

METHODS

Using JavaScript in web design

Michael Shoikhedbrod*

Electromagnetic Impulse Inc., North York, Ontario, Canada

***Correspondence:**Michael Shoikhedbrod,
michaelshoikhedbrod@bell.net**Received:** 18 April 2023; **Accepted:** 15 May 2023; **Published:** 21 June 2023

JavaScript was created to bring a web page to life. Scripts are programs written in JavaScript. Each script is connected directly to HyperText Markup Language (HTML) in any browser and is immediately executed when the web page is loaded. JavaScript scripts are written in plain text that does not require special training, which is why JavaScript is very different from the Java programming language, to which it is very similar. A special interpreter program allows JavaScript to be used not only in the browser but anywhere, and therefore the process of executing script commands is called “interpretation.” The interpreter executes the program’s JavaScript source code (script) “as is” in browsers. Modern interpreters convert JavaScript to machine code before execution, optimize it, and only then perform it. Because of this, JavaScript is very fast. All known browsers have a JavaScript interpreter built into them, which is why they can execute scripts on the page. However, JavaScript can be used outside of the browser. This is a full-fledged language, with programs that can be run on a server and even on consumer electronic devices, if they have the appropriate interpreter installed. This article presents various JavaScript applications, used by the author, embedded in HTML files that allow a web designer to make a text animation presentation on a web page, to make a slide show, to create a questionnaire with ready-made answers so that the user reading the web page can easily master the necessary material contained in the questionnaire, to conduct an exam with the user on any topic of interest to him or her in a multitask mode with assessments of the exam results and a table of correct answers, place scientific applications on the web page, for example, a table of chemical elements by Mendeleev with printing of all parameters of a chemical element at the request of the user or calculators that the user can use online. The author has managed to build HTML using JavaScript that allows displaying beautiful Java applets accompanied by a banner, designed as a Java applet, contained within a moving window that expands from the left end and moves to the right end at the top of the web page.

Keywords: Java programming language, JavaScript, Java applets, HTML, browser

Introduction

JavaScript was created to bring a web page to life. Scripts are programs written in JavaScript. Each script is connected directly to HyperText Markup Language (HTML) in any browser and is immediately executed when the web page is loaded (1–4).

JavaScript scripts are written in plain text that does not require special training, which is why JavaScript is very different from the Java programming language, to which it is very similar. A special interpreter program allows JavaScript to be used not only in the browser, but anywhere, and

therefore the process of executing script commands is called “interpretation.”

The interpreter executes the program’s JavaScript source code (script) “as is” in browsers.

Modern interpreters convert JavaScript to machine code before execution, optimize it, and only then perform it. Because of this, JavaScript is very fast.

All known browsers have a JavaScript interpreter built into them, which is why they can execute scripts on the page.

However, JavaScript can be used outside of the browser.

This is a full-fledged language, with programs that can be run on a server and even on consumer electronic devices if they have the appropriate interpreter installed.

JavaScript is a “safe” general-purpose programming language, as it does not use low-level memory, processors, or manipulation tools.

When JavaScript was created, it was originally aimed at browsers that do not use low-level means of working with memory, such as the processor.

JavaScript manifests its capabilities depending on the environment in which it works.

In a browser environment, JavaScript can perform all manipulations with a web page and communicate with the user as well as with the server (5–11):

Creating new HTML tags, deleting existing ones, changing element styles, hiding, and showing elements, etc.

Provide a friendly user interface by handling mouse clicks, cursor movements, and keystrokes, etc.

Send requests to the server and download data without reloading the page.

This article presents various JavaScript applications, used by the author, embedded in HTML files that allow a web designer to make a text animation presentation on a web page, to make a slide show; to create a questionnaire with ready-made answers so that the user reading the web page can easily master the necessary material, contained in the questionnaire; conduct an exam with the user on any topic of interest to him in a multi-task mode with assessments of the exam results and a table of correct answers; place scientific applications on the web page, for example, a table of chemical elements of D.I. Mendeleev with printing of all parameters of a chemical element at the request of the user or calculators that the user can use online.

The author has managed to build HTML using JavaScript that allows displaying beautiful Java applets accompanied by a banner, designed as a Java applet, contained within a window that expands from the left end and moves to the right end at the top of the web page.

