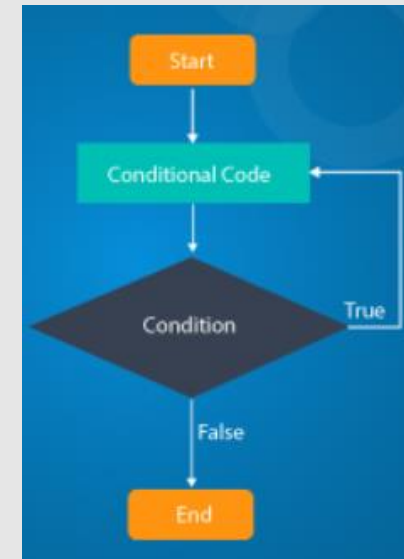


Desenvolvimento Web

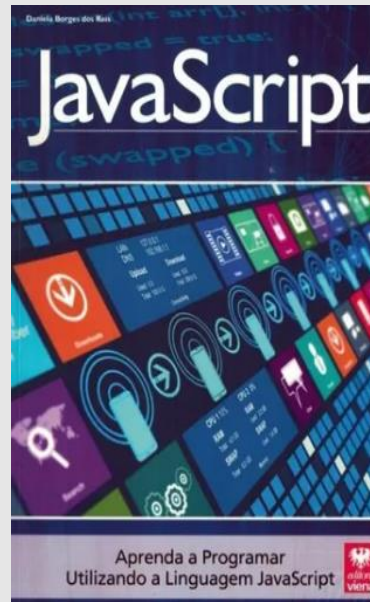
Unidade 2 – Parte 2

JS – Estruturas de Controle



Prof. Aparecido V. de Freitas
 Doutor em Engenharia
 da Computação pela EPUSP
aparecido.freitas@online.uscs.edu.br
aparecidovfreitas@gmail.com

Bibliografia



Programação com Desvios – Introdução

- ◆ Para que um programa de computador possa tomar **decisões**, é necessário que se criem **condições** para que a decisão ocorra;
- ◆ Assim, para se tomar uma decisão, é necessário que exista alguma condição;
- ◆ Uma condição pode produzir uma de duas respostas: uma resposta **verdadeira** ou uma resposta **falsa**.



Programação com Desvios

- ♦ A **condição**, pode ser definida por meio de **operadores** relacionais.

Operador Relacional	Descrição
==	Igual a
!=	Diferente de
>	Maior que
<	Menor que
>=	Maior do que ou igual a
<=	Menor do que ou igual a

Operadores Relacionais

☀ São condições válidas as seguintes declarações relacionais:

$A == B$, $A != B$, $A > B$, $A < B$, $A >= B$, $A <= B$ ou
 $A == 5$, $A != 5$, $A > 5$, $A < 5$, $A >= 5$, $A <= 5$.

- ☀ Por meio da **condição**, pode-se em um programa estabelecer-se ações de **desvios condicionais**;
- ☀ Para uma condição **verdadeira**, uma ação pode ser **executada**;
- ☀ Se **falsa**, outra ação pode ou não ser executada;
- ☀ Assim, desvios condicionais podem ser simples ou compostos.

Desvios Condicionais Simples

- ◆ Um desvio condicional simples ocorrerá quando a avaliação da expressão lógica for verdadeira;
- ◆ Se for **falsa**, **não ocorrerá desvio**. Nesse caso, o controle do fluxo do programa seguirá normalmente.

```
if <(condição)>
{
    <instrução 1 executada quando condição for verdadeira>;
    <instrução 2 executada quando condição for verdadeira>;
    <instrução N executada quando condição for verdadeira>;
}
<instrução executada após condição ser verdadeira e/ou falsa>;
```


Desvios Condicionais Simples

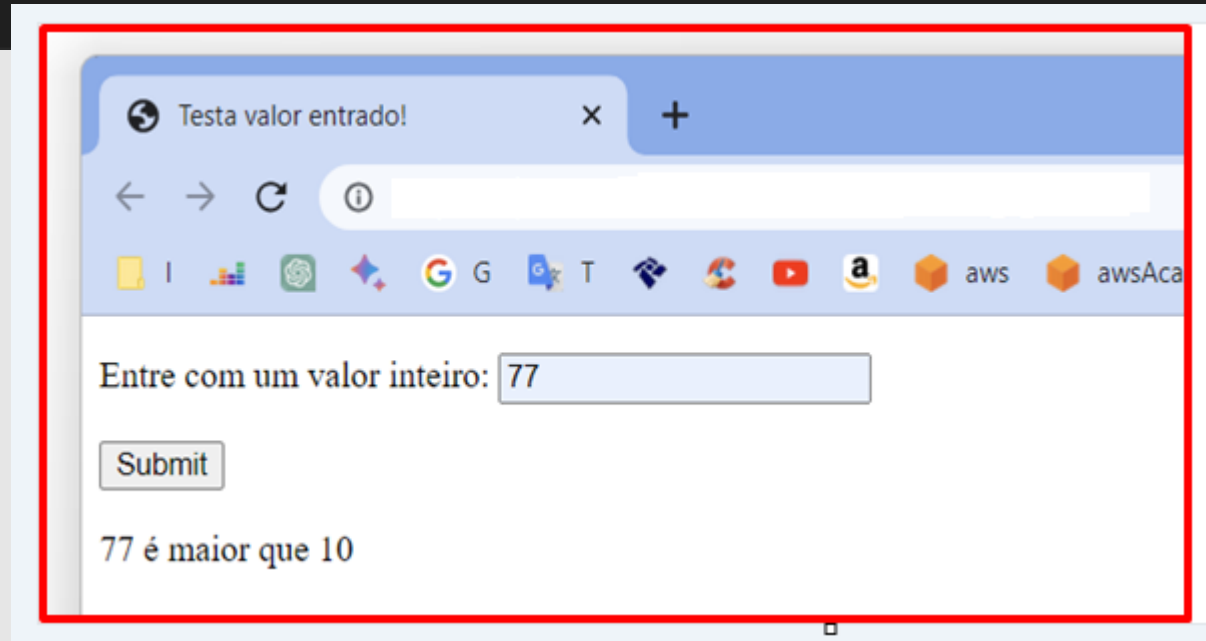
```

1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <!------->
4  <head>
5      <meta charset="UTF-8">
6      <meta http-equiv="X-UA-Compatible" content="IE=edge">
7      <meta name="viewport" content="width=device-width, initial-scale=1.0">
8
9      <script>
10         function testaValor() {
11
12             let valor = document.getElementById("valor").value;
13             let resposta = document.getElementById("resposta");
14
15             valor = Number(valor);
16
17             if (valor > 10) {
18                 resposta.textContent = valor + " é maior que 10";
19             }
20         }
21     </script>
  
```

Desvios Condicionais Simples

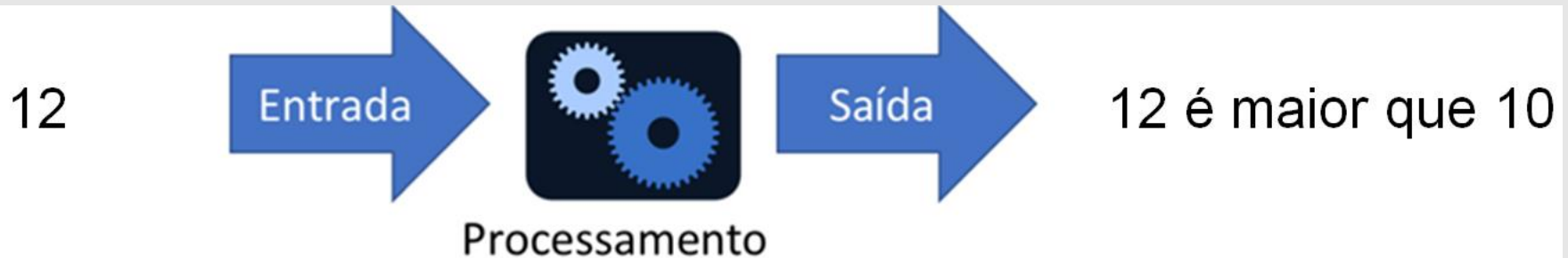
```

22 | <title>Testa valor entrado!</title>
23 | </head>
24 | <!------->
25 | <body>
26 |   <form>
27 |     <p><label>Entre com um valor inteiro: <input type="text" name = "valor" id="valor"></label></p>
28 |     <input type="button" value="Submit" onclick="testaValor()">
29 |   </form>
30 |   <p id="resposta"></p>
31 | </body>
32 | <!------->
33 | </html>
  
```



Desvios Condicionais Simples

- ❖ **Exemplo:** Codificar um algoritmo que receba um número e exiba uma mensagem caso o valor entrada seja maior que 10.



Desvios Condicionais Simples

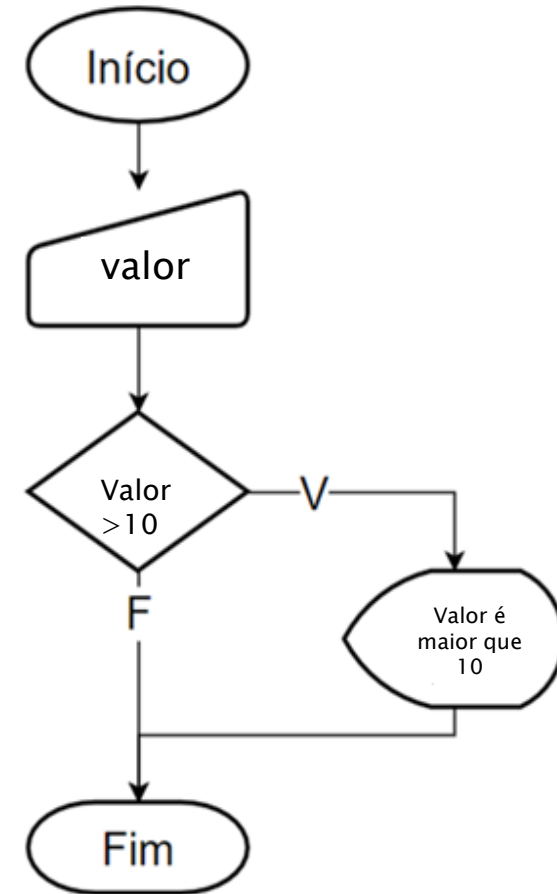
- ◆ **Exemplo:** Codificar um algoritmo que receba um número e exiba uma mensagem caso o valor entrada seja maior que 10.

```
<script>
function testaValor() {

    let valor = document.getElementById("valor").value;
    let resposta = document.getElementById("resposta");

    valor = Number(valor);

    if (valor > 10) {
        resposta.textContent = valor + " é maior que 10";
    }
}
</script>
```



Desvios Condicionais Simples

- ♦ **Exemplo:** Codificar um algoritmo que receba um número e exiba uma mensagem caso o valor entrada seja **maior** que 10.

Teste de mesa com a entrada abaixo:



Desvios Condicionais Simples

```
<script>
function testaValor() {
  let valor = document.getElementById("valor").value;
  let resposta = document.getElementById("resposta");

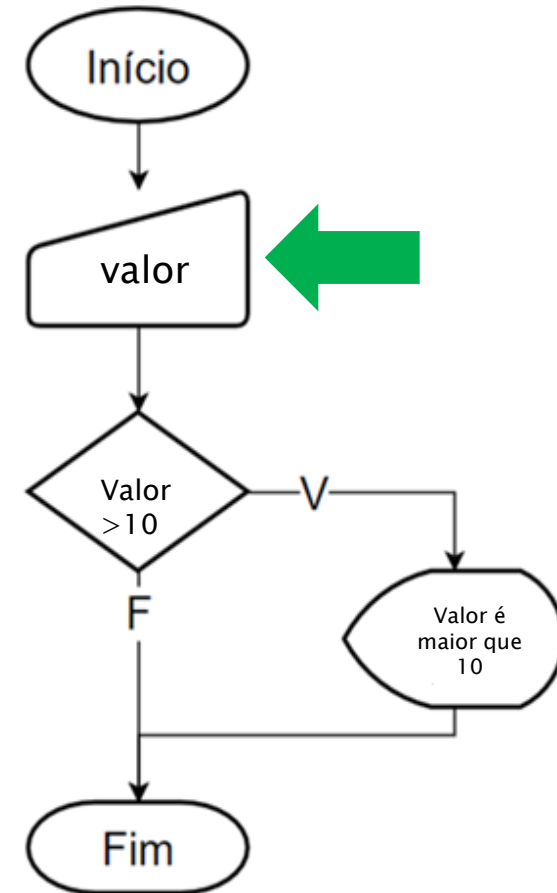
  valor = Number(valor);

  if (valor > 10) {
    resposta.textContent = valor + " é maior que 10";
  }
}
</script>
```

valor

12

Memória do Programa: os dados ficam na memória durante a execução do programa



Desvios Condicionais Simples

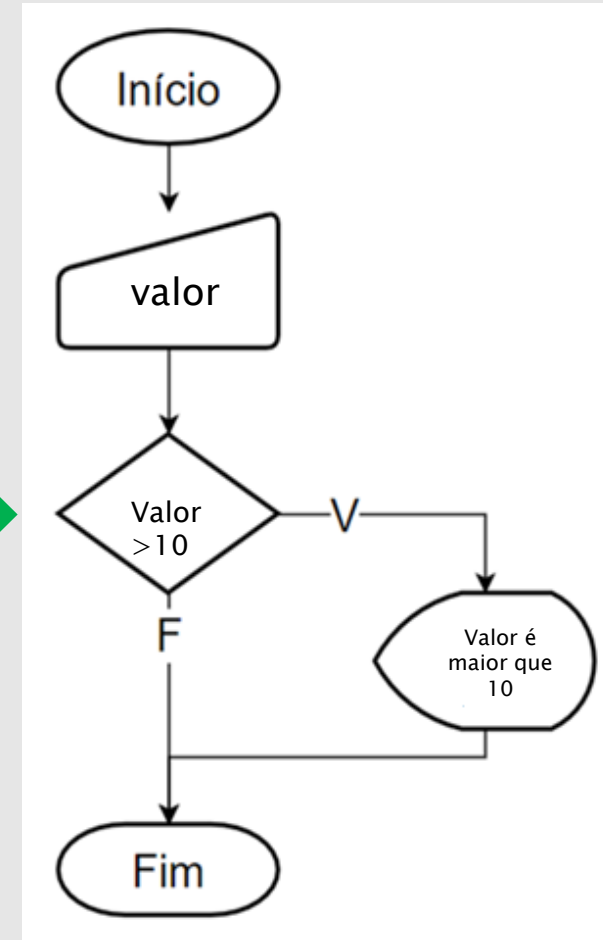
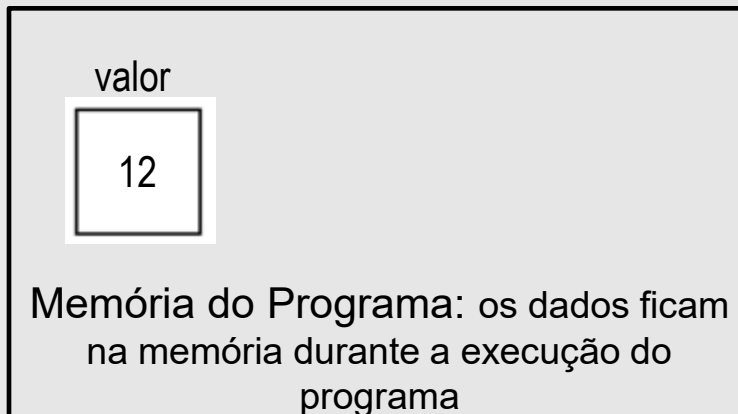
```
<script>
function testaValor() {

    let valor = document.getElementById("valor").value;
    let resposta = document.getElementById("resposta");

    valor = Number(valor);

    if (valor > 10) {
        resposta.textContent = valor + " é maior que 10";
    }
}
</script>
```

verdadeiro



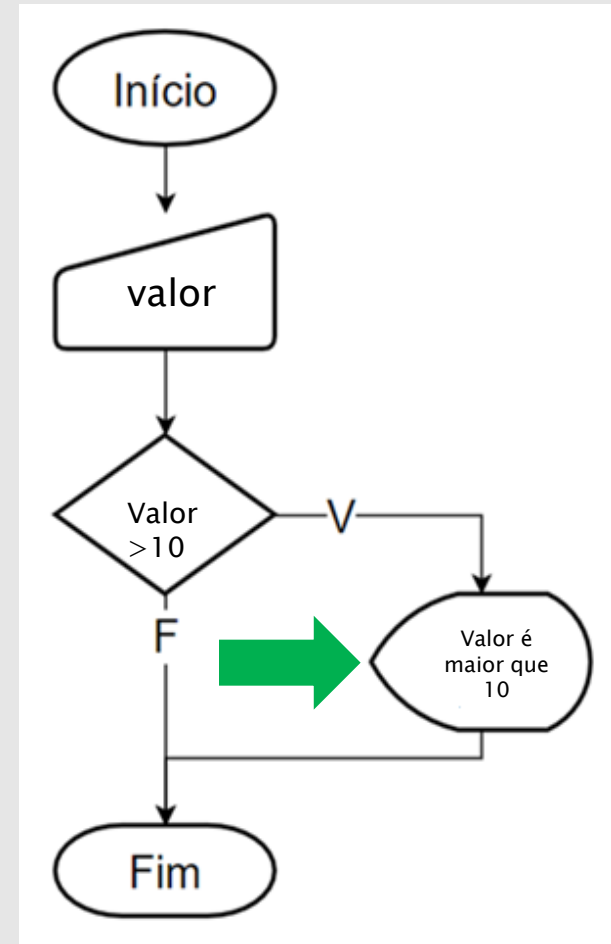
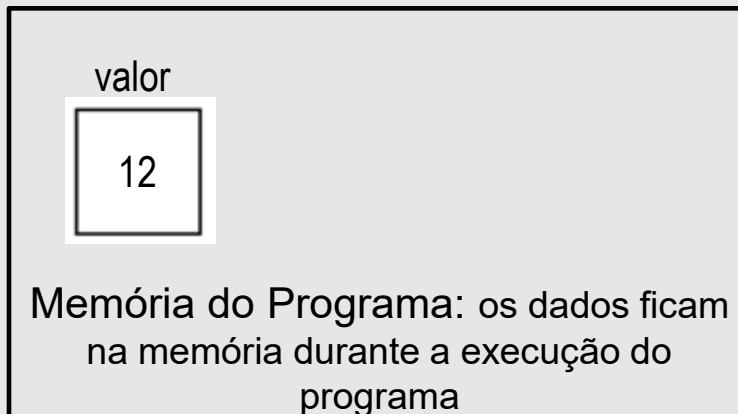
Desvios Condicionais Simples

```
<script>
function testaValor() {

    let valor = document.getElementById("valor").value;
    let resposta = document.getElementById("resposta");

    valor = Number(valor);

    if (valor > 10) {
        resposta.textContent = valor + " é maior que 10";
    }
}
</script>
```



Desvios Condicionais Simples

```
<script>
function testaValor() {

    let valor = document.getElementById("valor").value;
    let resposta = document.getElementById("resposta");

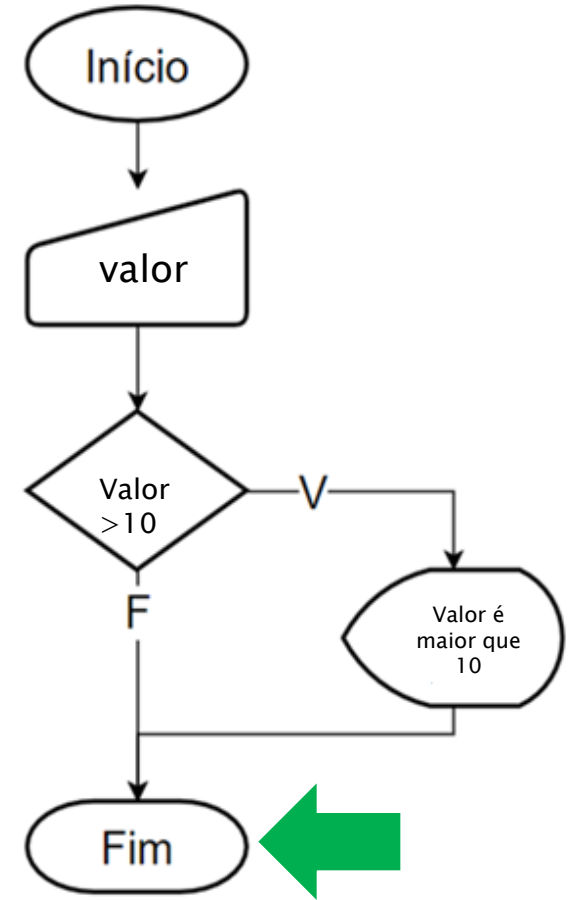
    valor = Number(valor);

    if (valor > 10) {
        resposta.textContent = valor + " é maior que 10";
    }
}
</script>
```

Valor

12

Memória do Programa: os dados ficam na memória durante a execução do programa



Desvios Condicionais Simples

- ♦ **Exemplo:** Codificar um algoritmo que receba um número e exiba uma mensagem caso o valor entrada seja **menor** que 10.

Teste de mesa com a entrada abaixo:



Desvios Condicionais Simples

```
<script>
function testaValor() {
  let valor = document.getElementById("valor").value;
  let resposta = document.getElementById("resposta");

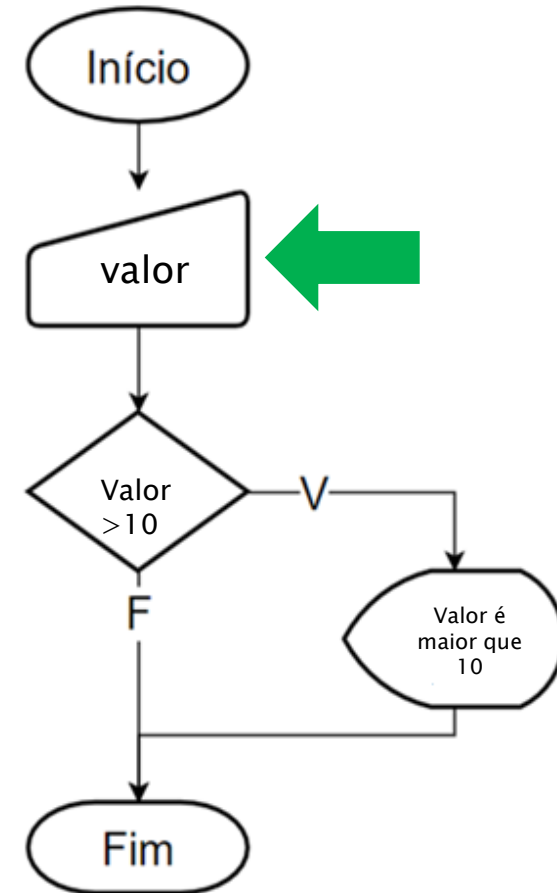
  valor = Number(valor);

  if (valor > 10) {
    resposta.textContent = valor + " é maior que 10";
  }
}
</script>
```

valor

3

Memória do Programa: os dados ficam na memória durante a execução do programa



Desvios Condicionais Simples

```
<script>
function testaValor() {

    let valor = document.getElementById("valor").value;
    let resposta = document.getElementById("resposta");

    valor = Number(valor);

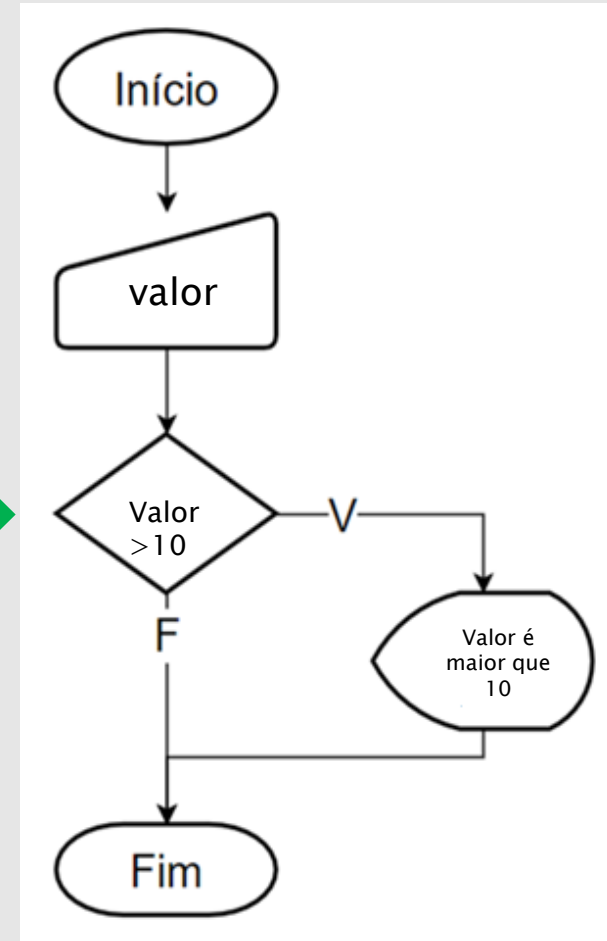
    if (valor > 10) {
        resposta.textContent = valor + " é maior que 10";
    }
}
</script>
```

falso

valor

3

Memória do Programa: os dados ficam na memória durante a execução do programa



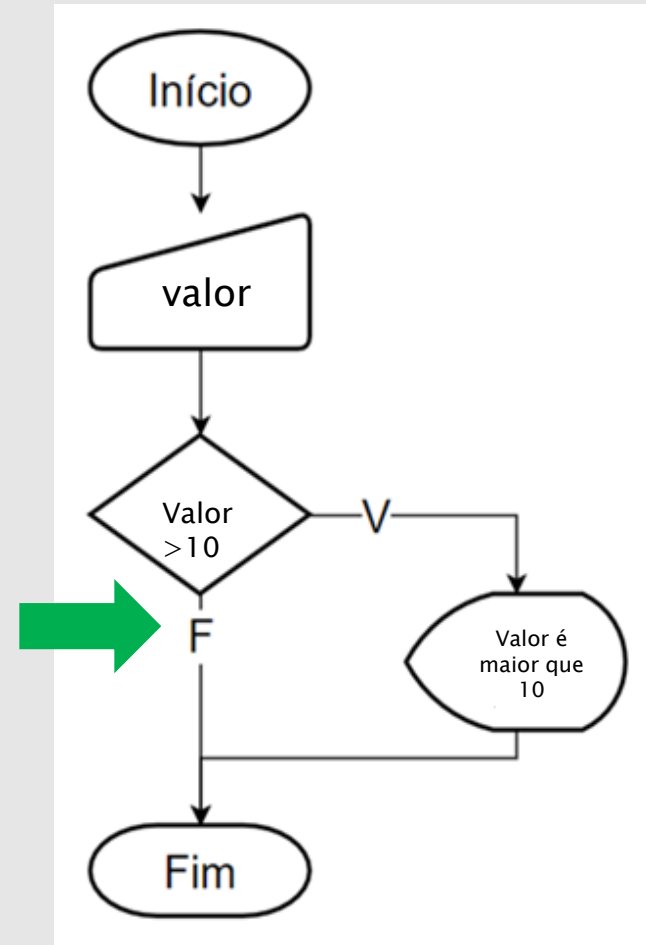
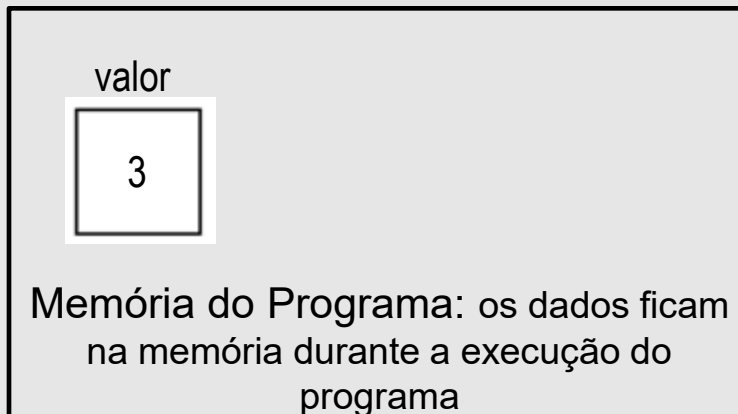
Desvios Condicionais Simples

```
<script>
function testaValor() {

    let valor = document.getElementById("valor").value;
    let resposta = document.getElementById("resposta");

    valor = Number(valor);

    if (valor > 10) {
        resposta.textContent = valor + " é maior que 10";
    }
}
</script>
```



Desvios Condicionais Simples

```
<script>
function testaValor() {

    let valor = document.getElementById("valor").value;
    let resposta = document.getElementById("resposta");

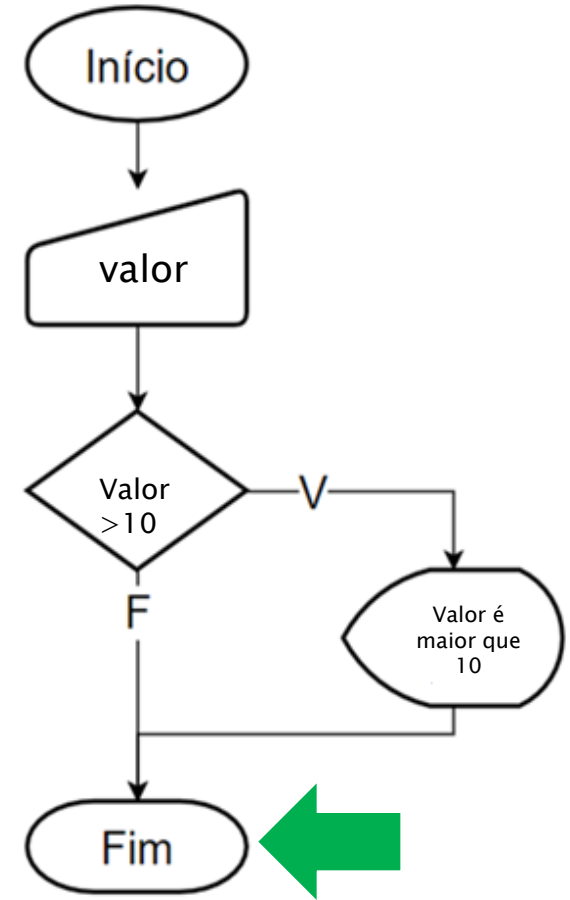
    valor = Number(valor);

    if (valor > 10) {
        resposta.textContent = valor + " é maior que 10";
    }
}
</script>
```

Valor

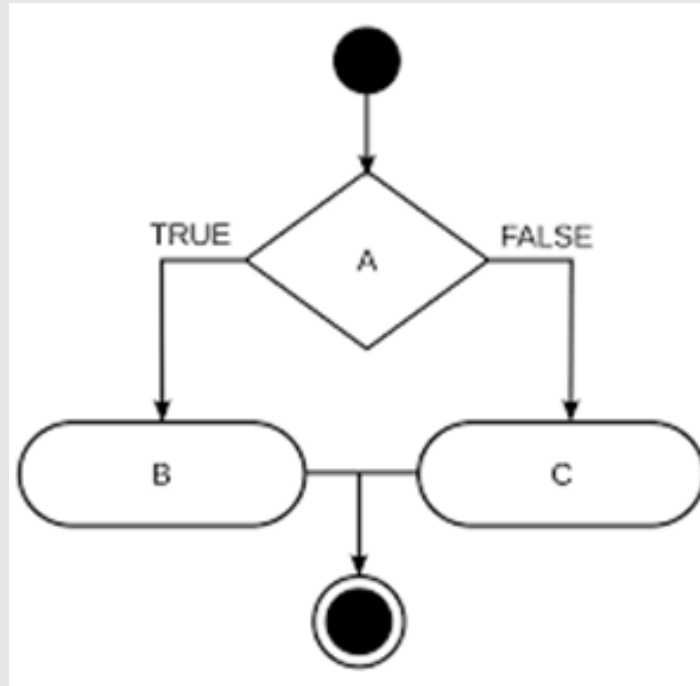
3

Memória do Programa: os dados ficam na memória durante a execução do programa



Desvio Condicional Composto

- ❖ Em uma instrução **if... else**, se a condição for verdadeira, será executada a instrução posicionada entre as instruções **if** e **else**.
- ❖ Caso a condição seja **falsa**, será executada a instrução posicionada logo após o **else**.

