

La librería **functions** de JSTL (TAGLIB) nos permite trabajar con Strings y Arrays en el lenguaje EL de JSP resolviendo la mayor parte de las necesidades típicas en programación. Aquí muestro los ejemplos de las más utilizadas.

Debemos incluir en nuestra página JSP la llamada a la JSTL functions:

```
1 <%@ taglib prefix="fn" uri="http://java.sun.com/jsp/jstl/functions" %>
2 <c:set var="tempStr" value="Me gusta mucho el lenguaje EL"></c:set>
```

Obtener la cantidad de caracteres de un String o la cantidad de elementos de una colección (array o lista):

```
1 ${fn:length(tempStr)}
```

Comprobar si el String contiene la cadena "test"

```
1 ${fn:contains(tempStr,"test")}
```

Comprobar si el String contiene la cadena "Test" ignorando si son mayúsculas o minúsculas

```
1 ${fn:containsIgnoreCase(tempStr,"test")}
```

Convertir el String a mayúsculas:

```
1 ${fn:toUpperCase(tempStr)}
```

Convertir el String a minúsculas:

```
1 ${fn:toLowerCase(tempStr)}
```

Dividir un String en un array de String indicando un carácter separador, en este ejemplo el "espacio".

```
1 ${fn:split(tempStr," ")}
```

Obtener la cantidad de Strings obtenidos tras el split

```
1 ${fn:length(fn:split(tempStr," "))}
```

Eliminar caracteres vacíos al final del string

```
1 ${fn:trim(tempStr)}
```

Comprobar si un String termina con el sufijo indicado:

```
1 ${fn:endsWith(tempStr,"sufijo")}
```

Escapar los caracteres que podrían ser interpretados en un formato XML

```
${fn:escapeXml(tempStr)}
```

Devuelve el índice donde aparece la primera ocurrencia del substring especificado:

```
1 ${fn:indexOf(tempStr,"prefijo")}
```

Une todos los elementos de un array en un String:

```
1 ${fn:join(ARRAY[], tempStr)}
```

Devuelve el String resultado de reemplazar en el String de entrada todas las ocurrencias del segundo String por el tercero:

```
1 ${fn:replace(tempStr,"stringacambiar","stringnuevo")}
```

Comprueba si el String empieza por el segundo string especificado:

```
1    ${fn:startsWith(tempStr, "comienzo")}
```

Devuelve un subconjunto del String:

```
1    ${fn:substring(tempStr, inicio, fin)}
```

Devuelve un subconjunto del string siguiente al String indicado:

```
1    ${fn:substringAfter(tempStr, "stringcomiezo")}
```

Devuelve un subconjunto del string anterior al String indicado:

```
1    ${fn:substringBefore(tempStr, "stringfinal")}
```