Materials for research

The developed HTML, using JavaScript applications, are as follows: TextAnim.html; slide.html; Question and Answer.html; popquiz.htm, results.htm; plselect.html; PeriodTableMendeleev.html; Clock Local time.html; calculator.html; and Banner2.html, bunner2 + lens.html, bunner2 + lake1.html, bunner2 + lake.html, bunner2 + guci.html; animation.html.

Research and results

JavaScript programs can be inserted anywhere in HTML using the SCRIPT tag.

The script tag contains the executable code. Previous HTML standards required the type attribute to be specified, but this is no longer required.

A simple `<script>` is enough. Browser when it sees `<script>`:

1. Starts displaying the page, showing part of the document before the script.
2. Having met the script tag, it switches to JavaScript mode and does not show but executes its content.
3. When finished, it switches back to HTML mode and only then displays the rest of the document.

The JavaScript, used by the author, embedded in the HTML file TextAnim.html, allows a web designer to create a text animation presentation, using the following JavaScript functions: `function nextSize(i,incMethod,textLength);` `function sizeCycle(text,method,dis),` loaded by `function doWave(n),` using HTML tag `<body onload = “doWave(0)”>` and variable `var theText = “Michael Shoikhedbrod presents.”`

Figure 1, which is a screenshot of the HTMLTextAnim.html file operations, illustrates the wavy animation of the text “Michael Shoikhedbrod presents.”

To create a slideshow on a web page that can be used by any user who accessed the web page on the Internet, the author used the JavaScript function `imageArray()` and variable of `images: var imgz = new imageArray(“0.jpg,” “1.jpg,” “2.jpg,” “3.jpg,” “4.jpg,” “5.jpg,” “6.jpg,” “7.jpg,” “8.jpg,” “9.jpg”)` in the slide.html file.

Figure 2, which is a screenshot of the three frames of the slide.html operation, illustrates the three frames of the slideshow on the website.

JavaScript, used by the author and embedded in an HTML animation.html file, allows a web designer to create animation presentations on a web page.

Figure 3 shows a screenshot of the animated movement of a red cube on a web page.

The most interesting feature is the creation of a questionnaire with ready-made answers, using JavaScript, so that the user reading the web page can easily master the material he or she needs.

The HTML Question and Answer.html file that creates such a questionnaire with ready-made answers uses the function `showAnswer(whatForm)` the JavaScript function, which, using an HTML option tag such as `<option value = “JavaScript is a scripting language that allows you to create dynamic content updates, control multimedia, image animation, and more.”>` `What is JavaScript, and what is it used for? </option>` selects the question: `What is JavaScript and what is it used for?` and right there in a special window is the answer: `JavaScript is a scripting language that allows you to create dynamic content updates, media control, image animation, and much more.`

Figure 4, which is a screenshot of the two windows of the browser with question and answer, illustrates the question and answer.html operations.



FIGURE 1 | Screenshot of the html TextAnim.html file operation, illustrates the wavy animation of the text “Michael Shoikhedbrod presents.”

Of particular interest is the user’s ability to conduct an online exam on a website on any topic of interest to him or her in a multitasking mode with exam scores and a table of correct answers.

HTML files: popquiz.htm, results.htm, and solution.js permit it.

Figure 5 illustrates two screenshots of the HTML popquiz.htm file, illustrating a quiz given to the user with the ability to mark the correct answers according to the user’s opinion.

After answering the quiz, the user, by clicking on “Grade Me!” will receive an assessment of his or her quiz with the help of an HTML results.htm file (**Figure 6**).

The HTML results.htm file has two buttons: one “Take the quiz again” for returning to the quiz, and the other “View solution” for printing the correct answers using the JavaScript solution.js file.

Figure 7 shows the screenshot of the quiz solution in a separate window, obtained by the JavaScript solution.js file in the background of instant quiz results.

For scientific users, it will be of interest for use online, placed by the author on the website tools: a calculator, a distance calculator, a clock of current local date and time, and the periodic table of chemical elements of Mendeleev.

The HTML calculate.html file, which uses the JavaScript language, allows performing online on a calculator all the necessary arithmetic operations: addition, subtraction, division, multiplication, and taking the square root.

