

Lab 07 - Expression Language

Neste laboratório faremos uso das **Expression Language** do **JSP 2.0**.

Exercícios

Exercício 1: Uso de diversos exemplos de **Expression Language**

Exercício 1 - Uso de diversos exemplos de Expression Language

1.1. Básico da Aritmética

1. Crie um **projeto Dynamic Web** chamado **ExpressionLanguage** e crie um arquivo **basico-aritmetica.jsp** com o código abaixo. Analise o código e veja que a página é escrita utilizando duas vezes o mesmo código para cada **<TD>**, mas a primeira utiliza o caractere de escape, então você terá na primeira coluna a exibição do código escrito e na segunda coluna o resultado do código utilizando **EL**.

```
<html>
<head>
<title>JSP 2.0 Expression Language - Aritmetica basica</title>
</head>
<body>
<h1>JSP 2.0 Expression Language - Aritmetica basica</h1>
<hr>
```

Este exemplo ilustra o basico da aritmetica usado na ExpressionLanguage. Adição(+), subtração(-), multiplicação (*), divisão (/ ou div), e módulo (% ou mod) são todos suportados. Erro de condições, como divisão por zero, são tratados graciosamente.

```
<br>
<blockquote>
<code>
<table border="1">
  <thead>
    <td><b>Expressão</b></td>
    <td><b>Resultado</b></td>
  </thead>
  <tr>
    <td>\${1}</td>
    <td>${1}</td>
  </tr>
  <tr>
    <td>\${1 + 2}</td>
    <td>${1 + 2}</td>
  </tr>
  <tr>
    <td>\${1.2 + 2.3}</td>
    <td>${1.2 + 2.3}</td>
  </tr>
  <tr>
    <td>\${1.2E4 + 1.4}</td>
    <td>${1.2E4 + 1.4}</td>
  </tr>
  <tr>
    <td>\${-4 - 2}</td>
    <td>${-4 - 2}</td>
  </tr>
  <tr>
```

```

        <td>\${21 * 2}</td>
        <td>${21 * 2}</td>
    </tr>
    <tr>
        <td>\${3/4}</td>
        <td>${3/4}</td>
    </tr>
    <tr>
        <td>\${3 div 4}</td>
        <td>${3 div 4}</td>
    </tr>
    <tr>
        <td>\${3/0}</td>
        <td>${3/0}</td>
    </tr>
    <tr>
        <td>\${10%4}</td>
        <td>${10%4}</td>
    </tr>
    <tr>
        <td>\${10 mod 4}</td>
        <td>${10 mod 4}</td>
    </tr>
    <tr>
        <td>\${(1==2) ? 3 : 4}</td>
        <td>${(1==2) ? 3 : 4}</td>
    </tr>
</table>

</code>
</blockquote>
</body>
</html>

```

2. Execute o arquivo **basico-aritmetica.jsp** e veja o resultado conforme imagem abaixo.



JSP 2.0 Expression Language - Aritmetica basica

Este exemplo ilustra o basico da aritmetica usado na ExpressionLanguage. Adição(+), subtração(-), multiplicação(*), divisão (/ ou div), e módulo (% ou mod) são todos suportados. Erro de condições, como divisão por zero, são tratados graciosamente.

Expressão	Resultado
<code>\${1}</code>	1
<code>\${1 + 2}</code>	3
<code>\${1.2 + 2.3}</code>	3.5
<code>\${1.2E4 + 1.4}</code>	12001.4
<code>\${-4 - 2}</code>	-6
<code>\${21 * 2}</code>	42
<code>\${3/4}</code>	0.75
<code>\${3 div 4}</code>	0.75
<code>\${3/0}</code>	Infinity
<code>\${10%4}</code>	2
<code>\${10 mod 4}</code>	2
<code>\${(1==2) ? 3 : 4}</code>	4

1.2. Comparações Básicas

1. Crie um arquivo **comparacoes-basicas.jsp** e insira o código abaixo. Este exercício irá mostrar que podemos utilizar os caracteres comuns de comparação ou palavras reservadas equivalentes do **JSTL**.

```
<html>
<head>
<title>JSP 2.0 Expression Language - Comparações Básicas</title>
</head>
<body>
```

Esse exemplo ilustra as Comparações Básicas da Expression Language. Os seguintes operadores de comparação são suportados:

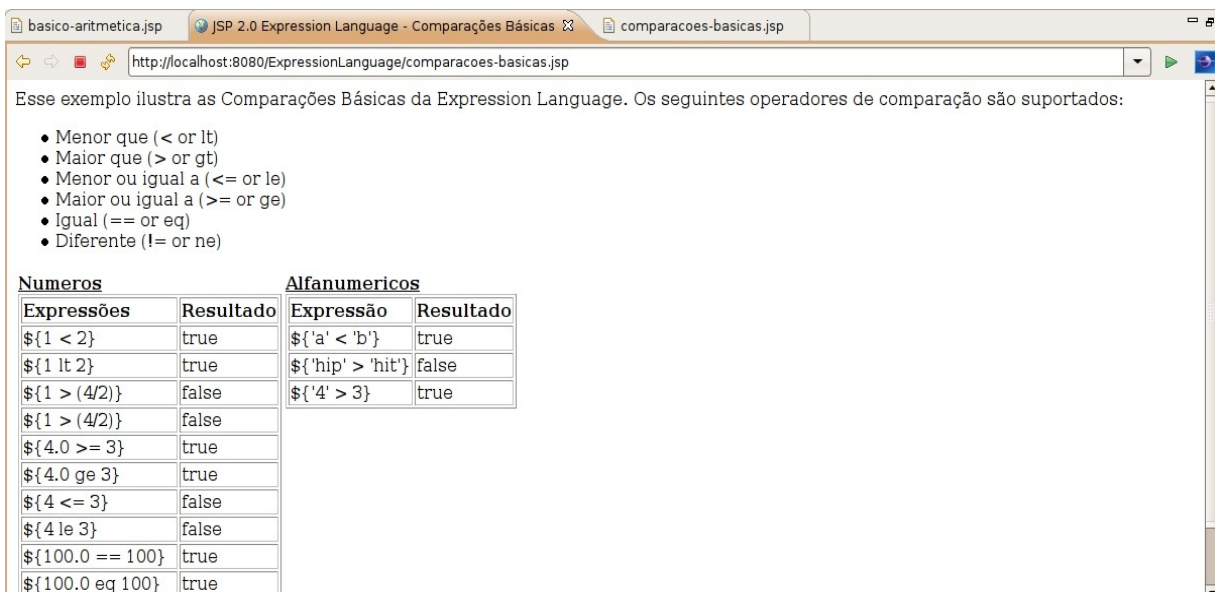
```
<ul>
  <li>Menor que (< or lt)</li>
  <li>Maior que (> or gt)</li>
  <li>Menor ou igual a (<= or le)</li>
  <li>Maior ou igual a (>= or ge)</li>
  <li>Igual (== or eq)</li>
  <li>Diferente (!= or ne)</li>
</ul>
<table border="0">
  <tr>
    <td valign="top"><u><b>Numeros</b></u> <code>
      <table border="1">
        <thead>
          <td><b>Expressão</b></td>
          <td><b>Resultado</b></td>
        </thead>
        <tr>
          <td>\${1 < 2}</td>
          <td>\${1 < 2}</td>
        </tr>
        <tr>
          <td>\${1 < 2}</td>
          <td>\${1 < 2}</td>
        </tr>
        <tr>
          <td>\${1 > (4/2)}</td>
          <td>\${1 > (4/2)}</td>
        </tr>
        <tr>
          <td>\${1 > (4/2)}</td>
          <td>\${1 > (4/2)}</td>
        </tr>
        <tr>
          <td>\${4.0 >= 3}</td>
          <td>\${4.0 >= 3}</td>
        </tr>
        <tr>
          <td>\${4.0 >= 3}</td>
          <td>\${4.0 >= 3}</td>
        </tr>
        <tr>
          <td>\${4 <= 3}</td>
          <td>\${4 <= 3}</td>
        </tr>
        <tr>
          <td>\${4 <= 3}</td>
          <td>\${4 <= 3}</td>
        </tr>
        <tr>
          <td>\${100.0 == 100}</td>
```

```

        <td>${100.0 == 100}</td>
    </tr>
    <tr>
        <td>\${100.0 eq 100}</td>
        <td>${100.0 eq 100}</td>
    </tr>
    <tr>
        <td>\${(10*10) != 100}</td>
        <td>${(10*10) != 100}</td>
    </tr>
    <tr>
        <td>\${(10*10) ne 100}</td>
        <td>${(10*10) ne 100}</td>
    </tr>
</table>
</code></td>
<td valign="top"><u><b>Alfanumericos</b></u> <code>
<table border="1">
    <thead>
        <td><b>Expressão</b></td>
        <td><b>Resultado</b></td>
    </thead>
    <tr>
        <td>\${'a' &lt; 'b'}</td>
        <td>${'a' < 'b'}</td>
    </tr>
    <tr>
        <td>\${'hip' &gt; 'hit'}</td>
        <td>${'hip' > 'hit'}</td>
    </tr>
    <tr>
        <td>\${'4' &gt; 3}</td>
        <td>${'4' > 3}</td>
    </tr>
</table>
</code></td>
</tr>
</table>
</body>
</html>

```

2. Execute o arquivo **comparacoes-basicas.jsp** e veja o resultado conforme imagem abaixo.



Esse exemplo ilustra as Comparações Básicas da Expression Language. Os seguintes operadores de comparação são suportados:

- Menor que (< or lt)
- Maior que (> or gt)
- Menor ou igual a (<= or le)
- Maior ou igual a (>= or ge)
- Igual (== or eq)
- Diferente (!= or ne)

Numeros		Alfanumericos	
Expressões	Resultado	Expressão	Resultado
\${1 < 2}	true	\${'a' < 'b'}	true
\${1 lt 2}	true	\${'hip' > 'hit'}	false
\${1 > (4/2)}	false	\${'4' > 3}	true
\${1 > (4/2)}	false		
\${4.0 >= 3}	true		
\${4.0 ge 3}	true		
\${4 <= 3}	false		
\${4 le 3}	false		
\${100.0 == 100}	true		
\${100.0 eq 100}	true		

1.3. Objetos implícitos

1. Crie um arquivo **objetos-implicitos.jsp** e adicione o código abaixo. Analise o código e veja as várias formas que podemos utilizar os **objetos implícitos** dentro do **JSTL** para recuperar valores de variáveis.

```
<html>
<head>
<title>JSP 2.0 Expression Language (EL) - Objetos Implícitos</title>
</head>
<body>
Este exemplo ilustra alguns dos objetos implícitos disponíveis na
Expression Language. Os seguintes objetos estão implicitamente
disponíveis (não todos ilustrados aqui):
<table cellpadding="0" cellspacing="0" border="1"
    style="font-size: 12px; font-family: Verdana;" align="center">
<tr style="font-weight: bold;" align="center" bgcolor="#CCCCCC">
    <td>EL Map</td>
    <td>Mapeia nome de</td>
    <td>Para seu</td>
    <td>Corresponde na API</td>
</tr>
<tr>
    <td>pageScope</td>
    <td>Atributo no escopo da página</td>
    <td>Valor</td>
    <td>PageContext.getAttribute(String nome)</td>
</tr>
<tr>
    <td>requestScope</td>
    <td>Atributo no escopo da requisição</td>
    <td>Valor</td>
    <td>ServletRequest.getAttribute(String nome)</td>
</tr>
<tr>
    <td>sessionScope</td>
    <td>Atributo no escopo da sessão</td>
    <td>Valor</td>
    <td>HttpSession.getAttribute(String nome)</td>
</tr>
<tr>
    <td>applicationScope</td>
    <td>Atributo no escopo da Aplicação</td>
    <td>Valor</td>
    <td>ServletContext.getAttribute(String nome)</td>
</tr>
<tr>
    <td>param</td>
    <td>Parâmetro da Requisição</td>
    <td>Valor único String</td>
    <td>ServletRequest.getParameter(String nome)</td>
</tr>
<tr>
    <td>paramValues</td>
    <td>Parâmetro da Requisição</td>
    <td>String[] de Valores</td>
    <td>ServletRequest.getParameterValues(String nome)</td>
</tr>
<tr>
    <td>header</td>
    <td>Cabeçalho da Requisição</td>
```

