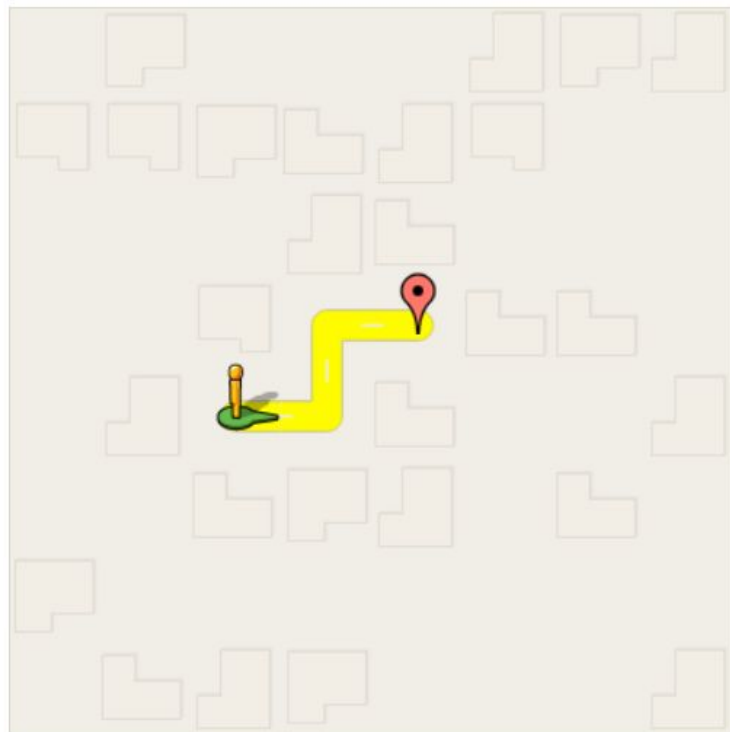


Curso de Javascript

Dayana Karoline

Lógica de programação



Jogos Blockly : Labirinto

Mover para a frente

gire para a esquerda ↶ ▾

gire para a direita ↷ ▾

Lógica de programação



▶ Executar o programa

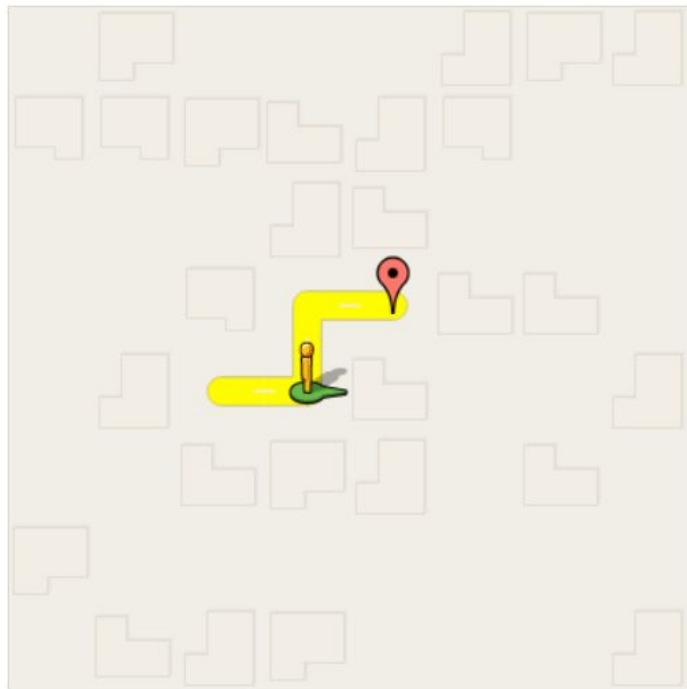
Mover para a frente

gire para a esquerda ↶

gire para a direita ↷

Mover para a frente

Lógica de programação



✖ Reiniciar

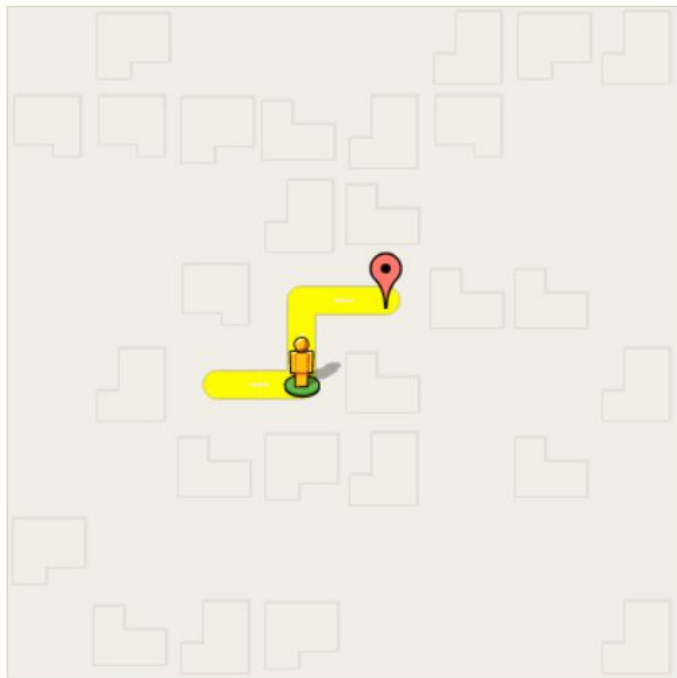
Mover para a frente

gire para a esquerda ↶

gire para a direita ↷

Mover para a frente

Lógica de programação



✖ Reiniciar

Mover para a frente

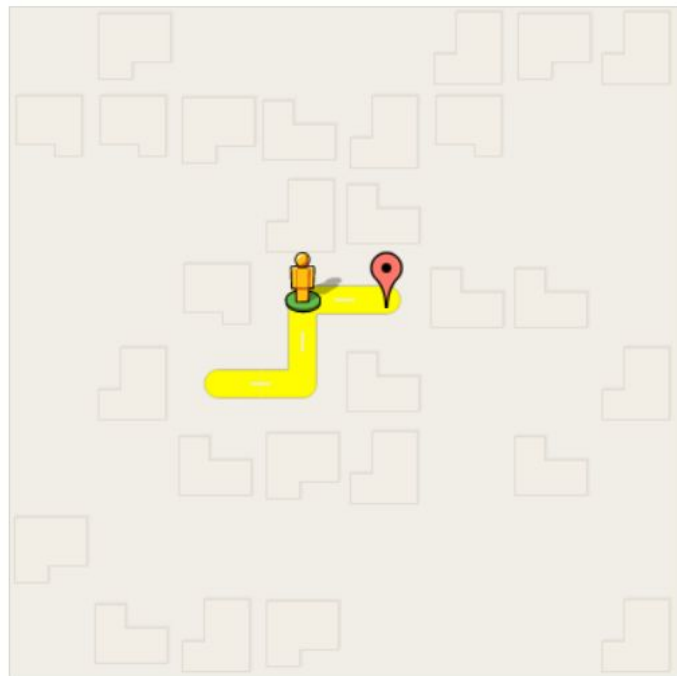
gire para a esquerda ↶

gire para a direita ↷

Mover para a frente

gire para a esquerda ↶

Lógica de programação



✖ Reiniciar

Mover para a frente

gire para a esquerda ↶

gire para a direita ↷

Mover para a frente

gire para a esquerda ↶

Mover para a frente

Lógica de programação



✖ Reiniciar

Mover para a frente

gire para a esquerda ↶

gire para a direita ↷

