

Principais funcionalidades:

Variáveis: Declaradas com `let`, `const` ou `var` para armazenar dados.

Exemplo:

```
let x = 10;.
```

Funções: Blocos de código reutilizáveis. Definidas com `function` ou como `arrow functions`.

Exemplo:

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

Condicionais: Estruturas para tomar decisões. Usando `if`, `else if` e `else`.

Exemplo:

```
if (x > 5) { ... }.
```

Laços de repetição: Para iterar sobre dados. Usando `for`, `while` ou `forEach`.

Exemplo:

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

Objetos: Estruturas que armazenam pares chave-valor.

Exemplo:

```
let pessoa = { nome: 'João', idade: 30 };
```

Arrays: Listas ordenadas de valores.

Exemplo:

```
let numeros = [1, 2, 3];.
```

Manipulação de DOM: Interagindo com elementos HTML.

Exemplo:

```
document.getElementById('meuElemento').innerText = 'Olá!';.
```

Especificações:

Variáveis: As variáveis são usadas para armazenar dados temporários. Em JavaScript, podem ser declaradas com:

- `let`: permite a reatribuição de valores e é limitada ao bloco onde é definida.

exemplo:

```
let x = 10; // x pode ser reatribuído
```

- `const`: para declarar constantes, ou seja, variáveis que não podem ser reatribuídas.

exemplo:

```
const PI = 3.14; // PI não pode ser mudado
```

- `var`: tem escopo de função e pode ser reatribuído, mas seu uso é desencorajado devido a problemas de escopo.

exemplo:

```
var y = 5; // y pode ser reatribuído
```

Funções: Funções são blocos de código que podem ser reutilizados. Existem duas formas principais de defini-las:

- Declaração de função:

exemplo:

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

- Arrow functions: (funções de seta), que são uma forma mais concisa de escrever funções:

exemplo:

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

Condicionais: Estruturas que permitem a execução de diferentes blocos de código com base em condições:

- **if**: executa um bloco se a condição for verdadeira.

- **else if**: permite testar múltiplas condições.

- **else**: executa um bloco se nenhuma das condições anteriores for verdadeira.

exemplo:

```
if (x > 5) {  
    console.log('Maior que 5');  
} else if (x === 5) {  
    console.log('Igual a 5');  
} else {  
    console.log('Menor que 5');  
}
```

Laços de repetição: Usados para iterar sobre dados, permitindo a execução repetida de um bloco de código:

- **for**: executa um bloco por um número específico de vezes.

exemplo:

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

- **while**: executa enquanto a condição for verdadeira.

exemplo:

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

- **forEach**: um método de arrays que executa uma função para cada elemento do array.

exemplo:

```
let numeros = [1, 2, 3];
numeros.forEach(num => console.log(num));
```

Objetos: Estruturas que armazenam dados em pares chave-valor, permitindo representar entidades do mundo real:

exemplo:

```
let pessoa = {
  nome: 'João',
  idade: 30,
  cidade: 'São Paulo'
};
```

Acessa-se valores usando a notação de ponto ou colchetes:

exemplo:

```
console.log(pessoa.nome); // 'João'
```

Arrays: Listas ordenadas de valores, que podem ser de diferentes tipos e são manipuladas através de métodos:

exemplo:

```
let numeros = [1, 2, 3, 4, 5];
```

Pode-se acessar elementos pelo índice:

exemplo:

```
console.log(numeros[0]);
```

Manipulação de DOM: Permite interagir e modificar elementos HTML da página através do JavaScript:

exemplo:

```
document.getElementById('meuElemento').innerText = 'Olá!'; // Altera o texto do elemento com ID 'meuElemento'
```