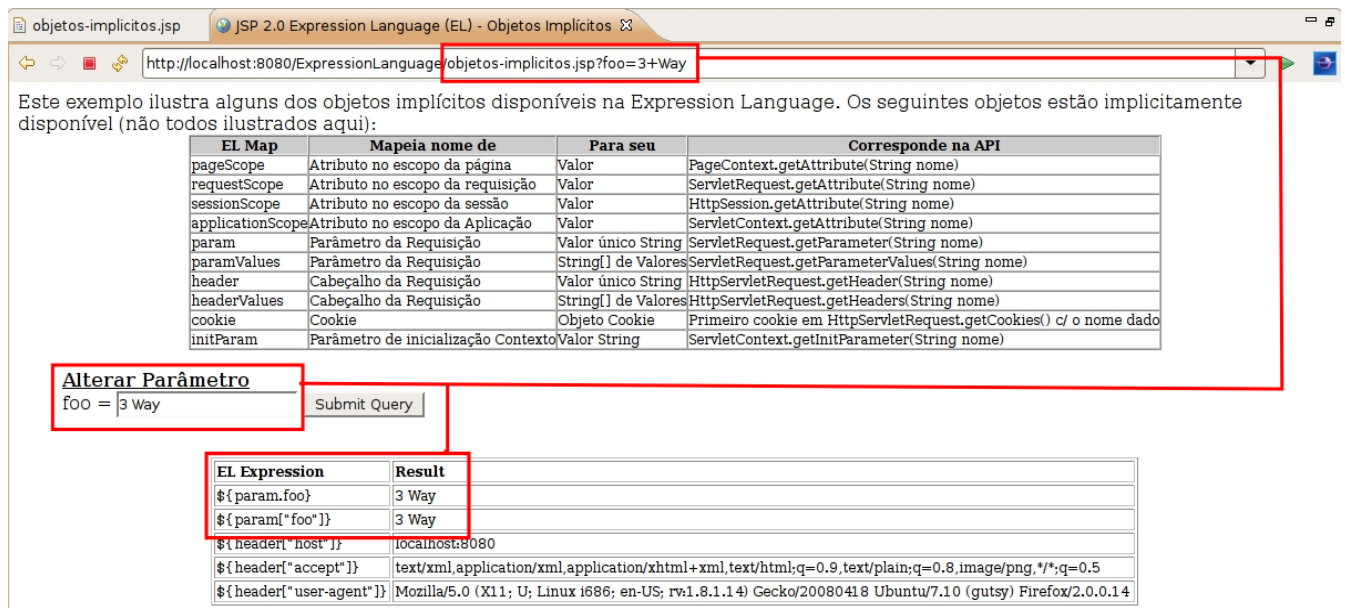
```

        <td>Valor único String</td>
        <td>HttpServletRequest.getHeader(String nome)</td>
    </tr>
    <tr>
        <td>headerValues</td>
        <td>Cabeçalho da Requisição</td>
        <td>String[] de Valores</td>
        <td>HttpServletRequest.getHeaders(String nome)</td>
    </tr>
    <tr>
        <td>cookie</td>
        <td>Cookie</td>
        <td>Objeto Cookie</td>
        <td>Primeiro cookie em HttpServletRequest.getCookies() c/ o nome
        dado</td>
    </tr>
    <tr>
        <td>initParam</td>
        <td>Parâmetro de inicialização Contexto</td>
        <td>Valor String</td>
        <td>ServletContext.getInitParameter(String nome)</td>
    </tr>
</table>

<blockquote><u><b>Alterar Parâmetro</b></u>
<form action="objetos-implicitos.jsp" method="GET">foo = <input
    type="text" name="foo" value="\${param["foo"]}"> <input
    type="submit"></form>
<br>
<code>
<table border="1" style="font-size: 12px; font-family: Verdana;"
    align="center">
    <thead>
        <td><b>EL Expression</b></td>
        <td><b>Result</b></td>
    </thead>
    <tr>
        <td>\${param.foo}</td>
        <td>\${param["foo"]} &nbsp;</td>
    </tr>
    <tr>
        <td>\${param["foo"]}</td>
        <td>\${param["foo"]} &nbsp;</td>
    </tr>
    <tr>
        <td>\${header["host"]}</td>
        <td>\${header["host"]} &nbsp;</td>
    </tr>
    <tr>
        <td>\${header["accept"]}</td>
        <td>\${header["accept"]} &nbsp;</td>
    </tr>
    <tr>
        <td>\${header["user-agent"]}</td>
        <td>\${header["user-agent"]} &nbsp;</td>
    </tr>
</table>
</code></blockquote>
</body>
</html>

```

2. Execute o projeto e acesse a página adicionando algum parâmetro ao campo *Foo* conforme imagem abaixo e veja o resultado.



Este exemplo ilustra alguns dos objetos implícitos disponíveis na Expression Language. Os seguintes objetos estão implicitamente disponíveis (não todos ilustrados aqui):

EL Map	Mapeia nome de	Para seu	Corresponde na API
pageScope	Atributo no escopo da página	Valor	PageContext.getAttribute(String nome)
requestScope	Atributo no escopo da requisição	Valor	ServletRequest.getAttribute(String nome)
sessionScope	Atributo no escopo da sessão	Valor	HttpSession.getAttribute(String nome)
applicationScope	Atributo no escopo da Aplicação	Valor	ServletContext.getAttribute(String nome)
param	Parâmetro da Requisição	Valor único String	ServletRequest.getParameter(String nome)
paramValues	Parâmetro da Requisição	String[] de Valores	ServletRequest.getParameterValues(String nome)
header	Cabeçalho da Requisição	Valor único String	HttpServletRequest.getHeader(String nome)
headerValues	Cabeçalho da Requisição	String[] de Valores	HttpServletRequest.getHeaders(String nome)
cookie	Cookie	Objeto Cookie	Primeiro cookie em HttpServletRequest.getCookies() c/ o nome dado
initParam	Parâmetro de inicialização Contexto	Valor String	ServletContext.getInitParameter(String nome)

Alterar Parâmetro
foo = 3 Way

EL Expression	Result
<code>\${param.foo}</code>	3 Way
<code>\${param["foo"]}</code>	3 Way
<code>\${header["host"]}</code>	localhost:8080
<code>\${header["accept"]}</code>	text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5
<code>\${header["user-agent"]}</code>	Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.8.1.14) Gecko/20080418 Ubuntu/7.10 (gutsy) Firefox/2.0.0.14