

Get 3 months access to 400+ books and courses for \$3/m! 23:58:04

Get Access Now (https://www.sitepoint.com/premium/products/Z2lkOi8vbGVhcm5hYmxlL1Byb2R1Y3QvMzA3?utm_source=smartbar&utm_medium=blog&utm_campaign=3-dollar-deal)
(<https://www.sitepoint.com/>) Blog Community Jobs Library Login

Get Premium
(/Premium/Products/Z2lkOi8vbGVhcm5hYmxlL1Byb2R1Y3QvMzA3?
Ref_source=Sitepoint&Ref_medium=Topnav&Ref_campaign=Add-A-Web-
Console-To-Your-Toolbox-Part-2&Ref_content=Html-Css)

HTML & CSS (<https://www.sitepoint.com/html-css/>) > May 10, 2012 > By [Jeff Friesen](https://www.sitepoint.com/author/jfriesen/) (<https://www.sitepoint.com/author/jfriesen/>).

Add a Web Console to Your Toolbox, Part 2

A *console* is a software artifact for reading line-oriented textual input from the keyboard and writing line-oriented textual output to the screen. Part 1 of this two-part series introduced you to a console library for embedding a console in a web page, and demonstrated the usefulness of such a console via a browser shell. Part 2 shows you how the console library is implemented.

Discover how Console Works

The console library is fairly complete, but you might want to extend it with new capabilities (e.g., echo asterisks while inputting a password). Alternatively, you might want to improve performance or implement error checking. Regardless of your purpose, you'll need to understand how the library works. The first step in gaining this knowledge is to grasp the library's overall architecture. Listing 1 presents an overview.



```
var Console =
```

```
{
```

```
  init: function(canvasName, numCols, numRows)
```

Get Access Now ([https://www.sitepoint.com/premium/products/Z2lkOi8vbGVhcm5hYmxlL1Byb2R1Y3QvMzA3?](https://www.sitepoint.com/premium/products/Z2lkOi8vbGVhcm5hYmxlL1Byb2R1Y3QvMzA3?utm_source=smartbar&utm_medium=blog&utm_campaign=3-dollar-deal)

(<https://www.sitepoint.com/>) [Blog](#) [Community](#) [Jobs](#) [Library](#) [Login](#)

utm_source=smartbar&utm_medium=blog&utm_campaign=3-dollar-deal)

```
  },
```

```
  clear: function()
```

```
  {
```

```
  },
```

```
  getLine: function(callback)
```

```
  {
```

```
  },
```

```
  echo: function(msg)
```

```
  {
```

```
  },
```

```
  render: function()
```

```
  {
```

```
  },
```

```
  writeChar: function(ch)
```

```
  {
```

```
  },
```

```
  scroll: function()
```

```
  {
```

```
  }
```

```
}
```

Get 3 months access to 400+ books and courses for \$3/m! 23:58:04

[Get 3 months Premium](#)

(/Premium/Products/Z2lkOi8vbGVhcm5hYmxlL1Byb2R1Y3QvMzA3?

Ref_source=Sitepoint&Ref_medium=Topnav&Ref_campaign=Add-A-Web-

Console-To-Your-Toolbox-Part-2&Ref_content=Html-Css)



Listing 1: The console library's skeletal structure

Get 3 months access to 400+ books and courses for \$3/m! 23:58:04

Listing 1 reveals a global object named `Console` consisting of seven function properties. The first four properties comprise the public API, whereas the last three (`writeChar`, `writeLine`, and `writeLn`) are considered private and not accessed. This list of properties is far from complete because the `init(canvasName, numCols, numRows)` function introduces additional properties.

Ref_source=Sitepoint&Ref_medium=Topnav&Ref_campaign=Add-A-Web-Console-To-Your-Toolbox-Part-2&Ref_content=Html-Css

Note

I could have "hidden" the final three properties by introducing expressions such as `Console.writeChar = function(ch) { /* code here */ }` within `init(canvasName, numCols, numRows)`. I chose to not do so to keep `init(canvasName, numCols, numRows)` from getting any longer.