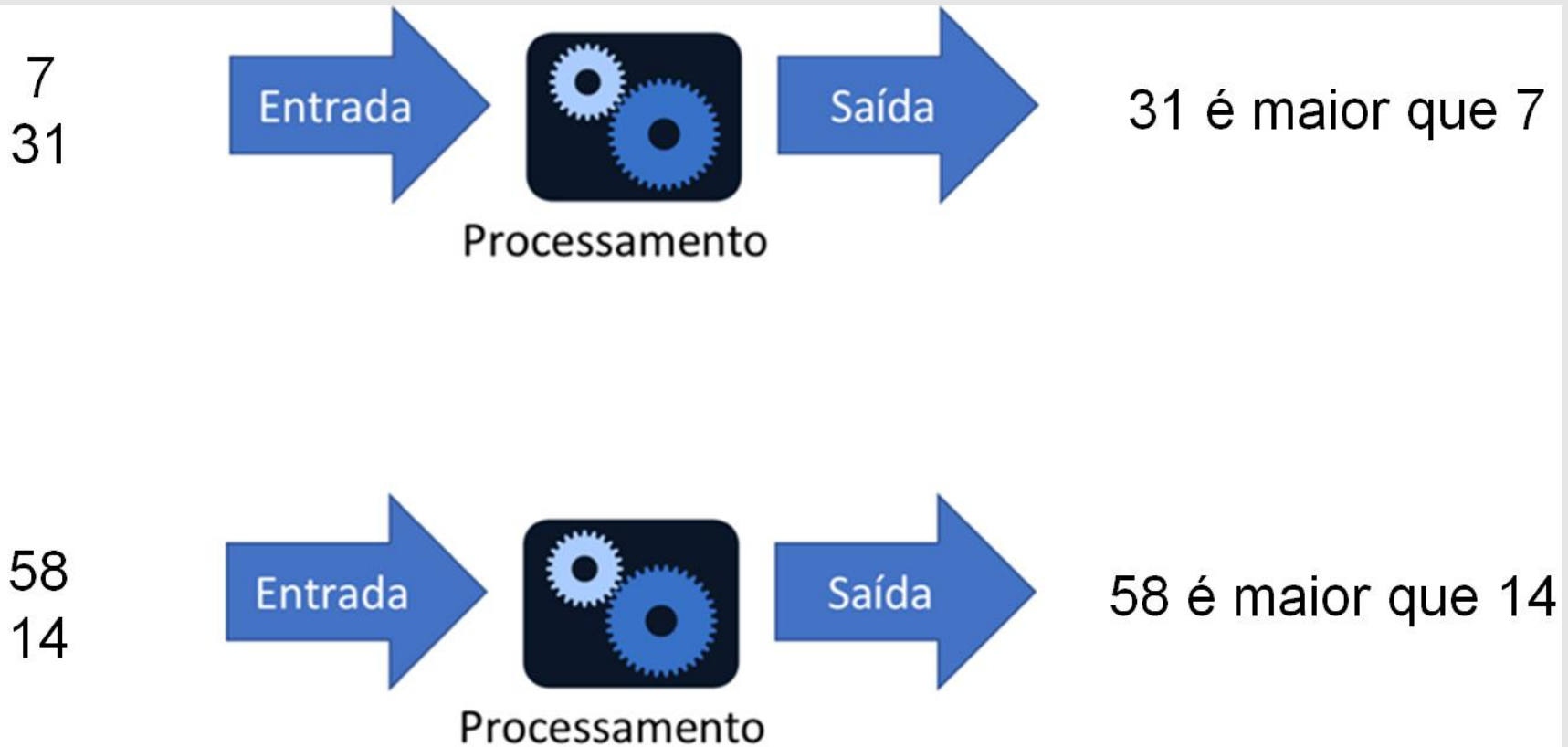


Desvio Condicional Composto

```
if <(condição)>
{
    <instrução 1 executada quando condição for verdadeira>;
    <instrução 2 executada quando condição for verdadeira>;
    <instrução N executada quando condição for verdadeira>;
}
else
{
    <instrução 1 executada quando condição for falsa>;
    <instrução 2 executada quando condição for falsa>;
    <instrução N executada quando condição for falsa>;
}
<instrução executada após condição ser verdadeira ou falsa>;
```

Desvio Condicional Composto

- ◆ **Exemplo:** Codificar um algoritmo que leia dois valores inteiros distintos e informe qual é o maior.



Desvio Condicional Composto

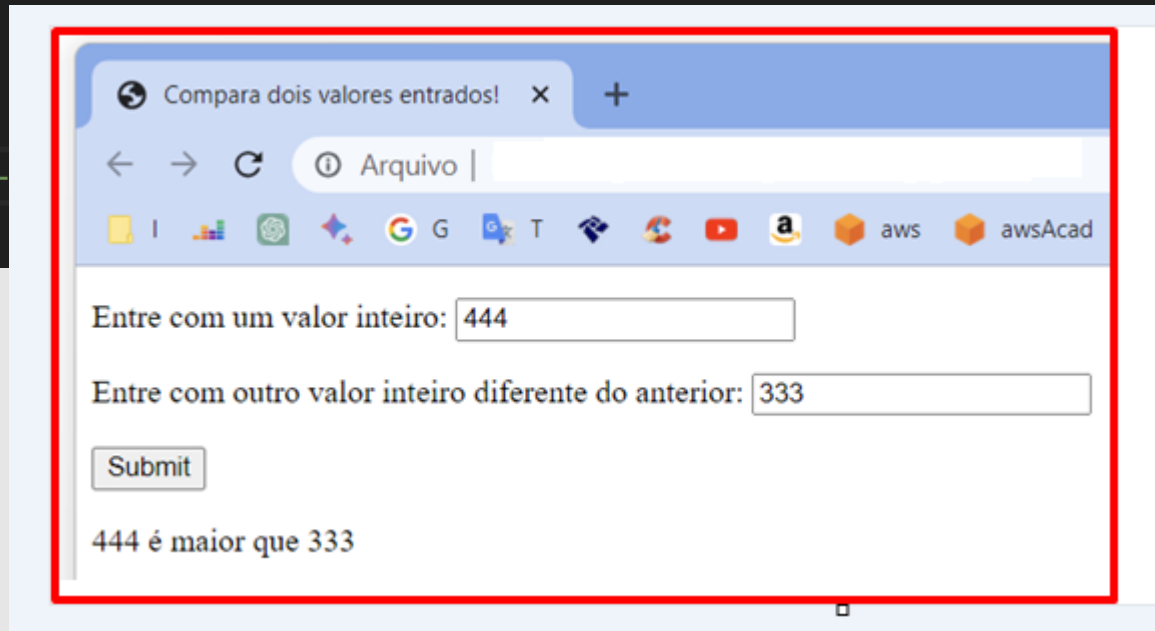
```
1 <!DOCTYPE html>
2 <html lang="pt-br">
3 <!------->
4 <head>
5   <meta charset="UTF-8">
6   <meta http-equiv="X-UA-Compatible" content="IE=edge">
7   <meta name="viewport" content="width=device-width, initial-scale=1.0">
8
9   <script>
10     function comparaDoisValores() {
11
12       let x = document.getElementById("x").value;
13       let y = document.getElementById("y").value;
14
15
16       let resposta = document.getElementById("resposta");
17
18       x = Number(x);
19       y = Number(y);
20
21       if (x > y) {
22         resposta.textContent = x + " é maior que " + y;
23       }
24       else {
25         resposta.textContent = y + " é maior que " + x;
26       }
27     }
28   </script>
```

Desvio Condicional Composto

```

29  <title>Compara dois valores entrados!</title>
30  </head>
31  <!------->
32  <body>
33    <form>
34      <p><label>Entre com um valor inteiro: <input type="text" name = "x" id="x"></label></p>
35      <p><label>Entre com outro valor inteiro diferente do anterior: <input type="text" name = "y" id="y"></label></p>
36      <input type="button" value="Submit" onclick="comparaDoisValores()">
37    </form>
38    <p id="resposta"></p>
39  </body>
40  <!------->
41  </html>

```



A screenshot of a web browser window titled "Compara dois valores entrados!". The browser's address bar shows "Arquivo |". The page contains two text input fields. The first field is labeled "Entre com um valor inteiro:" and contains the value "444". The second field is labeled "Entre com outro valor inteiro diferente do anterior:" and contains the value "333". Below these fields is a "Submit" button. At the bottom of the form, the text "444 é maior que 333" is displayed. The entire browser window is outlined with a red border.

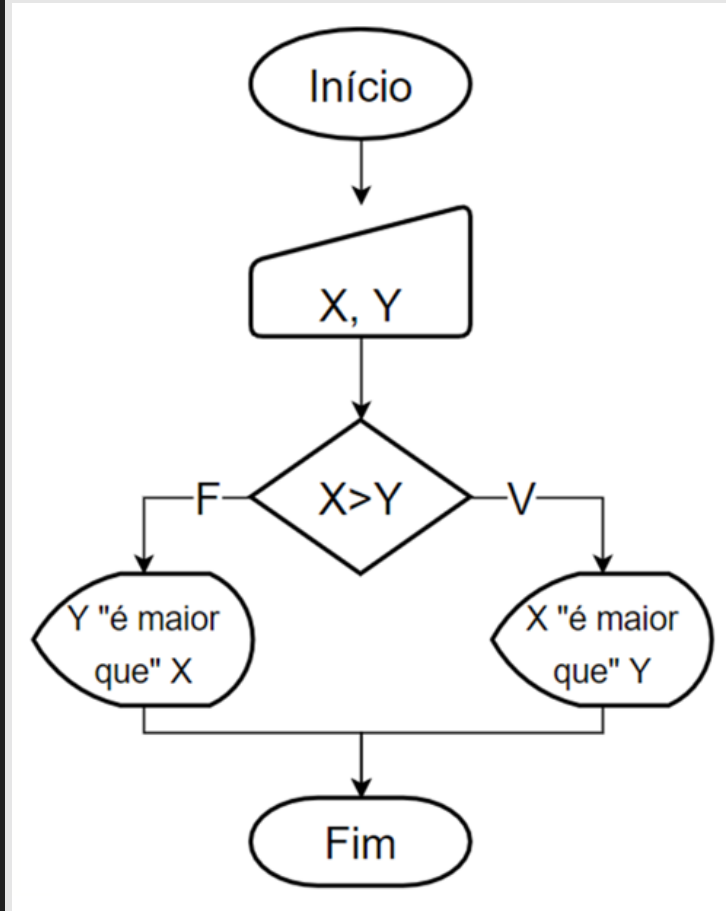
Desvio Condicional Composto

- ◆ **Exemplo:** Codificar um algoritmo que leia dois valores inteiros distintos e informe qual é o maior.

```

9  <script>
10  function comparaDoisValores() {
11
12      let x = document.getElementById("x").value;
13      let y = document.getElementById("y").value;
14
15
16      let resposta = document.getElementById("resposta");
17
18      x = Number(x);
19      y = Number(y);
20
21      if (x > y) {
22          resposta.textContent = x + " é maior que " + y;
23      }
24      else {
25          resposta.textContent = y + " é maior que " + x;
26      }
27  }
28  </script>

```



Desvio Condicional Composto

- ◆ **Exemplo:** Codificar um algoritmo que leia dois valores inteiros distintos e informe qual é o maior.

Teste de mesa com a entrada abaixo:

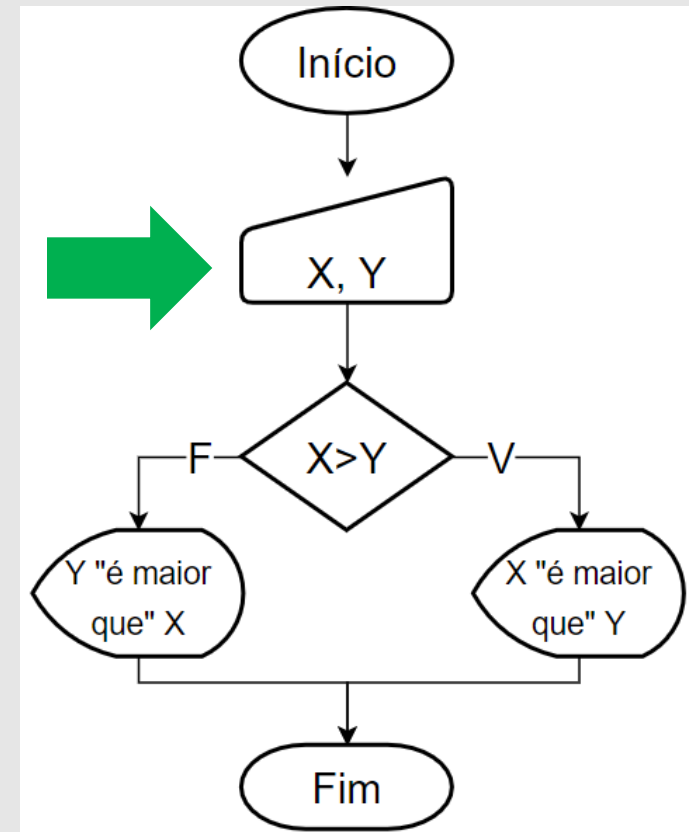


Desvio Condicional Composto

```

9  <script>
10  function comparaDoisValores() {
11
12      let x = document.getElementById("x").value;
13      let y = document.getElementById("y").value;
14
15
16      let resposta = document.getElementById("resposta");
17
18      x = Number(x);
19      y = Number(y);
20
21      if (x > y) {
22          resposta.textContent = x + " é maior que " + y;
23      }
24      else {
25          resposta.textContent = y + " é maior que " + x;
26      }
27  }
28  </script>

```



x	y
58	

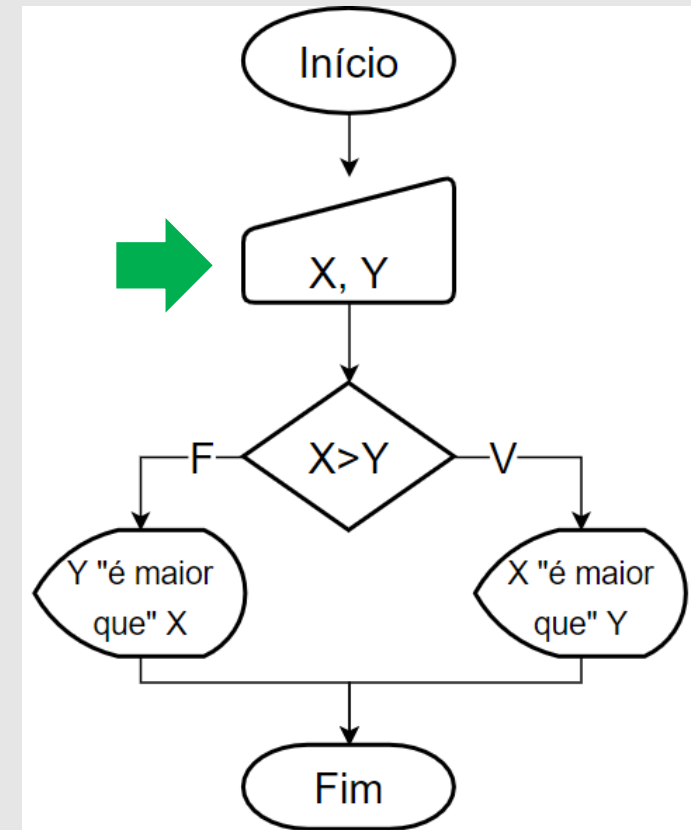
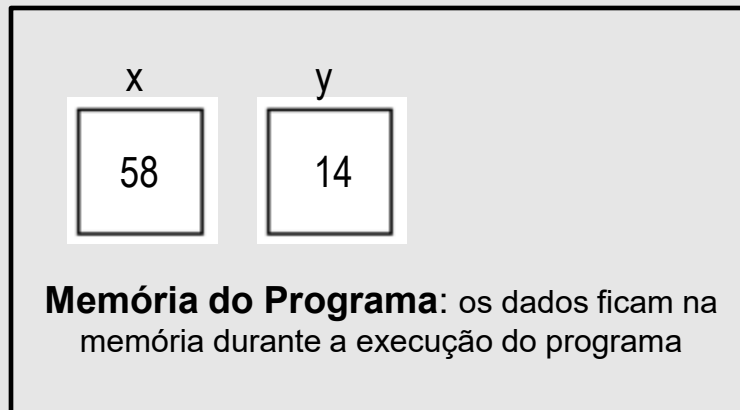
Memória do Programa: os dados ficam na memória durante a execução do programa

Desvio Condicional Composto

```

9   <script>
10  function comparaDoisValores() {
11
12      let x = document.getElementById("x").value;
13      let y = document.getElementById("y").value;
14
15
16      let resposta = document.getElementById("resposta");
17
18      x = Number(x);
19      y = Number(y);
20
21      if (x > y) {
22          resposta.textContent = x + " é maior que " + y;
23      }
24      else {
25          resposta.textContent = y + " é maior que " + x;
26      }
27  }
28  </script>

```

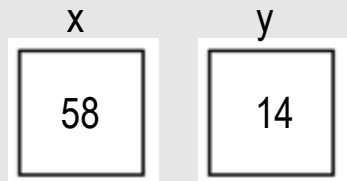
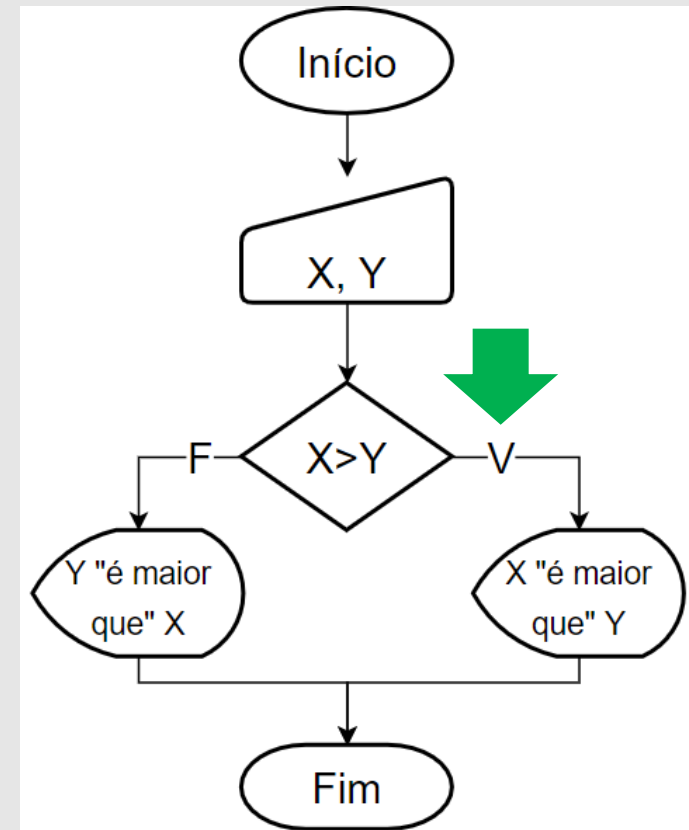


Desvio Condicional Composto

```

9      <script>
10     function comparaDoisValores() {
11
12         let x = document.getElementById("x").value;
13         let y = document.getElementById("y").value;
14
15
16         let resposta = document.getElementById("resposta");
17
18         x = Number(x);
19         y = Number(y);
20
21         verdadeiro
22         if (x > y) {
23             resposta.textContent = x + " é maior que " + y;
24         }
25         else {
26             resposta.textContent = y + " é maior que " + x;
27         }
28     }
29 </script>

```



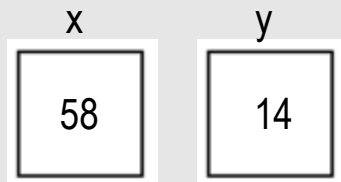
Memória do Programa: os dados ficam na memória durante a execução do programa

Desvio Condicional Composto

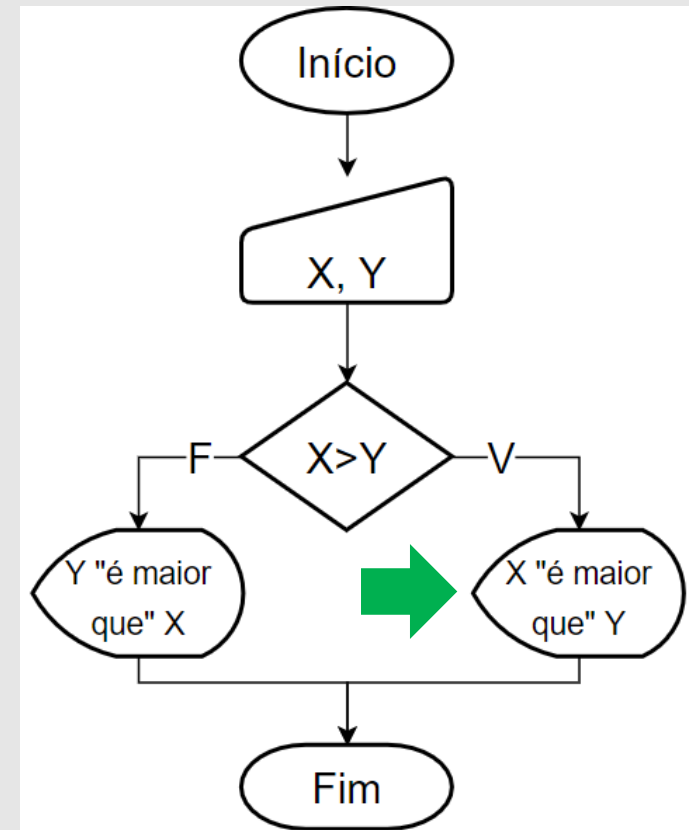
```

9  <script>
10  function comparaDoisValores() {
11
12      let x = document.getElementById("x").value;
13      let y = document.getElementById("y").value;
14
15
16      let resposta = document.getElementById("resposta");
17
18      x = Number(x);
19      y = Number(y);
20
21      if (x > y) {
22          resposta.textContent = x + " é maior que " + y;
23      }
24      else {
25          resposta.textContent = y + " é maior que " + x;
26      }
27  }
28  </script>

```



Memória do Programa: os dados ficam na memória durante a execução do programa

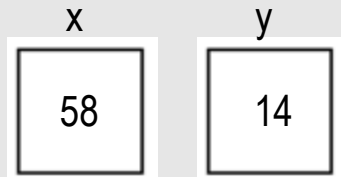


Desvio Condicional Composto

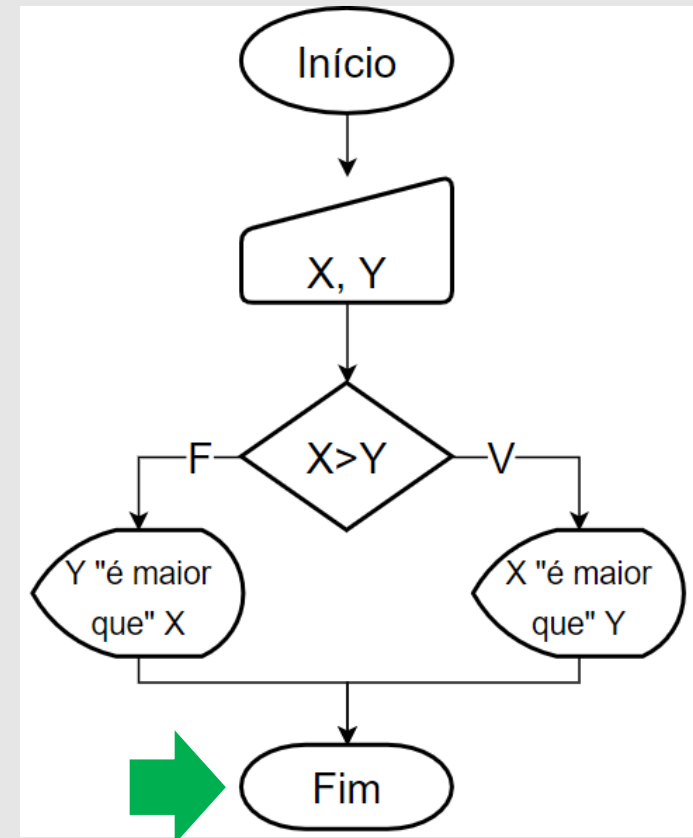
```

9  <script>
10  function comparaDoisValores() {
11
12      let x = document.getElementById("x").value;
13      let y = document.getElementById("y").value;
14
15
16      let resposta = document.getElementById("resposta");
17
18      x = Number(x);
19      y = Number(y);
20
21      if (x > y) {
22          resposta.textContent = x + " é maior que " + y;
23      }
24      else {
25          resposta.textContent = y + " é maior que " + x;
26      }
27  }
28  </script>

```



Memória do Programa: os dados ficam na memória durante a execução do programa



Desvio Condicional Composto

- ◆ **Exemplo:** Codificar um algoritmo que leia dois valores inteiros distintos e informe qual é o maior.

Teste de mesa com a entrada abaixo:

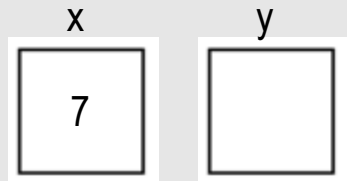


Desvio Condicional Composto

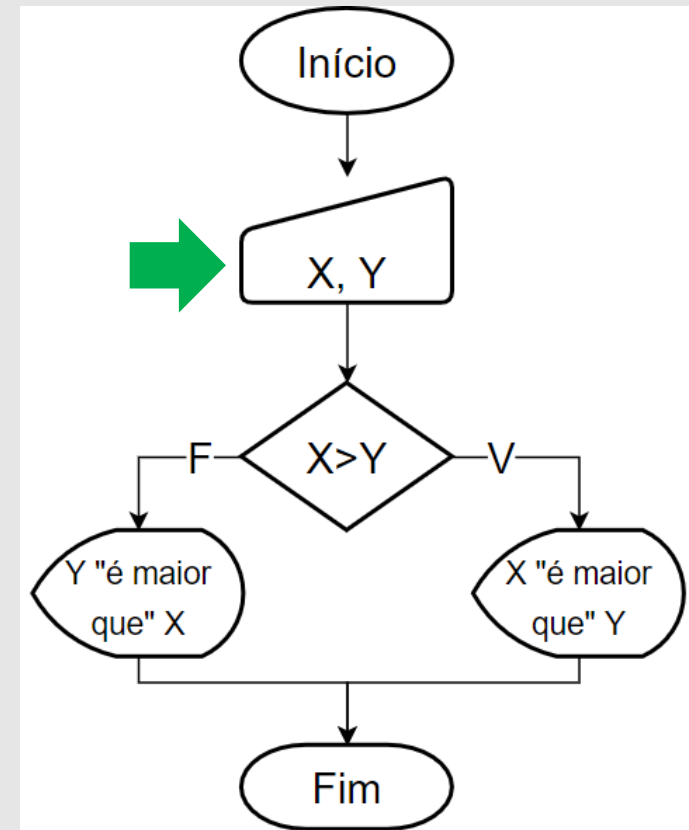
```

9  <script>
10  function comparaDoisValores() {
11
12      let x = document.getElementById("x").value;
13      let y = document.getElementById("y").value;
14
15
16      let resposta = document.getElementById("resposta");
17
18      x = Number(x);
19      y = Number(y);
20
21      if (x > y) {
22          resposta.textContent = x + " é maior que " + y;
23      }
24      else {
25          resposta.textContent = y + " é maior que " + x;
26      }
27  }
28  </script>

```



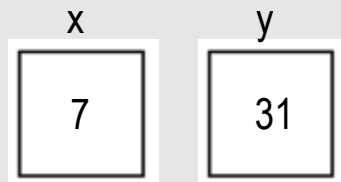
Memória do Programa: os dados ficam na memória durante a execução do programa



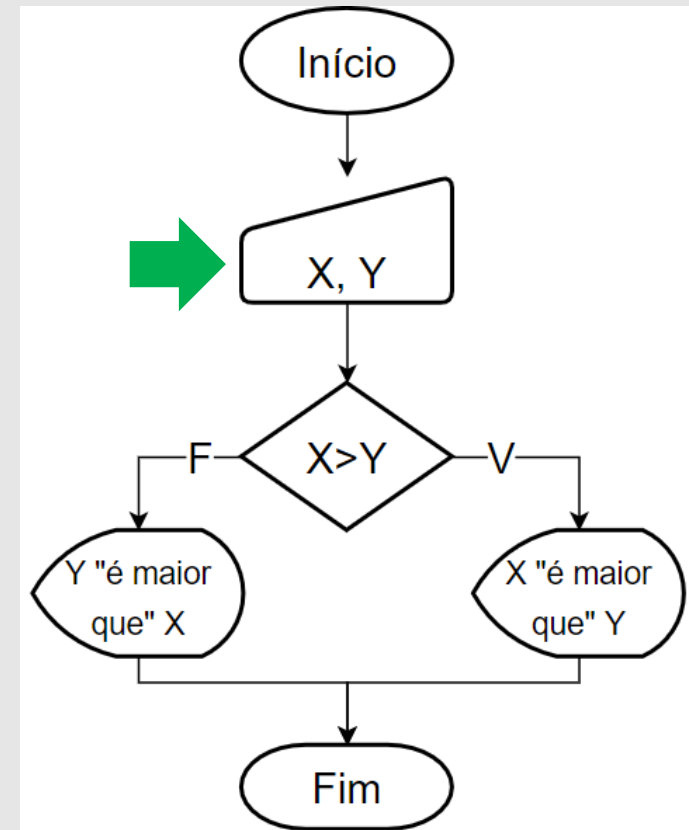
Desvio Condicional Composto

```

9   <script>
10  function comparaDoisValores() {
11
12      let x = document.getElementById("x").value;
13      let y = document.getElementById("y").value;
14
15
16      let resposta = document.getElementById("resposta");
17
18      x = Number(x);
19      y = Number(y);
20
21      if (x > y) {
22          resposta.textContent = x + " é maior que " + y;
23      }
24      else {
25          resposta.textContent = y + " é maior que " + x;
26      }
27  }
28  </script>
  
