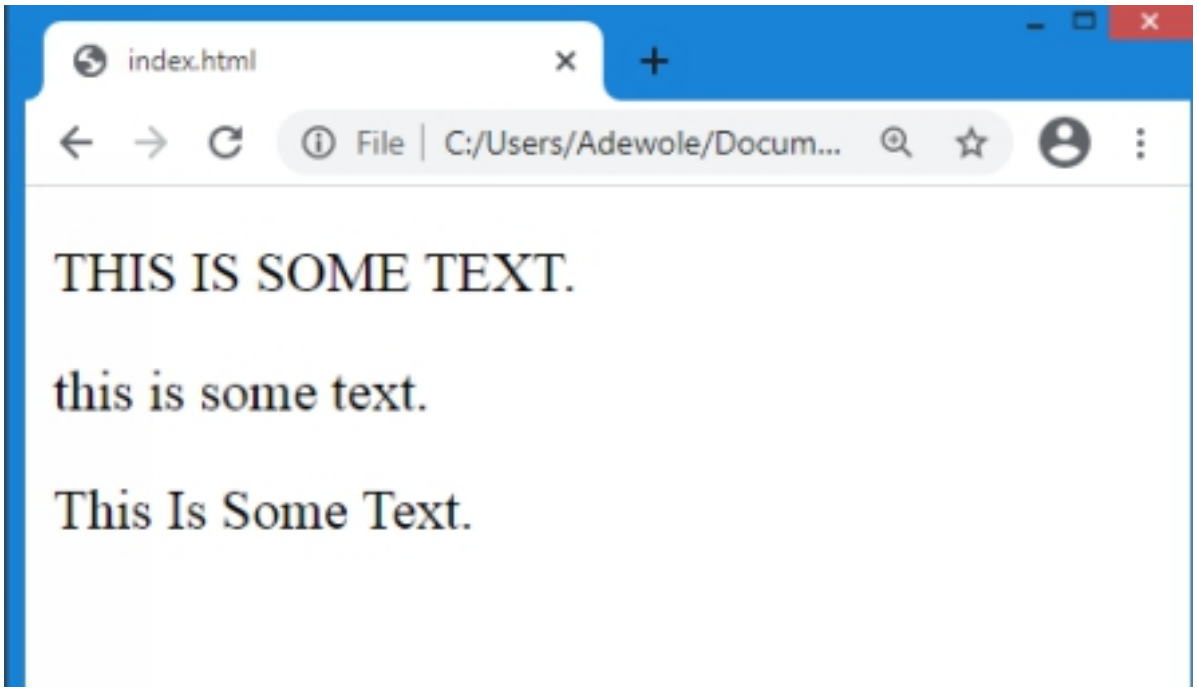
```
<!DOCTYPE html>
<html>

<head>
 <style>
  h1 {
     text-decoration: overline;
  }

  h2 {
     text-decoration: line-through;
  }

  h3 {
     text-decoration: underline;
  }
 </style>
</head>

<body>

 <h1>This is heading 1</h1>
 <h2>This is heading 2</h2>
 <h3>This is heading 3</h3>

</body>

</html>
```

Decorate the Text


index.html
File | C:/Users/Adewole/Docum...
This is heading 1
This is heading 2
This is heading 3

```
<!DOCTYPE html>
<html>

<head>
 <style>
   p.uppercase {
      text-transform: uppercase;
   }

   p.lowercase {
      text-transform: lowercase;
   }

   p.capitalize {
      text-transform: capitalize;
   }
 </style>
</head>

<body>

 <p class="uppercase">This is some text.</p>
 <p class="lowercase">This is some text.</p>
 <p class="capitalize">This is some text.</p>

</body>

</html>
```

Control the Letters in a Text


index.html
File | C:/Users/Adewole/Docum...
THIS IS SOME TEXT.
this is some text.
This Is Some Text.

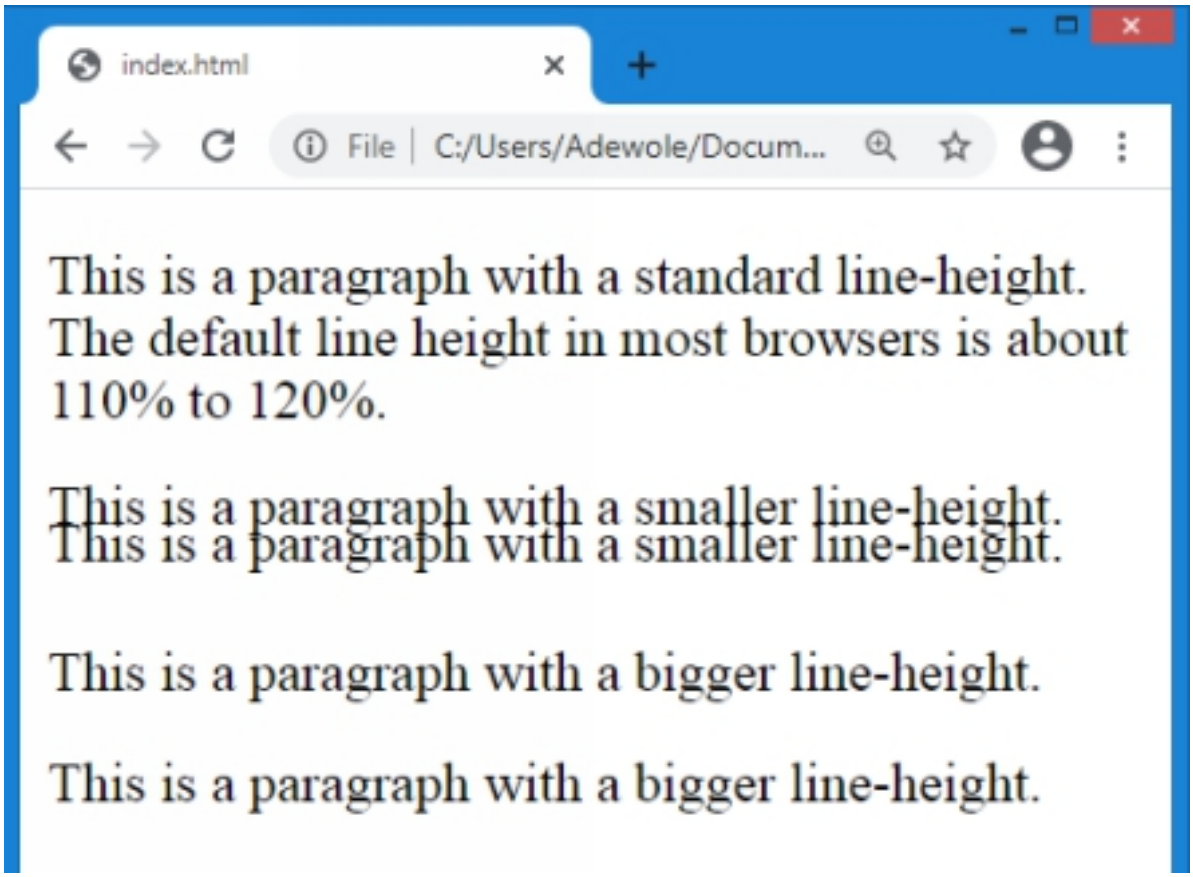
Specify the Space between Characters

```
<!DOCTYPE html>
<html>

<head>
 <style>
  h1 {
     letter-spacing: 3px;
  }

  h2 {
     letter-spacing: -3px;
  }
 </style>
</head>

<body>

 <h1>This is heading 1</h1>
 <h2>This is heading 2</h2>

</body>

</html>
```