Figure 8 shows the screenshot of the calculator in the HTML calculate.html file, permitting performing online all the necessary arithmetic operations: addition, subtraction, division, multiplication, and taking the square root.

The HTML Distcalculate.html file, using JavaScript, allows, on request, to calculate and display online the distance between cities around the world.

Figure 9 demonstrates the air distance between, for example, Moscow and New York.

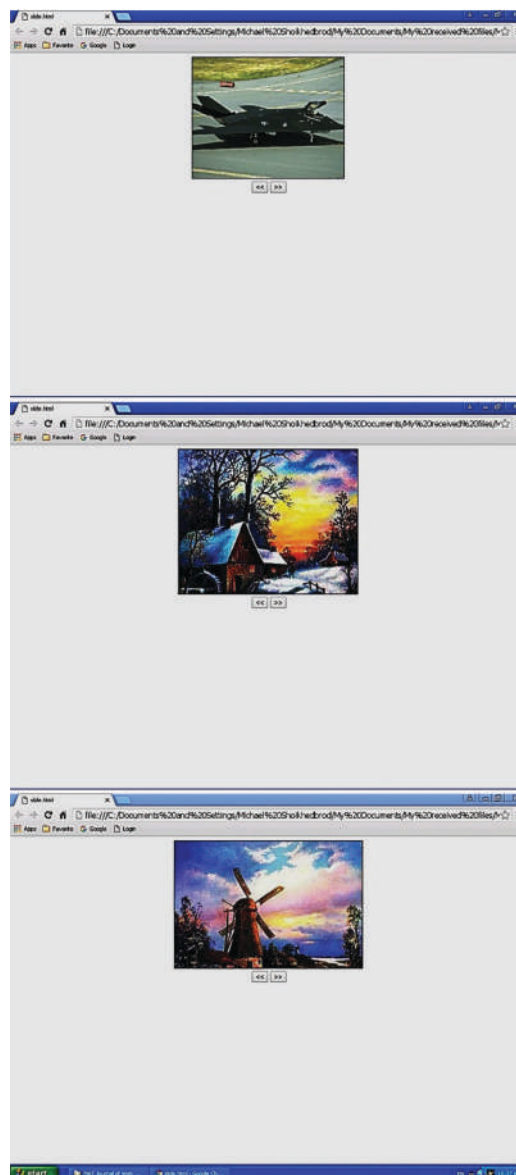


FIGURE 2 | Screenshot of the frames of the slide.html job, illustrating the three frames of the slideshow.

The HTML localtime.html file, using JavaScript, allows you to click “here” on a web page (**Figure 10**) to get the current date and exact time (**Figure 11**).

The most useful for scientists is the HTML PeriodTableMendeleev.html file, using JavaScript, which not only displays the Mendeleev periodic table of chemical elements but also prints online on the user’s request the chemical element itself, its number, weight, shells, orbitals, melt, and boil (**Figure 12**).

The author has succeeded in creating HTML that allows displaying the beautiful Java applets, accompanied by a banner made as a Java applet, contained in a window that expands from the left end and moves to the right end at the top of the web page.

