

**UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE**  
**INSTITUTO METRÓPOLE DIGITAL**

# **AULA 13**

**INTRODUÇÃO AO JAVASCRIPT**

**PROF. JANIHERYSON FELIPE**

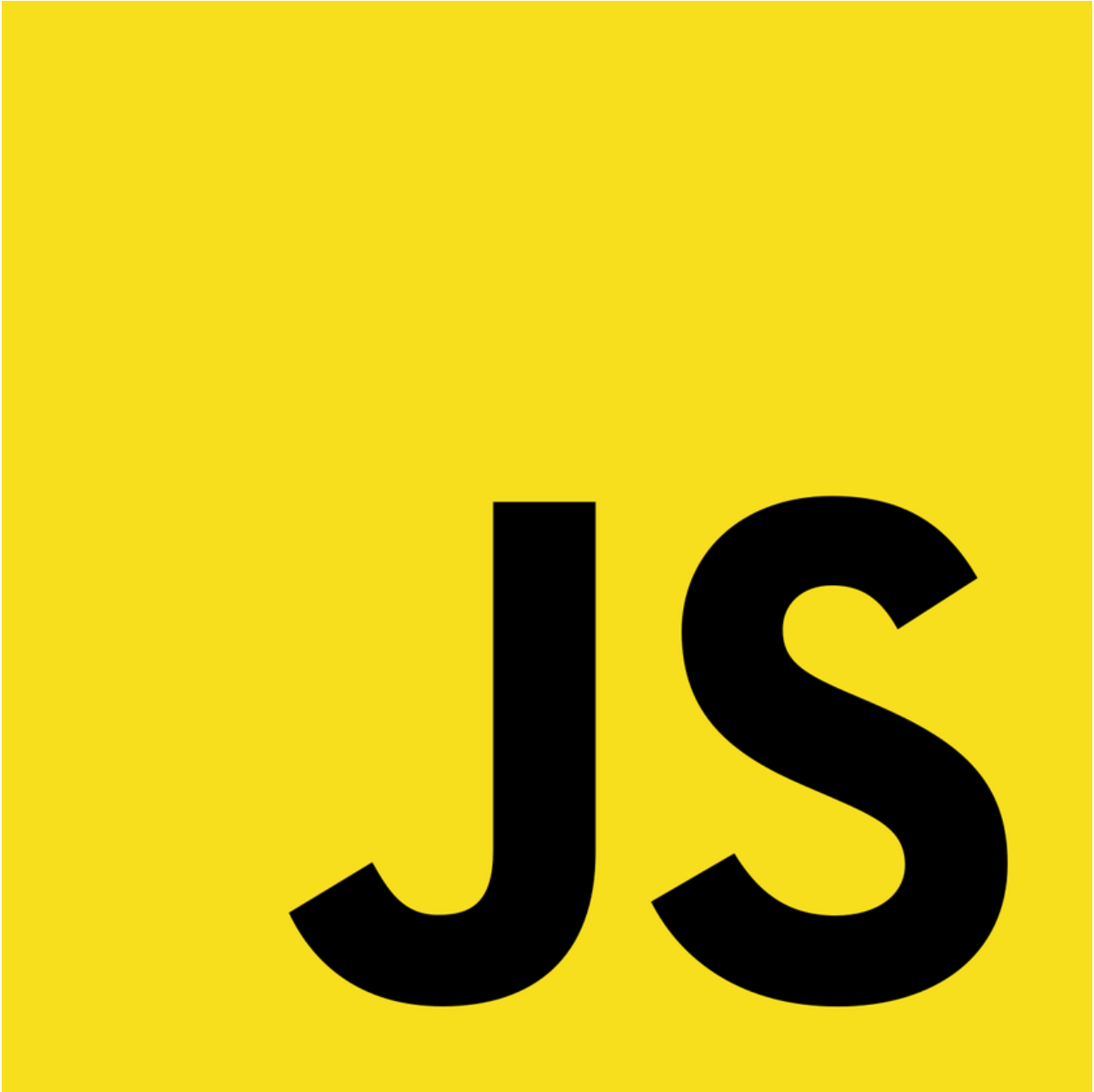
---

# CONTEÚDO DESSA AULA

- **CONHECER A HISTORIA DO JAVASCRIPT**
- **CONHECER AS CARACTERISTICA DA LINGUAGEM**
- **CONHECER AS FORMAS DE LEITURA E ESCRITA**
- **DISCUSSÕES E DÚVIDAS GERAIS.**

# JAVASCRIPT

- É uma linguagem de programação **interpretada estruturada**, de **script** em alto nível com tipagem dinâmica fraca e multiparadigma (protótipos, orientado a objeto, imperativo e funcional). Juntamente com HTML e CSS, o JavaScript é uma das três principais tecnologias da World Wide Web.

