

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.  ↵
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.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
QWTKZyjpPEjISv5WaRU90FeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH"
crossorigin="anonymous">↵
12. </head>↵
13. ↵
14. <body>↵
15.   <!-- barra de navegação -->↵
16.   <nav class="navbar navbar-expand-lg bg-body-tertiary">↵
```

```

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

```

```

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

```

```

135.         </tr><tr><td class="posicao lugar"><a href="historico.php?
nome=lugar-1-0">
</a></td><
136.         <td class="posicao lugar"><a href="historico.php?
nome=lugar-1-1">
</a></td><
137.         <td class="posicao vazio"></td><
138.         <td class="posicao lugar"><a href="historico.php?
nome=lugar-1-3">
</a></td><
139.     </tr><tr><td class="posicao lugar"><a href="historico.php?
nome=lugar-2-0">
</a></td><
140.     <td class="posicao lugar"><a href="historico.php?
nome=lugar-2-1">
</a></td><
141.     <td class="posicao vazio"></td><
142.     <td class="posicao lugar"><a href="historico.php?
nome=lugar-2-3">
</a></td><
143.     </tr><tr><td class="posicao lugar"><a href="historico.php?
nome=lugar-3-0">
</a></td><
144.     <td class="posicao lugar"><a href="historico.php?
nome=lugar-3-1">
</a></td><
145.     <td class="posicao vazio"></td><
146.     <td class="posicao porta">
</td><
147.     </tr><tr><td class="posicao lugar"><a href="historico.php?
nome=lugar-4-0">
</a></td><
148.     <td class="posicao lugar"><a href="historico.php?
nome=lugar-4-1">
</a></td><
149.     <td class="posicao lugar"><a href="historico.php?
nome=lugar-4-2">
</a></td><
150.     <td class="posicao lugar"><a href="historico.php?
nome=lugar-4-3">
</a></td><
151.     </tr>                                </tbody><
152. </table><
153. </div><
154. </div><
155. </div><
156. <div class="col-sm-4 flex-column justify-content-around align-content-
center"><
157.     <p class="card-text">Bem vindo user: <b>teste</b></p><
158.     <!-- Espaço para implementação de video --><
159.     <hr><
160.     <
161.     <hr><
162.     <div class="atuadores"><
163.         <!--Atuador de Iluminação<
164.             apresenta os modos de iluminação disponíveis<
165.             para cada modo existe um botão com a identificação distinta para o
mesmo ser desativado de acordo com o modo atual (js)<

```

```

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

```

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



