

# 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 text input ☐ CSS

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Plataforma IoT</title>
  <link rel="stylesheet" href="style.css">
  <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
QWTKZyjpPEjISv5WaRU90FeRpok6YctnYmDr5pNlyT2bRjXh0JMHjY6hW+ALEwIH"
crossorigin="anonymous">
```

[Check](#)

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

[Message Filtering](#)

**Document checking completed. No errors or warnings to show.**

## Source

```
1.  ↵
2.  <!DOCTYPE html>↵
3.  <html lang="en">↵
4.  ↵
5.  <head>↵
6.    <meta charset="utf-8">↵
7.    <meta name="viewport" content="width=device-width, initial-scale=1">↵
8.    <title>Plataforma IoT</title>↵
9.    <link rel="stylesheet" href="style.css">↵
10.   <link
    href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
    rel="stylesheet" integrity="sha384-
    QWTKZyjpPEjISv5WaRU90FeRpok6YctnYmDr5pNlyT2bRjXh0JMHjY6hW+ALEwIH"
    crossorigin="anonymous">↵
11.  ↵
12. </head>↵
13.  ↵
14. <body>↵
15.  ↵
16.  ↵
```

```

17. <!-- barra de navegação -->
18. <nav class="navbar navbar-expand-lg bg-body-tertiary">
19.   <div class="container-fluid">
20.     <a class="navbar-brand" href="dashboard.php">Veículo</a>
21.     <button class=" navbar-toggler" type="button" data-bs-toggle="collapse"
data-bs-target="#navbarSupportedContent" aria-
controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle
navigation">
22.       <span class="navbar-toggler-icon"></span>
23.     </button>
24.     <div class="collapse navbar-collapse" id="navbarSupportedContent">
25.       <ul class="navbar-nav me-auto mb-2 mb-lg-0">
26.         <li class="nav-item">
27.           <a class="nav-link" href="historico.php">Histórico</a>
28.         </li>
29.       </ul>
30.       <form class="d-flex" action="logout.php" method="post">
31.         <button class="btn btn-outline-success"
type="submit">Logout</button>
32.       </form>
33.     </div>
34.   </div>
35. </nav>
36. <br>
37. <div class="container">
38.   <div class="row text-center">
39.     <div class="col-sm-2 flex-column justify-content-around align-content-
center">
40.       <!-- SENSOR DE TEMPERATURA -->
41.       <div class="card">
42.         <div class="card-header">
43.           <h6>Temperatura</h6>
44.         </div>
45.         <div class="card-body">
46.           <h1 id="temperatura">0</h1>
47.         </div>
48.         <div class="card-footer">
49.           <h6><a href="historico.php?nome=temperatura">Histórico</a></h6>
50.         </div>
51.       </div>
52.       <br>
53.       <!-- SENSOR DE HUMIDADE -->
54.       <div class="card ">
55.         <div class="card-header">
56.           <h6>Humidade</h6>
57.         </div>
58.         <div class="card-body">
59.           <h1 id="humidade">0</h1>
60.         </div>
61.         <div class="card-footer">
62.           <h6><a href="historico.php?nome=humidade">Histórico</a></h6>
63.         </div>
64.       </div>
65.       <hr>
66.       <!-- CONTROLADOR DE ATUADOR DE VENTONHA -->
67.       <div class="card">
68.         <div class="card-header">
69.           <h6>Ventilação</h6>
70.         </div>
71.         <div class="ventoinha d-flex ">
72.           <!-- 0 controlo de ligamento/desligamento da ventoinha deverá ser
desenvolvido -->
73.           Este campo apenas permite configurar os parametros para
controlo externo
74.           Caso o estado do atuador da ventoinha seja alterado, a imagem
é sinalizada de acordo com o estado atual-->
75.           
76.           <!-- envio de pedido post com as configurações -->
77.           <div class=" flex-column justify-content-around align-content-
center">

```

