

OS TIPOS DE DADOS

“

Os **tipos de dados permitem** que o JavaScript **conheça** as **características** e **funcionalidades** que estarão disponíveis para esses **dados**.



NUMÉRICOS (number)

```
{}
```

```
var idade = 35; // número inteiro  
var preço = 150.65; // com decimais
```



Como o JavaScript está escrito em inglês, vamos usar um ponto para separar os decimais.

CADEIAS DE CARACTERES (string)

```
{}
```

```
var nome = 'Madero'; // aspas simples  
var ocupação = "Mestre dos hamburgues"; // aspas  
duplas tem o mesmo resultado
```

LÓGICOS OU BOOLEANOS (boolean)

```
{}
```

```
var luzLigada = true;  
var temFeijoadaNoDomingo = false;|
```

OBJETO (object)

Ao contrário de outros tipos de dados que podem conter apenas um único dado, os objetos são coleções de dados e todos os dados acima podem existir dentro deles.

Nós podemos reconhecê-los porque eles são declarados com chaves {}.

```
var pessoa = {  
  nome: 'João', // string  
  idade: 34, // number  
  solteiro: true // boolean  
}
```

OBJETO (object)

Para acessar um elemento dentro de um objeto, usamos notação de ponto. Em outras palavras, o nome do objeto e o nome da propriedade separados por um ponto.

```
var pessoa = {  
    nome = 'Teresa', // string  
    idade = 23 // number  
}
```

```
pessoa.nome // Teresa  
pessoa.idade // 23
```

{}

ARRAY (object)

Como objetos, arrays são coleções de dados. Nós podemos reconhecê-los porque eles são declarados com colchetes [].

Os arrays são um tipo especial de objeto, então não **os consideramos como mais um tipo de dado**.

Nós os mencionamos de uma maneira especial, porque eles são muito comuns em todos os tipos de código.

```
var comidasFavoritas = ['Feijoada', 'Pizza',  
  'Filé à parmegiana'];
```

```
let numerosSorteados = [12, 45, 56, 324, 452];
```

ARRAY (object)

Para acessar um elemento dentro de um array, usamos o operador de índice (que é escrito com colchetes []). O que significa o nome do array e o número de índice que queremos acessar.

```
let numerosSorteados = [12, 45, 56, 324, 452];
```

```
numerosSorteados[0] // 12
```

```
numerosSorteados[1] // 45
```

{ }

OS TIPOS DE DADOS ESPECIAIS

“

Os **tipos de dados especiais**

permitir que o JavaScript determine
estados especiais que os **dados**
podem ter.



NaN (NOT A NUMBER)

```
{}
```

```
var divisaoRuim = "35" / 2; // NaN não é um número
```

NULL (VALOR NULO)

Nós o atribuímos para indicar um valor vazio ou desconhecido.

```
{}
```

```
var temperatura = null; // Não chegou um dado, algo falhou
```

UNDEFINED (valor indefinido)

As variáveis têm um valor indefinido até lhes atribuirmos um valor.

```
{}
```

```
var outraVariavel; // undefined, não tem valor  
outraVariavel = "Hello"; //Agora sim temos um valor
```

Os comentários são partes do nosso código que **não são executadas**.

Eles começam sempre com duas barras inclinadas.
//

Nós os utilizamos para explicar o que estamos fazendo, **e deixar informações úteis** para a nossa equipe ou para o nosso eu futuro.



```
// Math.round() retorna o valor arredondado para o  
inteiro mais próximo.
```

```
var arredondado = Math.round(20.49);
```