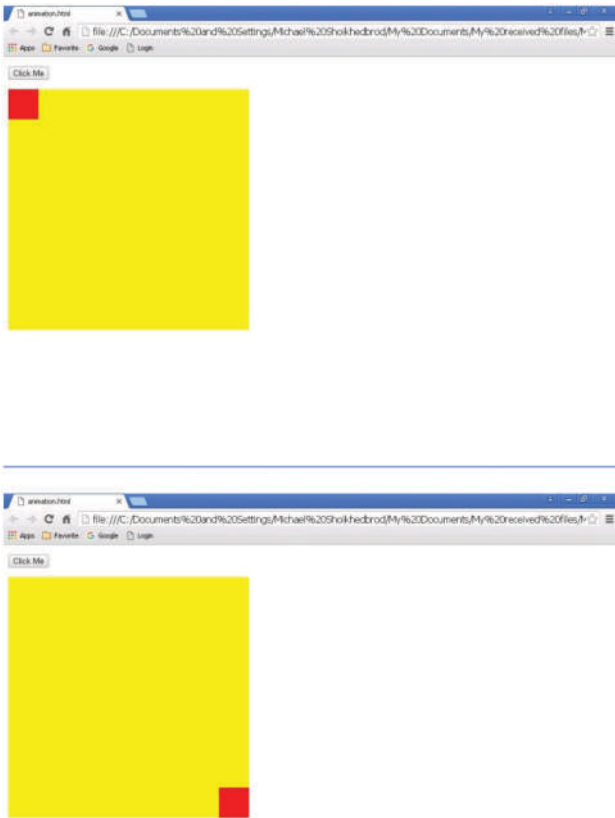


FIGURE 3 | Screenshot of the animated movement of a red cube on a web page.

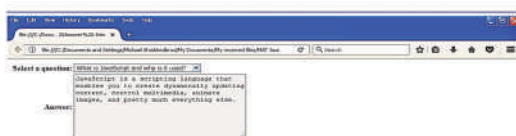
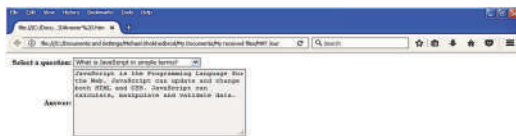


FIGURE 4 | Screenshots of the two windows of the browser with question and answer illustrate the question and answer.html operations.

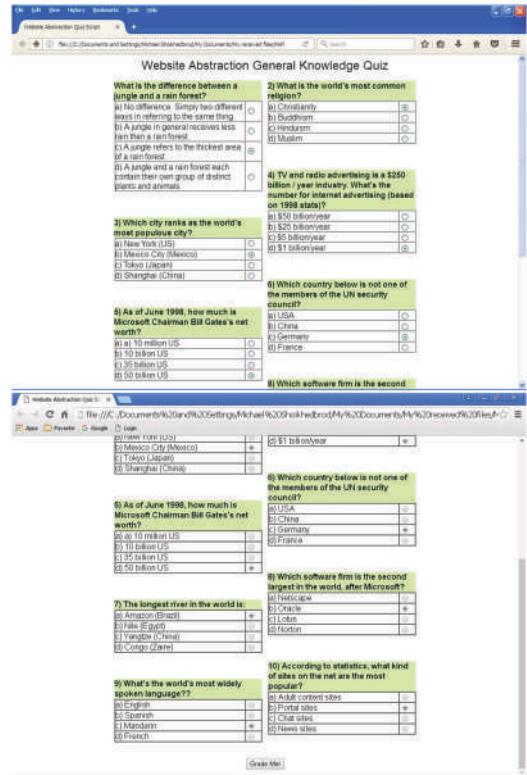


FIGURE 5 | Two screenshots of popquiz.html operations.

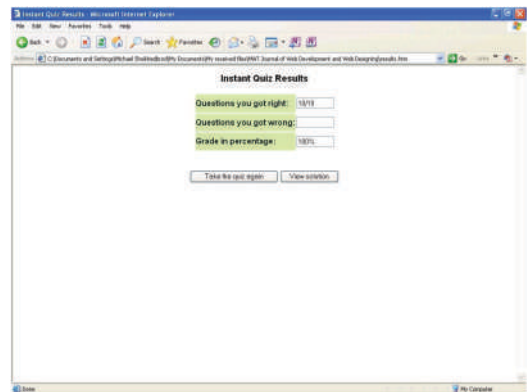


FIGURE 6 | Screenshot of results.html operations.

The developed HTML file includes two HTML files, bunner2 + lens.html and bunner2.html, and calls bunner2.html using the JavaScript of bunner2 + lens.html.