index.html
File | C:/Users/Adewole/Docum...
This is heading 1
This is heading 2

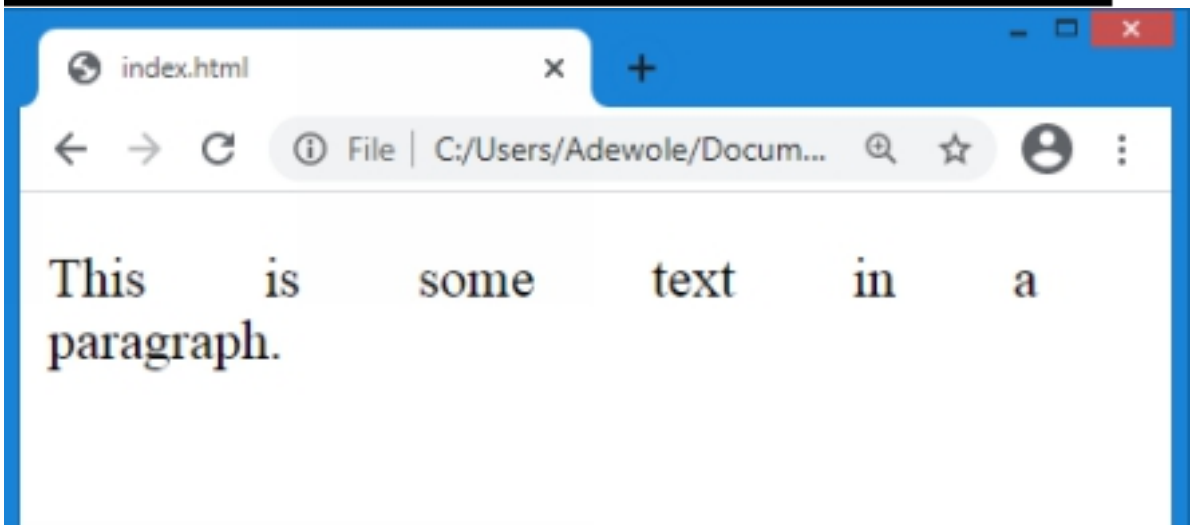
```
<!DOCTYPE html>
<html>

<head>
 <style>
  p.small {
    line-height: 70%;
  }

  p.big {
    line-height: 200%;
  }
 </style>
</head>

<body>

 <p>
  This is a paragraph with a standard line-height.<br>
The default line height in most browsers is about 110%
to 120%.<br>
 </p>
 <p class="small">
  This is a paragraph with a smaller line-height.<br>
This is a paragraph with a smaller line-height.<br>
 </p>

 <p class="big">
  This is a paragraph with a bigger line-height.<br>
This is a paragraph with a bigger line-height.<br>
 </p>
</body>
</html>
```

Specify the Space between Lines


index.html
File | C:/Users/Adewole/Docum...
This is a paragraph with a standard line-height. The default line height in most browsers is about 110% to 120%.
This is a paragraph with a smaller line-height.
This is a paragraph with a smaller line-height.
This is a paragraph with a bigger line-height.
This is a paragraph with a bigger line-height.

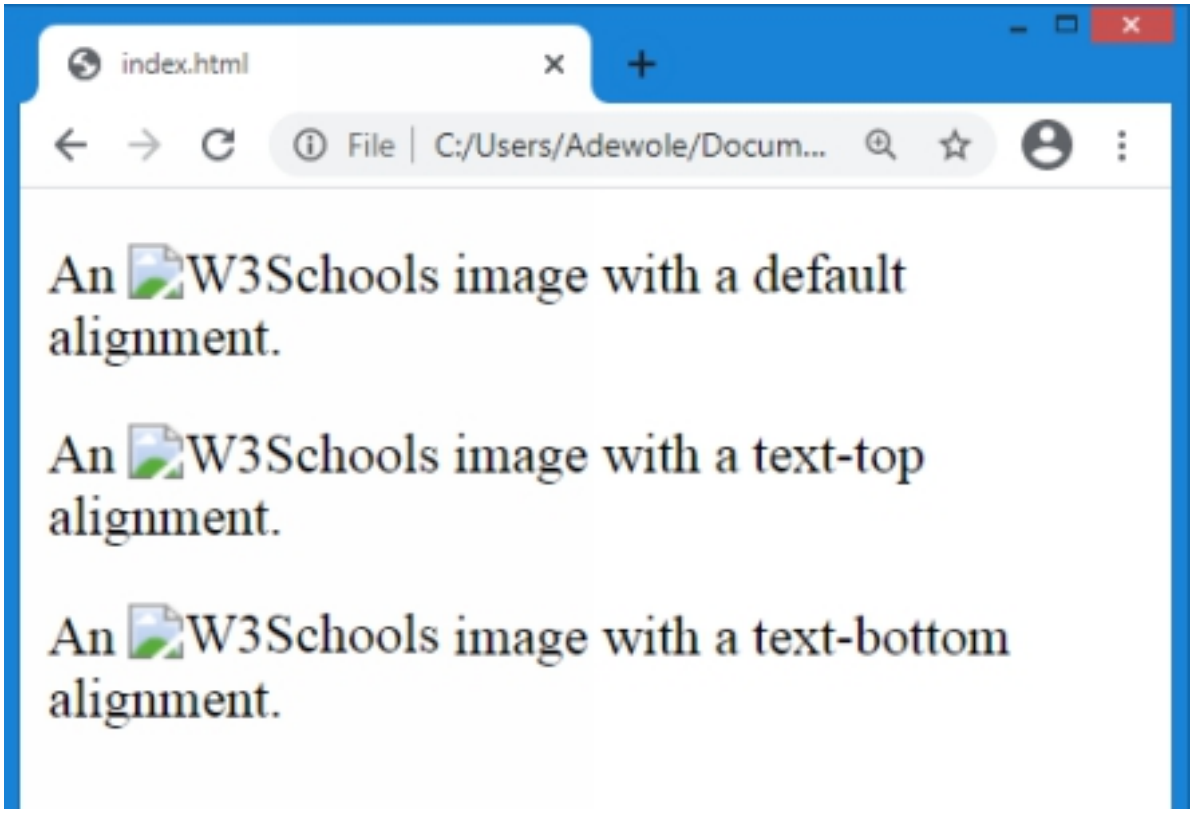
Increase White space between Words

```
<!DOCTYPE html>
<html>

<head>
 <style>
  p {
     word-spacing: 30px;
  }
 </style>
</head>

<body>

 <p>This is some text in a paragraph.</p>

</body>

</html>
```



Vertical Alignment of an Image inside Text

```
<!DOCTYPE html>
<html>

<head>
 <style>
  img.top {
      vertical-align: text-top;
  }

  img.bottom {
      vertical-align: text-bottom;
  }
 </style>
</head>

<body>

 <p>An <img src="http://apk-dl.com/detail/image/com.freeit.java-w250.png" alt="W3Schools" width="150" height="100"> image with a default alignment.</p>
 <p>An <img class="top" src="http://apk-dl.com/detail/image/com.freeit.java-w250.png" alt="W3Schools" width="150" height="100"> image with a text-top alignment.</p>
 <p>An <img class="bottom" src="http://apk-dl.com/detail/image/com.freeit.java-w250.png" alt="W3Schools" width="150" height="100"> image with a text-bottom alignment.</p>

</body>

</html>
```


index.html
File | C:/Users/Adewole/Docum...
An W3Schools image with a default alignment.
An W3Schools image with a text-top alignment.
An W3Schools image with a text-bottom alignment.

AN ELEMENT WITH A TOTAL WIDTH OF 250PX

```
<!DOCTYPE html>
<html>

<head>
 <style>
  div {
      width: 250px;
      padding: 10px;
      border: 5px solid gray;
      margin: 0;
  }
 </style>
</head>

<body>

 <img src="image.jpg" width="250" height="263"
alt="Klematis">
 <div>The picture above is 250px wide. The total width
of this element is also 250px.</ div>

</body>

</html>
```


index.html
File | C:/Users/Adewole/Docum...
The picture above is 250px wide. The total width of this element is also 250px.

Specify a black border for table, th, and td elements

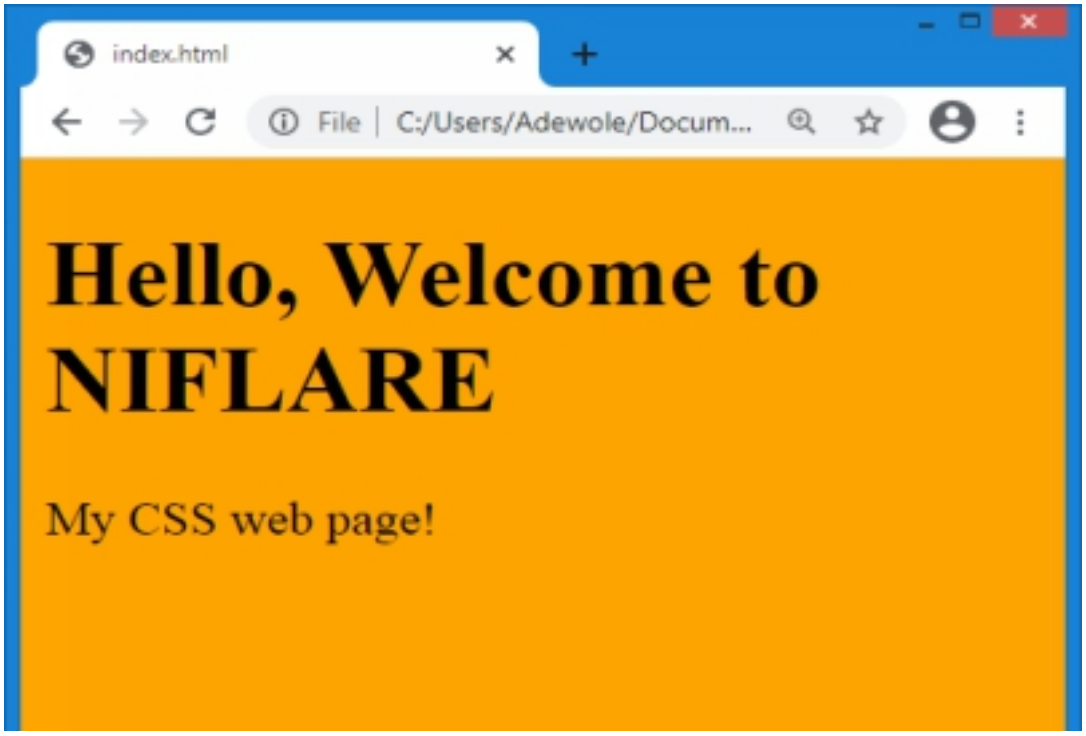
```
<!DOCTYPE html>
<html>
        <head>
 <style>
  table, th, td {
     border: 1px solid black;
  }
 </style>
        </head>

<body>
         <table>
   <tr>
    <th>Firstname</th>
    <th>Lastname</th>
   </tr>
   <tr>
    <td>Jade</td>
    <td>Gabi</td>
   </tr>
   <tr>
    <td>Lola</td>
    <td>Graffield</td>
   </tr>
 </table>

</body>
</html>
```

