



(✓) Curso Técnico

Quem Investe no futuro faz QI

Informática para Internet

Desenvolvimento de Sistemas

Web III

Unidade III

1. Operadores de incremento e decremento

Os operadores de incremento e decremento são operadores unários usados para adicionar ou subtrair 1 do valor de uma variável numérica. O operador de incremento é simbolizado por dois sinais de adição, ++, e o de decremento, por dois subtração, --.

x++	x = x + 1, soma um ao valor contido em x
x--	x = x - 1, subtrai um do valor contido em x
++x	x = x + 1, soma um ao valor contido em x, com uso do resultado na mesma linha
--x	x = x - 1, subtrai um do valor contido em x, com uso do resultado na mesma linha

2. Operadores lógicos

Os operadores lógicos relacionam valores lógicos como true/false, e o resultado de uma expressão lógica é outro valor lógico.

Os operadores lógicos são: E (AND), OU (OR), NÃO (NOT), NÃO-E (NAND), NÃO-OU (NOR), OU-EXCLUSIVO (XOR) e NÃO-OU-EXCLUSIVO (XNOR), mas os mais comumente utilizados na tecnologia Java são E, OU e o NÃO.

A forma de representação dos operadores lógicos na linguagem de programação varia, mas na Java a simbologia utilizada é a seguinte:

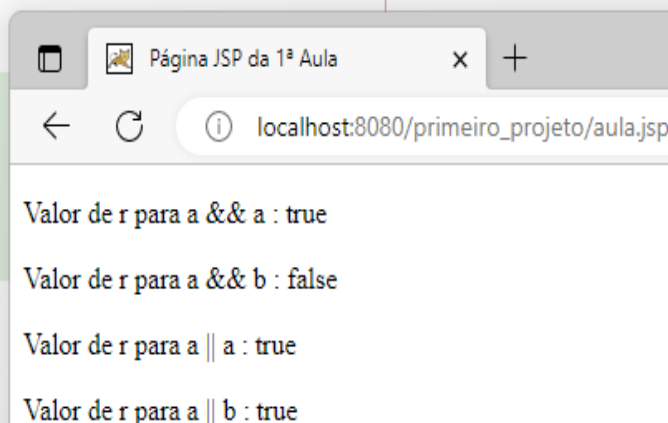
&&	“and” ou “e”
	“or” ou “ou”
!	“not” ou “inversão”

Importante ressaltar que o resultado de uma operação lógica será sempre false ou true na linguagem Java, conforme exemplos:

```

1  <%--
2      Document    : aula
3      Created on  : Em 2023
4      Author     : Conteudista
5  --%>
6  <%@page contentType="text/html" pageEncoding="UTF-8"%>
7  <!DOCTYPE html>
8  <html>
9      <head>
10         <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
11         <title>Página JSP da 1ª Aula</title>
12     </head>
13     <body>
14         <%
15             boolean a,b,r ;
16             a=true;
17             b=false;
18             r = a && a;
19         %>
20         <p>Valor de r para a && a : <%=r %></p>
21         <% r = a && b; %>
22         <p>Valor de r para a && b : <%=r %></p>
23         <% r = a || a; %>
24         <p>Valor de r para a || a : <%=r %></p>
25         <% r = a || b; %>
26         <p>Valor de r para a || b : <%=r %></p>
27     </body>
28 </html>

```



3. Operadores relacionais

Operadores relacionais são utilizados para comparar valores, preferencialmente do mesmo tipo, sendo que o resultado de uma expressão relacional é um valor booleano, i.e., false ou true.