The bunner2 + lens.html displays a beautiful Java applet, accompanied by a banner (bunner2.html), made as a Java applet, contained in a window that expands from the left end and moves to the right end at the top of the web page, and has the following source codes:

```

<HEAD> <TITLE> AnyfLens - realtime lens over a
GIF/JPG pic! </TITLE>
</HEAD>
<BODY aLink = #ff8000 bgColor = #404040 link = #5fff5f
onload = go() text = #ffff22

```

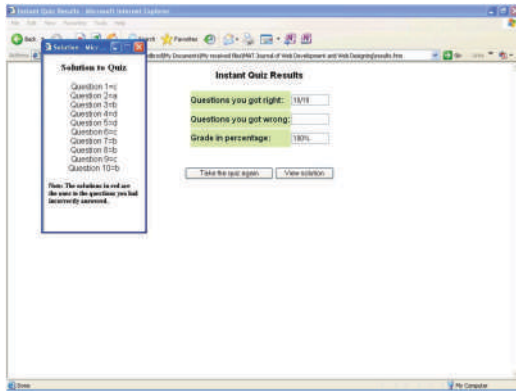


FIGURE 7 | Screenshot of the quiz solution in a separate window, obtained by the JavaScript solution.js file in the background of instant quiz results.

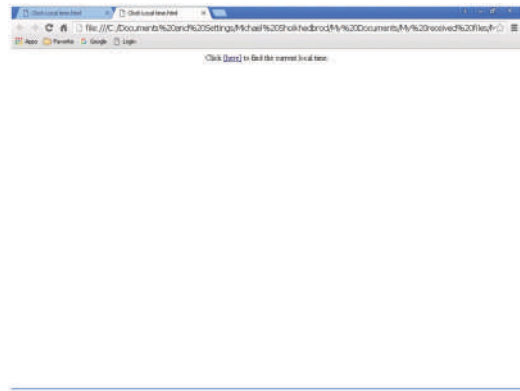


FIGURE 10 | Localtime.html file, using JavaScript, allows at click “here” on a web page to get the current date and exact time.

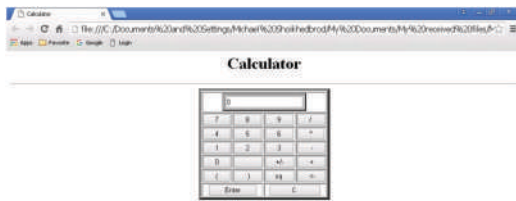


FIGURE 8 | Screenshot of the calculator in the HTML calculate.html file, permitting performing online all the necessary arithmetic operations: addition, subtraction, division, multiplication, and taking the square root.

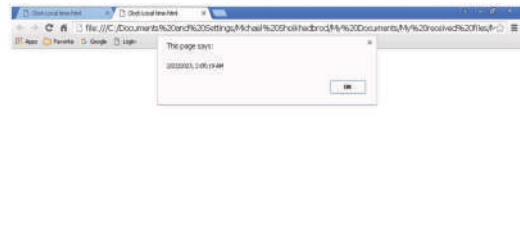


FIGURE 11 | Current date and exact time.



FIGURE 9 | Determination of distance between cities of the world.

```
vLink = #558a8b>
<SCRIPT>
var mylocation = "Banner2.html"
var winheight = 100
var winsize = 100
var x = 5
```

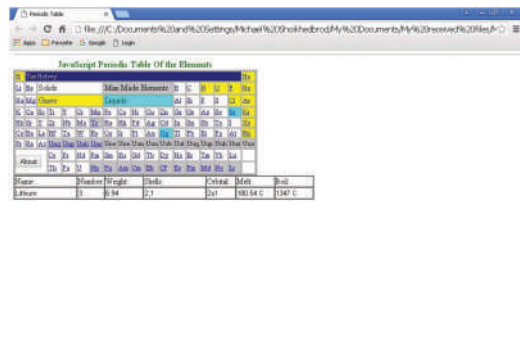


FIGURE 12 | Mendeleev’s periodic table of chemical elements and printing online on the user’s request the chemical element itself, its number, weight, shells, orbitals, melt, and boil.

```
function go(){
win2 = window.open(“”, “”, “scrollbars”)
if (!document.layers&&!document.all){
win2.location = mylocation
return
}
win2.resizeTo(100,120)
win2.moveTo(0,0)
```

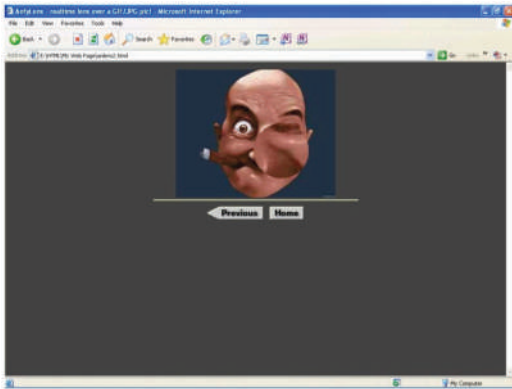



FIGURE 13 | Java applet lens is loaded and displayed on the website.