```



Memória do Programa: os dados ficam na memória durante a execução do programa



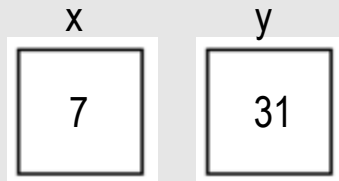
Desvio Condicional Composto

```

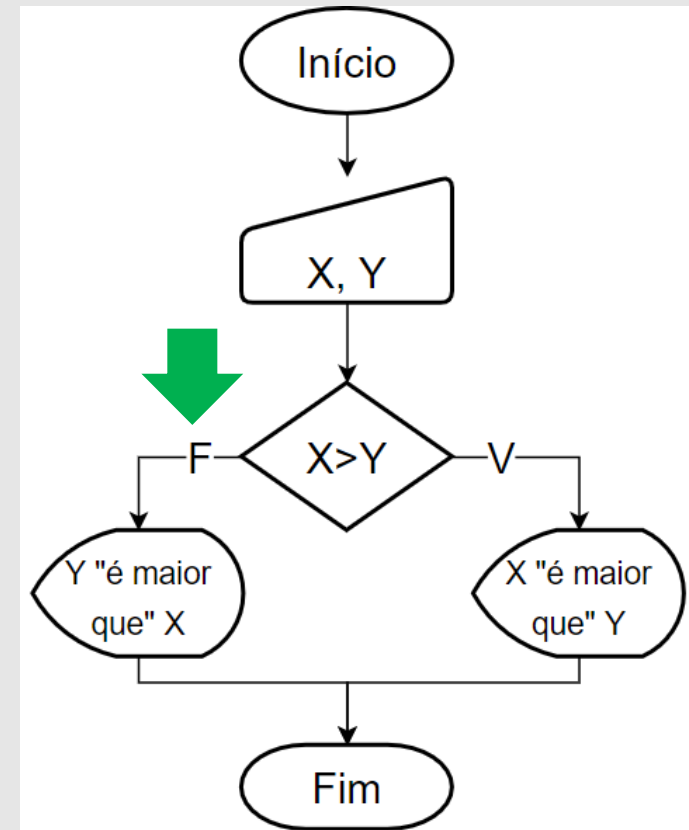
9  <script>
10  function comparaDoisValores() {
11
12      let x = document.getElementById("x").value;
13      let y = document.getElementById("y").value;
14
15
16      let resposta = document.getElementById("resposta");
17
18      x = Number(x);
19      y = Number(y);
20
21      if (x > y) {
22          resposta.textContent = x + " é maior que " + y;
23      }
24      else {
25          resposta.textContent = y + " é maior que " + x;
26      }
27  }
28  </script>

```

falso



Memória do Programa: os dados ficam na memória durante a execução do programa

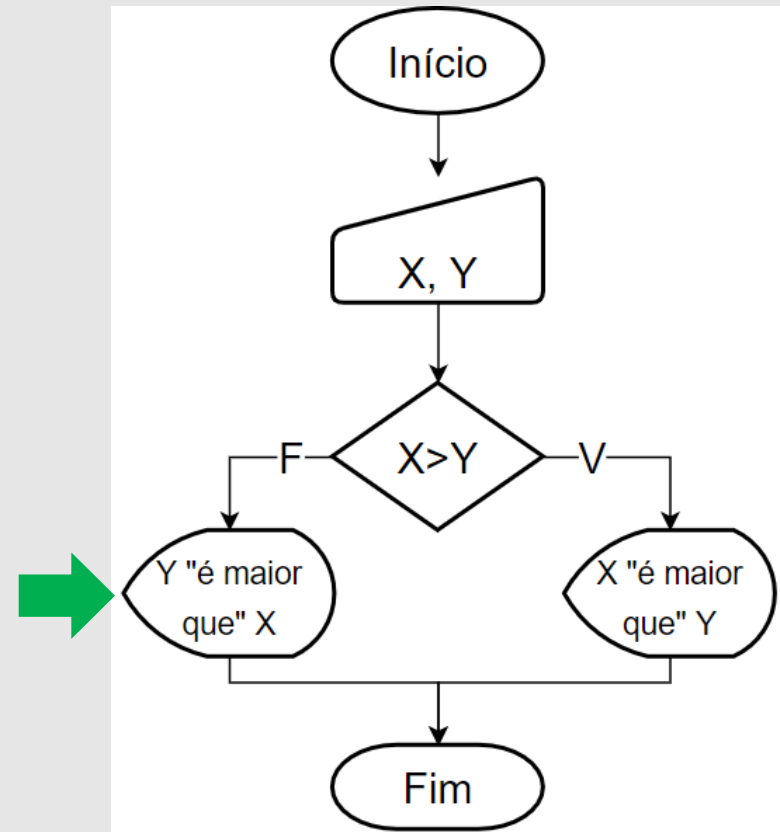
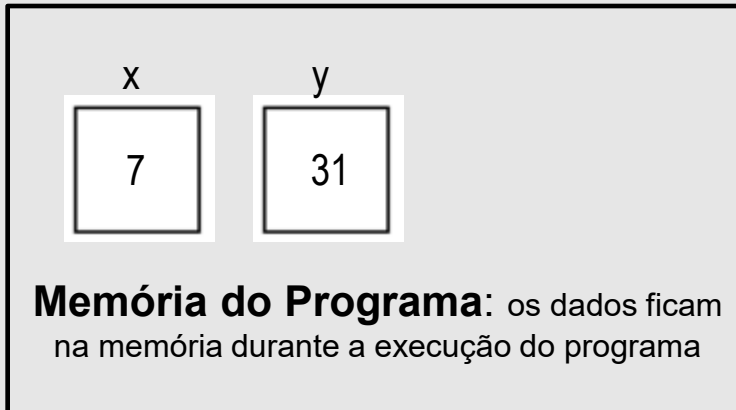


Desvio Condicional Composto

```

9  <script>
10  function comparaDoisValores() {
11
12      let x = document.getElementById("x").value;
13      let y = document.getElementById("y").value;
14
15
16      let resposta = document.getElementById("resposta");
17
18      x = Number(x);
19      y = Number(y);
20
21      if (x > y) {
22          resposta.textContent = x + " é maior que " + y;
23      }
24      else {
25          resposta.textContent = y + " é maior que " + x;
26      }
27  }
28  </script>

```

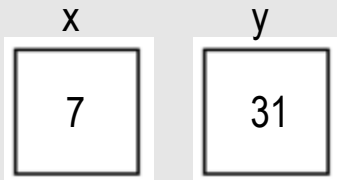


Desvio Condicional Composto

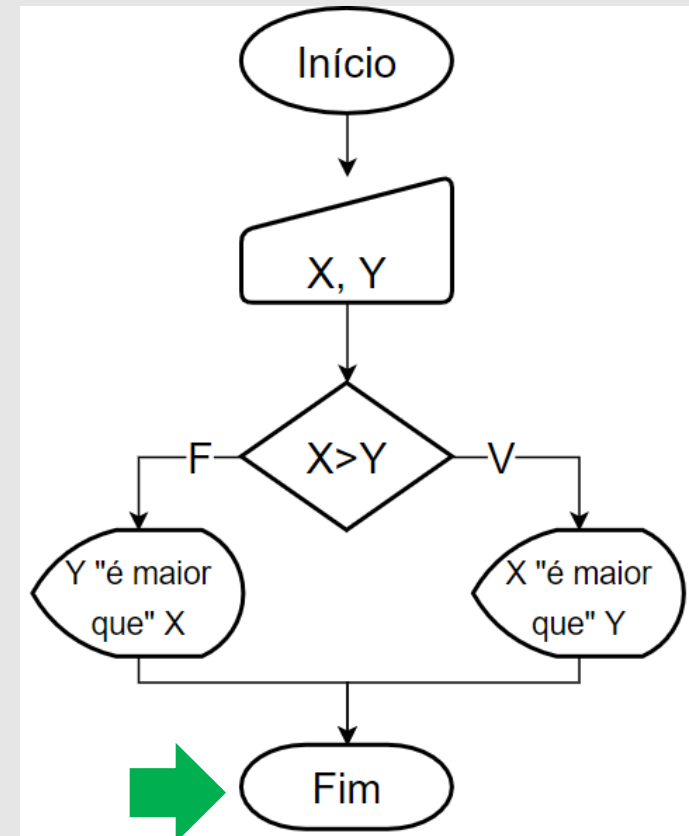
```

9  <script>
10  function comparaDoisValores() {
11
12      let x = document.getElementById("x").value;
13      let y = document.getElementById("y").value;
14
15
16      let resposta = document.getElementById("resposta");
17
18      x = Number(x);
19      y = Number(y);
20
21      if (x > y) {
22          resposta.textContent = x + " é maior que " + y;
23      }
24      else {
25          resposta.textContent = y + " é maior que " + x;
26      }
27  }
28  </script>

```



Memória do Programa: os dados ficam na memória durante a execução do programa



Desvio Condicional Seletivo

- ◆ Para grandes quantidades de desvios condicionais, pode-se empregar a estrutura de controle com múltipla escolha, denominada **switch**.



Desvio Condicional Seletivo

```
switch <variável>
{
    case <opção 1>: <operação 1>;      break;
    case <opção 2>: <operação 2>;      break;
    case <opção N>: <operação N>;      break;
    default       : <operação default>; break;
}
```

<variável> : nome da variável a ser controlada na decisão;

<opção> : conteúdo da variável a ser verificado;

<operação>: execução de alguma ação específica.

Desvio Condicional Seletivo

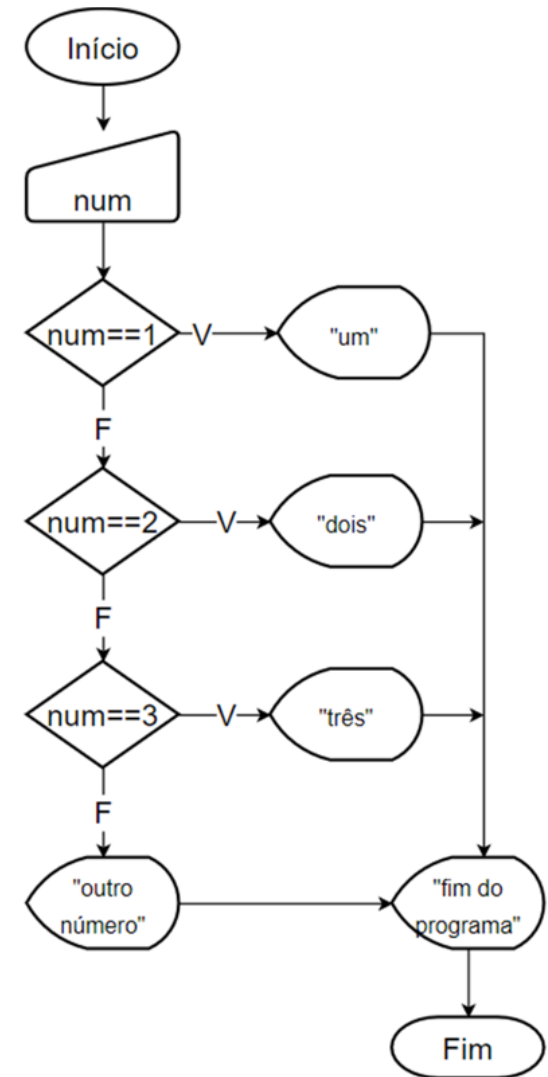
- ◆ A instrução **default** da estrutura de seleção **switch** e qualquer operação a ela associada são **opcionais**, não caracterizando obrigatoriedade de uso;
- ◆ A instrução **break** tem a finalidade de desviar o processamento para fora do comando **switch**.

Desvio Condicional Seletivo

```

1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <!------->
4
5  <head>
6    <meta charset="UTF-8">
7    <meta http-equiv="X-UA-Compatible" content="IE=edge">
8    <meta name="viewport" content="width=device-width, initial-scale=1.0">
9
10   <script>
11     function comparaDoisValores() {
12
13       let num = document.getElementById("num").value;
14       let f = document.getElementById("f");
15
16       let resposta = document.getElementById("resposta");
17
18       num = Number(num);
19

```

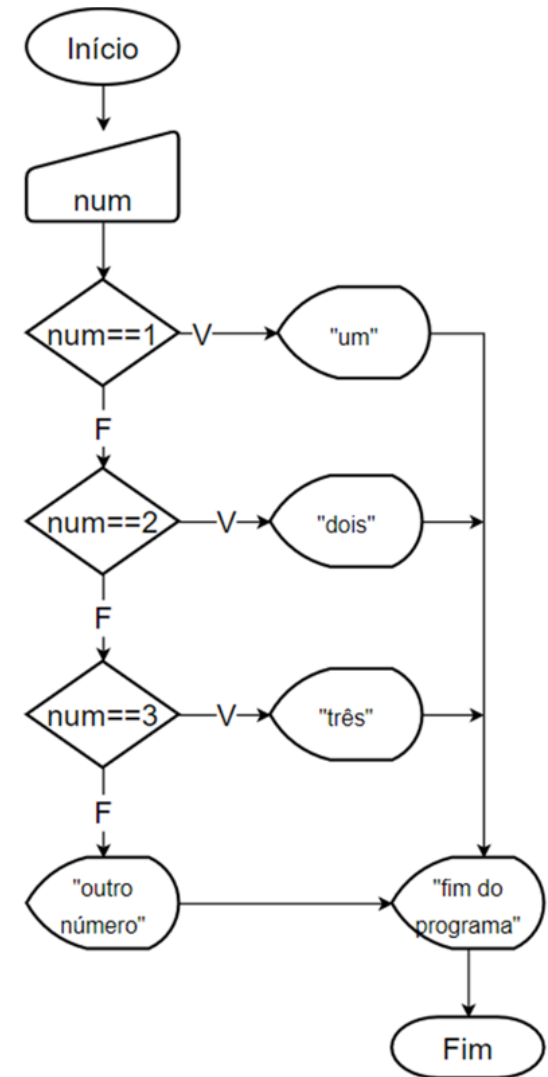


Desvio Condicional Seletivo

```

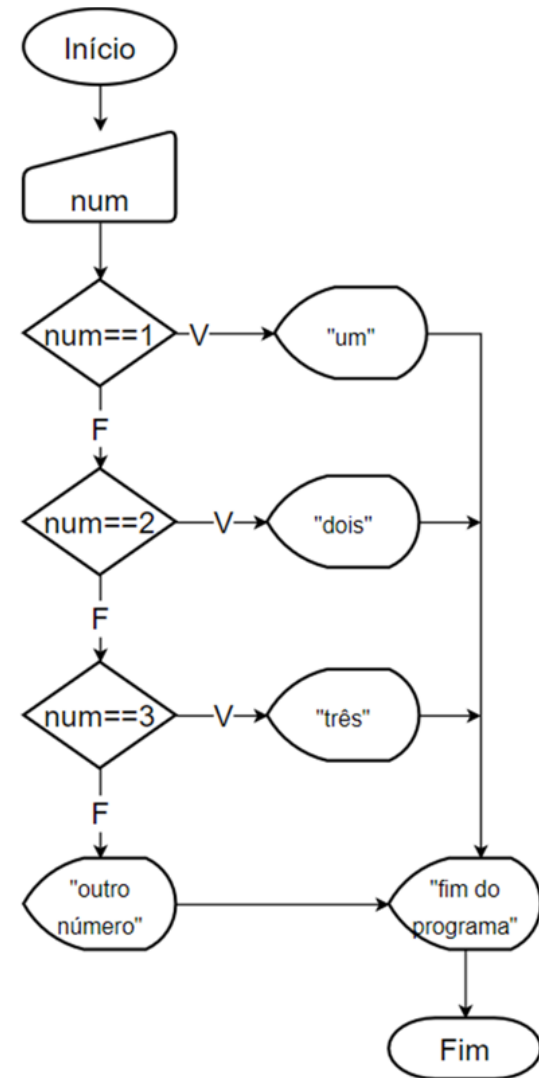
20  switch (num) {
21      case 1:
22          resposta.textContent = "um";
23          break;
24
25      case 2:
26          resposta.textContent = "dois";
27          break;
28
29      case 3:
30          resposta.textContent = "três";
31          break;
32
33      default:
34          resposta.textContent = "outro número";
35      }
36      f.textContent = "fim do programa";
37  }
38  </script>

```



Desvio Condicional Seletivo

```
40 </head>
41 <!------->
42
43 <body>
44   <form>
45     <p><label>Entre com um valor numérico : <input type="text" name="num" id="num"></label></p>
46
47     <input type="button" value="Submit" onclick="comparaDoisValores()">
48   </form>
49   <p id="resposta"></p>
50   <p id="f"></p>
51 </body>
52 <!------->
53
54 </html>
```



Desvio Condicional Seletivo

```
40 </head>
41 <!------->
42
43 <body>
44   <form>
45     <p><label>Entre com um valor numérico : <input type="text" name="num" id="num"></label></p>
46
47     <input type="button" value="Submit" onclick="comparaDoisValores()">
48   </form>
49   <p id="resposta"></p>
50   <p id="f"></p>
51 </body>
52 <!------->
53
54 </html>
```

Switch em Javascript

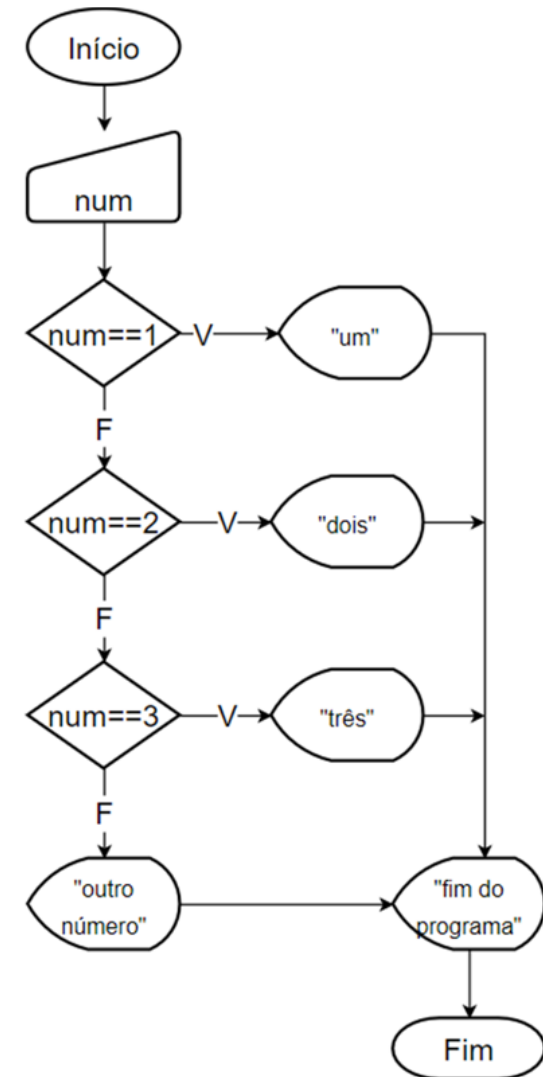
Arquivo | D:/Fontes_JS/Un...

Entre com um valor numérico : 2

Submit

dois

fim do programa



Desvio Condicional Seletivo

Teste de mesa com a entrada abaixo:



Desvio Condicional Seletivo

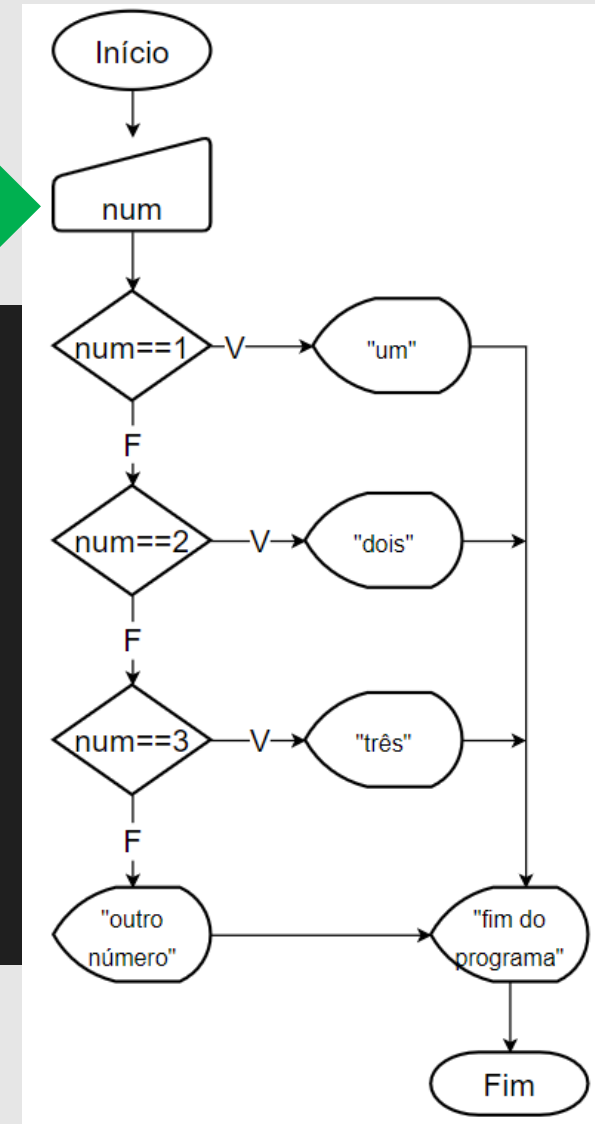
```

<script>
function comparaDoisValores() {

    let num = document.getElementById("num").value;
    let f = document.getElementById("f");

    let resposta = document.getElementById("resposta");

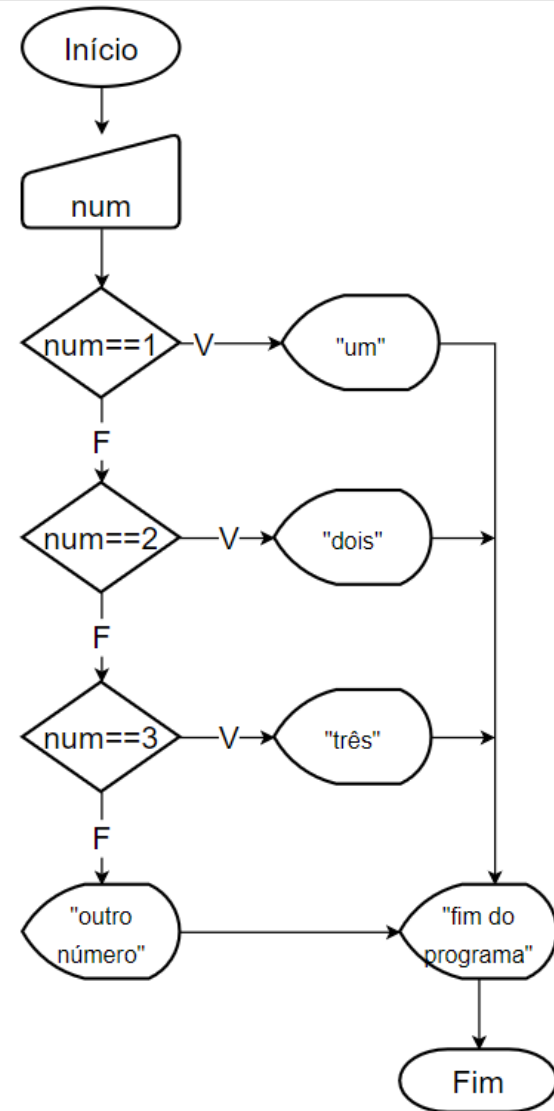
    num = Number(num);
  
```



Desvio Condicional Seletivo

```

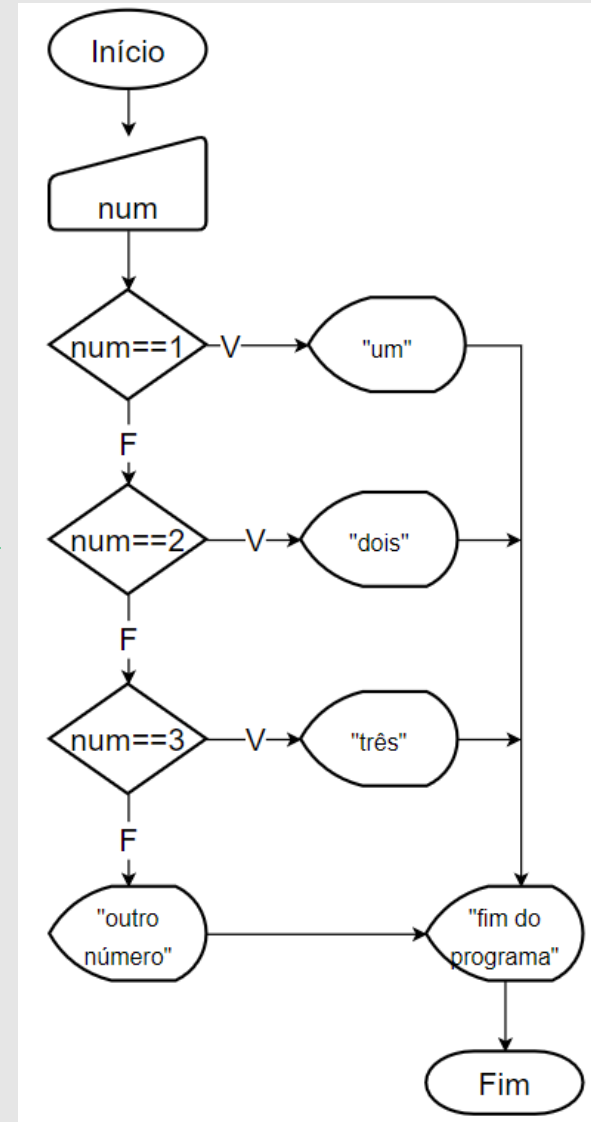
20  switch (num) {
21      case 1:
22          resposta.textContent = "um";
23          break;
24
25      case 2:
26          resposta.textContent = "dois";
27          break;
28
29      case 3:
30          resposta.textContent = "três";
31          break;
32
33      default:
34          resposta.textContent = "outro número";
35      }
36      f.textContent = "fim do programa";
37  }
38  </script>
  
```



Desvio Condicional Seletivo

```

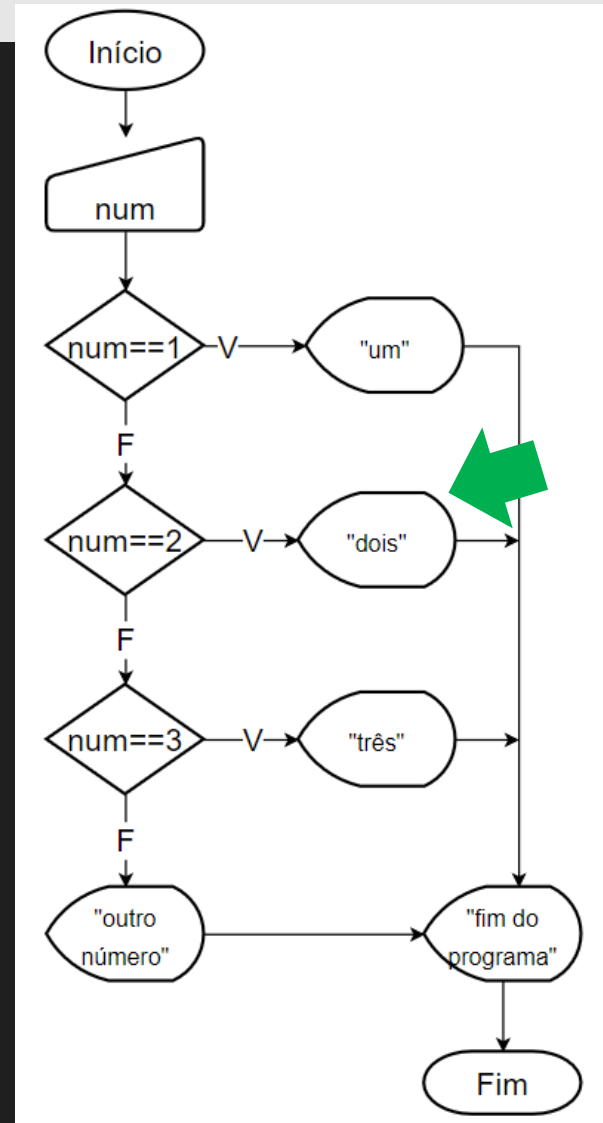
20      switch (num) {
21          case 1:
22              resposta.textContent = "um";
23              break;
24
25      case 2:
26          resposta.textContent = "dois";
27          break;
28
29      case 3:
30          resposta.textContent = "três";
31          break;
32
33      default:
34          resposta.textContent = "outro número";
35      }
36      f.textContent = "fim do programa";
37  }
38  </script>
  
```



Desvio Condicional Seletivo

```

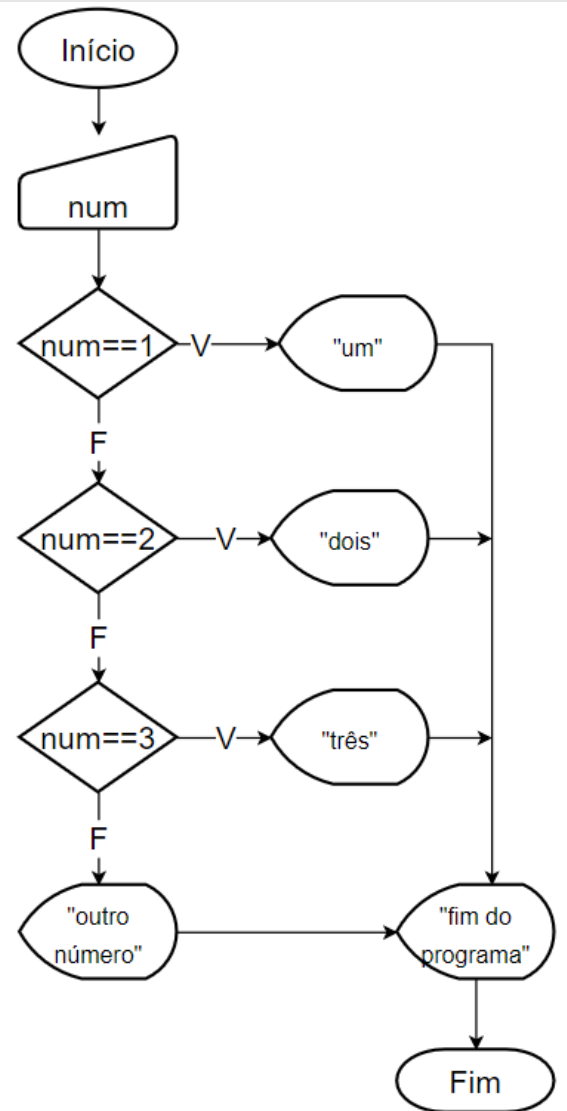
20  switch (num) {
21      case 1:
22          resposta.textContent = "um";
23          break;
24
25      case 2:
26          resposta.textContent = "dois";
27          break;
28
29      case 3:
30          resposta.textContent = "três";
31          break;
32
33      default:
34          resposta.textContent = "outro número";
35      }
36      f.textContent = "fim do programa";
37  }
38  </script>
  
