

Training

SIERRA PRINT TEMPLATES Using Java Strings

In addition to customizing the content of Sierra output by selecting the elements to be included, defining the characteristics of the elements, and adding images and URL links, Print Templates can include special instructions in the form of Java Strings.

Java Strings allow for conditional application of elements and formatting of the elements contained within the Print Template. For example, if a message is to be included only for Juvenile patrons, a check can be placed on the patron type and only display the message if it matches the Juvenile values. Another example, for a Hold Slip inserted into material placed holdshelf, an identifier can be constructed to include parts of the patron name and barcode such that it is recognizable by the patron, but not others.

Java Strings are managed by Oracle and are described [here](#).

Java strings are appended to the end of an element in the format `#{element}.javastring()`. For example, `#{callAlphaStart}.equals("")` checks for the presence of the beginning letters of the call number. Note that every Java String is followed with parentheses, but there is not always content within the parentheses.

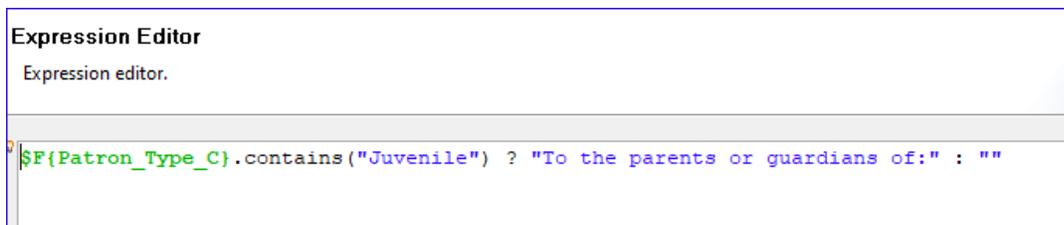
Example 1: Conditionally including a string.

Java String: contains()

Statement: if patron type is Juvenile, then include a message "To the parents/guardian of" before the patron name. If the patron type is not Juvenile, then do not include the message.

```
#{Patron_Type_C}.contains("Juvenile") ? "To the parents or guardians of:" : ""
```

To enter, drag the Patron_Type_C element on to the Title band and then double-click to add the Java String.



`#{Patron_Type_C}.contains("Juvenile")` provides the conditional statement that the patron type either contains or does not contain the string "Juvenile"

Training

? "To the parents or guardians of:" provides instruction if this condition is true. The question mark is the symbol that indicates if true, do the following.

: "" provides instruction (print nothing) if this condition is false. The colon is the symbol that indicates if false, do the following. The two quotes with nothing between indicate print nothing.

The result is each patron record on the circulation notices will either meet or not meet the condition and follow the associated instruction.

<table><tr><td>Preferred Name</td><td>Garner, David</td></tr><tr><td>Barcode</td><td>20102220523203</td></tr><tr><td>Patron Type</td><td>50 Resident Juvenile</td></tr></table>	Preferred Name	Garner, David	Barcode	20102220523203	Patron Type	50 Resident Juvenile	<p>To the parents or guardians of: Garner, David 51 Cedar Street</p> <p>Fifarek, Simon 123 Main Street SE</p>
Preferred Name	Garner, David						
Barcode	20102220523203						
Patron Type	50 Resident Juvenile						
<table><tr><td>Preferred Name</td><td>Fifarek, Simon</td></tr><tr><td>Barcode</td><td>7766554433</td></tr><tr><td>Patron Type</td><td>12 Adult</td></tr></table>	Preferred Name	Fifarek, Simon	Barcode	7766554433	Patron Type	12 Adult	
Preferred Name	Fifarek, Simon						
Barcode	7766554433						
Patron Type	12 Adult						

The patron with "Juvenile" in the patron type includes the message. The patron with "Juvenile" not in the patron type does not include the message.

Example 2: Create a code based on the first 4 characters of the patron last name and the last 4 digits of the barcode

Java String: substring()

Statement: Extract the first 4 characters of the patron last name and first four digits of the patron barcode and merge the two values for a single code.