A large, bold, black 'JS' logo is centered on a bright yellow square background. The letters are thick and stylized, with the 'J' and 'S' connected at the bottom. The yellow square is positioned on the right side of the slide, partially overlapping a blue vertical bar on the far right edge.

# JAVASCRIPT

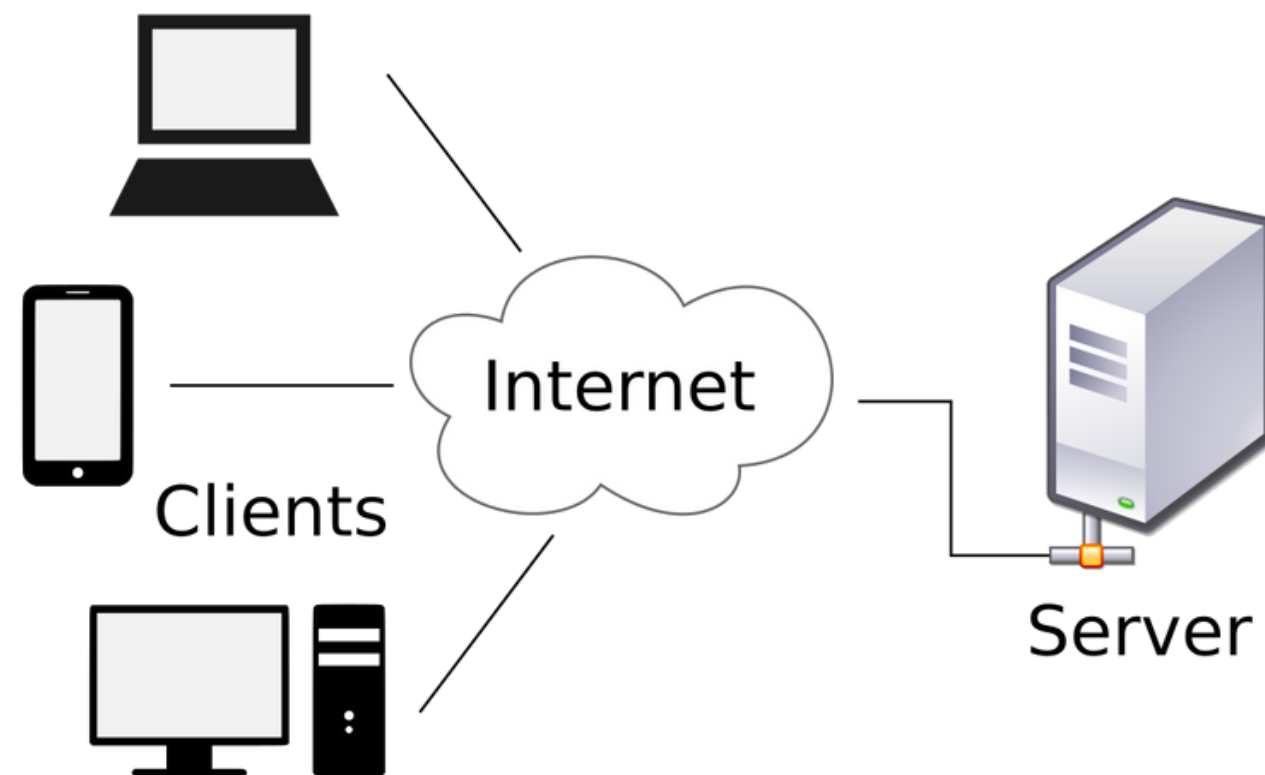
- A linguagem JavaScript é atualmente uma das mais conhecidas no mercado e na academia, sendo utilizada por boa parte dos programadores do mundo inteiro.
- O desenvolvimento do JavaScript teve início quando a **Netscape** percebeu, em **1995**, que as páginas web precisavam ser mais **dinâmicas**. Assim, observou-se que o HTML precisava de algum tipo de linguagem de programação que fosse **simples** para designers e programadores iniciantes.

