

Como podemos acessar o elemento que dispomos um evento? `event.currentTarget`
" " adicionar um evento a um elemento através do `on`? `element.addEventListener()`
O que faz o método `preventDefault`? Impede o comportamento padrão de um evento

Trabalhando com formulários no jQuery

`element.addEventListener("event", callback function)`

`document.querySelector('input[name="nome"]').value`

pega elemento \downarrow atributo \downarrow nome \downarrow valor

Quando o input for um 'select', ele retorna o valor selecionado.

`let salad = ""` esse input é uma \downarrow `select` \downarrow `input` \downarrow `value` \downarrow `nome` \downarrow `valor`
`document.querySelectorAll('input[name="salad"]:checkbox').forEach(` verifica se o input está \downarrow `checked`

`function(item) { salad += "-" + item.value + " \n" })`

Está sendo feita uma iteração para cada input. Nessa iteração está sendo chamada uma função que recebe como parâmetro o item (cada input). Dentro da função, está sendo feita uma concatenação com o valor de cada input.

Evitando o comportamento padrão dos formulários

O comportamento padrão dos formulários é atualizar a página rapidamente quando submetidos.

`form.addEventListener('submit', function(event) {`

`event.preventDefault()`

`}`

\downarrow prevenindo o comportamento padrão das elementos, no caso a página

do form é de atualizar e mostrar os dados conforme os atributos `action` e `method`. Alguns elementos possuem comportamento padrão, em caso de necessidade consulte a documentação.

A utilidade disso é caso você queira um formulário que não envie os dados para outro lugar. Vale destacar que, caso queira limpar os inputs do formulário basta alterar o `value` deles. Por exemplo: `document.getElementById("element").value = ""`