Os operadores relacionais são:

igual	==
diferente	!=
maior	>
menor	<
maior ou igual	>=
menor ou igual	<=

No exemplo abaixo é possível notar que as variáveis comparadas são do tipo int, mas a variável que recebe o resultado da comparação é do tipo boolean:

```

1 <%@page contentType="text/html" pageEncoding="UTF-8"%>
2 <!DOCTYPE html>
3 <html>
4 <head>
5     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
6     <title>Página JSP da 1ª Aula</title>
7 </head>
8 <body>
9     <%
10         int a,b;
11         boolean r ;
12         a=2;
13         b=3;
14         r = a == b;
15     %>
16     <p>Valor de r para a == a : <%=r %></p>
17     <% r = a != b; %>
18     <p>Valor de r para a != b : <%=r %></p>
19     <% r = a > b; %>
20     <p>Valor de r para a > b : <%=r %></p>
21     <% r = a < b; %>
22     <p>Valor de r para a < b : <%=r %></p>
23     <% r = a >= b; %>
24     <p>Valor de r para a >= b : <%=r %></p>
25     <% r = a <= b; %>
26     <p>Valor de r para a <= b : <%=r %></p>
27 </body>
28 </html>

```

Valor de r para a == a : false

Valor de r para a != b : true

Valor de r para a > b : false

Valor de r para a < b : true

Valor de r para a >= b : false

Valor de r para a <= b : true

Vale ainda ressaltar que para os dados do “tipo” String, quando entende-se que expressões ou palavras deverão ser comparadas, a função equals da classe String deverá ser utilizada no detrimento do operador de igualdade ==, mesmo porque na linguagem Java a classe String permite a declaração/criação de objetos/variáveis que possibilitam o armazenamento de caracteres de qualquer tipo, salientando que os objetos/variáveis do “tipo” String terão métodos , conforme exemplifica-se:

```

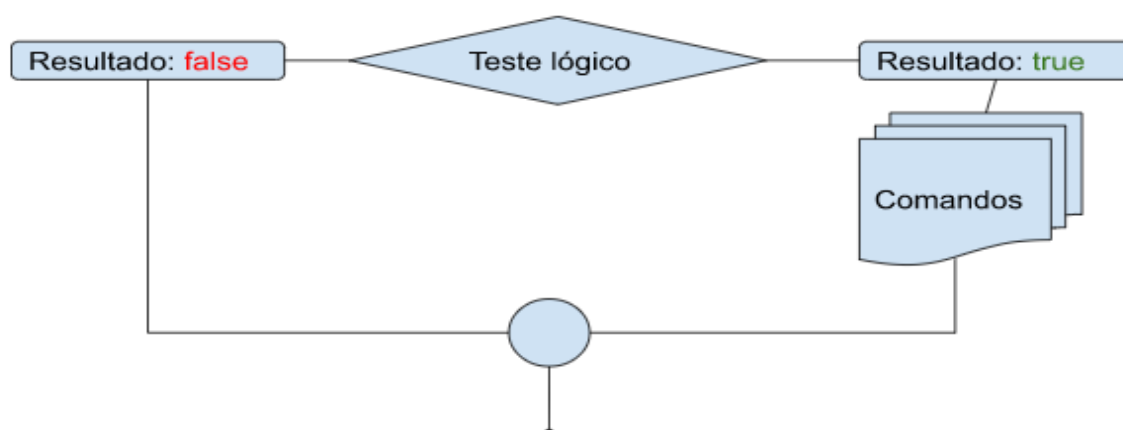
1  <%@page contentType="text/html" pageEncoding="UTF-8"%>
2  <!DOCTYPE html>
3  <html>
4  <head>
5      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
6      <title>Página JSP da 1ª Aula</title>
7  </head>
8  <body>
9      <%
10         String a,b;
11         a = new String("teste");// poderia ser a="teste";
12         b = new String("teste");//poderia ser b="teste";
13         boolean r ;
14         r = a == b;
15     %>
16     <p>Valor de r para a == b : <%=r %></p>
17     <% r = a.equals(b); %>
18     <p>Valor de r para a.equals(b) : <%=r %></p>
19 </body>
20 </html>

```



4. Estrutura condicional

A estrutura condicional na linguagem Java permite o condicionamento da execução de comandos ao resultado de um teste lógico, observado o seguinte fluxograma:



Na estrutura condicional os comandos serão executados quando o resultado do teste lógico for **true**, neste contexto ao utilizar a linguagem Java as seguintes estruturas condicionais poderão ser utilizadas:

5. Estrutura condicional if (se)

A estrutura condicional ou de controle ou de tomada de decisão if (se), é utilizada para condicionar a execução de comandos apenas se a sua condição for true (verdadeira), contando ainda com o recurso else (senão) para os casos em que há comandos para execução senão for verdadeira a condição imposta.