# JAVASCRIPT

- Como a **Netscape** tinha uma colaboração com a **Sun Microsystems**, detentora dos direitos de Java, decidiu-se que esta linguagem deveria **complementar o Java** e ter uma **sintaxe semelhante**. **Brendan Eich** desenvolveu o primeiro protótipo da linguagem em apenas **dez dias** e o apresentou em **maio de 1995**.
- Ao longo dos anos, a linguagem foi então padronizada e teve sua **primeira versão**, cujo nome oficial era **ECMAScript 1, lançada em 1997**, e evoluiu até que, em 2018, foi lançada a ECMAScript 2018.

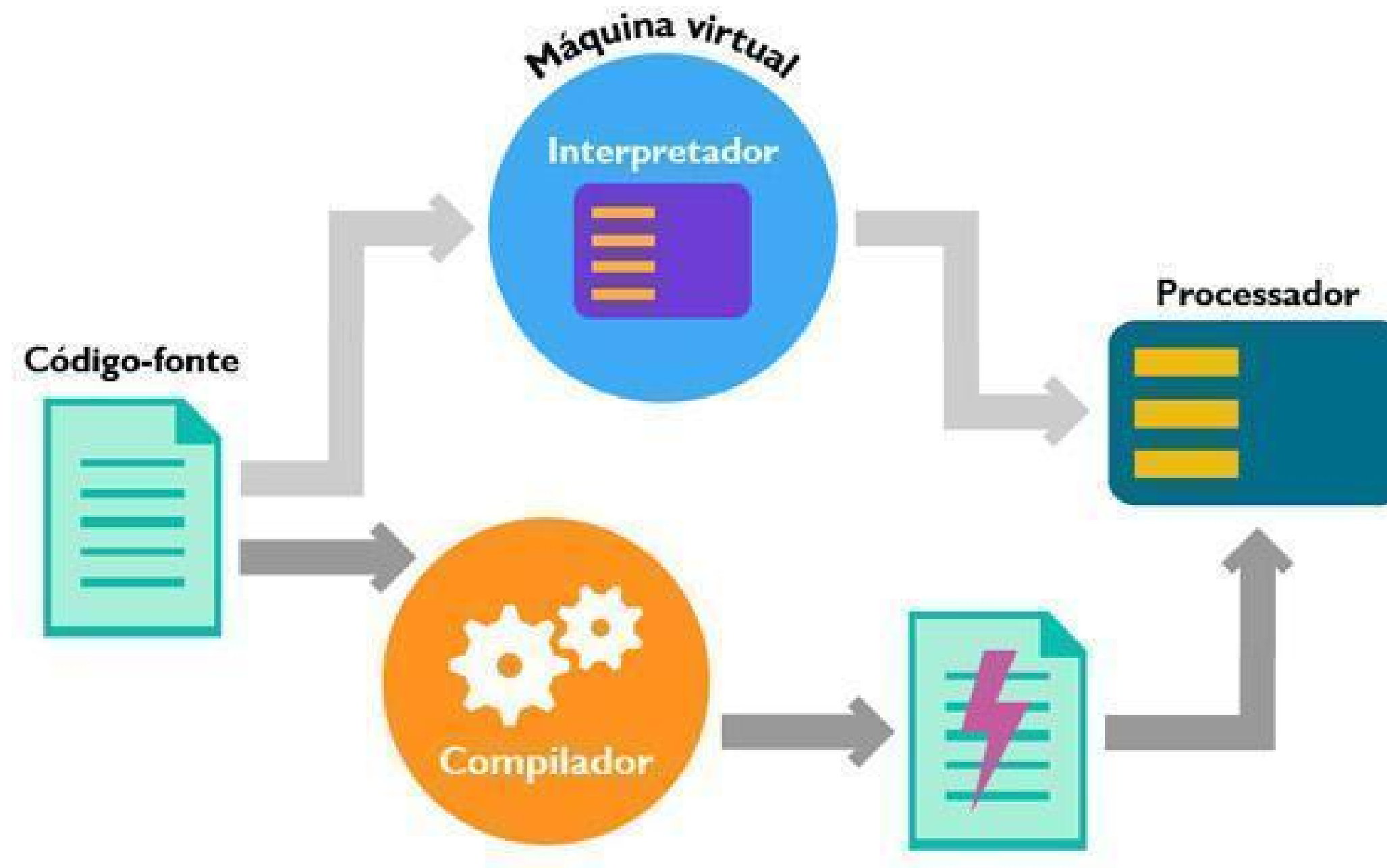
# JAVASCRIPT

Hoje, o JavaScript é a principal linguagem para programação **cliente** de sistemas web, isto é, JavaScript é a principal linguagem para programação da **interatividade das páginas** que são exibidas em nossos navegadores. No entanto, ela também é bastante utilizada do **lado do servidor**, através do **Node.js**.