Discover `init(canvasName, numCols, numRows)`

Listing 2 presents `init(canvasName, numCols, numRows)`.



```
init: function(canvasName, numCols, numRows)
```

The next few lines create and initialize a buffer to support double buffering. The goal is to avoid flicker by drawing into the buffer and then copying the buffer content to the canvas. Code snippets (e.g., <http://stackoverflow.com/questions/2795269/does-html5-canvas-support-double-buffering>) suggest that current browsers support

Get 3 months access to 400+ books and courses for \$3/m! 23:58:04

Get Access Now (https://www.sitepoint.com/premium/products/Z2lkOi8vbGVhcm5hYmxlL1Byb2R1Y3QvMzA3?utm_source=smartbar&utm_medium=blog&utm_campaign=3-dollar-deal)

Library Login

(/Premium/Products/Z2lkOi8vbGVhcm5hYmxlL1Byb2R1Y3QvMzA3?Ref_source=Sitepoint&Ref_medium=Topnav&Ref_campaign=Add-A-Web-Console-To-Your-Toolbox-Part-2&Ref_content=Html-Css)

X

A two-dimensional **screen** array for storing characters that are echoed to the console is now created. JavaScript implements a two-dimensional array as a one-dimensional row array of one-dimensional column arrays. The **Array** object is used to create the row array and then, for each row element, a column array whose reference is assigned to the row array element.

Although each row in this table could potentially store a different number of columns (which is known as a *ragged array*), I've chosen to fix the number of columns to the value passed to **Array**'s constructor. An element in the **Screen** array is accessed via syntax **Console.screen[row][col]** — row and column indexes are zero-based.

Moving on, the **Array** object is used to create a **keyQueue** array for storing character and special keystrokes (e.g., newline). This array behaves as a queue in which keystrokes are added at one end and removed at the other. Code that adds keystrokes to this queue is contained in a pair of event-handling functions that are registered with the canvas to respond to key-down and key-press events.

The key-down event handler responds to the backspace key only. I would have preferred to handle this key via key-press, but that event handler is not called when backspace is pressed in Internet Explorer, Chrome, or Safari contexts. After adding **b** to the queue, key-down executes **event.preventDefault()** to prevent the current page from being replaced by the previous page in Chrome's page history.

The key-press event handler also responds to the backspace key for Firefox and Opera. It ignores this key under these browsers (it would not be a good idea to add a second **b** code to the queue), but executes **event.preventDefault()** to prevent the current page from being replaced by the previous page in Opera's page history.

The key-press event handler also responds to the Enter/Return key by adding a newline character to the queue, and responds to keys whose codes range from 32 through 126 by calling **String.fromCharCode()** function on the code and adding the equivalent character to the queue. On Firefox, **keyCode** contains 0 for a character key (e.g., A), and the appropriate code must be obtained from **charCode**.



Note

Get 3 months access to 400+ books and courses for \$3/m! 23:58:04

Opera does not support `charCode`, but `keyCode` distinguishes between uppercase and lowercase characters in a key-press context.

Get Access Now ([https://www.sitepoint.com/premium/products/Z2lkOi8vbGVhcm5hYmxlL1Byb2R1Y3QvMzA3?](https://www.sitepoint.com/premium/products/Z2lkOi8vbGVhcm5hYmxlL1Byb2R1Y3QvMzA3?utm_source=smartbar&utm_medium=blog&utm_campaign=3-dollar-deal))

(<https://www.sitepoint.com/>) Blog Community Jobs Library Login

HTML provides a `tabindex` attribute and a `tabIndex` DOM property for placing elements into tab order.