Firstname	Lastname
Jade	Gabi
Lola	Graffield

PLAYGROUND (BACKGROUND)

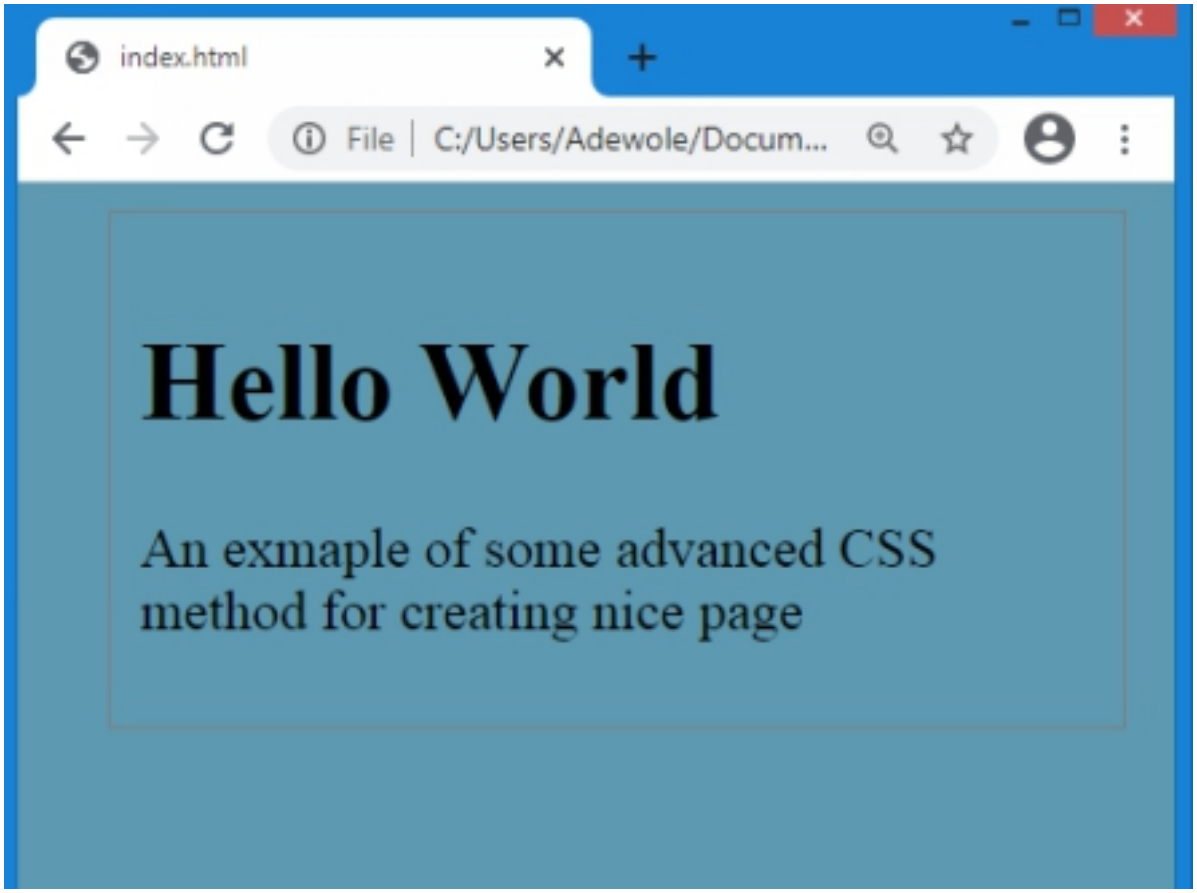
Background Color of a Page

```
<!DOCTYPE html>
<html>

<head>

 <style>
   body{
      background-color: orange
      }
  </style>
 </head>

  <body>
     <h1>Hello, Welcome to NIFLARE</h1>
     <p>My CSS web page!</p>
  </body>
  </html>
```


index.html
File | C:/Users/Adewole/Docum...
Hello, Welcome to NIFLARE
My CSS web page!

```
<!DOCTYPE html>
<html>

<head>
 <style>
    body {
       margin-left:20px;
       background:#5d9ab2
    }
    .container{text-align:center;}

    .center_div{
       border:1px solid gray;
       margin-left: auto;
       margin-right: auto;
       width: 90%;
       background-color: #d0f0f6;
       text-align: left;
       padding: 8px;
    }
   </style>
  </head>
   <body>
    <div class="container">
    <div class="center_div">
      <h1>Hello World</h1>
      <p>An exmaple of some advanced CSS method for
creating nice page</p>
       </div>
      </div>

   </body>
   </html>
```

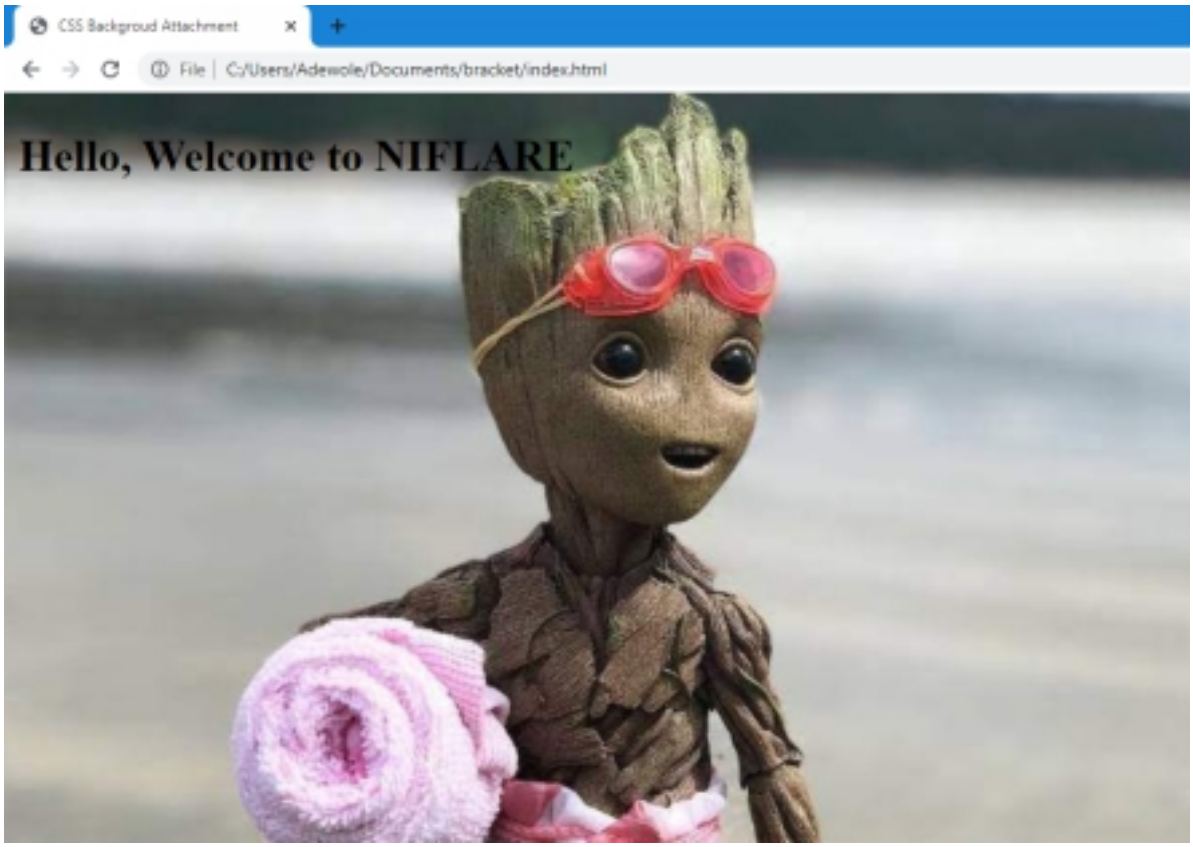
Backgrounds


index.html
File | C:/Users/Adewole/Docum...
Hello World
An exmaple of some advanced CSS method for creating nice page

