Revisão de Sintaxe e Fundamentos de JavaScript

Nivelamento

Ana Júlia Lima

Desenvolvido para
Young 1

Ctrl Play Santa Mônica
Vila Velha, Brasil
Fevereiro 2026

Sumário

1 Revisar as sintaxes de JavaScript

1.1 Introdução

JavaScript (JS) é uma linguagem usada para dar **vida** às páginas: botões que funcionam, formulários que validam, animações e interações. Também pode ser usada no **back-end** (Node.js), em apps e até em jogos simples.

Nesta apostila, você vai revisar:

- Variáveis e tipos básicos;
- Operadores;
- Funções;
- Condicionais;
- Loops;
- Eventos (principalmente no navegador).

1.2 Como rodar JavaScript (3 formas simples)

- **Console do navegador:** aperte F12, aba "Console" e digite JS.
- **Arquivo HTML + JS:** crie um `index.html` e linke um `script.js`.
- **Node.js:** rode no terminal com `node arquivo.js`.

1.2.1 Exemplo HTML + JS

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Teste JS</title>
</head>
<body>
  <button id="btn">Clique</button>

  <script src="script.js"></script>
</body>
</html>

// script.js
console.log("Ola, JavaScript!");
```

2 Variáveis: var, let e const

2.1 O que é uma variável?

Variável é um **nome** que guarda um valor. Esse valor pode mudar (ou não) dependendo de como você declara a variável.

2.2 Regras simples para nomear variáveis

- Use **camelCase**: `nomeCompleto`, `totalCarrinho`.
- Não comece com número: `1nome` é inválido.
- Não use espaços: `nome completo` é inválido.
- Use nomes que expliquem o que é: `idadeAluno` é melhor que `x`.

2.3 var (legado / evitar)

`var` é antigo. Funciona, mas pode gerar confusão porque:

- Tem **escopo de função** (não respeita blocos como `if`).
- Pode causar bugs em loops e em código maior.

```
function exemploVar() {
  if (true) {
    var x = 10;
  }
  console.log(x); // 10 (mesmo fora do if)
}

exemploVar();
```

2.4 let (flexível)

`let` é a escolha quando você **precisa reatribuir** (mudar o valor).

- Respeita **escopo de bloco**.
- É moderno e recomendado.

```
let idade = 18;
idade = 19; // ok

if (true) {
  let nome = "Ana";
  console.log(nome); // Ana
}

// console.log(nome); // erro: nome nao existe fora do bloco
```

2.5 const (estável / recomendado por padrão)

`const` é a escolha padrão:

- Você não pode **reatribuir** (trocar) o valor da variável.
- Ainda assim, em objetos e arrays, você pode alterar o **conteúdo**.

```
const curso = "JavaScript";
// curso = "Python"; // erro

const numeros = [1, 2, 3];
numeros.push(4); // ok: alterou o conteudo
console.log(numeros); // [1,2,3,4]
```

2.6 Regra prática (bem simples)

- Use **const** sempre.
- Use **let** apenas se precisar mudar o valor.
- Evite **var**.

3 Tipos de dados (básico e essencial)

3.1 Principais tipos em JS

- **Number**: números inteiros e decimais
- **String**: textos
- **Boolean**: `true` ou `false`
- **null**: "vazio intencional"
- **undefined**: "não definido ainda"
- **Object**: objetos, arrays, funções

```
let idade = 20;             // Number
const nome = "Leon";          // String
const aprovado = true;        // Boolean
const vazio = null;           // null
let semValor;                 // undefined
const pessoa = { nome: "Ana", idade: 19 }; // Object
```

3.2 typeof (descobrir o tipo)

```
console.log(typeof 10);        // "number"
console.log(typeof "oi");      // "string"
console.log(typeof true);      // "boolean"
console.log(typeof undefined);// "undefined"
console.log(typeof null);      // "object" (curiosidade do JS)
```

4 Operadores

4.1 O que são operadores? Para que servem?

Operadores são símbolos usados para:

- fazer contas,
- comparar valores,
- juntar condições,
- atribuir (guardar) valores em variáveis.

4.2 Aritméticos

- Adição: `+`
- Subtração: `-`
- Multiplicação: `*`
- Divisão: `/`
- Módulo: `%` (resto)
- Exponenciação: `**`

