
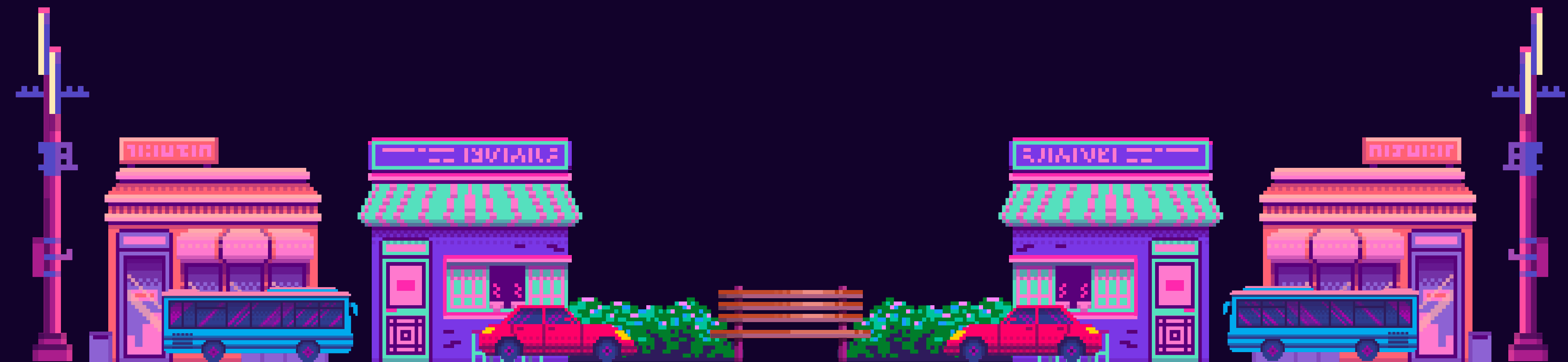




HELLO
WORLD



INTRODUÇÃO AO JAVASCRIPT



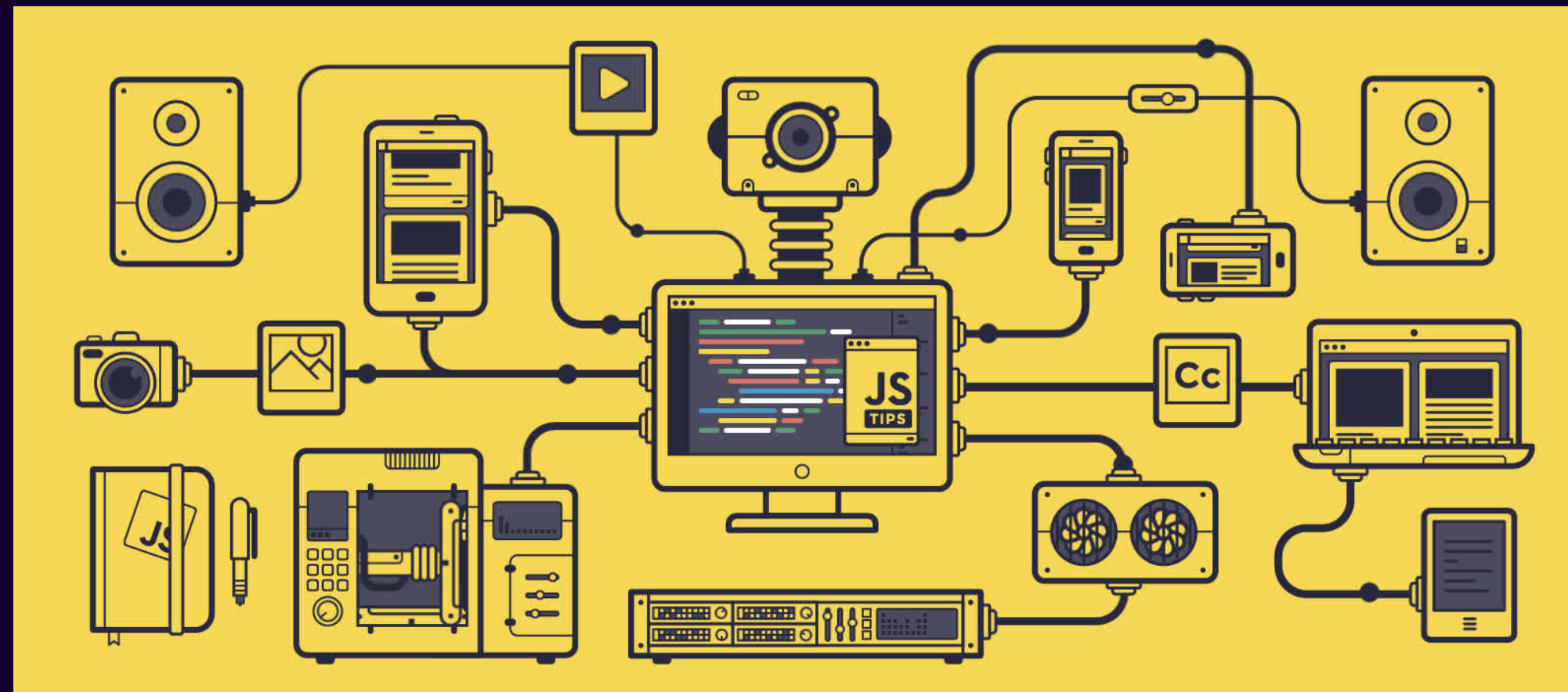
O QUE É O JAVASCRIPT?

