

## Javascript 1 algoritmos de ordenação

### 01 do papel para o código pegar livro mais barato

```
01DoPapelparaCodigo.js > ...
1  const precosLivros = [25, 15, 30, 50, 45, 20];
2
3  let maisBarato = 0;
4
5  for (let i = 0; i < precosLivros.length; i++) {
6    if (precosLivros[i] < precosLivros[maisBarato]) {
7      maisBarato = i;
8    }
9  }
10
11  console.log(`o livro mais barato custa ${precosLivros[maisBarato]}`);
```

### 02 exibindo livros

```
02listaLivros.js x 01DoPapelparaCodigo.js
02listaLivros.js > <unknown>
1  const precosLivros = [
2    {
3      titulo: 'Javascript',
4      preco: 25
5    },
6    {
7      titulo: 'PHP',
8      preco: 15
9    },
10   {
11     titulo: 'Java',
12     preco: 30
13   }
14 ];
15
16 const livros = require('./02listaLivros');
17
18 let maisBarato = 0;
19
20 for (let i = 0; i < livros.length; i++) {
21   if (livros[i].preco < livros[maisBarato].preco) {
22     maisBarato = i;
23   }
24 }
25
26 console.log(`o livro mais barato custa ${livros[maisBarato].preco}`);
```

### 03 selection sort

```
function menorValor(arrProdutos, posicaoInicial) {
  let maisBarato = posicaoInicial;

  for (let atual = posicaoInicial; atual < arrProdutos.length; atual++) {
    if (arrProdutos[atual].preco < arrProdutos[maisBarato].preco) {
      maisBarato = atual
    }
  }

  return maisBarato;
}

module.exports = menorValor;
```

```
03SelectionSort.js > menorValor
1  const livros = require('./02listaLivros');
2  const menorValor = require('./01menorValor');
3
4  for (let atual = 0; atual < livros.length - 1; atual++) {
5    let menor = menorValor(livros, atual)
6
7    let livroAtual = livros[atual];
8    console.log('posição atual', atual)
9    console.log('livro atual', livros[atual])
10   let livroMenorPreco = livros[menor];
11   console.log('livro menor preço', livros[menor])
12
13   livros[atual] = livroMenorPreco
14   livros[menor] = livroAtual
15 }
16
17 console.log(livros)
```

### 04 Insertion Sort

```
const livros = require('./02listaLivros');

function insertionSort(lista) {
  for (let atual = 0; atual < lista.length; atual++) {
    let analise = atual;

    while (analise > 0 && lista[analise].preco < lista[analise - 1].preco) {
      let itemAnalise = livros[analise];
      let itemAnterior = lista[analise - 1];

      lista[analise] = itemAnterior;
      lista[analise - 1] = itemAnalise;

      analise--;
    }
  }

  console.log(lista);
}

insertionSort(livros);
```

### Curso reconhecimento de Voz

<https://colorhunt.co/>

[https://developer.mozilla.org/en-US/docs/Web/API/Web\\_Speech\\_API](https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API)

[https://developer.mozilla.org/en-US/docs/Web/API/Web\\_Speech\\_API/Using\\_the\\_Web\\_Speech\\_API](https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API/Using_the_Web_Speech_API)

sortearNumero.js > numeroSecreto

```
const menorValor = 0;
const maiorValor = 100;
const numeroSecreto = 42;

function gerarNumeroAleatorio() {
  return parseInt(Math.random() * maiorValor + 1);
}

const elementoMenorValor = document.getElementById('menor-valor');
const elementoMaiorValor = document.getElementById('maior-valor');
```

recomecamentovoz.js >

```
1 const elementoChute = document.getElementById("chute");
2
3 window.SpeechRecognition = window.SpeechRecognition || window.webkitSpeechRecognition;
4
5 const recognition = new SpeechRecognition();
6 recognition.lang = "pt-BR";
7 recognition.addEventListener('result', listener);
8 // recognition.addEventListener('end', () =>
9 //   recognition.start()); deixa ligado
10
11 function onSpeak() {
12   recognition.start();
13 }
14
15 function listener(e) {
16   let chute = e.results[0][0].transcript
17   exibeChuteNaTela(chute);
18   verificaSeChuteValido(chute);
19 }
20
21 function exibeChuteNaTela(chute) {
22   elementoChute.innerHTML = `
23     <div>Você disse:</div>
24     <span class="box">${chute}</span>
25     <button onclick="location.reload()">Reiniciar</button>
26   `
27 }
```

