

HTMLHint

Static Code Analysis Tool for HTML

Robert Whitmer



HTMLHint is a Static Code Analysis Tool for HTML, you can use it with IDE or in build system.

build **passing** coverage **99%** npm **v0.9.13** downloads **89k/month** downloads **990k**

<https://github.com/yaniswang/HTMLHint>

Issues with analyzing our codebase

Our front-end is comprised of HTML template Pug files. Pug dynamically renders HTML files on runtime, so there is technically no HTML in the codebase to analyze, and there is no available Pug analysis tool.

Solution to analyzing Pug files

There are many tools available to convert HTML to Pug (Jade) and vice versa, so by converting our Pug file to HTML, we can then perform some analysis on it.

I used [html2jade](#) to convert our Pug files.



☒ SPACES ☐ TABS WIDTH OF INDENT

```
<!DOCTYPE html>
<html>
  <head>
    <title>PPA - Training Log</title>
    <!-- Compiled and minified CSS-->
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/materialize/0.96.0/css/materialize.min.css">
    <!-- Compiled and minified JavaScript-->
    <script src="https://cdnjs.cloudflare.com/ajax/libs/materialize/0.96.0/js/materialize.min.js"></script>
    <!-- Bootstrap-->
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
```

```
doctype html
html
  head
    title PPA - Training Log
    // Compiled and minified CSS
    link(rel='stylesheet', href='https://cdnjs.cloudflare.com/ajax/libs/materialize/0.96.0/css/materialize.min.css')
    // Compiled and minified JavaScript
    script(src='https://cdnjs.cloudflare.com/ajax/libs/materialize/0.96.0/js/materialize.min.js')
    // Bootstrap
    link(rel='stylesheet', href='https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css')
```

Analysis

Place your html code here:

Theme: Monokai

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>PPA - Training Log</title>
5 <!-- Compiled and minified CSS-->
6 <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/materialize/0.96.0/css/materialize.min.css">
7 <!-- Compiled and minified JavaScript-->
8 <script src="https://cdnjs.cloudflare.com/ajax/libs/materialize/0.96.0/js/materialize.min.js"></script>
9 <!-- Bootstrap-->
10 <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
11 <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>
12 <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
13 <!-- Import Google Icon Font-->
14 <link href="http://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">
15 <!-- Import materialize.css-->
16 <link type="text/css" rel="stylesheet" href="css/materialize.min.css" media="screen,projection">
17 <!-- Let browser know website is optimized for mobile-->
18 <meta name="viewport" content="width=device-width, initial-scale=1.0">
19 <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>
20 <!-- Load the AJAX API-->
21
```

Find Hints: 21

Previous

Next

Standard

- ☒ tagname-lowercase
- ☒ attr-lowercase
- ☒ attr-value-double-quotes
- ☐ attr-value-not-empty
- ☒ attr-no-duplication
- ☒ doctype-first
- ☒ tag-pair
- ☐ tag-self-close
- ☒ spec-char-escape
- ☒ id-unique
- ☒ src-not-empty
- ☒ title-require

Performance

- ☐ head-script-disabled

Accessibility

- ☐ alt-require

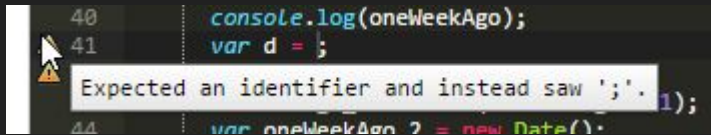
Specification

- ☐ doctype-html5
- ☐ id-class-value
- ☐ style-disabled
- ☐ inline-style-disabled
- ☐ inline-script-disabled
- ☐ space-tab-mixed-disabled
- ☐ id-class-ad-disabled
- ☐ href-abs-or-rel
- ☐ attr-unsafe-chars

Other

- ☐ csslint
- ☒ jshint

List of Issues

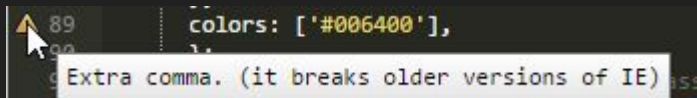


This issue is caused by the passing in of data to a Pug file. I was forced to use string interpolation to get this to work, so on the Pug end you will see:

```
var d = !{data_w};
```

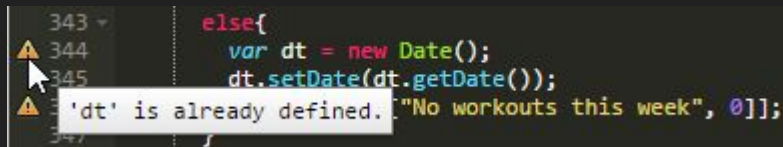
This issue can reasonably be ignored because there should always be data being passed in, and if not, the user will not see any output anyway.

List of Issues



A screenshot of a code editor with a dark theme. Line 89 contains the code `colors: ['#006400'],`. A yellow warning icon is on the left. A white tooltip box points to the comma at the end of the array, containing the text: "Extra comma. (it breaks older versions of IE)".

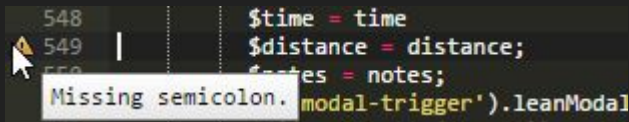
This issue is a simple fix that will hopefully allow use to have more cross-platform compatibility.



A screenshot of a code editor with a dark theme. Lines 343-347 show an `else{}` block. Line 344: `var dt = new Date();`. Line 345: `dt.setDate(dt.getDate());`. Line 346: `'dt' is already defined. "No workouts this week", 0];`. A yellow warning icon is on the left. A white tooltip box points to the `dt` in the array, containing the text: "'dt' is already defined."

This issue is not necessary to fix in my opinion because the variables are “defined” in the if portion, so only one can possibly occur.

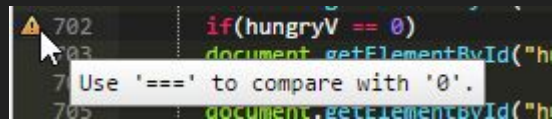
List of Issues



A screenshot of a code editor with a dark theme. Line 548 contains the code `$time = time`. Line 549 contains `$distance = distance;`. Line 550 contains `$notes = notes;`. Line 551 contains `modal-trigger').leanModal`. A yellow warning icon is on the left of line 551. A tooltip points to the end of line 551 with the text "Missing semicolon."

```
548 $time = time
549 $distance = distance;
550 $notes = notes;
551 modal-trigger').leanModal
```

A simple mistake; easy fix



A screenshot of a code editor with a dark theme. Line 702 contains the code `if(hungryV == 0)`. Line 703 contains `document.getElementById("h`. Line 704 contains `Use '===' to compare with '0'.`. Line 705 contains `document.getElementById("h`. A yellow warning icon is on the left of line 702. A tooltip points to the `==` operator with the text "Use '===' to compare with '0'."

```
702 if(hungryV == 0)
703 document.getElementById("h
704 Use '===' to compare with '0'.
705 document.getElementById("h
```

Another simple fix +
JavaScript style that I
was unaware of.

Other Pug files

Due to the nature of our site, the other Pug files we used all generated similar (no new) issues when analyzed. The myworkouts Pug file is by far the largest file we are using, so it is no surprise it contains all of the unique issues.

Interesting Conclusion about this analysis

The jslint tool was turned on along with HTMLHint, and **all of the issues** found were about the JavaScript inside the HTML.

I think the consequence of using Pug (which automatically generates the HTML) is that you avoid having any issues with the HTML itself. As long as the Pug file is not broken, it will produce good HTML code.