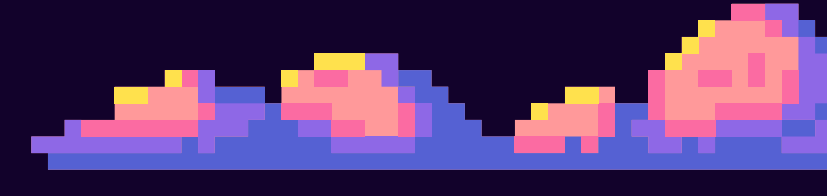
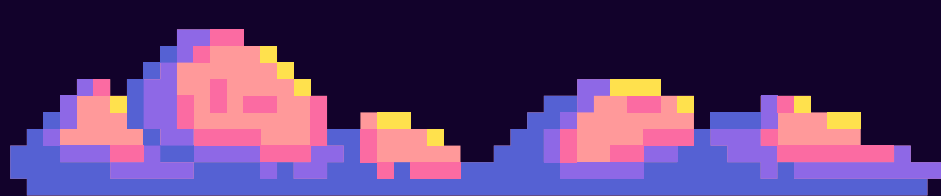
JavaScript é uma linguagem de programação que adiciona interatividade ao seu site. Isso acontece em jogos, no comportamento das respostas quando botões são pressionados ou com entrada de dados em formulários; com estilo dinâmico; com animação, etc. Este artigo ajuda você a começar a usar o JavaScript e aumenta sua compreensão do que é possível.

JavaScript 

JavaScript é uma poderosa linguagem de programação que pode adicionar interatividade a um site. Foi inventado por Brendan Eich. JavaScript é versátil e amigável para iniciantes. Com mais experiência, você poderá criar jogos, gráficos 2D e 3D animados, aplicativos abrangentes baseados em banco de dados e muito mais!

- Interfaces de programação de aplicativos de navegador (APIs) incorporadas a navegadores da Web, fornecendo funcionalidades como criação dinâmica de HTML e definição de estilos CSS; coletar e manipular um fluxo de vídeo da webcam de um usuário ou gerar gráficos 3D e amostras de áudio.
- APIs de terceiros que permitem aos desenvolvedores incorporar funcionalidades em sites de outros provedores de conteúdo, como Twitter ou Facebook.
- Estruturas e bibliotecas de terceiros que você pode aplicar ao HTML para acelerar o trabalho de construção de sites e aplicativos.





VARIÁVEIS

Variables são contêineres que armazenam valores. Você começa declarando uma variável com a palavra-chave `let`, seguida do nome que você dá à variável:

```
let myVariable;
```

Depois de declarar uma variável, você pode atribuir um valor a ela:

```
myVariable = "Bob";
```

Além disso, você pode fazer essas duas operações na mesma linha:

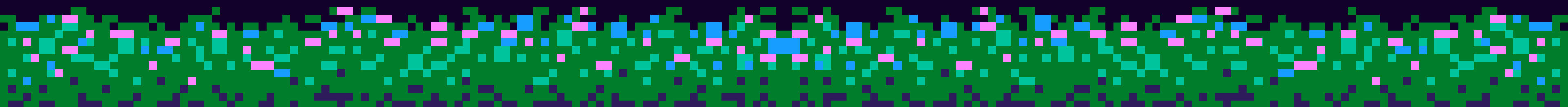
```
let myVariable = "Bob";
```

Você recupera o valor chamando o nome da variável:

```
myVariable;
```

Depois de atribuir um valor a uma variável, você pode alterá-lo posteriormente no código:

```
let myVariable = "Bob";  
myVariable = "Steve";
```