```

78.         <button type="button" class="btn btn-warning btn-sm"
onclick="resetControlador();">RESET</button>↵
79.         <br>↵
80.         <button type="button" class="btn btn-success btn-sm"
onclick="saveControlador();">SAVE</button>↵
81.     </div>↵
82. </div>↵
83. <div class="card-footer">↵
84.     <!-- valores a serem usados para atualizar o controlador -->↵
85.     <div class="d-flex">↵
86.         <div class="col-sm-3">↵
87.             ↵
88.         </div>↵
89.         <div class="col-sm-3">↵
90.             <input type="text" style="width: 30px;"
id="controlo_temperatura">↵
91.         </div>↵
92.         <div class="col-sm-3">↵
93.             ↵
94.         </div>↵
95.         <div class="col-sm-3">↵
96.             <input type="text" style="width: 30px;"
id="controlo_humidade">↵
97.         </div>↵
98.         <div class="col-sm-1">↵
99.         </div>↵
100.     </div>↵
101.     <hr>↵
102.     <h6><a href="historico.php?nome=ventoinha">Histórico</a></h6>↵
103. </div>↵
104. </div>↵
105. </div>↵
106. <div class="col-sm-6 flex-column justify-content-around align-content-
center">↵
107.     <!-- Monitorização de Sensores de Lugares -->↵
108.     <div class="card">↵
109.         <div class="card-header">↵
110.             <h4>Lugares</h4>↵
111.             <hr>↵
112.             <div class="row">↵
113.                 <!--Contadoes de lugares ocupados/disponíveis-->↵
114.                 <div class="col-sm-4">↵
115.                     <h6>Disponiveis</h6>↵
116.                     <h6 id="lugares_livres">0</h6>↵
117.                 </div>↵
118.                 <div class="col-sm-4">↵
119.                     <h6>Ocupados</h6>↵
120.                     <h6 id="lugares_ocupados">0</h6>↵
121.                 </div>↵
122.                 <div class="col-sm-4">↵
123.                     <h6>Lotação</h6>↵
124.                     <h6 id="lugares_total">0</h6>↵
125.                 </div>↵
126.             </div>↵
127.         </div>↵
128.         <div class="card-body">↵
129.             <!-- tabela de lugares↵
130.                 constroi a tabela de lugares interpretando o array $lugares-
->↵
131.                 <table class="table">↵
132.                     <tbody>↵
133.                         <tr><td class="posicao lugar"><a href="historico.php?
nome=lugar-0-0">
</a></td>↵
134.                         <td class="posicao lugar"><a href="historico.php?
nome=lugar-0-1">
</a></td>↵
135.                         <td class="posicao vazio"></td>↵

```

```
136.         <td class="posicao lugar"><a href="historico.php?
nome=lugar-0-3">
</a></td><↵
137.         </tr><tr><td class="posicao lugar"><a href="historico.php?
nome=lugar-1-0">
</a></td><↵
138.         <td class="posicao lugar"><a href="historico.php?
nome=lugar-1-1">
</a></td><↵
139.         <td class="posicao vazio"></td><↵
140.         <td class="posicao lugar"><a href="historico.php?
nome=lugar-1-3">
</a></td><↵
141.         </tr><tr><td class="posicao lugar"><a href="historico.php?
nome=lugar-2-0">
</a></td><↵
142.         <td class="posicao lugar"><a href="historico.php?
nome=lugar-2-1">
</a></td><↵
143.         <td class="posicao vazio"></td><↵
144.         <td class="posicao lugar"><a href="historico.php?
nome=lugar-2-3">
</a></td><↵
145.         </tr><tr><td class="posicao lugar"><a href="historico.php?
nome=lugar-3-0">
</a></td><↵
146.         <td class="posicao lugar"><a href="historico.php?
nome=lugar-3-1">
</a></td><↵
147.         <td class="posicao vazio"></td><↵
148.         <td class="posicao porta">
</td><↵
149.         </tr><tr><td class="posicao lugar"><a href="historico.php?
nome=lugar-4-0">
</a></td><↵
150.         <td class="posicao lugar"><a href="historico.php?
nome=lugar-4-1">
</a></td><↵
151.         <td class="posicao lugar"><a href="historico.php?
nome=lugar-4-2">
</a></td><↵
152.         <td class="posicao lugar"><a href="historico.php?
nome=lugar-4-3">
</a></td><↵
153.         </tr>                                </tbody><↵
154.     </table><↵
155. </div><↵
156. </div><↵
157. </div><↵
158. <div class="col-sm-4 flex-column justify-content-around align-content-
center"><↵
159.     <p class="card-text">Bem vindo user: <b>user</b></p><↵
160.     <!-- Espaço para implementação de video --><↵
161.     <hr><↵
162.     <↵
163.     <hr><↵
164. </div class="atuadores"><↵
```