Background Attribute

Background attachment determines whether a background images is fixed or scrolls with the rest of the page.

```
<!DOCTYPE html>
<html>

<head>
  <title>CSS Backgroud Attachment</title>

 <style>
   body{
      background-image:url(image.jpg);
      background-attachment: scroll;
      }
  </style>
 </head>

  <body>
     <h2>Hello, Welcome to NIFLARE</h2>
  </body>
  </html>
```


CSS Backgroud Attachment
File | C:/Users/Adewole/Documents/bracket/index.html
Hello, Welcome to NIFLARE

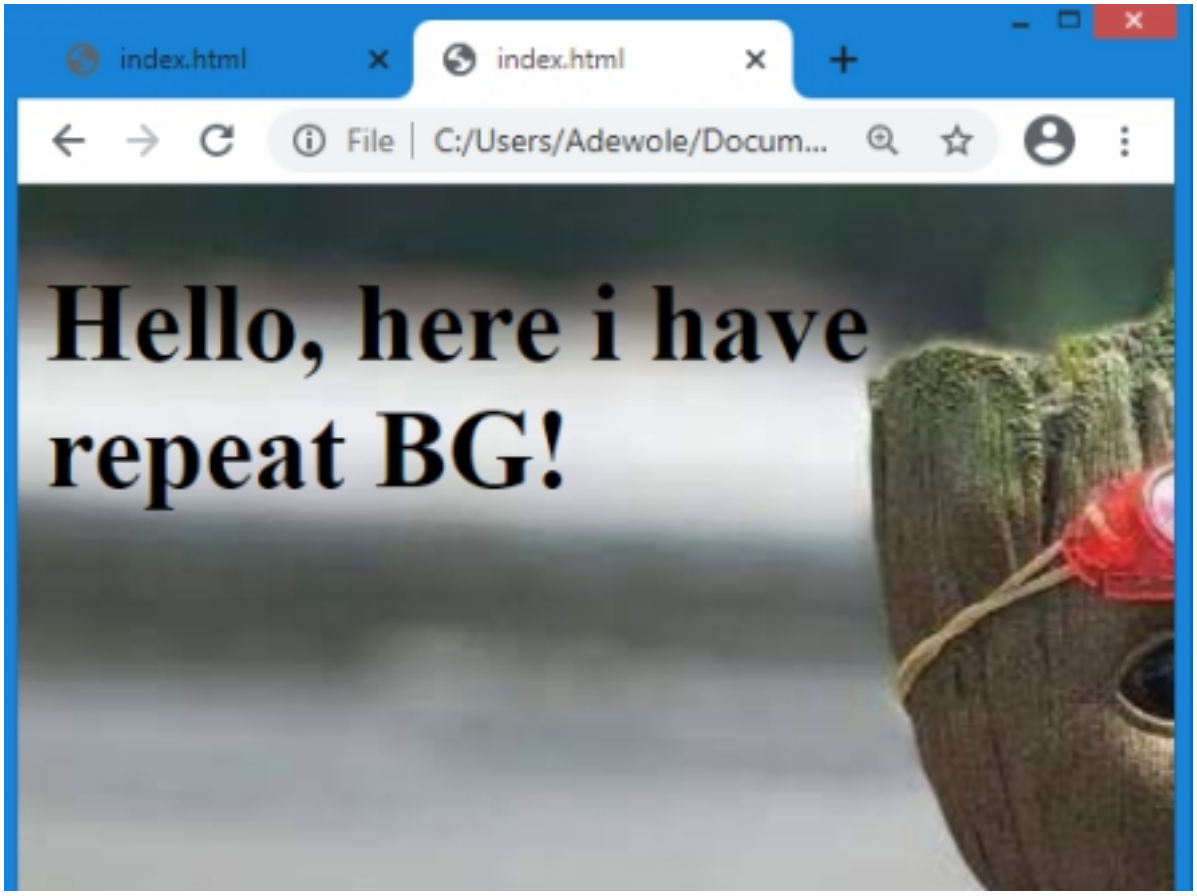
Background Color of different Elements

```
<!DOCTYPE html>
<html>

<head>

 <style>
    h1{background-color: red}
    p{background-color: green}
    div{background-color: yellow}

  </style>
 </head>

  <body>
     <h1>CSS background-color example</h1>
     <div>
        A text inside a div element.
     <p>A paragraph with its own background
color</p>
        All these are still inside the div element.
  </div>

  </body>
  </html>
```


index.html
File | C:/Users/Adewole/Docum...
CSS background-color example
A text inside a div element.
A paragraph with its own background color
All these are still inside the div element.

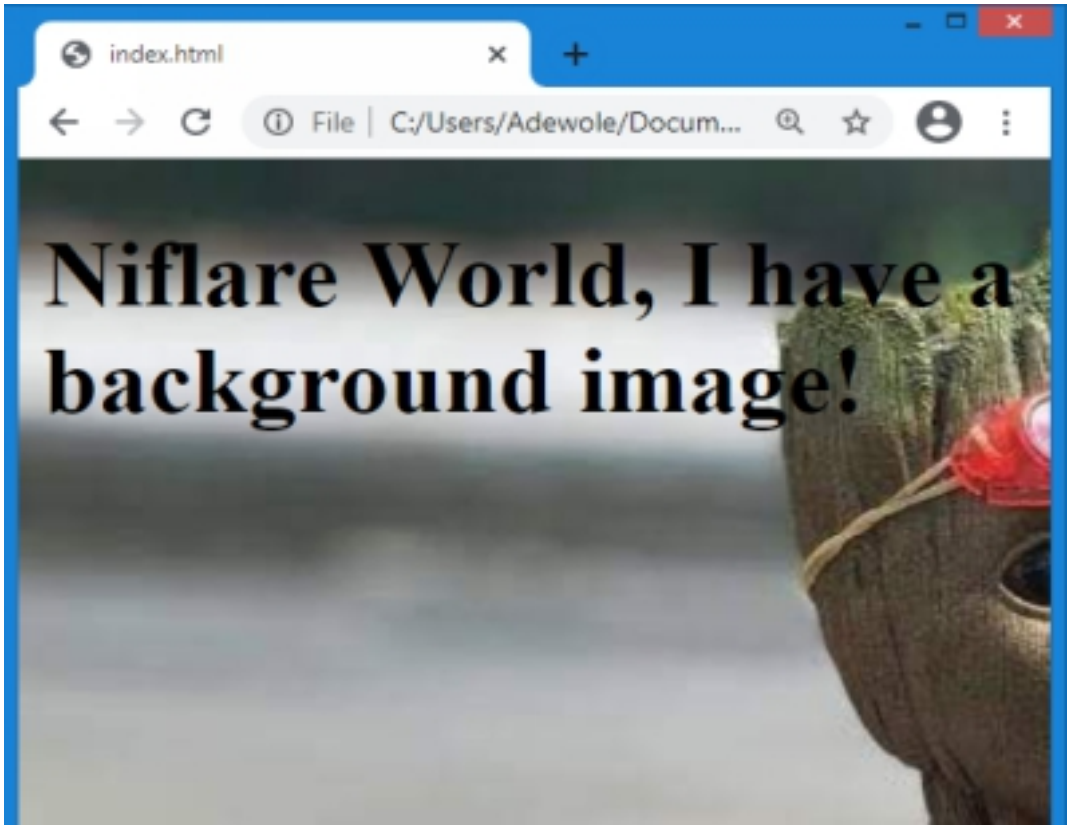
```
<!DOCTYPE html>
<html>

 <style>
    body{background-image: url(image.jpg);
    background-repeat: repeat-x;}

   </style>

   <body>
      <h1>Hello, here i have repeat BG!</h1>
   </body>
   </html>
```

Background Image only Horizontally


index.html
index.html
File | C:/Users/Adewole/Docum...
Hello, here i have repeat BG!

```
<!DOCTYPE html>
<html>

<head>
 <style>
    body{background-image: url(image.jpg);
    background-repeat: repeat-x;}

  </style>
 </head>
  <body>
     <h1>Niflare World, I have a background
image!</h1>
  </body>
  </html>
```

Background Image



Position a Background Image

```
<!DOCTYPE html>
<html>

<head>
 <style>
    body{background-image: url(image.jpg);
    background-repeat:no-repeat;
    background-position: right top;
    margin-right: 700px;}

  </style>
 </head>
  <body>
     <h1>Niflare World</h1>
     <p>This is a background no-repeat, set position
example.</p>
     <p>Now the background image is only shown
once, and positioned away from the text</p>
     <p>In this example we have also added a margin
on the right side, so th background image will never
disturb the text.</p>
  </body>
  </html>
```

index.html

File | C:/Users/Adewole/Documents/bracket/index.html

Niflare World

This is a background no-repeat, set position example.