Mover para a frente

gire para a esquerda ↶

Mover para a frente

gire para a direita ↷

Lógica de programação



✖ Reiniciar

Mover para a frente

gire para a esquerda ↶

gire para a direita ↷

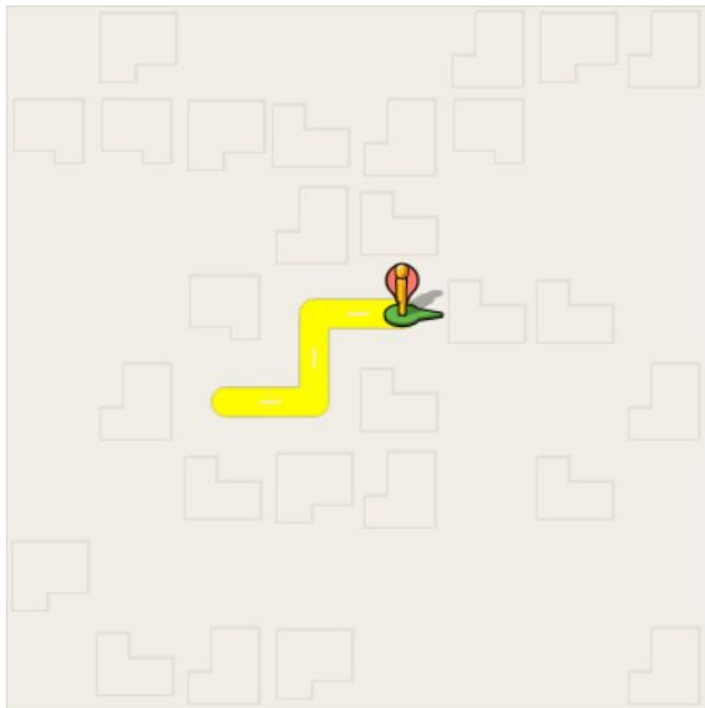
Mover para a frente

gire para a esquerda ↶

Mover para a frente

gire para a direita ↷

Lógica de programação



✖ Reiniciar

Mover para a frente

gire para a esquerda ↶

gire para a direita ↷

Mover para a frente

gire para a esquerda ↶

Mover para a frente

gire para a direita ↷

Mover para a frente

Linguagem de programação

São métodos e regras desenvolvidas para comunicação entre o homem e a máquina.

Escritos na Língua Inglesa de forma que tanto o homem quanto a máquina conseguem interpretar.

O objetivo é passar instruções para a máquina executar.

Por que aprender Javascript ?

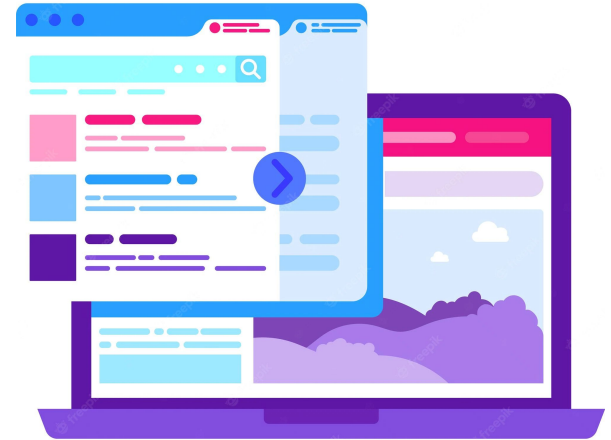
Javascript é uma das 3 linguagens que todos os desenvolvedores devem aprender.

- HTML : Definição do conteúdo da página web;
- CSS : Especificação do layout da página;