# CARACTERÍSTICAS DA LINGUAGEM JAVASCRIPT

## Uma linguagem Interpretada



# CARACTERÍSTICAS DA LINGUAGEM JAVASCRIPT

## Uma linguagem fracamente tipada

- Essa é uma característica da maioria das linguagens de script, como JavaScript. Nessas linguagens, tipos são associados com valores e não com variáveis. Na prática, você não precisa definir no programa qual o tipo de uma variável. Por exemplo, uma variável `x` pode receber um valor inteiro em um determinado momento do programa e, mais tarde, receber uma string.



# CARACTERÍSTICAS DA LINGUAGEM JAVASCRIPT

## Uma linguagem multiparadigma

- Permite a utilização de diferentes paradigmas de programação em seus programas. Os paradigmas de programação determinam como o programador enxerga o seu programa



# POTENCIALIDADES DO JAVASCRIPT

O javascript é uma uma poderosa ferramenta na manipulação de paginas HTML/CSS.

- Manipulação de elementos HTML (exemplo 1);
- Alterar valores de atributos HTML (exemplo 2);
- Manipulação de Estilos CSS (exemplo 3);

# INCLUSÃO DE JAVASCRIPT NO HTML

O javascript pode ser incluído de quatro formas em um documento HTML.

1. Na seção do cabeçalho do HTML
2. Na seção do corpo do HTML
3. Em um arquivo JavaScript externo disponível localmente
4. Em um arquivo JavaScript disponível em uma URL

## ELEMENTOS BÁSICOS

- O ponto-e-virgula é facultativo, mas aconselha-se a usar.
- As instruções JavaScript geralmente começam com uma palavra-chave para identificar a ação JavaScript a ser executada, por exemplo, você já aprendeu a palavra-chave `function` para definir nossas funções.
- Comentários são feitos usando `//` (linha única) e `/**/` (Múltiplas linhas);

## ELEMENTOS BÁSICOS

```
<script>  
    //Comentario js  
    var x, y, z;  
    x = 10;  
    y = 20;  
    z = x + y;  
    var x,y,z; x = 10; y = 20; z = x + y;  
  
    document.getElementById("meu_texto_1").innerHTML = z;  
</script>
```

# ELEMENTOS BÁSICOS

```
<h2>Sequência de Instruções</h2>
<p id='meu_texto_1'></p>
<p id='meu_texto_2'></p>

<script>
    var nome = "Pedro";
    document.getElementById("meu_texto_1").innerHTML = nome;

    nome = 10;

    document.getElementById("meu_texto_2").innerHTML = nome;
</script>
```

# ENTRADA DE DADOS

- Os programas muitas vezes exigirão valores de entrada para que algum algoritmo seja executado gerando o seu resultado, ou seja, sua saída.

```
<script>
  function minhaOutraFuncao() {
    var x = document.getElementById("meuTexto").value;
    document.getElementById("meu_texto_6").innerHTML = x;
  }
</script>
```

# PRECEDÊNCIA DAS OPERAÇÕES

- A interpretação de uma operação em javascript é feita da esquerda para a direita. Sendo assim as operações abaixo produzem distintos resultados;

```
<script>
  var resultado = "Ola mundo" + 50 + 5;
  document.getElementById("p01").innerHTML = resultado;

  var resultado = 50 + 5 + "Ola mundo";
  document.getElementById("p01").innerHTML = resultado;

  var resultado = "Ola mundo" + (50 + 5);
  document.getElementById("p01").innerHTML = resultado;
</script>
```



# TIPOS DE DADOS

## Os tipos básicos são:

- [number] para números de qualquer tipo: inteiro ou ponto flutuante;
- [string] para strings, ou seja, para cadeias de um ou mais caracteres - em JavaScript não há um tipo de caractere único separado;
- [boolean] para os valores booleanos true e false;
- [null] para valores desconhecidos - este é um tipo independente que possui um único valor null; e,

# TIPOS DE DADOS

## Os tipos básicos são:

- [undefined] para valores não atribuídos - este também é um tipo independente que possui um único valor undefined

## Os principais tipos complexos são:

- [function] para funções;
- [symbol] para identificadores exclusivos; e,
- [object] para estruturas de dados mais complexas.