```



Desvio Condicional Seletivo

```

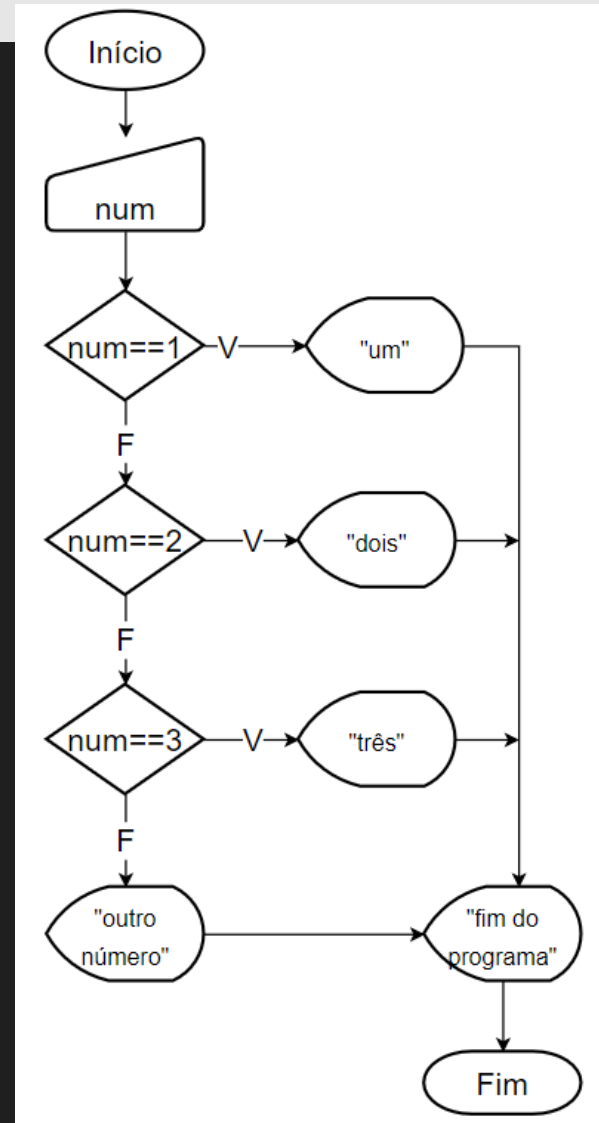
20  switch (num) {
21      case 1:
22          resposta.textContent = "um";
23          break;
24
25      case 2:
26          resposta.textContent = "dois";
27          break;
28
29      case 3:
30          resposta.textContent = "três";
31          break;
32
33      default:
34          resposta.textContent = "outro número";
35      }
36      f.textContent = "fim do programa";
37  }
38  </script>
  
```



Desvio Condicional Seletivo

```

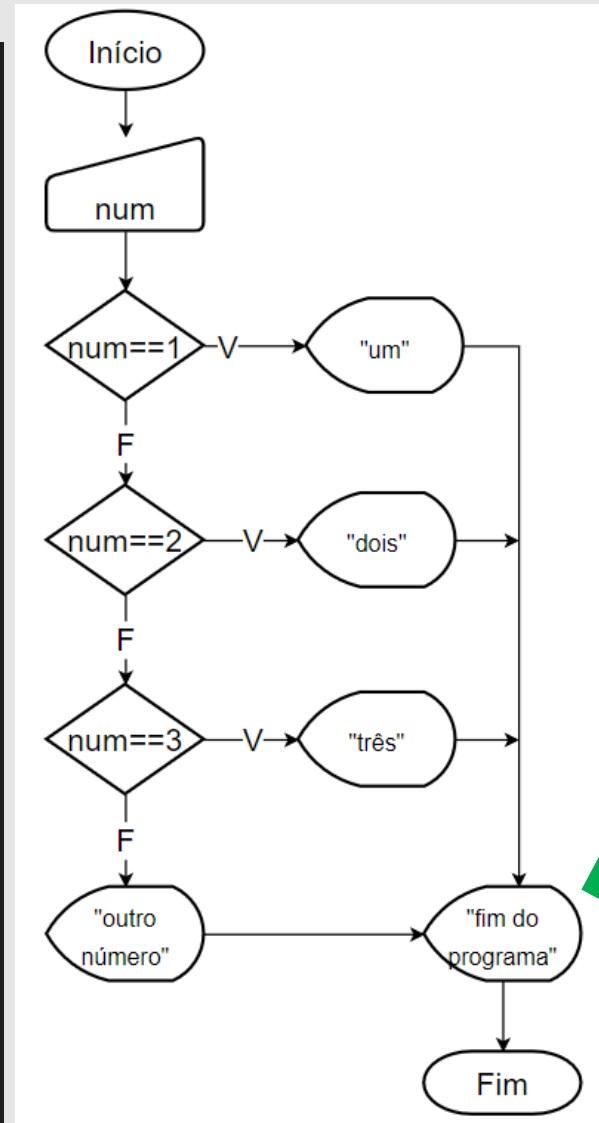
20  switch (num) {
21      case 1:
22          resposta.textContent = "um";
23          break;
24
25      case 2:
26          resposta.textContent = "dois";
27          break;
28
29      case 3:
30          resposta.textContent = "três";
31          break;
32
33      default:
34          resposta.textContent = "outro número";
35      }
36      f.textContent = "fim do programa";
37  }
38  </script>
  
```



Desvio Condicional Seletivo

```

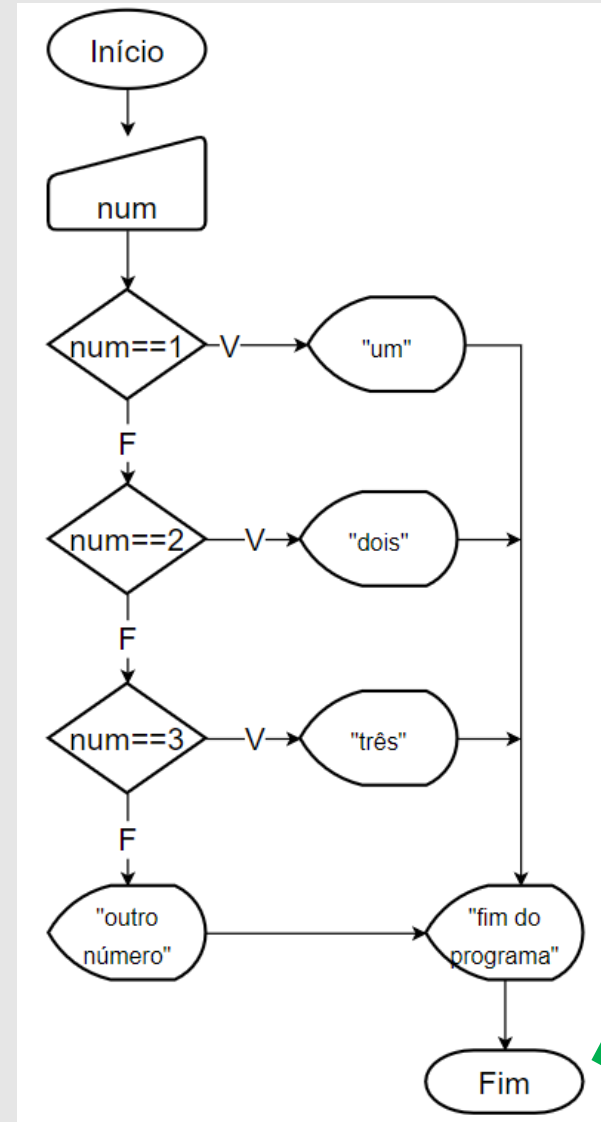
20  switch (num) {
21      case 1:
22          resposta.textContent = "um";
23          break;
24
25      case 2:
26          resposta.textContent = "dois";
27          break;
28
29      case 3:
30          resposta.textContent = "três";
31          break;
32
33      default:
34          resposta.textContent = "outro número";
35  }
36  f.textContent = "fim do programa";
37  }
38  </script>
  
```



Desvio Condicional Seletivo

```

20  switch (num) {
21      case 1:
22          resposta.textContent = "um";
23          break;
24
25      case 2:
26          resposta.textContent = "dois";
27          break;
28
29      case 3:
30          resposta.textContent = "três";
31          break;
32
33      default:
34          resposta.textContent = "outro número";
35      }
36      f.textContent = "fim do programa";
37  }
38  </script>
  
```



Operadores Lógicos

- ◆ Os operadores lógicos **&&** e **||** permitem mais de uma condição para a tomada de uma única decisão;
- ◆ Já o operador logico **!** tem por finalidade a negação do estado lógico de uma condição.

Operador Lógico	Descrição
&&	AND = E
 	OR = Ou
!	NOT = Não

Operadores Lógicos

◆ Tabela Verdade E

A	B	A && B
V	V	V
V	F	F
F	V	F
F	V	F

◆ Tabela Verdade OU

A	B	A B
V	V	V
V	F	V
F	V	V
F	V	F

◆ Tabela Verdade NOT

A	!A
V	F
F	V

Divisibilidade

- ◆ Quando o resto de uma divisão de números naturais é igual a zero, tem-se **divisibilidade**, ou seja, resultado de divisão exata;
- ◆ A linguagem **Javascript** possui como operador aritmético para auxiliar o cálculo de divisibilidade o símbolo **%** (porcentagem);
- ◆ Emprega-se esse operador aritmético quando se necessita calcular o valor do resto de uma divisão de valores inteiros.
- ◆ Exemplo:

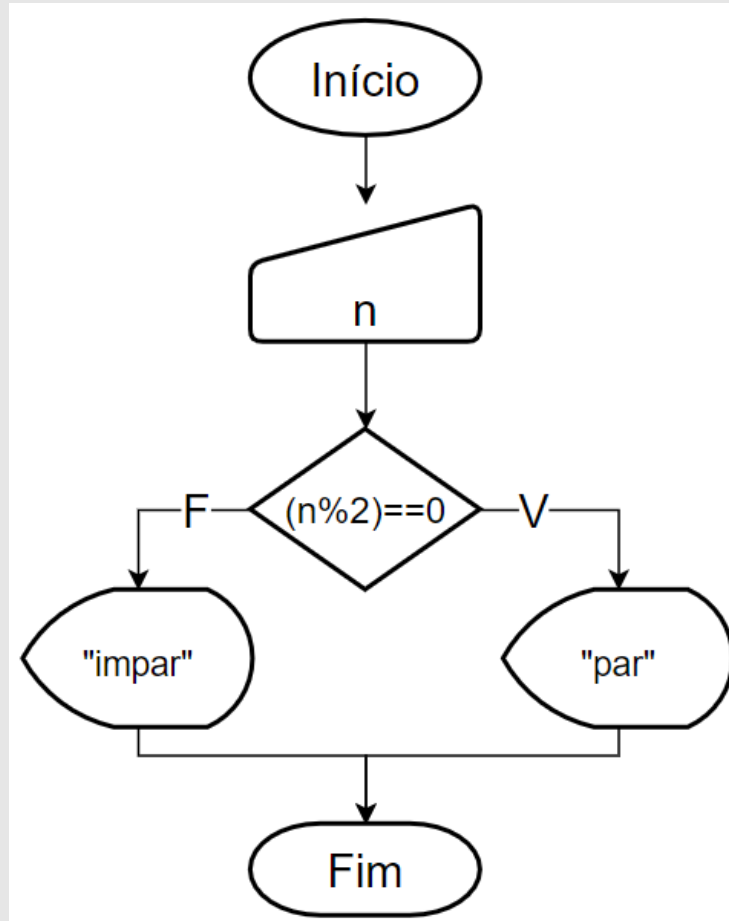
$X = 4\%2;$ $//$ x é zero

$Y = 5\%3;$ $//$ y é 2

$z = 6\%3;$ $//$ z é zero

Divisibilidade

- Programa escreve na tela se o número digitado pelo usuário é par ou ímpar.



Divisibilidade

- Programa escreve na tela se o número digitado pelo usuário é par ou ímpar.

```

1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <!------->
4
5  <head>
6    <meta charset="UTF-8">
7    <meta http-equiv="X-UA-Compatible" content="IE=edge">
8    <meta name="viewport" content="width=device-width, initial-scale=1.0">
9
10   <script>
11     function parOuImpar() {
12
13       let n = document.getElementById("n").value;
14       let f = document.getElementById("f");
15       let resposta = document.getElementById("resposta");
16
17       n = Number(n);
18
19       if (n % 2 == 0) {
20         resposta.textContent = "par";
21       }
22       else {
23         resposta.textContent = "ímpar";
24       }
25       f.textContent = "fim do programa";
26     }
27   </script>

```

Divisibilidade

- Programa escreve na tela se o número digitado pelo usuário é par ou ímpar.

```

28  <title>Switch em Javascript</title>
29  </head>
30  <!------->
31
32  <body>
33    <form>
34      <p><label>Entre com um valor numérico : <input type="text" name="n" id="n"></label></p>
35
36      <input type="button" value="Submit" onclick="parOuImpar()">
37    </form>
38    <p id="resposta"></p>
39    <p id="f"></p>
40  </body>
41  <!------->
42
43  </html>

```

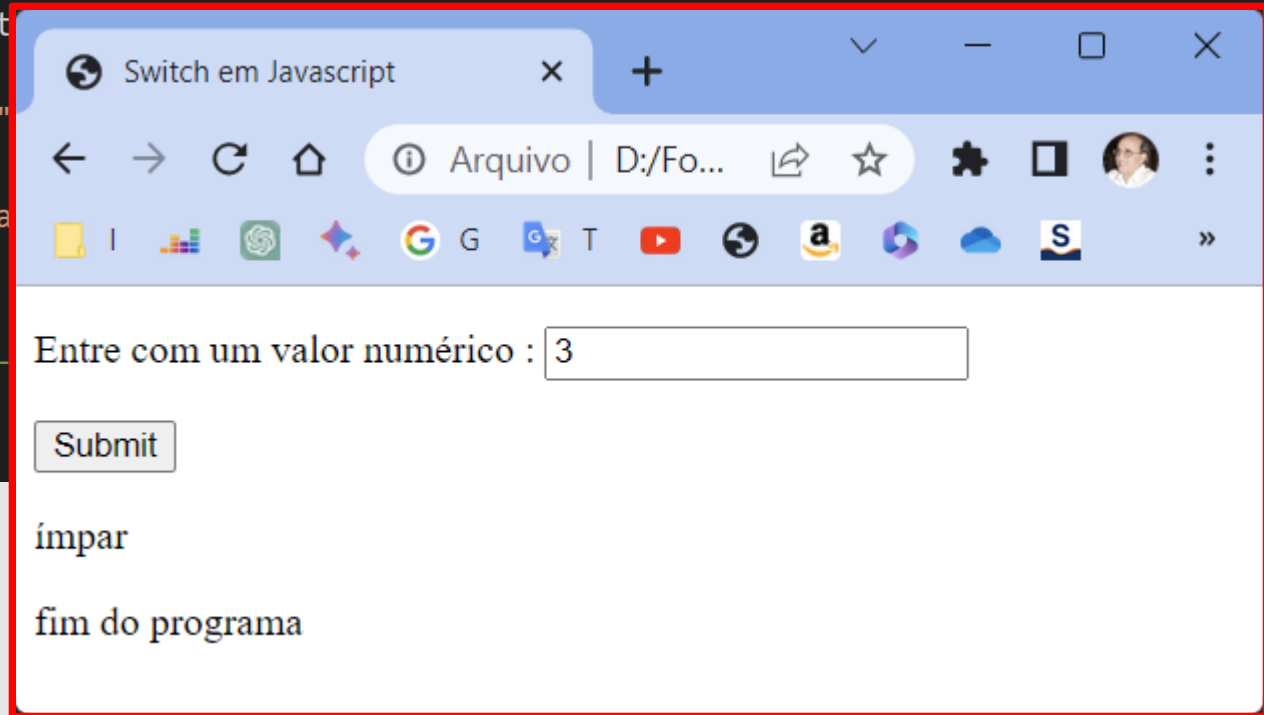
Divisibilidade

- Programa escreve na tela se o número digitado pelo usuário é par ou ímpar.

```

28 <title>Switch em Javascript</title>
29 </head>
30 <!------->
31
32 <body>
33   <form>
34     <p><label>Ent
35
36     <input type="
37   </form>
38   <p id="resposta
39   <p id="f"></p>
40 </body>
41 <!-------
42
43 </html>

```



Switch em Javascript

Arquivo | D:/Fo...

Entre com um valor numérico : 3

Submit

ímpar

fim do programa

Exercícios

1. Implementar, com a Linguagem Javascript, um programa que solicita dois números ao usuário e exiba “igual” caso sejam iguais ou mostre qual o maior.
2. Implementar, com a Linguagem Javascript, um programa que receba três inteiros e mostre qual deles é o maior e qual o menor.
3. Implementar, com a Linguagem Javascript, um programa que recebe um inteiro e exiba na tela se ele é múltiplo de 7 ou não.
4. Para doar sangue é necessário ter entre 18 e 67 anos. Implementar com a Linguagem Javascript um programa que pergunte a idade de uma pessoa e diga se ela pode doar sangue ou não. Utilize alguns dos operadores lógicos **OU** (||) e **E** (&&).

Programação com Laços – Objetivos

- ◆ Apresentar detalhes sobre o uso de **laços** para repetição de trechos de programas;
- ◆ São apresentadas duas formas populares de **laços**:
condicional pré-teste e condicional pós-teste.

Laços e Detalhes Operacionais

- ◆ Programas (obtidos a partir de algoritmos) são sequências de instruções que impõem certa ordem de execução a um computador;
- ◆ Por vezes algumas dessas ordens devem ser repetidas , e para fazer esse tipo de ação, pode-se empregar a técnica de laços de repetição.

Laços e Detalhes Operacionais

- Laços empregados com a **Linguagem Javascript**:
 - ✱ Ação condicional pré-teste (com a condição de controle no início do laço);
 - ✱ Ação condicional pós-teste (com a condição de controle no fim do laço).

Laço Condicional Pré-Teste

- O laço condicional pré-teste na **linguagem Javascript** é implementado por meio da instrução **while**;
- A instrução **while** permite se executar um conjunto de instruções enquanto a condição verificada permanecer válida;
- No momento em que essa condição **não** for válida, o processamento da rotina será desviado para **fora** do laço.

Instrução while – Sintaxe

```
while <(condição)>
{
    <instrução 1 executada enquanto a condição for verdadeira>
    <instrução 2 executada enquanto a condição for verdadeira>
    <instrução 3 executada enquanto a condição for verdadeira>
    <instrução N executada enquanto a condição for verdadeira>
}
```

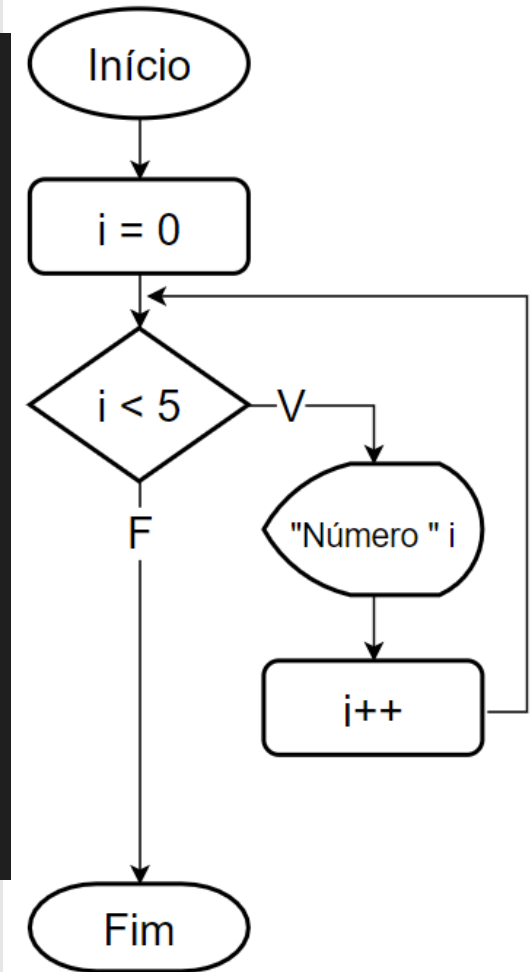
Instrução while – Exemplo

- Exibir na página HTML os números 0, 1, 2, 3 e 4.

```

10  <script>
11      function exibeNumeros() {
12          let i = 0;
13
14          while (i < 5) {
15              document.write("<p>" + "Número " + i + "<p>");
16              i++;
17          }
18
19      }
20
21      window.onload = exibeNumeros;
22  </script>

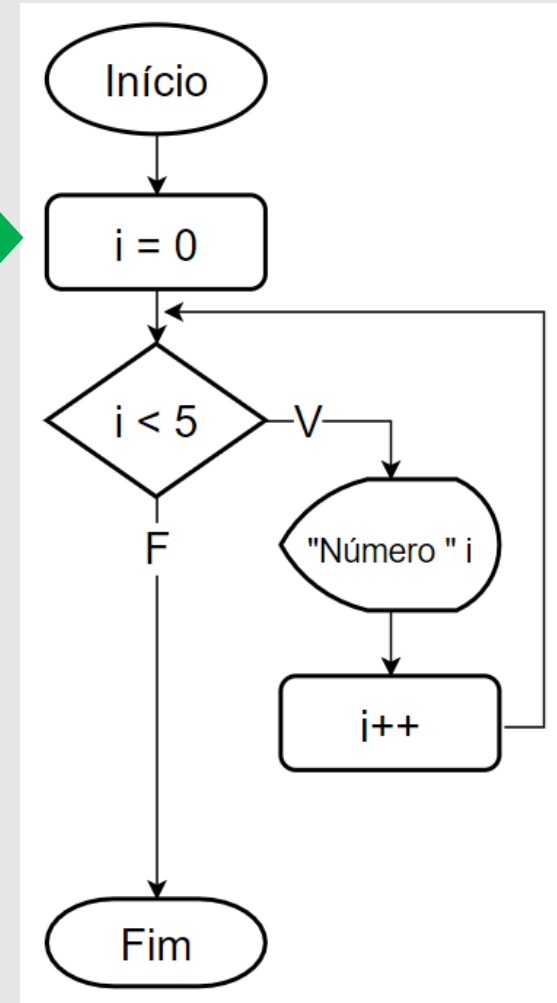
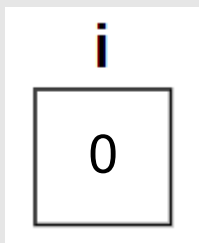
```



Instrução while – Exemplo

```

10  <script>
11      function exibeNumeros() {
12
13  →  let i = 0;
14
15      while (i < 5) {
16          document.write("<p>" + "Número " + i + "<p>");
17          i++;
18      }
19
20  }
21      window.onload = exibeNumeros;
22  </script>
  
```

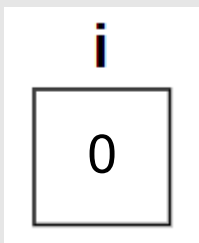
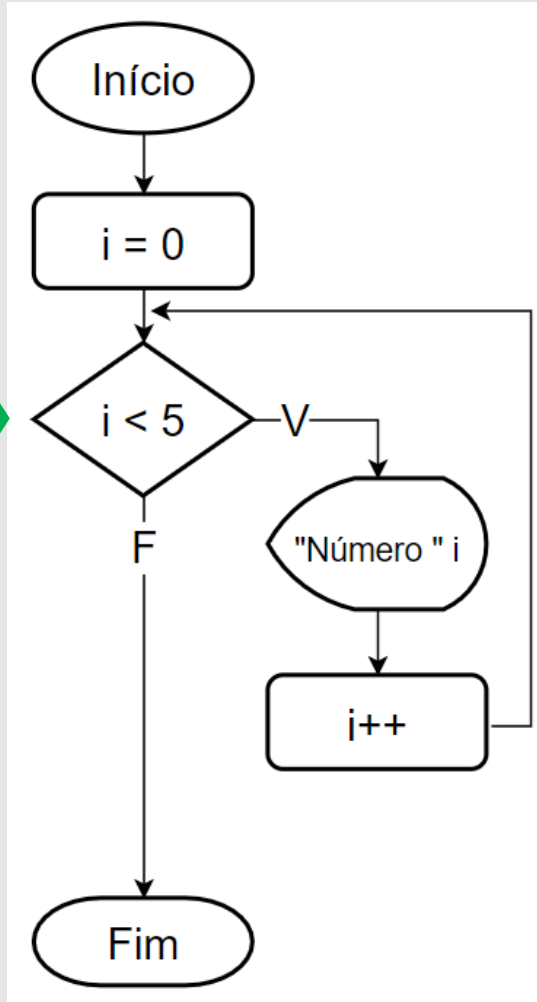


Instrução while – Exemplo

```

10  <script>
11      function exibeNumeros() {
12
13          let i = 0;
14
15          while (i < 5) {
16              document.write("<p>" + "Número " + i + "<p>");
17              i++;
18          }
19
20      }
21      window.onload = exibeNumeros;
22  </script>

```

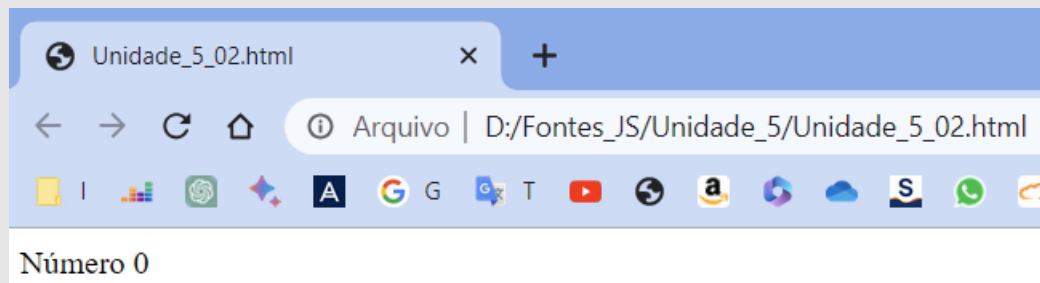
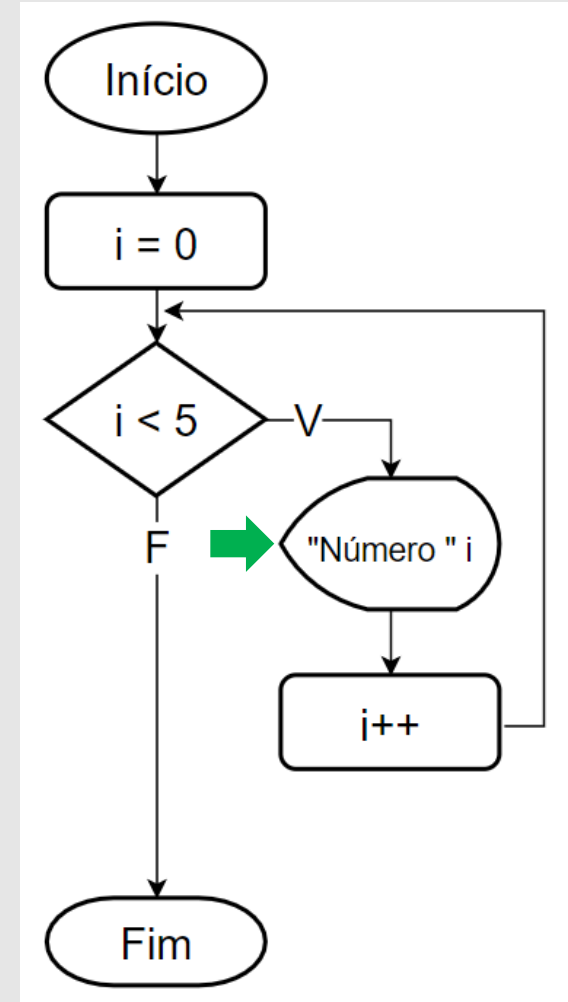


Instrução while – Exemplo

```

10 <script>
11   function exibeNumeros() {
12
13     let i = 0;
14
15     while (i < 5) {
16       document.write("<p>" + "Número " + i + "<p>");
17       i++;
18     }
19
20   }
21   window.onload = exibeNumeros;
22 </script>

```



i

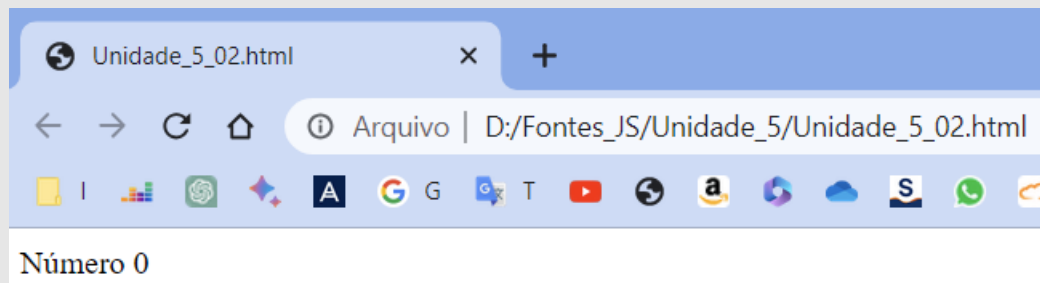
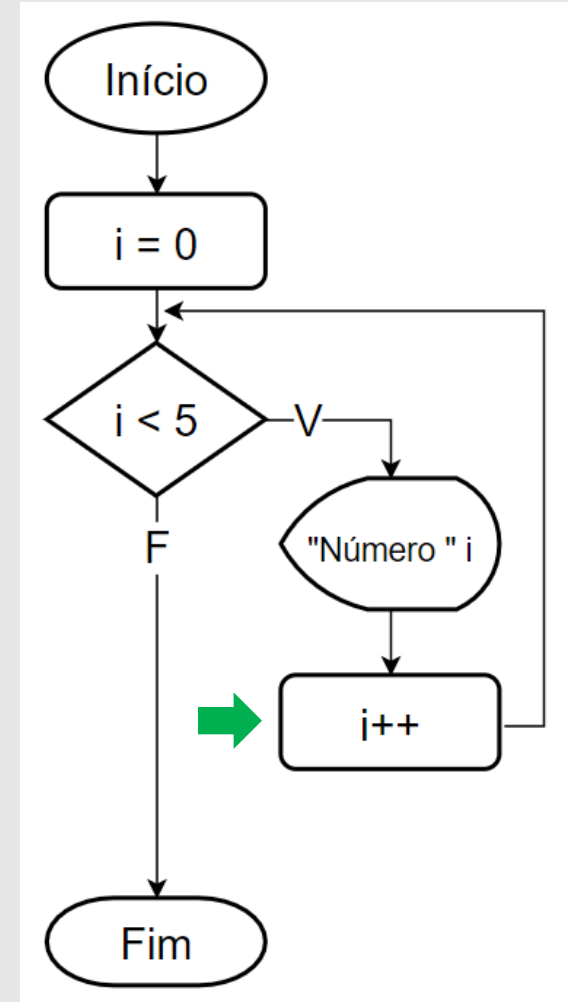
0

Instrução while – Exemplo

```

10 <script>
11   function exibeNumeros() {
12
13     let i = 0;
14
15     while (i < 5) {
16       document.write("<p>" + "Número " + i + "<p>");
17       i++;
18     }
19
20   }
21   window.onload = exibeNumeros;
22 </script>

