|  |  |  |
| --- | --- | --- |
| **HTML** | | |
| <!DOCTYPE html> | **é uma declaração que define o tipo de documento como HTML5. É usada para indicar ao navegador qual versão do HTML está sendo utilizada no código.** | <!DOCTYPE html>  <html>  <body>  </body>  </html> |
| <html></html> | **É utilizado para abrir e fechar um código HTML** | <html>  </html> |
| lang=”” | **É oque vai dizer a linguagem que a página vai ter** | <html lang=”pt-BR“>  </html> |
| <title> </title> | **Para colocar título na aba** | <title>Comandos basicos</title> |
| <h1> </h1> | **Serve para colocar algo escrito em letra grande no topo da página** | <h1> Comandos basicos</h1> |
| <p></p> | **Cria um paragrafo** | <p>grande parágrafo lindo</p> |
| <script></script> | **Serve para começar a parte de Javascript** | <script>document.write("Olá mundo");  alert("Olá mundo");console.log("Oi");</script> |
| <br> | **É um comando que pode ser usado para pular linha** | document.write(<br>); |
| <meta> | **A tag <meta> no HTML fornece metadados essenciais sobre um documento, como conjunto de caracteres, descrição e palavras-chave. Essas informações ajudam a descrever e definir o comportamento da página.** | <meta charset="UTF-8">  <meta http-equiv="X-UA-Compatible" content="IE=edge">  <meta name="viewport" content="width=device-width, initial-scale=1.0"> |
| charset=”” | **O atributo charset define o conjunto de caracteres usado no documento HTML para exibir corretamente os caracteres especiais.** | <meta charset="UTF-8"> |
| http-equiv=”” | **é usado para fornecer informações adicionais ao navegador ou ao servidor HTTP. Ele permite especificar metadados sobre a forma como o navegador deve tratar a página ou a solicitação HTTP.** | <meta http-equiv="X-UA-Compatible"> |
| content="" | **é usado para especificar o valor associado a uma propriedade ou diretiva. Ele fornece informações adicionais ou dados relevantes para o atributo específico em questão.** | <meta http-equiv="X-UA-Compatible" contente="IE=edge"> |
| name=”” | **é usado para definir um nome ou identificador para o conteúdo associado. Ele fornece um rótulo ou uma chave para especificar o propósito ou o tipo de informação contida no elemento <meta>. O valor do atributo name varia dependendo do contexto e do propósito do elemento <meta>.** | <meta name="viewport" content="width=device-width, initial-scale=1.0"> |
| <head></head> | **é uma seção do documento HTML que contém metadados e informações sobre o documento, como links para arquivos CSS, scripts JavaScript, título da página, codificação de caracteres, entre outros elementos não visíveis ao usuário** | <!DOCTYPE html>  <html>  <head>  </head>  <body>  </body>  </html> |
| <style></syle> | **É aqui que entra todo o css pagina quando é colocado em head** | <style>background-color:blue;</style> |
| <body></body> | **É aqui que vem as informações que ficam na tela como h1,br e p** | <body><h1>titulo</h1></body> |
| <script></script> | **É o lugar aonde é colocado os comandos de HTML** | <script>console.log(‘Olá mundo’)</script> |
| <Strong></Strong> | **Deixa em negrito a palavra** | <Strong>Olá mundo</Strong> |
| Id=”” | **O id é algo único que serve para ser chamado no Javascript ou no CSS** | id= "msg" |
| <div></div> | **Serve para organizar uma pagina dividindo ela em divs** | <div id= "msg"> Clique em mim</div> |
| onclick=" " | **Comando acionado quando há clique do mouse a partir do HTML** | onclick="clique()" |
| id | **Comando acionado quando o ponteiro do mouse esta em cima do elemento** | onmouseenter=” dentro()” |
| onmouseout=" " | **Comando acionado quando o ponteiro do mouse esta fora do elemento** | onmouseout=” fora()” |
