

Chapter 6: Menus, Navigation and Webpage protection (12 marks)

Status bar - builds a static message, changing the message using rollover, moving the message along the status bar.

Banner - loading and displaying banner advertisement. Linking a banner advertisement to *url*.

Slide Show - creating a slide show

Menus- creating a pulldown menu, dynamically changing a menu, validating menu selection, Floating menu, chain select menu, tab menu, pop-up menu, sliding menu, highlighted menu, folding a tree menu, context menu, scrollable menu, side bar menu.

Protecting web page - hiding your code, disabling the right mouse button, JavaScript, concealing email address.

Frameworks of Javascript and its application.

Builds a static message

The content of the status bar is the value of the window object's status property.

To display a message on the status bar, you'll need to assign the message to the status property of the window object.

The following statement assigns a string to the status property, which appears on the status bar once the browser executes this statement.

window.status= 'Trade secrets are revealed in the Demystified Series.'

Refers to current
window

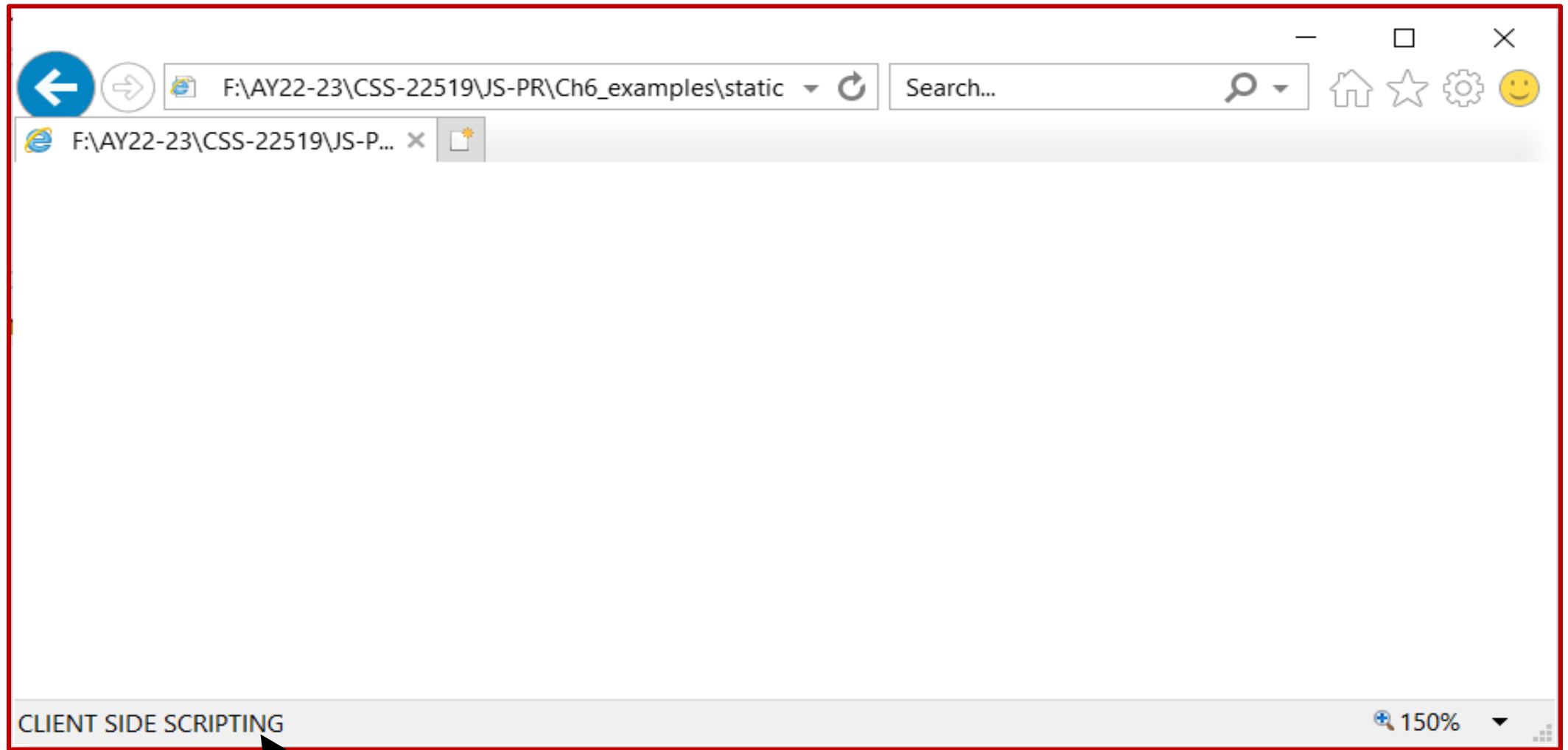
Property to write on