```

165.         <!--Atuador de Iluminação<
166.             apresenta os modos de iluminação disponíveis<
167.             para cada modo existe um butao com a identificacao distinta para o
mesmo ser desativado de acordo com o modo atual(js)<
168.             a funcao toggleIluminacao() e chamada com parametros distintos
quando e acionado o evento onclick em cada butao--><
169.             <div class="card col-sm-5"><
170.                 <div class="card-header"><
171.                     <h4>Iluminação</h4><
172.                 </div><
173.                 <div class="card-body"><
174.                     <div class="d-flex flex-column justify-content-around align-
content-center" style="height: 35vh;"><
175.                         <button type="button" class="btn btn-primary"
onclick="toggleIluminacao(0);" id="butao_iluminacao_0">OFF</button><
176.                         <button type="button" class="btn btn-primary"
onclick="toggleIluminacao(1);" id="butao_iluminacao_1">Baixa</button><
177.                         <button type="button" class="btn btn-primary"
onclick="toggleIluminacao(2);" id="butao_iluminacao_2">Alta</button><
178.                     </div><
179.                 </div><
180.                 <div class="card-footer"><
181.                     <h6><a href="historico.php?nome=iluminacao">Histórico</a></h6><
182.                 </div><
183.             </div><
184.         <!--Atuador de Porta<
185.             apresenta e alterna o estado das portas<
186.             a informacao e atualizada usando os identificadores estado_portas
,imagem_portas e butao_portas--><
187.             <div class="card col-sm-5"><
188.                 <div class="card-header"><
189.                     <h4>Portas</h4><
190.                 </div><
191.                 <div class="card-body"><
192.                     <div class="justify-content-around align-content-center"
style="height: 35vh;"><
193.                         <h4 id="estado_portas">Fechadas</h4><
194.                         <
195.                         <button type="button" class="btn btn-success"
onclick="togglePortas(1);" id="butao_portas">Abrir</button><
196.                     </div><
197.                 </div><
198.                 <div class="card-footer"><
199.                     <h6><a href="historico.php?nome=portas">Histórico</a></h6><
200.                 </div><
201.             </div><
202.         </div><
203.     </div><
204. </div><
205. <br><
206. <
207. <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js">
</script><
209. <script><
210.     // uso da funcao setInterval() para atualizacao dos componentes da
dashboard<
211.     const intervalSensores = setInterval(updateEstados, 2000);<
212.     updateEstados();<
213.     resetControlador();<
214. <
215.     //função fornecida para criação de timestamp<
216.     function getHora() {<
217.         var dateISO = new Date().toISOString();<
218.         var data0 = dateISO.split('T')[0];<
219.         var data1 = dateISO.split('T')[1];<
220.         var time = data1.split('.')[0];<
221.         var datahora = data0 + " " + time;<
222.         return datahora;<

```

