

The image features the word "JAVASCRIPT" in a bold, blue, sans-serif font, centered horizontally. The text is set against a white background. On the left and right sides of the text, there are large, curved, blue shapes that resemble stylized swooshes or the edges of a sphere, creating a dynamic and modern frame for the word.

JAVASCRIPT

TIPOS DE VARIÁVEIS BÁSICOS:

STRING: 'TEXT0', '123'

NUMBER : 123456

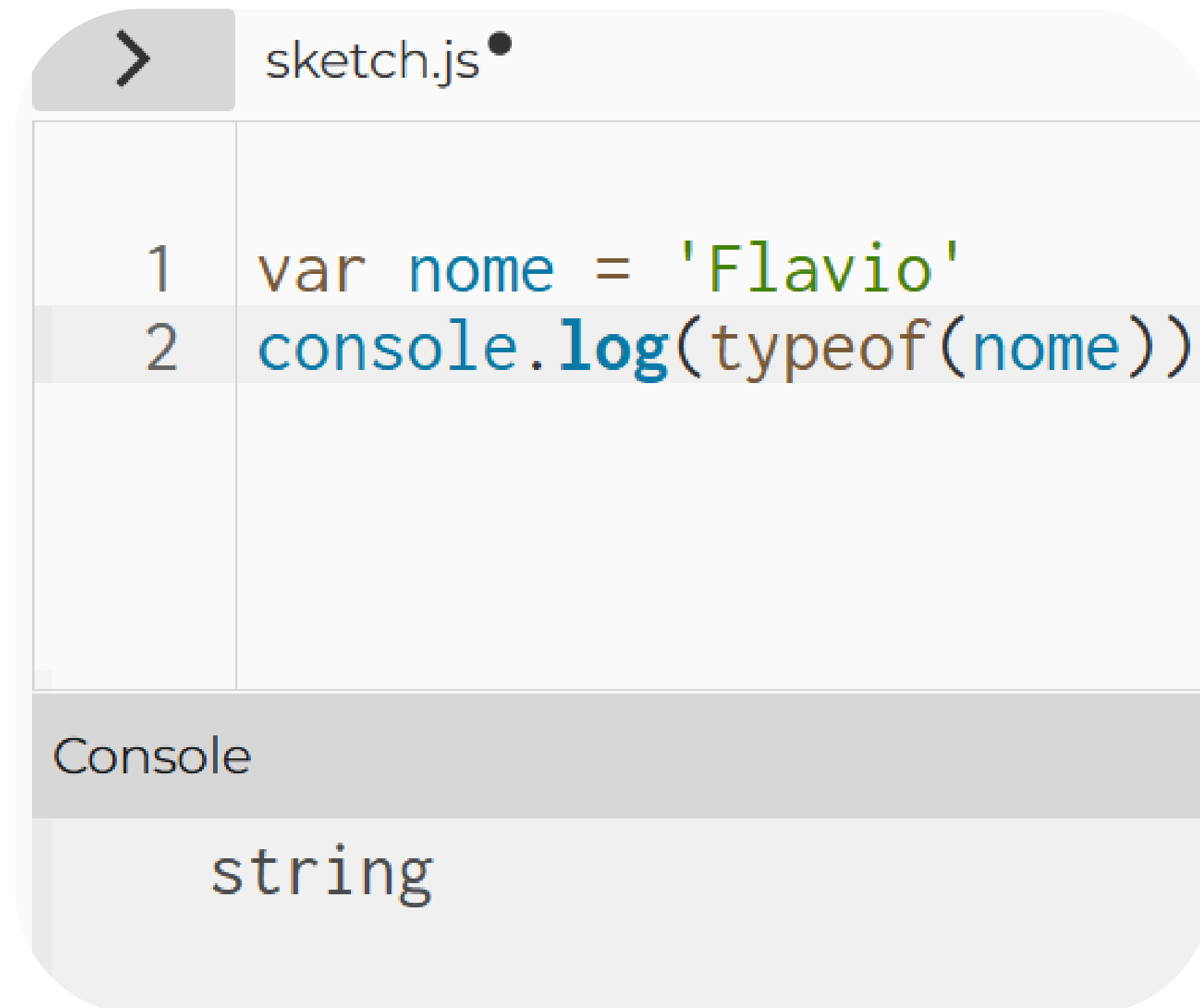
VAR IDADE = 28

BOOLEAN: FALSE OU TRUE

O COMANDO **typeof** RETORNA O TIPO DE VARIÁVEL



O COMANDO **typeof** RETORNA O TIPO DE VARIÁVEL



The image shows a code editor window titled 'sketch.js'. It contains two lines of JavaScript code: `var nome = 'Flavio'` on line 1 and `console.log(typeof(nome))` on line 2. Below the code editor is a console output area labeled 'Console' which displays the result `string`.

```
> sketch.js
```

1	<code>var nome = 'Flavio'</code>
2	<code>console.log(typeof(nome))</code>

Console

`string`

GUARDAR VARIÁVEIS :

usamos o comando : `window.prompt()`

ex:

```
var nome = window.prompt("Qual é o seu nome?");
```

para mostrar as variáveis no console :

ex:

```
console.log("Olá, " + nome + "! Seja bem-vindo!");
```

USO DO WINDOW.PROMPT E CONSOLE.LOG

>

sketch.js

1	var nome = window.prompt('Qual seu nome ? :')
2	console.log('Bem vindo '+nome)

Console

Clear

▼

Bem vindo flavio

PLACEHOLDERS

//placeholders são utilizados para concatenar strings com variáveis

//placeholders são representados por **`${}`**

//placeholders são utilizados com crases **```**

// Formatando Strings

```
var s = 'JavaScript'
```

```
'Eu estou aprendendo s' // não faz interpolação
```

```
'Eu estou aprendendo' + s // usa concatenação
```

```
`Eu estou aprendendo ${s}` // usa template string
```

ATRIBUIÇÃO DE VALOR ESPECÍFICO

NUMBER.PARSEINT
NUMBER.PARSEFLOAT
STRING

```
VAR N1 = NUMBER.PARSEINT(WINDOW.PROMPT("DIGITE UM  
NÚMERO:"));
```

```
VAR N2 = NUMBER(WINDOW.PROMPT("DIGITE OUTRO NÚMERO:"));  
VAR NOME = STRING(WINDOW.PROMPT("DIGITE UMA STRING:"));
```

//CONVERSÃO DE TIPOS EM RESUMO

Number.parseInt() - converte para inteiro

Number.parseFloat() - converte para decimal

Number() - converte para número

String() - converte para texto

Boolean() - converte para booleano

VAR LET E CONST

- Variáveis declaradas com **LET** têm escopo de bloco, instrução ou expressão e as
- Variáveis declaradas com **VAR** têm escopo global de função.
- Variáveis com **CONST** deixam de ser variáveis e passa a ser constantes
-

```
> if(true)
```



VAR LET E CONST

Operadores Aritméticos :

Ordem de
Precedencia -->