status bar

Message that we want to display on status bar

Example:

```
<html>
  <head>
    <script>
      window.status = 'CLIENT SIDE SCRIPTING'
    </script>
  </head>
</html>
```

Output window is
shown on next page for
example.



Status message on status
bar

Changing the message using rollover

Changing the message using rollover means we have to change status bar message on onmouseover event of a text/image or any relevant element.

```
<html>
  <body>
    <p onmouseover="window.status='Maharashtra State Board of Technical Education'"
      onmouseout="window.status=' '>MSBTE
    </p>
    <p onmouseover="window.status='All India Council for Technical Education'"
      onmouseout="window.status=' '>AICTE
    </p>
  </body>
</html>
```

Moving the Message Along the Status Bar

You can spice up any message on the status bar by displaying letters individually, giving the message a sense of movement.

M

Mo

Mov

Movi

Movin

Moving

Moving t

Moving te

Moving tex

Moving text

`Msg.substr(0,1)`

`Msg.substr(0,2)`

`Msg.substr(0,3)`

`Msg.substr(0,4)`

`Msg.substr(0,5)`

`Msg.substr(0,6)`

`Msg.substr(0,7)`

`Msg.substr(0,8)`

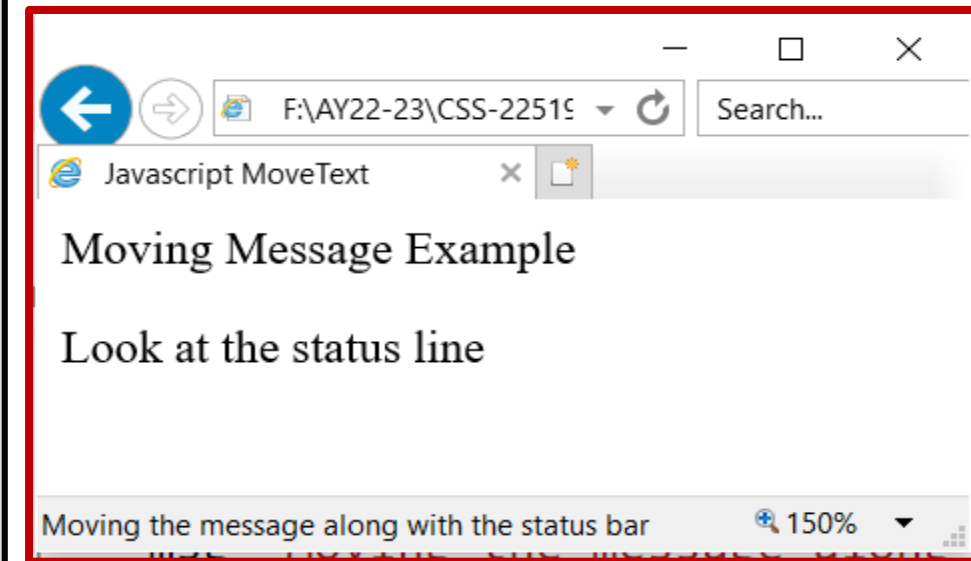
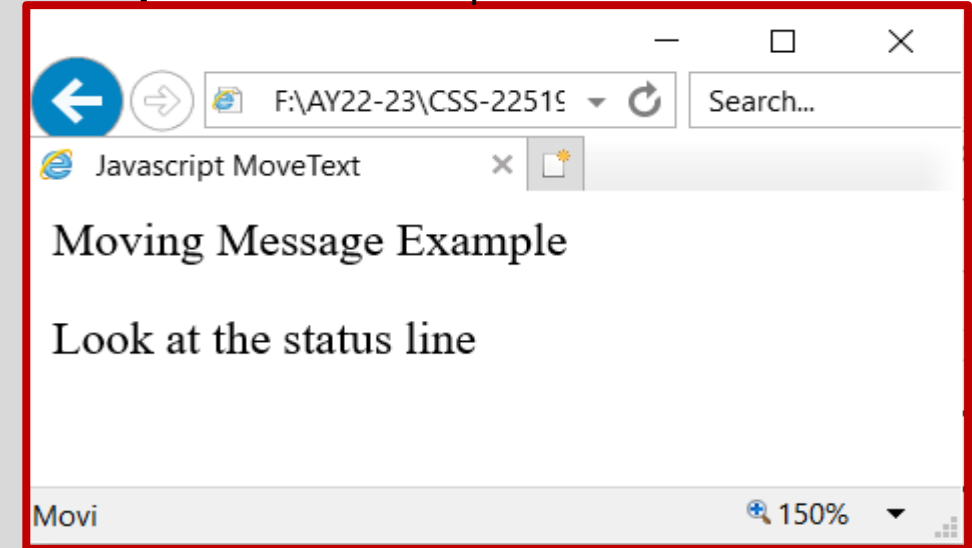
`Msg.substr(0,9)`