O else é um recurso facultativo ao if, mas o else não pode ser utilizado sozinho, ou seja, sempre que houver um else haverá um if.

As possíveis sintaxes de utilização do if na linguagem Java são:

6. Estrutura condicional if com uma instrução/comando

if(teste lógico)

uma instrução/comando;



```
1 <%@page contentType="text/html" pageEncoding="UTF-8"%>
2 <!DOCTYPE html>
3 <html>
4   <head>
5     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
6     <title>Página JSP da 1ª Aula</title>
7   </head>
8   <body>
9     <p>
10       <%
11         int a,b;
12         String r ;
13         r = "Teste lógico =false";
14         a=2;
15         b=3;
16         if(a>b)
17           r = "teste lógico = true";
18         out.print(r);
19       %>
20     </p>
21   </body>
22 </html>
```

Página JSP da 1ª Aula

localhost:8080/primeiro_proj

Teste lógico =false

7. Estrutura condicional if com mais de uma instrução/comando

```
if(teste lógico){
    primeira instrução/comando;
    segunda instrução/comando;
}
```

No exemplo abaixo a mensagem só irá aparecer se o teste for verdadeiro, pois as chaves determinam o bloco “condicionado/protegido” pelo `if`:

The screenshot shows a JSP code editor on the left and a web browser on the right. The code in the editor is as follows:

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Página JSP da 1ª Aula</title>
</head>
<body>
<p>
<%
    int a,b;
    String r ;
    r = "Teste lógico =false";
    a=2;
    b=3;
    if(a>b){
        r = "teste lógico = true";
        out.print(r);
    }
%>
</p>
</body>
</html>
```

The browser on the right shows the title "Página JSP da 1ª Aula" and the URL "localhost:8080/primeiro_projeto/operadores.jsp". The browser content area is empty, indicating that the output of the JSP code is not visible in the screenshot.

8. Estrutura condicional if com else

```
if(teste lógico){
    Instruções/comandos;
}else{
    Instruções/comandos;
}
```

No exemplo abaixo a mensagem "teste lógico = true" irá aparecer se o teste for verdadeiro, senão aparecerá o valor definido para variável na linha 13:

```

1  <%@page contentType="text/html" pageEncoding="UTF-8"%>
2  <!DOCTYPE html>
3  <html>
4      <head>
5          <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
6          <title>Página JSP da 1ª Aula</title>
7      </head>
8      <body>
9          <p>
10             <%
11                 int a,b;
12                 String r ;
13                 r = "Teste lógico =false";
14                 a=2;
15                 b=3;
16                 if(a>b) {
17                     r = "teste lógico = true";
18                     out.print(r);
19                 }else{
20                     out.print(r);
21                 }
22             %>
23         </p>
24     </body>
25 </html>

```

Teste lógico =false

9. Estrutura condicional if encadeado ou com múltiplas condições

```

if(teste lógico){
    Instruções/comandos;
}else if(teste lógico){
    Instruções/comandos;
}else{
    Instruções/comandos;
}

```

No exemplo abaixo a mensagem está uma estrutura condicional encadeada que verificará qual a situação do valor da variável “a” em relação a variável “b”, vale

salientar que mais estruturas condicionais poderiam ser encadeadas a estrutura do exemplo abaixo:

