

# DESENVOLVIMENTO DE APLICAÇÕES PARA INTERNET

## Parte V

Prof. Msc. Felipe Diniz Dallilo



# Códigos dentro do html

- Até o momento foi visto dentro do HTML:
  - Códigos CSS através da tag <Style>
  - Códigos Javascript através da tag <Script>

E agora veremos:

- **Java !**

# Jsp

- Como visto anteriormente :
  - Acrônimo de Java Server Pages
  - É um Servlet (são traduzidas em servlets em tempo de execução)
  - Exige um servidor web compatível com um container servlet
  - Permite intercalação entre o conteúdo de marcação estático e o código Java

# Jsp - Scriptlet

- As páginas desenvolvidas em JSP utilizam diversos delimitadores com funções de script.
- O delimitador mais básico é o **scriptlet**, onde é sinalizado dentro do JSP que um bloco de código Java será executado ao solicitar uma página.
- **A tag esperada (que indica um scriptlet) é:**
  - `<% ... %>`
  - `<% String a = "abc "; %>`

# Jsp – Expressões

- O Script é utilizado para o processamento da informação (com o código Java).
  - Para a apresentação desta informação, é utilizado outro delimitador, conhecido como **expressões**. Abaixo segue a sintaxe:
- 
- `<%= ... %>`
  - `<%= a %>`
  - Uma característica importante da Expressão é a ausência da utilização do carácter “;”.



# Como o JSP se comporta com o html?

- Código HTML:

```
<html>
```

```
<body>
```

```
  Olá mundo!
```

```
</body>
```

```
</html>
```

Código JSP:

```
<html>
```

```
<body>
```

```
  <% String a = "Olá mundo!" %>
```

```
  <%= a %>
```

```
</body>
```

```
</html>
```

# Como o JSP se comporta com o html?

- Código HTML:

```
<html>
<body>
  Contando de 1 a 1000 ! <br>
  Este número é : 1 <br>
  Este número é : 2 <br>
  Este número é : 3 <br>
  ....
  1000 <br>
  Contagem finalizada !
</body>
</html>
```

Código JSP:

```
<html>
<body>
  <% for (int i=1; i<=1000; i++) { %>
    <p>Este número é : <%= i %>.</p>
  <% } %>
</body>
</html>
```

# Recuperando parâmetros

```
<html>
  <body>
    <form action="teste.jsp" method="GET">
      Digite um valor: <input type="text" name="valor"/>
      <input type="submit" value="Enviar">
    </form>

    <%
      String res = request.getParameter("valor");
      out.print(res);
    %>
  </body>
< html >
```



# JSP - Diretivas

- São instruções processadas quando a página JSP é compilada em Servlet:

**Include** - Adiciona um arquivo estático (outro JSP).

```
<%@include file="page.jsp"%>
```

**Page** – Define atributos que são aplicados a todo o arquivo JSP.

```
<%@ page import="java.util.Date" %>
```

```
<%@ page extends="com.altieres.ocewcd.MyClass"%>
```

**Taglib** – Define uma tag library e seu prefixo a ser utilizado na página JSP (falaremos mais no futuro).

Também há as tags do próprio JSP (JSTL – JSP Standard Tag Library) (que também detalharemos mais no futuro).

# JSP - Declarações

- São similares com as declarações de variáveis em Java e definem variáveis para uso posterior em **expressões ou scriptlets**. São definidas entre `<%! E %>`
- Sintaxe:
- `<%! int i = 0; String a = ""; %>`

# Diferença entre Declaração e Scriptlet

- Ao se criar uma declaração: todas as instâncias abertas visualizam a mesma variável, ao modificar o valor de uma variável de instância, você estará modificando o valor de todas as outras instâncias.
- Ao utilizar a Tag Scriptlet: Cria variáveis independentes para cada instância
- \* Não confunda com as variáveis estáticas que são usadas para declarar variáveis de classes, e são compartilhadas entre todas as instâncias das classes. Podem ser criadas assim:
- `<%! static public int counter=0; %>`