CONSTANTES

Nos primeiros dias do JavaScript, não existiam constants. No JavaScript moderno, temos a palavra-chave `const`, que nos permite armazenar valores que nunca podem ser alterados:

```
const diasNaSemana = 7;  
const horasNoDia = 24;
```

`const` funciona exatamente da mesma maneira que `let`, exceto que você não pode atribuir um novo valor a `const`. No exemplo a seguir, a segunda linha geraria um erro:

```
const diasNaSemana = 7;  
diasNaSemana = 8;
```

TIPOS DE DADOS

Tipagem dinâmica e fraca

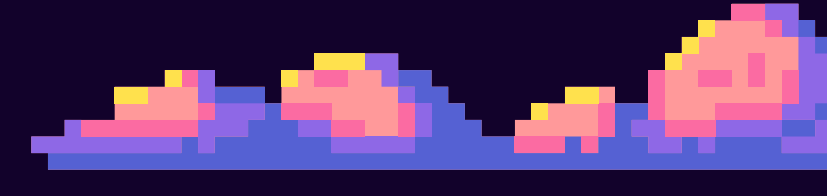
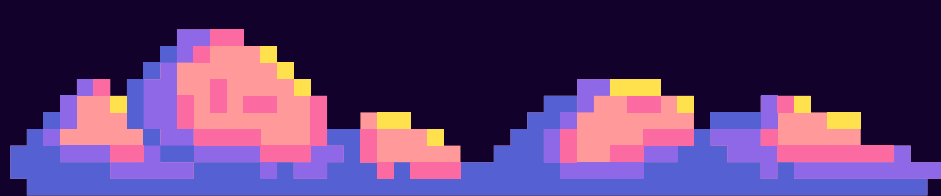
JavaScript é uma linguagem dinâmica com tipos dinâmicos. As variáveis em JavaScript não estão diretamente associadas a nenhum tipo de valor específico, e qualquer variável pode receber (e reatribuir) valores de todos os tipos:

Variável	Explicação	Exemplo
String	Esta é uma sequência de texto conhecida como string. Para significar que o valor for uma string, coloque-a entre aspas simples.	<pre>let myVariable = 'Bob';</pre>
Number	Isto é um número. Os números não têm aspas.	<pre>deixe minhaVariável = 10;</pre>
Boolean	Este é um valor Verdadeiro/Falso. As palavras <code>true</code> e <code>false</code> são palavras-chave especiais que não precisam de aspas.	<pre>let myVariable = true;</pre>
Array	Esta é uma estrutura que permite armazenar vários valores em um único referência.	<pre>let myVariable = [1, 'Bob', 'Steve', 10];</pre> <p>Refere-se a cada membro do array assim:</p> <pre>myVariable[0], myVariable[1], etc.</pre>
Object	Isso pode ser qualquer coisa. Tudo em JavaScript é um objeto e pode ser armazenados em uma variável. Tenha isso em mente enquanto aprende.	<pre>let myVariable = document.querySelector('h1');</pre> <p>Todos os exemplos acima também.</p>

OBJETOS

Um objeto é uma coleção de dados e/ou funcionalidades relacionadas (que geralmente consistem em diversas variáveis e funções — que são chamadas de propriedades e métodos quando estão dentro de objetos). Vamos trabalhar com um exemplo para entender como eles são.

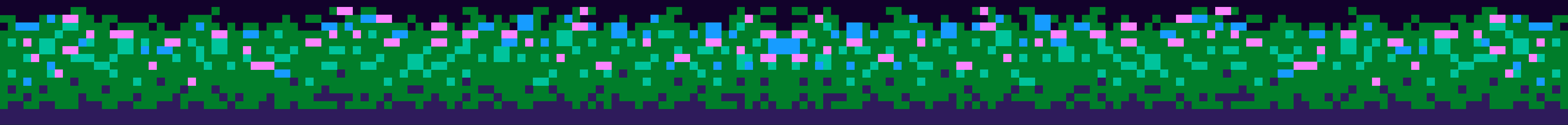
```
{
  nome: 'Ananias',
  idade: 20,
  endereco: {
    rua: "Vergueiro",
    cidade: "São Paulo"
  },
  telefones: ["91234-5678", '98765-4321'],
  "calcula-ano-nascimento": function(){
    //obtem ano atual
    var anoAtual = new Date().getFullYear();
    return anoAtual - this.idade;
  }
}
```

COMENTÁRIOS

```
/*  
Tudo no meio é um comentário.  
*/
```

```
// Isso é um comentário
```



OPERADORES

Um operador é um símbolo matemático que produz um resultado baseado em dois valores (ou variáveis). Na tabela a seguir, você pode ver alguns dos operadores mais simples, juntamente com alguns exemplos para experimentar no console JavaScript.