| <input> | **Ele cria elementos que servem para interatividade com caixa de texto, botões etc.** | <input type="button" value="verificar" onclick="calcular()"> |
| <link> | **Esse elemento faz uma conexão outro documento ou uma url** | <link rel="stylesheet" href="estilo.css"> |
| rel=”” | **Ele serve para nomear e vincula o arquivo com um nome** | rel="stylesheet":, rel="icon":, rel="preconnect, rel="nofollow" |
| href=”” | **é usado para especificar o destino de um link em um elemento HTML, como um <a> (link) ou <link> (referência de estilo). O valor do atributo href é uma URL (Uniform Resource Locator) que indica o endereço para onde o link deve direcionar quando clicado.** | <link rel="stylesheet" href="estilo.css"> |
| <header> </header> | **é usada para representar o cabeçalho de um documento HTML ou de uma seção específica de um documento. Geralmente, ela contém elementos como o logotipo, o título da página, menus de navegação e outros elementos relacionados ao cabeçalho do conteúdo.** | <header>  <h1>Titulo </h1>  </header>  \*elemento de organização\* |
| <section></section> | **é usada para definir uma seção ou uma parte específica de um documento HTML. Ela é utilizada para agrupar conteúdos relacionados que podem ser considerados independentes do restante do documento. Pode ser usada para dividir o conteúdo do documento em seções distintas, como capítulos, artigos, blocos de texto, galerias de imagens, entre outros.** | <section>  <div>  Testando...  </div>  <div>  Ola!  </div>  </section> |
| <footer></footer> | **é utilizada para representar a seção de rodapé de um documento HTML. Ela normalmente contém informações adicionais, como informações de contato, direitos autorais, links de navegação secundários ou qualquer outro conteúdo relevante para o final da página.** | <footer>  <p>&copy; NinjaExtreme</p>  </footer> |
| src=”" | **é importante para indicar a localização do recurso que será exibido ou executado no documento HTML. O valor do src pode ser uma URL, um caminho relativo ou um caminho absoluto, dependendo do contexto em que está sendo utilizado.** | <script src="script.js"></script> |
| onload=”” | **é um evento em JavaScript e HTML que é acionado quando um recurso, como uma imagem, um documento HTML ou um script, é completamente carregado. Esse evento é útil para executar determinadas ações ou funções quando todo o conteúdo da página ou um elemento específico é carregado.** | var imagem = new Image();  imagem.onload = function() {  console.log('A imagem foi carregada com sucesso!');  };  imagem.src = 'caminho/para/imagem.jpg'; |
| <img> | **é uma tag HTML usada para exibir imagens em uma página da web. Ela é um elemento vazio, o que significa que não possui uma tag de fechamento correspondente.** | <img src="caminho/para/imagem.jpg" alt="Descrição da imagem"> |
| name=”” | **em HTML é usado para fornecer um nome a um elemento de entrada de dados, como um campo de texto (<input type="text">), uma caixa de seleção (<select>) ou um botão (<button>). O valor do atributo name é enviado junto com o valor inserido pelo usuário quando o formulário é enviado.** | <input type="number" name="txtano" id="txtano" min="0"> |
| min=”” | **é usado em campos de entrada numérica, como <input type="number">, para especificar o valor mínimo permitido. Ele define o valor mínimo que um usuário pode inserir no campo.** | <input type="number" min="5"> |
| <label></label> | **é usada para criar um rótulo associado a um elemento de formulário, como um campo de entrada de texto ou um botão de opção. O rótulo fornece uma descrição ou instrução para o elemento de formulário, tornando-o mais acessível e compreensível para os usuários.** | <label for="nome">Nome:</label>  <input type="text" id="nome" name="nome"> |