```
290.         .then(data => document.getElementById("temperatura").innerHTML = data
+ "°C")↵
291.         .catch(error => console.error(error));↵
292.     ↵
293.     //Sensor Humidade - atualiza o campo de texto com o valor da humidade↵
294.     fetch("./api/api.php?valor=humidade")↵
295.         .then(response => response.text())↵
296.         .then(data => document.getElementById("humidade").innerHTML = data +
"%")↵
297.         .catch(error => console.error(error));↵
298.     ↵
299.     //Controlador Ventoinha - adiciona ou remove a class 'ocupado' à imagem
da ventoinha↵
300.     fetch("./api/api.php?valor=ventoinha")↵
301.         .then(response => response.text())↵
302.         .then(data => {↵
303.             var fanImage = document.getElementById('ventoinha');↵
304.             if (data == "1") fanImage.classList.remove('ocupado')↵
305.             else if (data == "0") fanImage.classList.add('ocupado')↵
306.         })↵
307.         .catch(error => console.error(error));↵
308.     ↵
309.     //Atuador Iluminacao - altera a cor de fundo (já associado às células da
tabela de lugares) de acordo com o nível de luminosidade↵
310.     // - coloca o botão correspondente como desativo, e os
restantes como ativos↵
311.     fetch("./api/api.php?valor=iluminacao")↵
312.         .then(response => response.text())↵
313.         .then(data => {↵
314.             const styleSheet = document.styleSheets[0];↵
315.             const ruleIndex = Array.from(styleSheet.cssRules).findIndex(rule =>
rule.selectorText === ".posicao");↵
316.             var b0 = document.getElementById('butao_iluminacao_0');↵
317.             var b1 = document.getElementById('butao_iluminacao_1');↵
318.             var b2 = document.getElementById('butao_iluminacao_2');↵
319.             switch (data) {↵
320.                 case "0":↵
321.                     (styleSheet.cssRules[ruleIndex]).style.setProperty("background-
color", "rgba(255, 255, 0, 0)", "important");↵
322.                     b0.disabled = true;↵
323.                     b1.disabled = false;↵
324.                     b2.disabled = false;↵
325.                     break;↵
326.                 case "1":↵
327.                     (styleSheet.cssRules[ruleIndex]).style.setProperty("background-
color", "rgba(255, 255, 0, 0.20)", "important");↵
328.                     b0.disabled = false;↵
329.                     b1.disabled = true;↵
330.                     b2.disabled = false;↵
331.                     break;↵
332.                 case "2":↵
333.                     (styleSheet.cssRules[ruleIndex]).style.setProperty("background-
color", "rgba(255, 255, 0, 0.40)", "important");↵
334.                     b0.disabled = false;↵
335.                     b1.disabled = false;↵
336.                     b2.disabled = true;↵
337.                     break;↵
338.             }↵
339.         })↵
340.         .catch(error => console.error(error));↵
341.     ↵
342.     //Atuador Portas - altera a imagem com o id "imagem_portas" (fechadas /
abertas)↵
343.     // - altera a classe do botão de controlo de portas
(success / danger)↵
344.     // - adiciona ou remove a class 'visible' à imagem da
porta na tabela de lugares↵
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.             document.getElementById("estado_portas").innerHTML = "Fechadas";↵
359.             butao.onclick = function() {↵
360.                 togglePortas(1);↵
361.             }↵
362.         } else if (data == "1") {↵
363.             imagem.src = "imagens/fechar_portas.png";↵
364.             butao.classList.remove('btn-success');↵
365.             butao.classList.add('btn-danger');↵
366.             butao.innerHTML = "Fechar";↵
367.             (stylesheet.cssRules[ruleIndex]).style.setProperty("visibility",
"hidden");↵
368.             document.getElementById("estado_portas").innerHTML = "Abertas";↵
369.             butao.onclick = function() {↵
370.                 togglePortas(0);↵
371.             }↵
372.         }↵
373.     })↵
374.     .catch(error => console.error(error));↵
375. ↵
376.     //Atualização dos lugares ocupados↵
377.     // - volta a pedir o array de lugares à API ↵
378. ↵
379.     fetch("./api/api.php?valor=lugares")↵
380.     .then(response => response.text())↵
381.     .then(data => {↵
382.         const lugares = JSON.parse(data);↵
383.         var lugaresOcupados = 0;↵
384.         var lugaresLivres = 0;↵
385.         var elementId = "";↵
386.         var X = Y = 0;↵
387.         lugares.forEach(function(linha) {↵
388.             Y = 0;↵
389.             linha.forEach(function(lugar) {↵
390.                 //identificador único de lugar↵
391.                 elementId = 'posicao-' + X + '-' + Y;↵
392.                 if (lugar < 0) {↵
393.                     // avalia se os lugares estão ocupados (valor negativo) para
aribuir/remover a classe correspondente↵
394.                     lugaresOcupados++;↵
395.                     document.getElementById(elementId).classList.add("ocupado");↵
396.                 } else if (lugar != 0 && lugar != 9) {↵
397.                     // avalia se não é um espaço vazio ou uma porta↵
398.                     lugaresLivres++;↵
399.
document.getElementById(elementId).classList.remove('ocupado');↵
400.                 }↵
401.                 Y++;↵
402.             })↵
403.             X++;↵
404.         });↵
405.         //atualiza a informação dos contadores de ligares
ocupados/disponiveis↵
406.         document.getElementById("lugares_livres").innerHTML =
lugaresLivres;↵
407.         document.getElementById("lugares_ocupados").innerHTML =
lugaresOcupados;↵
408.         document.getElementById("lugares_total").innerHTML = lugaresLivres +
lugaresOcupados;↵
409.     })↵
410.     .catch(error => console.error(error));↵

```



```
411.      }↵
412.    </script>↵
413.  ↵
414. </body>↵
415.  ↵
416. </html>
```

Used the HTML parser.

Total execution time 11 milliseconds.

[About this checker](#) • [Report an issue](#) • Version: 24.4.15