



Recibe una cálida:

# ¡Bienvenida!

---

Te estábamos esperando 😊



# » Java Server Pages

---

**Plan formativo:** Desarrollo de Aplicaciones Full Stack Java Trainee V2.0

# HOJA DE RUTA

¿Cuáles **skill** conforman el programa?



# REPASO CLASE ANTERIOR

En la clase anterior trabajamos :

- ✓ Concepto e implementación de tags: `c:if`, `c:choose` y `c:out`

# LEARNING PATHWAY

¿Sobre qué temas trabajaremos?

5.2

Start! 🏁

Java Server Pages

Tag c:forEach

C:FOREACH

Iterando con c:forEach

Utilizando funciones útiles en JSTL

# OBJETIVOS DE APRENDIZAJE

¿Qué aprenderemos?



*Reconocer la implementación de la etiqueta `c:forEach`*



*Comprender el uso de las funciones en JSTL*





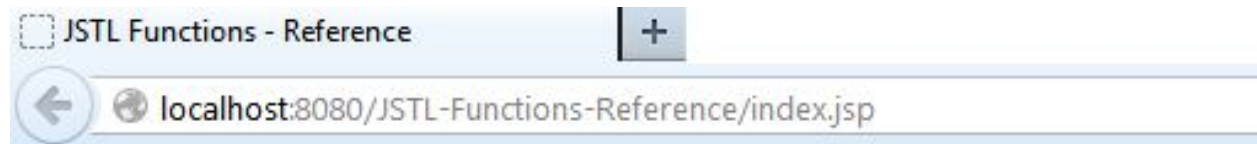
# Rompehielo 🧊

Ahora que ya sabemos que podemos utilizar código Java en las vistas:

**Respondan en el chat y levantando la mano 🙋🙋**

1- ¿Es posible iterar listas desde la vista? ¿Cómo?

2- ¿Y si quiero buscar una subcadena o medir la longitud de una cadena desde el JSP?

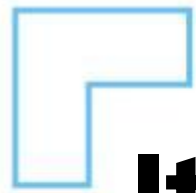


## JSTL Functions - Reference

<a href="#">fn:contains</a>	<a href="#">fn:containsIgnoreCase</a>	<a href="#">fn:endsWith</a>	<a href="#">fn:escapeXml</a>
<a href="#">fn:indexOf</a>	<a href="#">fn:join</a>	<a href="#">fn:length</a>	<a href="#">fn:replace</a>
<a href="#">fn:split</a>	<a href="#">fn:startsWith</a>	<a href="#">fn:substring</a>	<a href="#">fn:substringAfter</a>
<a href="#">fn:substringBefore</a>	<a href="#">fn:toLowerCase</a>	<a href="#">fn:toUpperCase</a>	<a href="#">fn:trim</a>

# › Iterando con <c:forEach>





# Iterando con `<c:forEach>`



## ¿Para qué sirve?:

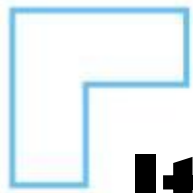
Se utiliza para **realizar iteraciones sobre una colección de elementos** en una página JSP. Permite recorrer listas, arrays, mapas y otros tipos de colecciones de datos.



La **sintaxis** básica de `<c:forEach>` es la siguiente:

```
<c:forEach items="colección" var="elemento" [otros atributos opcionales]>  
  <!-- Código a ejecutar por cada elemento de la colección -->  
</c:forEach>
```





# Iterando con `<c:forEach>`

Estas etiquetas se utilizan como una buena **alternativa para** incrustar un **bucle while, do-while o for** de Java a través de un scriptlet. La etiqueta `<c:forEach>` es la etiqueta más utilizada porque itera sobre una colección de objetos.

Veamos un ejemplo: en este caso, le estamos dando parámetros de comienzo y finalización. 

```
<body>
<c:forEach var="j" begin="1" end="3">
  Item <c:out value="${j}"/> <p>
</c:forEach>
</body>
```

```
Item 1
Item 2
Item 3
```

 Con la etiqueta `<c:out>` estamos imprimiendo 'Item + el valor de j' en cada iteración.

# LIVE DEMO

Ejemplo en vivo

## Probando, probando:

*En este ejemplo vamos probar todas las etiquetas JSTL que hemos aprendido. Vamos a setear algunos valores para que podamos realizar todo el ejercicio desde la página JSP. En este caso, simularemos algunos importes + IVA.*

*Comenzaremos con las siguientes líneas:*

```
<c:set value="500" var="importe" />
```

```
<c:out value="{importe}" />
```

```
<c:set value="Profe" var="nombre" />
```

```
<c:out value="{nombre}" />
```

# LIVE DEMO

Ejemplo en vivo

*A continuación vamos a calcular el IVA:*

```
<c:set value="\${importe*.21}" var="iva" />
```

```
<c:out value="\${iva}" />
```

```
<c:if test="\${iva > 100}">
```

```
<p>iQue caro!</p>
```

```
</c:if>
```