Start by dragging the Patron_Name_C element on to the Title band and then drag on the Patron_Barcode_C. We will need to combine these into one element.

<code>#{Patron_Name_C}</code>	<code>#{Patron_Barcode_C}</code>
-------------------------------	----------------------------------

Click into the Patron_Barcode_C element, highlight and copy the entire `#{Patron_Barcode_C}` value. Then double-click on the Patron_Name_C element and enter " + " (plus symbol) followed by the Patron_Barcode_C element.

Training

Expression Editor

Expression editor.

```
$F{Patron_Name_C} + $F{Patron_Barcode_C}
```

Now that we have the two elements together, the `$F{Patron_Barcode_C}` element is no longer needed and can be deleted.

We can add in the Substring command to both elements that provide the instruction to extract select characters. For this example, we will assume the barcode is 14 digits in length. In both cases, substring indicates the position of the characters to extract. Note that counting starts at 0, not 1. For example, if the instruction is `$F{Patron_Name_C}.substring(2, 5)`, then characters 3 through 5 are extracted. On the name

Hamilton, that would be **mil**
01234567

So, `substring(0,4)` will extract the first 4 characters of the name and `substring(10,14)` will extract the last 4 digits of the barcode.

```
$F{Patron_Name_C}.substring(0,4) +  
$F{Patron_Barcode_C}.substring(10,14)
```

Preferred Name	Koehler, Michelle
Barcode	20102120047592

Koehler, Michelle
Koeh7592

Patron last name **Koehler** and barcode 2010212004**7592** becomes Koeh7592. One question that may come up is that while the name is written in title case, could the code contain all capitals? Yes, that is another substring called `toUpperCase()` which is appended to the `Patron_Name_C` element, and would be included as follows:

```
$F{Patron_Name_C}.substring(0,4).toUpperCase() +  
$F{Patron_Barcode_C}.substring(10,14)
```

Koehler, Michelle
KOEH7592

Another question is whether we can accommodate last names that are not 4 characters in length. Since we are extracting the data from the `Patron_Name_C` element which contains the entire patron name (last

Training

name, first name middle name/initial), a last name shorter than 4 characters would include the comma and what follows. Here is how a 3-character last name would appear:

```
Lee, Annabelle  
LEE,3606
```

We can introduce a conditional statement that checks if the comma is character 0, 1, 2, or 3 – remembering that the positions begin with 0. This would mean that the last name is length 0 through 3.

The substring indexOf() checks for the position of the character in the parentheses. In our case that is

```
 $\$F\{Patron\_Name\_C\}.indexOf(",") < 4$ 
```

This catches our short last-named patron because if the position of the comma is 3 or or less.

```
Lee,  
0123
```

We will return to the use of the “?” and “:” to say what happens when the statement is true or false. For the last name, will use the position of the comma as part of the substring command. Instead of using the number 4, we will use the index of the comma.

```
 $\$F\{Patron\_Name\_C\}.substring(0, \$F\{Patron\_Name\_C\}.indexOf(","))$ 
```

Putting it all together, the full expression looks like this:

```
( $\$F\{Patron\_Name\_C\}.indexOf(",") < 4 ?$   
 $\$F\{Patron\_Name\_C\}.substring(0,$   
 $\$F\{Patron\_Name\_C\}.indexOf(",")).toUpperCase() :$   
 $\$F\{Patron\_Name\_C\}.substring(0,4).toUpperCase() ) +$   
 $\$F\{Patron\_Barcode\_C\}.substring(10,14)$ 
```

Expression Editor
Expression editor.

```
( $\$F\{Patron\_Name\_C\}.indexOf(",") < 4 ?$   
 $\$F\{Patron\_Name\_C\}.substring(0,$   
 $\$F\{Patron\_Name\_C\}.indexOf(",")).toUpperCase() :$   
 $\$F\{Patron\_Name\_C\}.substring(0,4).toUpperCase() ) +$   
 $\$F\{Patron\_Barcode\_C\}.substring(10,14)$ 
```

Training

The entire Patron_Name_C expression is enclosed in parentheses to ensure that is addressed first followed by the Patron_Barcode_C expression.