Operador	Explicação	Símbolo(s)	Exemplo	Não, não é igual	Isso retorna o valor logicamente oposto do que precede. Acontece um <code>true</code> em um <code>false</code> , etc. Quando é usado ao lado do operador de igualdade, o operador de negação testa se dois os valores <i>não</i> são iguais.	!, !==	<p>Para "Not", a expressão básica é <code>true</code>, mas o a comparação retorna <code>false</code> porque nós a negamos:</p> <pre>let myVariable = 3; !(myVariable === 3);</pre> <p>"Não é igual" dá basicamente o mesmo resultado com diferentes sintaxe. Aqui estamos testando "é <code>myVariable</code> NÃO igual a 3". Isso retorna <code>false</code> porque <code>myVariable</code> É igual a 3:</p> <pre>let myVariable = 3; myVariable !== 3;</pre>
Adição	Adicione dois números ou combine duas strings.	+	<code>6 + 9;</code> <code>'olá ' + 'mundo!';</code>				
Subtração, Multiplicação, Divisão	Eles fazem o que você espera que façam em matemática básica.	- , * , /	<code>9 - 3;</code> <code>8 * 2; // multiplicar em JS é um asterisco</code> <code>9 / 3;</code>				
Atribuição	Como você já viu: isso atribui um valor a uma variável.	=	<code>let myVariable = 'Bob';</code>				
Igualdade estrita	Isso executa um teste para ver se dois valores são iguais. Ele retorna um Resultado <code>true / false</code> (booleano).	===	<code>let myVariable = 3;</code> <code>myVariable === 4;</code>				

OPERADORES MATEMÁTICOS

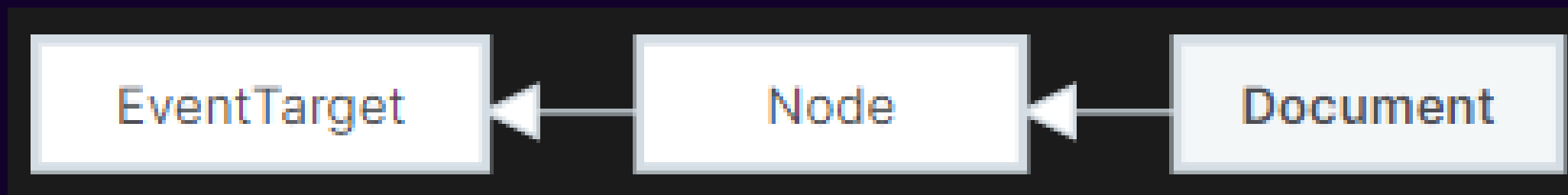
Um operador é um símbolo matemático que produz um resultado baseado em dois valores (ou variáveis). Na tabela a seguir, você pode ver alguns dos operadores mais simples, juntamente com alguns exemplos para experimentar no console JavaScript.

Operator	Name	Purpose	Example	Shortcut for
<code>+=</code>	Atribuição de adição	Adiciona o valor à direita ao valor da variável à esquerda e, em seguida, retorna o novo valor da variável	<code>x = 3; x += 4;</code>	<code>x = 3; x = x + 4;</code>
<code>-=</code>	Atribuição de subtração	Subtrai o valor à direita do valor da variável à esquerda e retorna o novo valor da variável	<code>x = 6; x -= 3;</code>	<code>x = 6; x = x - 3;</code>
<code>*=</code>	Atribuição de multiplicação	Multiplica o valor da variável à esquerda pelo valor à direita e retorna o novo valor da variável	<code>x = 2; x *= 3;</code>	<code>x = 2; x = x * 3;</code>
<code>/=</code>	Atribuição de divisão	Divide o valor da variável à esquerda pelo valor à direita e retorna o novo valor da variável	<code>x = 10; x /= 5;</code>	<code>x = 10; x = x / 5;</code>

