



Certified Tech Developer

The Ultimate Degree

Javascript: Próximos passos

console.log()

Agora que conhecemos o console como parte das ferramentas que devemos usar durante o desenvolvimento, vamos colocar algumas linhas de código em prática.

O clássico seria mostrar uma mensagem por console, para isso devemos implementar o método **log()** do objeto **Console**. Então, como seria?

```
console.log("Esta é uma mensagem por console");
```

Podemos tentar executar a mesma instrução com mensagens diferentes, mesmo com tipos de dados diferentes, e devemos ter o mesmo resultado. Por exemplo:

```
console.log("Olá, sou outro texto!");  
console.log(25);  
console.log(true);
```

🌟 É sempre importante colocar seu código em prática, experimentar diferentes possibilidades e não se preocupar com bugs. Além disso, o aparecimento de erros é fundamental para o processo de aprendizagem, se não os encontrarmos dificilmente estaremos exigindo o suficiente de nós mesmos.

Variantes

Como alternativas ao **log()**, mas raramente implementado, é bom saber que existem outros métodos, tais como:

.error() → Escreva um erro no console.

.warn() → Escreva um aviso no console.

.table() → Escreva uma tabela no console.

O último é outra maneira de exibir um objeto ou array no console de uma forma um pouco mais visual. Por exemplo:

```
let meuObjeto = {  
  mensagem: "Mensagem de texto",  
  utilidade: "Prova de JS"  
};  
  
let meuArray = [ "Primeira mensagem do array", "Segunda mensagem  
do array", "Terceira mensagem do array"]  
  
console.table(meuObjeto);  
console.table(meuArray);
```

Continue tentando alternativas, familiarize-se com o console do navegador e implemente o código para obter resultados diferentes. Você também pode realizar **operações aritméticas e relacionais**.

```
let numero = 5;  
let numeroEmLetras = "5";  
  
console.log(numero == numeroEmLetras);  
console.log(numero === numeroEmLetras);
```

alert()

Assim como exibimos mensagens por console, que são muito úteis para depurar nosso código, também temos **mensagens de alerta que o usuário pode ver**.

O método alert() pertence ao objeto window, mas para usá-lo podemos implementá-lo diretamente no console.

```
alert("Isso é um alerta.");
```

O método mostra uma caixa de alerta, com a mensagem que passamos como parâmetro, e um botão OK. Ele é implementado precisamente para mostrar ao usuário

certas informações que consideramos importantes. Desta forma simples e rápida já podemos nos comunicar com o usuário.

Não é a interface mais bonita do mundo, mas pelo menos assim podemos mostrar mensagens mais práticas.

📌 Estes são os métodos mais simples, talvez isolados não nos pareçam muito úteis, mas depois os complementaremos com outros que darão funcionalidade ao nosso código.