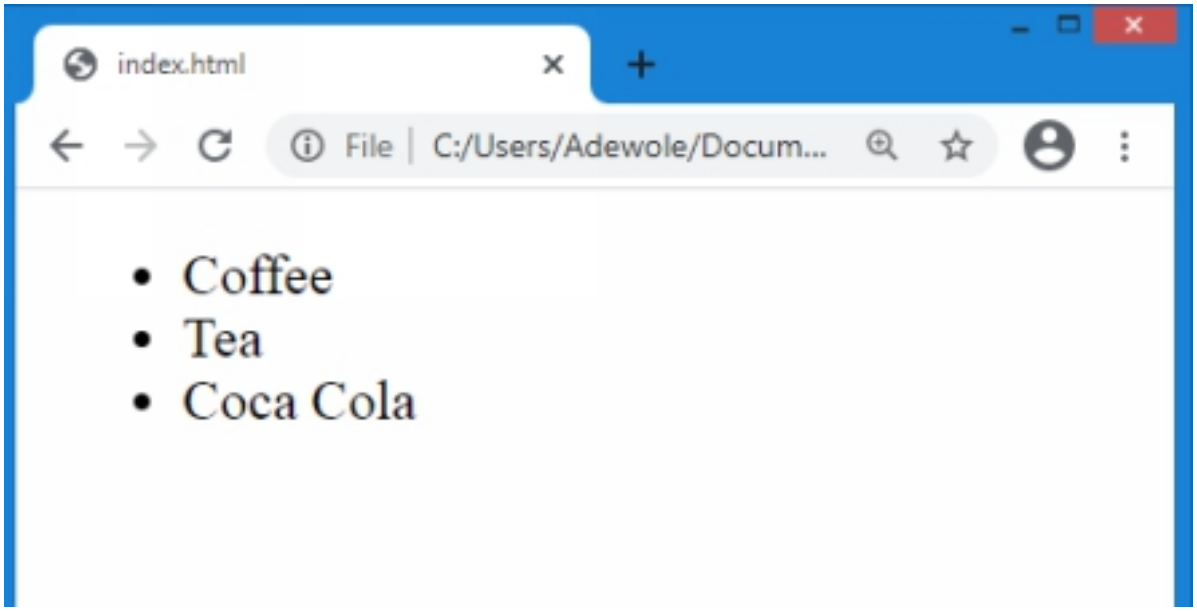
Now the background image is only shown once, and positioned away from the text

In this example we have also added a margin on the right side, so th background image will never disturb the text.

PLAYGROUND (LIST)

Set an image as the list-item marker

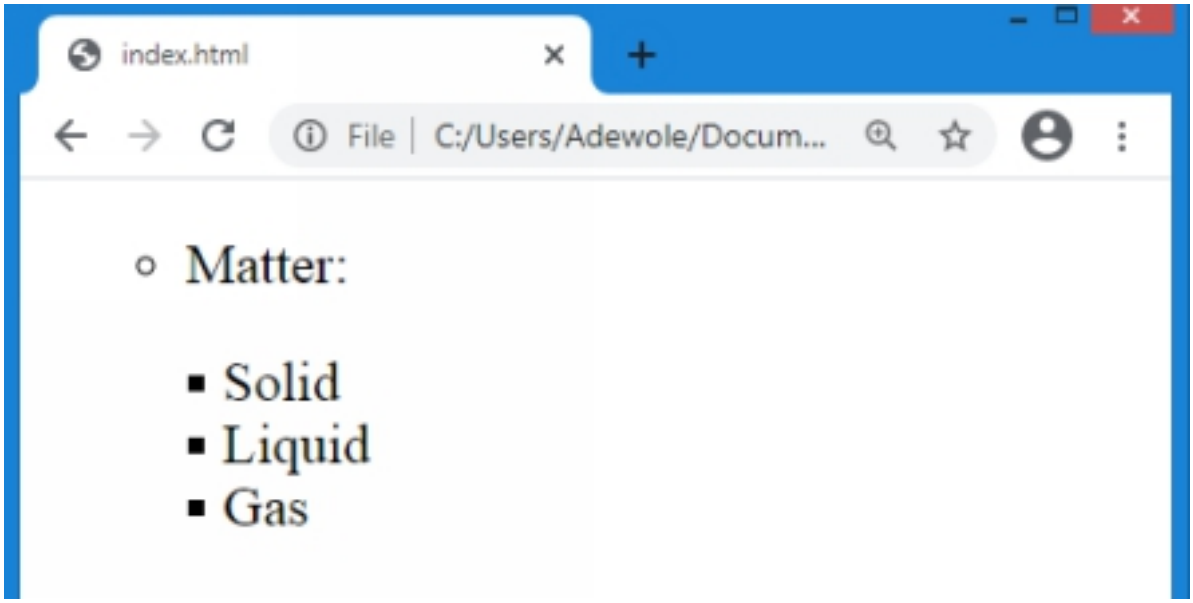
```
<!DOCTYPE html>
<html>

<head>
 <style>
  ul {
     list-style-image: url('http://www.niflare.com);
  }
 </style>
</head>

<body>

 <ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Coca Cola</li>
 </ul>

</body>

</html>
```


index.html
File | C:/Users/Adewole/Docum...
Coffee
Tea
Coca Cola

List Style

It specifies all the list properties in one declaration.

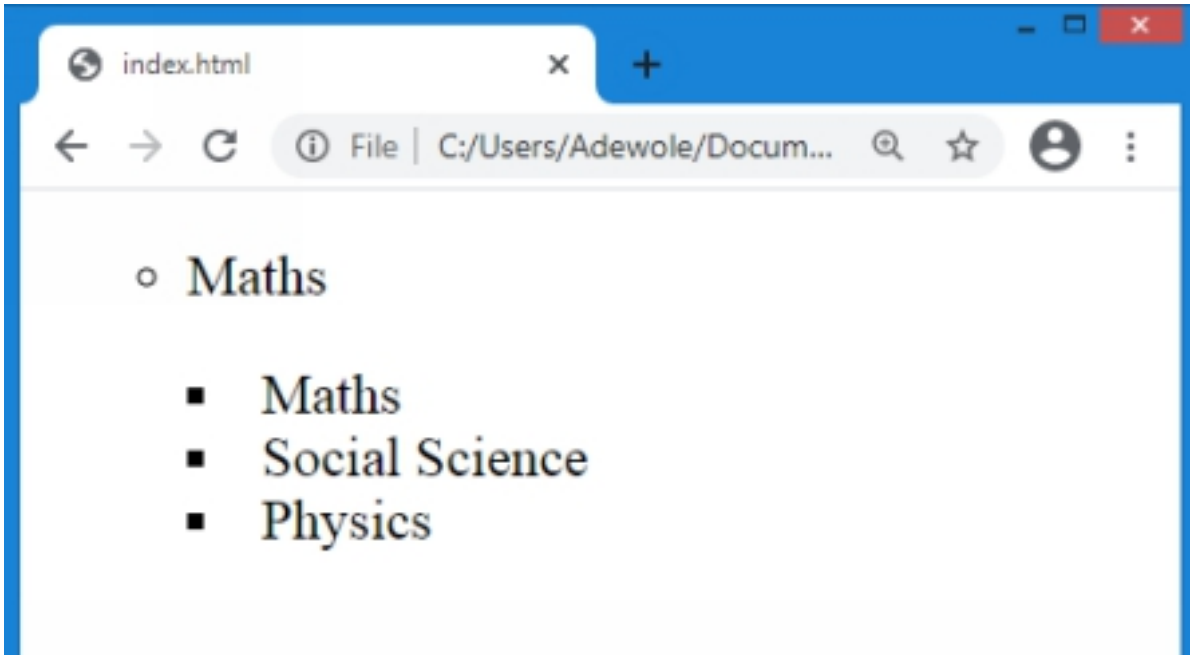
```
<!DOCTYPE html>
<html>

<head>
 <style>
  .main
        {
                list-style:circle outside url(bullet.png);
        }

        .sub
        {
                list-style:square inside url(bullet1.png);
        }
 </style>
</head>

<body>
 <ul class="main">

  <li>Matter:</li>

 </ul>

 <ul class="sub">
  <li>Solid</li>
  <li>Liquid</li>
  <li>Gas</li>
 </ul>
</body>

</html>
```


index.html
File | C:/Users/Adewole/Docum...
Matter:
Solid
Liquid
Gas

Link Style Position

The list-style-position property indicates whether the marker should appear inside or outside of the box containing the bullet points.

```
<!DOCTYPE html>
<html>

<head>
 <style>
  .main
        {
                list-style-type:circle;
                list-style-position:outside;
        }
        .sub
        {
                list-style-type:square;
                list-style-position:inside;
        }
 </style>
</head>
<body>
 <ul class="main">
  <li>Maths</li>
 </ul>
 <ul class="sub">
  <li>Maths</li>
  <li>Social Science</li>
  <li>Physics</li>
 </ul>
</body>

</html>
```