- JavaScript : Programação do comportamento da página.



HTML

- HTML é a linguagem de marcação padrão para páginas da Web.
- Com HTML você pode criar seu próprio site.



Primeira Página Web

```
<!DOCTYPE html>
<html lang="pt">
<head>
  <title>Minha Página</title>
</head>
<body>
  <h1>Olá! Esta é minha primeira página web </h1>
  <p>E este é o meu primeiro parágrafo! </p>
</body>
</html>
```

CSS

- CSS é a linguagem que usamos para dar estilo e descrever como os elementos HTML devem ser exibidos.



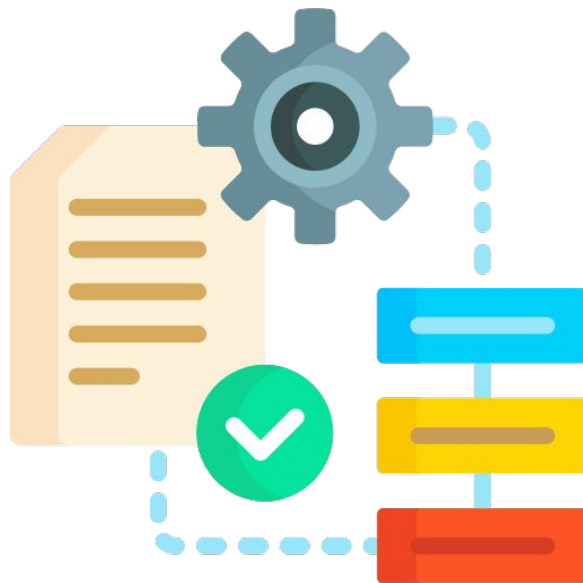
Criando meu primeiro CSS

```
<!DOCTYPE html>
<html lang="pt">
<head>
  <title>Minha página</title>
  <link rel="stylesheet"
href="\home\lab25dp\Documents\style.css" >
</head>
<body>
  <h1>Olá! Esta é minha primeira página
web</h1>
  <p>E este é o meu primeiro parágrafo! </p>
</body>
</html>
```

```
body {
  background-color: lightblue;
}
h1 {
  color: white;
  text-align: center;
}
p {
  text-align: center;
  font-family: verdana;
  font-size: 20px;
}
```

