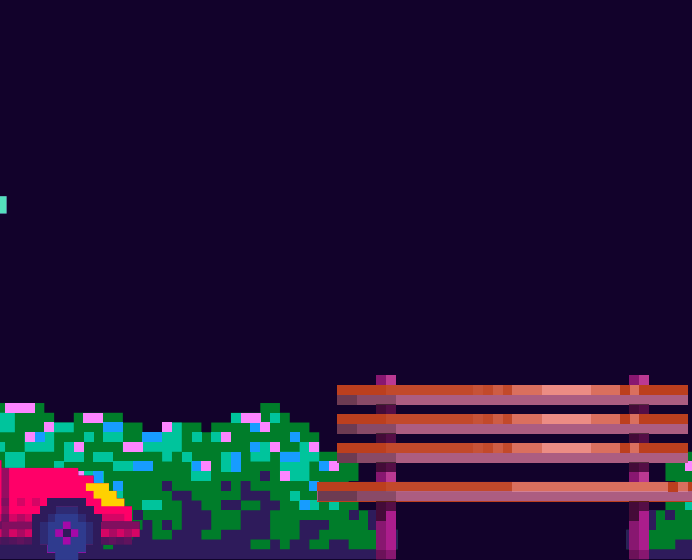




HELLO
WORLD



LUGARES E INTERAÇÕES



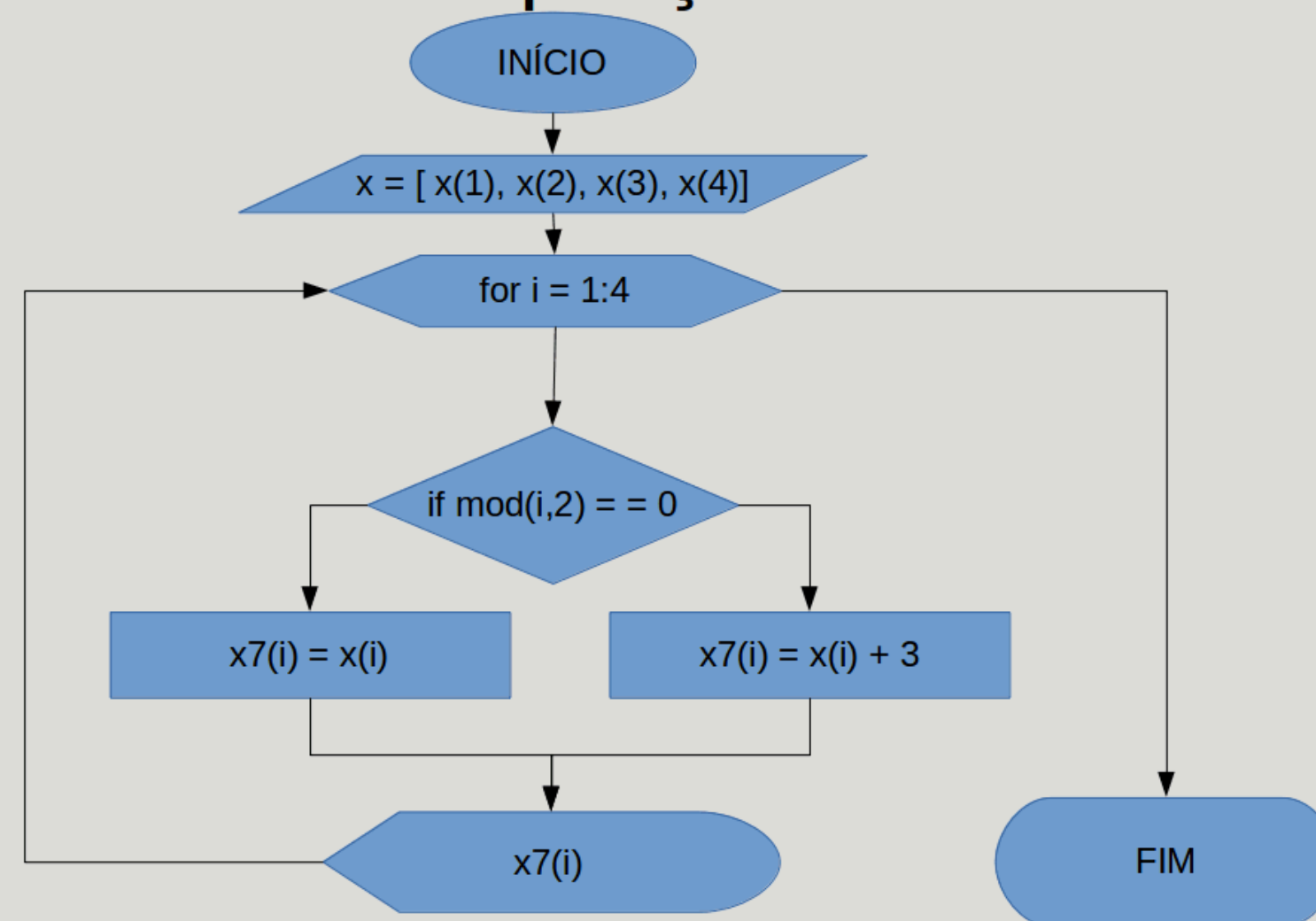
LAÇOS

Laços oferecem um jeito fácil e rápido de executar uma ação repetidas vezes.

Você pode pensar em um laço de repetição como um jogo onde você manda o seu personagem andar X passos em uma direção e Y passos em outra; por exemplo, a ideia "vá 5 passos para leste" pode ser expressa em um laço desta forma:

```
var passo;  
for (passo = 0; passo < 5; passo++) {  
    // Executa 5 vezes, com os valores de passos de 0 a 4.  
    console.log("Ande um passo para o leste");  
}
```

Estrutura de repetição e condicional



DECLARAÇÃO FOR

Quando um for é executado, ocorre o seguinte:

1. A expressão `expressao Inicial` é inicializada e, caso possível, é executada. Normalmente essa expressão inicializa um ou mais contadores, mas a sintaxe permite expressões de qualquer grau de complexidade. Podendo conter também declaração de variáveis.
2. A expressão `condicao` é avaliada. caso o resultado de `condicao` seja verdadeiro, o laço é executado. Se o valor de `condicao` é falso, então o laço terminará. Se a expressão `condicao` é omitida, a `condicao` é assumida como verdadeira.
3. A instrução é executada. Para executar múltiplas declarações, use uma declaração em bloco (`{ ... }`) para agrupá-las.
4. A atualização da expressão `incremento`, se houver, executa, e retorna o controle para o passo 2.

```
for ([expressaoInicial]; [condicao]; [incremento])  