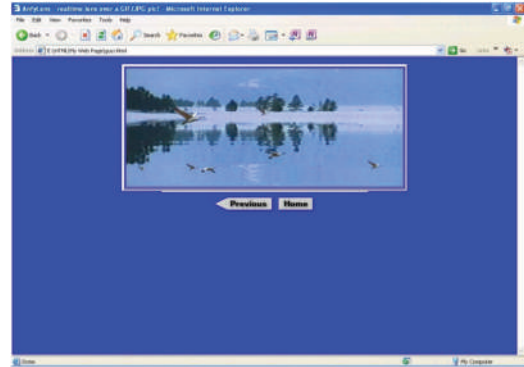


FIGURE 16 | Java applet Coowild is loaded and displayed on the website.

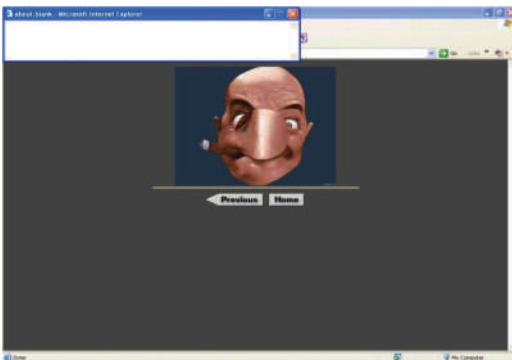


FIGURE 14 | The Java applet lens is then accompanied by a moving window that expands from the left edge to the right end of the right edge at the top of the web page.



FIGURE 17 | Bunner2 + coowild.html presents an applet that brings to life a picture of geese flying over a lapping lake with a banner at the top with changing advertising.



FIGURE 15 | The Java applet lens, accompanied by a designed banner (bunner2.html), contained in a moving window that expands from the left edge to the right end of the right edge at the top of the web page.

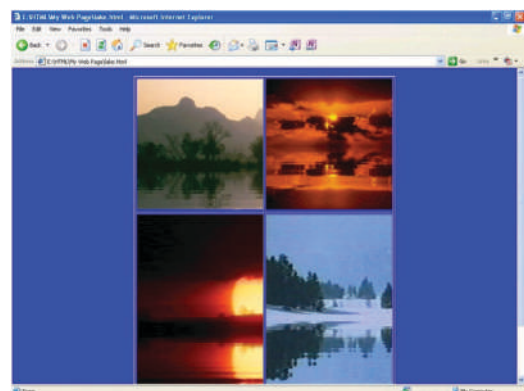


FIGURE 18 | Java applet lake is loaded and displayed on the website.

```

go2()
}
function go2(){
if (winheight >= screen.availHeight-3)
x = 0
win2.resizeBy(5,0)
// was win2.resizeBy(5,x)

```

```

winheight += 5
winsize += 5
if (winsize >= screen.width-5){
win2.location = mylocation
winheight = 100
winsize = 100
x = 5
return
}

```

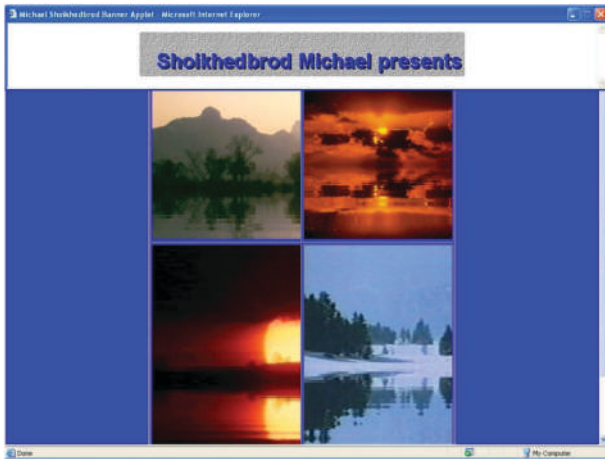


FIGURE 19 | Bunner2 + lake.html presents an applet that brings to life pictures of lapping lakes with a banner at the top with changing advertising.