If the last name is short, display the name up to the comma. If the name is 4 characters or greater, take the first 4 characters afterwards. In either case, convert to upper case and then append the last 4 digits of the barcode.

Lee, Annabelle
LEE3606

Koehler, Michelle
KOEH7592

Useful Java Strings

JAVA STRING	DESCRIPTION	EXAMPLE
concat(string)	Concatenates the specified string to the end of this string.	<code>#{callAlphaStart}.concat(#{callNumericStart})</code> <i>Merge start of alpha call number with start of numeric call number</i> DF2991
contains(string)	Returns true if and only if this string contains the specified sequence of char values.	<code>#{PONumber}.contains("ADR")</code> <i>Does PO Number contain 'ADR'?</i> True / False
endsWith(string)	Tests if this string ends with the specified suffix.	<code>#{Patron_Type}.endsWith("Resident")</code> <i>Does Patron Type end with 'Resident'?</i> True / False
equals(string)	Compares this string to the specified object.	<code>#{Item_Location}.equals("cendv")</code> <i>Is the location code 'cendv'?</i> True / False
equalsIgnoreCase(string)	Compares this String to another String, ignoring case considerations.	<code>#{Item_Location}.equalsIgnoreCase("CENDV")</code> <i>Is the location code 'CENDV', 'cendv', 'CeNdv', etc?</i> True / False
indexOf(character)	Returns the index within this string of the first occurrence of the specified character, starting the search at the specified index.	<code>#{Item_Message_C}.indexOf(":")</code> <i>Find the position of the colon within the item message.</i> 9
length(string)	Returns the length of this string.	<code>#{Item_Price}.length()</code> <i>What is the length of the price field?</i>

Training

		5
matches (regular expression)	Tells whether this string matches the given regular expression.	<code>#{Patron_Home_Library}.matches("ar[a-z]{3}")</code> <i>Does the Home Library start with 'ar' and then followed by 3 alphabetic characters?</i> True / False
replaceAll(string to replace or regular expression, replacement string)	Replaces each substring of this string that matches the given regular expression with the given replacement.	<code>#{Item_Title}.replaceAll(" ", "_")</code> <i>Replace all the spaces in the title with underscores.</i> Fiction_and_the_figures_of_life
startsWith(string)	Tests if this string starts with the specified prefix.	<code>#{Patron_Type}.startsWith("Resident")</code> <i>Does Patron Type begin with 'Resident'?</i> True / False
substring()	Returns a new string that is a substring of this string.	<code>#{Item_Message}.substring(0,14)</code> <i>Extract the first 14 characters of the item message.</i> Tue Oct 06 2020
toLowerCase()	Converts all of the characters in this String to lower case using the rules of the default locale.	<code>#{Item_Call_Number}.toLowerCase()</code> <i>Convert call number to lower case.</i> hb103.k47 h28
toUpperCase()	Converts all of the characters in this String to upper case using the rules of the default locale.	<code>#{Item_Call_Number}.toUpperCase()</code> <i>Convert call number to upper case.</i> HB103.K47 H28
trim()	Returns a copy of the string, with leading and trailing spaces omitted.	<code>#{item_author}.trim()</code> <i>Remove spaces before and after the author field.</i> Meriwether, Louise.

This publication is supplied for the exclusive use of customers of Innovative Interfaces with the understanding that it shall not be shown or distributed to anyone outside of the customer's organization without the prior written permission of Innovative Interfaces. This publication may be copied only if the copies are for the exclusive use of staff members of libraries that have purchased the Innovative system.

© 2022, Innovative Interfaces, Inc.