🔍 JMeter String Functions – Detailed Guide with Examples

JMeter provides a variety of **built-in string functions** that help with **string manipulation, formatting, encoding, and transformations** during performance testing. These functions can be used in **Regular Expression Extractors, Assertions, Debugging, CSV Data Processing, and Parameterization.**

📌 1. List of String Functions in JMeter

Function	Usage	Example	Output
${__strReplace(source, search, replace, variable)}	Replaces a substring within a string	${__strReplace(Hello World, World, JMeter, result)}	Hello JMeter
${__substring(source, start, end, variable)}	Extracts a substring from a string	${__substring(JMeterTest, 0, 6, result)}	JMeter
${__char(value)}	Returns the character representation of ASCII code	${__char(65)}	A
${__unescapeHtml(html_string)}	Converts HTML entities back to normal text	${__unescapeHtml(&lt;b&gt;bold&lt;/b&gt;)}	<b>bold</b>
${__escapeHtml(text)}	Converts special characters to HTML entities	${__escapeHtml(<b>bold</b>)}	&lt;b&gt;bold&lt;/b&gt;
${__unescapeJava(escaped_string)}	Converts escaped Java	${__unescapeJava(Hello\nWorld)}	Hello (new line) World

Function	Usage	Example	Output
	sequences to normal text		
${__unescapeXml(xml_string)}	Converts XML escaped sequences to normal text	${__unescapeXml(&lt;tag&gt;value &lt;/tag&gt;)}	<tag>value</tag>
${__escapeXml(text)}	Converts text into XML-safe characters	${__escapeXml(<tag>value</tag>)}	&lt;tag&gt;value&lt;/tag&gt;
${__toLowerCase(text)}	Converts text to lowercase	${__toLowerCase(JMETER)}	jmeter
${__toUpperCase(text)}	Converts text to uppercase	${__toUpperCase(jmeter)}	JMETER
${__trim(text)}	Removes leading and trailing spaces	${__trim(JMeter)}	JMeter
${__split(source, delimiter, variable)}	Splits a string based on a delimiter	${__split(apple,banana,grape, ",", fruit)}	${fruit_1} = apple, ${fruit_2} = banana, ${fruit_3} = grape
${__StringFromFile(filepath, variable)}	Reads a string from a file	${__StringFromFile(/path/to/file.txt, myVar)}	Contents of file.txt
${__stringify(value)}	Converts a variable into a string	${__stringify(${randomNum})}	String representation of ${randomNum}

📌 2. Detailed Explanation & Usage with Examples

🔹 1. Replacing Substrings in JMeter

💡 **Scenario**: Replace "Hello World" with "Hello JMeter".

${__strReplace(Hello World, World, JMeter, result)}

✅ **Result**: Hello JMeter

🔹 Usage in a JMeter HTTP Request

```
<HTTPSamplerProxy>
  <stringProp name="Argument.value">${__strReplace(hello world, world, JMeter)}</stringProp>
</HTTPSamplerProxy>
```

🔹 2. Extracting a Substring

💡 **Scenario**: Extract first 6 characters of "JMeterTest".

${__substring(JMeterTest, 0, 6, result)}

✅ **Result**: JMeter

🔹 Usage in a JMeter Variable

```
<UserDefinedVariables>
  <stringProp name="testString">JMeterTesting</stringProp>
</UserDefinedVariables>
<HTTPSamplerProxy>
  <stringProp name="Argument.value">${__substring(${testString}, 0, 6)}</stringProp>
</HTTPSamplerProxy>
```

🔹 3. Changing Case (Uppercase & Lowercase)

💡 **Scenario**: Convert "jmeter" to uppercase.

${__toUpperCase(jmeter)}

✅ **Result**: JMETER

💡 **Scenario**: Convert "JMeter" to lowercase.

${__toLowerCase(JMeter)}

✅ **Result**: jmeter

🔹 **Usage in CSV Data Set Config**

```
<CSVDataSet>
  <filename>users.csv</filename>
  <variableNames>name</variableNames>
</CSVDataSet>
<HTTPSamplerProxy>
  <stringProp name="Argument.value">${__toUpperCase(${name})}</stringProp>
</HTTPSamplerProxy>
```

🔹 **4. Removing Leading and Trailing Spaces**

💡 **Scenario**: Trim spaces from " JMeter ".

```
${__trim(   JMeter   )}
```

✅ **Result**: JMeter

🔹 **Usage in PreProcessor**

```
String rawValue = vars.get("rawData");
String trimmedValue = rawValue.trim();
vars.put("cleanData", trimmedValue);
```

🔹 **5. Encoding and Decoding HTML/XML**

💡 **Scenario**: Convert HTML to readable text.

```
${__unescapeHtml(&lt;b&gt;bold&lt;/b&gt;)}
```

✅ **Result**: <b>bold</b>

💡 **Scenario**: Escape XML.

```
${__escapeXml(<tag>value</tag>)}
```

✅ **Result**: &lt;tag&gt;value&lt;/tag&gt;

🔹 **6. Splitting a String**

💡 **Scenario**: Split "apple,banana,grape" using , as a delimiter.

```
${__split(apple,banana,grape, ",", fruit)}
```

✅ **Results**:

- ${fruit_1} = apple
- ${fruit_2} = banana
- ${fruit_3} = grape

🔹 **Usage in a Loop Controller**

```
<LoopController>
  <stringProp name="LoopController.loops">${fruit_matchNr}</stringProp>
</LoopController>
```

🔹 **7. String Manipulation with BeanShell**

💡 **Scenario**: Extract substring from a variable in BeanShell.

```
String text = vars.get("responseText");
String extracted = text.substring(5, 10);
vars.put("subStringVar", extracted);
log.info("Extracted String: " + extracted);
```

📌 **3. Real-world Use Cases**

📌 **Case 1: Extracting and Formatting a JSON Response**

💡 **Scenario**: Extract username and convert to **uppercase**.

✅ **Steps**

1. Use **JSON Extractor**:

   ```
   $.user.username
   ```

   Stores result in ${username}.

2. Convert to uppercase in a request:

   ${__toUpperCase(${username})}

📌 **Case 2: Reading from CSV and Replacing Values**

💡 **Scenario**: Read usernames from users.csv and replace _ with -.

✅ **Steps**

1. **CSV Data Set Config**
   - **File:** users.csv
   - **Variable Name:** username
2. **Use strReplace in Request**

   ${__strReplace(${username}, _, -, formattedUsername)}

📌 **Case 3: Using String Functions in Assertions**

💡 **Scenario**: Verify API response contains "SUCCESS" in **uppercase**.

✅ **Steps**

1. **Regular Expression Extractor**
   - Extracts "success" as ${status}.
2. **Response Assertion**
   - Expected: ${__toUpperCase(${status})}
   - Pattern: SUCCESS

🎯 Conclusion

Function	Use Case
__strReplace()	Replacing substrings dynamically
__substring()	Extracting parts of strings
__toUpperCase() / __toLowerCase()	Converting case
__trim()	Removing extra spaces
__split()	Splitting string data
__escapeHtml() / __unescapeHtml()	Handling HTML entities
__escapeXml() / __unescapeXml()	Handling XML