```
const a = 10;
const b = 3;

console.log(a + b); // 13
console.log(a - b); // 7
console.log(a * b); // 30
console.log(a / b); // 3.333...
console.log(a % b); // 1  (resto)
console.log(a ** b); // 1000
```

4.3 Atribuição

- Simples: `=`
- Soma: `+=`
- Subtração: `-=`
- Multiplicação: `*=`
- Divisão: `/=`
- Módulo: `%=`

```
let pontos = 10;
pontos += 5;  // 15
pontos -= 2;  // 13
pontos *= 2;  // 26
pontos /= 2;  // 13
pontos %= 5;  // 3
console.log(pontos);
```

4.4 Comparação (muito importante)

- Igualdade: `==` (evitar)
- Igualdade estrita: `===` (recomendado)
- Desigualdade: `!=` (evitar)
- Desigualdade estrita: `!==` (recomendado)
- `> >= < <=`

4.4.1 Por que evitar ==?

Porque == tenta **converter** tipos e pode enganar.

```
console.log(2 == "2");   // true  (converte string para numero)
console.log(2 === "2");  // false (compara tipo e valor)
```

4.5 Lógicos

- AND: `&&` (as duas condições precisam ser verdade)
- OR: `||` (basta uma condição ser verdade)
- NOT: `!` (inverte)

```
const temIngresso = true;
const estaComDocumento = false;

console.log(temIngresso && estaComDocumento); // false
console.log(temIngresso || estaComDocumento); // true
console.log(!temIngresso); // false
```

4.6 Operador ternário (?)

É um `if/else` curtinho:

```
const idade = 17;
const msg = (idade >= 18) ? "Maior de idade" : "Menor de idade";
console.log(msg);
```

papapapapa olhar aqui

5 Condicionais

5.1 O que são condicionais? Para que servem?

Condicionais servem para tomar decisões:

- se algo for verdadeiro, faça uma coisa,
- senão, faça outra.

5.2 if, else if, else

```
const nota = 6;

if (nota >= 7) {
  console.log("Aprovado");
} else if (nota >= 5) {
  console.log("Recuperacao");
} else {
  console.log("Reprovado");
}
```

5.3 Boas práticas em condicionais

- Use `===` e `!==` sempre que possível.
- Deixe as condições claras (sem exagerar na complexidade).

5.3.1 Exemplo: validar login

```
const usuarioCorreto = "ana";
const senhaCorreta = "1234";

const usuario = "ana";
const senha = "1234";

if (usuario === usuarioCorreto && senha === senhaCorreta) {
  console.log("Login permitido");
} else {
  console.log("Usuario ou senha incorretos");
}
```

6 Loops (repetição)

6.1 O que são loops? Para que servem?

Loops repetem um bloco de código para:

- contar repetições (1 a 10),
- percorrer listas (arrays),
- repetir até algo acontecer (senha correta).

6.2 for (quando você sabe a quantidade)

```
for (let i = 1; i <= 5; i++) {
  console.log("Numero:", i);
}
```

6.2.1 for percorrendo array

```
const nomes = ["Ana", "Leon", "Andy", "Saphirra"];

for (let i = 0; i < nomes.length; i++) {
  console.log(nomes[i]);
}
```

6.3 while (enquanto for verdadeiro)

```
let energia = 3;

while (energia > 0) {
  console.log("Energia:", energia);
  energia--;
}

console.log("Acabou a energia!");
```

6.4 do...while (executa pelo menos uma vez)

```
let tentativas = 0;

do {
  tentativas++;
```

```
  console.log("Tentativa:", tentativas);
} while (tentativas < 3);
```

6.5 Dica: break e continue

- `break`: para o loop
- `continue`: pula para a próxima repetição

```
for (let i = 1; i <= 5; i++) {
  if (i === 3) continue; // pula o 3
  if (i === 5) break;    // para no 5
  console.log(i);
}
```

7 Funções

7.1 O que é uma função? Para que serve?

Funções são "máquinas":

- recebem entradas (parâmetros),
- fazem um processamento,
- devolvem uma saída (return).

7.2 Estrutura básica