Operator	Name	Purpose	Example
<code>===</code>	Igualdade estrita	Verifica se o valor da esquerda e o da direita são idênticos entre si.	<code>5 === 2 + 4</code>
<code>!==</code>	Não-igualdade-estrita	Verifica se o valor da esquerda e o da direita não são idênticos entre si.	<code>5 !== 2 + 3</code>
<code><</code>	Menor que	Verifica se o valor da esquerda é menor que o valor da direita.	<code>10 < 6</code>
<code>></code>	Maior que	Verifica se o valor da esquerda é maior que o valor da direita.	<code>10 > 20</code>
<code><=</code>	Menor ou igual que	Verifica se o valor da esquerda é menor que ou igual ao valor da direita.	<code>3 <= 2</code>
<code>>=</code>	Maior ou igual que	Verifica se o valor da esquerda é maior que ou igual ao valor da direita.	<code>5 >= 4</code>

DOCUMENT

Para cada página carregada no browser, existe um objeto Document. A interface Document serve como um ponto de entrada para o conteúdo da Página (a árvore DOM, incluindo elementos como <body> e <table>) e provê funcionalidades globais ao documento (como obter a URL da página e criar novos elementos no documento).



Um objeto document pode ser obtido por meio de várias APIs:

- A mais comum; você trabalha com o script do documento que está rodando o script com o objeto document. (O mesmo document também pode ser referenciado como window.document.)
- O documento de um iframe através da propriedade contentDocument.
- O responseXML de um objeto XMLHttpRequest.
- O documento ao qual um determinado node ou elemento pertence pode ser recuperado usando a propriedade do node ownerDocument.

FUNÇÕES

Functions são uma forma de empacotar a funcionalidade que você deseja reutilizar. É possível definir um corpo de código como uma função que é executada quando você chama o nome da função em seu código. Esta é uma boa alternativa para escrever repetidamente o mesmo código. Você já viu alguns usos de funções. Por exemplo:

```
let myVariable = document.querySelector("h1");
```


```
alert("olá!");
```

Biblioteca do Javascript, Math:

```
Math.sqrt(4)
```

FUNÇÕES PERSONALIZADAS

Você também pode definir suas próprias funções. No próximo exemplo, criamos uma função simples que recebe dois números como argumentos e os multiplica:



```
1 function multiplique(num1, num2) {  
2     let result = num1 * num2;  
3     return result;  
4 }  
5  
6 multiplique(2, 3);  
7
```

Os parâmetros (variáveis) colocadas no escopo de uma função, só são acessados dentro da função, não sendo conhecidos fora de seu escopo, a não ser que retornados

PROPRIEDADE

- `document.getElementById(id):`
 - Retorna um elemento com o ID especificado.
- `document.querySelector(selector):`
 - Retorna o primeiro elemento que corresponde ao seletor CSS especificado.
- `document.querySelectorAll(selector):`
 - Retorna uma lista de todos os elementos que correspondem ao seletor CSS especificado.
- `document.createElement(tagName):`
 - Cria um novo elemento com o nome de tag especificado.
- `document.body:`
 - Retorna o elemento `<body>` do documento.
- `document.head:`
 - Retorna o elemento `<head>` do documento.
- `document.title:`
 - Obtém ou define o título do documento.
- `document.URL:`
 - Retorna o URL completo da página atual



GAME RULES

01

Cria um componente HTML, e buscá-lo com o javascript

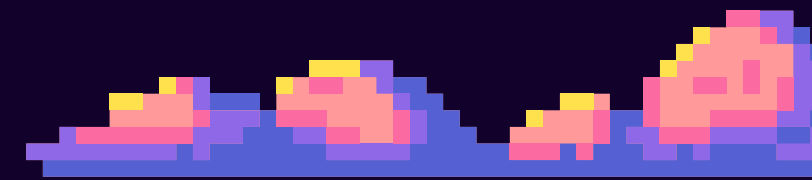
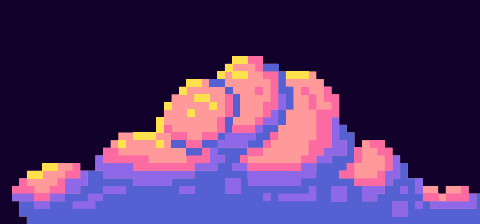
02

Criar uma função que vai sortear um número de 1 à 10.