JavaScript

- Uma das linguagens de programação mais populares do mundo.
- Linguagem de programação da Web.
- Fácil de aprender.



Meu primeiro Javascript

```
<!DOCTYPE html>
<html lang="pt">
<head>
  <title>Minha página</title>
  <link rel="stylesheet"
href="\home\lab25dp\Documents\style.css" >
</head>
<body>
  <h1>Olá! Esta é minha primeira página web </h1>
  <p>E este é o meu primeiro parágrafo! </p>
  <script
src="\home\lab25dp\Documents\script.js" ></script>
</body>
</html>
```

```
alert("Olá mundo!");
```

Saída de Dados (Output)

- `getElementById()` - Altera conteúdo de um elemento HTML;
- `innerHTML` - Escreve em um elemento HTML;
- `document.write()` - Escreve a saída como elemento HTML;
- `alert()` - Escreve em uma caixa de alerta;
- `console.log()` - Escreve no console do navegador.

Alterando elemento HTML

```
<!DOCTYPE html>
<html lang="pt">
<head>
  <meta charset="utf-8"/>
  <title>Minha página</title>
  <link rel="stylesheet" href="\home\lab25dp\Documents\style.css" >
</head>
<body>
  <p id = "p1"> </p>
  <script src="\home\lab25dp\Documents\script.js" ></script>
</body>
</html>
```

Alterando elemento HTML

```
document.getElementById("p1").innerHTML = "Escrevendo no parágrafo";
```

Escrevendo no HTML

```
<!DOCTYPE html>
<html lang="pt">
<head>
  <meta charset="utf-8"/>
  <title>Minha página</title>
  <link rel="stylesheet" href="\home\lab25dp\Documents\style.css" >
</head>
<body>
  <h1>Olá! Esta é minha primeira página web </h1>
  <p>E este é o meu primeiro parágrafo! </p>
  <script src="\home\lab25dp\Documents\script.js" ></script>
</body>
</html>
```

Escrevendo no HTML

```
document.write("Olá mundo!");
```

Escrevendo no console do navegador

```
console.log("Olá mundo!")
```

Variáveis

São espaços de memória que servem para guardar (armazenar) dados.

Existem 4 maneiras de declarar uma variável em JavaScript:

- `var`
- `let`
- `const`
- `nada`

Exemplo 1

```
const y = 5;  
const x = 6;  
let total = x + y;  
alert(total);
```

Saídas:

11

Exemplo 2

```
const y = 5;  
const x = 6;  
var total = x * y;  
alert(total);
```

Saídas:

30

Operador de Atribuição

Em JavaScript, o sinal de igual (=) é um operador de "atribuição", não um operador "igual a";

Exemplo 3

```
var x = 1;  
x = x + 5;  
alert(x);
```

Saídas:
6

Tipos de Dados

As variáveis JavaScript podem conter números como 100 , valores de texto como "João" e valores booleanos como: verdadeiro ou falso.

```
const pi = 3.14;  
let nome = "Alice Braga";  
let CPF = "018.450.165-01";
```

Operadores Aritméticos

Operador	Descrição
+	Adição
-	Subtração
*	Multiplicação
**	Potência
/	Divisão
%	Resto da divisão
++	Incremento
--	Decremento

Operadores de Atribuição

Operador	Exemplo	Significado
=	$x = y$	$x = y$
+=	$x += y$	$x = x + y$
-=	$x -= y$	$x = x - y$
*=	$x *= y$	$x = x * y$
/=	$x /= y$	$x = x / y$
%=	$x \% = y$	$x = x \% y$
**=	$x ** = y$	$x = x ** y$

Exemplo 4

```
let x = 10;  
x += 5;  
alert(x);
```

Saídas:

15

Exemplo 5

```
const y = 5;  
const x = 6;  
x *= y;  
alert(x);
```

Saídas:

30

Concatenando String

O operador (+) também pode ser usado para adicionar (concatenar) strings.