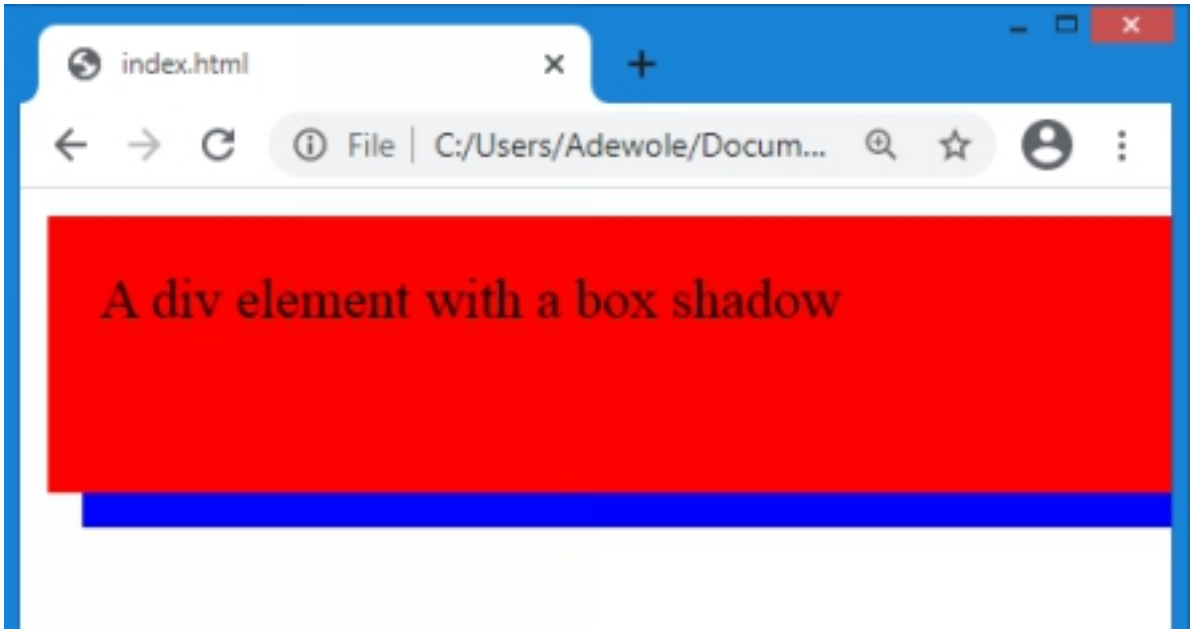

index.html
File | C:/Users/Adewole/Docum...
Maths
Maths
Social Science
Physics

PLAYGROUND (BOX MODEL)

Box Shadow

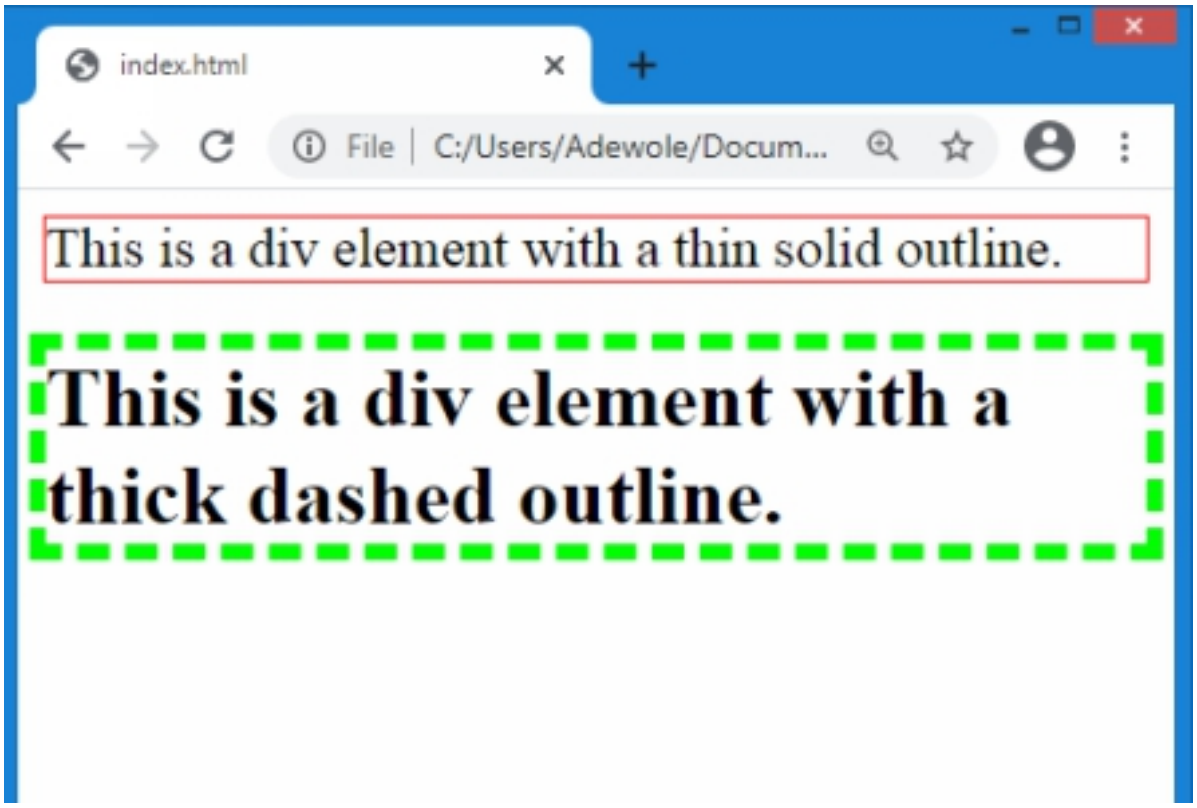
Box shadow property, CSS3 supported to add shadow to box.

```
<!DOCTYPE html>
<html>

<head>

 <style>
    div{width: 300px;
    height: 50px;
    padding: 15px;
    background-color: red;
    box-shadow: 10px 10px #0000FF}

  </style>
 </head>
  <body>
     <div>A div element with a box shadow</div>
  </body>
  </html>
```


index.html
File | C:/Users/Adewole/Docum...
A div element with a box shadow

```
<!DOCTYPE html>
<html>

<head>

 <style>
  div
  {
    outline:thin solid red;
  }

  h2
  {
    outline:thick dashed #00FF00;
  }
 </style>

</head>

<body>

 <div>This is a div element with a thin solid
outline.</div>

 <h2>This is a div element with a thick dashed
outline.</h2>

</body>

</html>
```

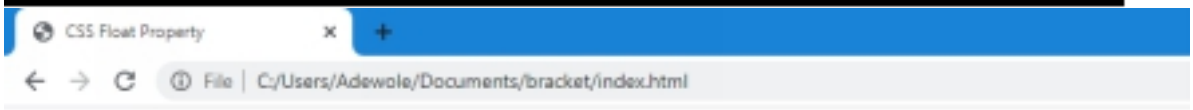
Outline


index.html
File | C:/Users/Adewole/Docum...
This is a div element with a thin solid outline.
This is a div element with a thick dashed outline.