# LIVE DEMO

Ejemplo en vivo

*Luego, vamos a mostrar mensajes por pantalla según la evaluación:*

```
<c:choose>
  <c:when test="{iva < 100}">
    <c:out value="{iva} menor de 100" />
  </c:when>
  <c:when test="{iva < 200}">
    <c:out value="{iva} menor de 200" />
  </c:when>
  <c:otherwise>
    + <c:out value="{iva} mayor de 200" />
  </c:otherwise>
</c:choose>
```

# LIVE DEMO

Ejemplo en vivo

*Finalmente, recorreremos los items y mostramos lo iterado:*

```
<c:forEach var="i" items="1,4,5,6,7,8,9">
```

```
Item <c:out value="Nº ${i}" />
```

```
<p>
```

```
</c:forEach>
```

  **Tiempo: 25 minutos**

# › Funciones JSTL



# Funciones JSTL



## ¿Para qué sirven?:

Se utilizan para realizar **toma de decisiones condicionales** en las páginas JSP. Estas etiquetas **permiten controlar el flujo de ejecución** en función de condiciones específicas.



El tag **<c:if>** se utiliza para evaluar una condición y ejecutar un bloque de código si la condición es verdadera.

Por otro lado, el tag **<c:choose>** se utiliza cuando se necesita realizar múltiples tomas de decisiones y ejecutar diferentes bloques de código en función de varias condiciones.







# Funciones JSTL

JSTL dispone de un conjunto de funciones que pueden emplearse desde dentro del lenguaje de expresiones, y permiten sobre todo manipular cadenas, para sustituir caracteres, concatenar, etc.



Para utilizar estas funciones, debemos cargar la directiva taglib correspondiente:



**`<%@ taglib uri="http://java.sun.com/jstl/ea/functions" prefix="fn" %>`**

Y luego utilizamos las funciones que hay disponibles dentro de una expresión del lenguaje:

La cadena tiene `<c:out value="${fn:length(miCadena)}"/>` caracteres





# Funciones JSTL

Algunas de sus funciones útiles son:



- **fn:length()**: Devuelve la longitud de una cadena, una lista, un array o una colección.
- **fn:toUpperCase()** y **fn:toLowerCase()**: Convierten una cadena a mayúsculas o minúsculas, respectivamente.
- **fn:substring(texto,inicio,final)**: Devuelve una subcadena de una cadena dada, especificando el índice de inicio y, opcionalmente, el índice de finalización.
- **fn:replace(texto, palabra1,palabra2)**: Reemplaza todas las apariciones de una subcadena por otra en una cadena dada.
- **fn:contains(var, ítema buscar)**: Verifica si una cadena contiene otra cadena dada y devuelve un valor booleano.



# LIVE CODING

Ejemplo en vivo

**Siguiendo en el ejercicio anterior:** vamos a cambiar el nombre 'Profe' a mayúsculas, y reemplazar la letra "E" por un '3'

Luego, vamos a evaluar si la cadena contiene un caracter '3' en sí misma. Si esto es verdadero, debemos mostrar por pantalla "Nombre inválido"

 **Tiempo: 15 minutos**



# Evaluación Integradora ✨

¿Listos para un nuevo desafío? En esta clase comenzamos a construir nuestro...

## Trabajo Integrador del Módulo 💪

Iremos completándolo progresivamente clase a clase.





# **Ejercicio N° 1**

# **C:FOREACH**



# C:FOREACH



## Manos a la obra: 🙌

Vamos a poner en práctica lo aprendido editando el archivo index.jsp del proyecto AlkeWallet.

## Consigna: ✍️

Vamos a calcular el interés de diversos medios de pago. En caso de que el medio elegido sea Efectivo, el interés será 0. Por otro lado el medio de pago tarjeta tiene un interés del 10%. Finalmente, el medio de pago depósito tiene un interés de 20%.

2. Cargar una lista con los 3 medios de pago y mostrar por pantalla el interés final de cada uno. El valor inicial será el que decidas.

**Tiempo** 🕒: 20 minutos



○

# ¿Alguna consulta?

+



# RESUMEN

¿Qué logramos en esta clase?

- ✓ **Comprender la utilización de la etiqueta `c:forEach`**
- ✓ **Entender la implementación de las funciones `JSTL`**





# #WorkingTime

Continuemos ejercitando

**¡Antes de cerrar la clase!** Te invitamos a: 📌 📌 📌

1. Repasar nuevamente la grabación de esta clase
2. Revisar el material compartido en la plataforma de Moodle (lo que se vio en clase y algún ejercicio adicional)
  - a. *Lectura Modulo 5, Lección 2: páginas 4 - 6*
3. Traer al próximo encuentro, todas tus dudas y consultas para verlas antes de iniciar nuevo tema.

Momento: ✚

# Time-out!

🕒 5 min.

