Menus, Navigation & Web Page Protection

Unit - VI

Status Bar

- The status property sets the text in the status bar at the bottom of the browser, or returns the previously set text.

  Syntax: *window.status = “some text”;*

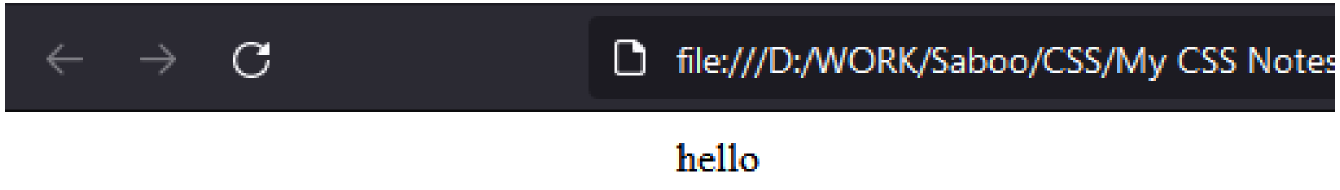
- The status property does not work in the default configuration of IE, Firefox, Chrome, Safari or Opera 15 and newer.

Moving the Message along Status Bar

```
<html>
<head> <title>Scrolling Text</title>
<script>
var scrollPos = 0,maxScroll = 100,blanks = “”;
function scrollText(text, milliseconds)
{ window.setInterval("displayText('"+text+"')", milliseconds);}
function displayText(text)
{ window.defaultStatus = blanks + text ;
++scrollPos;
blanks += " “;
if(scrollPos > maxScroll)
{ scrollPos = 0 blanks = "" } }
</script> </head>
<body onload="scrollText('Watch this text scroll!!!', 300)">
<p>Watch the text scroll at the bottom of this window!</p>
</body> </html>
```

Moving the Message along Status Bar

```
<html>
  <head><title>Moving Message
  </title><script>
  var scrollpos = 0, maxScroll = 100, blanks = "";
  function scrolltext(text,sec)
  {   window.setInterval(function()
  {   document.getElementById("show").innerHTML = blanks + text;
    ++scrollpos;
      blanks = blanks + "&nbsp";
      if (scrollpos>maxScroll)
      {   scrollpos = 0;
          blanks = "";
      }
  },sec);
  }</script></head>
  <body onload="scrolltext('hello',500)">
     <p id="show"></p>
  </body></html>
```



file:///D:/WORK/Saboo/CSS/My CSS Notes

hello

Banners

- Displaying banners ads is a common practice for showing advertisements on web pages to the visitors.

- Banner ads can be static or animated. Animated images are animated GIF files or flash movies. Flash movies are created using Macromedia Flash and the browsers must have installed flash plugin to view the movies.

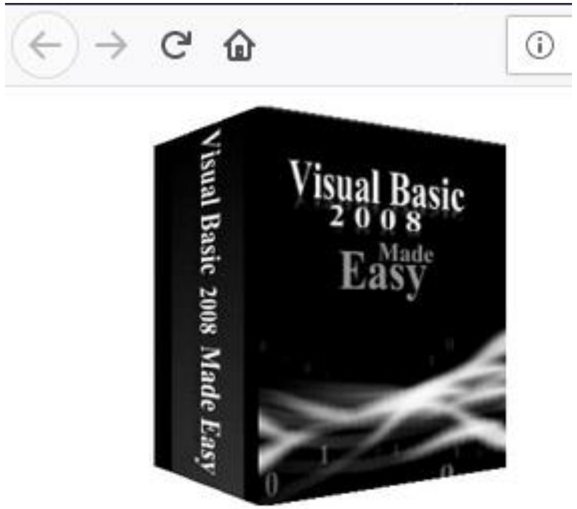
- On the other hand, you can create some animated effect using JavaScript, like rotating static banner ads at a certain time interval.

