

JS Bitwise

JS RegExp

Erros JS

Escopo JS

JS Hoisting

Modo JS Strict

JS esta palavra-chave

Função de seta JS

Classes JS

JS JSON

Depuração JS

Guia de estilo JS

JS Best Practices

JS Erros

JS Performance

Palavras Reservadas JS

Objetos JS

Definições de Objeto

Propriedades do objeto

26 de Junho

Venha ao GRANDE ANIVERSÁRIO

OFERTA DE 20.000.000kz EM COMPRAS

DESKONTÃO

Dá gosto poupar assim!

# Desempenho de JavaScript

< Anterior

Próximo >

Como acelerar seu código JavaScript.

## Reduza a atividade em loops

Os loops são frequentemente usados na programação.

Cada instrução em um loop, incluindo a instrução for, é executada para cada iteração do loop.

As instruções ou atribuições que podem ser colocadas fora do loop farão com que o loop seja executado mais rapidamente.

Ruim:

```
for (let i = 0; i < arr.length; i++) {
```

Código melhor:

```
let l = arr.length;
for (let i = 0; i < l; i++) {
```

O código incorreto acessa a propriedade length de uma matriz cada vez que o loop é iterado.

The better code accesses the length property outside the loop and makes the loop run faster.

## Reduce DOM Access

Accessing the HTML DOM is very slow, compared to other JavaScript statements.

If you expect to access a DOM element several times, access it once, and use it as a local variable:

Example

```
const obj = document.getElementById("demo");
obj.innerHTML = "Hello";
```

Try it Yourself >

## Reduce DOM Size

Keep the number of elements in the HTML DOM small.

This will always improve page loading, and speed up rendering (page display), especially on smaller devices.

Every attempt to search the DOM (like getElementsTagName) will benefit from a smaller DOM.

## Avoid Unnecessary Variables

Don't create new variables if you don't plan to save values.

Often you can replace code like this:

```
let fullName = firstName + " " + lastName;
document.getElementById("demo").innerHTML = fullName;
```

With this:

```
document.getElementById("demo").innerHTML = firstName + " " + lastName;
```

## Delay JavaScript Loading

Putting your scripts at the bottom of the page body lets the browser load the page first.

Durante o download de um script, o navegador não iniciará nenhum outro download. Além disso, todas as atividades de análise e renderização podem ser bloqueadas.

A especificação HTTP define que os navegadores não devem baixar mais de dois componentes em paralelo.

Uma alternativa é usar `defer="true"` na tag de script. O atributo defer especifica que o script deve ser executado após o término da análise da página, mas só funciona para scripts externos.

Se possível, você pode adicionar seu script à página por código, após o carregamento da página:

Exemplo

```
<script>
window.onload = function() {
  const element = document.createElement("script");
  element.src = "myScript.js";
  document.body.appendChild(element);
};
</script>
```

## Evite usar com

Evite usar a `with` palavra - chave. Isso tem um efeito negativo na velocidade. Ele também confunde os escopos do JavaScript.

A `with` palavra-chave **não é permitida** no modo estrito.

< Anterior

Próximo >

PROPAGANDA

COLOR PICKER

COMO NÓS

f i in

Get certified by completing a course today!

W3SCHOOLS CERTIFIED 2021

Get started

CODE GAME

Play Game