

FIAP GRADUAÇÃO

# DESENVOLVIMENTO DE SISTEMAS WEB

PROF. RAFAEL MATSUYAMA  
profrafael.matsuyama@fiap.com.br

JSP

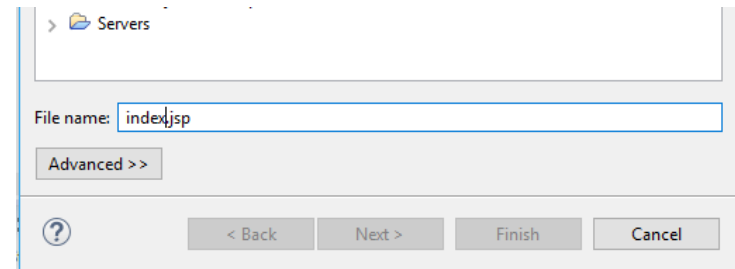
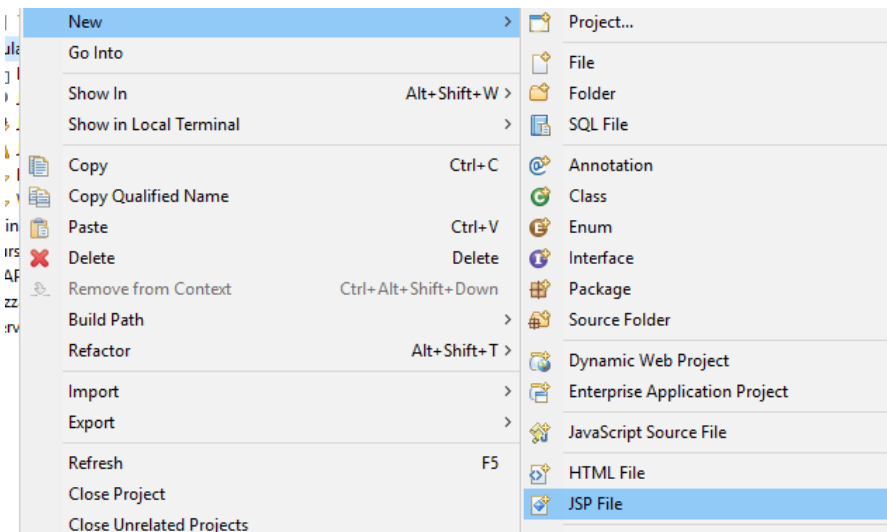
- Introdução
- Elementos JSP
  - Scriptlets
  - Expressions
  - Declarations
- Comentários
- Diretivas
  - Include
  - Taglib
  - Page
  - Import
- Objetos Implícitos

- Assim como as servlets, o JSP é utilizado para construção de páginas WEB. As páginas JSP podem ser vistas como uma extensão das funcionalidades da servlet, uma vez que também pode executar os recursos java.
- Uma página JSP consiste em Tags HTML e Tags JSP, ficando mais fácil separar a parte de desenvolvimento da parte de apresentação. Usando apenas Servlet precisamos misturar os códigos para gerar as páginas.
- O desenvolvimento é mais rápido, não há necessidade de recompilar e reimplementar.

# ❖ CRIANDO UMA PÁGINA JSP

Para criar nosso projeto vá em **File -> New -> Dynamic Web Project**, digite o nome do projeto **“Aula-JSP”** e se o TOMCAT já estiver configurado clique em **NEXT, NEXT**, marque o checkbox **para criação do arquivo web.xml** e clique em **FINISH**.

Agora clique com o botão direito no projeto **new -> JSP File**, dê o nome de **index.jsp** e **FINISH**.



# PRIMEIRA PÁGINA JSP

Vamos criar nossa primeira página JSP, digite o código abaixo.

```
9 <body>
10     <h1>Aula de JSP</h1>
11
12     <ul>
13         <% for (int i = 0; i < 10; i++) { %>
14             <li><% out.print(i);%></li>
15             <% } %>
16     </ul>
17     <p> <% out.print("primeira página JSP!"); %> </p>
18 </body>
19 </html>
```

- Execute o JSP no servidor, olhe no browser o código fonte da página...

# PASSANDO PARÂMETROS COM JSP

Agora vamos passar informações por parâmetro para outra página. Primeiro, monte o formulário abaixo na página index.jsp:

```
7 <title>Insert title here</title>
8 </head>
9 <body>
10 <form action="resposta.jsp" method="get">
11 <label for="idNome">Nome:</label>
12 <input type="text" name="nome" id="idNome">
13 <input type="submit" value="Enviar">
14 </form>
15 </body>
16 </html>
```

- Repare que no atributo action do formulário, indicamos que o formulário será enviado para a página “resposta.jsp”.

Nosso próximo passo será criar a página resposta.jsp. Após criar a página insira nela o código abaixo:

```
7 <title>Página de Resposta</title>
8 </head>
9 <body>
10 <h1>Resultado da Resposta</h1>
11 <h3><% out.print(request.getParameter("nome")); %></h3>
12 </body>
13 </html>
```

- Para recebermos parâmetros na página JSP, assim como na servlet, utilizamos **“request.getParameter(“nome”)”**.



# I EXERCÍCIO

Crie um projeto dinâmico que na página index.jsp tenha um formulário com os seguintes campos:

- Nome;
- Idade;
- Endereço;
- Cidade;
- Estado;
- Telefone;
- Email.

Agora crie outra página chamada cadastro.jsp para receber as informações de nosso formulário.