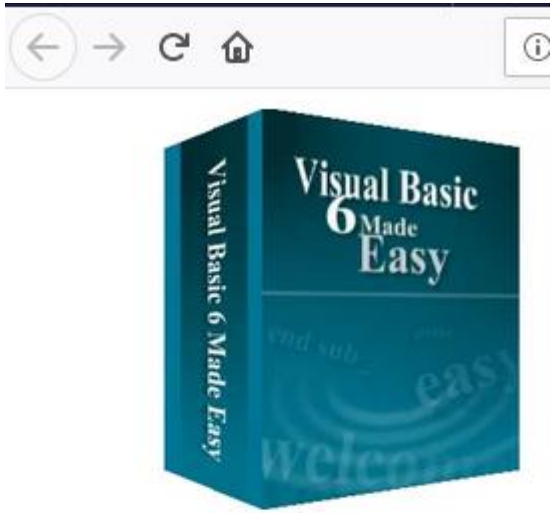
Rotating Banners

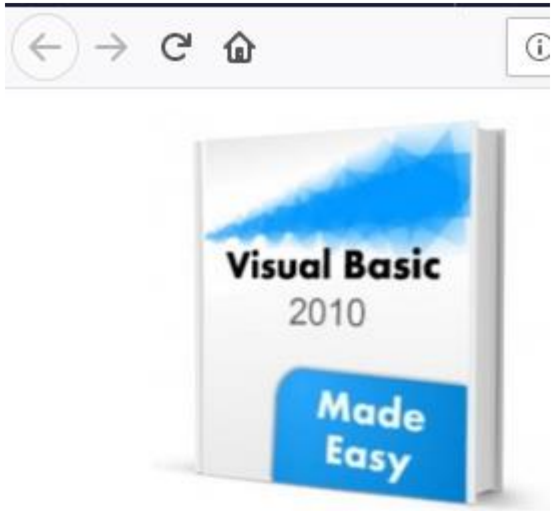
- Rotating banners ads comprises several banner images that constantly rotate on a webpage at a fix time interval.
- A rotating banner can be implemented by displaying multiple images on a certain time interval again and again with the help of img tag.

Rotating Banners

```
<html>
<head>
<script language="Javascript">
MyBanners=new Array('banner1.jpg','banner2.jpg','banner3.jpg')
banner=0
function ShowBanners()
{ if (document.images)
{ banner++
if (banner==MyBanners.length) {
banner=0}
document.ChangeBanner.src=MyBanners[banner]
setTimeout("ShowBanners()",1000)}}
</script>
<body onload="ShowBanners()">
<center>
<img src="banner1.jpg" width="200" height="200" name="ChangeBanner"/>
</center>
</body></html>
```