Exemplo 6

```
let texto1 = "João";  
let texto2 = "Lucas";  
let texto3 = texto1 + " " + texto2;
```

Saídas:

João Lucas

Exercício 1

Desenvolva um código em javascript que armazena em **x** e **y** os valores **10** e **5** respectivamente. Seu programa deve mostrar uma **caixa de texto** com o resultado da **soma, subtração, divisão e potência** entre **x** e **y**.

Saídas:

$$10 + 5 = 15$$

$$10 - 5 = 5$$

$$10 / 5 = 2$$

$$10 ^ 5 = 100000$$

Exercicio 1 - Resolvido

index.js x



```
1 alert("10 + 5 = "+(10+5));  
2 alert(`10 - 5 = %d`,(10-5));  
3 alert(`10 / 5 = %d`, (10/5));  
4 alert("10 ^ 5 = "+(10**5));  
5
```

Console x

Shell x



10 + 5 = 15

10 - 5 = 5

10 / 5 = 2

10 ^ 5 = 100000

Hint: hit control+c anytime to enter REPL.

>

Entrada de dados

- `prompt()` - é um método que exibe uma caixa de texto que solicita a entrada do usuário e retorna o valor de entrada se o usuário clicar em "OK", caso contrário, retorna `null`.

Exemplo 7

```
let nome = prompt("Por favor informe seu nome:");

if (nome != null) {
    document.getElementById("nome").innerHTML =
        "Olá " + nome + "! Como você está?";
}
```

Exercício 2

Desenvolva uma página web igual a página abaixo, onde os números 5 e 10 são informados com entrada de dados.

Olá, Esta é minha pagina web!

A soma de 5 + 10 é :

15

Operadores de Comparação

Operador	Descrição
==	valores iguais independente do tipo
===	valores iguais e tipos iguais
!=	diferente
!==	valor ou tipo diferente
>	Maior que
<	Menor que
>=	Maior igual
<=	Menor igual
?	Operador Ternário

Exemplo

```
let x = 10;  
let y = 20;  
var compara = x<=y ;  
  
document.write(compara ? "Sim!": "Não!");
```

Saída:

Sim!

Estruturas de Decisão

- **if** para especificar um bloco de código a ser executado, se uma condição especificada for verdadeira;
- **else** para especificar um bloco de código a ser executado, se a condição do **if** for falsa;
- **else if** para especificar uma nova condição a ser testada, se a primeira condição for falsa;
- **switch** para especificar muitos blocos alternativos de código a serem executados.

if

```
var idade = 21;  
  
if (idade <= 18) {  
    document.write("Pode Dirigir");  
}
```

if else

```
var idade = 21;

if (idade <= 18) {
    document.write("Pode Dirigir");
}
else{
    document.write("Não pode Dirigir");
}
```


Operadores Lógicos

Operador	Descrição
&&	e (and)
	ou (or)
!	negação (not)

else if

```
var idade = 21;

if (idade < 16) {
    document.write("Não pode votar nem dirigir!");
}
else if(idade >= 16 && idade <18){
    document.write("Pode votar, mas não pode dirigir!");
}else{
    document.write("Pode votar e Dirigir!");
}
```

Switch Case

```
var dia = 5;
switch (dia) {
  case 0:
    document.write("Domingo");
    break;
  case 1:
    document.write("Segunda");
    break;
  case 2:
    document.write("Terça");
    break;
```

```
  case 3:
    document.write("Quarta");
    break;
  case 4:
    document.write("Quinta");
    break;
  case 5:
    document.write("Sexta");
    break;
  case 6:
    document.write("Sábado");
  }
}
```

Exercício 3

Faça uma calculadora de média aritmética. Sua calculadora deve ter a opção do usuário digitar 3 notas e calcular a média, mostrando se o aluno está aprovado, em recuperação ou reprovado, seguindo as especificações abaixo.

