



*Open-source Education* →

# JavaServer Pages 2.0

Iniciativa Globalcode

## Palestrante

---

- Kleber Xavier
  - Formado em Engenharia Mecatrônica pela POLI-USP e Mestrando em Engenharia de Software pelo IME-USP
  - Experiência Internacional em projetos em Cingapura e Equador
  - **Certificações:** Java Programmer, Web Component Developer e Business Component Developer.
  - Instrutor Globalcode;

## Agenda

- **Novidades do JSP 2.0**
- **Expression Language**
- **JavaServer Pages Standard Tag Library**
- **Tag File**

## Agenda

- **Novidades do JSP 2.0**
- **Expression Language**
- **JavaServer Pages Standard Tag Library**
- **Tag File**

## Novidades do JSP 2.0

Entre as principais novidades encontram-se:

- Expression Language (EL);
- JSTL;
- Tag Files: tags customizados a partir de fragmentos de página JSP

## Agenda

- **Novidades do JSP 2.0**
- **Expression Language**
- **JavaServer Pages Standard Tag Library**
- **Tag File**

# Expression Language

## Objetivos:

- Simplificar o desenvolvimento de JavaServer Pages
  - Minimizar o uso de scriptlets `<% %>` e expressions `<%= %>`
  - Simplificar o acesso a variáveis de sessão, contexto, request e parâmetros.

# Expression Language

## Sintaxe:

- `${ EXPRESSÃO }`
- As expressões podem conter literais, operadores e variáveis.

## Exemplos:

`${usuario.nome}`

`${saldo - valor }`

`${1000 - 500}`



# Expression Language

## **Operadores:**

- A EL (Expression Language) oferece versões texto para a maioria dos operadores da linguagem Java.

## Expression Language

Operador	Descrição
+	Adição
-	Subtração
*	Multiplicação
/ ou div	Divisão
% ou mod	Resto da divisão
== ou eq	Igualdade
!= ou ne	Desigualdade
< ou lt	Menor

## Expression Language

Operador	Descrição
> ou gt	Maior
<= ou le	Menor ou igual
>= ou ge	Maior ou igual
&& ou and	E lógico
ou or	OU lógico
! ou not	Negação lógica
a ? b : c	Operador condicional
empty	Testa valor vazio ou null

# Expression Language

**Demo: exemploEL.jsp**

# Expression Language

## **Variáveis:**

- Variáveis são acessadas diretamente (pelo nome) na EL, podendo representar parâmetros de formulários, ou variáveis que se encontram nos escopos válidos, como HttpSession, ServletContext, HttpServletRequest.

## Expression Language

**Exemplo de leitura de parâmetro de formulário:**

**Trecho de HTML:**

```
<input type="text" name="txtNome">
```

**Leitura via EL:**

```
Bem-vindo ${param.txtNome}
```



## Expression Language

**Exemplo de leitura de atributo da sessão:**

**Trecho de um Servlet:**

```
session.setAttribute("dataAtual", new Date());
```

**Leitura via EL:**

A data atual é `${dataAtual}`



# Expression Language

## Navegação em atributos

- A EL facilita o acesso a atributos através dos operadores `.` e `[]`. Cada vez que um destes operadores é encontrado é feito o acesso ao atributo solicitado utilizando a convenção de getters do Java



# Expression Language

## **Exemplo sem utilizar EL:**

- `usuario.getEndereco().getRua();`

## **Utilizando EL temos o seguinte:**

- `${ usuario.endereco.rua}` ou
- `${usuario.endereco["rua"]}`

## Expression Language

Uma facilidade disponibilizada pelos operadores de acesso e por EL em geral é a robustez na **tolerância** a operações com **variáveis nulas**.

# Expression Language

## Navegação em arrays e Lists

- Utilizamos os operadores [ ] da mesma forma que utilizamos para arrays em Java.

# Expression Language

**Exemplo sem utilizar EL:**

- `usuarios[0].getNome()`

**Utilizando EL temos o seguinte:**

- `${ usuarios[0].nome }`

## Expression Language

### **Criação do List e armazenamento na request**

```
<%  
    List lista = new ArrayList( );  
    lista.add("elemento ");  
    request.setAttribute("lista",lista);  
%>
```

### **Acesso ao ArrayList utilizando EL:**

- `${lista[0]}`

# Expression Language

## **Objetos implícitos:**

- Da mesma forma que os JSPs, a EL oferece uma série de objetos implícitos criados automaticamente pelo container.
- Vejamos uma tabela com os principais objetos implícitos.

## Expression Language

<b>Objeto Implícito</b>	<b>Conteúdo</b>
applicationScope	Conjunto de variáveis de ServletContext
sessionScope	Conjunto de variáveis de HttpSession
requestScope	Conjunto de variáveis de ServletRequest
pageScope	Conjunto de variáveis de PageContext
pageContext	Objeto PageContext
param	Lista de parâmetros obtidos por: ServletRequest.getParameter( )
cookie	Lista de cookies obtidos do request
initParam	Conjunto de parâmetros de inicialização obtidos do ServletContext

# Expression Language

## **Desabilitando Scriptlets:**

- É possível desabilitar o uso de Scriptlets na aplicação através de configurações no deployment descriptor.



