runScript

This function is used by **Ignition's Expression language**.

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Runs a single line of Python code as an expression. If a poll rate is specified, the function will be run repeatedly at the poll rate. This is a very powerful way for you to add extensions to the expression language. For example, one could write a project script module function called <code>shared.weather.getTempAt(zip)</code> that queried a web service for the current temperature at a given zipcode, and then bind the value of a label to the return value of that function. The <strong>scriptFunction</strong> is entered as a <strong>string</strong> and the <strong>pollRate</strong> is in <strong>milliseconds</strong>. You can optionally add any function arguments after the poll rate.</td>
</tr>
</tbody>
</table>

---

**runScript Polling in Tags**

The runScript function can be used in expression tags, but the poll rate doesn't work exactly the same as in an expression binding. All Tags have a Scan Class that dictates the minimum amount of time between each evaluation. The runScript poll rate only polls **up to** the rate of the scan class set on the tag.

For example, if an Expression Tag is configured with runScript to run at a poll rate of 60 seconds and is using the "default" (1 second) scan class, the Tag's Expression will still execute every 1 second. So a scan class rate of 60 seconds will be necessary for a runScript expression to poll at any rate between 0 and 60 seconds.

---

**Syntax**

```python
runScript(scriptFunction[, pollRate][, args...])
```

- **Parameters**
  - `string scriptFunction` - A single line of python code. Typically the path to a script module.
  - `int pollRate` - Optional. The poll rate of the script.
  - `object args` - Optional. Any number of argument objects that will be passed into the given script.

- **Results**
  - `object` - The return value of the specified function.
Examples

Here is our scripting function we are going to run that is located in a shared script called textScript.

**Code Snippet - Python Function**

```python
def myFunc(text="Hello World!"): return text
```

**Code Snippet**

// run a shared function with this expression
runScript("shared.textScript.myFunc()", 0) // This would run the script and return "Hello World!".

**Code Snippet**

// run a shared function dynamically with this expression using string concatenation. A poll rate is unnecessary, as it will refresh when the tag value changes.
runScript("shared.textScript.myFunc(" +\{_gensim_/Writeable/WriteableString1} + ")", 0) // This would run the function and pass in the value of the WriteableString1 tag.

**Code Snippet**

// run a shared function dynamically with this expression using optional arguments. A poll rate is unnecessary, as it will refresh when the tag value changes.
// Note the missing "()" at the end of the scriptFunction string
runScript("shared.textScript.myFunc", 0, \{_gensim_/Writeable/WriteableString1})