# Comentário no JSP

- Comentários HTML e JSP:

- ✓ `<!-- comment -->`

- ✓ `<%-- comment --%>`

- ✓ `<% /* comment */ %>`

# Servlet ou JSP?

Para entender melhor qual é o mais adequado, algumas perguntas devem ser feitas ao utilizar o Servlet :

- ✓ Todo o HTML em uma “String” no Servlet é a melhor solução ?
- ✓ E se houver CSS e Javascript incluídos, como lidar ?
- ✓ Aspas dúplas dentro de String ?
- ✓ Manutenção de Strings ?



# Servlet ou JSP?

- Também devem ser feitas algumas perguntas ao utilizar o JSP:
  - ✓ Muitos scriptlets na página ?
  - ✓ Desenvolvedor e Designer trabalhando no mesmo arquivo ?
  - ✓ Lógica Java, Javascript, CSS e HTML na mesma página ?

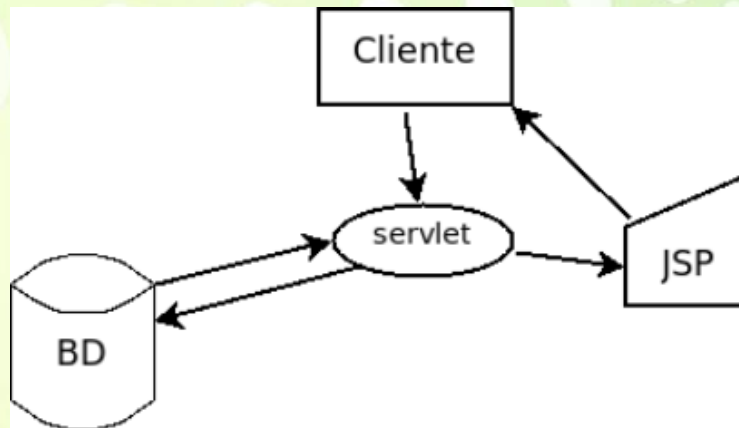
# Servlet ou JSP ?

- Por quê não utilizar o melhor dos dois ?



# Servlet e jsp

- O JSP foi feito apenas para apresentar o resultado do servidor, não deve fazer acessos a banco de dados e nem fazer a instanciação de objetos. Estas responsabilidades devem ser atribuídas ao Servlet!



# Prática

Crie o seguinte JSP:

- `<html>`
- `<body>`
- `Meu JSP`
- `</body>`
- `</html>`

Crie um Servlet e no método GET, adicione:

```
RequestDispatcher rd = request  
    .getRequestDispatcher("/nome-do-jsp.jsp");  
rd.forward(request,response);
```

# Referências

- Deitel, Paul J.; Deitel, Harvey M. Java: como programar - 8ª edição. Editora Pearson. ISBN: 9788576055631
- Barnes, David.; Kölling, Michael. Programação Orientada a Objetos com Java: uma introdução prática usando o Blue J - 1ª edição. 2013. Editora Pearson. ISBN: 9788576050124
- Puga, Sandra; Rissetti, Gerson. Lógica de Programação e Estruturas de Dados: com aplicações em Java. 1ª edição. 2013. Editora Pearson. ISBN: 9788587918826
- Lemay, Laura; Colburn, Rafe; Tyler, Denise. Aprenda a Criar Páginas Web com HTML e XHTML em 21 Dias. Editora Pearson. 2013. 1ª Edição. ISBN: 9788534614283
- Chak, Andrew. Como Criar Sites Persuasivos. Editora Pearson. 2012. 1ª Edição. ISBN: 9788534615112
- Fábio Flatschart; Clécio Bachini; Cesar Cusin. Open Web Platform. Editora Pearson. 2013. 1ª Edição. ISBN: 9788574526140