



**INSTITUTO FEDERAL**  
Santa Catarina

Ministério da Educação  
Secretaria de Educação Profissional e Tecnológica  
**INSTITUTO FEDERAL DE SANTA CATARINA**

**INSTITUTO FEDERAL DE SANTA CATARINA - CHAPECÓ**

**Curso: Técnico em Informática Integrado ao Ensino Médio**

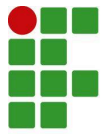
**Matéria: Tópicos Especiais**

**Docente: Alexandre Anderson dos Santos**

**SANDRO DE CASTRO, MATEUS TOMCZAK FAGUNDES, JOÃO VICTOR GILIOLI,  
RAFAEL ROSSA E VICTOR AZAMBUJA**

**Tópicos Especiais**

**16 de outubro de 2025**



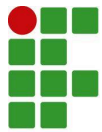
# Boas práticas com JavaScript

## **Use com consciência os comparadores de igualdade == e ===.**

Desenvolvedores com alguma experiência em linguagens cuja sintaxe seja semelhante à de JavaScript, como Java e C#, já estão acostumados a usar o operador == para comparação de igualdade. Todavia, apesar das semelhanças, em JavaScript o operador == possui um comportamento um pouco diferente. Para começar, em JS existem dois operadores distintos, o operador de igualdade duplo (==) e o triplo (===). A diferença entre esses consiste em que o operador triplo não realiza conversão de valores antes da comparação.

## **Sempre utilize o ponto e vírgula nos locais apropriados.**

Você já deve ter percebido que, embora a sintaxe de JavaScript exija o uso de ponto e vírgula ao final de uma expressão, na maioria das vezes, o seu código irá funcionar se você se esquecer de adicioná-los. Isso ocorre porque antes de executar o seu código o compilador de JS irá inserir ponto e vírgula nos locais onde julgar necessário. Essa característica pode levá-lo a um mau hábito que, em alguns casos, tende a gerar bugs difíceis de serem identificados.



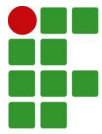
## **Evite declarar variáveis no escopo global**

Principalmente quando não se tem experiência em desenvolver grandes projetos em JavaScript, a tendência entre nós desenvolvedores é ir declarando e usando funções e objetos à medida que esses se fazem necessários. Dessa maneira, tem-se um código em que praticamente todas as funções e variáveis ficam acessíveis no escopo global. O problema com essa tendência é que, além de contribuir para a desorganização do código, ela torna o código extremamente vulnerável a ser afetado por outros trechos de código.

## **Tenha cuidado ao trabalhar com decimais**

Crie o hábito de utilizar o estilo de indentação K&R ou um de seus variantes.

Desenvolvedores acostumados a linguagens C-like costumam utilizar o estilo de indentação de Allman para definir escopos. Para os que não estão familiarizados com a nomenclatura, esse estilo coloca a chave de abertura do bloco de escopo em uma linha própria.



## **Atenção ao utilizar a variável this no JavaScript**

Diferentemente de linguagens que possuem o conceito de classe, como C# e Java, em JavaScript o valor da variável this é definido dinamicamente quando a função é executada, e não quando ela é definida. Além disso, em JS qualquer função, seja ela pertencente ou não a um objeto, pode utilizar a variável this. Assim, utilizar essa variável requer um pouco de atenção.

### **Referências:**

<https://medium.com/@gabrielmoraesmm/boas-pr%C3%A1ticas-para-programa%C3%A7%C3%A3o-web-html-e-css-c2a62f1b9560>

<https://www.dtidigital.com.br/blog/dicas-e-boas-praticas-com-javascript#Tenha-cuidado-ao-trabalhar-com-decimais>



**INSTITUTO FEDERAL**  
Santa Catarina

Ministério da Educação  
Secretaria de Educação Profissional e Tecnológica  
**INSTITUTO FEDERAL DE SANTA CATARINA**