Se a média é maior ou igual a 5:

⇒ Aprovado!

Se a média é maior ou igual a 3 e menor que 5:

⇒ Recuperação!

Se a média é menor que 3:

⇒ Reprovado!

P5.js Editor – Projeto Calculadora

Calculadora Média Aritmética

Nota 1

5.8

Nota 2

4.2

Nota 3

8.7

Calcular

Aprovado!
Média: 5.67

Vetores

Um vetor é uma variável especial, que pode conter mais de um valor.

```
let carro1 = "Volkswagen";  
let carro2 = "Hyundai";  
let carro3 = "Fiat";
```

```
const carros = ["Volkswagen", "Hyundai", "Fiat"];
```

Armazenando valor em uma posição do vetor

Declarando vetor e guardando o valor na primeira posição.

```
const carros = [];  
carros[0] = "Volkswagen";
```

Mostrando elemento do vetor

É possível mostrar um elemento específico do vetor, acessando o endereço onde esse elemento foi armazenado.

0

1

2

3

Volkswagen	Hyundai	Fiat	Renault
------------	---------	------	---------

Mostrando elemento do vetor

É possível mostrar um elemento específico do vetor, acessando o endereço onde esse elemento foi armazenado.

0	1	2	3
Volkswagen	Hyundai	Fiat	Renault
carros[0]	carros[1]	carros[2]	carros[3]

Mostrando elemento do vetor

Para mostrar apenas um elementos do vetor, podemos usar o comando `document.write()`

```
document.write(carros[0]);
```

Saída:

Volkswagen

Mostrando elementos do vetor

Para mostrar todos os elementos de um vetor, podemos usar o comando `console.log()`

```
console.log(carros);
```

Saída:

```
(3) ["Volkswagen", "Hyundai", "Fiat"]
```



Mostrando elementos do vetor

Para mostrar todos os elementos de um vetor, podemos usar o comando `document.write()`

```
document.write(carros);
```

Saída:

Volkswagen,Hyundai,Fiat

Mostrando elementos do vetor

Para mostrar todos os elementos de um vetor, podemos usar o comando `alert()`

```
alert(carros);
```

Saída:

Volkswagen,Hyundai,Fiat

OK

Repetição Contada for()

As repetições são úteis, se você quiser executar o mesmo código várias vezes, cada vez com um valor diferente.

```
var text = "";  
for (let i = 0; i < 5; i++) {  
  text += "O número é " + i + "\n";  
}  
console.log(text);
```

Saídas:

O número é 0
O número é 1
O número é 2
O número é 3
O número é 4

Exemplo

Mostrar todos os elementos de um vetor separando com espaço.

```
const carros = ["Volkswagen", "Hyundai", "Fiat" , "Renault"];

for(i=0; i<4; i++){
  document.write(carros[i]+" ");
}
```

Saída:

Volkswagen Hyundai Fiat Renault

Exemplo

Mostrar todos os elementos de um vetor pulando a linha:

```
const carros = ["Volkswagen", "Hyundai", "Fiat" , "Renault"];

for(i=0; i<4; i++){
    document.write(carros[i]+"<br/>");
}
```

Saída:

Volkswagen
Hyundai
Fiat
Renault

Exercício 4

Faça um programa usando o vetor frutas abaixo, que mostrará todos os elementos do vetor separados por traço (-) usando repetição contada. Após exibir o vetor, receba uma fruta na entrada, usando o comando `prompt()`, verifique usando repetição contada, se essa fruta pertence ao vetor recebido, mostrando a posição do vetor seguido da mensagem “ - **Encontrei!**” caso encontre a fruta e “ - **Não Encontrei!**” Caso contrário, sua saída deve ser exibida usando o comando `document.write()` pulando uma linha na página web.

```
1  var frutas = ["Abacaxi", "Uva", "Limão","Goiaba","Banana", "Laranja"];  
2
```

Exercício 4 - Saída do programa

Exercicio 3 -

Abacaxi - Uva - Limão - Goiaba - Banana - Laranja -

Buscando por Uva...