```

1  <%@page contentType="text/html" pageEncoding="UTF-8"%>
2  <!DOCTYPE html>
3  <html>
4  <head>
5      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
6      <title>Página JSP da 1ª Aula</title>
7  </head>
8  <body>
9      <p>
10         <%
11             int a,b;
12             String r ;
13             r = "";
14             a=2;
15             b=3;
16             if(a>b) {
17                 r = a+" é maior que "+b;
18             }else if(a<b) {
19                 r = b+" é maior que "+a;
20             }else{
21                 r = a+" é igual a "+b;
22             }
23             out.print(r);
24         %>
25     </p>
26 </body>
27 </html>

```

3 é maior que 2

10. Estrutura condicional ternário

A estrutura condicional ternário quase sempre resolve as mesmas questões da estrutura condicional if, mas de uma maneira mais simples, pois detém a seguinte sintaxe:

(teste lógico)? Instruções/comandos se o teste for true : Instruções/comandos se o teste for false;

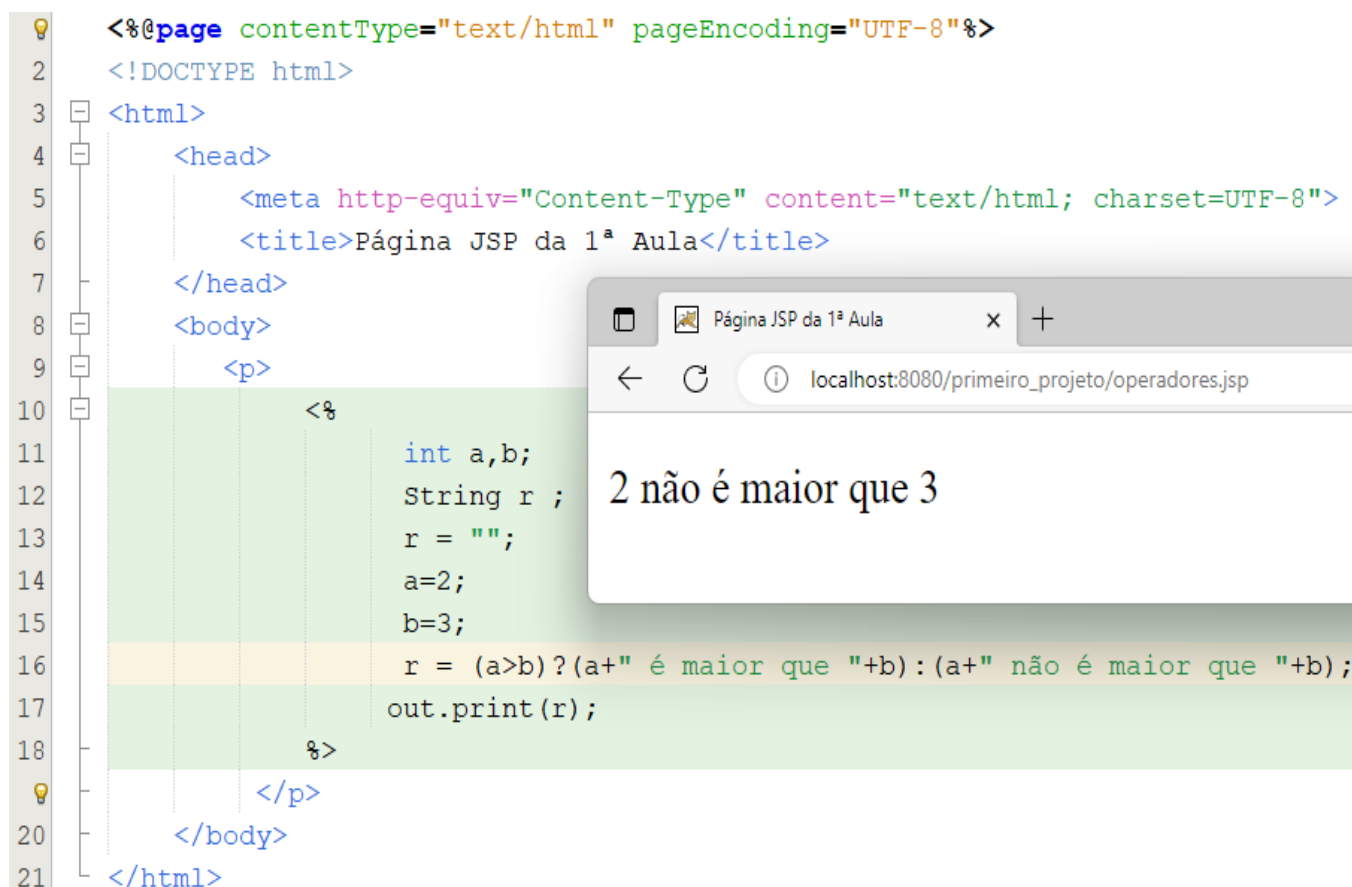
? é o if(se)

: é o else(senão)

Vale asseverar que uma variável poderá receber o resultado da estrutura ternário, ou poderá ser diretamente utilizado num comando de saída como print.

Vale ainda registrar que, caso o valor resultado da estrutura condicional ternário seja atribuído a uma variável, deve-se cuidar para que haja harmonia entre o tipo da variável que receberá o valor e os valores resultantes da condição true ou condição false;

Exemplo de uso da estrutura ternário:



```

1  <%@page contentType="text/html" pageEncoding="UTF-8"%>
2  <!DOCTYPE html>
3  <html>
4  <head>
5      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
6      <title>Página JSP da 1ª Aula</title>
7  </head>
8  <body>
9      <p>
10         <%
11             int a,b;
12             String r ;
13             r = "";
14             a=2;
15             b=3;
16             r = (a>b)?(a+" é maior que "+b):(a+" não é maior que "+b);
17             out.print(r);
18         %>
19     </p>
20 </body>
21 </html>

```

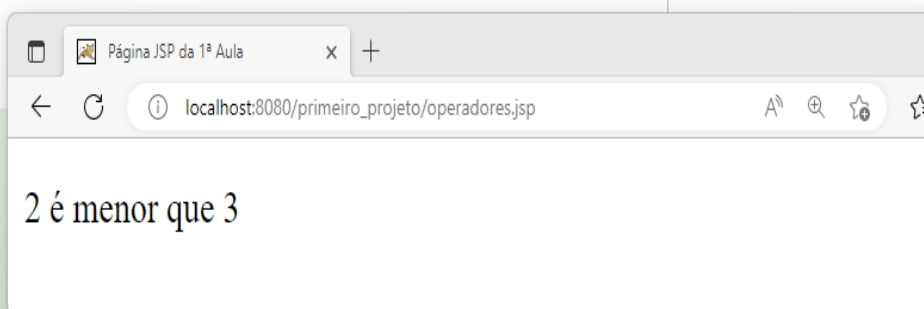
The browser window shows the output: 2 não é maior que 3

É possível encadear o ternário, realizando múltiplos testes lógicos para execução de códigos específicos, conforme exemplifica-se:

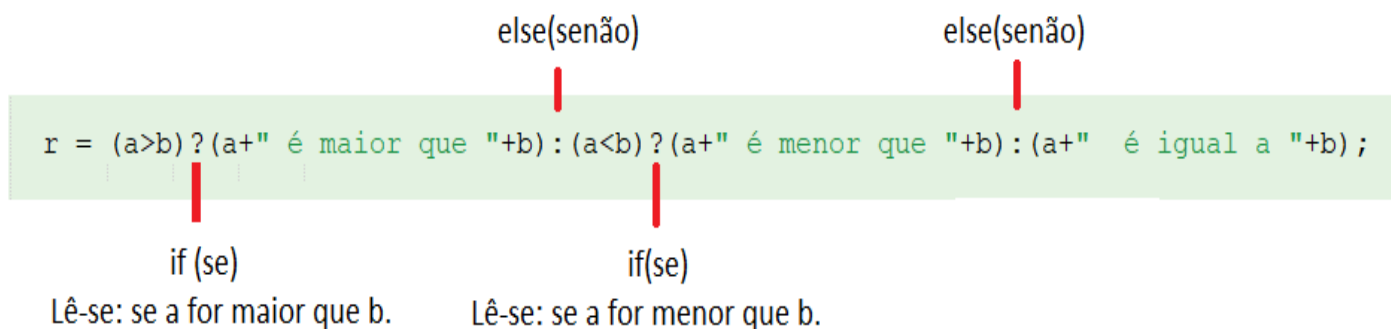
```

