

artilheiro.js:

Marcando Golaços com JavaScript



Aprenda todo o
básico
de JavaScript
com futebol!

Funções, Arrays,
DOMs e objetos,
aprenda tudo
neste eBook.

Arthur Guilherme

Introdução

Aprender JS com futebol

Aprender *JavaScript* pode parecer difícil no início, mas e se a gente misturar isso com futebol?

Neste eBook, você vai dar os primeiros toques no mundo da programação usando o futebol como inspiração. Cada conceito técnico é explicado de forma simples, leve e com exemplos do universo da bola.

Vista a camisa, entre em campo e comece sua jornada no JavaScript Futebol Clube!





O juiz apitou e o jogo começou!
Mas afinal, o que é esse tal de
JavaScript?

Apito Inicial

O Que é JavaScript?

JavaScript é a linguagem que faz a *web* acontecer. Sabe quando aparece um placar atualizado no site do Brasileirão?

Provavelmente tem *JavaScript* ali. É uma linguagem de programação que roda no navegador e permite interações com o usuário. Vamos ver um exemplo básico:



```
futebol_clube.js
```

```
console.log("A bola está rolando!");
```

Esse comando escreve no console do navegador. É como se fosse o narrador anunciando o início da partida.



VARIÁVEIS

Vamos escalar nosso time. Cada jogador precisa de nome, posição e número.

Variáveis

Escalando os Titulares

Variáveis são usadas para guardar informações. Pense nelas como jogadores no seu time — cada um tem um nome e uma função específica.

```
futebol_clube.js

let jogador = "Vinícius Jr.";
let numero = 7;
let posicao = "Atacante";

console.log(jogador + " é o camisa " + numero + " e joga como " + posicao);
```

Você também pode mudar esses valores durante o jogo, como um técnico trocando a estratégia

```
jogador = "Rodrygo";
numero = 10;
```



CONDIÇÕES

Se foi falta, o juiz apita. Se não foi,
segue o jogo!

Condições

Regras da Partida

Na programação, às vezes precisamos tomar decisões. O *if* funciona como o juiz: ele avalia a jogada e aplica a regra.

```
futebol_clube.js

let faltas = 4;

if (faltas >= 5) {
  console.log("Cartão vermelho!");
} else if (faltas === 4) {
  console.log("Cartão amarelo!");
} else {
  console.log("Segue o jogo!");
}
```

Aqui, o código avalia e dá o cartão amarelo. 4 faltas resulta em amarelo. 5, é vermelho. Simples, direto, e com lógica.

Essas condições tornam seu código inteligente.



LOOPS

Todo bom jogador treina até
dominar a jogada.

Loops

Treinando até Acertar

Use *loops* para repetir algo várias vezes!

```
futebol_clube.js

for (let i = 1; i <= 5; i++) {
  console.log("Chute número " + i);
}
```

Esse código simula um treino com 5 finalizações. Cada vez que roda, o valor de *i* aumenta, e a mensagem muda.

Também dá pra usar o *while*, pra repetir até alcançar uma condição:

```
let gols = 0;

while (gols < 3) {
  gols++;
  console.log("Gol número " + gols);
}
```



FUNÇÕES

Cada jogada ensaiada tem um nome. E quando você chama, ela acontece.

Funções

Jogadas Ensaaiadas

Funções são como jogadas ensaiadas. Você monta uma estratégia com nome e depois só chama quando quiser.

```
futebol_clube.js

function fazerGol(jogador) {
  console.log(jogador + " marcou um golão!");
}
```

Você pode até criar funções com mais lógica dentro:

```
function avaliarFalta(numeroFaltas) {
  if (numeroFaltas >= 5) {
    console.log("Expulsão!");
  } else {
    console.log("Continua no jogo.");
  }
}
```



OBJETOS

Cada jogador tem seus dados:
nome, idade, posição. Vamos
organizar isso!

Objetos

Ficha Técnica dos Craques

Objetos permitem agrupar várias informações de uma só vez, como se fosse uma ficha técnica completa.

```
let jogador = {  
  nome: "Alexia Putellas",  
  idade: 30,  
  posicao: "Meia",  
  gols: 12  
};  
  
console.log(jogador.nome + " tem " + jogador.idade + " anos e marcou " + jogador.gols + " gols.");
```

Você também pode acessar e alterar os dados:

```
jogador.gols += 1;  
console.log("Novo total de gols: " + jogador.gols);
```



ARRAYS

Temos que guardar os nomes dos times que estão no campeonato!

Arrays

A Tabela do Campeonato

Arrays são listas. Servem pra guardar coleções de dados, como nomes de times, jogadores ou artilheiros.

```
let times = ["Flamengo", "Palmeiras", "Grêmio", "Botafogo"];  
console.log("Time da rodada: " + times[2]); // Grêmio
```

Você também pode adicionar ou remover elementos:

```
times.push("Atlético-MG");  
times.splice(1, 1); // Remove Palmeiras
```



EVENTOS

Quando o jogador toca na bola,
algo acontece. No site, é igual!

Eventos

Tocou na Bola, Ação!

Eventos permitem interagir com o usuário. Um clique, um toque, uma ação — e o JavaScript responde.



```
<button onclick="chutar()">Chutar ao gol</button>

<script>
  function chutar() {
    console.log("GOOOOOOL!");
  }
</script>
```

Você pode fazer algo acontecer ao clicar, arrastar, digitar, etc. Isso é essencial pra deixar seu site com cara de jogo interativo.



DOM

O *DOM* é o campo onde os jogadores (elementos) se movimentam.

DOM

Controlando o Campo de Jogo

O *DOM* é como o campo do seu código. Com ele, o *JavaScript* pode mudar textos, estilos, imagens e mais, direto no *HTML*.

```
<p id="placar">Placar: 0</p>
<button onclick="marcarGol()">Marcar Gol</button>

<script>
  function marcarGol() {
    document.getElementById("placar").innerText = "Placar: 1";
  }
</script>
```

Dá até pra montar placares que somam pontos, mudar cor de fundo após um gol ou esconder elementos do campo. Um verdadeiro controle tático!



FIM DE JOGO

O jogo acabou, mas a temporada continua. Bora treinar pro próximo nível!

Fim de Jogo

Próximas Etapas

Você passou pelas principais posições do *JavaScript*: variáveis, condições, funções, objetos, eventos, e *DOM*. Com essa base, já dá pra criar placares dinâmicos, jogos simples e sistemas interativos.

```
let gols = 0;

function marcar() {
  gols++;
  console.log("GOL! Total: " + gols);
}
```

Esse é só o começo. Em breve, você pode partir para *APIs*, animações, e até *apps*.

Continue treinando. O craque do JS é você!



AGRADECIMENTOS

Este eBook foi gerado por IA, e diagramado por humano.

Esse conteúdo foi gerado com fins didáticos de construção, não foi realizado uma validação cuidadosa humana no conteúdo e pode conter erros gerados por uma IA.

AUTOR:

Arthur Guilherme

[GitHub](#) | [LinkedIn](#) | [Instagram](#)