```

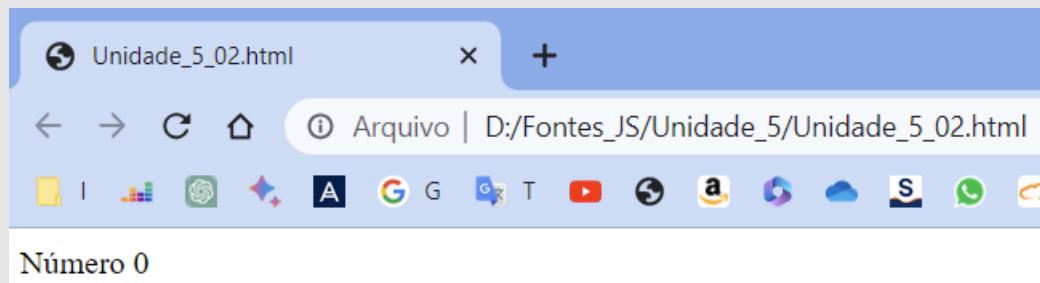
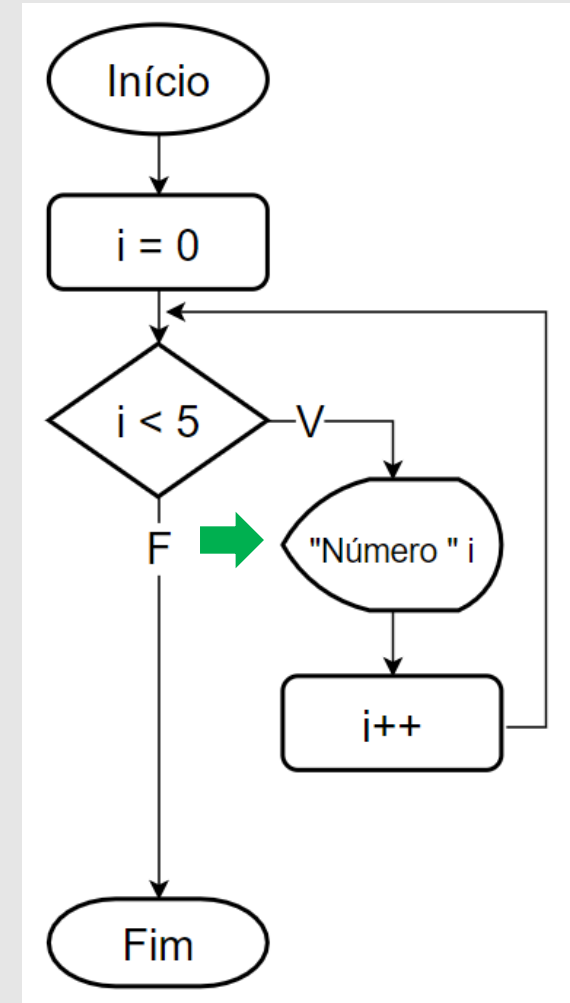


Instrução while – Exemplo

```

10  <script>
11      function exibeNumeros() {
12
13          let i = 0;
14
15      while (i < 5) {
16          document.write("<p>" + "Número " + i + "<p>");
17          i++;
18      }
19
20  }
21      window.onload = exibeNumeros;
22  </script>

```

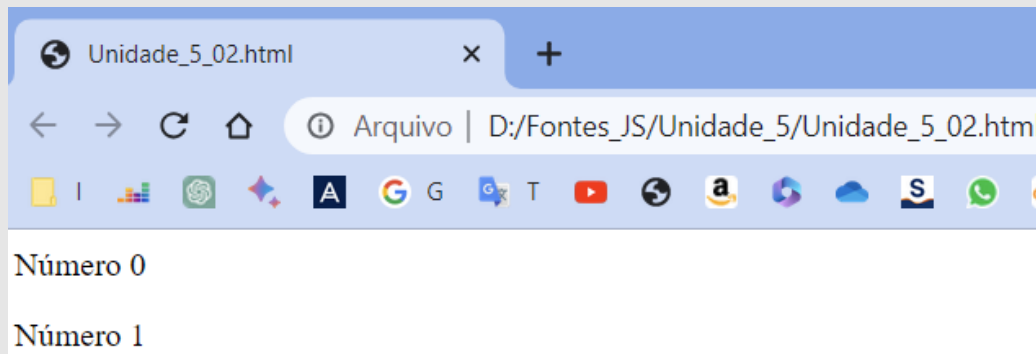
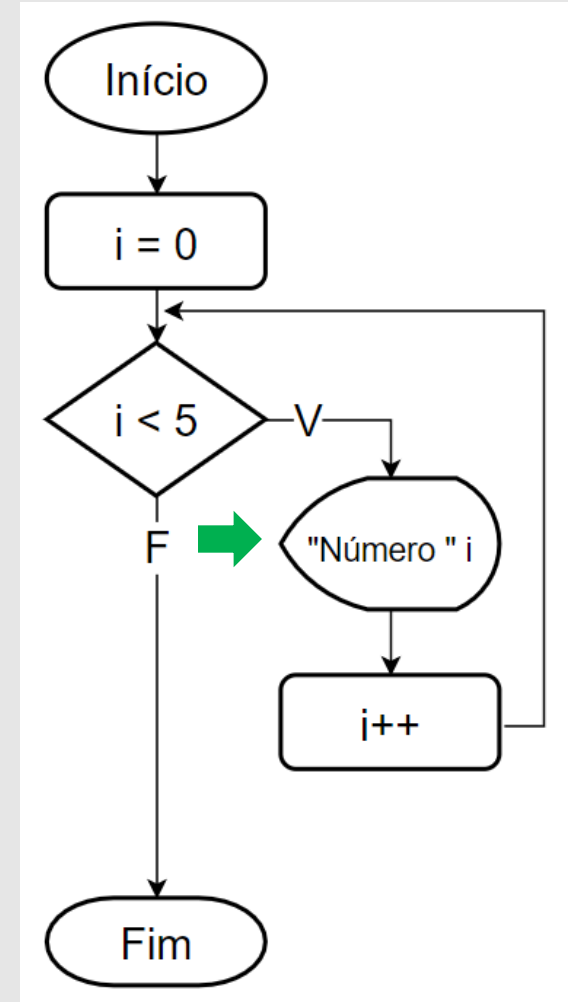


Instrução while – Exemplo

```

10 <script>
11   function exibeNumeros() {
12
13     let i = 0;
14
15     while (i < 5) {
16       document.write("<p>" + "Número " + i + "<p>");
17       i++;
18     }
19
20   }
21   window.onload = exibeNumeros;
22 </script>

```



i

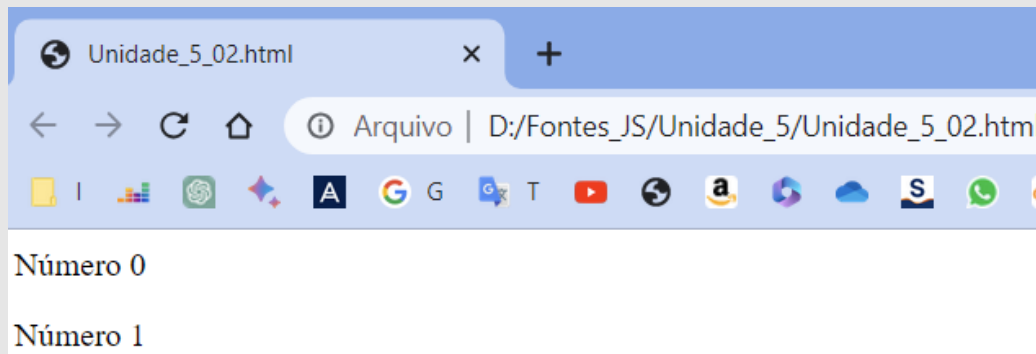
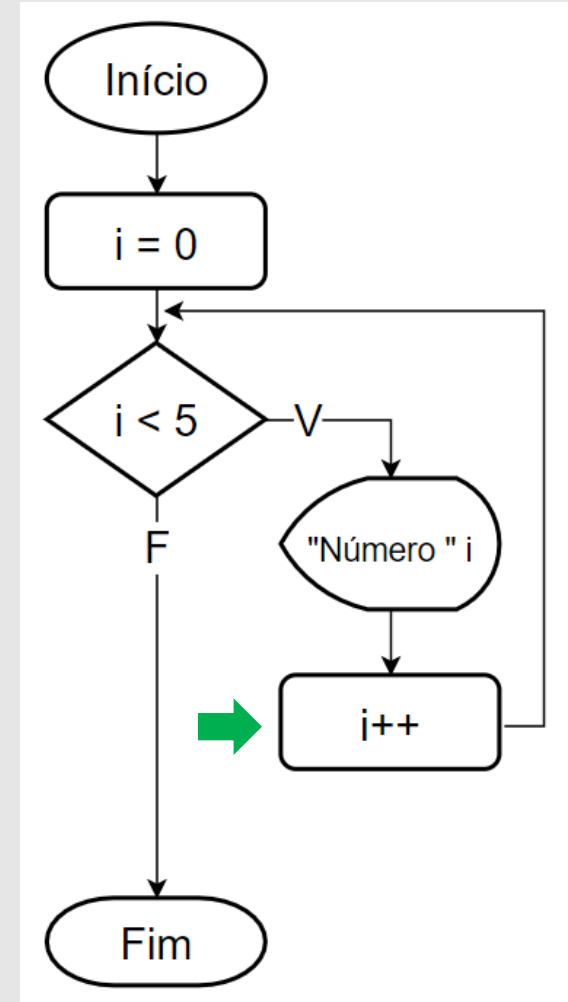
1

Instrução while – Exemplo

```

10 <script>
11   function exibeNumeros() {
12
13     let i = 0;
14
15     while (i < 5) {
16       document.write("<p>" + "Número " + i + "<p>");
17       i++;
18     }
19
20   }
21   window.onload = exibeNumeros;
22 </script>

```



i

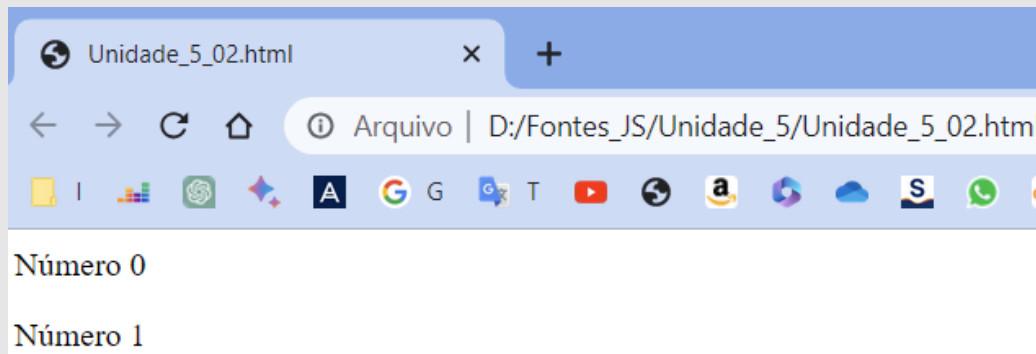
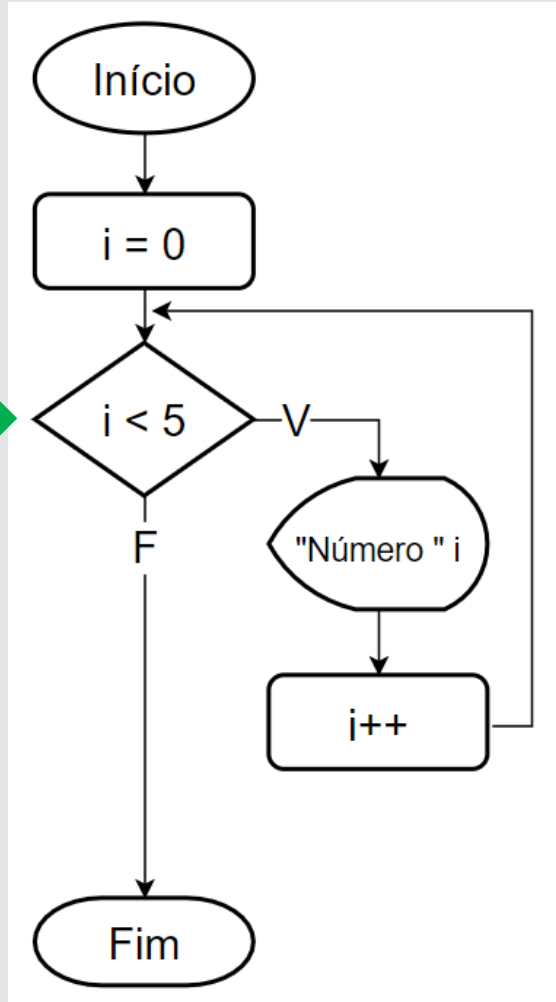
2

Instrução while – Exemplo

```

10 <script>
11   function exibeNumeros() {
12
13     let i = 0;
14
15     while (i < 5) {
16       document.write("<p>" + "Número " + i + "<p>");
17       i++;
18     }
19
20   }
21   window.onload = exibeNumeros;
22 </script>

```



i

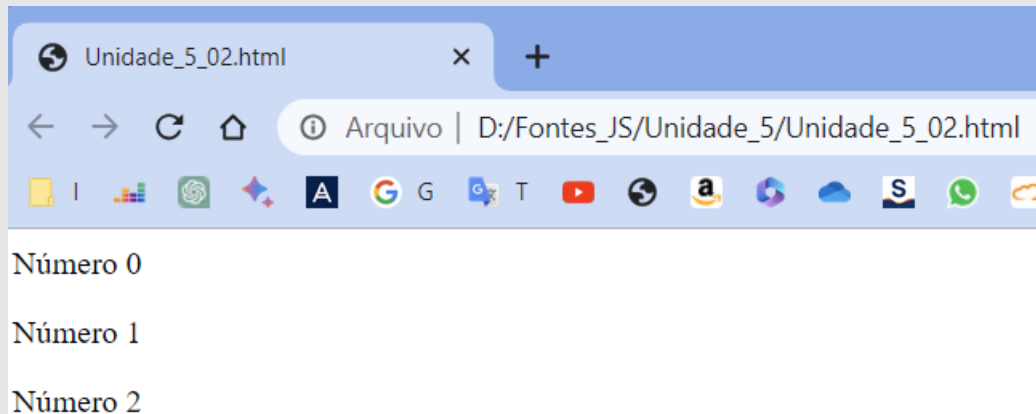
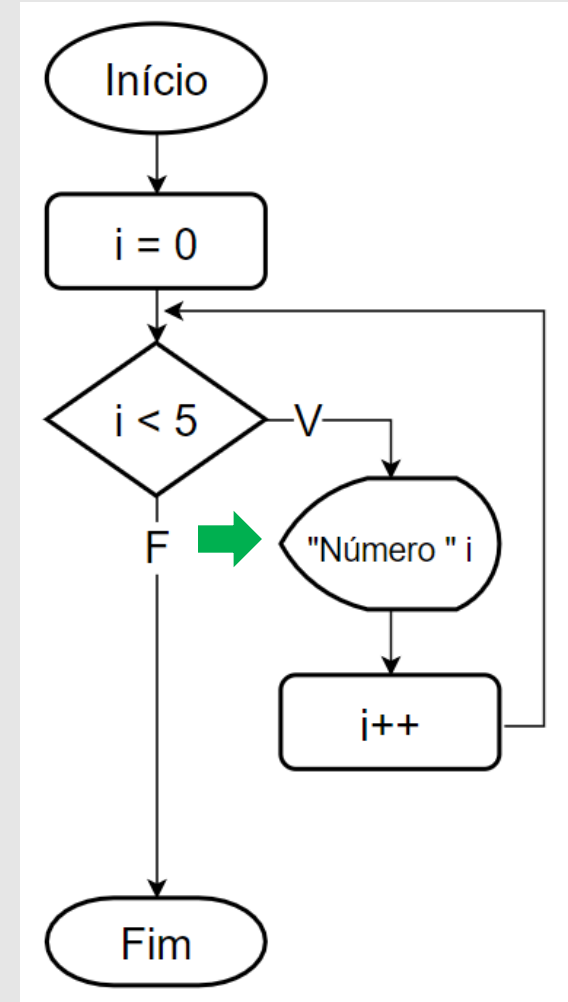
2

Instrução while – Exemplo

```

10 <script>
11   function exibeNumeros() {
12
13     let i = 0;
14
15     while (i < 5) {
16       document.write("<p>" + "Número " + i + "<p>");
17       i++;
18     }
19
20   }
21   window.onload = exibeNumeros;
22 </script>

```



i

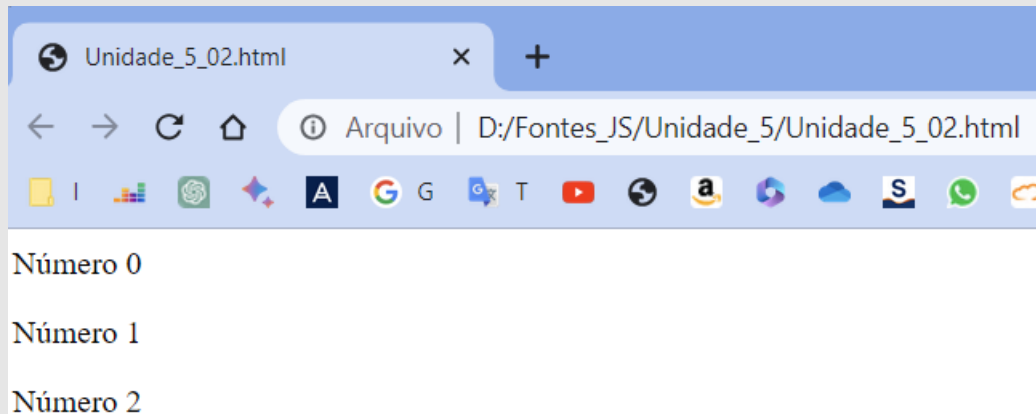
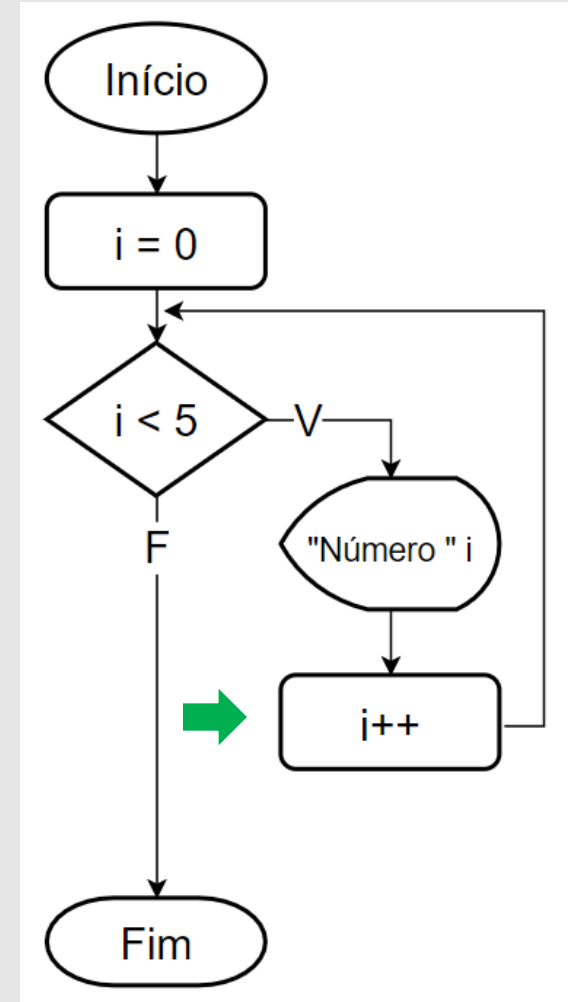
2

Instrução while – Exemplo

```

10 <script>
11   function exibeNumeros() {
12
13     let i = 0;
14
15     while (i < 5) {
16       document.write("<p>" + "Número " + i + "<p>");
17       i++;
18     }
19
20   }
21   window.onload = exibeNumeros;
22 </script>

```



i

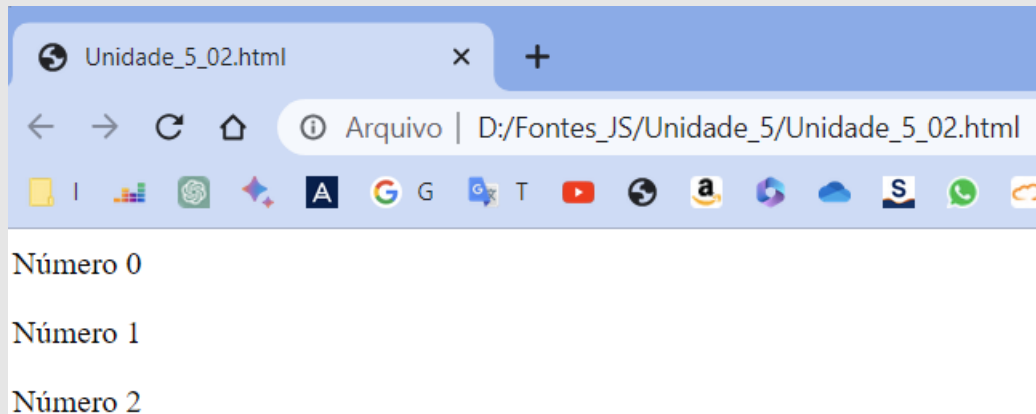
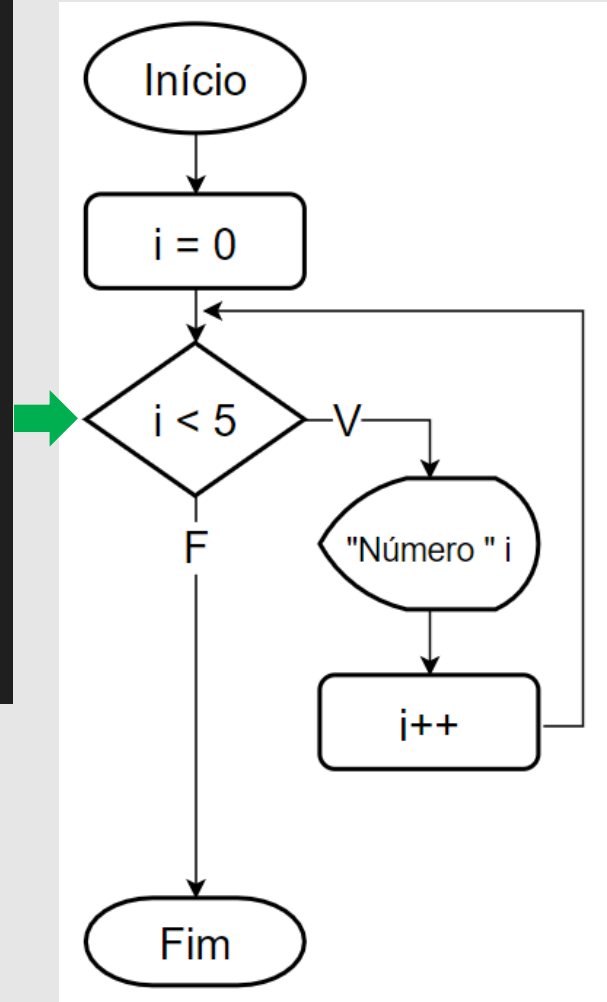
3

Instrução while – Exemplo

```

10 <script>
11   function exibeNumeros() {
12
13     let i = 0;
14
15     while (i < 5) {
16       document.write("<p>" + "Número " + i + "<p>");
17       i++;
18     }
19
20   }
21   window.onload = exibeNumeros;
22 </script>

```



i

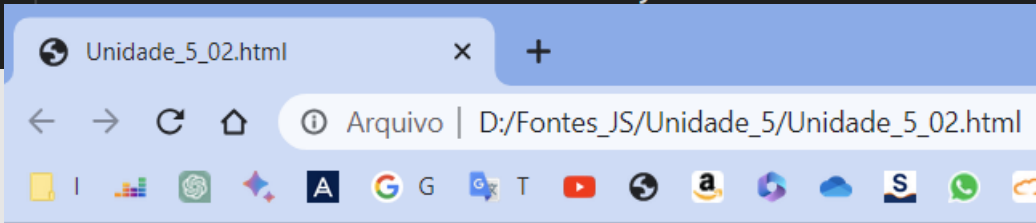
3

Instrução while – Exemplo

```

10 <script>
11   function exibeNumeros() {
12
13     let i = 0;
14
15     while (i < 5) {
16       document.write("<p>" + "Número " + i + "<p>");
17       i++;
18     }
19
20   }
21   window.onload = exibeNumeros;
22

```



Número 0

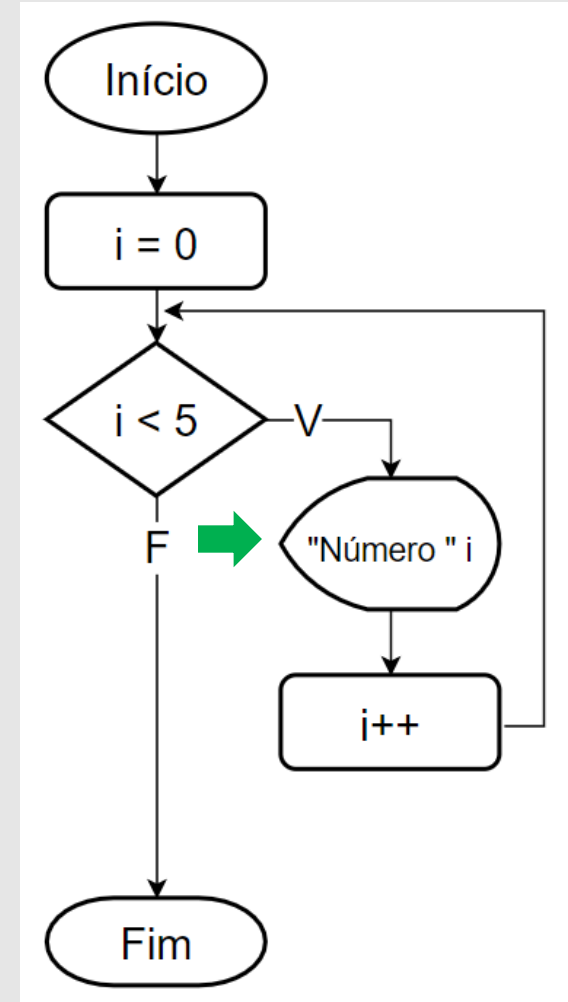
Número 1

Número 2

Número 3

i

3

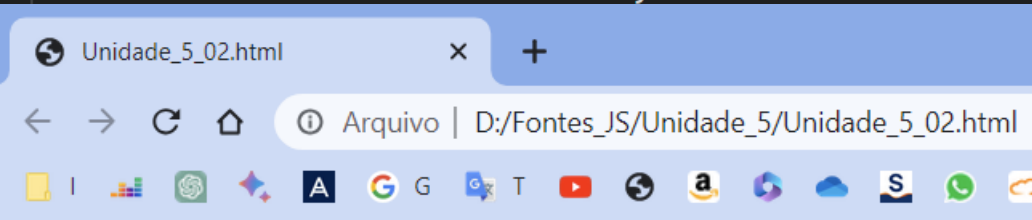
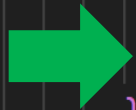


Instrução while – Exemplo

```

10 <script>
11   function exibeNumeros() {
12
13     let i = 0;
14
15     while (i < 5) {
16       document.write("<p>" + "Número " + i + "<p>");
17       i++;
18     }
19
20   }
21   window.onload = exibeNumeros;
22

```



Número 0

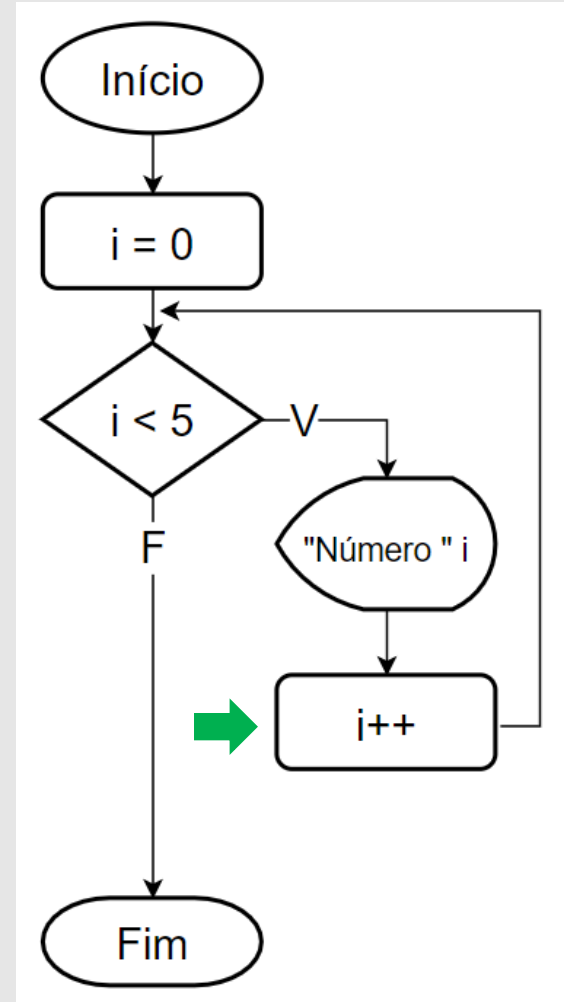
Número 1

Número 2

Número 3

i

4

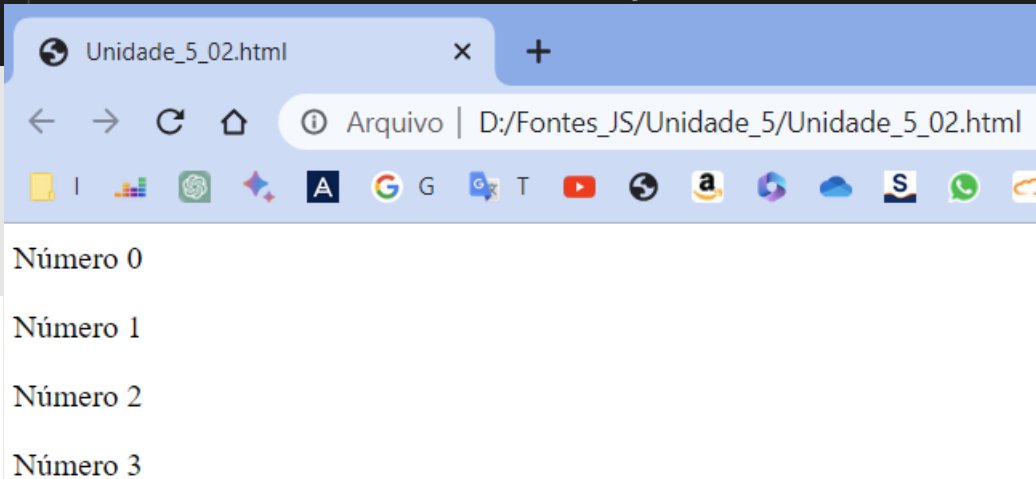
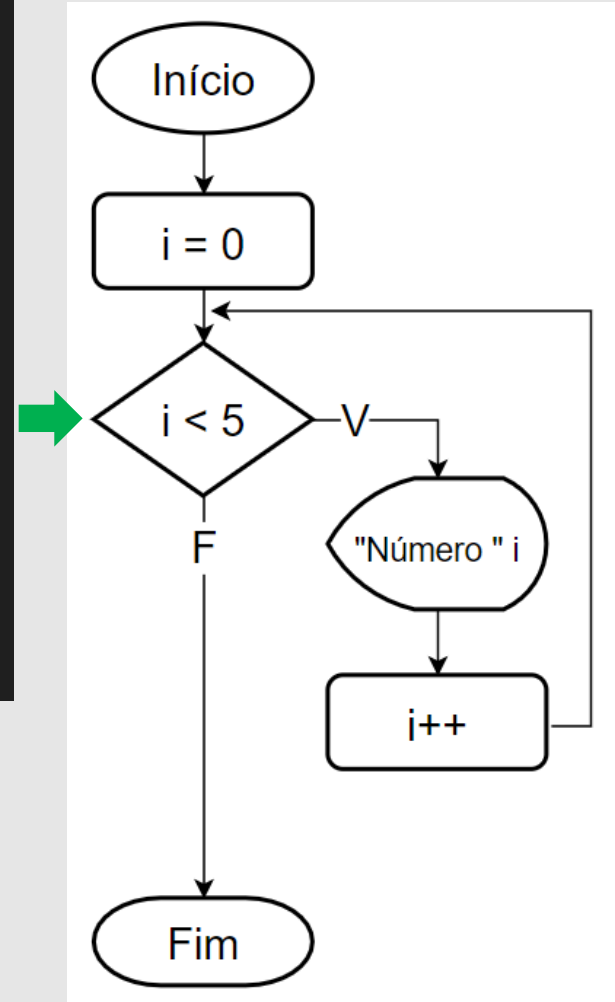