| For=”” | **O atributo for na tag <label> é usado para associar o rótulo a um elemento específico em um formulário. Esse atributo estabelece uma relação entre o rótulo e o elemento, permitindo que o usuário clique no rótulo para ativar o elemento associado.** | <label for="meuInput">Digite seu nome:</label>  <input type="text" id="meuInput" name="nome"> |
| checked | **O atributo checked é usado em elementos de input, como checkboxes e radio buttons, para indicar que o elemento está selecionado por padrão.** | <input type="checkbox" id="meuCheckbox" checked>  <label for="meuCheckbox">Opção 1</label> |
| size=”” | **O atributo size em HTML é usado para especificar o tamanho visual de um elemento de entrada, como <input> ou <select>. Ele determina quantos caracteres ou opções são exibidos ao mesmo tempo.** | <select size="5">  <option value="1">Opção 1</option>  <option value="2">Opção 2</option>  <option value="3">Opção 3</option>  <option value="4">Opção 4</option>  <option value="5">Opção 5</option>  </select> |
| <select></select> | **O elemento HTML <select> é usado para criar uma lista suspensa de opções para seleção. Ele permite que os usuários escolham uma opção de um conjunto de valores pré-definidos.** | <select>  <option value="opcao1">Opção 1</option>  <option value="opcao2">Opção 2</option>  <option value="opcao3">Opção 3</option>  </select> |
| <option></option> | **A tag <option></option> é usada dentro de um elemento <select> para definir uma opção na lista suspensa. Cada tag <option> representa uma escolha disponível para seleção.** | <select>  <option value="opcao1">Opção 1</option>  <option value="opcao2">Opção 2</option>  <option value="opcao3">Opção 3</option>  </select> |
| Placeholder=”” | **Serve para escrever normalmente em caixas de texto oque é para digitar ou ela significa** | <input type="number" name="base" id="base" placeholder="Base" class="text"> |
| <iframe></iframe> | **Serve para pegar um vídeo da internet estilo youtube** | <iframe width="500" height="315" src="https://www.youtube.com/embed/UXoBHQBT5OE" title="YouTube video player" frameborder="0" allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture; web-share" allowfullscreen></iframe> |

|  |  |  |
| --- | --- | --- |
| **Javascript** | | |
| /\*\*/ | **Serve para colocar um comentário de mais de uma linhas** | /\*bom dia para todos que estão nessa sala de aula incrível\*/ |
| // | **Comentário de uma linha só** | //vamos para a lua |
| document.write(); | **Serve para escrever algo na página** | document.write("Olá mundo"); |
| Alert(); | **É para criar uma caixa de aviso onde aparece quando você inicia a página** | alert("Olá mundo"); |
| Confirm ou windows.confirm | **Ele cria uma caixa onde devemos confirmar ou não** | window.confirm("esta gostando de javascript"); |
| console.log (); | **É para colocar uma mensagem no console do código fonte** | console.log("Oi"); |
| var | **Quando se tem que declarar uma variável** | var numero = 2; |
| let | **É uma variação de var** | let num = [5,8,9,7,6,4,1,0] |
| const | **Outra variação de var e let** | Const name = ‘christian’ |
| prompt (); | **Serve para escrever uma informação na caixa de alerta** | var nome = prompt ("Olá, qual seu nome"); |
| .lenght | **Ele conta a quantidade de caracteres/letras** | Nome.lenght |
| ParseInt | **Números inteiros** | var n1 = parseInt(prompt("digite um numero: ")) |
| ParseFloat | **Números Real** | var n1 = parseFloat(prompt("digite um numero: ")) |
| Number | **Números inteiros e real** | var n1 = Number(prompt("digite um numero: ")) |
| String(n) | **Transforma numero em string** | Document.write(“esse é seu numero”+ String(N)) |