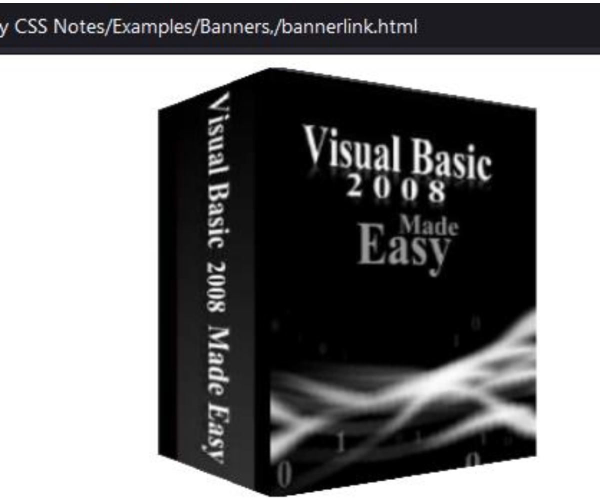


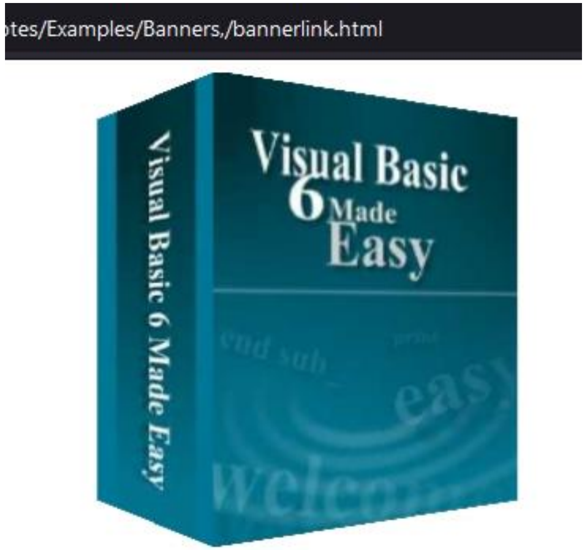


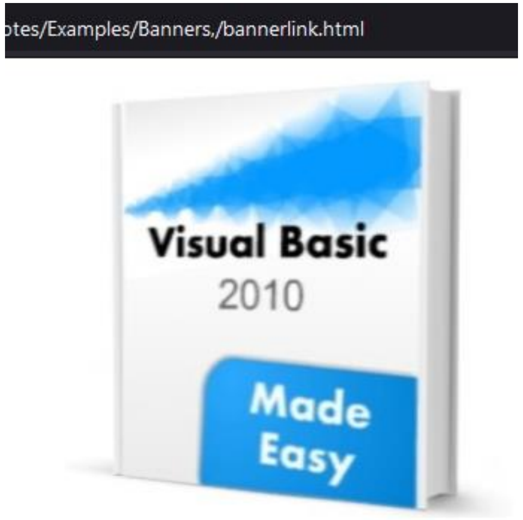
Rotating Banners as Hyperlink

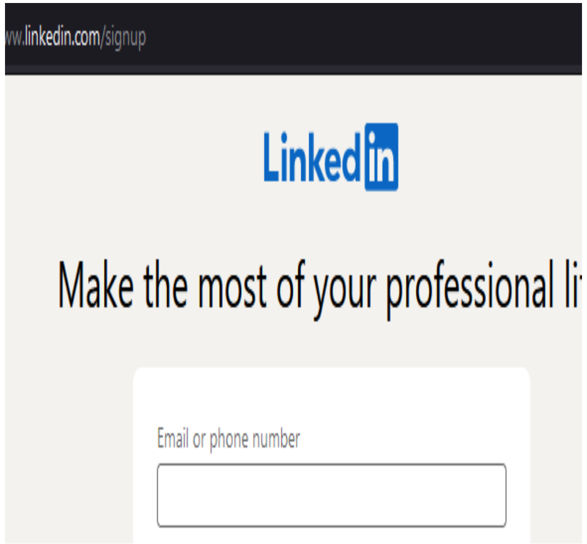
```
<html>
<head><title>Banners with hyperlink</title>
<script language="Javascript">
MyBanners=new Array('banner1.jpg','banner2.jpg','banner3.jpg')
mysite=new Array('https://www.google.com/','https://www.linkedin.com/signup','https://github.com/')
banner=0
function ShowBanners()
{ if (document.images)
{ banner++
if (banner==MyBanners.length) {
banner=0}
document.ChangeBanner.src=MyBanners[banner];
document.getElementById("an").href=mysite[banner];
setTimeout("ShowBanners()",3000)}}
</script>
<body onload="ShowBanners()">
    <a href="https://www.google.com/" target=_blank name=anchor id=an>
      <center><img src="banner1.jpg" width="300" height="300" name="ChangeBanner"/></center></a>
<p>Banner as Hyperlink</p></body></html>
```

