



# JS CHEATSHEET I

🌐 FTAMBERLINI.DEV.BR

📍 FTAMBERLINIDEV

👤 @FTAMBERLINI

## STRINGS

### #ATRIBUTOS E MÉTODOS

```
let str = "abcdefghij";
```

```
length           //retorna o tamanho da string = 10
str.length       // 10
charAt()         // retorna o caractere em uma posição específica
str.charAt(2)    // 'c'
charCodeAt()     // retorna o código Unicode do caractere na
posição indicada
str.charCodeAt(0) // 97
concat()         // concatena duas strings
str.concat('XYZ') // 'abcdefghijXYZ'
includes()       // verifica se a string contém um texto
str.includes('def') // true
endsWith()       // verifica se a string termina com um texto
str.endsWith('hij') // true
startsWith()     // verifica se a string começa com um texto
str.startsWith('abc') // true
indexOf()        // retorna o índice da primeira ocorrência
str.indexOf('d')  // 3
lastIndexOf()    // retorna o índice da última ocorrência
str.lastIndexOf('h') // 7
slice()          // retorna parte da string entre índices
str.slice(2, 6)  // 'cdef'
substring()      // retorna parte da string entre índices
str.substring(2, 6) // 'cdef'
toUpperCase()    // converte em letras maiúsculas
str.toUpperCase() // 'ABCDEFGHIJ'
toLowerCase()    // converte em letras minúsculas
str.toLowerCase() // 'abcdefghij'
trim()           // remove espaços no início e no fim
' abc '.trim()   // 'abc'
trimStart()      // remove espaços no início
' abc'.trimStart() // 'abc'
trimEnd()        // remove espaços no fim
'abc '.trimEnd() // 'abc'
replace()        // substitui a primeira ocorrência
str.replace('abc', 'XYZ') // 'XYZdefghij'
replaceAll()     // substitui todas as ocorrências
'aba'.replaceAll('a', 'x') // 'xbx'
split()          // divide a string em partes
str.split('e')   // ['abcd', 'fghij']
repeat()         // repete a string
str.repeat(2)    // 'abcdefghijabcdefghij'
padStart()       // adiciona caracteres no início
str.padStart(12, '*') // '**abcdefghij'
padEnd()         // adiciona caracteres no final
str.padEnd(12, '*')  // 'abcdefghij**'
toString()       // converte o objeto String em texto literal
str.toString()     // 'abcdefghij'
```

## EVENTOS

### #CHAMADA DE FUNÇÃO APÓS EVENTO EM UM ELEMENTO

```
<button onclick="funcaoExemplo();">
  Clique aqui!
</button>
```

### #CHAMADA DE FUNÇÃO APÓS EVENTO DISPARADO

```
<script>
  // Adiciona um listener para o evento de tecla pressionada
  document.addEventListener('keydown', function(evento) {
    console.log('Tecla pressionada:', evento.key);
  });
</script>
```

### #MOUSE

```
onclick           //ocorre quando o usuário clica no elemento
ondblclick        //ocorre quando o usuário dá um clique duplo
onmousedown       //ocorre quando botão do mouse é pressionado
onmouseup         //ocorre quando o botão do mouse é solto
onmousemove       //ocorre quando o mouse é movido sobre
onmouseover       //ocorre quando o mouse entra na área
onmouseout        //ocorre quando o mouse sai da área
oncontextmenu     //ocorre quando o botão direito é clicado
onwheel           //ocorre quando a roda do mouse é rolada
```

### #TECLADO

```
onkeydown //ocorre quando qualquer tecla é pressionada
onkeyup   //ocorre quando a tecla é solta
```

### #FORMULÁRIO

```
onsubmit // ocorre quando um formulário é enviado
onreset  // ocorre quando um formulário é resetado
onfocus // ocorre quando um campo recebe foco
onblur   // ocorre quando um campo perde foco
onchange // ocorre quando o valor de um campo perde foco
oninput  // ocorre quando o valor de um campo muda em
tempo real
onselect // ocorre quando o usuário seleciona texto em um
campo
```

### #JANELA/DOCUMENTO

```
onload // ocorre quando a página/img termina de carregar
onunload // ocorre quando a página é descarregada
onresize // ocorre quando a janela é redimensionada
onscroll // ocorre quando a página é rolada
onerror // ocorre quando acontece um erro no carregamento
```

