

**!** No JavaScript moderno, além de utilizar a palavra **var**, também podemos usar a palavra **let**.

```

var n1
var n2
var n3
    
```

**!** Lembre-se: além do **var**, podemos utilizar o **let**.

```

var n1 = 5
var n2 = 8.5
var n3 = 15
    
```

```

var s1
var s2
var s3
    
```

```

n1 5
n2 8.5
n3 15
s1 JavaScript
s2 Curso em Vídeo
s3 Guanabara
    
```

```

var s1 = "JavaScript"
var s2 = 'Curso em Vídeo'
var s3 = `Guanabara`
    
```

```

n1 5
n2 8.5
n3 15
s1 JavaScript
s2 Curso em Vídeo
s3 Guanabara
    
```

O "s" significa "string"

Em JS pode-se usar dois tipos de aspas ("" e '') e a crase (``) para delimitar uma string

### Identificadores

- Podem começar com **letra**, **\$** ou **\_**
- Não podem começar com **números**
- É possível usar **letras** ou **números**
- É possível usar **acentos** e **símbolos**
- Não podem conter **espaços**
- Não podem ser **palavras reservadas**

```

Node.js
> var nome = 'Silvio'
undefined
> nome
'Silvio'
> 'nome'
'nome'
> nome
'Silvio'
> nome = 'Jaque'
'Jaque'
> nome
'Jaque'
> .exit
    
```

### Dicas

- Maiúsculas e minúsculas** fazem diferença
- Tente escolher **nomes coerentes** para as variáveis
- Evite se tornar um **'programador alfabeto'** ou um **'programador contador'**

```

> var a = 12
undefined
> var A = 10
undefined
> a
12
> A
10
>
    
```

Maiúsculo e minúscula fazem diferença no JS

## typeof

```

> typeof 10
'number'
> typeof 4.5
'number'
> typeof '5'
'string'
> typeof []
'object'
> typeof {}
'object'
> typeof function(){}
'function'
> typeof undefined
'undefined'
> typeof NaN
'number'
> typeof infinity
'undefined'
> typeof null
'object'
> []
    
```

### Data Types

- number
- Infinity
- NaN
- string
- boolean
- null
- undefined
- object
- Array
- function

```

> var n = 200
undefined
> n
200
> typeof n
'number'
> n = "Silvio"
'Silvio'
> typeof n
'string'
> []
    
```

No JS a variável pode mudar de tipo como no exemplo ao lado.