| value | **Ela usada para transformar um valor numérico ou oque for em uma string** | resposta.value = inputTexto.value |
| toFixed(2) | **Indica quantas casa você quer depois da virgula** | n1.toFixed(2) |
| Replace(n) | **Substitui uma elemento da saída por outro** | Replace(‘.’,’,’) |
| toUpperCase() | **Serve para mostra um elemento em maiúsculo** | nome. toUpperCase() |
| toLowerCase() | **Serve para mostra um elemento em minúsculo** | nome.toLowerCase() |
| Var.toLocaleString(‘pt-BR’,{style: ‘currency’,currency: ‘moeda’}) | **Server para transforma em formatos de dinheiro** | N1.toLocaleString(‘pt-BR’,{style: ‘currency’,currency: ‘BRL’}) |
| + | **Soma de números e juntar dois termos** | 2+5=7, Documen.write(“esse é seu numero”+ String(N)) |
| - | **Subtração de números** | 5-2=3 |
| \* | **Multiplicação de números** | 5\*2=10 |
| / | **Divisão de números** | 5/2=2.5 |
| % | **Resto da divisão** | 5%2=1 |
| \*\* | **Número ao Quadrado** | 5\*\*2=25 |
| Var operador=num | **Serve para usar uma variável que já esta gravada e usar um operador para mudar ela** | Var n=3, n+=4: 7, n-=5: 2,n\*=4: 8,n/=2: 4, n\*\*=2: 16,n%=5: 1 |
| Incremento ou = | **Serve para somar ou subtrair 1** | Var n=9, n++=10,n--=9 |
| < | **inferioridade** | A<B |
| = ou== | **recebe e igualdade de conta de programas** | A=B e 5==1+4 |
| <= | **Não maior** | A<=B |
| != | **Diferença** | A!=B |
| >= | **Não menor** | A>=B |
| > | **Superioridade** | A>B |
| === | **Identidade que é algo idêntico** | A===B: verdadeiro, A===’B’: falso |
| !== | **Desigual restrito tem a mesma utilidade da identidade só que para desigualdade** | A!==B: verdadeiro, A!==’B’: falso |
| typeof | **Serve para descobrir o tipo de uma variável ou constante** | Typeof 1 : number, typeof ‘oi’ : string |
| ! | **Negação** | Não, primeiro |
| && | **Conjunção** | Em, segundo |
| || | **Disjunção** | Ou, terceiro |
| Ternário  Teste ? true : false | **Para fazer um teste** | Media > 7? ‘aprovado’ : ‘reprovado’  var res = x % 2 == 0? 5:9 |
| InnerHTML | **Serve para escrever no HTML** | InnerHTML(‘Oi’) |
| innerText | **É para escrever só o texto sem as modificações do HTML** | innerText(‘Oi’) |
| getElementsByTagName | **Serve para chamar um elemento nome do html ou css dentro do javascript** | var p1 = window.document.getElementsByTagName('p')[1]  aqui ele está chamando o segundo paragrafo (p) porque em Javascript vai de 0 a infinito e não de 1 a infinito |
| Variável.style.css | **Serve para criar um style css apartir do javascript** | p1.style.color = 'blue' |
| getElementById | **Serve para chamar um elemento html ou css pelo id** | var d = window.document.getElementById("msg")  aqui ele está chamando pela id que é dada pelo usuário a um elemento HTML nesse exemplo é essa daqui: <div id= "msg"> Clique em mim</div> |
| querySelector | **Serve para facilita a chamada de elementos** | var d = window.document.querySelector('div#msg') aqui ele esta chamando um elemento pelo id se ele quer chama pelo nome seria (‘div.msg’) |
| getElementsByClassName | **Serve para fazer a mesma coisa do id so que com class** | var elementosDestaque = document.getElementsByClassName("destaque"); ele esta chamando o elemento: <p class="destaque">Este é o primeiro parágrafo.</p> |
| addEventListener | **Ele adiciona um evento a um elemento e pode mudar o elemento** | a.addEventListener('click',clicar) |
| click | **Comando acionado quando há clique do mouse apartir do javascript** | a.addEventListener('click',clicar) |
| mouseenter | **Comando acionado quando o ponteiro do mouse esta em cima do elemento apartir do javascript** | a.addEventListener('mouseenter',entrar) |
| mouseout | **Comando acionado quando o ponteiro do mouse esta fora do elemento apartir do javascript** | a.addEventListener('mouseout',sair) |