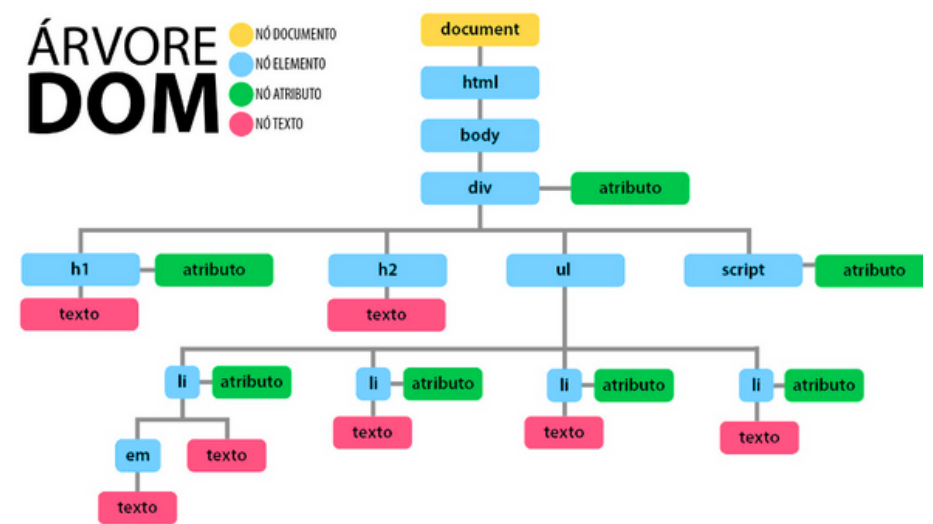
### #TOQUE

```
ontouchstart // ocorre quando o dedo toca a tela
ontouchmove  // ocorre quando o dedo se move na tela
ontouchend   // ocorre quando o dedo é levantado
ontouchcancel // ocorre quando o toque é interrompido (ex:
chamada telefônica)
```

### #OUTROS EVENTOS

```
oncopy // ocorre quando o usuário copia conteúdo
oncut   // ocorre quando o usuário recorta conteúdo
onpaste // ocorre quando o usuário cola conteúdo
ondrag  // ocorre durante o arrasto de um elemento
ondrop  // ocorre quando o elemento arrastado é solto
```

## DOM



### #SELEÇÃO DE ELEMENTOS

```
document.getElementById("ElementoID");
//retorna um único elemento com o id repassado
document.getElementsByClassName("nomeClasse");
//retorna uma coleção com os elementos da classe repassada
document.getElementsByTagName("nomeTag");
//retorna uma coleção com os elementos da tag repassada
document.getElementsByName("nome");
//retorna uma coleção com os elementos com nome repassado
document.querySelector("seletor");
//retorna o primeiro elemento que corresponde ao seletor
document.querySelectorAll("seletor");
//retorna todos os elementos que correspondem ao selector
const formularios = document.forms[0]
//retorna o 1º formulário do documento
const imagem = document.images["foto"]
//retorna a imagem com atributo name="foto"
```

### #ACESSO AO TEXTO DO ELEMENTO

```
const e = document.getElementById("ElementoID");
e.textContent //apenas texto do elemento id="ElementoID"
e.innerText   //texto com CSS do elemento id="ElementoID"
e.innerHTML   //texto com HTML do elemento id="ElementoID"
```

### #ACESSO AO ATRIBUTO DO ELEMENTO

```
const e = document.getElementById("ElementoID");
e.getAttribute("name") //acessa o atributo do elemento
e.removeAttribute("name") //remove o atributo "name"
e.attributes            //lista todos atributos
e.setAttribute("name", "novo nome") //altera o valor do
atributo "name" para "novo nome"
```

### #CRIAÇÃO, INSERÇÃO E EXCLUSÃO DE ELEMENTO

```
const novaDiv = document.createElement("div")
//cria o nome elemento DIV
const e = document.getElementById("ElementoID");
e.appendChild(novaDiv) //insere novaDIV como último filho
e.append(novaDiv)      //insere novaDIV como último filho
e.prepend(novaDiv)     //insere novaDIV como primeiro filho
e.before(novaDiv)      //insere novaDIV antes como irmão de e
e.after(novaDiv)       //insere novaDIV depois como irmão de e
e.remove()             //remove o elemento com id = "ElementoID"
e.removeChild("p")     //remove o filho <p> do elemento
```

FUNÇÕES

```
#FUNÇÃO DECLARADA
function soma(a, b) {
  return a + b;
}
x = soma(a, b);
#FUNÇÃO ANÔNIMA
const soma = function(a, b) {
  return a + b;
}
#FUNÇÃO CURTA (ARROW FUNCTION)
const soma = (a,b) => a + b;
```