0 - Não encontrei!

1 - Encontrei!

2 - Não encontrei!

3 - Não encontrei!

4 - Não encontrei!

5 - Não encontrei!

Resolução do Exercício 4

```
script.js x +
JS > ...
1 var frutas = ["Abacaxi", "Uva", "Limão", "Goiaba", "Banana", "Laranja"];
2
3 document.write("</br>");
4
5 ▼ for(j=0;j<6;j++){
6   document.write(frutas[j]+" - ");
7 }
8
9 var fruta = prompt("Digite uma fruta:");
10
11 document.write("</br>Buscando por "+fruta+"...</br>");
12
13 ▼ for(i=0;i<6;i++){
14   if(frutas[i]==fruta){
15     document.write(i+" - Encontrei! </br>");
16   }else{
17     document.write(i+" - Não encontrei! </br>")
18   }
19 }
```

Webview x Shell x +

https://Exercicio-3-curso-js.dayanakaroline.repl.co

Exercicio 3 -
Abacaxi - Uva - Limão - Goiaba - Banana - Laranja -
Buscando por Uva...
0 - Não encontrei!
1 - Encontrei!
2 - Não encontrei!
3 - Não encontrei!
4 - Não encontrei!
5 - Não encontrei!

Exercício 5

Crie uma animação usando vetores e imagens.

Repetição condicional While ()

O while percorre um bloco de código enquanto uma condição especificada for verdadeira.

```
var num = parseInt(prompt("Digite um numero:"));

while(num!=0){
    alert(num);
    num = parseInt(prompt("Digite um numero:"));
}
```

Exercício 6

Faça um programa que soma todos os valores recebidos até que o número digitado seja 100, quando 100 for digitado, a soma deve parar.

Atenção: o número 100 não deve ser somado!

Funções

Uma função JavaScript é um bloco de código projetado para executar uma tarefa específica, quando "algo" a invoca (chama).

- Funções sem retorno
- Funções com parâmetros
- Funções com retorno

Motivação: Você pode usar o mesmo código muitas vezes com argumentos diferentes, para produzir resultados diferentes.

Funções sem retorno

```
function hello() {  
    document.write("Olá Mundo!");  
}  
  
hello();
```


Funções sem retorno

Definição da função



```
function hello() {  
    document.write("Olá Mundo!");  
}  
  
hello();
```

Funções sem retorno

Definição da função

```
function hello() {  
    document.write("Olá Mundo!");  
}
```

```
hello();
```


Chamada da função

Funções com parâmetros

```
let x = soma(4, 3);  
  
function soma(a, b) {  
    document.write("Soma = "+ (a + b));  
}
```

Funções com parâmetros

Parâmetros



```
function soma(a, b) {  
    document.write("Soma = " + (a + b));  
}  
  
soma(4, 3);
```

Funções com parâmetros

Parâmetros

```
function soma(a, b) {  
    document.write("Soma = " + (a + b));  
}
```

```
soma(4, 3);
```

Argumentos

Funções com retorno

Retorno da função

```
function Celsius(f) {  
  return (5/9) * (f-32);  
}
```

```
var temp = Celsius(157);
```

```
document.write("Temperatura: "+temp+" °C");
```

Funções com retorno

```
function Celsius(f) {  
    return (5/9) * (f-32);  
}
```

```
var temp = Celsius(157);
```

```
document.write("Temperatura: "+temp+" °C");
```

Variável recebe o
retorno da função

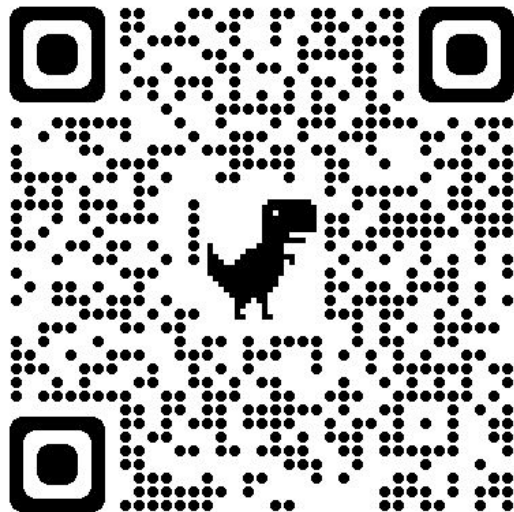
Exercício 8

Crie uma tela de menu, onde cada tela será uma função.