Rotating Banners as Hyperlink


y CSS Notes/Examples/Banners,/bannerlink.html


otes/Examples/Banners,/bannerlink.html

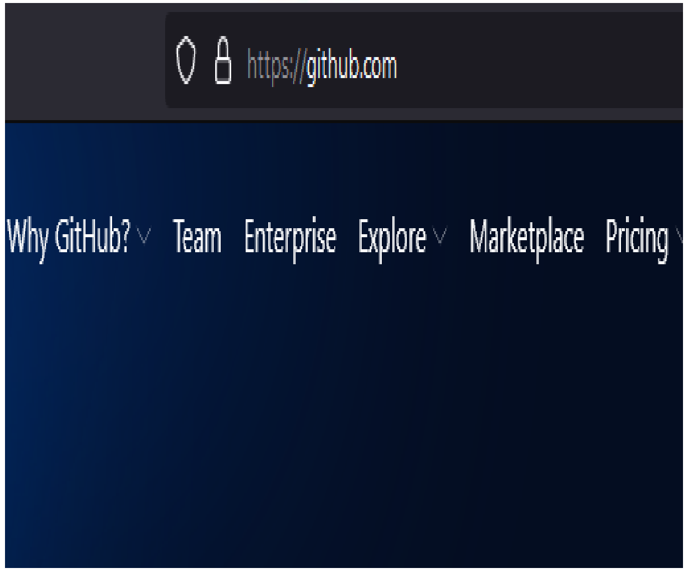

otes/Examples/Banners,/bannerlink.html

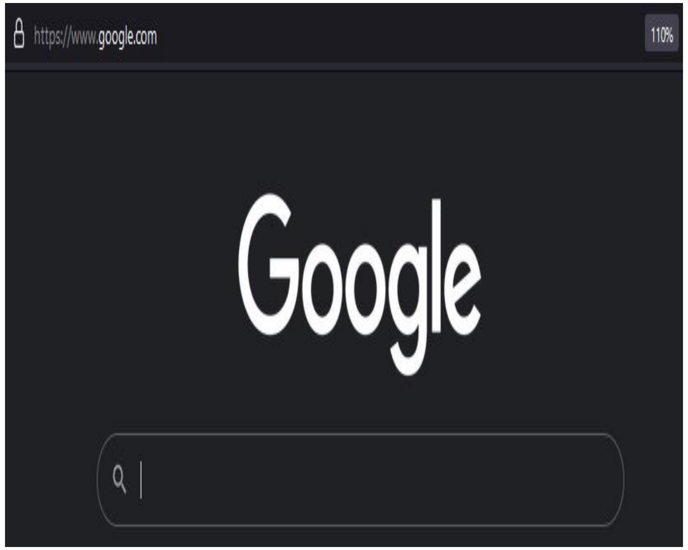

ww.linkedin.com/signup

LinkedIn

Make the most of your professional li

Email or phone number


https://github.com


https://www.google.com

JavaScript Framework

- When you set out to code a website, you could code every aspect of that site from scratch, but there are certain common website features that make more sense to apply from a template. And that’s where JavaScript frameworks come into play.

- JavaScript framework is an application framework written in JavaScript where the programmers can manipulate the functions and use them for their convenience.

Popular JavaScript Framework

- Angular:

  One of the most powerful, efficient, and open-source JavaScript frameworks is Angular. This framework is operated by Google and is basically implemented to use for developing Single Page Application (SPA).

- React:

  Created by Facebook, React framework has earned popularity within a short period of time. In fact, it is used to develop and operate the dynamic User Interface of the web pages with high incoming traffic.

- Ember:

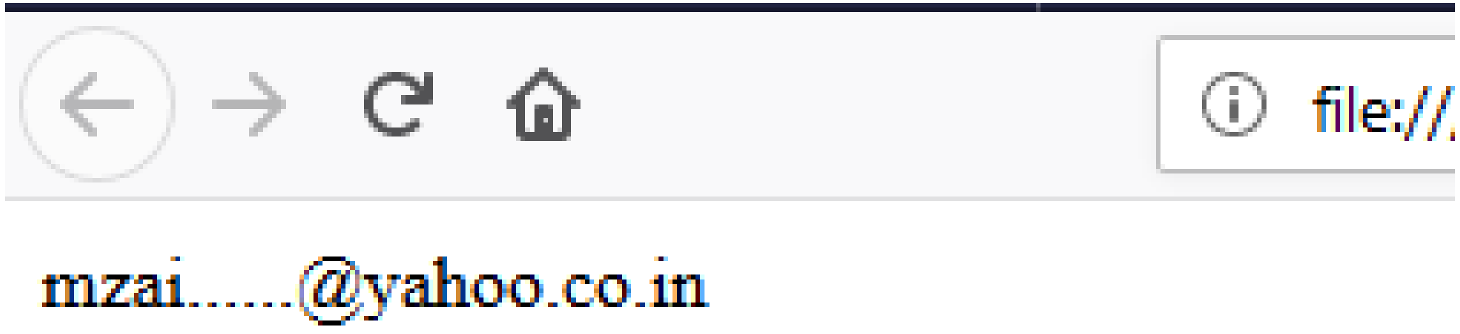
  Ember.js was introduced to the software market in 2015. It provides a reliable platform for handling the complicated User Interfaces. Popular websites like LinkedIn, Netflix, Nordstrom and many more use the Ember.JS platform for their websites.

Disabling Right Mouse Button

```
<html>
<head>
<script language="Javascript">
MyBanners=new Array('banner1.jpg','banner2.jpg','banner3.jpg')
banner=0
function ShowBanners()
{ if (document.images)
{ banner++
if (banner==MyBanners.length) {
banner=0}
document.ChangeBanner.src=MyBanners[banner]
setTimeout("ShowBanners()",1000)
}}</script>
<body onload="ShowBanners()" oncontextmenu="return false;">
<center>
<img src="banner1.jpg" width="200" height="200" name="ChangeBanner"/>
</center></body></html>
```

Concealing Email Address

```
<html>
<head>
<title>
Hiding Email Address
</title>
</head>
<body><script>
   function protect_email(useremail)
   {var avg,part1,part2;
   splitted = useremail.split("@");
   part1=splitted[0];
   part2=splitted[1];
   avg=part1.length/2;
   part1=part1.substring(0,(part1.length-avg));
   return part1+"......@"+part2;}
   document.write(protect_email("mzaidsa1@yahoo.co.in"));
</script></body>
</html>
```



```
mzai......@yahoo.co.in
```

Menus

- <div>
  - It defines a division or section in HTML document.
  - It is used as a container for HTML elements - which is then styled with CSS or manipulated with JavaScript.
  - It can be styled by using class or id rule.
- Background-color:
  - It sets the background color of an element.
  - Background is the total size of an element: including padding and border (but not margin)
- Color:
  - It can be used to set color of a text or border.
  - Colors can be specified using RGB values, HEX values, HSL (Hue, Saturation, Lightness) values or RGBA/HSLA values (A stands for Alpha - Transparency)

Menus

- Padding:
  - It is used to create space around an element’s content, inside of any defined borders.
  - Values: length px / pt / % / inherit(padding value should be inherited from parent elemment)
  - Properties: top – right – bottom – left
  - E.g.:

padding: 25px 50px 75px 100px	-> top(25) right(50) bottom(75) left(100)
padding: 25px 50px 75px	-> top(25) right and left (50) left (75)
padding: 25px 50px	-> top bottom (25) right left (50)
padding: 25px	-> All four paddings are 25

- Font-size:
  - Sets the font size of different elements. Default value is medium.
  - Values: small, medium, large, x-small/large, xx-small/large, smaller, larger, length px/cm, %

- Border:
  - It allows you to specify style, width and color of an element’s border.
  - Border-style: dotted, dash, solid, double, groove, ridge, inset, outset, none, hidden

Menus

- Cursor:
  - It generates a bunch of different mouse cursor.
  - Values: default, auto, cell, copy, crosshair, grab, help, pointer etc

- Position:
  - It specifies the type of positioning method used for an element
  - static: an element with position:static is not positioned in any special way, it is positioned according to the normal flow of the page,

    relative: an element with position:relative; is positioned relative to its normal position Setting the top, right, bottom, and left properties of a relatively-positioned element will cause it to be adjusted away from its normal position. Other content will not be adjusted to fit into any gap left by the element,

    fixed, absolute, sticky

Menus

- Position:
  - Fixed: an element with position:fixed; is positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled.