```
223.     }↵
224. ↵
225.     //POST - função para alterar o valor da iluminação, recebe como argumento
o novo estado do atuador↵
226.     function toggleIluminacao(valor) {↵
227.         const data = new URLSearchParams({↵
228.             nome: "iluminacao",↵
229.             valor: valor,↵
230.             hora: getHora()↵
231.         });↵
232.         fetch('./api/api.php', {↵
233.             method: 'POST',↵
234.             headers: {↵
235.                 'Content-Type': 'application/x-www-form-urlencoded'↵
236.             },↵
237.             body: data.toString()↵
238.         })↵
239.     }↵
240. ↵
241.     //POST - função para abertura/fecho de portas, recebe como argumento o
novo estado do atuador↵
242.     function togglePortas(valor) {↵
243.         const data = new URLSearchParams({↵
244.             nome: "portas",↵
245.             valor: valor,↵
246.             hora: getHora()↵
247.         });↵
248.         fetch('./api/api.php', {↵
249.             method: 'POST',↵
250.             headers: {↵
251.                 'Content-Type': 'application/x-www-form-urlencoded'↵
252.             },↵
253.             body: data.toString()↵
254.         })↵
255.     }↵
256. ↵
257.     //POST - função para atualização dos parametros para o controlador da
ventoinha↵
258.     function saveControlador() {↵
259.         const data = new URLSearchParams({↵
260.             nome: "controlador",↵
261.             temperatura: document.getElementById("controlo_temperatura").value,↵
262.             humidade: document.getElementById("controlo_humidade").value↵
263.         });↵
264.         fetch('./api/api.php', {↵
265.             method: 'POST',↵
266.             headers: {↵
267.                 'Content-Type': 'application/x-www-form-urlencoded'↵
268.             },↵
269.             body: data.toString()↵
270.         })↵
271.     }↵
272. ↵
273.     //GET - Atualiza os valores do controlador da ventoinha a partir da API↵
274.     function resetControlador() {↵
275.         fetch("./api/api.php?valor=controlador")↵
276.         .then(response => response.text())↵
277.         .then(data => {↵
278.             var inputTemperatura =
document.getElementById("controlo_temperatura");↵
279.             var inputHumidade = document.getElementById("controlo_humidade");↵
280.             valores_controlador = data.split(";");↵
281.             inputTemperatura.value = valores_controlador[0];↵
282.             inputHumidade.value = valores_controlador[1];↵
283. ↵
284.         })↵
285.         .catch(error => console.error(error));↵
286.     }↵
287. ↵
288.     //Atualiza a dashboard pedindo toda a informação à API↵
289.     function updateEstados() {↵
```



```
290. //Sensor Temperatura↵
291. fetch("./api/api.php?valor=temperatura")↵
292. .then(response => response.text())↵
293. .then(data => document.getElementById("temperatura").innerHTML = data
+ "°C")↵
294. .catch(error => console.error(error));↵
295. ↵
296. //Sensor Humidade↵
297. fetch("./api/api.php?valor=humidade")↵
298. .then(response => response.text())↵
299. .then(data => document.getElementById("humidade").innerHTML = data +
"%")↵
300. .catch(error => console.error(error));↵
301. ↵
302. //Controlador Ventoinha↵
303. fetch("./api/api.php?valor=ventoinha")↵
304. .then(response => response.text())↵
305. .then(data => {↵
306.     var fanImage = document.getElementById('ventoinha');↵
307.     if (data == "1") fanImage.classList.remove('ocupado')↵
308.     else if (data == "0") fanImage.classList.add('ocupado')↵
309. })↵
310. .catch(error => console.error(error));↵
311. ↵
312. //Atuador Iluminacao↵
313. fetch("./api/api.php?valor=iluminacao")↵
314. .then(response => response.text())↵
315. .then(data => {↵
316.     const styleSheet = document.styleSheets[0];↵
317.     const ruleIndex = Array.from(styleSheet.cssRules).findIndex(rule =>
rule.selectorText === ".posicao");↵
318.     var b0 = document.getElementById('butao_iluminacao_0');↵
319.     var b1 = document.getElementById('butao_iluminacao_1');↵
320.     var b2 = document.getElementById('butao_iluminacao_2');↵
321.     switch (data) {↵
322.         case "0":↵
323.             (styleSheet.cssRules[ruleIndex]).style.setProperty("background-
color", "rgba(255, 255, 0, 0)", "important");↵
324.             b0.disabled = true;↵
325.             b1.disabled = false;↵
326.             b2.disabled = false;↵
327.             break;↵
328.         case "1":↵
329.             (styleSheet.cssRules[ruleIndex]).style.setProperty("background-
color", "rgba(255, 255, 0, 0.20)", "important");↵
330.             b0.disabled = false;↵
331.             b1.disabled = true;↵
332.             b2.disabled = false;↵
333.             break;↵
334.         case "2":↵
335.             (styleSheet.cssRules[ruleIndex]).style.setProperty("background-
color", "rgba(255, 255, 0, 0.40)", "important");↵
336.             b0.disabled = false;↵
337.             b1.disabled = false;↵
338.             b2.disabled = true;↵
339.             break;↵
340.         }↵
341.     })↵
342. .catch(error => console.error(error));↵
343. ↵
344. //Atuador Portas↵
345. fetch("./api/api.php?valor=portas")↵
346. .then(response => response.text())↵
347. .then(data => {↵
348.     const styleSheet = document.styleSheets[0];↵
349.     const ruleIndex = Array.from(styleSheet.cssRules).findIndex(rule =>
rule.selectorText === ".estado_porta");↵
350.     var imagem = document.getElementById("imagem_portas");↵
351.     var butao = document.getElementById('butao_portas');↵
352.     if (data == "0") {↵
353.         imagem.src = "imagens/abrir_portas.png";↵
```