[Ref: Zero-Stepoint&Ref_medium=Topnav&Ref_campaign=Add-A-Web-Console-To-Your-Toolbox-Part-2&Ref_content=Html-Css](#)

HTML provides a `tabindex` attribute and a `tabIndex` DOM property for placing elements into tab order. The canvas's `tabIndex` property for this purpose. Next, the canvas's `focus()` function is invoked to give this element the keyboard focus. Although focus is given on Firefox, focus is not given on Internet Explorer — you must press the Tab key once or click the mouse on the canvas.

The canvas manages a cursor via `cursorOn`, `cursorCounter`, and `cursorCounterMax` properties. The cursor is visible when `true` is assigned to `cursorOn` (the default value), and the cursor remains visible until `cursorCounter` reaches `cursorCounterMax`, at which point it is reset to 0. It then becomes invisible and remains as such for the same duration.

There are two final tasks for `init(canvasName, numCols, numRows)` to perform. First, it assigns the empty string to the `line` property, which is a buffer for storing characters until Enter/Return is pressed. Second, it invokes the `clear()` function to clear the console and reset the location of the cursor to the upper-left character position.

Discover `clear()`

Listing 3 presents `clear()`.

```
clear: function()
{
    for (var row = 0; row < Console.numRows; row++)
        for (var col = 0; col < Console.numCols; col++)
            Console.screen[row][col] = " ";
    Console.row = 0;
    Console.col = 0;
    Console.render();
}
```

Listing 3: Clearing the console



Listing 3 clears the console by assigning a space to each `screen` array element. (Although not very performant, I'm emphasizing clarity. I could probably speed this up by using `Array.prototype.fill()`. I'll leave this as an exercise for you to try.)

by `Console`'s `col` and `row` properties) to the upper-left character position, and renders `screen`'s contents onto the canvas. (I

Get Access Now (<https://www.sitepoint.com/premium/products/Z2lkOi8vbGVhcm5hYmxlL1Byb2R1Y3QvMzA3?discuss=https://www.sitepoint.com/>) Blog Community Jobs Library Login
utm_source=smartbar&utm_medium=blog&utm_campaign=3-dollar-deal

Get Premium
(/Premium/Products/Z2lkOi8vbGVhcm5hYmxlL1Byb2R1Y3QvMzA3?
Ref_source=Sitepoint&Ref_medium=Topnav&Ref_campaign=Add-A-Web-
Console-To-Your-Toolbox-Part-2&Ref_content=Html-Css)

Discover `getLine(callback)`

Listing 4 presents `getLine(callback)`.



```
getLine: function(callback)
```

{ **Get 3 months access to 400+ books and courses for \$3/m! 23:58:04**

```
    Console.render(); // update cursor
```

Get Access Now ([https://www.sitepoint.com/premium/products/Z2lkOi8vbGVhcm5hYmxlL1Byb2R1Y3QvMzA3?](https://www.sitepoint.com/premium/products/Z2lkOi8vbGVhcm5hYmxlL1Byb2R1Y3QvMzA3?utm_source=smartbar&utm_medium=blog&utm_campaign=3-dollar-deal))

[Blog](#) [Community](#) [Jobs](#) [Library](#) [Login](#)

utm_source=smartbar&utm_medium=blog&utm_campaign=3-dollar-deal

```
    if (Console.keyQueue.length == 0)
```

```
    {
```

```
        if (Console.line.length == 0)
```

```
        {
```

```
            if (callback != undefined)
```

```
                callback();
```

```
        }
```

```
        return null;
```

```
    }
```

```
    var ch = Console.keyQueue.shift();
```

```
    if (ch == "b") // handle backspace
```

```
    {
```

```
        if (Console.line.length != 0)
```

```
        {
```

```
            Console.line = Console.line.substr(0,
```

```
                Console.line.length-1);
```

```
            Console.echo(ch);
```

```
        }
```

```
        return null;
```

```
    }
```

```
    Console.echo(ch);
```

```
    if (ch == "n") // handle newline
```

```
    {
```

```
        var temp = Console.line;
```

```
        Console.line = "";
```

```
        return temp;
```