    declaracao
```

```
1 <form name="selectForm">
2   <p>
3     <label for="tipoMusica"
4       >Escolha alguns tipos de música, em seguida, clique no botão
5       abaixo:</label>
6     >
7     <select id="tipoMusica" name="tipoMusica" multiple="multiple">
8       <option selected="selected">R&B</option>
9       <option>Jazz</option>
10      <option>Blues</option>
11      <option>New Age</option>
12      <option>Classico</option>
13      <option>Ópera</option>
14    </select>
15  </p>
16  <p><input id="btn" type="button" value="Quantos foram selecionados?" /></p>
17 </form>
18
19 <script>
20   function howMany(selectObject) {
21     var numeroSelecionadas = 0;
22     for (var i = 0; i < selectObject.options.length; i++) {
23       if (selectObject.options[i].selected) {
24         numeroSelecionadas++;
25       }
26     }
27     return numeroSelecionadas;
28   }
29
30   var btn = document.getElementById("btn");
31   btn.addEventListener("click", function () {
32     alert(
33       "Total de opções selecionadas: " + howMany(document.selectForm.tipoMusica)
34     );
35   });
36 </script>
37
```



DECLARAÇÃO WHILE

Uma declaração while executa suas instruções, desde que uma condição especificada seja avaliada como verdadeira. Segue uma declaração while.

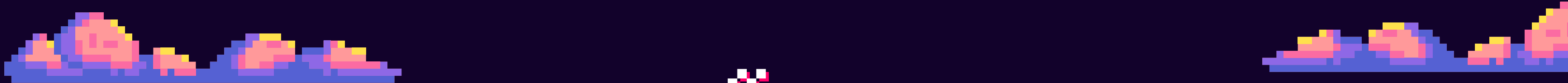
Se a condição se tornar falsa, a declaração dentro do laço para a execução e o controle é passado para a instrução após o laço.