| If(){} | **Serve para você criar uma condição e se essa condição for verdade vai criar um comando** | if (vel > 60) {  res.innerHTML = “você esta acima da velocidade  } |
| If(){} else {} | **Ele faz a mesma coisa do if normal só que se a condição for errada ele cria uma outra condição** | if (nac = 'Brasil') {  res.innerHTML += ' <p> Voce é Brasileiro</p> '}  else {  res.innerHTML += ' <p> Voce é estrangeiro </p> '} |
| event | **permite que você manipule e responda a eventos específicos que ocorrem em seu código JavaScript, permitindo interações com o usuário e personalizando o comportamento do seu aplicativo ou página da web.** | if (event.keyCode === 13) {  // Aciona o clique no botão  botao.click();  } |
| keyCode | **Ele é um tipo de comando que chama uma tecla pelo código dela** | event.keyCode === 13 |
| keypress | **Quando for preciona uma tecla esse comando será ativado** | a.addEventListener('keypress',enter) |
| new | **é utilizado em JavaScript para criar uma nova instância de um objeto a partir de uma função construtora. Ele é seguido pelo nome da função construtora e, opcionalmente, pelos argumentos que serão passados para o construtor.** | var data = new Date(); |
| Date(); | **A função Date() em JavaScript é usada para criar um objeto que representa uma data e hora. Quando chamada sem argumentos, ela retorna a data e hora atuais.** | var dataAtual = new Date();  console.log(dataAtual); |
| getHours() | **O método getHours() é usado em objetos Date em JavaScript para obter o valor da hora, especificado em formato de 24 horas, da data representada pelo objeto** | var dataAtual = new Date();  var horaAtual = dataAtual.getHours();  console.log(horaAtual); |
| getFullYear() | **O método getFullYear() é um método da classe Date em JavaScript que retorna o ano completo de uma data específica. Ele retorna um número de 4 dígitos representando o ano da data atual ou da data especificada.** | const dataAtual = new Date();  const ano = dataAtual.getFullYear();  console.log(ano); |
| setAttribute() | **O método setAttribute() é um método em JavaScript que permite definir o valor de um atributo em um elemento HTML. Ele recebe dois parâmetros: o nome do atributo que será definido e o valor a ser atribuído.** | const elemento = document.getElementById('meuElemento');  elemento.setAttribute('atributo', 'valor'); |
| createElement() | **O método createElement() é uma função em JavaScript que permite criar um novo elemento HTML. Ele recebe como argumento o nome da tag do elemento que será criado e retorna um objeto representando esse elemento.** | const novoElemento = document.createElement('tag'); |
| .appendChild | **O método appendChild() é uma função em JavaScript que permite adicionar um elemento filho a um elemento pai existente no documento HTML. Ele recebe como argumento o elemento que será adicionado como filho e o insere como o último elemento dentro do elemento pai.** | O método appendChild() é uma função em JavaScript que permite adicionar um elemento filho a um elemento pai existente no documento HTML. Ele recebe como argumento o elemento que será adicionado como filho e o insere como o último elemento dentro do elemento pai. |
| For(){} | **O for é uma estrutura de controle em JavaScript que permite executar um bloco de código repetidamente com base em uma condição específica. É comumente usado para percorrer e iterar por elementos de uma lista, como arrays.** | for (inicialização; condição; expressão final) {  // Bloco de código a ser executado  }  E  const array = [1, 2, 3, 4, 5];  for (let i = 0; i < array.length; i++) {  console.log(array[i]);  } |