```
function soma(a, b) {
  return a + b;
}

console.log(soma(2, 3)); // 5
```

7.3 Parâmetros e retorno

```
function calcularMedia(n1, n2, n3) {
  const media = (n1 + n2 + n3) / 3;
  return media;
}

console.log(calcularMedia(10, 8, 7)); // 8.333...
```

7.4 Arrow function (muito usada no JS moderno)

```
const dobro = (n) => n * 2;

console.log(dobro(5)); // 10
```

7.5 Funções com valores padrão

```
function saudacao(nome = "Pessoa") {
  console.log("Ola,", nome);
}

saudacao("Ana");
saudacao();
```

8 Padrão de estilo do JavaScript (ESLint / Airbnb / Prettier)

8.1 Qual é o "PEP8" do JavaScript?

JavaScript não tem um único padrão oficial, mas é MUITO comum usar ferramentas:

- **ESLint**: encontra erros e aplica regras de estilo/boas práticas.
- **Airbnb**: um conjunto famoso de regras para ESLint.
- **Prettier**: formata o código automaticamente (espaços, quebra de linha).

8.2 Regras práticas (para iniciante)

- Use `const` por padrão.
- Use `let` só quando precisar mudar o valor.
- Use `===` e `!==`.
- Nomes claros e consistentes.

9 Eventos

9.1 O que são eventos? Para que servem?

Eventos são ações do usuário ou do navegador:

- clique,
- digitar,
- mover o mouse,
- enviar formulário,
- carregar a página.

JavaScript consegue "escutar" eventos e responder com uma função.

9.2 Exemplo 1: evento de click

```
const botao = document.querySelector("#btn");

botao.addEventListener("click", () => {
  console.log("Clicou no botao!");
});
```

9.3 Exemplo 2: evento de input (pegar o que a pessoa digita)

```
const campo = document.querySelector("#nome");

campo.addEventListener("input", () => {
  console.log("Digitando:", campo.value);
});
```

9.4 Exemplo 3: submit (não recarregar a página)

```
const form = document.querySelector("#formulario");

form.addEventListener("submit", (event) => {
  event.preventDefault(); // impede recarregar
  console.log("Formulario enviado!");
});
```

9.5 Eventos comuns

- click
- input
- change
- submit
- keydown
- load

10 Lista de exercícios e nivelamento

10.1 Variáveis

1. Crie `const` para `nome` e `cidade`. Crie `let` para `idade` e incremente 1.
2. Crie um array com 5 nomes e imprima todos usando `for`.
3. Crie um objeto `pessoa` com `nome` e `idade`. Imprima uma frase usando esses dados.

10.2 Operadores

1. Dado `a=20` e `b=6`, mostre soma, subtração, multiplicação, divisão e módulo.
2. Faça um código que diga se `n` é múltiplo de 3.
3. Compare `"10"` e `10` com `==` e `===` e explique (em comentário) por que deu diferente.

10.3 Condicionais

1. Leia uma nota e retorne: aprovado (¿=7), recuperação (¿=5), reprovado (¡5).
2. Faça um código que diga se um número é par ou ímpar.
3. Use operador ternário para retornar `"Pode"` ou `"Nao pode"` dependendo de uma condição.

10.4 Loops

1. Imprima os números de 1 a 50 pulando os múltiplos de 5.
2. Some os números de 1 a 100 usando `while`.
3. Faça um loop que procura um nome dentro de um array e pare quando encontrar (`break`).

10.5 Funções

1. Crie `somar(a,b)`, `subtrair(a,b)`, `multiplicar(a,b)`, `dividir(a,b)`.
2. Crie `ehPar(n)` que retorna `true` ou `false`.
3. Crie uma função que receba um array de notas e retorne a média.

10.6 Eventos

1. Crie um botão e faça aparecer um `alert("Ola!")` ao clicar.
2. Crie um input e mostre o texto digitado dentro de uma `div` na tela.
3. Crie um formulário e valide se o campo nome está vazio antes de “enviar”.

11 Gabarito — Soluções dos Exercícios

11.1 Variáveis

1) Const nome e cidade; let idade e incremente 1

```
const nome = "Ana";
const cidade = "Sao Paulo";

let idade = 18;
idade++;

console.log(nome);
console.log(cidade);
console.log(idade); // 19
```

2) Array com 5 nomes usando for