```

354.         butao.classList.remove('btn-danger');↵
355.         butao.classList.add('btn-success');↵
356.         butao.innerHTML = "Abrir";↵
357.         (styleSheet.cssRules[ruleIndex]).style.setProperty("visibility",
"visible");↵
358.         butao.onclick = function() {↵
359.             togglePortas(1);↵
360.         }↵
361.     } else if (data == "1") {↵
362.         imagem.src = "imagens/fechar_portas.png";↵
363.         butao.classList.remove('btn-success');↵
364.         butao.classList.add('btn-danger');↵
365.         butao.innerHTML = "Fechar";↵
366.         (styleSheet.cssRules[ruleIndex]).style.setProperty("visibility",
"hidden");↵
367.         butao.onclick = function() {↵
368.             togglePortas(0);↵
369.         }↵
370.     }↵
371. })↵
372.     .catch(error => console.error(error));↵
373. ↵
374.     /*Atualização dos lugares ocupados↵
375.     volta a pedir o array de lugares à API ↵
376.     avalia se os lugares estão ocupados para atribuir/remover a classe
correspondente↵
377.     */↵
378.     fetch("./api/api.php?valor=lugares")↵
379.     .then(response => response.text())↵
380.     .then(data => {↵
381.         const lugares = JSON.parse(data);↵
382.         var lugaresOcupados = 0;↵
383.         var lugaresLivres = 0;↵
384.         var elementId = ""↵
385.         var X = Y = 0;↵
386.         lugares.forEach(function(linha) {↵
387.             Y = 0;↵
388.             linha.forEach(function(lugar) {↵
389.                 //identificador único↵
390.                 elementId = 'posicao-' + X + '-' + Y;↵
391.                 if (lugar < 0) {↵
392.                     lugaresOcupados++;↵
393.                     document.getElementById(elementId).classList.add("ocupado");↵
394.                 } else if (lugar != 0 && lugar != 9) {↵
395.                     lugaresLivres++;↵
396.                 }
document.getElementById(elementId).classList.remove('ocupado');↵
397.                 }↵
398.                 Y++;↵
399.             })↵
400.             X++;↵
401.         });↵
402.         //atualiza a informação dos contadores de ligares
ocupados/disponiveis↵
403.         document.getElementById("lugares_livres").innerHTML =
lugaresLivres;↵
404.         document.getElementById("lugares_ocupados").innerHTML =
lugaresOcupados;↵
405.         document.getElementById("lugares_total").innerHTML = lugaresLivres +
lugaresOcupados;↵
406.     })↵
407.     .catch(error => console.error(error));↵
408. }↵
409. </script>↵
410. ↵
411. </body>↵
412. ↵
413. </html>

```

Used the HTML parser.



Total execution time 10 milliseconds.

---

[About this checker](#) • [Report an issue](#) • Version: 24.4.22