## Expression Language

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://java.sun.com/xml/ns/j2ee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-
    instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns
    /j2ee http://java.sun.com/xml/ns/j2ee/web-
    app_2_4.xsd" version="2.4">
  <jsp-config>
    <jsp-property-group>
      <url-pattern>*.jsp</url-pattern>
      <scripting-invalid>true</scripting-
invalid>
    </jsp-property-group>
  </jsp-config>
</web-app>
```

# Expression Language

## **Desabilitando EL:**

- É possível desabilitar o uso de EL, e caso seja utilizado EL no JSP o código será interpretado como texto.

## Expression Language

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns="http://java.sun.com/xml/ns/j2ee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-
    instance"

  xsi:schemaLocation="http://java.sun.com/xml/ns
    /j2ee http://java.sun.com/xml/ns/j2ee/web-
    app_2_4.xsd" version="2.4">
  <jsp-config>
    <jsp-property-group>
      <url-pattern>*.jsp</url-pattern>
      <el-ignored>true</el-ignored>
    </jsp-property-group>
  </jsp-config>
</web-app>
```

## Agenda

- **Novidades do JSP 2.0**
- **Expression Language**
- **JavaServer Pages Standard Tag Library**
- **Tag File**

# Java Standard Tag Library

## O que é JSTL?

- JSTL = JavaServer Pages Standard Tag Library

Uma biblioteca de tags criada pelo grupo Jakarta e incorporada ao JSP 2.0, mas não obrigatório que todos os containers J2EE a tragam embutida.

## Java Standard Tag Library

### **Download:**

- [www.jakarta.apache.org](http://www.jakarta.apache.org)

A biblioteca JSTL pode ser encontrada pelo nome “standard”.

Além das bibliotecas taglib.jar e standard.jar no arquivo carregado existem vários exemplos de utilização de JSTL.

## Java Standard Tag Library

### Instalação em um servidor J2EE :

- Basta colocar as bibliotecas **taglib.jar** e **standard.jar** no diretório **WEB-INF/lib** da nossa aplicação.

# Java Standard Tag Library

## **Como utilizar uma JSTL em uma página JSP**

- O uso da JSTL é idêntico ao uso de qualquer biblioteca de tags customizados em uma página JSP. Ou seja, devemos utilizar a diretiva taglib para configurar sua utilização.



## Java Standard Tag Library

Biblioteca	URI	Prefix
Core	<a href="http://java.sun.com/jsp/jstl/core">http://java.sun.com/jsp/jstl/core</a>	c
Processamento de XML	<a href="http://java.sun.com/jsp/jstl/xml">http://java.sun.com/jsp/jstl/xml</a>	x
Internacionalização	<a href="http://java.sun.com/jsp/jstl/fmt">http://java.sun.com/jsp/jstl/fmt</a>	fmt
Acesso a banco de dados	<a href="http://java.sun.com/jsp/jstl/sql">http://java.sun.com/jsp/jstl/sql</a>	sql
Funções	<a href="http://java.sun.com/jsp/jstl/functions">http://java.sun.com/jsp/jstl/functions</a>	fn

# Java Standard Tag Library

## Utilização de JSTL em uma página JSP

```
<%@taglib uri=http://java.sun.com/jsp/jstl/core prefix="c"  
    %>
```

```
<%@taglib uri=http://java.sun.com/jsp/jstl/fmt prefix="fmt"  
    %>
```

# Java Standard Tag Library

## Principais bibliotecas JSTL

- **Biblioteca Core:** tags de uso geral como controle de fluxo, manipulação de variáveis, etc.
- **Biblioteca de Internacionalização:** tags para formatação de números e datas conforme a região e para configuração do idioma utilizado para mensagens.