Get 3 months access to 400+ books and courses for \$3/m! 23:58:04

(/Premium/Products/Z2lkOi8vbGVhcm5hYmxlL1Byb2R1Y3QvMzA3?
Ref_source=Sitepoint&Ref_medium=Topnav&Ref_campaign=Add-A-Web-
Console-To-Your-Toolbox-Part-2&Ref_content=Html-Css)




```
}
```

Get 3 months access to 400+ books and courses for \$3/m! 23:58:04

```
Console.line += ch;
```

Get Access Now (https://www.sitepoint.com/premium/products/Z2lkOi8vbGVhcm5hYmxlL1Byb2R1Y3QvMzA3?utm_source=smartbar&utm_medium=blog&utm_campaign=3-dollar-deal)
(<https://www.sitepoint.com/>) Blog Community Jobs Library Login

Get 3 months Premium
Access to 400+ books and courses for \$3/m!
Ref_source=Sitepoint&Ref_medium=Topnav&Ref_campaign=Add-A-Web-Console-To-Your-Toolbox-Part-2&Ref_content=Html-Css

Listing 4: Getting a line of input

Listing 4 describes a polling function that continually checks for input and processes this input one character at a time. The first task is to show or hide the cursor, and this task is accomplished by invoking `Console.render()`. Because `getLine()` is continuously invoked by the console demo and browser shell applications, the illusion of a blinking cursor is maintained.

Note

The cursor's blink rate depends upon the delay value passed to `setInterval()`. The larger the delay value, the slower the cursor blinks.

The next task is to determine whether any characters are present in the queue. If the queue is empty, `getLine()` can return. However, it first needs to invoke any callback function passed as an argument, but can only invoke this function when the `line` buffer is empty (a line of input is not in progress), to prevent screwing up the input line as demonstrated while discussing the browser shell.

At this point, the queue contains a character that is subsequently removed. If this character is the backspace, and if the `line` buffer is not empty, the rightmost character is removed from the buffer and the backspace is echoed to the console to keep the `screen` array synchronized, and `null` is returned because a complete line of input is not yet available.

After echoing the character to the console, `getLine()` checks the current character to see if it is a newline. If so, the `line` buffer is reset to the empty string in anticipation of the next line of input, and its previous contents are returned. Otherwise, the current character is appended to this buffer, and `null` is returned because a complete line of input is not yet available.

Discover `echo(msg)`

Listing 5 presents `echo(msg)`.



```
echo: function(msg)
```

```
{
```

```
  for (var i = 0; i < msg.length; i++)
```

```
    (https://www.sitepoint.com/premium/products/Z2lkOi8vbGVhcm5hYmxlL1Byb2R1Y3QvMzA3?utm_source=smartbar&utm_medium=blog&utm_campaign=3-dollar-deal) console.log(msg.charAt(i)); console.render();
```

Get 3 months access to 400+ books and courses for \$3/m! 23:58:04

Get Access Now (https://www.sitepoint.com/premium/products/Z2lkOi8vbGVhcm5hYmxlL1Byb2R1Y3QvMzA3?utm_source=smartbar&utm_medium=blog&utm_campaign=3-dollar-deal)

(https://www.sitepoint.com/premium/products/Z2lkOi8vbGVhcm5hYmxlL1Byb2R1Y3QvMzA3?utm_source=smartbar&utm_medium=blog&utm_campaign=3-dollar-deal)

Login

Get 3 months access to 400+ books and courses for \$3/m! 23:58:04

(/Premium/Products/Z2lkOi8vbGVhcm5hYmxlL1Byb2R1Y3QvMzA3?Ref_source=Sitepoint&Ref_medium=Topnav&Ref_campaign=Add-A-Web-

Console-To-Your-Toolbox-Part-2&Ref_content=Html-Css)

Listing 5: Echoing a string to the console

Listing 5 echoes a string of characters to the console one character at a time, updating the current cursor position in the process. The code employs `writeChar(ch)` for this purpose, and I will explain this function shortly. After writing out the string, `Console.render()` is invoked to update the canvas with the contents of the `screen` array.