```
const nomes = ["Ana", "Leon", "Andy", "Saphirra", "Nico"];

for (let i = 0; i < nomes.length; i++) {
  console.log(nomes[i]);
}
```

3) Objeto pessoa

```
const pessoa = {
  nome: "Ana",
  idade: 19
};

console.log(`A pessoa se chama ${pessoa.nome} e tem ${pessoa.idade} anos.`);
```

11.2 Operadores

1) Operações com a=20 e b=6

```
const a = 20;
const b = 6;

console.log("Soma:", a + b);
console.log("Subtracao:", a - b);
console.log("Multiplicacao:", a * b);
console.log("Divisao:", a / b);
console.log("Modulo:", a % b);
```

2) Verificar se n é múltiplo de 3

```
const n = Number(prompt("Digite um numero:"));

if (n % 3 === 0) {
  console.log("Multiplo de 3");
} else {
  console.log("Nao e multiplo de 3");
}
```

3) Comparação == e ===

```
console.log("10" == 10);
// true porque o == converte string para numero antes de comparar

console.log("10" === 10);
// false porque o === compara valor e tipo
```

11.3 Condicionais

1) Nota

```
const nota = Number(prompt("Digite um número: "));

if (nota >= 7) {
  console.log("Aprovado");
} else if (nota >= 5) {
  console.log("Recuperacao");
} else {
  console.log("Reprovado");
}
```

2) Par ou ímpar

```
const numero = Number(prompt("Digite um número: "));

if (numero % 2 === 0) {
  console.log("Par");
} else {
  console.log("Impar");
}
```

3) Operador ternário

```
const idade =  Number(prompt("Digite um número: "));

const resultado = (idade >= 18) ? "Pode" : "Nao pode";
console.log(resultado);
```

11.4 Loops

1) 1 a 50 pulando múltiplos de 5

```
for (let numero = 1; numero <= 50; numero++) {
  if (numero % 5 === 0) continue;
  console.log(i);
}
```

2) Soma de 1 a 100 usando while

```
let soma = 0;
let i = 1;

while (i <= 100) {
  soma += i;
  i++;
}
```

```
console.log("Soma:", soma); // 5050
```

3) Procurar nome com break

```
const lista = ["Ana", "Leon", "Andy", "Saphirra", "Nico"];
const procurado = "Andy";

for (let i = 0; i < lista.length; i++) {
  if (lista[i] === procurado) {
    console.log("Encontrado na posicao:", i);
    break;
  }
}
```

11.5 Funções

1) Operações básicas

```
function somar(a, b) {
  return a + b;
}

function subtrair(a, b) {
  return a - b;
}

function multiplicar(a, b) {
  return a * b;
}

function dividir(a, b) {
  return a / b;
}
```

2) Função ehPar

```
function ehPar(n) {
  return n % 2 === 0;
}
```

3) Média de notas

```
function mediaNotas(notas) {
  let soma = 0;

  for (let i = 0; i < notas.length; i++) {
    soma += notas[i];
  }

  return soma / notas.length;
}

const notas = [10, 8, 7];
console.log(mediaNotas(notas));
```

11.6 Eventos

1) Botão com alert

```
<button id="btn">Clique aqui</button>

const botao = document.querySelector("#btn");

botao.addEventListener("click", () => {
  alert("Ola!");
});
```

2) Input mostrando texto na tela

```
<input id="campo" type="text" />
<div id="saida"></div>

const campo = document.querySelector("#campo");
const saida = document.querySelector("#saida");

campo.addEventListener("input", () => {
  saida.textContent = campo.value;
});
```

3) Formulário validando campo vazio

```
<form id="form">
  <input id="nome" type="text" />
  <button type="submit">Enviar</button>
</form>
<p id="msg"></p>

const form = document.querySelector("#form");
const nomeInput = document.querySelector("#nome");
const msg = document.querySelector("#msg");

form.addEventListener("submit", (event) => {
  event.preventDefault();

  if (nomeInput.value.trim() === "") {
    msg.textContent = "Campo nome esta vazio.";
    return;
  }

  msg.textContent = "Formulario enviado com sucesso!";
});
```