    Absolute: An element with position: absolute; is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed).

    Sticky: An element with position: sticky; is positioned based on the user's scroll position. A sticky element toggles between relative and fixed, depending on the scroll position. It is positioned relative until a given offset position is met in the viewport - then it "sticks" in place (like position:fixed).

Menus

- Display: It specifies the display behavior of an element.

  Values:

  inline: displays an element as an inline element. Any width and height property will have no effect.

  Block: displays an element a block element (like <p>). It starts on a new line and takes up the whole width.

  None: The element is completely removed. Can be used to hide an element.

Menus

- Overflow: This property controls what happens to content that is too big to fit into an area. It specifies whether to clip the content or to add scroll bars in the element.

  Values:

  visible: default. The overflow is not clipped, content renders outside the element's box.

  hidden: The overflow is clipped, and the rest of the content will be invisible.

  scroll: The overflow is clipped and scroll bar is added to the rest of the content.

  auto: similar to scroll, but it adds scrollbars only when necessary.

Menus

- Z-index: It specifies the stack order of an element. An element with greater stack order is always in front of an element with a lower stack order.
  - It only works on positioned items.
  - It two positioned elements overlap without z-index specified, the element positioned last in the HTML code will be shown on top.
  - Auto: Default. Sets the stack order equal to its parent.
  - Number: Sets the stack order of an element. Negative numbers are allowed.
  - Initial: Sets this property to its default value.
  - Inherit: Inherits this property from its parent element.

Menus

- float: It specifies how an element should float, used for positioning and formatting content. E.g. an image should float left to the text in a container.
    - Values:

      left: element floats to the left of its container.

      right: element floats to the right of its container.

      None: Element will not float. It will be displayed just where it occurs in the text. (default).

      Inherit: Element inherits float value of its parents.