`Msg.substr(0,10)`

Moving the Message Along the Status Bar

```
<html>
  <head>
    <title>Javascript MoveText</title>
    <script language="javascript">
      msg="Moving the message";
      n=0;
      function MoveMessage()
      {
        window.status=msg.substring(0,n);
        n++;
        if(n>msg.length)
          n=0;
        window.setTimeout("MoveMessage()",150); //Recursion
      }
      MoveMessage();    //calling first time
    </script>
  </head>
  <body>
    <p>Moving Message Example</p>
    <p>Look at the status bar</p>
  </body>
</html>
```

Output



Banner Advertisement

The banner advertisement is the hallmark of every commercial web page. It is typically positioned near the top of the web page, and its purpose is to get the visitor's attention by doing all sorts of clever things.

Nearly all banner advertisements are in a file format such as a GIF, JPG, TIFF, or other common graphic file formats. Some are animated GIFs files.

You need to do three things to incorporate a banner advertisement in your web page:

1. Create several banner advertisements using a graphics tool such as PhotoShop. You'll want to make more than one advertisement so you can rotate them on your web page using a JavaScript.
2. Create an `` element in your web page with the height and width necessary to display the banner advertisement.
3. Build a JavaScript that loads and displays the banner advertisements in conjunction with the `` element.

Loading and displaying Banner advt.

- create an image element on your web page using the `` tag. You'll need to set four attributes of the `` tag: ***src***, ***width***, ***height***, and ***name***.
- Set the ***src*** attribute to the file name of the first banner advertisement that you want to display.
- Set the ***name*** attribute to a unique name for the image element. You'll be using the ***name*** attribute in the JavaScript when you change from one banner to the next.
- The image element (banner) should be cantered in the page using the `<center>` tag within the `<body>` tag.
- The final step is to build the JavaScript that will rotate the banners on your web page.

The JavaScript must do the following:

1. Load banner advertisements into an array.
2. Determine whether the browser supports the image object.
3. Display a banner advertisement.
4. Pause before displaying the next banner advertisement.

Detailed program is given on next page

```
<html>
<head>
<script language="Javascript">
MyBanners=new Array('banner1.jpg','banner2.jpg','banner3.jpg','banner4.jpg')
banner=0
function ShowBanners()
    { if (document.images)
        { banner++
          if (banner==MyBanners.length)
            {
              banner=0
            }
          document.ChangeBanner.src=MyBanners[banner]
          setTimeout("ShowBanners()",5000)
        }
    }
</script>
<body onload="ShowBanners()">
<center>

</center>
</body>
</html>
```

Linking Banner Ads with URL Links

Creating rotating banner images will provide the visitor to your webpage with some basic information. However, if you want the visitor to get more information by clicking on the banner images, you need to create rotating banner ads that contain URL links.