Float

The float property causes an element to be moved to one side of the parent element's content area, which allows other content to flow around it.

```
<!DOCTYPE html>
<html>

<head>
  <title>CSS Float Property</title>
 <style>
   h1{float: left}
  </style>
 </head>
  <body>
    <h1>CSS float example</h1>
    <p>This text will be flowing around the
heading.</p>
  </body>
  </html>
```

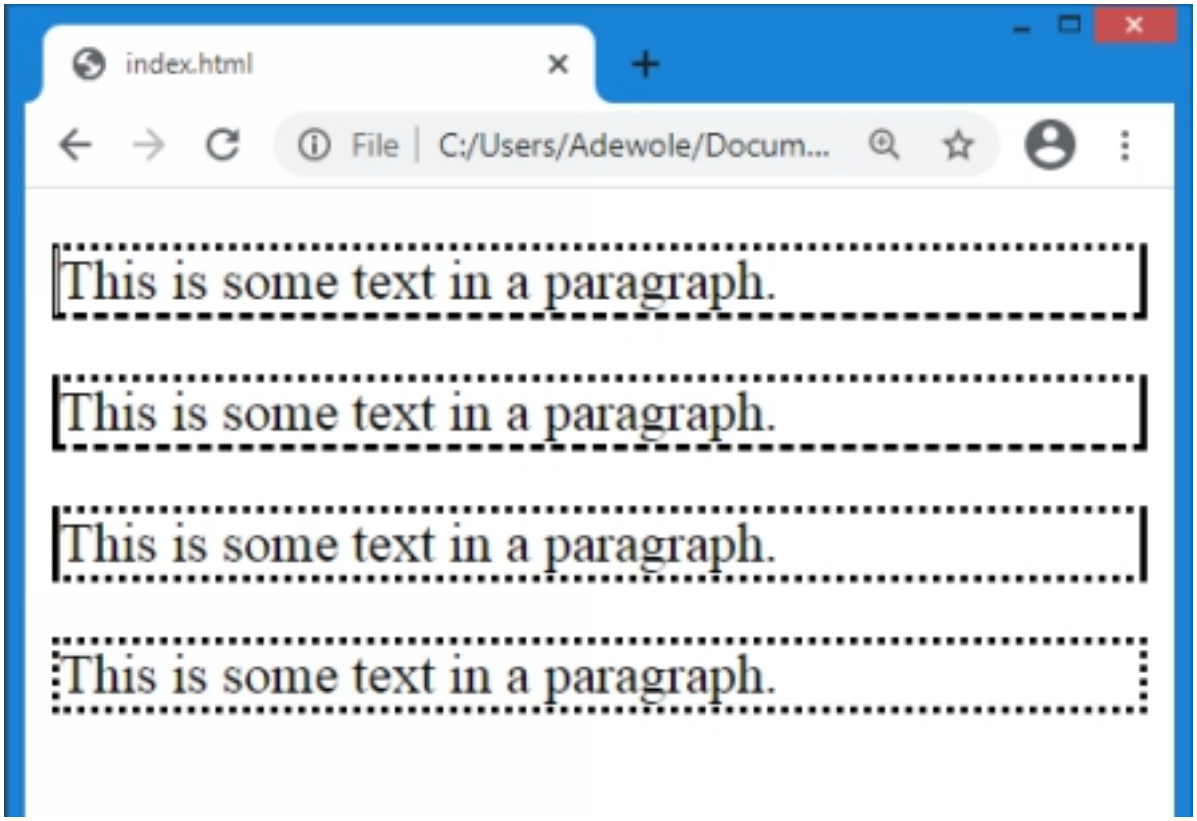


This text will be flowing around the heading.

CSS float example

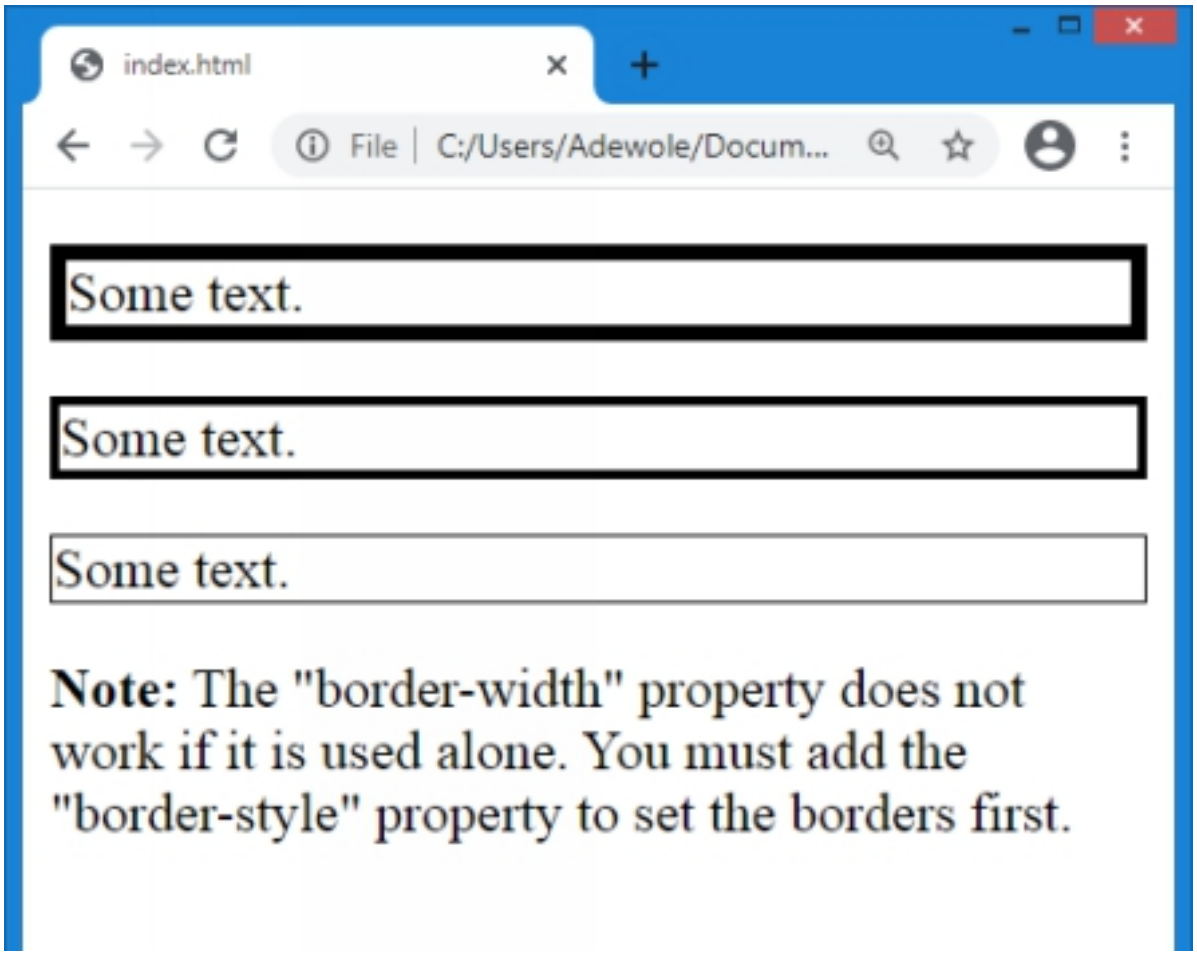
```
<!DOCTYPE html>
<html>
        <head>
 <style>
  p.one {
     border-style: dotted solid dashed double;
  }

  p.two {
     border-style: dotted solid dashed;
  }

  p.three {
     border-style: dotted solid;
  }

  p.four {
     border-style: dotted;
  }
 </style>
        </head>
<body>
 <p class="one">This is some text in a paragraph.</p>
 <p class="two">This is some text in a paragraph.</p>
 <p class="three">This is some text in a paragraph.</p>
 <p class="four">This is some text in a paragraph.</p>

</body>
</html>
```

Set different borders on each sides


index.html
File | C:/Users/Adewole/Docum...
This is some text in a paragraph.
This is some text in a paragraph.
This is some text in a paragraph.
This is some text in a paragraph.