ARRAY

```
#ATRIBUTOS E MÉTODOS
let arr = [1, 2, 3, 4, 5];
arr[0] // 1 arr[4] //5 arr[5] //undefined
arr.length // 5
push() // adiciona um ou mais elementos ao final
arr.push(6) // retorna 6 (novo comprimento) → arr = [1,2,3,4,5,6]
pop() // remove o último elemento do array
arr.pop() // retorna 6 → arr = [1,2,3,4,5]
unshift() // adiciona um ou mais elementos no início
arr.unshift(0) // retorna 6 → arr = [0,1,2,3,4,5]
shift() // remove o primeiro elemento
arr.shift() // retorna 0 → arr = [1,2,3,4,5]
concat() // combina dois arrays
arr.concat([6,7]) // [1,2,3,4,5,6,7]
join() // junta os elementos em uma string
arr.join('-') // '1-2-3-4-5'
indexOf() // retorna o índice da primeira ocorrência
arr.indexOf(3) // 2
reverse() // inverte a ordem dos elementos
arr.reverse() // [5,4,3,2,1]
sort() // ordena os elementos (como strings por padrão)
[3,1,2].sort() // [1,2,3]
toString() // converte o array em string
arr.toString() // '1,2,3,4,5'
fill() // preenche o array com um valor
arr.fill(0,2,4) // [1,2,0,0,5]
filter() // cria um novo array com os elementos que
passam no teste
arr.filter(x => x % 2 === 0) // [2,4]
map() // cria um novo array aplicando uma função
arr.map(x => x * 2) // [2,4,6,8,10]
forEach() // executa uma função para cada elemento (sem
retornar)
arr.forEach(x => console.log(x)) // imprime 1 2 3 4 5
reduce() // reduz o array a um único valor (acumulador)
arr.reduce((soma, x) => soma + x, 0) // 15
find() // retorna o primeiro elemento que satisfaz uma
condição
arr.find(x => x > 3) // 4
```

OBJETOS E CLASSES

```
#OBJETO
const pessoa = { //nome do objeto
  nome:"Pedro", //atributos
  sobrenome:'da Silva',
  peso:75,
  altura:1.80,
  calcularIMC(){ //método
    return this.peso/this.altura**2
  }
}
//Acessando o atributo do objeto
console.log("Meu nome é " + pessoa.nome);
//Invocando o método do objeto
console.log(`O meu IMC é ${pessoa.calcularIMC()}`);
//Criando um novo atributo
pessoa.idade=20;

#CLASSE (SUPERCLASSE)
class Pessoa { //definição
  constructor(nome, sobrenome, peso, altura){
    this.nome = nome;
    this.sobrenome = sobrenome;
    this.altura = altura;
    this.#peso = peso //atributo privado
  }
  //método público para calcular o IMC
  calcularIMC(){
    return this.peso/this.altura**2
  }
  //método público para mostrar nome completo
  nomeCompleto(){
    return `${this.nome} ${this.sobrenome}`;
  }
}
//Criando um objeto a partir da classe
const pessoa = new Pessoa("Ana","Silva",1.70,65);
//Invocando o método do objeto pessoa
console.log(`O meu IMC é ${pessoa.calcularIMC()}`);
//Acessando um atributo do objeto pessoa
console.log("Minha altura é " + pessoa.altura);

#CLASSE COM HERANÇA (SUBCLASSE)
class Aluno extends Pessoa {
  constructor(nome, sobrenome, peso, altura, matricula){
    //chama construtor da classe pai
    super(nome, sobrenome, peso, altura, matricula);
    this.matricula = matricula;
  }
}
//Criando um objeto a partir da classe
const aluno = new Aluno("Ana","Silva",1.70,65,2025020001);
//Invocando o método herdado da classe pai
console.log(`O meu IMC é ${pessoa.calcularIMC()}`);
//Acessando um atributo do objeto aluno
console.log("Minha matricula é " + aluno.matricula);
```



JS CHEATSHEET I

FTAMBERLINI.DEV.BR  
FTAMBERLINIDEV  
@FTAMBERLINI

TIPOS DE DADOS (DATA TYPES)

```
#DECLARAÇÃO DE VARIÁVEIS
var a = 1 //escopo global/função
//permite redeclaração no mesmo bloco
//é içada (hoisted)
//evite usar em código moderno

let b = 2 //escopo de bloco e não pode ser redeclarada
//pode ser reatribuída

const c = 3 //escopo de bloco e não pode ser redeclarada
//NÃO pode ser reatribuída
//Atributos de objeto const podem ser alterados

#VALORES E TIPOS
false, true //boolean
18, 3.14, NaN //number (int, dec, not number)
0b10000, 0o000, 0x0010 //number (binary, octa, hexa)
"Maria" , `Pedro` //string
undefined, null, Infinity //special
let a = [1,2,3] //array
let pessoa={
  nome: "Maria",
  idade: 18} //object
const dobro = (x) => x * 2;//function
```

ESTRUTURA BÁSICA

```
#SCRIPT NA PÁGINA
<script type="text/javascript">
  console.log("Olá, mundo!");
</script>

#SCRIPT ARQUIVO EXTERNO
<script src="script.js" ></script>

#COMENTÁRIOS
/* Comentário de
  múltiplas linhas */

// Comentário de uma única linha

#SAÍDAS (OUTPUT)
console.log("Olá!") //escrita no console do navegador
document.write("Olá!") //escrita no arquivo html
alert("Olá!") //janela de alerta
confirm("Confirma?") //caixa de diálogo (sim/não)
prompt("Idade?","0") //caixa de diálogo de entrada
//2º argumento é o valor inicial
```