1  <%@page contentType="text/html" pageEncoding="UTF-8"%>
2  <!DOCTYPE html>
3  <html>
4  <head>
5      <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
6      <title>Página JSP da 1ª Aula</title>
7  </head>
8  <body>
9      <p>
10         <%
11             int a,b;
12             String r ;
13             r = "";
14             a=2;
15             b=3;
16             r = (a>b)?(a+" é maior que "+b):(a<b)?(a+" é menor que "+b):(a+" é igual a "+b);
17             out.print(r);
18         %>
19     </p>
20 </body>
21 </html>

```



Detalha-se abaixo a estrutura:



11. Estrutura condicional switch (case)

Como uma alternativa a estrutura condicional if...else, a estrutura condicional switch seleciona um dos muitos blocos de instruções/comandos no caso de correspondência da expressão com o caso, observados os seguintes passos:

- A switch expression é avaliada uma vez.
- O valor da expressão é comparado com os valores de cada um case.
- Se houver uma correspondência, o bloco de código associado será executado.

- As palavras-chave `break` (interrompe a execução da estrutura) e `default` (especifica algum código a ser executado se não houver correspondência) são opcionais.

Sintaxe genérica para a estrutura condicional `switch`:

Sintaxe

```
switch(expression) {
  case x:
    // code block
    break;
  case y:
    // code block
    break;
  default:
    // code block
}
```

No exemplo abaixo a variável `x` será testada na estrutura `switch`, possibilitando perceber que em função da falta de correspondência nos `cases`, a instrução da palavra-chave `default` será executada:

```
<p>
<%
int x = 5;
switch (x) {
  case 1:
    out.println("Feminino");
    break;
  case 2:
    out.println("Masculino");
    break;
  default:
    out.println("Não informado!");
}
%>
</p>
```

Não informado!

Já neste outro exemplo abaixo a variável x será testada na estrutura **switch**, possibilitando perceber que em função da correspondência no primeiro **case** e a falta da palavra-chave **break**, determinam a execução de todas as instruções:

The screenshot shows a JSP page with a switch statement. The code is as follows:

```
<%
int x = 1;
switch (x) {
    case 1:
        out.println("Feminino");

    case 2:
        out.println("Masculino");

    default:
        out.println("Não informado!");
}
%>
```

The browser window shows the output: "Feminino Masculino Não informado!". The URL is localhost:8080/primeiro_projeto/operadores.jsp.

Infer-se até o momento que as estruturas condicionais ou estruturas de tomadas de decisão são cruciais para a maioria esmagadora dos programas de computadores e na linguagem Java não é diferente. Importante mencionar que uma lista de comandos é um arquivo de lote e não um programa, logo a capacidade de alterar a execução de código de acordo com diferentes parâmetros e variáveis é o que faz com que um programa. As estruturas condicionais ou estruturas de tomada de decisão são os mecanismos capazes de alterar o processamento de dados de acordo com circunstâncias diferentes.

- ★ <http://www.deinf.ufma.br/~mario/grad/poo/aulas/Controle.pdf>
- ★ <http://www.nce.ufrj.br/ginape/js/conteudo/decisao/ternario.htm>
- ★ <http://www.universidadejava.com.br/java/java-if-else/>
- ★ <https://intranet.ifs.ifsuldeminas.edu.br/michelle.nery/Java%20-%20Subsequente/>