Discover render()

Listing 6 presents `render()`.



```
render: function()
```

```
{ Get 3 months access to 400+ books and courses for $3/m! 23:58:04
```

```
    Console.bufferCtx.fillStyle = "#000"; // black
```

Get Access Now ([https://www.sitepoint.com/premium/products/Z2lkOi8vbGVhcm5hYmxlL1Byb2R1Y3QvMzA3?](https://www.sitepoint.com/premium/products/Z2lkOi8vbGVhcm5hYmxlL1Byb2R1Y3QvMzA3?utm_source=sponsor&utm_medium=blog&utm_campaign=3-dollar-deal))

([https://www.sitepoint.com/premium/products/Z2lkOi8vbGVhcm5hYmxlL1Byb2R1Y3QvMzA3?](https://www.sitepoint.com/premium/products/Z2lkOi8vbGVhcm5hYmxlL1Byb2R1Y3QvMzA3?utm_source=sponsor&utm_medium=blog&utm_campaign=3-dollar-deal))

utm_source=sponsor&utm_medium=blog&utm_campaign=3-dollar-deal

```
    Console.ctx.canvas.height);
```

```
    Console.bufferCtx.fillStyle = "#0f0"; // green
```

```
    var y = 0;
```

```
    for (var row = 0; row < Console.numRows; row++)
```

```
    {
```

```
        var x = 0;
```

```
        for (var col = 0; col < Console.numCols; col++)
```

```
        {
```

```
            var s = Console.screen[row][col]+"";
```

```
            Console.bufferCtx.fillText(s, x+5, y+5);
```

```
            x += Console.charWidth;
```

```
        }
```

```
        y += Console.charHeight;
```

```
    }
```

```
    if (Console.cursorOn)
```

```
        Console.bufferCtx.fillStyle = "#0f0"; // green
```

```
    else
```

```
        Console.bufferCtx.fillStyle = "#000"; // black
```

```
    Console.bufferCtx.fillText("_", Console.col*Console.charWidth+5, Console.row*Console.charHeight+5); if
```

```
(++Console.cursorCounter == Console.cursorCounterMax) { Console.cursorCounter = 0; Console.cursorOn =
```

```
!Console.cursorOn; } Console.ctx.drawImage(Console.buffer, 0, 0); }
```

(/Premium/Products/Z2lkOi8vbGVhcm5hYmxlL1Byb2R1Y3QvMzA3?

Ref_source=Sitepoint&Ref_medium=Topnav&Ref_campaign=Add-A-Web-

Console-To-Your-Toolbox-Part-2&Ref_content=Html-Css)



Listing 6: Rendering the `screen` array to the buffer (and more)

Listing 6 is responsible for rendering the `screen` array to the buffer and updating the cursor. It renders to the background buffer, and ultimately copies this buffer to the canvas side buffer. It then covers the `screen` array with the background buffer. The background buffer is first cleared to black to remove potential garbage from a previous rendering.

Get Access Now ([https://www.sitepoint.com/premium/products/Z2lkOi8vbGVhcm5hYmxlL1Byb2R1Y3QvMzA3?](https://www.sitepoint.com/premium/products/Z2lkOi8vbGVhcm5hYmxlL1Byb2R1Y3QvMzA3?utm_source=smartbar&utm_medium=blog&utm_campaign=3-dollar-deal)
(<https://www.sitepoint.com/>) Blog Community Jobs Library Login
After setting the drawing color to green, `render()` iterates over the `screen` array, invoking the Canvas `drawText()` method to draw each character. An offset of five pixels is added to the character's upper-left corner to support an empty border that's drawn around the console. Characters are separated horizontally by `charWidth` pixels and vertically by `charHeight` pixels.