Projeto 1

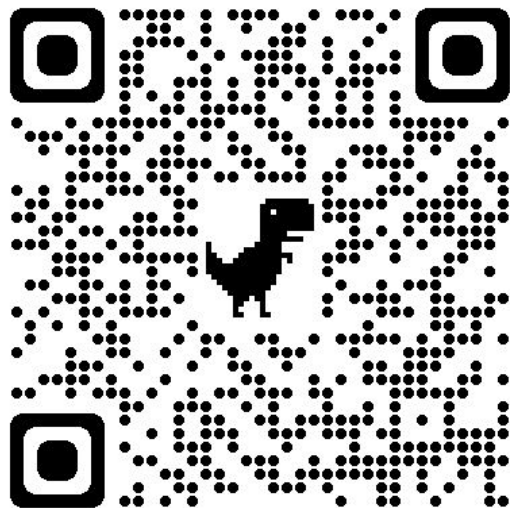
Jogo : Pedra Papel ou Tesoura?

Referências



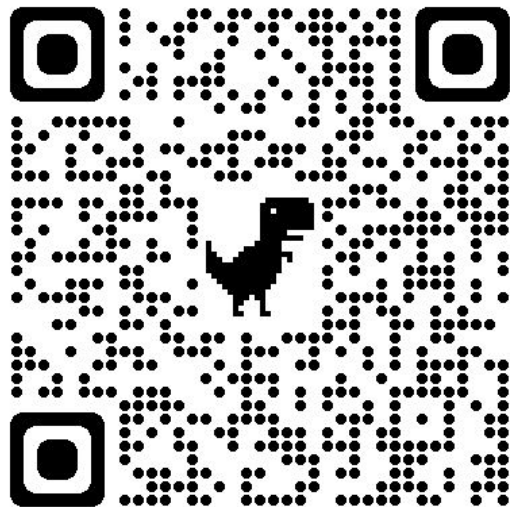
[W3schools JavaScript Tutorial](#)

Referências



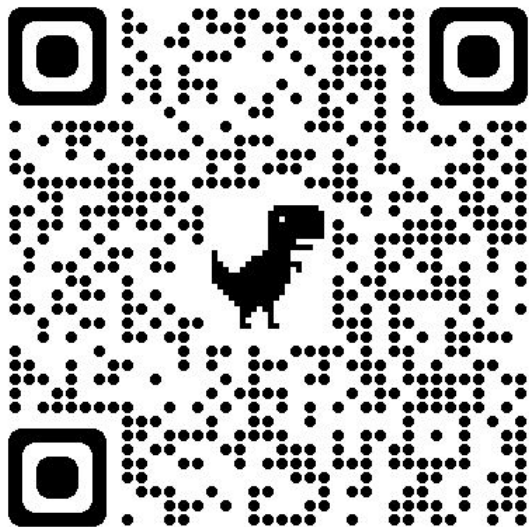
[Referências P5JS](#)

Referências



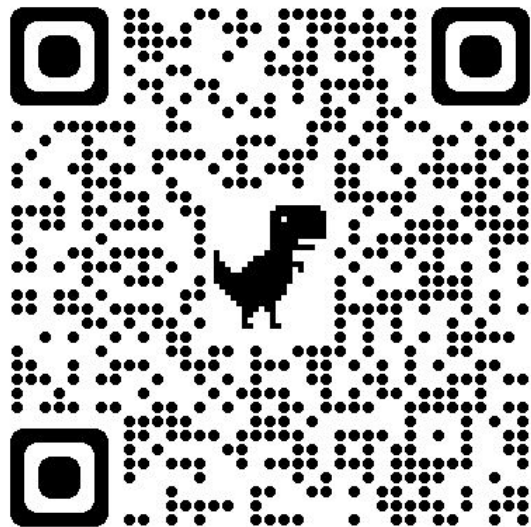
[Replit](#)

Referências



[Open Game Art](#)

Referências



[Minha Página](#)