The script is basically the same as the previous one but we need to add another array that comprises the links, as follows:

```
MyBannerLinks =new Array('URL1','URL2','URL3/','URL4/')
```

You need to make sure that the arrangement of the links is in corresponding order with the banner images in the first array. Next, we create the ShowLinks function to link the current banner image to the relevant URL and then assign the URL to the href attribute of the anchor tag.

To load the banner images with URL links, we insert an anchor tag within the <body></body> section before the tag that displays the current banner image. The anchor tag's attribute href is used to call the ShowLinks() function when the visitor clicks on the banner.

Detailed example is shown on next page in banner.html file ...

```
<html><head>
<script language="Javascript">
    MyBanners=new Array('html_banner.jpg','css_banner.jpg','js_banner.jpg')
    MyBannerLinks=new Array('w3schools.com/html/default.asp','w3schools.com/css/default.asp','w3schools.com/js/default.asp')
    banner=0
    function ShowLinks()
    {
        document.location.href="http://www." + MyBannerLinks[banner]
    }
    function ShowBanners()
    {
        if (document.images)
        {
            banner++
            if (banner==MyBanners.length)
            {
                banner=0
            }
            document.ChangeBanner.src=MyBanners[banner]
            setTimeout("ShowBanners()",5000)
        }
    }
}
</script>
<body onload="ShowBanners()">
    <center>
        <a href="javascript: ShowLinks()">
            
        </a>
    </center>
</body>
</html>
```

Creating Slideshow

A slideshow is similar in concept to a banner advertisement in that a slideshow rotates multiple images on the web page. However, unlike a banner advertisement, a slideshow gives the visitor the ability to change the image that's displayed:

the visitor can click the Forward button to see the next image and the Back button to see the previous image.

Next example shows how to implement a slideshow....

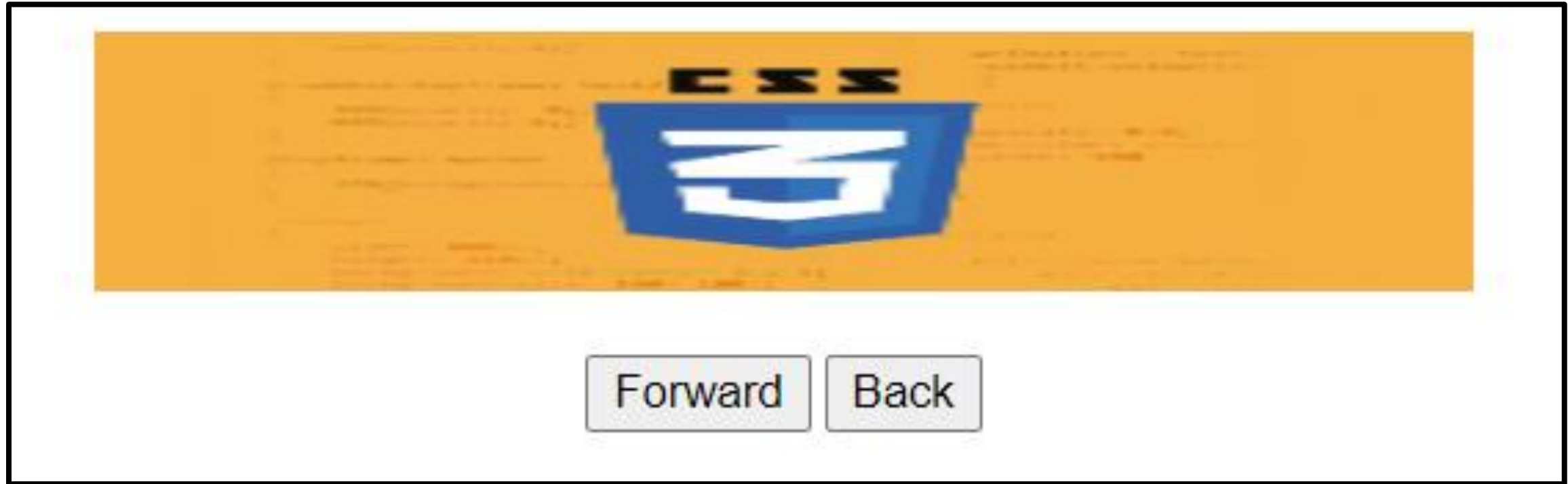
slideshow.html

```
<html>
<head>
  <title>Slideshow</title>
  <script src="slideshow.js">
  </script>
</head>
<body>
  <p align="center">
    
  </p>
  <center>
    <table border="0">
      <tr>
        <td align="center">
          <input type="button" value="Forward"
            onclick="RunSlideShow(1)">
          <input type="button" value="Back"
            onclick="RunSlideShow(-1)">
        </td>
      </tr>
    </table>
  </center>
</body>
</html>
```