03

O resultado da execução deve ser impresso no navegador.





DICAS

01

Usar biblioteca Math do Javascript

02

Você pode usar o getElementById

03

Usar o <script> dentro do HTML

O número sorteado aparecerá aqui

Sortear

O número sorteado é: 5

Sortear

MEUS CONTATOS:



27 99500-7495



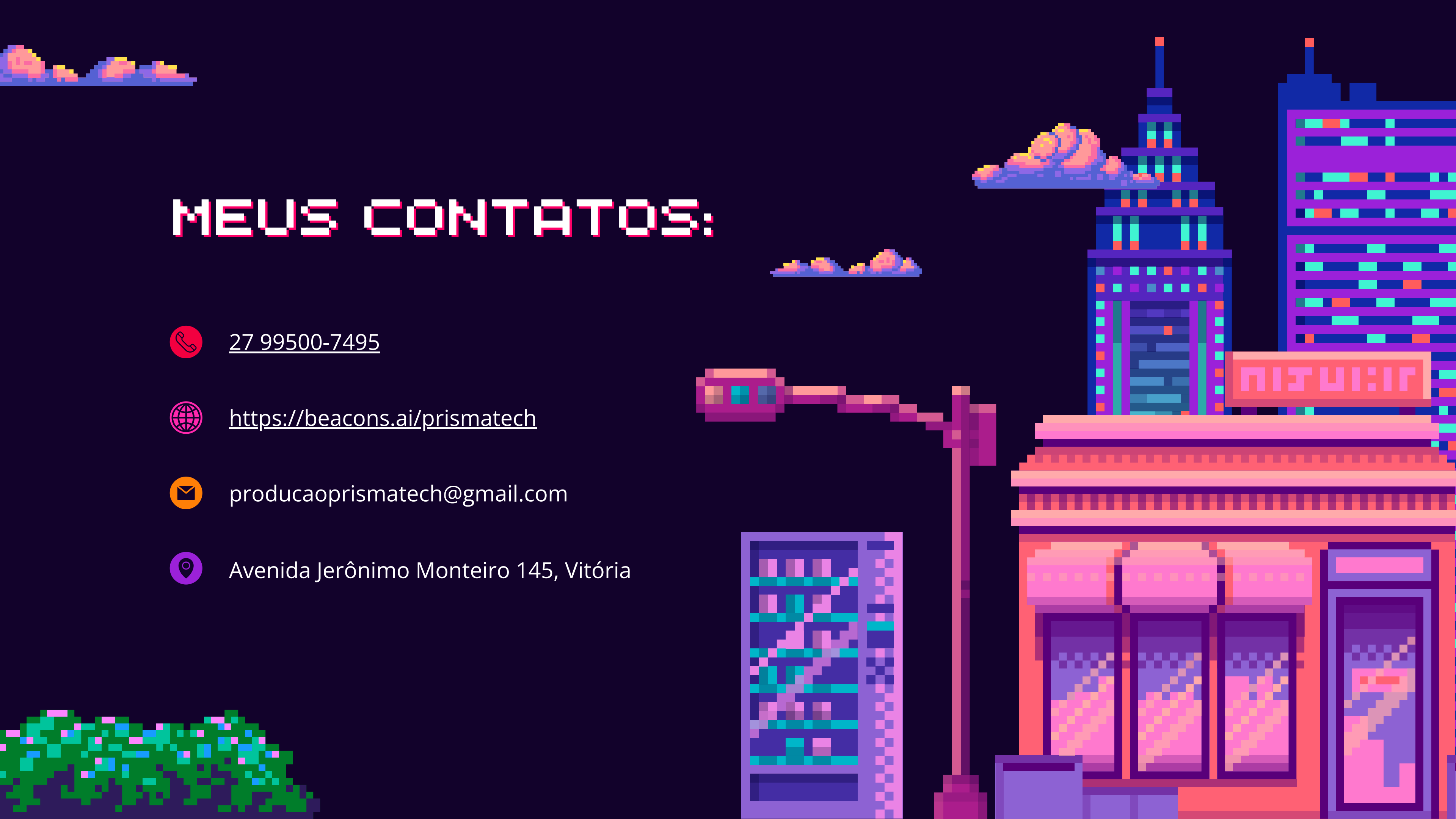
<https://beacons.ai/prismatech>



producaoprismatech@gmail.com



Avenida Jerônimo Monteiro 145, Vitória





THANK
YOU