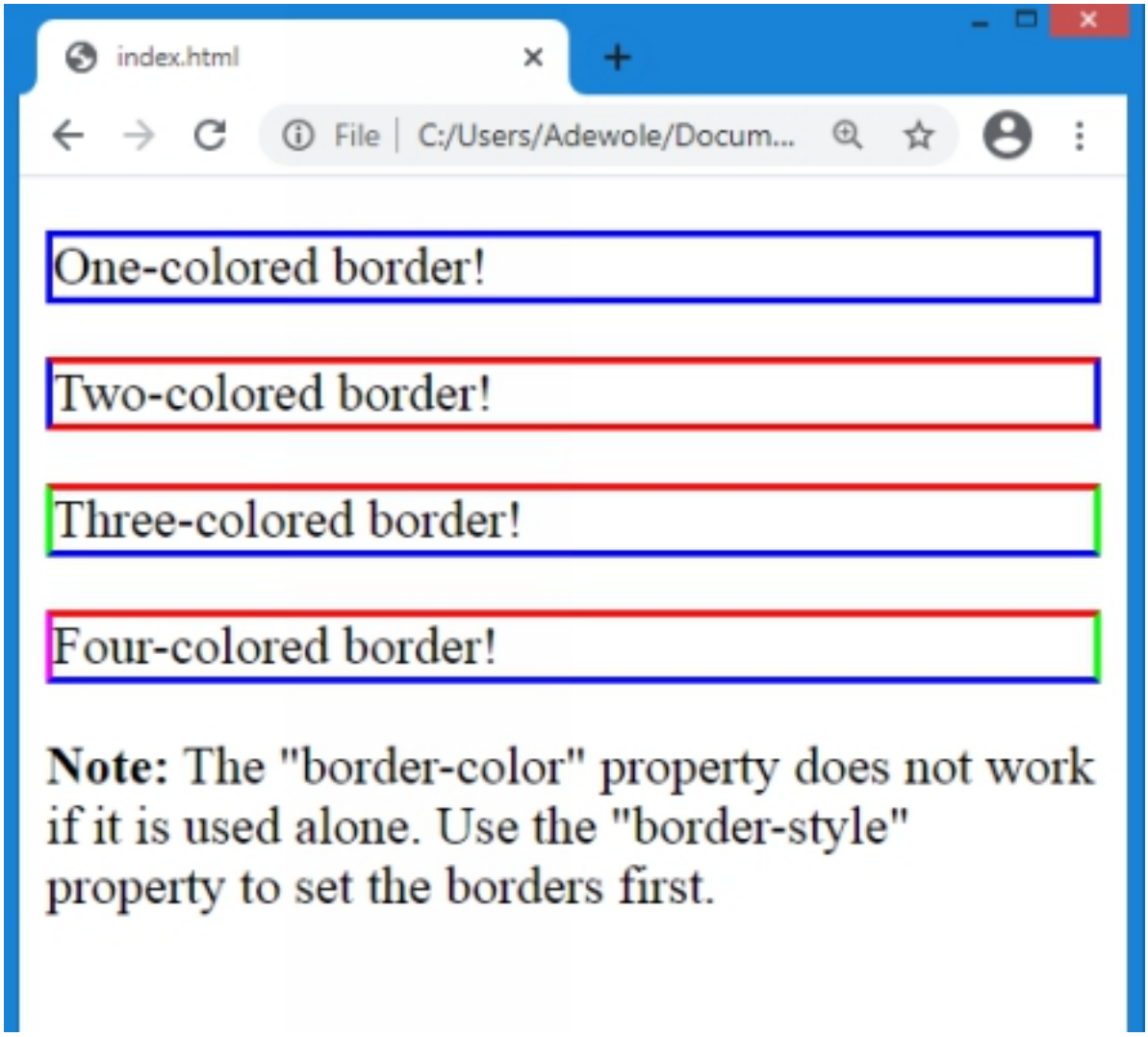
```
<!DOCTYPE html>
<html>
        <head>
 <style>
  p.one {
     border-style: solid;
     border-width: 5px;
  }

  p.two {
     border-style: solid;
     border-width: medium;
  }

  p.three {
     border-style: solid;
     border-width: 1px;
  }
 </style>
        </head>

<body>

 <p class="one">Some text.</p>
 <p class="two">Some text.</p>
 <p class="three">Some text.</p>
 <p><b>Note:</b>The "border-width" property does
not work if it is used alone. You must add the "border-
style" property to set the borders first.</p>

</body>
</html>
```

Set the width of the four borders


index.html
File | C:/Users/Adewole/Docum...
Some text.
Some text.
Some text.
Note: The "border-width" property does not work if it is used alone. You must add the "border-style" property to set the borders first.

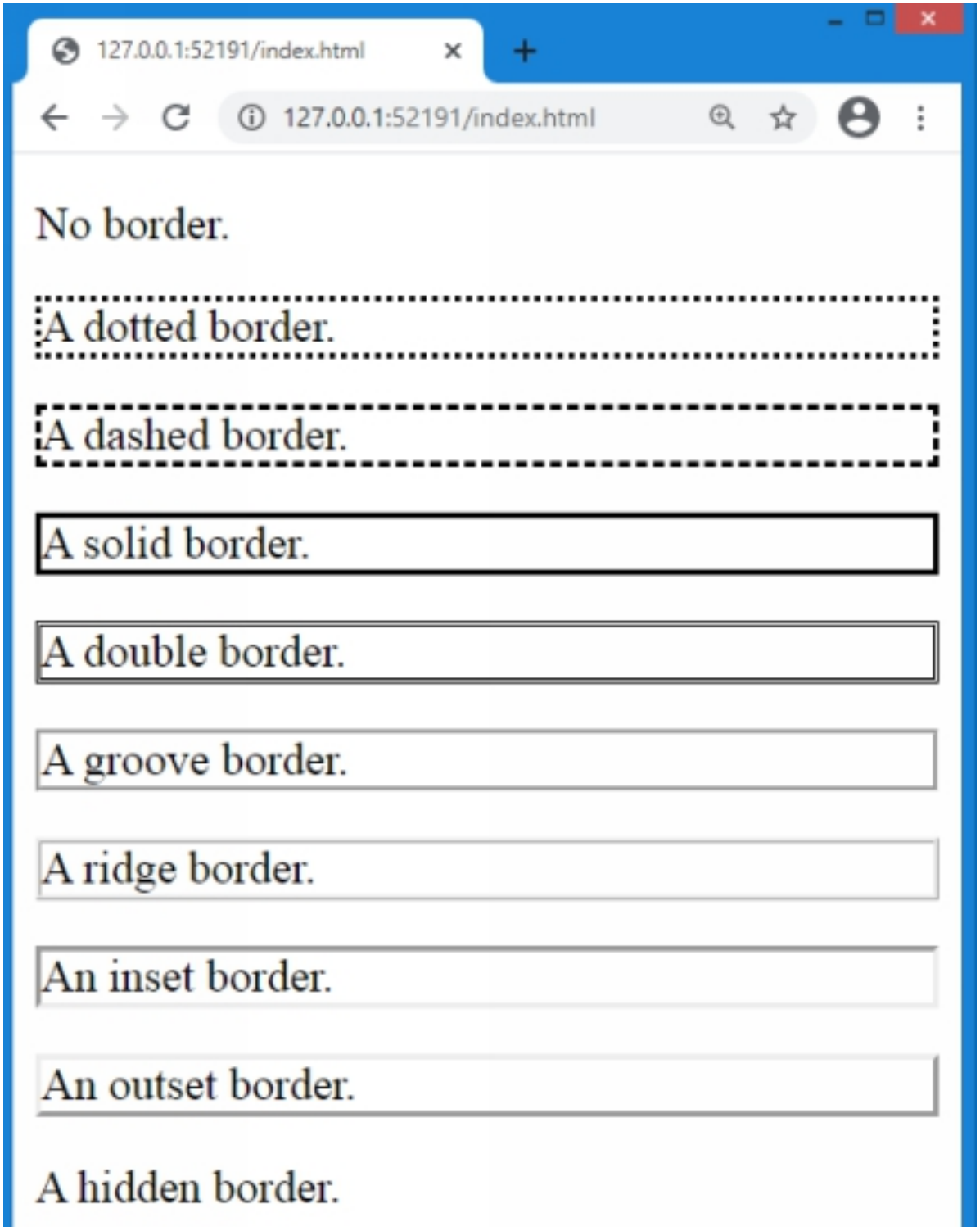
```
<!DOCTYPE html>
<html>
        <head>
 <style>
  p.one {
     border-style: solid;
     border-color: #0000ff;
  }
  p.two {
     border-style: solid;
     border-color: #ff0000 #0000ff;
  }

  p.three {
     border-style: solid;
     border-color: #ff0000 #00ff00 #0000ff;
  }
     p.four {
     border-style: solid;
     border-color: #ff0000 #00ff00 #0000ff rgb(250,0,255);
  }
 </style>
</head>
        <body>
 <p class="one">One-colored border!</p>
 <p class="two">Two-colored border!</p>
 <p class="three">Three-colored border!</p>
 <p class="four">Four-colored border!</p>
 <p><b>Note:</b>The "border-color" property does not work if it is used alone. Use the "border-style" property to set the borders first.</p>
        </body>
</html>
```

Set the color of the four borders


index.html
File | C:/Users/Adewole/Docum...
One-colored border!
Two-colored border!
Three-colored border!
Four-colored border!
Note: The "border-color" property does not work if it is used alone. Use the "border-style" property to set the borders first.

```
<!DOCTYPE html>
<html>
	<head>
 <style>
   p.none {border-style: none;}
   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.hidden {border-style: hidden;}
 </style>
	</head>

<body>
 <p class="none">No border.</p>
 <p class="dotted">A dotted border.</p>
 <p class="dashed">A dashed border.</p>
 <p class="solid">A solid border.</p>
 <p class="double">A double border.</p>
 <p class="groove">A groove border.</p>
 <p class="ridge">A ridge border.</p>
 <p class="inset">An inset border.</p>
 <p class="outset">An outset border.</p>
 <p class="hidden">A hidden border.</p>

</body>
</html>
```

Set the style of the four borders


127.0.0.1:52191/index.html
127.0.0.1:52191/index.html
No border.
A dotted border.
A dashed border.
A solid border.
A double border.
A groove border.
A ridge border.
An inset border.
An outset border.
A hidden border.