

## ■ FUNDAMENTOS DE JAVASCRIPT – RESUMO COMPLETO (VITOR)

### ■ AULA 1 – SENTENÇAS E BLOCOS DE CÓDIGO

```
console.log('Hi guys'); // Sentença de código
```

```
{
{
{
{
console.log("Bloco de código 02");
console.log("Bloco de código"); // Padrão do curso
}
}
}
}
```

// Explicação:

Uma sentença é uma instrução completa que o JS executa.

As chaves { } definem um bloco de código — um agrupamento de sentenças.

Blocos são usados em funções, condicionais e loops.

### ■ AULA 2 – VARIÁVEIS E CONSTANTES

```
var a = 3;
let b = 4;
console.log(a, b);
```

```
const c = 5;
```

// Explicação:

var → antiga (escopo de função, evite usar)

let → moderna, pode mudar o valor

const → constante, valor fixo

Resumo:

var → função

let → bloco

const → bloco

### ■ AULA 3 – NÚMEROS

```
const peso1 = 1.0;
const peso2 = Number('2.0');
```

```
const avaliacao1 = 9.871;
const avaliacao2 = 6.871;
const total = avaliacao1 * peso1 + avaliacao2 * peso2;
const media = total / (peso1 + peso2);
```

```
console.log(media.toFixed(2));
console.log(media.toString(2));
console.log(typeof media);
```

```
console.log(typeof Number);
```

// Explicação:

Number() → converte texto em número

.toFixed(2) → mostra 2 casas decimais

.toString(2) → converte para binário

typeof → mostra o tipo do valor

---

## ■ AULA 4 – STRINGS

---

```
const escola = "vitOr";
```

```
console.log(escola.charAt(4));
console.log(escola.charAt(6));
console.log(escola.charCodeAt(0));
console.log(escola.indexOf('O'));
console.log(escola.substring(1));
console.log(escola.substring(0, 3));
console.log('Escola '.concat(escola).concat("!"));
console.log(escola.replace(3, 'v'));
console.log('Ana, Maria, Pedro'.split(','));
```

// Explicação:

charAt → retorna caractere

charCodeAt → código ASCII

indexOf → posição do caractere

substring → fatia a string

concat → junta strings

replace → substitui

split → divide string em array

---

## ■ AULA 5 – BOOLEANOS

---

```
let isAtivo = false;
console.log(isAtivo);
```

```
isAtivo = true;
console.log(isAtivo);
```

```
isAtivo = 1;
console.log(!isAtivo);
console.log(!isAtivo);
```

// Explicação:

! → negação

!! → converte em verdadeiro/falso lógico

Booleans servem para decisões (if/else).

---

## ■ AULA 6 – TEMPLATE STRINGS

---

```
const nome = 'vitor';
const concatenacao = 'Ola ' + nome + '!';
const template = `
```

```
ola
${nome}!
```

```
console.log(concatenacao, template);
console.log(`1 + 1 = ${1 + 1}`);
```

```
const up = texto => texto.toUpperCase();
console.log(`Ei.... ${up('cuidado')}!`);
```

// Explicação:  
Template strings usam crase (``) e permitem expressões \${}.  
Permitem múltiplas linhas e interpolação de variáveis.

---

## ■ AULA 7 – OBJETO MATH

---

```
const raio = 5.6;
const area = Math.PI * Math.pow(raio, 2);
console.log(area);
console.log(typeof Math);
```

// Explicação:  
Math é um objeto global com funções matemáticas.  
Math.PI → valor de  $\pi$   
Math.pow(base, expoente) → potência

---

## ■ AULA 8 – ARRAYS

---

```
const valores = [7.7, 8.9, 6.3, 9.2];
console.log(valores[0], valores[3]);
console.log(valores[4]); // undefined
```

```
valores[4] = 10;
console.log(valores);
console.log(valores.length);
```

```
valores.push({id: 3}, false, null, 'teste');
console.log(valores);
```

```
console.log(valores.pop());
delete valores[0];
console.log(valores);
```

```
console.log(typeof valores);
```

// Explicação:  
Arrays armazenam listas de valores.  
.push() → adiciona  
.pop() → remove o último  
.length → tamanho  
delete → apaga elemento (deixa vazio)

---

## ■ AULA 9 – OBJETOS

---

```
const prod1 = {};  
prod1.nome = 'Celular Ultra Mega';  
prod1.preco = 15000.90;  
prod1['Desconto Legal'] = 0.40;
```

```
const prod2 = {  
  nome: 'Camisa Polo',  
  preco: 79.90,  
  obj: {  
    blablabla: 1,  
    obj: {  
      blablabla: 2  
    }  
  }  
}  
console.log(prod1);  
console.log(prod2);
```

// Explicação:  
Objetos armazenam dados em pares chave:valor.  
Acesso por . ou ['chave']  
Podem conter outros objetos dentro (aninhados).  
typeof objeto → "object"