The next section of code is responsible for rendering the cursor. An appropriate fill style is chosen based on the value of `cursorOn`: green when true and black when false. Either a green or black underline is then displayed at the current cursor position. The black underline completely erases what was previously displayed.

Finally, the cursor's on/off interval is established: the cursor is visible over five calls to `render()` and invisible over five calls to this function.

Because `render()` is called by `getLine()`, and because `getLine()` is invoked repeatedly by the application under the control of `setInterval()`, a blinking cursor is observed.

Discover writeChar(ch)

Listing 7 presents `writeChar(ch)`.



```
writeChar: function(ch)
```

{Get 3 months access to 400+ books and courses for \$3/m! 23:58:04

```
    if (ch == "b")
        Get Access Now (https://www.sitepoint.com/premium/products/Z2lkOi8vbGVhcm5hYmxlL1Byb2R1Y3QvMzA3?
        (https://www.sitepoint.com/) Blog Community Jobs Library Login
        utm_source=smartbar&utm_medium=blog&utm_campaign=3-dollar-deal)
        if (Console.col == 0 && Console.row == 0)
            return; // cannot backspace past the upper-left corner
        Console.col--;
        if (Console.col < 0)
        {
            Console.col = Console.numCols-1;
            Console.row--;
        }
        Console.screen[Console.row][Console.col] = " "; return; } if (ch == "n") { Console.col = 0; if
(++Console.row >= Console.numRows) Console.scroll(); return; } Console.screen[Console.row][Console.col] = ch; if
(++Console.col >= Console.numCols) { Console.col = 0; if (++Console.row >= Console.numRows) Console.scroll(); } }
```

Get Premium
(/Premium/Products/Z2lkOi8vbGVhcm5hYmxlL1Byb2R1Y3QvMzA3?
Ref_source=Sitepoint&Ref_medium=Topnav&Ref_campaign=Add-A-Web-
Console-To-Your-Toolbox-Part-2&Ref_content=Html-Css)



Listing 7: Writing a single character to the `screen` array

Listing 7 writes its single character argument to the `screen` array and updates the current cursor position. If the character is a backspace, the appropriate element in the `screen` array is removed. Otherwise, if the character is a newline, the current row advances and the console scrolls vertically upward when necessary. Otherwise, the character is stored at the current position, which advances and possibly scrolls upward.

Discover `scroll`

Listing 8 presents `scroll()`.

```
scroll: function()
```

```
{ Get 3 months access to 400+ books and courses for $3/m! 23:58:04
```

```
    Console.row = Console.numRows-1;
```

```
    (https://www.sitepoint.com/?row=0; Console.numRows-1; row++) Library. Login
```

```
    Get Access Now (https://www.sitepoint.com/premium/products/Z2lkOi8vbGVhcm5hYmxlL1Byb2R1Y3QvMzA3?utm_source=smartbar&utm_medium=blog&utm_campaign=3-dollar-deal)
```

```
    for (var col = 0; col < Console.numCols; col++)
```

```
        Console.screen[row][col] = Console.screen[row+1][col];
```

```
    for (var col = 0; col < Console.numCols; col++)
```

```
        Console.screen[Console.numRows-1][col] = " "; }
```

Get Premium
(/Premium/Products/Z2lkOi8vbGVhcm5hYmxlL1Byb2R1Y3QvMzA3?
Ref_source=Sitepoint&Ref_medium=Topnav&Ref_campaign=Add-A-Web-
Console-To-Your-Toolbox-Part-2&Ref_content=Html-Css)

Listing 8: Scrolling the console upward one row

Listing 8 performs a simple scrolling operation that moves the contents of the `screen` array up by one row. The first row's contents are replaced by the second row's contents, and the final row is set to spaces. I've coded this function for clarity, but it could be improved from a performance perspective.

Conclusion

Console's implementation leaves lots of room for improvement. You can add missing features (e.g., echo asterisks while entering a password), boost `screen`-oriented loop performance (perhaps via `Array`'s `splice()` function), and make the library more robust through argument validation (e.g., compare what is passed with `undefined`) and exception throwing. Have fun.