| while(){} | **A estrutura de repetição while em JavaScript permite executar um bloco de código repetidamente enquanto uma condição especificada for verdadeira. A sintaxe básica do while é a seguinte:** | var contador = 1;  while (contador <= 5) {  console.log(contador);  contador++;  } |
| do{}while() | **O operador do...while em JavaScript é uma estrutura de controle que executa um bloco de código repetidamente até que uma determinada condição seja falsa. A estrutura básica do do...while é a seguinte:** | var contador = 1;  do {  console.log("Contagem: " + contador);  contador++;  } while (contador <= 5); |
| return | **A palavra-chave return em JavaScript é usada para especificar o valor de retorno de uma função. Quando uma função é invocada e a instrução return é encontrada, a execução da função é interrompida e o valor especificado após o return é retornado como resultado da função.** | function somar(a, b) {  return a + b;  }  var resultado = somar(3, 4);  console.log(resultado); // Saída: 7 |
| indexOf | **O método indexOf em JavaScript é usado para encontrar a posição de uma determinada substring em uma string. Ele retorna o índice da primeira ocorrência da substring na string original. Se a substring não for encontrada, o método retorna -1.** | ar str = 'Olá, mundo!';  var posicao = str.indexOf('mundo');  console.log(posicao); // Saída: 5 |
| push | **O método push em JavaScript é utilizado para adicionar um ou mais elementos no final de um array. Ele modifica o array original, adicionando os elementos fornecidos como argumentos no final do array.** | var array = [1, 2, 3];  array.push(4, 5);  console.log(array); // Saída: [1, 2, 3, 4, 5] |
| focus | **O método focus em JavaScript é usado para definir o foco em um elemento HTML, como um campo de entrada de texto, um botão ou um link. Quando um elemento recebe o foco, ele se torna o alvo principal para interações do usuário, como digitação de texto ou clique.** | var inputElement = document.getElementById('meuInput');  inputElement.focus(); |
| array | **Em JavaScript, um array é uma estrutura de dados que permite armazenar e acessar vários valores em uma única variável. Ele é uma coleção ordenada de elementos, onde cada elemento é identificado por um índice numérico.** | var frutas = ['maçã', 'banana', 'laranja'];  console.log(frutas[0]); // saída: maçã  frutas.push('manga');  console.log(frutas); // saída: ['maçã', 'banana', 'laranja', 'manga']  frutas.pop();  console.log(frutas); // saída: ['maçã', 'banana', 'laranja'] |
| object | **Em JavaScript, um objeto é uma estrutura de dados que permite armazenar múltiplos valores relacionados em pares de chave-valor. As chaves são strings (ou símbolos) que atuam como identificadores únicos para acessar os valores correspondentes dentro do objeto.** | var pessoa = {  nome: "João",  idade: 25,  profissao: "Engenheiro"  };  console.log(pessoa.nome); // saída: "João"  console.log(pessoa['idade']); // saída: 25  pessoa.endereco = "Rua A, 123";  pessoa.idade = 26;  delete pessoa.profissao; |
| sort() | **O método sort() em JavaScript é usado para ordenar os elementos de um array em ordem alfabética (ou em ordem crescente se os elementos forem números). Esse método altera o próprio array e retorna o array ordenado.** | var numeros = [10, 5, 8, 2, 1];  numeros.sort(function(a, b) {  return a - b;  });  console.log(numeros); // saída: [1, 2, 5, 8, 10] |
| null | **Em JavaScript, null é um valor primitivo que representa a ausência intencional de qualquer valor de objeto. Ele indica que uma variável não aponta para nenhum objeto.** | let x = null;  console.log(x); // Saída: null |
| toString() | **toString() é um método de todo objeto JavaScript usado para exibir um objeto como texto ou usá-lo como uma string.** | let num = 15;  let result = num.toString();  console.log(result); // Saída: "15" |