FIGURE 21 | Bunner2 + lake.html presents an applet that brings to life pictures of lapping lakes with a banner at the top with changing advertising.

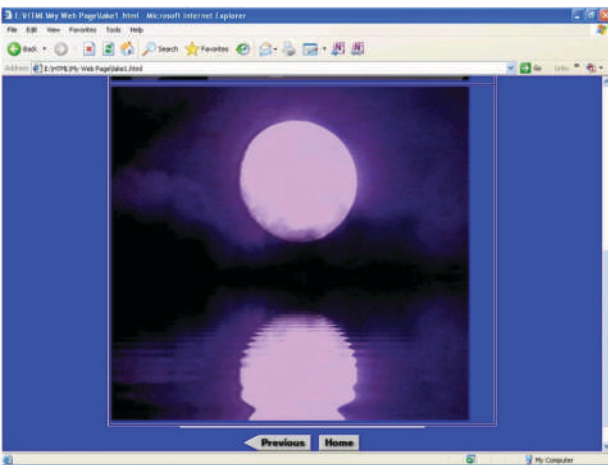


FIGURE 20 | Java applet lake is loaded and displayed on the website.

```

setTimeout("go2()",50)
}
</SCRIPT>
<CENTER>
<applet archive = "AnLens.jar" code = "AnLens.class"
width = 320 height = 256>
<param name = credits value = "Applet by Fabio Ciucci
(www.anfyjava.com)">
<param name = regcode value = "NO">
<param name = reglink value = "NO">
<param name = renewframe value = "YES">
<param name = regframe name value = "_blank">
<param name = statusmsg value = "Lens applet">
<param name = image value = "facet.jpg">
<param name = lenswidth value = "55">
<param name = distdval value = "10">
<param name = distoxy value = "0">
<param name = zoomfactor value = "12">
<param name = interactive value = "YES">

```

```

<param name = distort value = "NO">
<param name = spdX value = "2">
<param name = spdY value = "1">
<param name = overimg value = "NO">
<param name = overimgX value = "0">
<param name = overimgY value = "0">
<param name = memdelay value = "1000">
<param name = priority value = "3">
<param name = MinSYNC value = "10">

```

Sorry, your browser doesn't support Java.

```
</applet>
```

```
<TABLE border = 0 cellPadding = 0 cellSpacing = 0
width = 411>
```

```
<TBODY>
```

```
<TR>
```

```
<TD bgColor = #ffffcc height = 2> <IMG align = bottom
height = 2
```

```
src = "spacer.gif" width = 409> </TD> </TR> </TBODY>
```

```
</TABLE> <A
```

```
href = "demo.html"> <IMG
```

```
align = bottom border = 0 height = 27
```

```
src = "btn_PreviousEnabled.gif"
```

```
vspace = 12> </A> &nbsp; <A
```

```
href = "index.html"> <IMG
```

```
align = bottom border = 0 height = 27
```

```
src = "btn_HomeEnabled.gif"
```

```
vspace = 12> </A>
```

```
</P>
```

```
</CENTER>
```

```
</BODY> </HTML>
```

Bunner2.html has the following source codes:

```
<HTML>
```

```
<HEAD>
```

```
Michael Shoikhedbrod Banner Applet </TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```

<CENTER>
<APPLET CODE = "Banner2.class" WIDTH = 550
HEIGHT = 75>
<PARAM NAME = "text1" VALUE = "Shoikhedbrod
Michael presents">
<PARAM NAME = "text2" VALUE = "Shoikhedbrod Igor
presents">
<PARAM NAME = "text3" VALUE = "Shoikhedbrod
Ariel presents">
<PARAM NAME = "text4" VALUE = "Shoikhedbrod Irina
presents">
<PARAM NAME = "text5" VALUE = "Shoikhedbrod
Elena presents">
<PARAM NAME = "typeface" VALUE = "Helvetica">
<PARAM NAME = "fontcolor" VALUE = "blue">
<PARAM NAME = "backcolor" VALUE = "#FFE0FF">
<PARAM NAME = "backimage" VALUE = "Latar.gif">
<PARAM NAME = "maxfont" VALUE = "35">
<PARAM NAME = "delay" VALUE = "30">
<PARAM NAME = "shadow" VALUE = "true">
</APPLET>
</CENTER>
</BODY>
</HTML>

```

The developed HTML file works as follows:

First, the well-known Java applet lens is loaded and displayed on the website (Figure 13).

