

Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for contents of text-input area

Checker Input

Show ☒ source ☐ outline ☐ image report [Options...](#)

Check by ☐ CSS

```

        var answerContainer = document.createElement('div');
        answerContainer.innerHTML = '<li>' + data.content + '</li>'
<p>Answered by: ' + data.user.username +
        '<p>Status: <th>' + (data.is_correct ? 'Correct' : 'None') +
        '</th></p><p>Date: ' + data.date + '</p>';

        // Create form for deleting the answer
        var deleteForm = document.createElement('form');
        deleteForm.action = 'http://localhost:8000/question/1000/answer';
        deleteForm.method = 'POST';

        // Add CSRF token input
        var csrfInput = document.createElement('input');
        csrfInput.type = 'hidden';
        csrfInput.name = '_token';

```

[Check](#)

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

[Message Filtering](#)

- Error** Start tag seen without seeing a doctype first. Expected `<!DOCTYPE html>.`
[From line 1, column 1; to line 2, column 8](#)
`<<<script><<docum`
- Error** Stray doctype.
[From line 126, column 1; to line 126, column 15](#)
`/script><<<!DOCTYPE html><<<html`
- Error** Stray start tag `html`.
[From line 127, column 1; to line 127, column 16](#)
`YPE html><<<html lang="en"><<<`
- Fatal Error** Cannot recover after last error. Any further errors will be ignored.
[From line 127, column 1; to line 127, column 16](#)

YPE html><html lang="en"><

Document checking completed.

Source

```

1.  ↵
2.  <script>↵
3.  document.addEventListener('DOMContentLoaded', function () {↵
4.      var answerForm = document.getElementById('answerForm');↵
5.  ↵
6.      answerForm.addEventListener('submit', function (e) {↵
7.          e.preventDefault();↵
8.  ↵
9.          var formData = new FormData(answerForm);↵
10. ↵
11.         // Make an AJAX request↵
12.         fetch('http://localhost:8000/question/1000/answer', {↵
13.             method: 'POST',↵
14.             body: formData,↵
15.             headers: {↵
16.                 'X-CSRF-TOKEN': 'fXRuuNVJIHpLk09Ke1ElHLRm5QkGSEMFhN2y22dF',↵
17.             },↵
18.         })↵
19.         .then(response => response.json())↵
20.         .then(data => {↵
21.             console.log(data);↵
22.             var answersContainer =
document.getElementById('answersContainer');↵
23. ↵
24.             // Create container div for the answer↵
25.             var answerContainer = document.createElement('div');↵
26.             answerContainer.innerHTML = '<li>' + data.content + '</li>'
<p>Answered by: ' + data.user.username + ↵
27.             '<p>Status: <th>' + (data.is_correct ? 'Correct' : 'None') +
'</th></p><p>Date: ' + data.date + '</p>';↵
28. ↵
29.             // Create form for deleting the answer↵
30.             var deleteForm = document.createElement('form');↵
31.             deleteForm.action =
'http://localhost:8000/question/1000/answer';↵
32.             deleteForm.method = 'POST';↵
33. ↵
34.             // Add CSRF token input↵
35.             var csrfInput = document.createElement('input');↵
36.             csrfInput.type = 'hidden';↵
37.             csrfInput.name = '_token';↵
38.             csrfInput.value = 'fXRuuNVJIHpLk09Ke1ElHLRm5QkGSEMFhN2y22dF';↵
39.             deleteForm.appendChild(csrfInput);↵
40. ↵
41.             // Add method override input↵
42.             var methodInput = document.createElement('input');↵
43.             methodInput.type = 'hidden';↵
44.             methodInput.name = '_method';↵
45.             methodInput.value = 'DELETE';↵
46.             deleteForm.appendChild(methodInput);↵
47. ↵
48.             // Add answer ID input↵
49.             var answerIdInput = document.createElement('input');↵
50.             answerIdInput.type = 'hidden';↵
51.             answerIdInput.name = 'answer_id';↵
52.             answerIdInput.value = data.answer_id; ↵
53.             deleteForm.appendChild(answerIdInput);↵
54. ↵
55.             // Add delete button↵
56.             var deleteButton = document.createElement('button');↵