Instrução while – Exemplo

```

10 <script>
11   function exibeNumeros() {
12
13     let i = 0;
14
15     while (i < 5) {
16       document.write("<p>" + "Número " + i + "<p>");
17       i++;
18     }
19
20   }
21   window.onload = exibeNumeros;
22

```

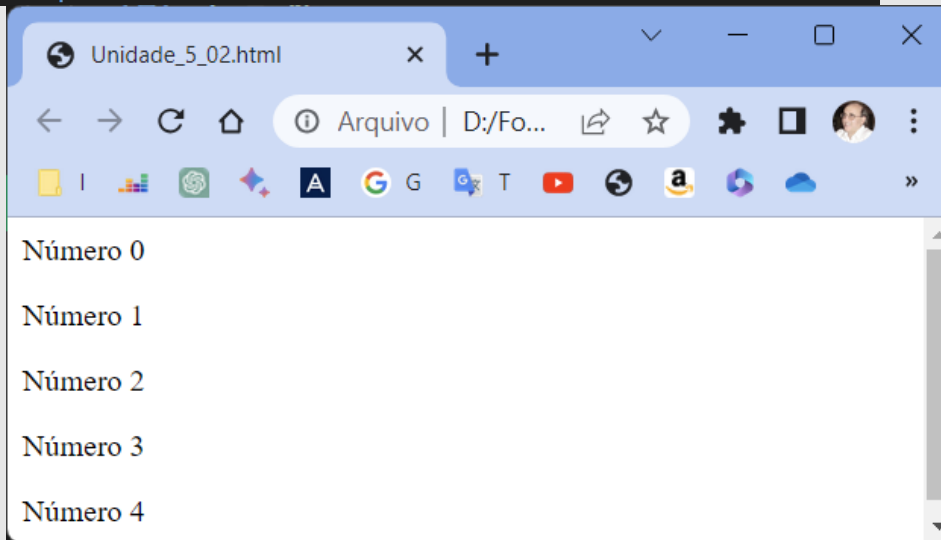
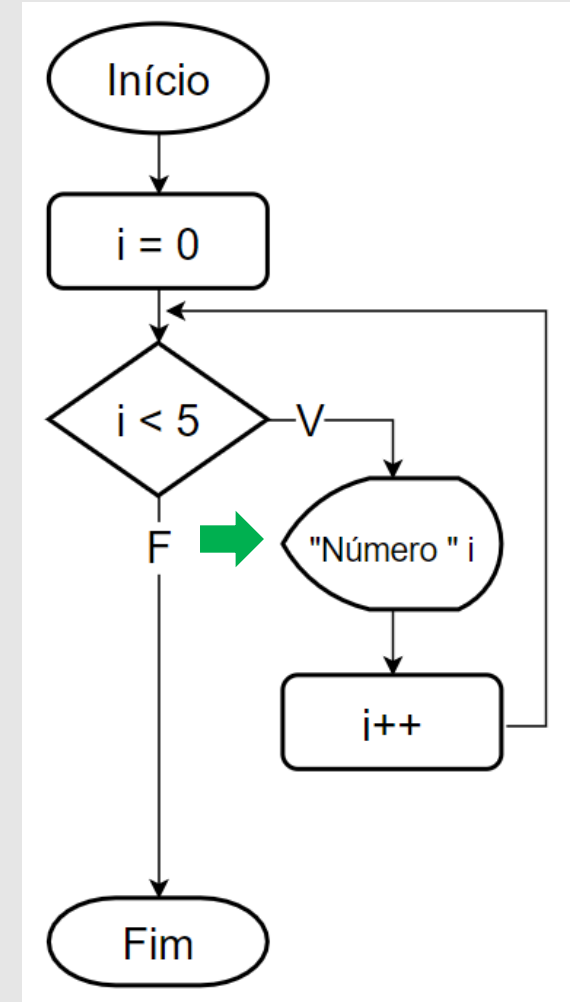


Instrução while – Exemplo

```

10  <script>
11    function exibeNumeros() {
12
13      let i = 0;
14
15      while (i < 5) {
16        document.write("<p>" + "Número " + i + "<p>");
17        i++;
18      }
19
20    }
21    window.onload = exibeNumeros;
22  </script>

```



i

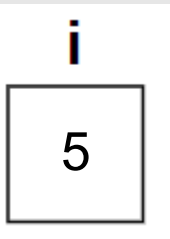
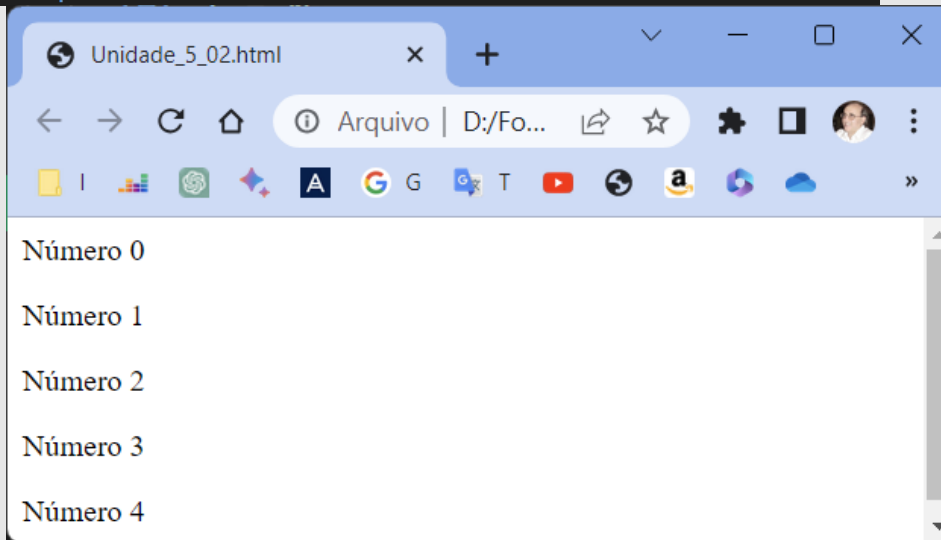
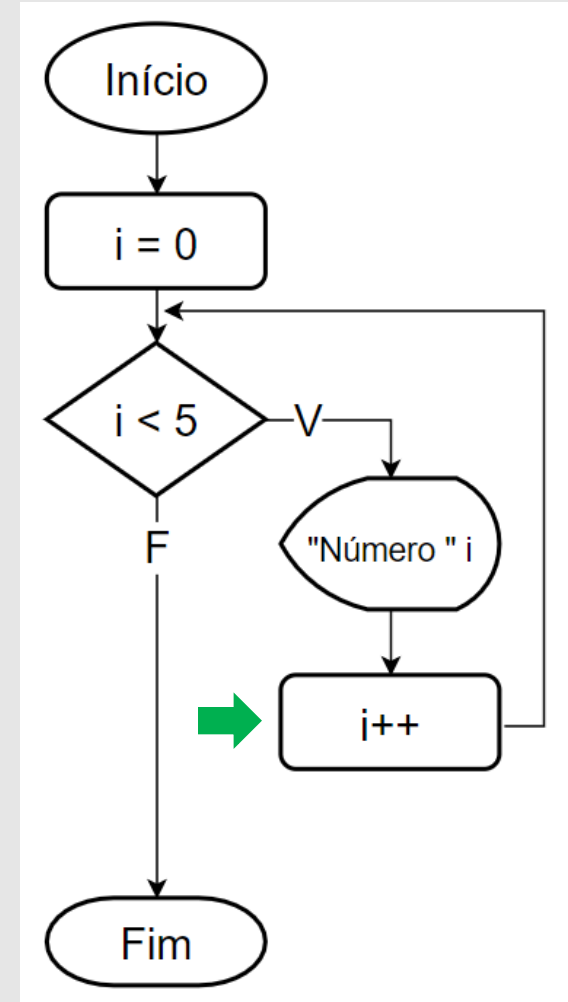
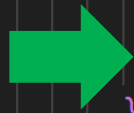
4

Instrução while – Exemplo

```

10  <script>
11    function exibeNumeros() {
12
13      let i = 0;
14
15      while (i < 5) {
16        document.write("<p>" + "Número " + i + "<p>");
17        i++;
18      }
19
20    }
21    window.onload = exibeNumeros;
22  </script>

```

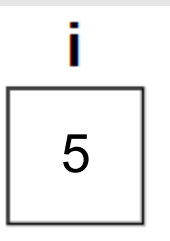
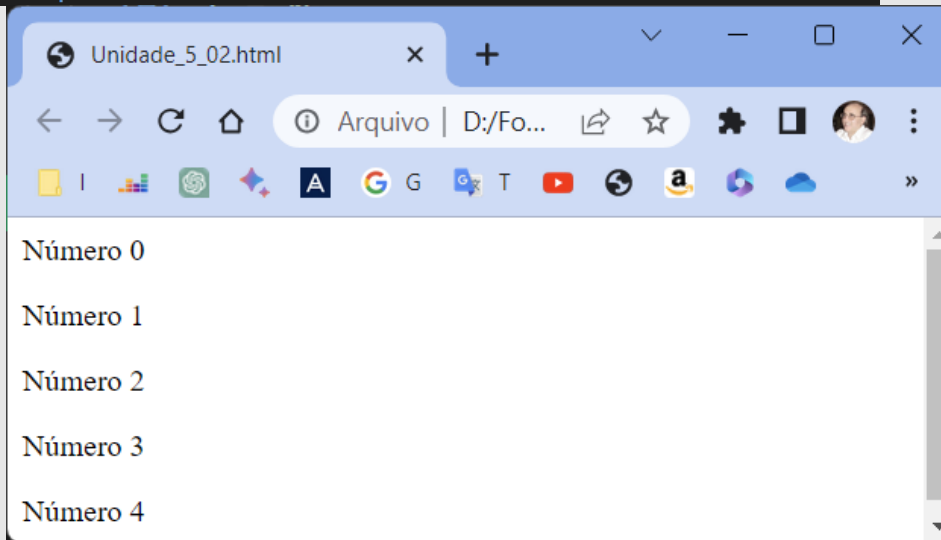
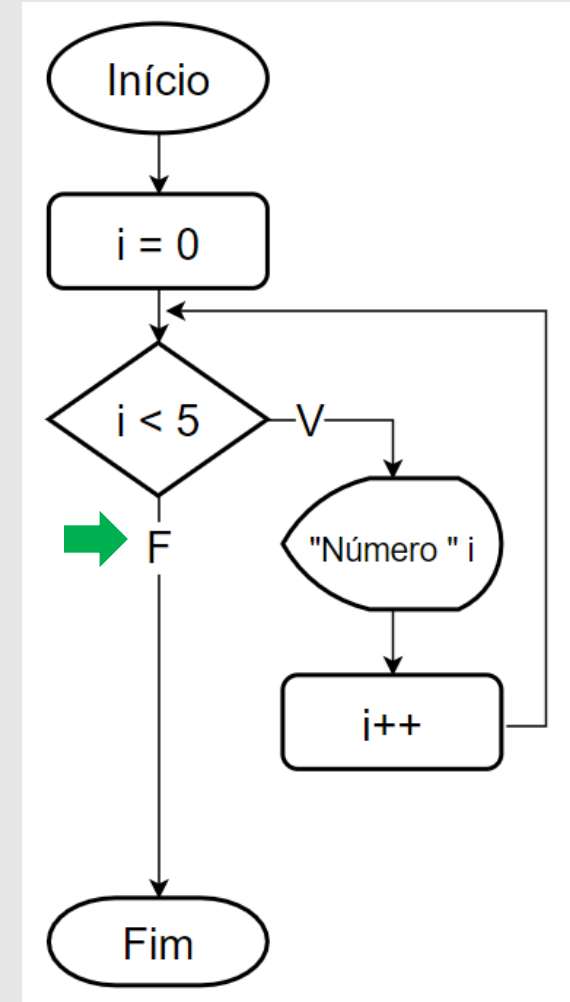


Instrução while – Exemplo

```

10  <script>
11    function exibeNumeros() {
12
13      let i = 0;
14
15      while (i < 5) {
16        document.write("<p>" + "Número " + i + "<p>");
17        i++;
18      }
19
20    }
21    window.onload = exibeNumeros;
22  </script>

```

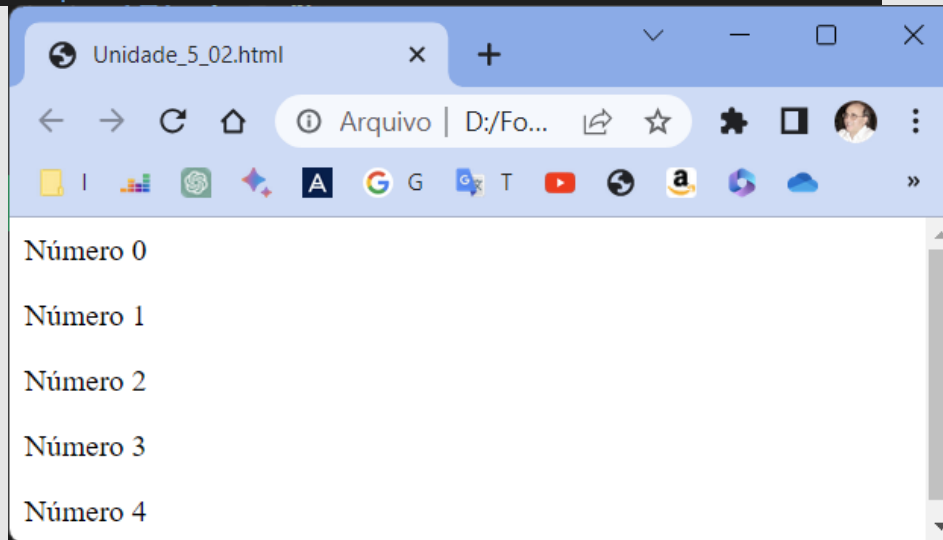
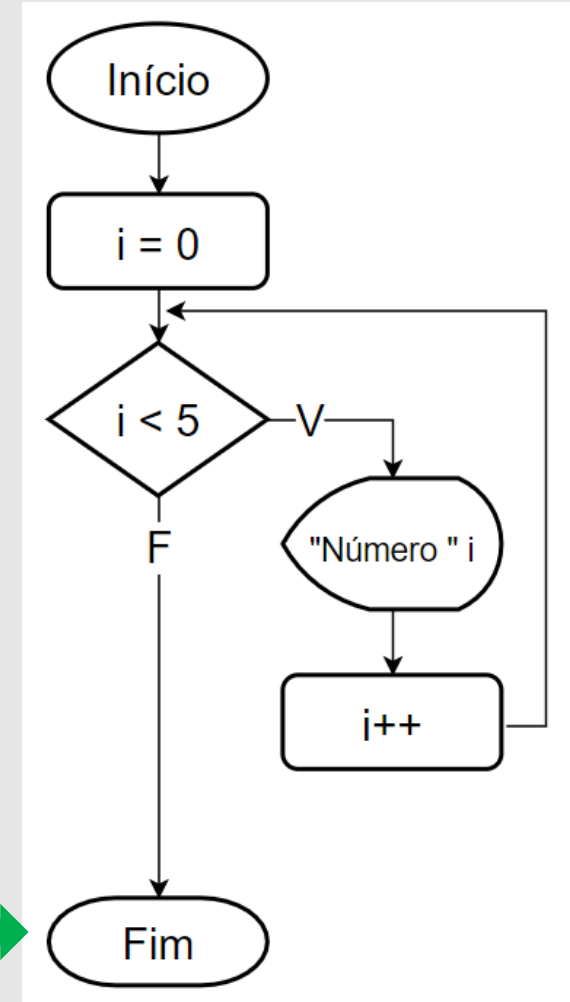


Instrução while – Exemplo

```

10  <script>
11    function exibeNumeros() {
12
13      let i = 0;
14
15      while (i < 5) {
16        document.write("<p>" + "Número " + i + "<p>");
17        i++;
18      }
19
20  }
21  window.onload = exibeNumeros;
22  </script>

```



i

5

Laço Condicional Pós-Teste

- Para a execução de laço condicional pós-teste a Linguagem C# oferece a instrução **do ... while**, que faz um teste lógico no final do laço, verificando se é possível executar novamente o trecho de instruções subordinadas a ele;
- Esse laço tem como característica operacional sempre executar o bloco subordinado ao laço no mínimo uma vez;
- **do ... while** executa o bloco de instruções subordinadas ao laço pelo menos uma vez antes de verificar a validade da condição estabelecida, diferentemente do laço **while**, que executa o bloco somente quando a condição for favorável.

Laço Condicional Pós-Teste

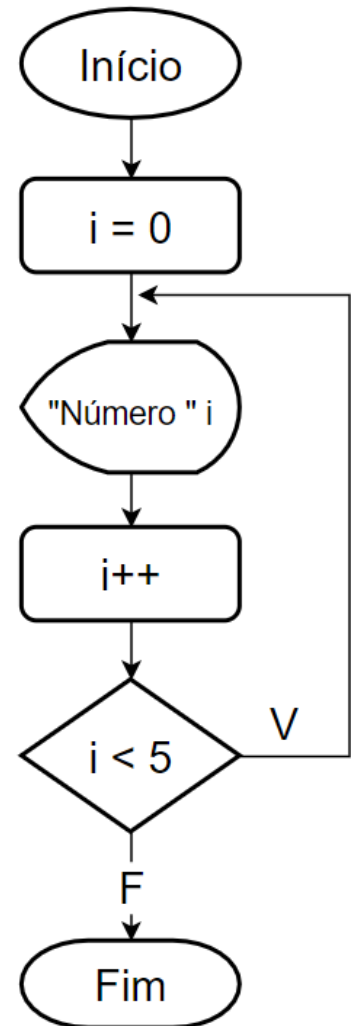
- 🔗 Dessa forma, **do ... while** sempre processa o bloco pelo menos uma vez, mesmo que a condição não seja válida.
- 🔗 Sintaxe:

```
do
{
    ...
    block of statements;
    ...
} while( condition ) ;
```


do ... while – Exemplo

Exibir na página HTML os números: 0, 1, 2, 3 e 4.

```
10  <script>
11      function exibeNumeros() {
12
13          let i = 0;
14
15          do {
16              document.write("<p>" + "Número " + i + "<p>");
17              i++;
18          } while (i < 5)
19
20      }
21      window.onload = exibeNumeros;
22  </script>
```

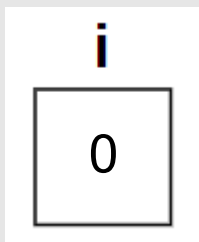
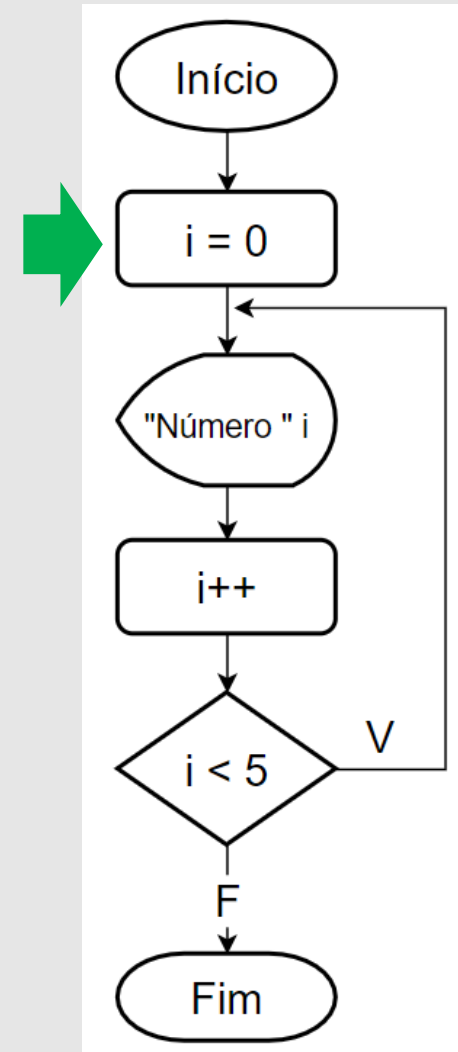


do ... while - Exemplo

```

10  <script>
11      function exibeNumeros() {
12
13  →  let i = 0;
14
15      do {
16          document.write("<p>" + "Número " + i + "<p>");
17          i++;
18      } while (i < 5)
19
20  }
21      window.onload = exibeNumeros;
22  </script>

```

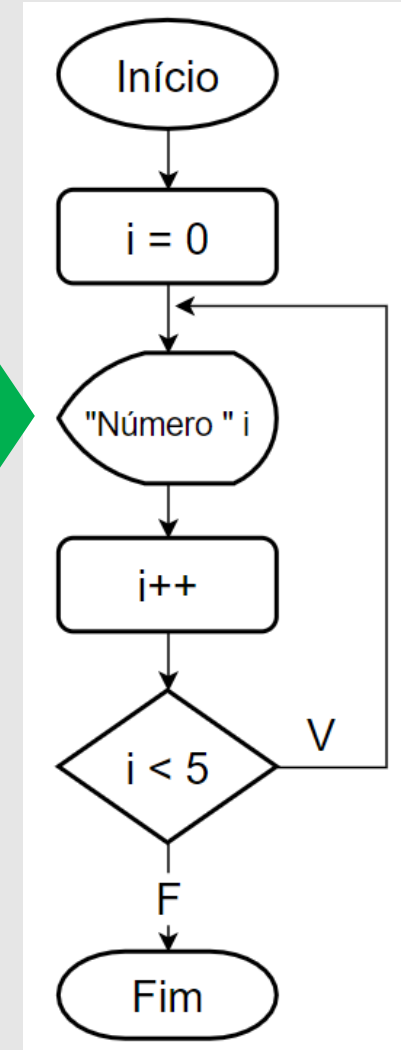


do ... while – Exemplo

```

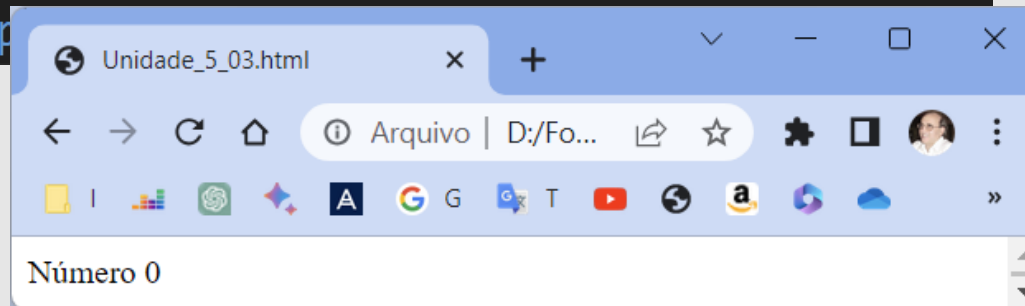
10  <script>
11      function exibeNumeros() {
12
13          let i = 0;
14
15          do {
16              document.write("<p>" + "Número " + i + "<p>");
17              i++;
18          } while (i < 5)
19
20      }
21      window.onload = exibeNumeros;
22  </script>

```



i

0

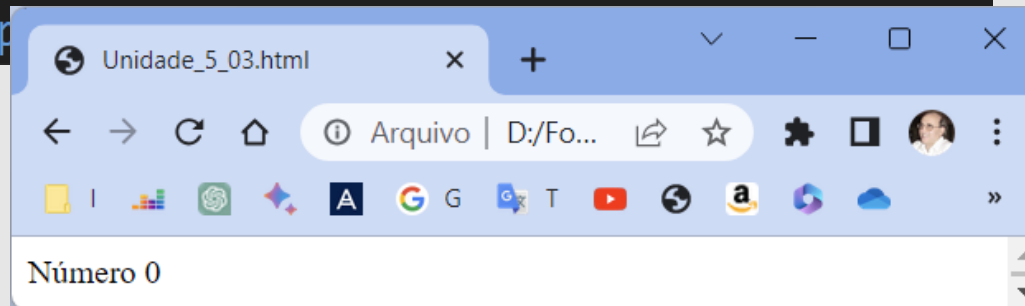
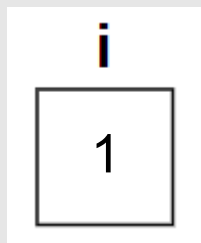
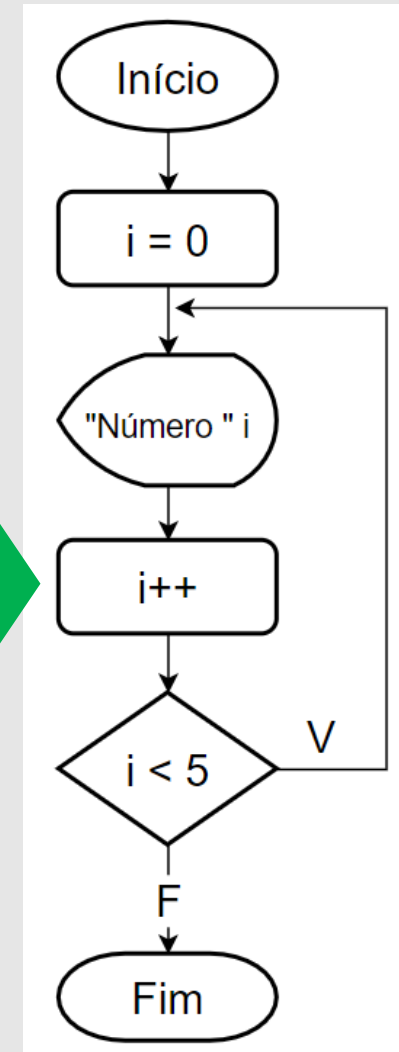


do ... while - Exemplo

```

10 <script>
11   function exhibeNumeros() {
12
13     let i = 0;
14
15     do {
16       document.write("<p>" + "Número " + i + "<p>");
17       i++;
18     } while (i < 5)
19
20   }
21   window.onload = exhibeNumeros;
22 </script>

```

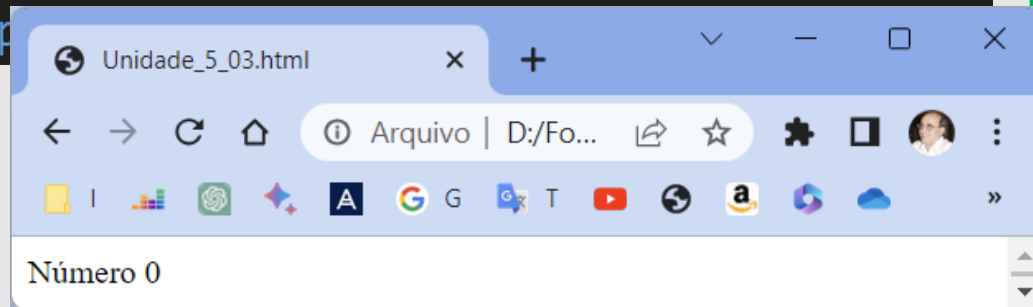
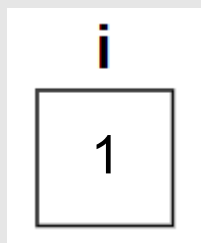
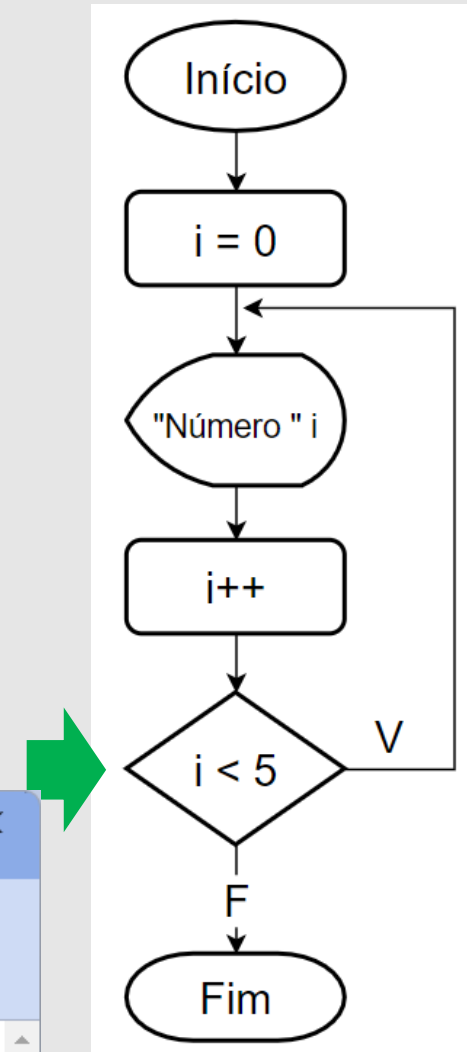


do ... while - Exemplo

```

10  <script>
11      function exibeNumeros() {
12
13          let i = 0;
14
15          do {
16              document.write("<p>" + "Número " + i + "<p>");
17              i++;
18          } while (i < 5)
19
20      }
21      window.onload = exibeNumeros;
22  </script>