# Java Standard Tag Library

## **Biblioteca Core:**

- Suporte a variáveis
- Controle de Fluxo
- Manipulação de URLs
- Outros Tags

## Java Standard Tag Library

**Suporte a variáveis: <c:set>**

**Exemplo utilizando JSTL:**

```
<c:set var="nome" value="Globalcode"  
      scope="session" />
```

**Sem utilizar JSTL:**

```
<%  
    session.setAttribute("nome", "Globalcode");  
%>
```

## Java Standard Tag Library

**Suporte a variáveis:** `<c:remove>`

**Exemplo utilizando JSTL:**

```
<c:remove var="nome" scope="session" />
```

**Sem utilizar JSTL:**

```
<%  
    session.removeAttribute("nome");  
%>
```

# Java Standard Tag Library

## **Biblioteca Core: Controle de Fluxo**

- `<c:if>`
- `<c:choose>`
- `<c:forEach>`
- `<c:forTokens>`

# Java Standard Tag Library

## **Controle de Fluxo: <c:if>**

### **Exemplo :**

```
<c:if test="${contador == 0} >  
    Este é o seu primeiro acesso !!  
</c:if>
```

**ou**

```
<c:if test="${contador == 0}  
    var="primeiroAcesso" />
```



# Java Standard Tag Library

## **Controle de Fluxo: <c:choose>**

### **Exemplo :**

```
<c:choose>
  <c:when test="{contador == 0}" >
    Este é o seu primeiro acesso !!
  </c:when>
  <c:when test="{contador == 1}" >
    Você já acessou o sistema 1 vez !
  </c:when>
  <c:otherwise>
    Você já acessou o sistema {contador} vezes!
  </c:otherwise>
</c:choose>
```

# Java Standard Tag Library

## **Controle de Fluxo: <c:forEach>**

### **Exemplo :**

```
<table>
<c:forEach var="produto" items="${produtos}"
  varStatus="contador">
  <tr><td>${contador.count} -
  ${produto}</td></tr>
</c:forEach>
</table>
```

## Java Standard Tag Library

**Controle de Fluxo:** `<c:forTokens>`

**Exemplo :**

```
<c:forTokens var="elemento" items="um,dois,tres"
              delims=", ">
```

```
  <c:out value="${elemento}" /><br>
```

```
</c:forTokens>
```

# Java Standard Tag Library

## **Biblioteca Core: Manipulação de URLs**

- `<c:import>`
- `<c:redirect>`
- `<c:url>`

## Java Standard Tag Library

**Controle de Fluxo:** `<c:import>`

**Exemplo :**

```
<c:import  
url="http://www.globalcode.com.br/controller?  
command=load" />
```

# Java Standard Tag Library

## Controle de Fluxo: <c.redirect>

## Exemplo :

```
<c:redirect
  url="http://www.globalcode.com.br/controller?
                                command=load" />
```

## Java Standard Tag Library

**Controle de Fluxo:** <c:url>

**Exemplo :**

```
<c:url value="http://www.globalcode.com.br/  
  controler" var="destino" >  
  <c:param name="command" value="load" />  
</c:url>  
  
<a href='${destino}'>clique aqui </a>
```

A diagram consisting of a thin black arrow. The arrow starts at the word 'destino' inside the 'var' attribute of the first <c:url> tag. It points downwards and to the right, ending at the word 'destino' inside the 'href' attribute of the <a> tag. This illustrates how the variable 'destino' is passed from the URL tag to the link tag.

# Java Standard Tag Library

## **Biblioteca de Internacionalização:**

- Locale
- Mensagens
- Formatação de Números e Datas



## Agenda

- **Novidades do JSP 2.0**
- **Expression Language**
- **JavaServer Pages Standard Tag Library**
- **Tag File**

## Tag File

---

### O que são Tag Files

- Um tipo de Custom Tag com maior facilidade de criação podendo ser criadas por autores de páginas JSP.
- Ao invés de criar uma classe customizada para implementação da interface Tag **basta salvar um trecho de um JSP em um arquivo com a extensão .tag**
- Não requer a criação de um TLD (Tag Library Descriptor)

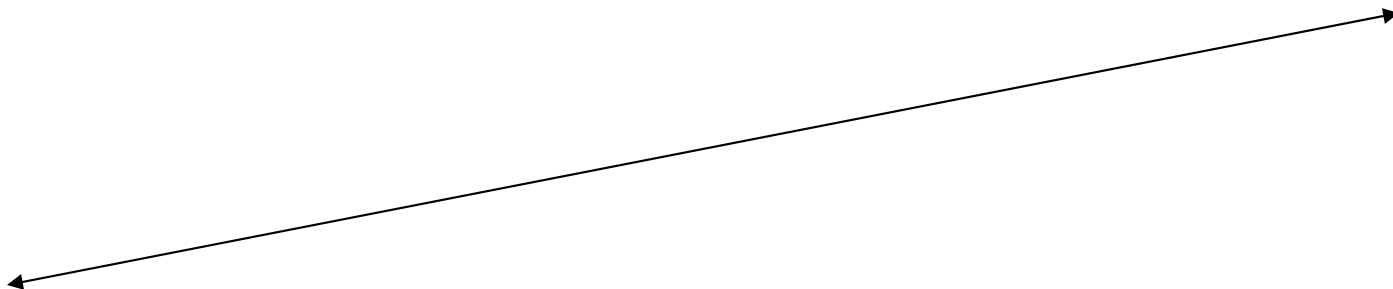
# Tag File

---

## Utilização de Tag Files em páginas JSP

```
<@taglib tagdir ="/WEB-INF/tags/" prefix="gc"%>
```

```
<gc:exemplo/>
```



## Tag File

---

### Demo: Tag Files