📓 Glossário Completo de JavaScript para Iniciantes

Este glossário foi desenvolvido para ajudar **iniciantes** que desejam aprender JavaScript e começar a programar para a web. Ele contém explicações claras, exemplos de código e as principais funcionalidades dessa linguagem poderosa.

📌 O que é JavaScript?

JavaScript (**JS**) é uma linguagem de programação **interpretada** e **orientada a objetos**, usada principalmente para adicionar **interatividade** em páginas web.

Exemplo prático:

```
<button onclick="alert('Olá!')">Clique em mim</button>
```

📌 Como incluir JavaScript no HTML

```
<script>
  console.log("Olá, mundo!");
</script>
```

Ou de forma externa:

```
html
CopiarEditar
<script src="script.js"></script>
```

📌 Variáveis

Declaração de variáveis com `let`, `const` e `var`:

```
let nome = "Ana";
const idade = 30;
var cidade = "São Paulo";
```

- `const`: valor fixo (não muda)
- `let`: valor que pode ser alterado
- `var`: forma antiga (evitar)

📌 Tipos de Dados

Tipo	Exemplo
String	`"Olá"`
Number	`42`, `3.14`
Boolean	`true`, `false`
Array	`[1, 2, 3]`
Object	`{ nome: "Ana" }`
null	`null`
undefined	`undefined`

📌 Operadores

Tipo	Exemplo
Aritméticos	`+`, `-`, `*`, `/`
Comparação	`==`, `===`, `!=`
Lógicos	`&&`, `

📌 Estruturas Condicionais

```
let idade = 18;

if (idade >= 18) {
  console.log("Maior de idade");
} else {
  console.log("Menor de idade");
}
```

📌 Laços de Repetição

```
for (let i = 0; i < 5; i++) {
  console.log(i);
}

let i = 0;
while (i < 5) {
  console.log(i);
  i++;
}
```

📌 Funções

```
function saudacao(nome) {
```

```
  return "Olá, " + nome + "!";
}

console.log(saudacao("Carlos"));
```

📌 Funções Arrow (=>)

```
const soma = (a, b) => a + b;
```

📌 Arrays

```
const frutas = ["Maçã", "Banana", "Pera"];
console.log(frutas[0]); // Maçã
```

📌 Objetos

```
const pessoa = {
  nome: "Ana",
  idade: 25,
  falar: function () {
    console.log("Olá, meu nome é " + this.nome);
  },
};

pessoa.falar();
```

📌 Manipulação do DOM

```
document.querySelector("h1").innerText = "Novo título";
```

📌 Eventos

```
document.querySelector("button").addEventListener("click", () => {
  alert("Botão clicado!");
});
```

📌 JSON

```
const json = '{"nome":"Ana","idade":30}';
const obj = JSON.parse(json);
```

📌 Classes (Orientação a Objetos)

```
class Pessoa {
  constructor(nome) {
    this.nome = nome;
  }

  falar() {
    console.log("Olá, meu nome é " + this.nome);
  }
}

const p1 = new Pessoa("Carlos");
p1.falar();
```

📌 Promises e Assíncrono

```
const promessa = new Promise((resolve, reject) => {
  setTimeout(() => resolve("Tudo certo!"), 2000);
});

promessa.then((resultado) => console.log(resultado));
```

📌 Função Assíncrona (async/await)

```
async function buscarDados() {
  const resposta = await fetch("https://api.exemplo.com");
  const dados = await resposta.json();
  console.log(dados);
}
```

📌 Boas Práticas

 Nomear variáveis de forma clara
 Usar `const` e `let` (evitar `var`)
 Comentar o código quando necessário
 Utilizar funções para reutilizar código

📌 Relacionamento com HTML e CSS

- HTML → Estrutura
- CSS → Aparência
- JS → Comportamento (interações)