```

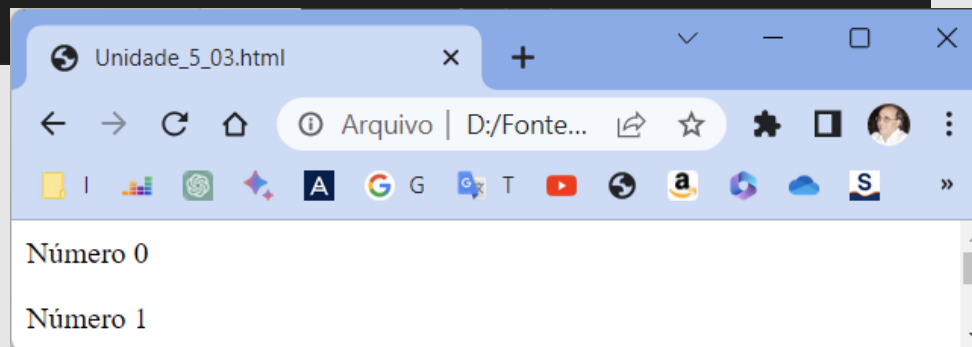
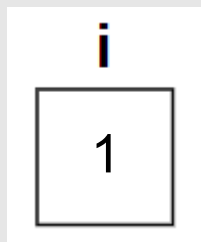
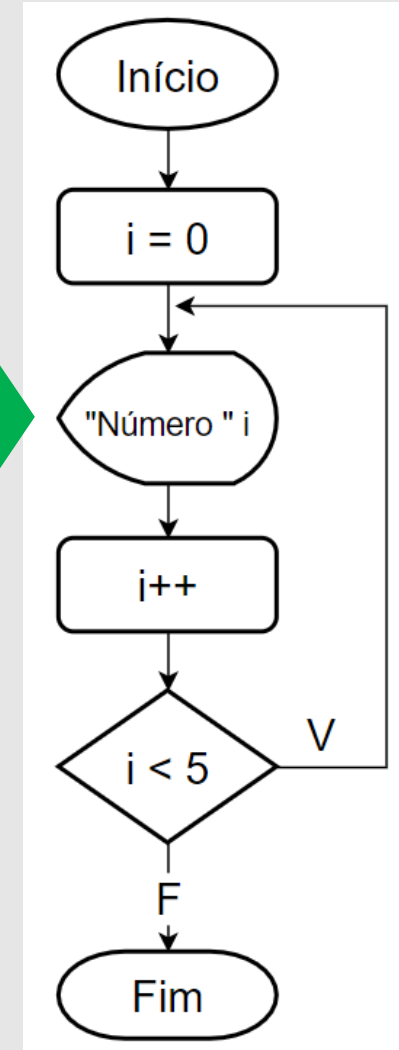


do ... while – Exemplo

```

10 <script>
11   function exhibeNumeros() {
12
13     let i = 0;
14
15     do {
16       document.write("<p>" + "Número " + i + "<p>");
17       i++;
18     } while (i < 5)
19
20   }
21   window.onload = exhibeNumeros;
22 </script>

```

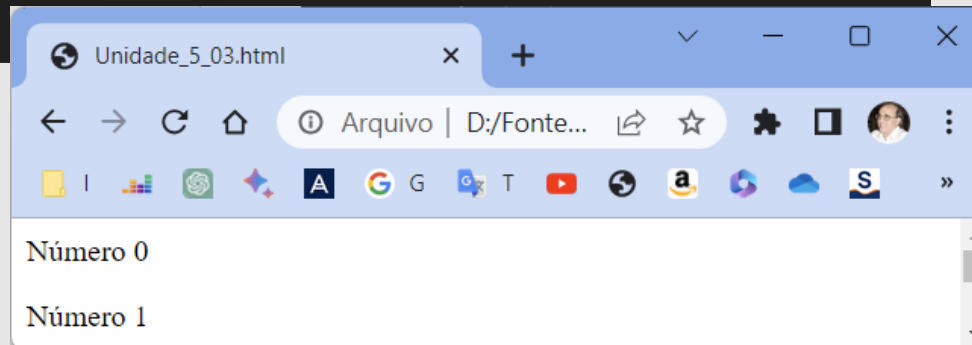
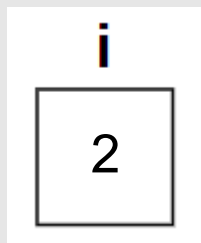
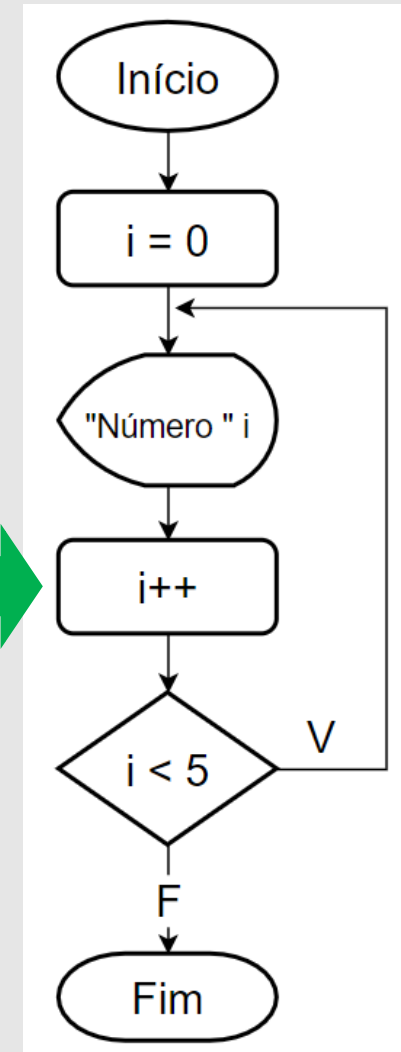


do ... while - Exemplo

```

10 <script>
11   function exibeNumeros() {
12
13     let i = 0;
14
15     do {
16       document.write("<p>" + "Número " + i + "<p>");
17       i++;
18     } while (i < 5)
19
20   }
21   window.onload = exibeNumeros;
22 </script>

```

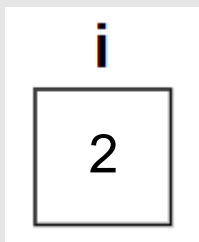
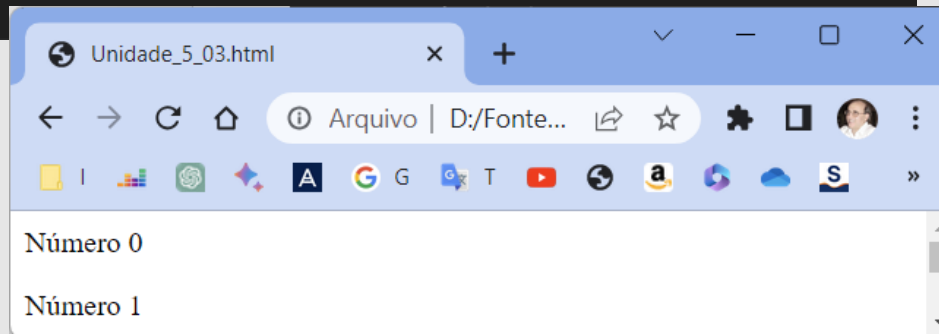
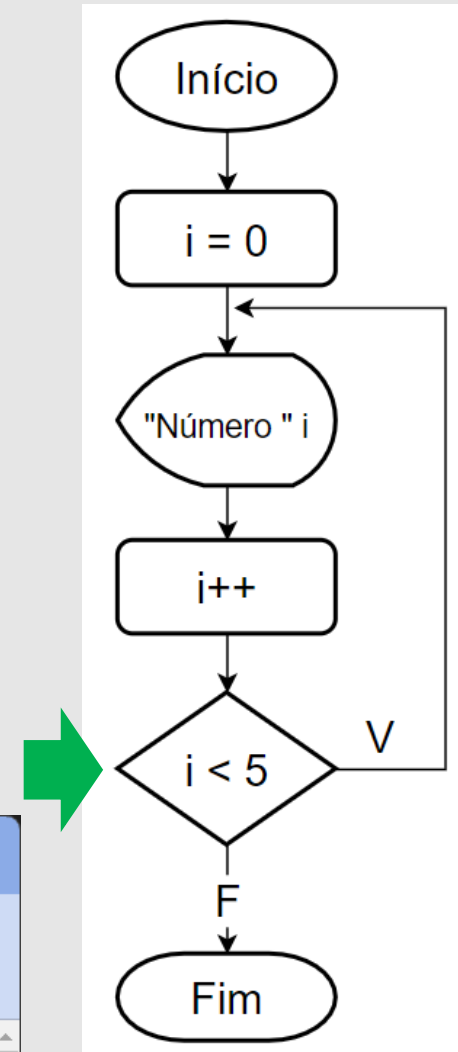


do ... while - Exemplo

```

10  <script>
11      function exibeNumeros() {
12
13          let i = 0;
14
15          do {
16              document.write("<p>" + "Número " + i + "<p>");
17              i++;
18          } while (i < 5)
19
20      }
21      window.onload = exibeNumeros;
22  </script>

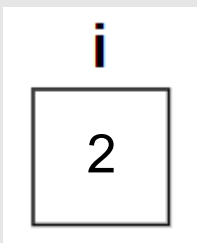
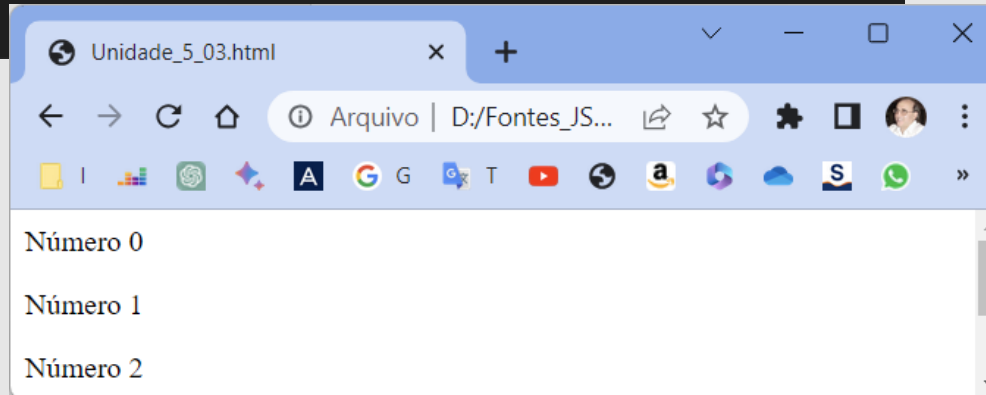
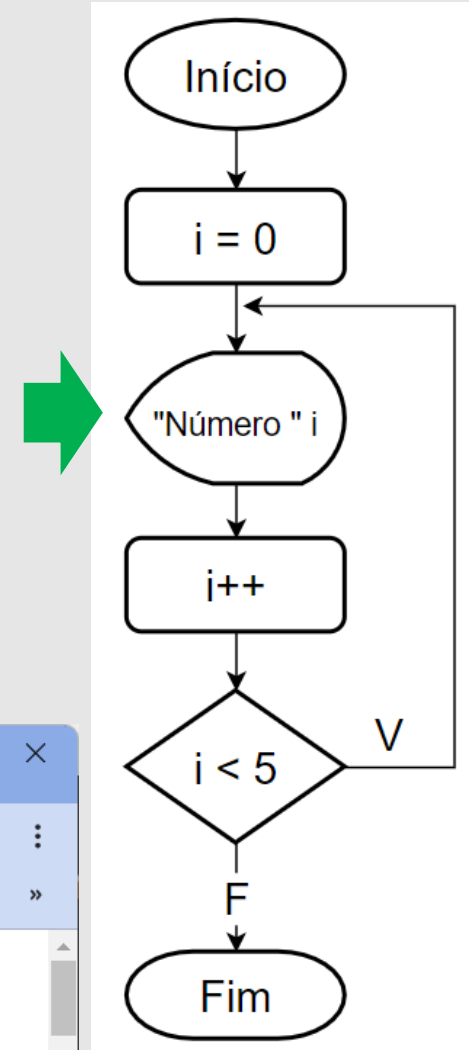
```



do ... while – Exemplo

```

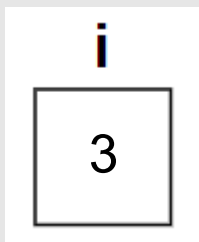
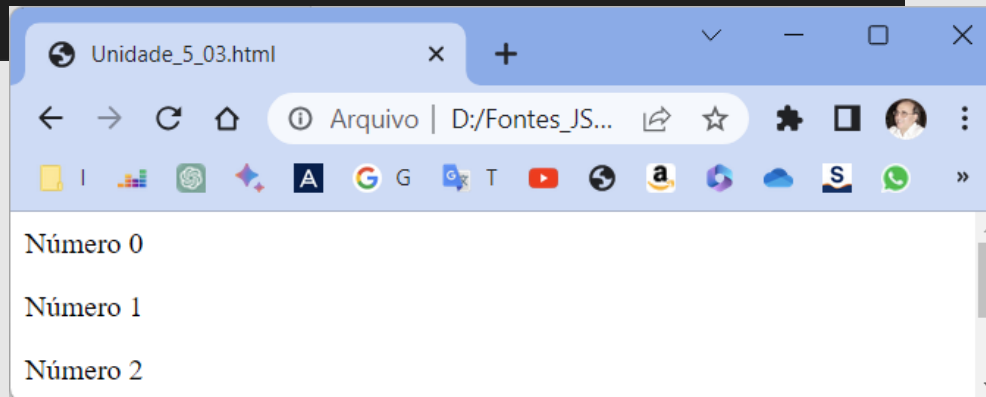
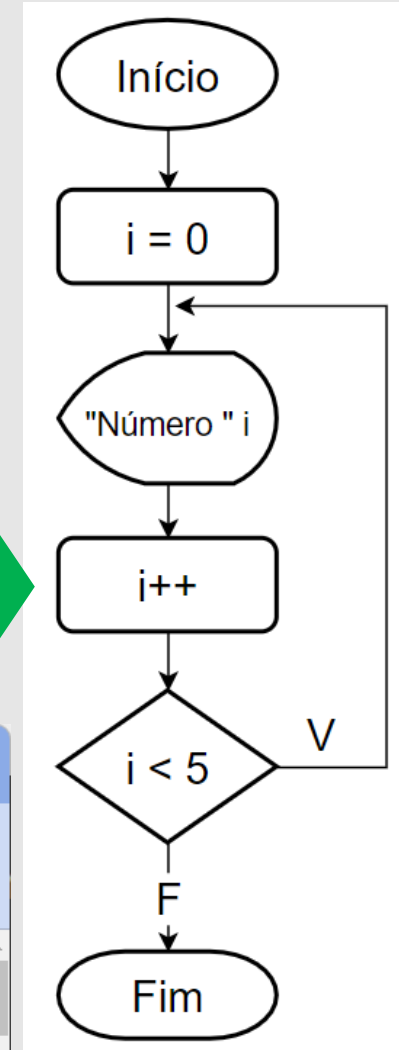
10  <script>
11      function exibeNumeros() {
12
13          let i = 0;
14
15          do {
16              document.write("<p>" + "Número " + i + "<p>");
17              i++;
18          } while (i < 5)
19
20      }
21      window.onload = exibeNumeros;
22  </script>
    
```



do ... while – Exemplo

```

10  <script>
11      function exibeNumeros() {
12
13          let i = 0;
14
15          do {
16              document.write("<p>" + "Número " + i + "<p>");
17              i++;
18          } while (i < 5)
19
20      }
21      window.onload = exibeNumeros;
22  </script>
    
```

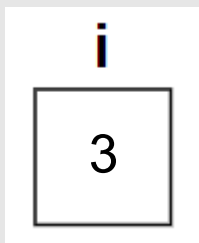
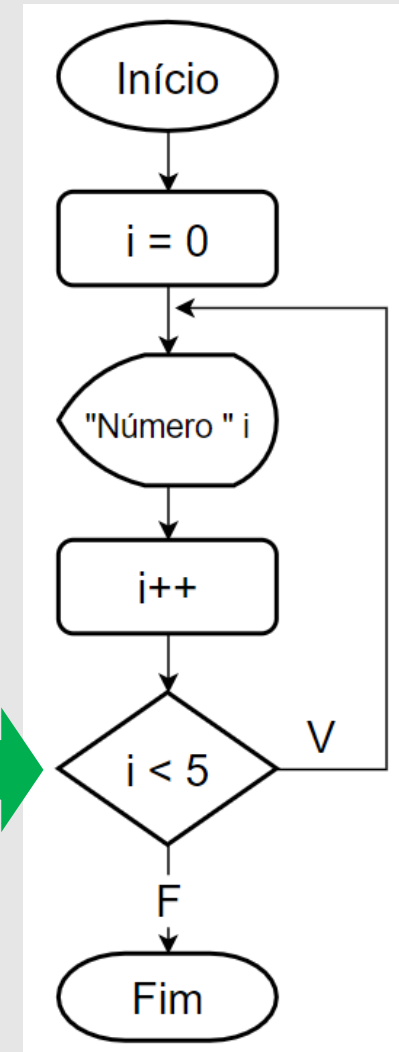
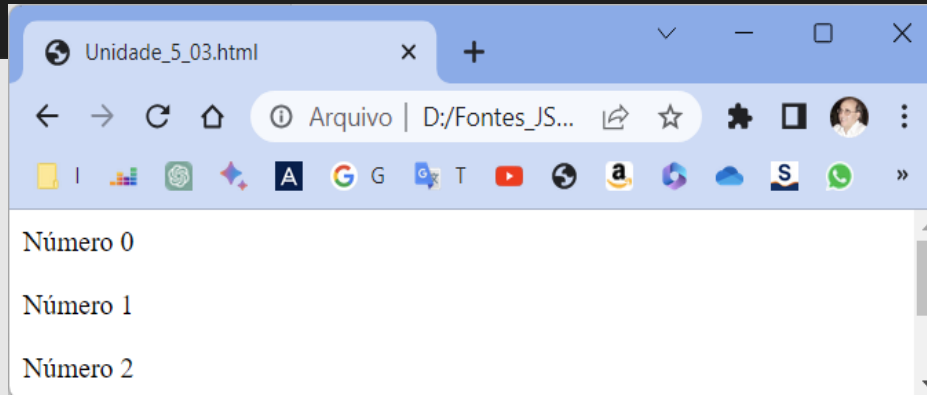


do ... while - Exemplo

```

10  <script>
11    function exibeNumeros() {
12
13      let i = 0;
14
15      do {
16        document.write("<p>" + "Número " + i + "<p>");
17        i++;
18      } while (i < 5)
19
20    }
21    window.onload = exibeNumeros;
22  </script>

```



do ... while - Exemplo

```

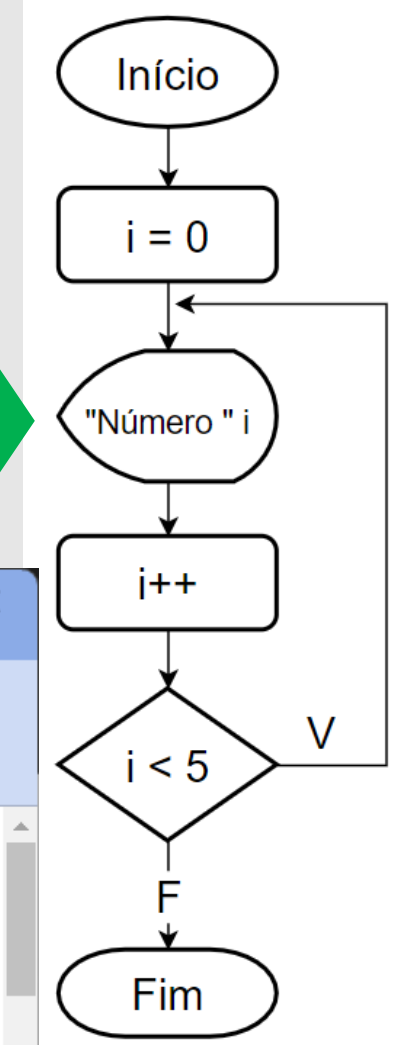
10 <script>
11   function exibeNumeros() {
12
13     let i = 0;
14
15     do {
16       document.write("<p>" + "Número " + i + "<p>");
17       i++;
18     } while (i < 5)
19
20   }
21
22 </script>

```



i

3

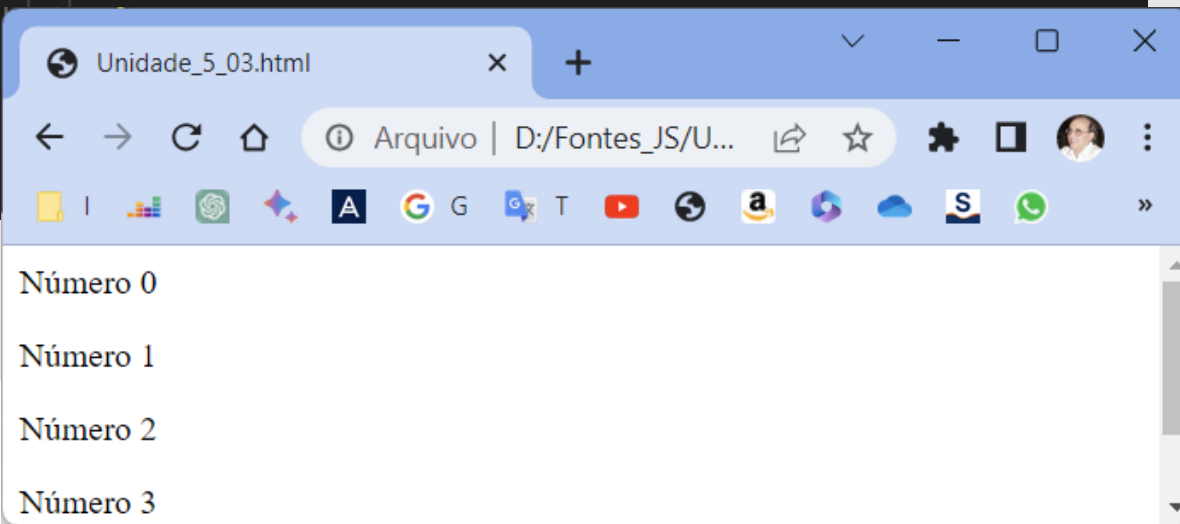


do ... while - Exemplo

```

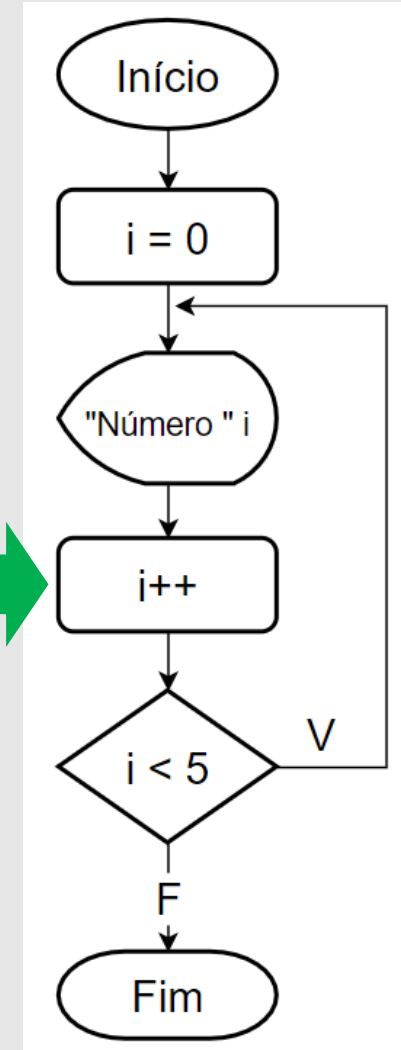
10 <script>
11   function exibeNumeros() {
12
13     let i = 0;
14
15     do {
16       document.write("<p>" + "Número " + i + "<p>");
17       i++;
18     } while (i < 5)
19
20
21
22

```



i

4

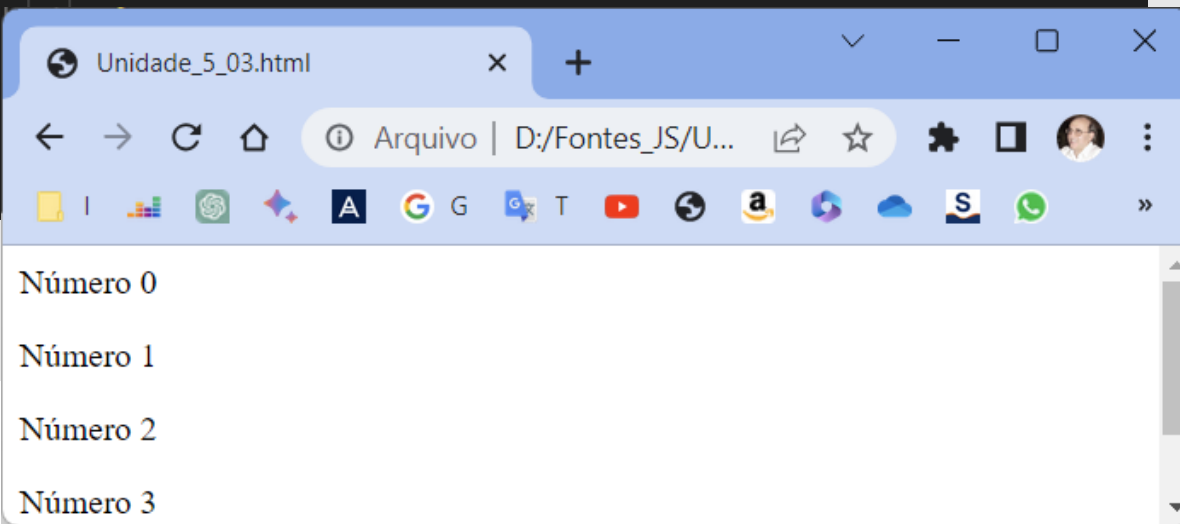


do ... while - Exemplo

```

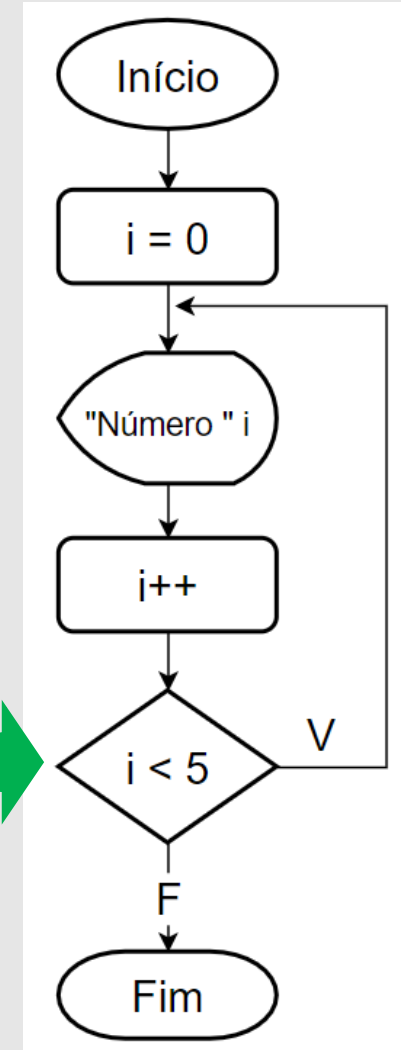
10 <script>
11   function exibeNumeros() {
12
13     let i = 0;
14
15     do {
16       document.write("<p>" + "Número " + i + "<p>");
17       i++;
18     } while (i < 5)
19
20
21
22

```



i

4

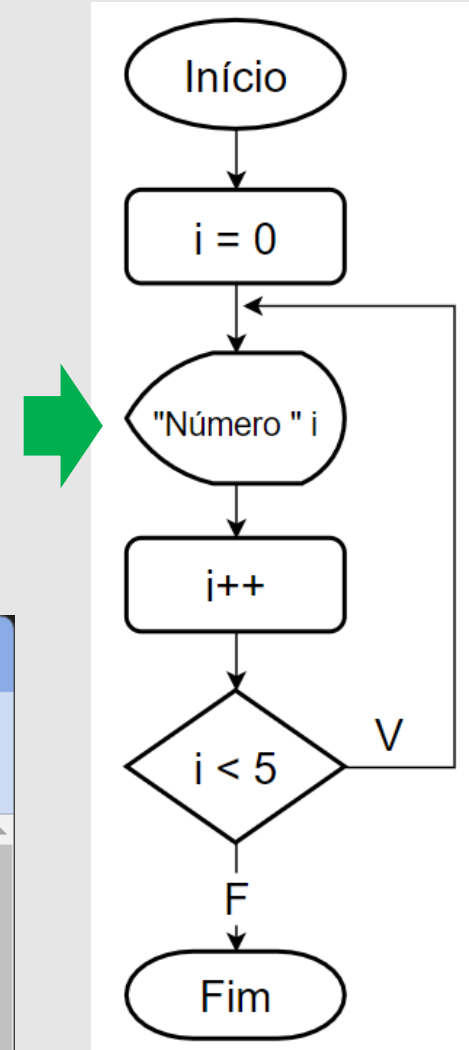
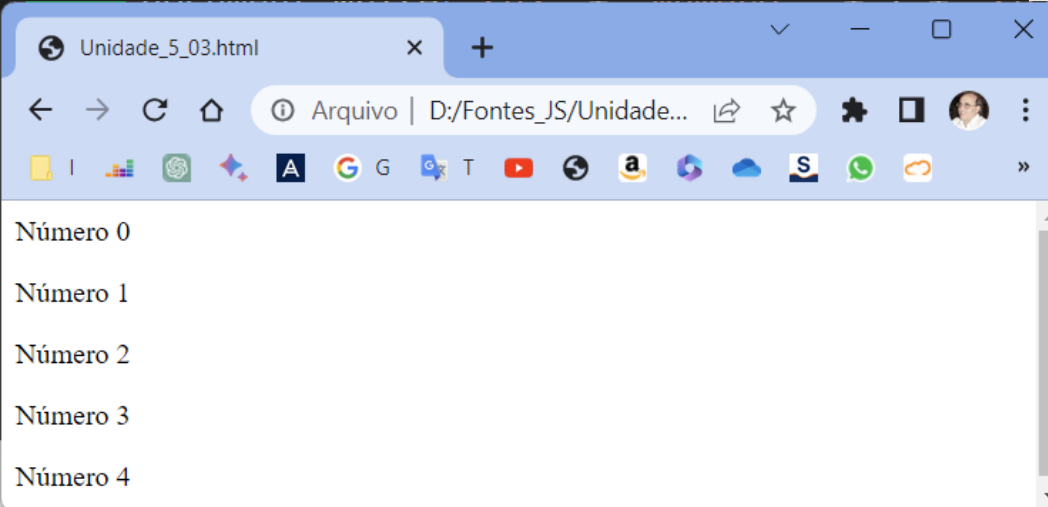


do ... while – Exemplo

```

10 <script>
11   function exibeNumeros() {
12
13     let i = 0;
14
15     do {
16       document.write("<p>" + "Número " + i + "<p>");
17       i++;
18     } while (i < 5)
19
20   }
21
22 </script>