slideshow.js

```
Pictures = new Array("html_banner.jpg","css_banner.jpg","js_banner.jpeg")
CurrentPicture = 0
function RunSlideShow(ForwardBack)
{
  if (document.images)
  {
    CurrentPicture = CurrentPicture + ForwardBack
    if (CurrentPicture > (Pictures.length - 1))
    {
      CurrentPicture = 0
    }
    if (CurrentPicture < 0)
    {
      CurrentPicture = Pictures.length - 1
    }
    document.PictureDisplay.src= Pictures[CurrentPicture]
  }
}
```

Output for slideshow



Webpage Protection

There is nothing secret about your web page. Anyone with a little computer knowledge can use a few mouse clicks to display your HTML code, including your JavaScript, on the screen. Although you cannot entirely prevent prying eyes from looking inside your web page, you can take a few steps to stop all.

Ways of Protecting your webpage

1. You can *disable use of the right mouse button* on your site so the visitor can't access the View Source menu option on the context menu.
2. You can *store your JavaScript on your web server* instead of building it into your web page. (External JS File)
3. You can *conceal your emails* to prevent spam emails.

Disabling the Right Mouse Button

Disable right click for window

```
<!-- Disable right click for window using HTML -->
<html>
  <body oncontextmenu="disable_right(); return false">

  </body>
  <script>
    function disable_right() {
      alert('Right Click is Disabled')
    }
  </script>
</html>
```

```
<!-- Disable right click for window using JS -->
<html>
  <body>

  </body>
  <script>
    window.addEventListener('contextmenu', e => e.preventDefault())
  </script>
</html>
```

Disabling the Right Mouse Button continue....

Disable right click for element on window

```
<!-- Disable right click for image element using HTML -->
<html>
  <body>
    
  </body>
  <script>
    function disable_right() {
      alert('Right Click is Disabled')
    }
  </script>
</html>
```

```
<!-- Disable right click for image element using JS -->
<html>
  <body>
    
  </body>
  <script>
    document.display.addEventListener('contextmenu', e => e.preventDefault())
  </script>
</html>
```

Conceal Email Address

Some spammers create programs called bots that surf the Net looking for e-mail addresses that are embedded into web pages,

Typically, bots identify e-mail addresses in two ways: by the `mailto:` attribute that tells the browser the e-mail address to use when the visitor wants to respond to the web page, and by the `@` sign that is required of all e-mail addresses.

Developer job is to confuse the bots by using a JavaScript to generate the e-mail address dynamically.

To conceal an e-mail address, you need to create strings that contain part of the e-mail address and then build a JavaScript that assembles those strings into the e-mail address, which is then written to the web page.

The following example illustrates one of many ways to conceal an e-mail address.

Conceal Email Address

```
<html>
<head>
<title>Conceal Email Address</title>
<script language=JavaScript>
    function CreateEmailAddress(){
        var x = 'help&example*c_o_m'
        var y = 'mai'
        var z = 'lto'
        var s = '?subject=Customer Help'
        x = x.replace('&', '@')
        x = x.replace('*', '.')
        x = x.replace('_', '')
        x = x.replace('_', '')
        var b = y + z + ':' + x + s
        window.location=b                //to redirect
    }
</script>
</head>
<body>
<input type="button" value="Help" onclick="CreateEmailAddress()">
</body>
</html>
```

Hiding Your JavaScript

You can hide your JavaScript from a visitor by storing it in an external file on your web server. The external file should have the **.js** file extension.

The browser then calls the external file whenever the browser encounters a JavaScript element in the web page. If you look at the source code for the web page, you'll see reference to the external **.js** file, but you won't see the source code for the JavaScript.