



Boas práticas de Javascript



**Certified
Developer**

The Ultimate Tech Degree

DigitalHouse >
Coding School



Evite usar variáveis globais

Sempre que possível, evite utilizar variáveis globais. Isso inclui todos os tipos de dados, objetos e funções, pois eles podem ser sobrescritos por outras bibliotecas ou códigos espalhados pelo seu site/aplicativo.





Utilização dos ponto-e-vírgulas

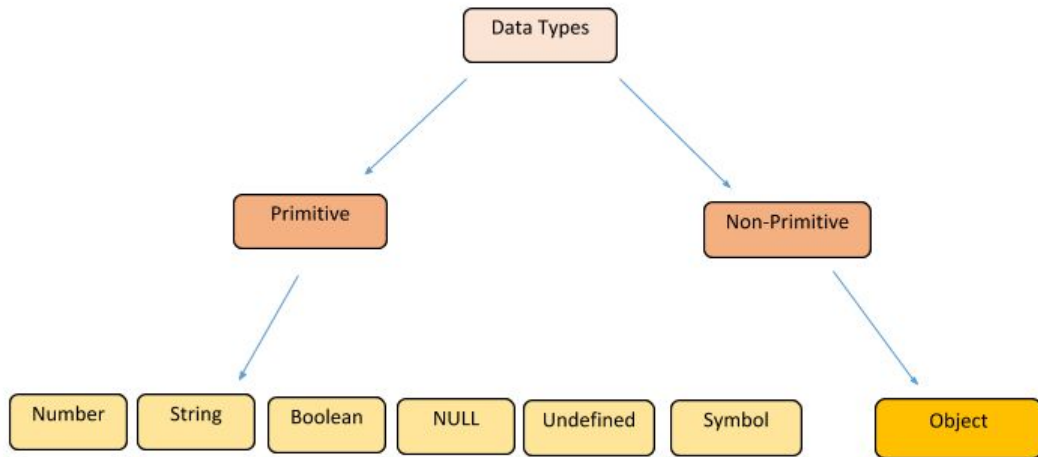
A maioria dos interpretadores permite que os ponto-e-vírgulas sejam omitidos de certas partes do código (ex.: antes do fechamento das chaves). No entanto, isso não é necessariamente uma regra, e pode levar seu código a executar de forma não desejada.





Nunca declare números, strings e booleanos como objetos.

Declarar variáveis de tipos “primitivos” como objetos causa um impacto na performance do código que é desnecessário em 99% das vezes. Sempre que possível declare-os utilizando os valores primitivos. Ex.: **let umaString = “teste”;**





JavaScript puro sempre pode ser mais rápido que usar uma biblioteca

Bibliotecas como o jQuery, bootstrap.js, w3.js, entre outras, são ótimas ferramentas para reduzir o tempo de desenvolvimento de aplicativos e websites. No entanto, quando carregadas por seus usuários e as funções que elas fornecem podem ser significativamente mais lentas que a utilização do JavaScript puro.





Coloque os scripts na parte final da sua página

Algumas bibliotecas e frameworks já tem instruído os desenvolvedores, mas vale a pena lembrar: na linguagem HTML, sempre que possível, chame os scripts no final antes do `</body>`, pois desta forma você estará permitindo que o HTML e o CSS renderizem em tela antes de carregar os scripts. Isto dá uma sensação de velocidade maior para o usuário.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="UTF-8"/>
5   <title>Hello World Com Javascript</title>
6 </head>
7 <body>
8   <script>
9     |
10  </script>
11 </body>
12 </html>
```



Minimize as chamadas HTTP

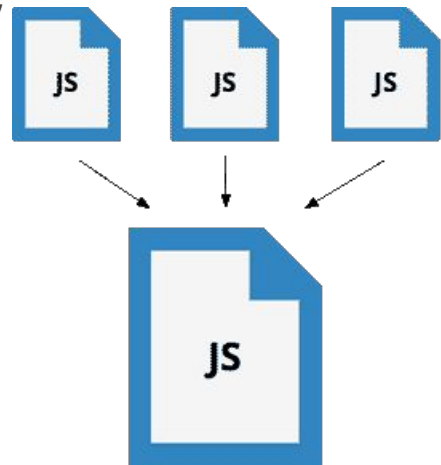
Cerca de 80% do tempo de resposta para o usuário final é gasto no front-end e boa parte deste tempo é baixando imagens, stylesheets, scripts, etc. É aconselhável que você entregue todo o código JavaScript de uma vez só para economizar chamadas ao servidor HTTP. Para isso existem diversas ferramentas que permitem minificar e unir diversos arquivos JavaScript em um só.





Deixe o código JavaScript em arquivos externos

Normalmente o uso de arquivos externos (colocar o código em um **arquivo js** ao invés da tag script dentro do HTML) gera uma página com tempo de carregamento mais rápido pois o navegador costuma fazer cache de arquivos JS e CSS. Se o estilo e o script do código existirem dentro do HTML, eles serão baixados juntos com a página toda vez que ela for carregada. Por mais que essa prática reduza o número de chamadas HTTP, ela aumenta o tempo base fixo para carregamento do website.





Minifique e use Gzip

Apesar da indentação e comentários serem ótimos para melhorar a leitura, os interpretadores não precisam deles. E, na verdade, precisam trabalhar mais para isso. Por isso, utilize uma ferramenta para criar uma **cópia minificada do seu código javascript** quando for colocá-lo em produção.

Outra ferramenta que pode reduzir o tamanho de seus arquivos em mais da metade é o **gzip**, reduzindo o tempo necessário para que o conteúdo seja baixado pelo usuário final.





Comente seu código

O código comentado facilita muito a comunicação do que está (ou deveria estar) acontecendo no programa, e permite que outros programadores possam navegar com menos trabalho em scripts que não são deles. Para aqueles que trabalham sozinhos, e acreditam que não precisam comentar seu próprio código: Você conseguiria pegar o código fonte de um programa complexo, escrito por você há 2 anos atrás e dizer de cara o que cada função faz?

```
</head>
<body>
<h1>Comentários em HTML e PHP </h1>
<?php
    echo "<h2>Vamos comentar o código PHP.</h2>
        <!-- <h2>Esse trecho não será exibido.</h2> -->
        <h2>Esse conteúdo será exibido!</h2>"
    /* Comentário que ocupa várias linhas.
    -----
    Bloco de código. */
    //echo <h2>Essa parte também não será exibida!</h2>
?>
</body>
</html>
•
```

DigitalHouse>
Coding School