The Java applet lens is then accompanied by a moving window that expands from the left edge to the right end of the right edge at the top of the web page (Figure 14).

Figure 15 shows the Java applet lens, accompanied by a designed banner (bunner2.html), contained in a moving window that expands from the left edge to the right end of the right edge at the top of the web page.

The HTML bunner2 + coowild.html files work similarly, presenting a Java applet that brings to life a picture of geese flying over a lapping lake with a banner at the top with changing advertising.

First, the well-known Java applet Coowild is loaded and displayed on the website (Figure 16).

The Java applet Coowild is then accompanied by a designed banner (bunner2.html), contained in a moving window that expands from the left edge to the right end of the right edge at the top of the web page (Figure 17).

The HTML bunner2 + lake.html files present a Java applet that brings to life a picture of lapping lakes with a banner at the top with changing advertising.

First, the well-known Java applet lake is loaded and displayed on the website (Figure 18).

The Java applet lake is then accompanied by a designed banner (bunner2.html), contained in a moving window that expands from the left edge to the right end of the right edge at the top of the web page (Figure 19).

Figures 20, 21 show the use of bunner2 + lake.html, presenting an applet that brings to life another picture of lapping lakes with a banner in the top with changing advertising.

Conclusion

Thus, the various JavaScript applications, presented by the author and embedded in HTML files, showed the great possibilities that JavaScript provides to a web designer to build impressive live web pages that allow to create a text animation presentation on a web page, create a slide show, create questionnaires with ready-made answers so that the user, reading the web page, can easily master the necessary material contained in the questionnaire, conduct an exam with the user on any topic of interest to him or her in a multitasking mode with assessments of the exam results and a table of correct answers; place scientific applications on the web page, such as the table of chemical elements of Mendeleev with a printout of all the parameters of a chemical element, selected from the table, at the request of the user, and calculators that the user can use online.

The author has managed to create HTML, using JavaScript, that allows displaying beautiful animated Java applets, accompanied by a banner, implemented as a Java applet, contained in a moving window that expands from the left end and moves to the right end at the top of the web page.

References

1. Tilkov S, Vinoski S. Using JavaScript to build high-performance network programs. *IEEE Internet Comput.* (2010) 14:80–3. doi: 10.1109/MIC.2010.145
2. Quigley E. *JavaScript by example*. Hoboken, NJ: Hall Professional (2004).
3. White A. *JavaScript programmer's reference*. Hoboken, NJ: John Wiley & Sons (2010).
4. Doernhoefer M. JavaScript. *ACM Sigsoft Softw Eng Notes.* (2006) 31:16–24.
5. Bradenbaugh J. *JavaScript application cookbook*. Newton, MA: O'Reilly Media, Inc (1999).
6. Sun K, Ryu S. Analysis of JavaScript programs: challenges and research trends. *ACM Comput Surv.* (2017) 50:1–34. doi: 10.1145/3106741
7. De Leeuw JR. A JavaScript library for creating behavioral experiments in a Web browser. *Behav Res Methods.* (2015) 47:1–12.
8. Madsen M, Livshits B, Fanning M. Practical static analysis of JavaScript applications in the presence of frameworks and libraries. *Proceedings of the 2013 9th joint meeting on foundations of software engineering*. New York, NY: Association for Computing Machinery (2013). p. 499–509.
9. Flanagan D. *JavaScript: the definitive guide*. Newton, MA: O'Reilly Media, Inc (2006).
10. Myers M. *A smarter way to learn Javascript, ASmarterWayToLearn.* (2014). Available online at: <https://www.pdfdrive.com/a-smarter-way-to-learn-javascript-cppe>
11. Freeman E, Robson E. *Head first JavaScript programming: a brain-friendly guide*. Newton, MA: O'Reilly Media, Inc (2014).