# OPERAÇÕES COM NUMEROS

```
<script>  
  var num1 = 10;  
  var num2 = 10.5;  
  
  var multiplicacao = num1 * num2;  
  var divisao = num1/num2;  
  var soma = num1 + num2;  
  var subtracao = num1 - num2;  
  var resto = num1 % num2;  
  
</script>
```

## TIPOS DE DADOS

- Além dos números regulares, existem os chamados valores numéricos especiais, que também pertencem a esses tipos de dados, são eles: **infinity** e **-infinity**, que representam os números fora do intervalo de números possíveis. E **NaN**, "Not a Number", que representa um número ilegal como, por exemplo, o resultado de dividir 100 pelo texto "programa".

# TIPOS DE DADOS

Valores do tipo string podem ser inseridos no programa usando aspas duplas ou simples, como nos textos "seja bem-vindo! " e 'até breve! '.

```
//Formas aceitas
var texto1 = "ola mundo";
var texto2 = 'ola mundo';
```

```
//Formas não aceitas
var texto3 = 'ola mundo';
var texto4 = "ola
                mundo";
```

# TIPOS DE DADOS

- boolean , é o tipo lógico de JavaScript. Ele tem apenas dois valores: true e false , o verdadeiro e o falso.
- Esses valores lógicos podem ser o resultado de uma comparação e são usados nas condições de comandos de seleção como o se-então, e de controle de loops,

```
var verdadeiro = true;  
var falso = true;  
var verdadeiro2 = 3 > 1;
```

## TIPOS DE DADOS

- Os últimos dois tipos básicos são o null e o undefined , que possuem apenas um valor cada, respectivamente, null e undefined. Em JavaScript, null não é uma "referência a um objeto não existente" ou um "ponteiro nulo", como em algumas outras linguagens. É apenas um valor especial que representa "nada", "vazio" ou "valor desconhecido"

```
var x = null;
```

# TIPOS DE DADOS

- Por outro lado, o significado do undefined é "valor não atribuído". Se uma variável é declarada, mas não atribuída, seu valor é indefinido.

```
var y;  
var z = undefined;
```

- Na verdade, undefined e null têm valores iguais, mas têm tipos diferentes.



# TIPOS DE DADOS

- Podemos ver o tipo de um dado através da propriedade `typeof` atribuindo a uma variável.

```
var x = null;  
  
var y;  
var z = undefined;  
  
var tipo1 = typeof z;
```

## TIPOS DE DADOS

- JavaScript tem um operador especial que é `===`. Esse comparador compara não apenas os valores, como também os tipos de ambas as expressões.

```
var x = null;  
var z = undefined;  
  
var w1 = x == z; //true  
var w2 = x === z; //false
```

## TIPOS DE DADOS

- Constantes em javascript são criadas com a palavra chave const. Não podem ser alteradas apos criadas, nem ciadas sem valor inicial.

```
//Constantes
const contante01; // Operação não permitida
const constante02 = 10;
constante02 = 20; // Operação não permitida
console.log(constante02);
```

# NOMES DE VARIÁVEIS ACEITOS

```
//Nomenclatura de variaveis  
var Total = 10; //É case sensitive  
var total = 20;  
  
var 1tes = 10; //Numeros no inicio não são aceitos  
  
var $dfgdf = "teste";  
var _teste = "teste";
```

# HOISTING

- Hoisting é o comportamento padrão de JavaScript que move as declarações das variáveis para o topo do programa
- É importante ressaltar que JavaScript leva apenas as declarações para o topo. As inicializações não são movidas.

```
document.getElementById("p01").innerHTML = x;  
var x;  
x = 100;  
  
var x;  
document.getElementById("p01").innerHTML = x;  
x = 100;
```

## VAR X LET

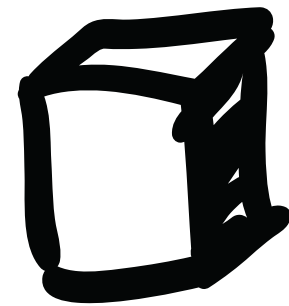
- As variáveis do tipo let tem seu escopo de bloco, sendo, portanto, esquecida ao concluir um bloco.
- Já as variáveis do tipo var tem seu escopo de função e podem ser acessadas de qualquer parte de uma função.

```
function exemploVarLet() {  
  if (true) {  
    var x = 10; // visível em toda a função  
    let y = 20; // visível apenas dentro deste bloco  
  }  
  console.log(x); // 10  
  console.log(y); // ReferenceError: y is not defined  
}
```

## MODO RIGOROSO "USE STRICT"

- O uso rigoroso impede que o interpretador JS deixe passar sentença erradas ou incompletas, como exemplo podemos citar quando inicializamos uma variável sem declara-la previamente.





DÚVIDAS?