

```

// Função para atualizar os dados na página
function updateData(temperature, humidity) {
  document.getElementById("temperature").textContent = temperature;
  document.getElementById("humidity").textContent = humidity;
}

//make the chart with the data from the api
function makeChart() {
  const ctx = document.getElementById('myChart').getContext('2d');
  const chart = new Chart(ctx, {
    type: 'line',
    data: {
      labels: [],
      datasets: [{
        label: 'Temperatura',
        backgroundColor: 'rgb(255, 99, 132)',
        borderColor: 'rgb(255, 99, 132)',
        data: []
      }]
    },
    options: {}
  });
  return chart;
}

// Função para realizar uma requisição GET e atualizar os dados na página
function fetchData() {
  fetch("https://api.thingspeak.com/channels/2317881/feeds.json?api_key=MAU6RGEX5HD66KPF&results=2").then(response => response.json()).then(data => {
    const entry = data.feeds[0];
    const temperature = parseFloat(entry.field1);
    const humidity = parseFloat(entry.field2).toFixed(1); // Manter uma casa decimal
    updateData(temperature + '°C', humidity + '%');
    console.log(temperature);
  })
}

```

```

1 document.getElementById("input_sel_arq").addEventListener("change", function (event) {
2   const file = event.target.files[0];
3
4   if (file) {
5     const reader = new FileReader();
6
7     reader.onload = function (e) {
8       const fileContent = e.target.result;
9       const rows = fileContent.split('\n');
10
11       const table = document.getElementById("tab_geral");
12       table.innerHTML = ""; // Clear existing table
13
14       // Add custom row
15       const customRow = table.insertRow(-1);
16       const cell1 = customRow.insertCell(0);
17       const cell2 = customRow.insertCell(1);
18       cell1.textContent = "Temperatura (°C)";
19       cell2.textContent = "Umidade (%)";
20
21       for (let i = 0; i < rows.length; i++) {
22         const row = table.insertRow(-1);
23         const columns = rows[i].split(';'); // Separate columns using ';'
24
25         for (let j = 0; j < columns.length; j++) {
26           const cell = row.insertCell(-1);
27           cell.textContent = columns[j].trim(); // Trim leading and trailing whitespace
28         }
29       }
30     };
31
32     reader.readAsText(file);
33   }
34 });

```

```

1 // CRIAR OBJETOS E VARIÁVEIS
2 const obj_body = document.querySelector('body')
3 const obj_exibe_data = document.querySelector('#exibe_data')
4
5 const obj_input_data_ini = document.querySelector('#input_data_ini')
6 const obj_input_data_fim = document.querySelector('#input_data_fim')
7
8 // Formato de Data do Firefox: YYYY/MM/DD
9 let data_ini = new Date('9999/12/31')
10 let data_fim = new Date('0001/01/01')
11
12 // EVENTOS
13 obj_body.addEventListener('load', funPreencherData(), funLimparFormularios())
14
15 obj_link_importar.addEventListener('click', function() { funExibir('#imp'); funTelaResImp() } )
16 obj_link_saldo.addEventListener('click', function() { funExibir('#saldo'); funTelaSaldo() } )
17 obj_link_home.addEventListener('click', function() { funEsconderTudo(); funTelaHome() } )
18
19 obj_input_sel_arq.addEventListener('change', funLerArquivoLocal)
20 conteudo_arq.addEventListener('load', funCarregarArquivoLocal)
21
22 obj_sel_conta_saldo.addEventListener('change', funCalcularSaldo)
23
24 obj_bt_extrato.addEventListener('click', funValidarCampos)
25
26 obj_bt_car_arq_nuv.addEventListener('click', funLerArquivoNuvem)
27 arq_nuvem.addEventListener('readystatechange', funCarregarArquivoNuvem)
28
29
30 // FUNÇÕES
31 function funPreencherData() {
32     let agora = new Date
33     obj_exibe_data.innerHTML = agora.toLocaleDateString('pt-br')
34 }
35
36 function funTelaHome() {
37     obj_home.setAttribute('class', 'exibe')
38 }
39
40 function funTelaHome() {
41     obj_home.setAttribute('class', 'exibe')
42 }
43
44 function funTratamentoDadas() {
45     for ( var i = 0; i < dados_arquivo.length; i++ ) {
46         if ( dados_arquivo[i]['data'] < data_ini ) {
47             data_ini = dados_arquivo[i]['data']
48         }
49         if ( dados_arquivo[i]['data'] > data_fim ) {
50             data_fim = dados_arquivo[i]['data']
51         }
52     }
53 }
54
55 // Conversão para o mesmo formato de data do input: YYYY-MM-DD (ISO)
56 let converte = data_ini.toISOString().slice(0, 10)
57 obj_input_data_ini.setAttribute('value', converte)
58 obj_input_data_ini.setAttribute('min', converte)
59 obj_input_data_fim.setAttribute('min', converte)
60
61 converte = data_fim.toISOString().slice(0, 10)
62 obj_input_data_fim.setAttribute('value', converte)
63 obj_input_data_ini.setAttribute('max', converte)
64 obj_input_data_fim.setAttribute('max', converte)
65 }

```