

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

```

        }
      })
      .catch(error => console.error(error));

//Atuador Portas
fetch("./api/api.php?valor=portas")
  .then(response => response.text())
  .then(data => {
    const styleSheet = document.styleSheets[0];
    const ruleIndex = Array.from(styleSheet.cssRules).findIndex(rule =>
rule.selectorText === ".porta");
    var imagem = document.getElementById("imagem_portas");
    var butao = document.getElementById('butao_portas');
    if (data == "0") {
      imagem.src = "imagens/abrir_portas.png";
    }
  });

```

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-
      QWTKZyjpPEjISv5WaRU90FeRpok6YctnYmDr5pNlyT2bRjXh0JMhY6hW+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. ↵
37. <div class="container">↵
38.   <div class="row text-center">↵
39.     <div class="col-sm-2">↵
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.       <!-- 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.         <!---0 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>↵

```

```

79.         <button type="button" class="btn btn-success btn-sm"
onclick="saveControlador();">SAVE</button>↵
80.     </div>↵
81. </div>↵
82. <div class="card-footer">↵
83.     <!-- valores a serem usados para atualizar o controlador -->↵
84.     <div class="d-flex">↵
85.         <div class="col-sm-3">↵
86.             ↵
87.         </div>↵
88.         <div class="col-sm-3">↵
89.             <input type="text" style="width: 30px;"
id="controlelo_temperatura">↵
90.         </div>↵
91.         <div class="col-sm-3">↵
92.             ↵
93.         </div>↵
94.         <div class="col-sm-3">↵
95.             <input type="text" style="width: 30px;"
id="controlelo_humidade">↵
96.         </div>↵
97.         <div class="col-sm-1">↵
98.         </div>↵
99.     </div>↵
100.    <hr>↵
101.    <h6><a href="historico.php?nome=ventoinha">Histórico</a></h6>↵
102. </div>↵
103. </div>↵
104. </div>↵
105. <div class="col-sm-6">↵
106.     <!-- Monitorização de Sensores de Lugares -->↵
107.     <div class="card">↵
108.         <div class="card-header">↵
109.             <h4>Lugares</h4>↵
110.             <hr>↵
111.             <div class="row">↵
112.                 <!-- Contadores de lugares ocupados/disponíveis -->↵
113.                 <div class="col-sm-4">↵
114.                     <h6>Disponíveis</h6>↵
115.                     <h6 id="lugares_livres">0</h6>↵
116.                 </div>↵
117.                 <div class="col-sm-4">↵
118.                     <h6>Ocupados</h6>↵
119.                     <h6 id="lugares_ocupados">0</h6>↵
120.                 </div>↵
121.                 <div class="col-sm-4">↵
122.                     <h6>Lotação</h6>↵
123.                     <h6 id="lugares_total">0</h6>↵
124.                 </div>↵
125.             </div>↵
126.         </div>↵
127.         <div class="card-body">↵
128.             <!-- tabela de lugares↵
129.                 constroi a tabela de lugares interpretando o array $lugares-
->↵
130.             <table class="table">↵
131.                 <tbody>↵
132.                     <tr><td class="posicao lugar"><a href="historico.php?
nome=lugar-0-0">
</a></td>↵
133.                     <td class="posicao lugar"><a href="historico.php?
nome=lugar-0-1">
</a></td>↵
134.                     <td class="posicao vazio"></td>↵
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"><
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 butao com a identificacao 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↵
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. ↵
205.     <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js">
</script>↵
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.         //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. //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("controleo_temperatura").value,↵
259.         humidade: document.getElementById("controleo_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("controleo_temperatura");↵
276.         var inputHumidade = document.getElementById("controleo_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 à API↵
286. function updateEstados() {↵
287.     //Sensor 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↵
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↵
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↵
310. fetch("./api/api.php?valor=iluminacao")↵
311. .then(response => response.text())↵
312. .then(data => {↵
313.   const styleSheet = document.styleSheets[0];↵
314.   const ruleIndex = Array.from(styleSheet.cssRules).findIndex(rule =>
rule.selectorText === ".posicao");↵
315.   var b0 = document.getElementById('butao_iluminacao_0');↵
316.   var b1 = document.getElementById('butao_iluminacao_1');↵
317.   var b2 = document.getElementById('butao_iluminacao_2');↵
318.   switch (data) {↵
319.     case "0":↵
320.       (styleSheet.cssRules[ruleIndex]).style.setProperty("background-
color", "rgba(255, 255, 0, 0)", "important");↵
321.       b0.disabled = true;↵
322.       b1.disabled = false;↵
323.       b2.disabled = false;↵
324.       break;↵
325.     case "1":↵
326.       (styleSheet.cssRules[ruleIndex]).style.setProperty("background-
color", "rgba(255, 255, 0, 0.20)", "important");↵
327.       b0.disabled = false;↵
328.       b1.disabled = true;↵
329.       b2.disabled = false;↵
330.       break;↵
331.     case "2":↵
332.       (styleSheet.cssRules[ruleIndex]).style.setProperty("background-
color", "rgba(255, 255, 0, 0.40)", "important");↵
333.       b0.disabled = false;↵
334.       b1.disabled = false;↵
335.       b2.disabled = true;↵
336.       break;↵
337.     }↵
338.   })↵
339. .catch(error => console.error(error));↵
340. ↵
341. //Atuador Portas↵
342. fetch("./api/api.php?valor=portas")↵
343. .then(response => response.text())↵
344. .then(data => {↵
345.   const styleSheet = document.styleSheets[0];↵
346.   const ruleIndex = Array.from(styleSheet.cssRules).findIndex(rule =>
rule.selectorText === ".porta");↵
347.   var imagem = document.getElementById("imagem_portas");↵
348.   var butao = document.getElementById('butao_portas');↵
349.   if (data == "0") {↵
350.     imagem.src = "imagens/abrir_portas.png";↵
351.     butao.classList.remove('btn-danger');↵
352.     butao.classList.add('btn-success');↵
353.     butao.innerHTML = "Abrir";↵
354.     (styleSheet.cssRules[ruleIndex]).style.setProperty("visibility",
"visible");↵
355.     butao.onclick = function() {↵
356.       togglePortas(1);↵
```

```

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

```

Used the HTML parser.

Total execution time 10 milliseconds.