| switch | **A instrução switch em JavaScript avalia uma expressão e executa instruções associadas a um caso específico que corresponde ao valor da expressão1. É semelhante a várias instruções if...else encadeadas.** | let x = 1;  switch (x) {  case 1:  console.log('x é igual a 1');  break;  case 2:  console.log('x é igual a 2');  break;// Saída: x é igual a 1 |
| case |  |  |
| Math | **Math é um objeto embutido em JavaScript que possui propriedades e métodos para constantes e funções matemáticas. Ao contrário de outros objetos globais, Math não é um construtor. Todas as propriedades e métodos de Math são estáticos** | let x = Math.PI;  console.log(x); // Saída: 3.141592653589793 |
| pow() | **O método Math.pow() em JavaScript retorna a base elevada ao expoente, ou seja, base^expoente. É um método estático de Math, então você sempre o usará como Math.pow(), em vez de usá-lo como um método de um objeto do tipo Math que você tenha criado** | let x = Math.pow(2, 3);  console.log(x); // Saída: 8 |
| sqrt() | **O método Math.sqrt() em JavaScript retorna a raiz quadrada de um número1. É um método estático de Math, então você sempre o usará como Math.sqrt(), em vez de usá-lo como um método de um objeto do tipo Math que você tenha criado** | let x = Math.sqrt(9);  console.log(x); // Saída: 3 |

|  |  |  |
| --- | --- | --- |
| **CSS** | | |
| Comando{} | **Serve para pegar um elemento HTML e mudar o estilo** | Body{ colo:black: } |
| Background-color | **É para mudar a cor de fundo do elemento referido** | background-color: blue; |
| Color | **É para mudar os elementos dentro do elemento HTML referido,normalmente são a letras** | color: white; |
| font | **Ele muda o tamanho e tipo de fonte** | font: 20px arial; |
| height | **Ele muda a altura de um elemento** | height: 200px; |
| width | **Ele muda a largura de um elemento** | width: 200px; |
| line-height | **define o espaçamento vertical entre as linhas de texto em um elemento. Ela controla a altura da linha, afetando o espaço entre as palavras e o espaçamento entre as linhas** | line-height: 200px; |
| Text-align | **Serve para escolher o posicionamento de um elemento** | text-align: center; |
| Border-radius | **Ele pega um elemento e cria bordas arredondadas nele** | border-radius: 10px; |
| border | **Ele cria uma borda e pode se escolher o tamanho e a cor** | border: 2px solid; |
| padding | **Ele cria um espaçamento geral dentro do elemento** | padding: 15px; |
| padding -left | **Ele cria um espaçamento para a esquerda dentro do elemento** | padding -left: auto; |
| padding -right | **Ele cria um espaçamento para direita dentro do elemento** | padding -right: auto; |
| padding -top | **Ele cria um espaçamento para cima dentro do elemento** | padding -top: 20px; |
| padding -bottom | **Ele cria um espaçamento para baixo dentro do elemento** | padding -bottom: 20px; |
| margin | **Ele cria um espaçamento geral fora do elemento** | margin: auto; |
| Margin-left | **Ele cria um espaçamento para a esquerda fora do elemento** | margin-left: auto; |
| Margin-right | **Ele cria um espaçamento para direita fora do elemento** | margin-right: auto; |
| Margin-top | **Ele cria um espaçamento para cima fora do elemento** | Margin-top: 20px; |
| Margin-bottom | **Ele cria um espaçamento para baixo fora do elemento** | Margin-bottom: 20px; |
| box-shadow | **Ele cria um tipo de sombra do elemento para fora dele** | box-shadow: 3px 3px 10px rgba(0, 0, 0, 0.671); |
| font-style | **Ele diz qual vai ser a fonte que vai ser usada no elemento** | font-style: italic; |
| font-size | **Ele muda o tamanha da letra** | font-size: 15px; |
|  |  |  |