

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.jsdelivrivr.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.  ↵
3.  <!DOCTYPE html>↵
4.  <html lang="en">↵
5.  ↵
6.  <head>↵
7.    <meta charset="utf-8">↵
8.    <meta name="viewport" content="width=device-width, initial-scale=1">↵
9.    <title>Plataforma IoT</title>↵
10.   <link rel="stylesheet" href="style.css">↵
11.   <link
href="https://cdn.jsdelivrivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
QWTKZyjpPEjISv5WaRU90FeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH"
crossorigin="anonymous">↵
12.  ↵
13. </head>↵
14.  ↵
15. <body>↵
16.   <!-- barra de navegação -->↵
```

```

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

```

```

78.         <br><br>
79.         <button type="button" class="btn btn-success btn-sm"
onclick="saveControlador();">SAVE</button><br>
80.     </div><br>
81. </div><br>
82.     <div class="card-footer"><br>
83.         <!-- valores a serem usados para atualizar o controlador --><br>
84.         <div class="d-flex"><br>
85.             <div class="col-sm-3"><br>
86.                 <br>
87.             </div><br>
88.             <div class="col-sm-3"><br>
89.                 <input type="text" style="width: 30px;"
id="controlelo_temperatura"><br>
90.             </div><br>
91.             <div class="col-sm-3"><br>
92.                 <br>
93.             </div><br>
94.             <div class="col-sm-3"><br>
95.                 <input type="text" style="width: 30px;"
id="controlelo_humidade"><br>
96.             </div><br>
97.             <div class="col-sm-1"><br>
98.             </div><br>
99.         </div><br>
100.        <hr><br>
101.        <h6><a href="historico.php?nome=ventoinha">Histórico</a></h6><br>
102.    </div><br>
103. </div><br>
104. </div><br>
105. <div class="col-sm-6 flex-column justify-content-around align-content-
center"><br>
106.     <!-- Monitorização de Sensores de Lugares --><br>
107.     <div class="card"><br>
108.         <div class="card-header"><br>
109.             <h4>Lugares</h4><br>
110.             <hr><br>
111.             <div class="row"><br>
112.                 <!--Contadoes de lugares ocupados/disponíveis--><br>
113.                 <div class="col-sm-4"><br>
114.                     <h6>Disponiveis</h6><br>
115.                     <h6 id="lugares_livres">0</h6><br>
116.                 </div><br>
117.                 <div class="col-sm-4"><br>
118.                     <h6>Ocupados</h6><br>
119.                     <h6 id="lugares_ocupados">0</h6><br>
120.                 </div><br>
121.                 <div class="col-sm-4"><br>
122.                     <h6>Lotação</h6><br>
123.                     <h6 id="lugares_total">0</h6><br>
124.                 </div><br>
125.             </div><br>
126.         </div><br>
127.         <div class="card-body"><br>
128.             <!-- tabela de lugares<br>
129.                 constroi a tabela de lugares interpretando o array $lugares-
-><br>
130.                 <table class="table"><br>
131.                     <tbody><br>
132.                         <tr><td class="posicao lugar"><a href="historico.php?
nome=lugar-0-0">
</a></td><br>
133.                         <td class="posicao lugar"><a href="historico.php?
nome=lugar-0-1">
</a></td><br>
134.                         <td class="posicao vazio"></td><br>
135.                         <td class="posicao lugar"><a href="historico.php?
nome=lugar-0-3">
</a></td><↵
136.         </tr><tr><td class="posicao lugar"><a href="historico.php?
nome=lugar-1-0">
</a></td><↵
137.         <td class="posicao lugar"><a href="historico.php?
nome=lugar-1-1">
</a></td><↵
138.         <td class="posicao vazio"></td><↵
139.         <td class="posicao lugar"><a href="historico.php?
nome=lugar-1-3">
</a></td><↵
140.         </tr><tr><td class="posicao lugar"><a href="historico.php?
nome=lugar-2-0">
</a></td><↵
141.         <td class="posicao lugar"><a href="historico.php?
nome=lugar-2-1">
</a></td><↵
142.         <td class="posicao vazio"></td><↵
143.         <td class="posicao lugar"><a href="historico.php?
nome=lugar-2-3">
</a></td><↵
144.         </tr><tr><td class="posicao lugar"><a href="historico.php?
nome=lugar-3-0">
</a></td><↵
145.         <td class="posicao lugar"><a href="historico.php?
nome=lugar-3-1">
</a></td><↵
146.         <td class="posicao vazio"></td><↵
147.         <td class="posicao porta">
</td><↵
148.         </tr><tr><td class="posicao lugar"><a href="historico.php?
nome=lugar-4-0">
</a></td><↵
149.         <td class="posicao lugar"><a href="historico.php?
nome=lugar-4-1">
</a></td><↵
150.         <td class="posicao lugar"><a href="historico.php?
nome=lugar-4-2">
</a></td><↵
151.         <td class="posicao lugar"><a href="historico.php?
nome=lugar-4-3">
</a></td><↵
152.         </tr>                                </tbody><↵
153.     </table><↵
154. </div><↵
155. </div><↵
156. </div><↵
157. <div class="col-sm-4 flex-column justify-content-around align-content-
center"><↵
158.     <p class="card-text">Bem vindo user: <b>teste</b></p><↵
159.     <!-- Espaço para implementação de video --><↵
160.     <hr><↵
161.     <↵
162.     <hr><↵
163.     <div class="atuadores"><↵
164.         <!--Atuador de Iluminação<↵
165.         apresenta os modos de iluminação disponíveis<↵

```

```

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

```

```

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

```



```

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

```

```

356.         document.getElementById("estado_portas").innerHTML = "Fechadas";↵
357.         butao.onclick = function() {↵
358.             togglePortas(1);↵
359.         }↵
360.     } else if (data == "1") {↵
361.         imagem.src = "imagens/fechar_portas.png";↵
362.         butao.classList.remove('btn-success');↵
363.         butao.classList.add('btn-danger');↵
364.         butao.innerHTML = "Fechar";↵
365.         (styleSheet.cssRules[ruleIndex]).style.setProperty("visibility",
"hidden");↵
366.         document.getElementById("estado_portas").innerHTML = "Abertas";↵
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.                 }
397.                 document.getElementById(elementId).classList.remove('ocupado');↵
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.