Note

All files pertaining to this article are located in [code.zip](http://www.sitepoint.com/wp-content/uploads/2012/05/code.zip) (<http://www.sitepoint.com/wp-content/uploads/2012/05/code.zip>).



Meet the author

Jeff Friesen (<https://www.sitepoint.com/author/jfriesen/>)

Jeff Friesen is a freelance tutor and software developer with an emphasis on Java and mobile technologies. In addition to writing Java and Android books for Addison-Wesley, Jeff has written numerous articles on Java and other technologies for SitePoint, InformIT, JavaWorld, Java.net, and DevSource.

Get 3 months access to 400+ books and courses for \$3/m! 23:58:04

Get Access Now (https://www.sitepoint.com/premium/products/Z2lkOi8vbGVhcm5hYmxlL1Byb2R1Y3QvMzA3?utm_source=smartbar&utm_medium=blog&utm_campaign=3-dollar-deal)
(<https://www.sitepoint.com/>) [Blog](#) [Community](#) [Jobs](#) [Library](#) [Login](#)

Get Premium
(/Premium/Products/Z2lkOi8vbGVhcm5hYmxlL1Byb2R1Y3QvMzA3?
Ref_source=Sitepoint&Ref_medium=Topnav&Ref_campaign=Add-A-Web-
Console-To-Your-Toolbox-Part-2&Ref_content=Html-Css)

Stuff we do

- [Premium \(/premium/\)](/premium/)
- [Forums \(/community/\)](/community/)
- [Corporate memberships \(https://sitepoint.typeform.com/to/fNY7XG\)](https://sitepoint.typeform.com/to/fNY7XG)
- [Become an affiliate \(https://sitepoint.tapfiliate.com/\)](https://sitepoint.tapfiliate.com/)
- [Remote Jobs \(/jobs/\)](/jobs/)

About

- [Our story \(/about-us/\)](/about-us/)
- [Press room \(/press/\)](/press/)

Contact

- [Contact us \(/contact-us/\)](/contact-us/)
- [FAQ \(https://sitepoint.zendesk.com/hc/en-us\)](https://sitepoint.zendesk.com/hc/en-us)
- [Publish your book with us \(https://sitepoint.typeform.com/to/HtAXVN\)](https://sitepoint.typeform.com/to/HtAXVN)
- [Write an article for us \(https://sitepoint.typeform.com/to/DMmYfn\)](https://sitepoint.typeform.com/to/DMmYfn)
- [Advertise \(/advertise/\)](/advertise/)

Legals

- [Terms of use \(/legals/\)](/legals/)
- [Privacy policy \(/privacy-policy/\)](/privacy-policy/)

Connect



(<https://www.facebook.com/sitepoint>)



(<http://twitter.com/sitepointdotcom>)



(<https://www.sitepoint.com/feed/>)



(<https://plus.google.com/+sitepoint>)

Get 3 months access to 400+ books and courses for \$3/m! 23:58:04

Get Access Now ([https://www.sitepoint.com/premium/products/Z2lkOi8vbGVhcm5hYmxlL1Byb2R1Y3QvMzA3?](https://www.sitepoint.com/premium/products/Z2lkOi8vbGVhcm5hYmxlL1Byb2R1Y3QvMzA3?utm_source=smartbar&utm_medium=blog&utm_campaign=3-dollar-deal)
(<https://www.sitepoint.com/>) [Blog](#) [Community](#) [Jobs](#) [Library](#) [Login](#)

Get 3 months access to 400+ books and courses for \$3/m! 23:58:04
Premium
(/Premium/Products/Z2lkOi8vbGVhcm5hYmxlL1Byb2R1Y3QvMzA3?
Ref_source=Sitepoint&Ref_medium=Topnav&Ref_campaign=Add-A-Web-
Console-To-Your-Toolbox-Part-2&Ref_content=Html-Css)