validacao.js > ...

```
function verificaSeChuteValido(chute) {
  const numero = +chute

  if (Number.isNaN(numero)) {
    elementoChute.innerHTML = `
      <div>Valor inválido, apenas números</div>
      <button onclick="location.reload()">Reiniciar</button>
    `
    return false;
  }

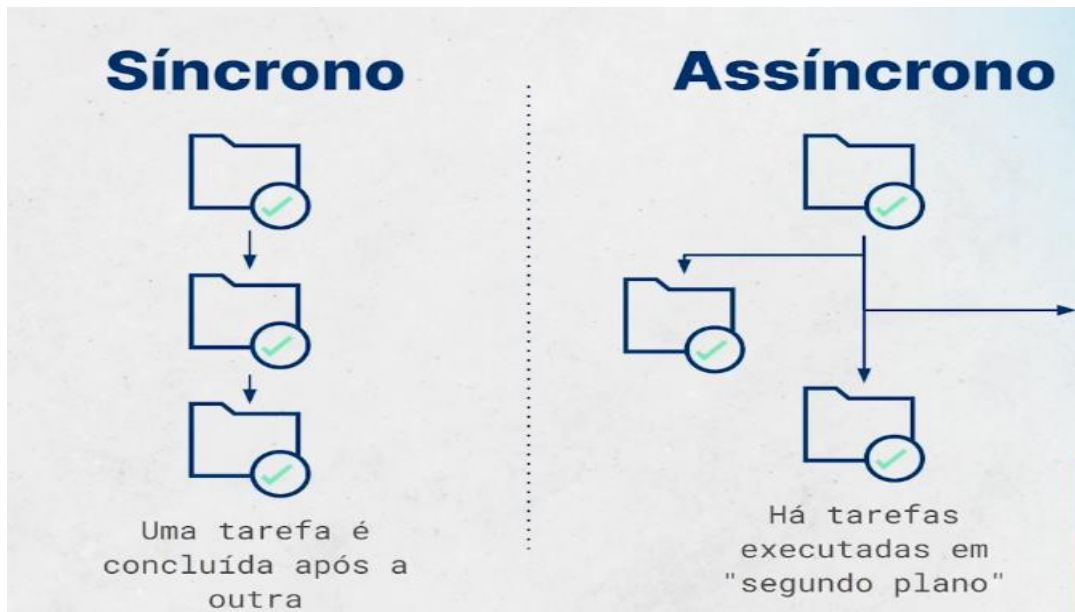
  if (numero > maiorValor || numero < menorValor) {
    elementoChute.innerHTML += `
      <div>O valor precisa estar entre ${menorValor} e ${maiorValor}</div>
    `
  } else if (numero > numeroSecreto) {
    elementoChute.innerHTML += `
      <div>O número secreto é menor</div>
    `
  } else {
    elementoChute.innerHTML += `
      <div>O número secreto é maior</div>
    `
  }

  if (numero === numeroSecreto) {
    document.body.innerHTML = `
      <h2>Parabéns Você acertou!</h2>
      <h3>O número secreto era ${numeroSecreto}</h3>
      <button onclick="location.reload()">Reiniciar</button>
    `
    return;
  }
}

document.body.addEventListener('click', e => {
  if (e.target.id == 'jogar-novamente') {
    window.location.reload()
  }
})
```

## JAVASCRIPT CONSUMINDO E TRATANDO DADOS DE UM API

### 01 síncrono e assíncrono



<https://viacep.com.br/>

```
javascript > 03JavaScriptConsumindoTratandoDadosAPI > script.js > ...
1 let consultaCEP = fetch('http://viacep.com.br/ws/01001000/json/')
2   .then(res => res.json())
3   .then(res => console.log('res: ', res))
4

script > 03JavaScriptConsumindoTratandoDadosAPI > script.js > consultaCEP
let consultaCEP = fetch('http://viacep.com.br/ws/01001250/json/')
  .then(res => res.json())
  .then(res => {
    if (res.erro) {
      throw Error("tente novamente mais tarde");
    }
    console.log('res: ', res)
  })
  .catch(error => console.log(error));

script > 03JavaScriptConsumindoTratandoDadosAPI > script.js > ...
let consultaCEP = fetch('http://viacep.com.br/ws/01001250/json/')
  .then(res => res.json())
  .then(res => {
    if (res.erro) {
      throw Error("tente novamente mais tarde");
    }
    console.log('res: ', res)
  })
  .catch(error => console.log(error))
  .finally(msg => console.log('Finalizado'))

script > 03JavaScriptConsumindoTratandoDadosAPI > script.js > ...
1 async function buscaEndereco() {
2   let consultaCEP = await fetch('http://viacep.com.br/ws/01001000/json/');
3   let res = await consultaCEP.json();
4   console.log(res);
5 }
6
7 buscaEndereco();
```

```
async function buscaEndereco() {  
  try {  
    let consultaCEP = await fetch('http://viacep.com.br/ws/01001000/json/');  
    let res = await consultaCEP.json();  
    if (res.erro) {  
      throw Error("Cep não existe...");  
    }  
    console.log(res);  
  } catch(err) {  
    console.log(err);  
  }  
}  
  
buscaEndereco();
```