O teste da condição ocorre antes que o laço seja executado. Desta forma se a condição for verdadeira o laço executará e testará a condição novamente. Se a condição for falsa o laço termina e passa o controle para as instruções após o laço. Para executar múltiplas declarações, use uma declaração em bloco ({ ... }) para agrupar essas declarações.

```
while (condicao)
    declaracao
```



```
1 <body>
2   <h1>Exemplo de Loop while</h1>
3   <div id="resultado"></div>
4
5   <script>
6     let n = 0;
7     let x = 0;
8     let resultadoDiv = document.getElementById("resultado");
9
10    while (n < 3) {
11      n++;
12      x += n;
13      resultadoDiv.innerHTML += `n: ${n}, x: ${x}<br>`;
14    }
15  </script>
```

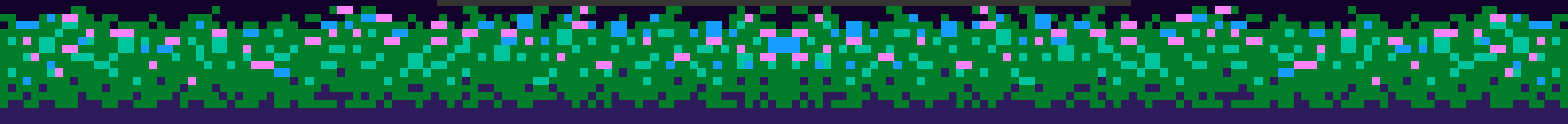


DECLARAÇÃO DO...WHILE

A instrução do...while repetirá até que a condição especificada seja falsa.

A instrução será executada uma vez antes da condição ser verificada. Para executar múltiplas instruções utilize uma declaração de bloco ({ ... }) para agrupá-las. Caso a condicao seja verdadeira, então o laço será executado novamente. Ao final de cada execução, a condicao é verificada. Quando a condição contida no while for falsa a execução do laço é terminada e o controle é passado para a instrução seguinte a do...while.

```
do  
  
    declaracao  
  
while (condicao);
```





```
1 <h1>Exemplo de Loop do...while</h1>
2   <div id="resultado"></div>
3
4   <script>
5       let i = 0;
6       let resultadoDiv = document.getElementById("resultado");
7
8       do {
9           i += 1;
10          resultadoDiv.innerHTML += `Valor de i: ${i}<br>`;
11      } while (i < 5);
12  </script>
```



DECLARAÇÃO BREAK

Use break para terminar laços, switch, ou um conjunto que utiliza label.

- Quando você utiliza break sem um label, ele encerrará imediatamente o laço mais interno while, do-while, for, ou switch e transferirá o controle para a próxima instrução.
- Quando você utiliza break com um label, ele encerrará o label específico.

Segue a sintaxe do break:

1. break;
2. break label;

```
Enquanto (condição verdadeira) {  
    // Bloco de código a ser executado  
    Se (condição de saída) {  
        Quebre o loop  
    }  
    // Mais código aqui, se necessário  
}
```

```
1 <body>
2   <h1>Exemplo de Loop while com break</h1>
3   <div id="resultado"></div>
4
5   <script>
6     let z = 1;
7     let x = 0;
8     let resultadoDiv = document.getElementById("resultado");
9
10    // Rotulagem do loop
11    labelCancelaLaco: while (true) {
12      // Executa apenas se z for menor ou igual a 5
13      if (z ≤ 5) {
14        resultadoDiv.innerHTML += `Laço interior: ${z}<br>`;
15        z += 1;
16      } else {
17        // Se z for maior que 5, verifica a condição de saída
18        if (z === 10 && x === 10) {
19          resultadoDiv.innerHTML += `Condição de saída atingida: z = ${z}, x = ${x}<br>`;
20          break labelCancelaLaco;
21        } else if (z === 10) {
22          resultadoDiv.innerHTML += `Z atingiu 10, saindo do loop<br>`;
23          break;
24        }
25      }
26    }
27  </script>
28 </body>
```



GAME RULES

01

Um campo de entrada onde o usuário pode digitar o número para o qual deseja gerar a tabuada.

Um campo de entrada adicional para definir até que número a tabuada deve ser gerada (por exemplo, de 1 a 10).

02

Utilize um laço de repetição (for, while, etc.) para calcular e exibir os resultados da tabuada na tela.

Cada resultado da multiplicação deve ser exibido de forma clara e organizada na interface do usuário.

03

Permita que o usuário altere o número base da tabuada e o limite de geração sem recarregar a página.

Atualize dinamicamente a visualização da tabuada conforme o usuário faz alterações nos campos de entrada.



MEUS CONTATOS:



27 99500-7495



<https://beacons.ai/prismatech>



producaoprismatech@gmail.com



Avenida Jerônimo Monteiro 145, Vitória





THANK
YOU