```

```

57.         deleteButton.type = 'submit';↵
58.         deleteButton.textContent = 'Delete Answer';↵
59.         deleteButton.addEventListener('click', function () {↵
60.             return confirm('Are you sure you want to delete this
        answer?');↵
61.         });↵
62. ↵
63.         deleteForm.appendChild(deleteButton);↵
64.         answerContainer.appendChild(deleteForm);↵
65. ↵
66.         // Create form for editing the answer↵
67.         var editForm = document.createElement('form');↵
68.         editForm.action =
        'http://localhost:8000/question/1000/answer/REPLACE_WITH_ANSWER_ID/edit'.replac
        e('REPLACE_WITH_ANSWER_ID', data.answer_id);↵
69.         editForm.method = 'GET';↵
70. ↵
71.         // Add edit button↵
72.         var editButton = document.createElement('button');↵
73.         editButton.type = 'submit';↵
74.         editButton.textContent = 'Edit Answer';↵
75. ↵
76.         editForm.appendChild(editButton);↵
77.         answerContainer.appendChild(editForm);↵
78. ↵
79.         // Add mark as correct form and View Comments link↵
80.         if (data.question_user === 1000) {↵
81.             var markAsCorrectForm = document.createElement('form');↵
82.             markAsCorrectForm.action =
        'http://localhost:8000/question/REPLACE_WITH_ANSWER_ID/mark-as-
        correct'.replace('REPLACE_WITH_ANSWER_ID', data.answer_id);↵
83.             markAsCorrectForm.method = 'POST';↵
84. ↵
85.             // Add CSRF token input for mark as correct form↵
86.             var markAsCorrectCsrInput =
        document.createElement('input');↵
87.             markAsCorrectCsrInput.type = 'hidden';↵
88.             markAsCorrectCsrInput.name = '_token';↵
89.             markAsCorrectCsrInput.value =
        'fXRuuNVJIHpLk09Ke1ElHLRm5QkGSEMFhN2y22dF';↵
90.             markAsCorrectForm.appendChild(markAsCorrectCsrInput);↵
91. ↵
92.             // Add hidden input for mark as correct↵
93.             var markAsCorrectInput = document.createElement('input');↵
94.             markAsCorrectInput.type = 'hidden';↵
95.             markAsCorrectInput.name = 'correct';↵
96.             markAsCorrectInput.value = data.is_correct ? 0 : 1;↵
97.             markAsCorrectForm.appendChild(markAsCorrectInput);↵
98. ↵
99.             // Add mark as correct button↵
100.            var markAsCorrectButton =
        document.createElement('button');↵
101.            markAsCorrectButton.type = 'submit';↵
102.            markAsCorrectButton.textContent = data.is_correct ? 'Remove
        Correct' : 'Mark as Correct';↵
103.            ↵
104.            markAsCorrectForm.appendChild(markAsCorrectButton);↵
105.            answerContainer.appendChild(markAsCorrectForm);↵
106.        }↵
107. ↵
108.        var viewCommentsLink = document.createElement('a');↵
109.        viewCommentsLink.className = 'button';↵
110.        viewCommentsLink.href = '/question/' + data.question_id +
        '/answer/' + data.answer_id + '/comments';↵
111.        viewCommentsLink.textContent = 'View Comments';↵
112. ↵
113.        answerContainer.appendChild(viewCommentsLink);↵
114. ↵
115.        answersContainer.appendChild(answerContainer);↵
116. ↵
117.        answerForm.reset();↵

```

```

118.         });↵
119.         .catch(error => {↵
120.             console.error('Error:', error);↵
121.         });↵
122.     });↵
123. });↵
124. </script>↵
125. ↵
126. <!DOCTYPE html>↵
127. <html lang="en">↵
128.     <head>↵
129.         <meta charset="utf-8">↵
130.         <meta http-equiv="X-UA-Compatible" content="IE=edge">↵
131.         <meta name="viewport" content="width=device-width, initial-scale=1">↵
132.     ↵
133.         <!-- CSRF Token -->↵
134.         <meta name="csrf-token"
content="fXRuuNVJIHpLk09Ke1ElHLRm5QkGSEMFhN2y22dF">↵
135.     ↵
136.         <title>BrainShare</title>↵
137.         <link rel='shortcut icon' type='image/x-icon'
href="http://localhost:8000/images/logo.png" />↵
138.     ↵
139.         <!-- Styles -->↵
140.         <link href="http://localhost:8000/css/milligram.min.css"
rel="stylesheet">↵
141.         <link href="http://localhost:8000/css/app.css" rel="stylesheet">↵
142.         <script type="text/javascript">↵
143.             // Fix for Firefox autofocus CSS bug↵
144.             // See: http://stackoverflow.com/questions/18943276/html-5-
autofocus-messes-up-css-loading/18945951#18945951↵
145.         </script>↵
146.         <script type="text/javascript" src=http://localhost:8000/js/app.js
defer>↵
147.         </script>↵
148.     </head>↵
149.     <body>↵
150.         <main>↵
151.             <header>↵
152.                 <nav>↵
153.                     <a href="http://localhost:8000/home"></a>↵
154.                     <h1><a href="http://localhost:8000/home">BrainShare</a></h1>↵
155.                     <ul>↵
156.                         <li><a href="http://localhost:8000/about">About us</a>
</li>↵
157.                         <li><a href="http://localhost:8000/faq">FAQ</a></li>↵
158.                         <li><a
href="http://localhost:8000/contact">Contacts</a></li>↵
159.                         <li><a
href="http

```

Used the HTML parser.

Total execution time 4 milliseconds.

[About this checker](#) • [Report an issue](#) • Version: 23.11.28