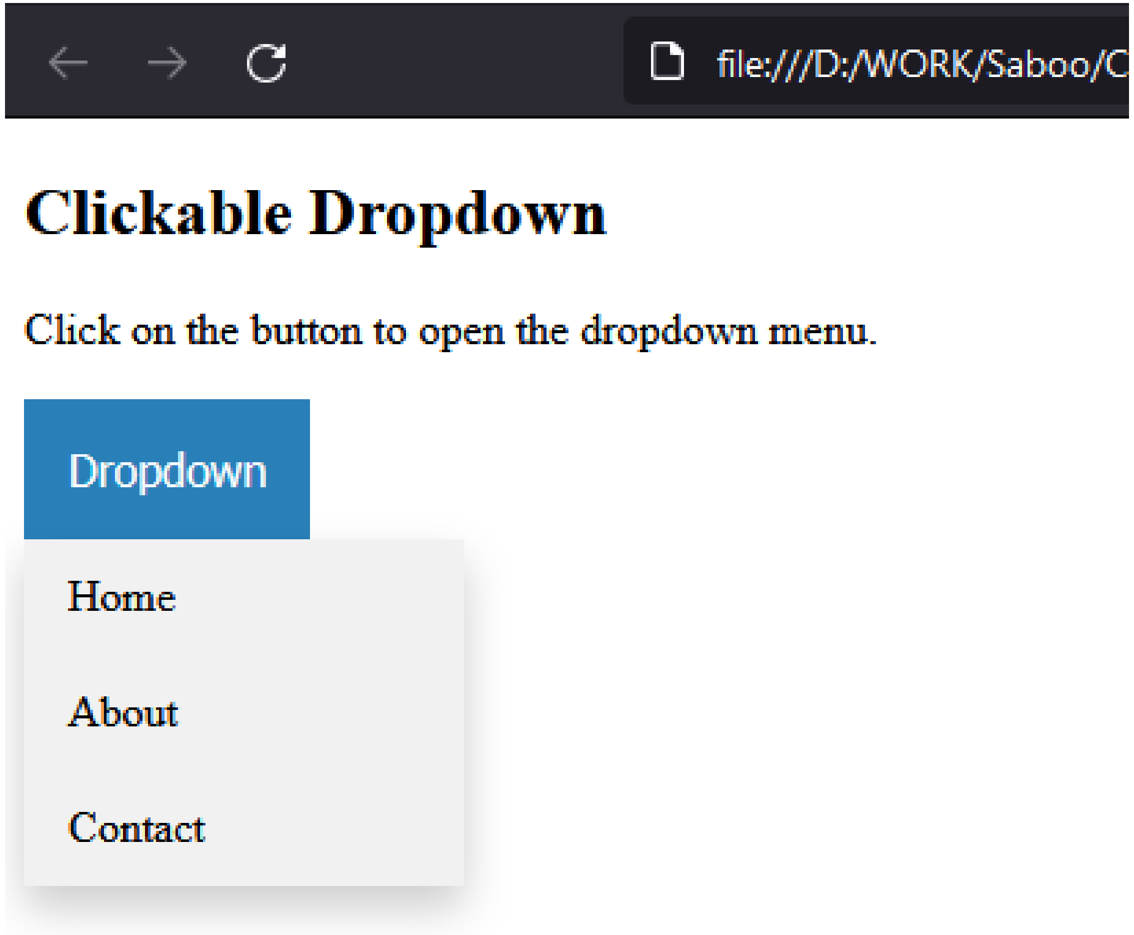
Menus

- margin: It is used to create space around elements, outside of any defined borders.
  - Values: top, right, bottom, left

- Outline: It is a line drawn outside the element’s border.
  - Properties: Style, color, width, offset (adds space between the outline and the border)

Clickable Dropdown Menu

```
<style> .dropbtn {  background-color: #3498DB;
  color: white;
  padding: 16px;
  font-size: 16px;
  border: none;
  cursor: pointer;}
.dropbtn:hover, .dropbtn:focus {
  background-color: #2980B9; }
.dropdown { position: relative;
  display: inline-block;}
.dropdown-content { display: none;
  position: absolute;
  background-color: #f1f1f1;
  min-width: 160px;
  overflow: auto;
  box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);
  z-index: 1; }
.dropdown-content a {  color: black;
  padding: 12px 16px;
  text-decoration: none;
  display: block;}
.dropdown a:hover {background-color: #ddd;}
.show {display: block;} </style>
```



Clickable Dropdown

Click on the button to open the dropdown menu.

Clickable Dropdown Menu

```
<body>
<h2>Clickable Dropdown</h2>
<p>Click on the button to open the dropdown menu.</p>
<div class="dropdown">
  <button onclick="myFunction()" class="dropbtn">Dropdown</button>
  <div id="myDropdown" class="dropdown-content">
    <a href="#home">Home</a>    <a href="#about">About</a>    <a href="#contact">Contact</a>
  </div> </div>
<script> /* When the user clicks on the button, toggle between hiding and showing the dropdown content */
function myFunction() {
  document.getElementById("myDropdown").classList.toggle("show"); }
// Close the dropdown if the user clicks outside of it
window.onclick = function(event) {
  if (!event.target.matches('.dropbtn')) {
    var dropdowns = document.getElementsByClassName("dropdown-content");
    var i;
    for (i = 0; i < dropdowns.length; i++) {      var openDropdown = dropdowns[i];
      if (openDropdown.classList.contains('show')) {
        openDropdown.classList.remove('show');
      }    }  }   }</script></body>
```