

Conceitos avançados de JavaScript

Ana Júlia Rita Lima Cardoso

Outubro de 2025

1 Funções em JavaScript

1.1 O que são?

As funções são blocos de código reutilizáveis que executam uma tarefa específica. Servem para organizar e “esconder” instruções que compartilham um objetivo comum, tornando o código mais limpo e modular.

Exemplo sem função:

```
console.log("Olá, Ana!");  
console.log("Olá, João!");  
console.log("Olá, Pedro!");
```

Com função:

```
function cumprimentar(nome) {  
  console.log("Olá, " + nome + "!");  
}  
  
cumprimentar("Ana");  
cumprimentar("João");  
cumprimentar("Pedro");
```

1.2 Reuso de código

A principal vantagem das funções é o reuso. Em vez de repetir a mesma lógica em várias partes do código, basta chamar a função.

```
function calcularArea(base, altura) {  
    return base * altura;  
}  
  
let a1 = calcularArea(5, 10);  
let a2 = calcularArea(7, 3);  
  
console.log(a1, a2); // 50, 21
```

1.3 Como definir funções

A forma mais comum é com a palavra-chave `function`:

```
function nomeDaFuncao(param1, param2) {  
    // corpo da função  
    return valor;  
}
```

Exemplo:

```
function somar(a, b) {  
    return a + b;  
}  
  
let resultado = somar(3, 4);  
console.log("Resultado:", resultado);
```

1.4 Escopo

O escopo define onde uma variável pode ser acessada. Variáveis declaradas dentro de uma função são locais a ela.

```
function exemplo() {  
    let x = 10; // escopo local  
    console.log(x);  
}  
  
exemplo();  
console.log(x); // ERRO
```

1.5 Funções internas do JavaScript: `getDay()` e `getMonth()`

Essas funções pertencem ao objeto `Date` e retornam informações sobre a data atual.

```
const hoje = new Date();

console.log("Dia da semana:", hoje.getDay()); // 0 a 6
console.log("Mês:", hoje.getMonth());         // 0 a 11
```

1.6 Arrow Functions

Forma moderna e simplificada de criar funções.

```
const dobrar = (n) => n * 2;
console.log(dobrar(5)); // 10
```

Equivalente a:

```
function dobrar(n) {
  return n * 2;
}
```

1.7 Exemplos práticos

```
function exibirDataAtual() {
  const data = new Date();
  const dia = data.getDate();
  const mes = data.getMonth() + 1;
  const ano = data.getFullYear();
  console.log(`${dia}/${mes}/${ano}`);
}
```

```
exibirDataAtual();
```

```
const saudacao = (nome) => `Bem-vindo, ${nome}!`;
console.log(saudacao("Annelise"));
```

2 Eventos em JavaScript

2.1 O que são eventos?

Um **evento** é toda ação executada por um visitante ou pelo navegador enquanto a página está aberta. O JavaScript permite “ouvir” (monitorar) esses eventos e reagir a eles por meio de funções chamadas **manipuladores de eventos** (*event handlers*).

- Clicar com o mouse;
- Pressionar ou soltar uma tecla;
- Mover o cursor;
- Redimensionar a janela;
- Carregar a página.

Em resumo:

Um evento é uma **ação**, e uma função associada a ele é a **reação**.

2.2 Estrutura geral de um evento

Há duas formas principais de associar eventos a elementos HTML:

a) Atributo HTML (forma simples)

```
<button onclick="mensagem()">Clique aqui</button>
<script>
  function mensagem() {
    alert("Você clicou no botão!");
  }
</script>
```

b) Método JavaScript moderno: `addEventListener()`

```
<button id="btn">Clique aqui</button>
<script>
  const botao = document.getElementById("btn");
  botao.addEventListener("click", function() {
    alert("Evento capturado com addEventListener!");
  });
</script>
```

Diferença: `addEventListener` é mais moderno e permite associar vários eventos ao mesmo elemento sem sobrescrever outros.

2.3 Eventos do Mouse

Os eventos do mouse são disparados quando o usuário interage com o cursor na página.

| Evento | Descrição |
|-------------|---|
| onclick | Quando o elemento é clicado |
| ondblclick | Clique duplo |
| onmousedown | Quando o botão do mouse é pressionado |
| onmouseup | Quando o botão é solto |
| onmouseover | Quando o cursor entra na área do elemento |
| onmouseout | Quando o cursor sai da área do elemento |

Exemplo:

```
<button id="teste">Passe o mouse</button>
<script>
  const botao = document.getElementById("teste");

  botao.onmouseover = () => botao.innerText = "Mouse em cima!";
  botao.onmouseout  = () => botao.innerText = "Mouse saiu!";
  botao.onclick     = () => alert("Clicou!");
</script>
```

Dica: Quando o evento é acionado, o navegador cria um objeto `event` com informações como posição do cursor e botão pressionado.