```



i

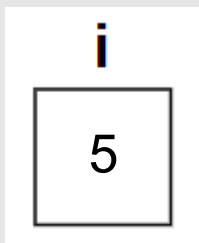
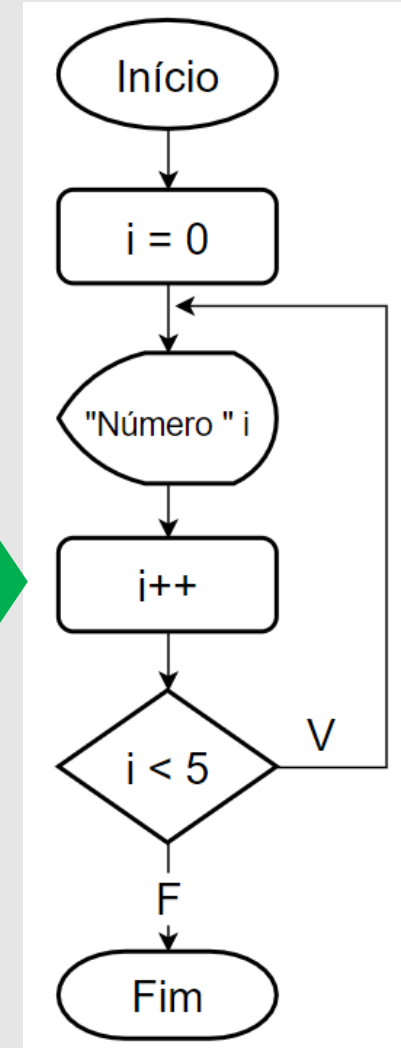
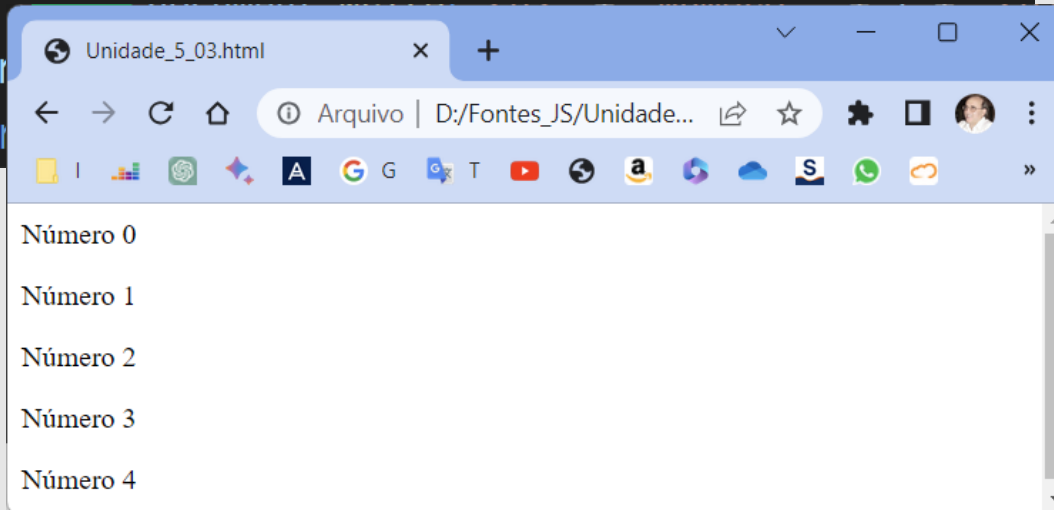
4

do ... while - Exemplo

```

10 <script>
11   function exibeNumeros() {
12
13     let i = 0;
14
15     do {
16       document.write("<p>" + "Número " + i + "<p>");
17       i++;
18     } while (i < 5)
19
20   }
21
22 </script>

```

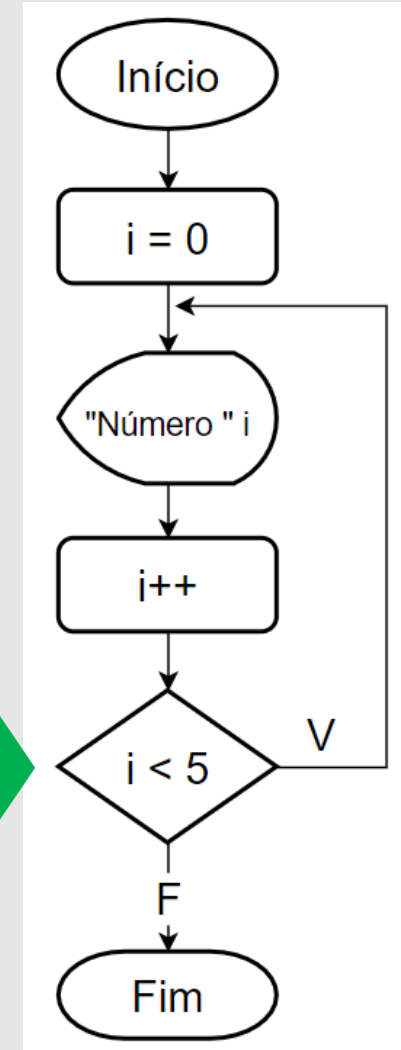
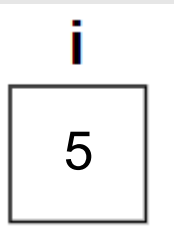
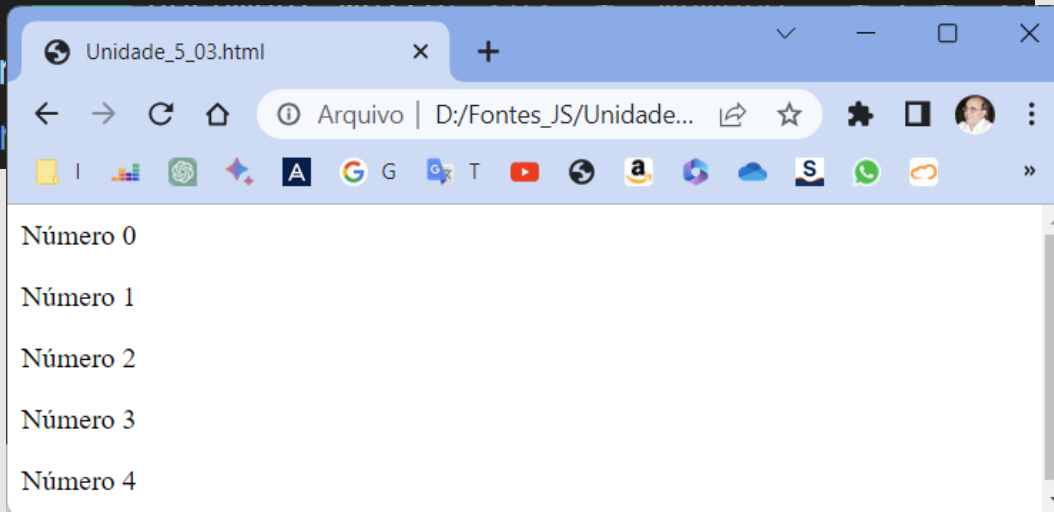


do ... while - Exemplo

```

10  <script>
11    function exibeNumeros() {
12
13      let i = 0;
14
15      do {
16        document.write("<p>" + "Número " + i + "<p>");
17        i++;
18      } while (i < 5)
19
20    }
21
22  </script>

```

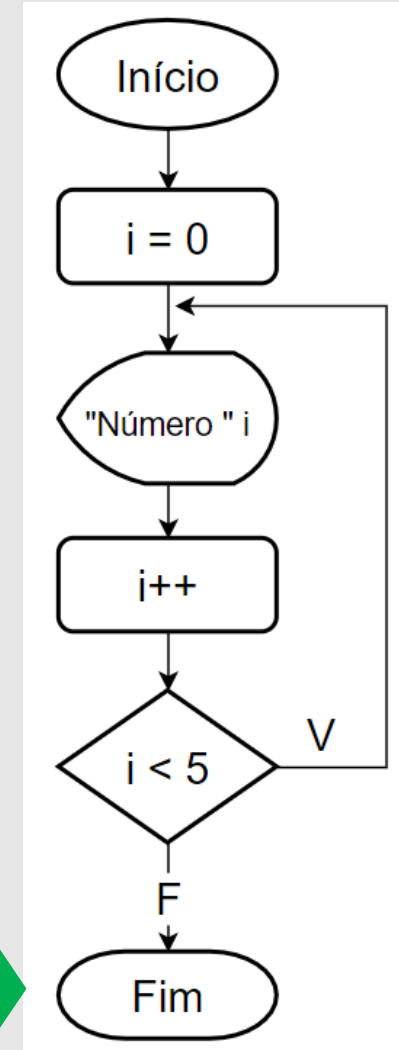
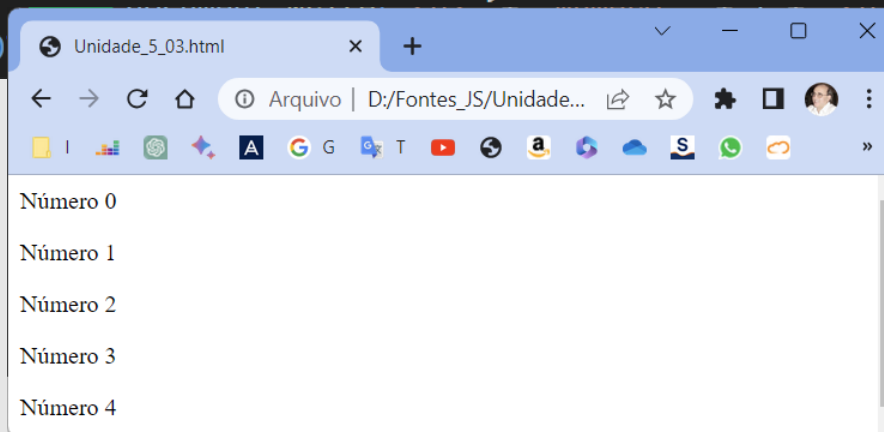
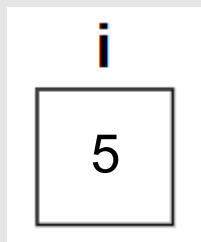


do ... while – Exemplo

```

10 <script>
11   function exibeNumeros() {
12
13     let i = 0;
14
15     do {
16       document.write("<p>" + "Número " + i + "<p>");
17       i++;
18     } while (i < 5)
19
20   }
21   window.onload = exibeNumeros;
22 </script>

```



Laço for

```
for (<início>; <fim>; <incremento>)  
{  
    <executa instrução 1>  
    <executa instrução 2>  
    <executa instrução 3>  
    <executa instrução N>  
}
```

Laço for

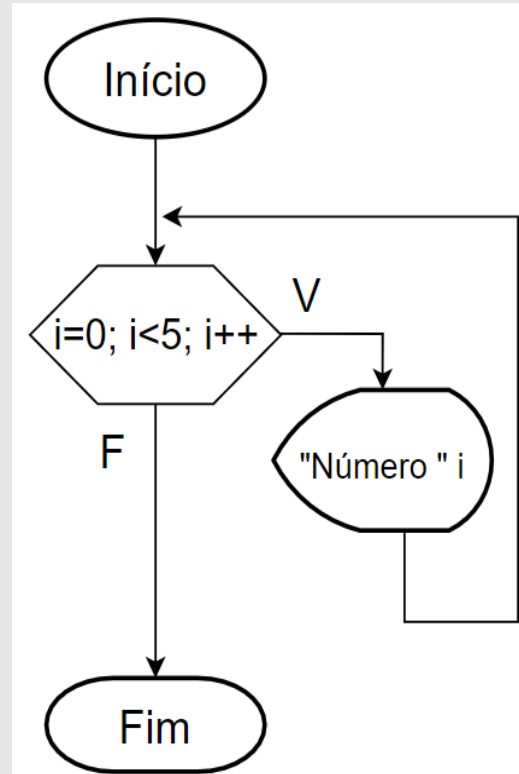
- O laço **for** é executado por uma instrução que recebe três parâmetros de trabalho, separados por ponto e vírgula;
- O primeiro parâmetro é representado por uma variável e seu valor inicial;
- O segundo parâmetro, pela condição de finalização da execução do laço;
- O terceiro e último parâmetro, pelo contador de incremento que ocorre entre os valores inicial e final da contagem.

Exemplo - for

```

1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <!------->
4
5  <head>
6    <meta charset="UTF-8">
7    <meta http-equiv="X-UA-Compatible" content="IE=edge">
8    <meta name="viewport" content="width=device-width, initial-scale=1.0">
9
10   <script>
11     function exibeNumeros() {
12
13       let i = 0;
14
15       for (i = 0; i < 5; i++)
16         document.write("<p>" + "Número " + i + "<p>");
17     }

```

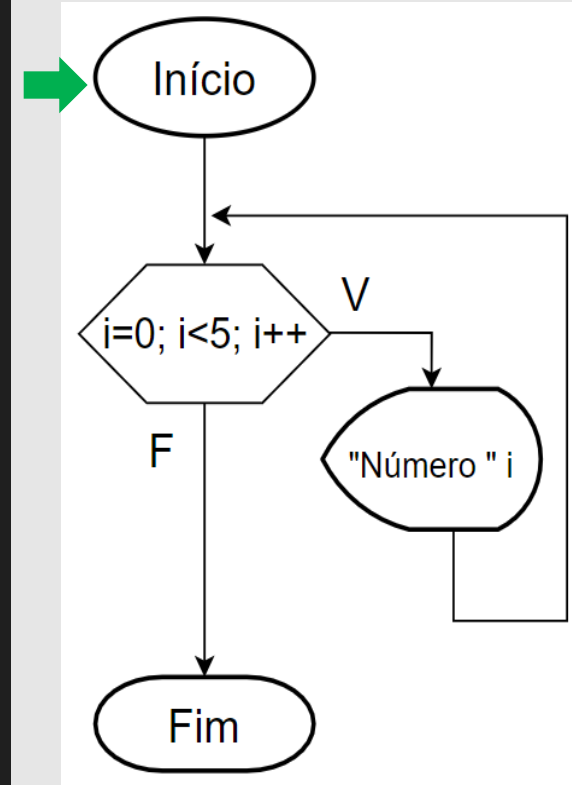


Exemplo - for

```

1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <!------->
4
5  <head>
6    <meta charset="UTF-8">
7    <meta http-equiv="X-UA-Compatible" content="IE=edge">
8    <meta name="viewport" content="width=device-width, initial-scale=1.0">
9
10   <script>
11     function exibeNumeros() {
12
13       let i = 0;
14
15       for (i = 0; i < 5; i++)
16         document.write("<p>" + "Número " + i + "<p>");
17     }

```

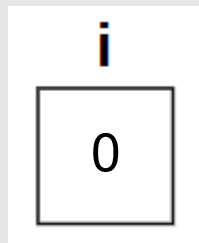
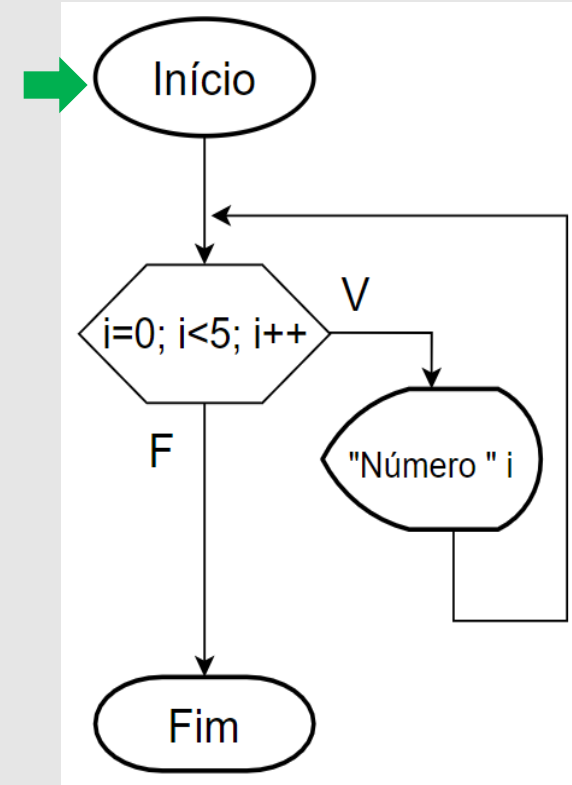


Exemplo - for

```

1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <!------->
4
5  <head>
6    <meta charset="UTF-8">
7    <meta http-equiv="X-UA-Compatible" content="IE=edge">
8    <meta name="viewport" content="width=device-width, initial-scale=1.0">
9
10   <script>
11     function exibeNumeros() {
12
13     → let i = 0;
14
15       for (i = 0; i < 5; i++)
16         document.write("<p>" + "Número " + i + "<p>");
17     }

```

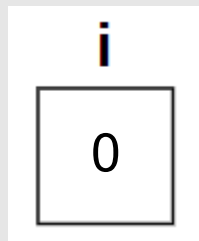
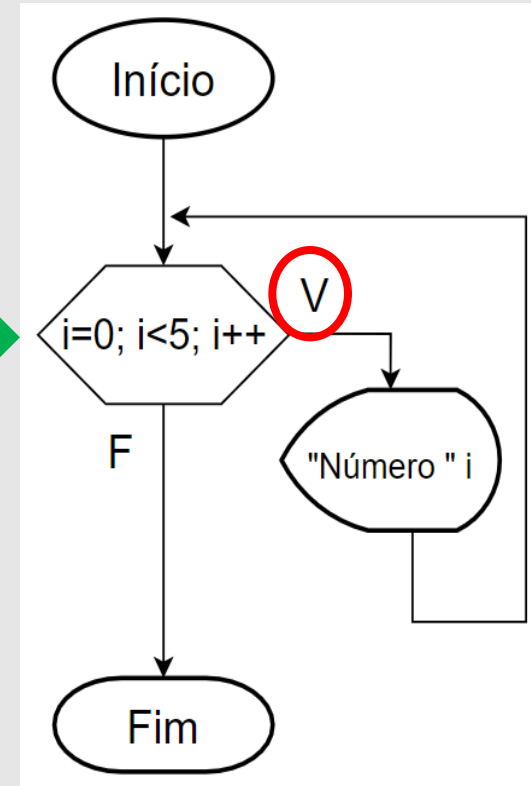


Exemplo - for

```

1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <!------->
4
5  <head>
6    <meta charset="UTF-8">
7    <meta http-equiv="X-UA-Compatible" content="IE=edge">
8    <meta name="viewport" content="width=device-width, initial-scale=1.0">
9
10   <script>
11     function exibeNumeros() {
12
13       let i = 0;
14
15       for (i = 0; i < 5; i++)
16         document.write("<p>" + "Número " + i + "<p>");
17     }

```

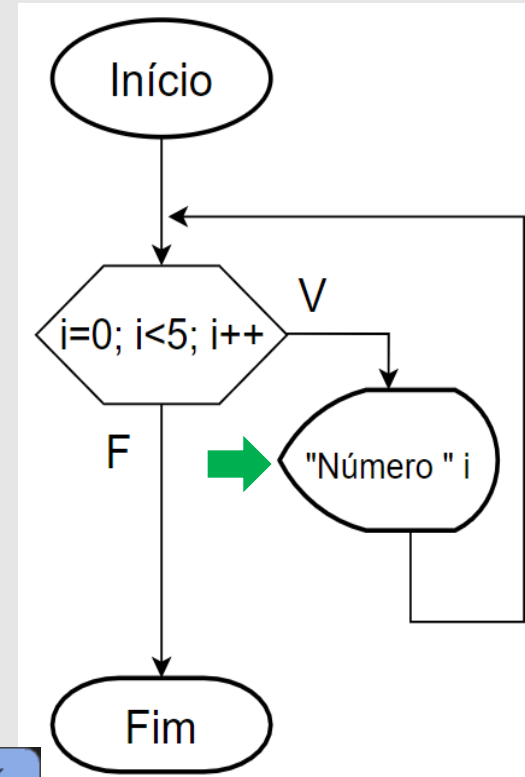


Exemplo - for

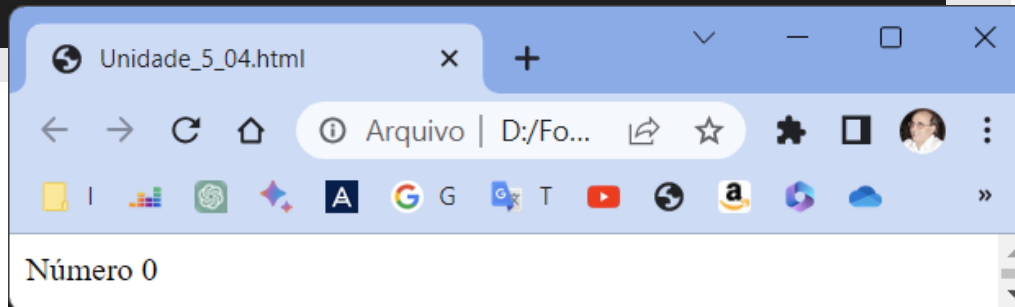
```

1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <!------->
4
5  <head>
6    <meta charset="UTF-8">
7    <meta http-equiv="X-UA-Compatible" content="IE=edge">
8    <meta name="viewport" content="width=device-width, initial-scale=1.0">
9
10   <script>
11     function exibeNumeros() {
12
13       let i = 0;
14
15       for (i = 0; i < 5; i++)
16         document.write("<p>" + "Número " + i + "<p>");
17     }

```



i
0

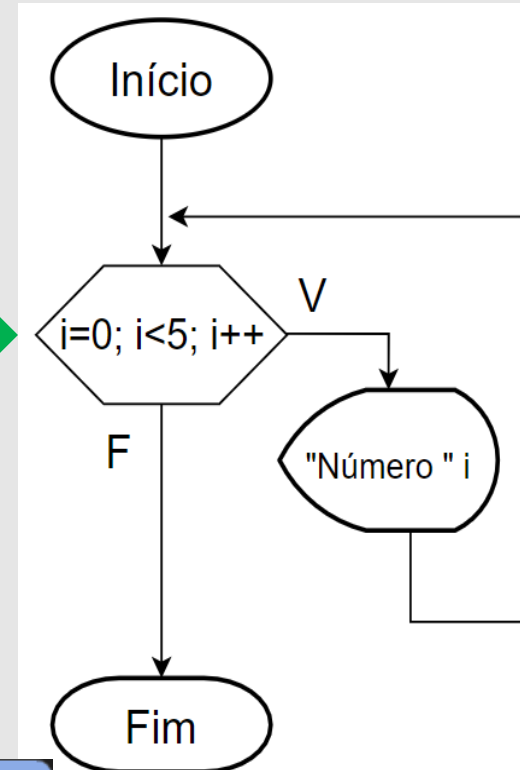


Exemplo - for

```

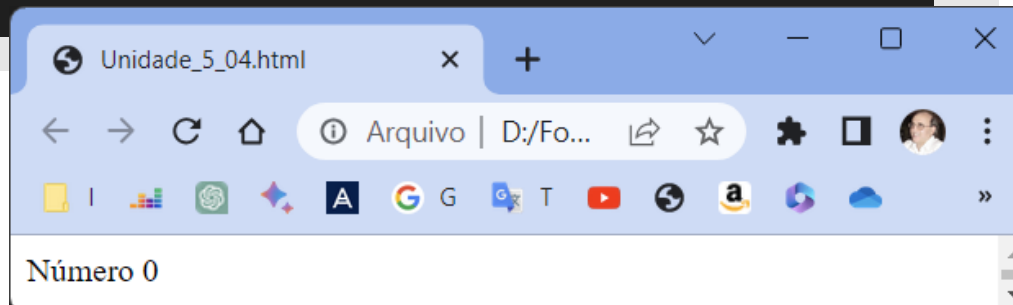
1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <!------->
4
5  <head>
6    <meta charset="UTF-8">
7    <meta http-equiv="X-UA-Compatible" content="IE=edge">
8    <meta name="viewport" content="width=device-width, initial-scale=1.0">
9
10   <script>
11     function exibeNumeros() {
12
13       let i = 0;
14
15       for (i = 0; i < 5; i++)
16         document.write("<p>" + "Número " + i + "<p>");
17     }

```



i

1

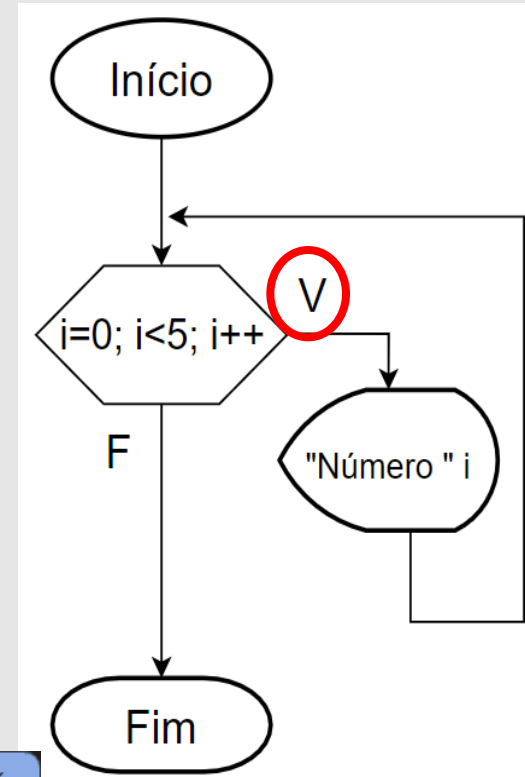


Exemplo - for

```

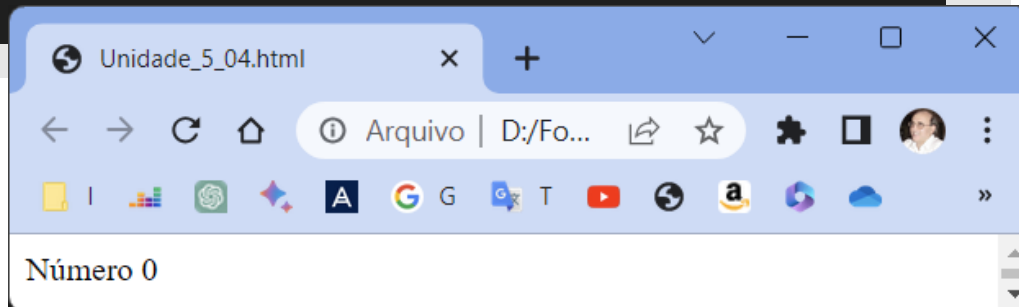
1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <!------->
4
5  <head>
6    <meta charset="UTF-8">
7    <meta http-equiv="X-UA-Compatible" content="IE=edge">
8    <meta name="viewport" content="width=device-width, initial-scale=1.0">
9
10   <script>
11     function exibeNumeros() {
12
13       let i = 0;
14
15       for (i = 0; i < 5; i++)
16         document.write("<p>" + "Número " + i + "<p>");
17     }

```



i

1

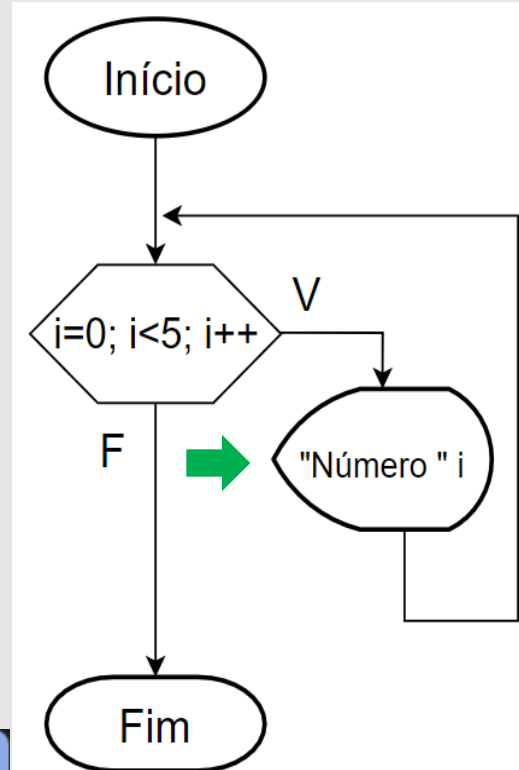


Exemplo - for

```

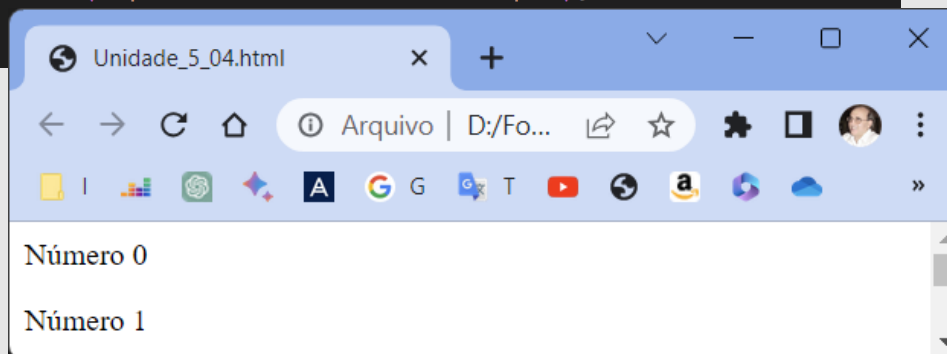
1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <!------->
4
5  <head>
6    <meta charset="UTF-8">
7    <meta http-equiv="X-UA-Compatible" content="IE=edge">
8    <meta name="viewport" content="width=device-width, initial-scale=1.0">
9
10   <script>
11     function exibeNumeros() {
12
13       let i = 0;
14
15       for (i = 0; i < 5; i++)
16         document.write("<p>" + "Número " + i + "<p>");
17     }

```



i

1

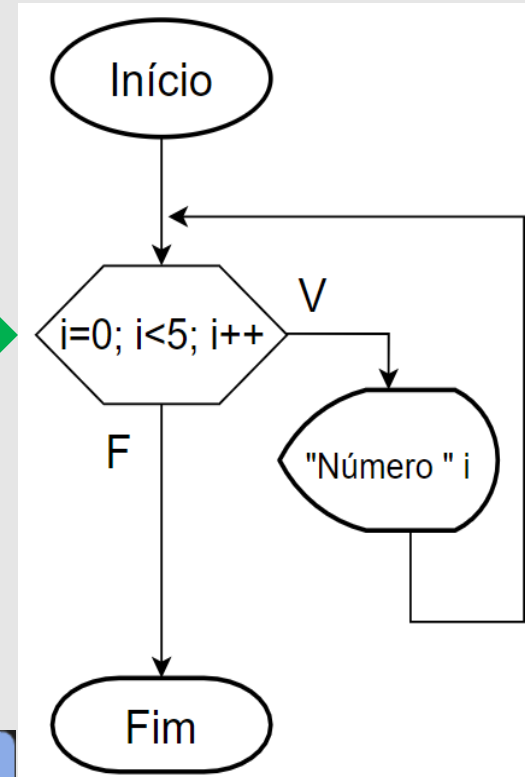


Exemplo - for

```

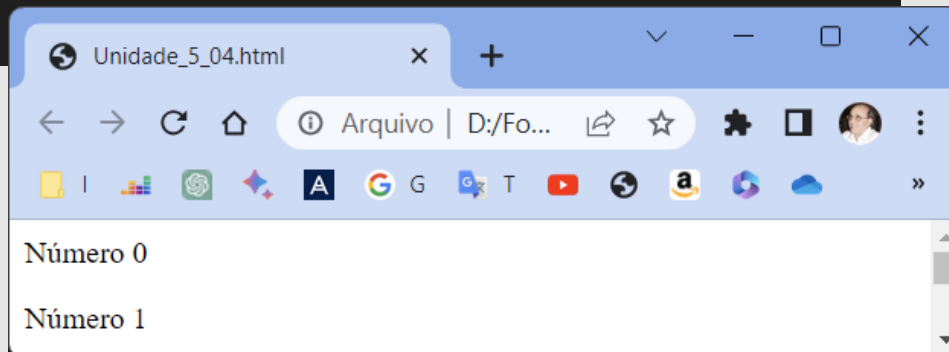
1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <!------->
4
5  <head>
6    <meta charset="UTF-8">
7    <meta http-equiv="X-UA-Compatible" content="IE=edge">
8    <meta name="viewport" content="width=device-width, initial-scale=1.0">
9
10   <script>
11     function exibeNumeros() {
12
13       let i = 0;
14
15       for (i = 0; i < 5; i++)
16         document.write("<p>" + "Número " + i + "<p>");
17     }

```



i

2



Exemplo - for

```

1  <!DOCTYPE html>
2  <html lang="pt-br">
3  <!------->
4
5  <head>
6    <meta charset="UTF-8">
7    <meta http-equiv="X-UA-Compatible" content="IE=edge">
8    <meta name="viewport" content="width=device-width, initial-scale=1.0">
9
10   <script>
11     function exibeNumeros() {
12
13       let i = 0;
14
15       for (i = 0; i < 5; i++)
16         document.write("<p>" + "Número " + i + "<p>");
17     }

```