- Scriptlets
- Expressions
- Declarations

- São um bloco de código Java inserido no JSP
- Um scriptlet deve iniciar com `<%` e terminar com `%>`

```
<%  
    ArrayList<String> lista = new ArrayList<String>();  
    lista.add("Algoritmos");  
    lista.add("LTP");  
    lista.add("Web");  
%>
```

- Código Java inserido no JSP
- Deve resultar em uma String, que será exibida na página resultante (HTML)
- Uma expression deve iniciar com **<%=** e terminar com **%>**

```
<%=lista.get(0) %>
```

# I DECLARATIONS

- Código Java inserido no JSP
- Utilizada para declarar atributos e métodos
- Uma expression deve iniciar com **<%!** e terminar com **%>**

```
<%!  
public int retorna(int  
n){  
return n*3;  
}  
%>  
<%= retorna(5) %>
```

- **OBS. Evite colocar regras de negócio em páginas JSP!**

- Dentro dos scriptlets, os comentários são iguais aos utilizados em classes Java:

```
<%  
    //Comentário de linha  
  
    /*  
    Comentário de bloco  
    */  
%>
```

- Fora dos scriptlets, a notação é:

```
<!-- Comentário -->
```

# EXERCÍCIO

Crie um projeto com uma página index.jsp contendo o formulário abaixo:

## FORM

Name	Method	Action
preferencias	Get	cadastrar.jsp

## ELEMENTOS

Label	Id	Name	Type	Value
Nome:	idNome	nome	Text	
E-Mail:	idEmail	email	Text	
Cor favorita:	idCor	corFavorita	Select	Azul, Preto, Vermelho, Verde, Amarelo, Branco
Manequim:	idManequim	manequim	Radio	Pequeno, Medio, Grande, Extra Grande
Estilo preferido:	idEstilo	estilo	CheckBox	Despojado, Casual, Esporte, Social
			Submit	Enviar
			Reset	Limpar

Agora crie uma página chamada “cadastrar.jsp” para receber as informações.

- Elas começam com `<%@` e terminam com `%>`
- Podem ser de 3 tipos:
  - Include:
    - Inclui código de um arquivo externo no JSP
  - Taglib:
    - Define uma tag library
  - Page:
    - Define propriedades da página



# I DIRETIVA INCLUDE

- Permite incluir um arquivo externo no JSP
- Ideal para incluir informações que se repetem em várias páginas, ex: Menu, Cabeçalhos..

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

# I DIRETIVA TAGLIB

- Permite referenciar tag libraries na página
- Podemos utilizar uma infinidade de tags para construir páginas JSPs

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

# I EXERCÍCIO

Usando a diretiva `include`, crie um cabeçalho e um rodapé para os dois projetos anteriores.

Os dois devem receber estilização com CSS.

- Define propriedades para a página JSP
- Podemos definir diversas propriedades:
  - language
  - contentType
  - pageEncoding
  - import

- Os atributos language, contentType e pageEncoding definem a linguagem e codificação da página
- Ela é definida automaticamente quando criamos páginas JSPs utilizando o Eclipse.

```
<%@ page language="java" contentType="text/html;  
charset=ISO-8859-1" pageEncoding="ISO-8859-1"%>
```

# I DIRETIVA PAGE - IMPORT

- O atributo import é utilizado para importar classes que serão utilizados na página JSP
- Funciona de forma parecida com o import em classes
- Devemos utilizar o pacote + nome da classe, ou pacote + .\* para importar todas as classes do pacote

```
<%@ page import="java.util.ArrayList,  
java.util.Calendar" %>
```

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

# I OBJETOS IMPLÍCITOS

- São objetos disponíveis nas páginas, dentro dos scriptlets

<% %>

Objeto Implícito	Classe	Descrição
Out	JspWriter	Saída de dados
Application	ServletContext	Configuração
Config	ServletConfig	Configuração
Request	HttpServletRequest	Request
Response	HttpServletResponse	Response
Session	HttpSession	Sessão
pageContext	PageContext	Contexto da página

- No exemplo abaixo, estamos utilizando o objeto out para imprimir tags e texto na página HTML. E também estamos utilizando o objeto session para recuperar um atributo da sessão do usuário.

```
<%  
    out.print("<h1>Olá Mundo</h1>");  
    session.getAttribute("usuario");  
%>
```



- Resgata um valor da aplicação.

Ex: Insira o código abaixo no seu arquivo web.xml.

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema"
  <display-name>Aula-JSP</display-name>
  <welcome-file-list>
    <welcome-file>index.jsp</welcome-file>
  </welcome-file-list>
  <context-param>
    <param-name>cidade</param-name>
    <param-value>São Paulo</param-value>
  </context-param>
</web-app>
```

Agora vamos recuperar este valor utilizando o application:

```
<%= application.getInitParameter("cidade") %>
```

- Fica ativa em todo projeto até ser destruída. Usamos ela para controlar a atividade de usuários nas páginas

Ex: Vamos criar uma sessão na página index.

```
<% session.setAttribute("usuario", "Luís Carlos"); %>
```

Agora vamos resgatar nossa sessão nos arquivos “resposta.jsp” e “resposta1.jsp” esse segundo temos que criar :

```
<%= "Usuário logado: " + session.getAttribute("usuario") %>
```

Copyright © 2018 Prof. Rafael Matsuyama / Prof. Luís Carlos de S Silva

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito, do Professor (autor).