2.4 Eventos do Teclado

Permitem detectar quando o usuário pressiona, mantém ou solta uma tecla. Muito usados em formulários, atalhos de teclado e jogos.

| Evento | Quando ocorre |
|------------|--|
| onkeydown | Quando uma tecla é pressionada |
| onkeypress | Enquanto a tecla é mantida pressionada |
| onkeyup | Quando a tecla é liberada |

Propriedades do objeto `event`:

| Propriedade | Descrição | Exemplo |
|-------------|-----------------------------------|-----------------|
| key | Nome da tecla pressionada | "a", "Enter" |
| code | Código físico da tecla | "KeyA", "Space" |
| altKey | true se ALT estiver pressionado | |
| ctrlKey | true se CTRL estiver pressionado | |
| shiftKey | true se SHIFT estiver pressionado | |

Exemplo:

```
<input id="campo" placeholder="Digite algo..." />

<script>
  const campo = document.getElementById("campo");

  campo.addEventListener("keydown", (event) => {
    console.log("Tecla pressionada:", event.key);
    console.log("Código:", event.code);
    console.log("Shift?", event.shiftKey);
  });
</script>
```

2.5 Eventos da Janela (window)

Esses eventos estão associados à própria janela do navegador.

| Evento | Quando ocorre |
|----------|-------------------------------------|
| onload | Quando a página termina de carregar |
| onresize | Quando a janela é redimensionada |

Exemplo:

```
<script>
  window.onload = () => {
    console.log("Página carregada com sucesso!");
  };

  window.onresize = () => {
    console.log("Janela redimensionada:", window.innerWidth, "x", window.innerHeight);
  };
</script>
```

2.6 Exemplo completo integrando eventos de mouse, teclado e janela

```
<button id="botao">Clique ou pressione Enter</button>

<script>
  const botao = document.getElementById("botao");

  // Evento do mouse
  botao.addEventListener("click", () => alert("Clicou no botão!"));
```

```
// Evento do teclado
document.addEventListener("keydown", (e) => {
  if (e.key === "Enter") {
    alert("Você pressionou Enter!");
  }
});

// Evento da janela
window.onload = () => console.log("Página carregada!");
</script>
```

2.7 Resumo e boas práticas

1. Use `addEventListener()` sempre que possível — é mais moderno e flexível.
2. Prefira nomear funções separadas em vez de escrever tudo dentro do evento.
3. Use `event` para acessar detalhes do que ocorreu (tecla, posição do mouse, etc.).
4. Combine eventos para criar interações dinâmicas e acessíveis.

Um bom uso de eventos torna sua página viva, interativa e inteligente.

3 Atividades Práticas — Funções e Eventos em JavaScript

3.1 Atividade 1 — Contador Interativo

Objetivo: Praticar o uso de funções e eventos do mouse para atualizar elementos na tela de forma dinâmica.

Instruções:

1. Crie uma página HTML com um número inicial igual a 0.
2. Adicione dois botões: “Aumentar” e “Diminuir”.
3. Cada botão deve chamar uma função JavaScript diferente ao ser clicado.
4. A função deve atualizar o número exibido na tela.

Desafio:

- Impedir que o contador fique negativo.
- Mudar a cor do número conforme o valor (verde para positivo, vermelho para negativo, preto para zero).

3.2 Atividade 2 — Exibir Data e Hora

Objetivo: Trabalhar com funções internas do objeto `Date` e o evento `onload`.

Instruções:

1. Ao carregar a página, exibir na tela a data e a hora atuais.
2. Utilize os métodos `getDay()`, `getMonth()`, `getFullYear()` e `getHours()`.
3. Formate o texto de forma legível, como: “Hoje é quarta-feira, 30 de outubro de 2025 — 14:35”.

Desafio:

- Atualizar automaticamente a hora a cada segundo com `setInterval()`.

3.3 Atividade 3 — Detectando Teclas Pressionadas

Objetivo: Usar eventos de teclado para capturar informações do objeto `event`.

Instruções:

1. Crie um campo de texto (`<input>`) e associe a ele um evento de teclado.
2. Sempre que o usuário pressionar uma tecla, exiba:
 - O nome da tecla (`event.key`);
 - O código físico da tecla (`event.code`);
 - Se as teclas ALT, CTRL ou SHIFT estavam pressionadas.
3. Mostre as informações em um parágrafo ou caixa HTML.

Desafio:

- Fazer a cor de fundo da página mudar conforme certas teclas forem pressionadas (ex.: B → azul, R → vermelho, W → branco).

3.4 Atividade 4 — Caixinha Mágica (Eventos do Mouse)

Objetivo: Compreender os principais eventos do mouse (`onclick`, `onmouseover`, `onmouseout`) e manipular elementos com JavaScript.

Instruções:

1. Crie uma `<div>` de 200x200 pixels com uma cor de fundo.
2. Ao passar o mouse sobre a caixa, mude a cor.
3. Ao tirar o mouse de cima, volte à cor original.
4. Ao clicar na caixa, exiba uma mensagem com `alert()`.

Desafio:

- Fazer a cor da caixa mudar aleatoriamente a cada vez que o mouse entra.

3.5 Atividade 5 — Redimensionamento da Janela

Objetivo: Usar eventos da janela (`onload` e `onresize`) para exibir informações dinâmicas sobre o navegador.

Instruções:

1. Crie uma página que exiba a largura e a altura atuais da janela do navegador.
2. Atualize automaticamente os valores quando a janela for redimensionada.

Desafio:

- Exibir também uma mensagem de boas-vindas ao carregar a página (`window.onload`).

3.6 Atividade Bônus — Painel Interativo

Objetivo: Integrar os conceitos de funções, eventos do mouse, teclado e janela.

Instruções:

1. Crie uma página com:
 - Um campo de texto para o nome do usuário;
 - Um botão “Entrar”;
 - Uma caixa colorida (`<div>`) para interações.

2. Ao clicar no botão:
 - Exibir uma saudação personalizada com o nome digitado;
 - Mostrar a data e hora atuais.
3. Ao passar o mouse na caixa, mude a cor de fundo;
4. Ao clicar na caixa, exiba uma mensagem;
5. Ao redimensionar a janela, exiba as novas dimensões;
6. Ao pressionar certas teclas, altere a cor de fundo da página.

Desafio:

- Atualizar a hora automaticamente a cada segundo.
- Escolher um esquema de cores temático (ex.: modo dia/noite).