

Apache NiFi 式言語一覽

(Ver1.0)

MIT License

Copyright (c) 2025 NTT InfraNet

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

商標および知的財産権について

Apache NiFiは、The Apache Software Foundationの米国およびその他の国における登録商標または商標です。

前提事項

本資料は、Apache NiFiの公式ドキュメントを翻訳して作成した資料となります。
より詳細な内容については、Apache NiFiの公式ドキュメントをご参照ください。

Apache NiFi 式言語一覧

No	種別	頻度	関数	引数	戻り値	例
1	判定		isNull()	なし	Attributeがnull : true、それ以外 : false	<code>\${filename:isNull()}</code>
2	判定		notNull()	なし	Attributeに値が存在する : true、それ以外 : false	<code>\${filename:notNull()}</code>
3	判定		isEmpty()	なし	Attributeがnull、文字が含まれていないもしくは空白のみが含まれる : true それ以外 : false	<code>\${filename:isEmpty()}</code>
4	比較	☆	equals()	比較したい文字	Attributeと引数が等しい : true それ以外 : false	<code>\${filename>equals('test.csv')}</code>
5	比較		equalsIgnoreCase()	比較したい文字	Attributeと引数が等しい : true それ以外 : false ただし、大文字小文字のみ差異があってもtrueを返します。	<code>\${filename>equalsIgnoreCase('Test.csv')}</code>
6	比較		gt()	比較したい数値	Attributeが引数より大きい : true それ以外 : false	<code>\${fileSize:gt(1024)}</code>
7	比較		ge()	比較したい数値	Attributeが引数より大きい、または等しい : true それ以外 : false	<code>\${fileSize:ge(1024)}</code>
8	比較		lt()	比較したい数値	Attributeが引数より小さい : true それ以外 : false	<code>\${fileSize:lt(1048576)}</code>
9	比較		le()	比較したい数値	Attributeが引数より小さい、または等しい : true それ以外 : false	<code>\${fileSize:le(1048576)}</code>
10	判定		and()	NiFi式	andの左右の式を両方満たす : true それ以外 : false	<code>\${filename>equals('test.csv'):and(\${filename:length():ge(5)})}</code>
11	判定		or()	NiFi式	orの左右の式のどちらかを満たす : true それ以外 : false	<code>\${filename>equals('test.csv'):or(\${filename:length():ge(5)})}</code>
12	判定		not()	なし	指定した条件式の論理値を反転 (true → false、false → true) して返します。	<code>\${filename>equals('test1.csv'):not()}</code>
13	判定		ifElse()	条件式がtrueの場合に返す値、条件式がfalseの場合に返す値	条件式が true の場合は第1引数を、false の場合は第2引数を返します。 右例) filenameが「test.csv」の時、foundを返します。	<code>\${filename>equals('test.csv'):ifElse('found', 'not_found')}</code>
14	判定		isJson()	なし	AttributeがJSON配列またはオブジェクト : true それ以外 : false	<code>\${jsonObj:isJson()}</code>
15	文字変換		toUpper()	なし	Attributeの文字列を大文字に変換して返します。	<code>\${filename:toUpper()}</code>
16	文字変換		toLower()	なし	Attributeの文字列を小文字に変換して返します。	<code>\${filename:toLower()}</code>
17	文字変換		trim()	なし	Attributeの先頭または末尾の空白を削除します。	<code>\${filename:trim()}</code>
18	抽出		substring()	切り取り開始位置、切り取り終了位置	引数に応じて、Attributeの文字を返します。 右例) filenameが「test.csv」の時、 1つ目は「test」が、2つ目は「st.csv」を返します。 <注意> 「切り取り開始位置」または「切り取り終了位置」がAttributeの長さより大きい場合や「0」未満の時、空白を返します。	<code>\${filename:substring(0,4)}</code> <code>\${filename:substring(2)}</code>
19	抽出	☆	substringBefore()	検索したい文字	Attributeの文字を検索し、引数が最初に出現する直前の文字まで返します。引数がAttributeに存在しない場合は、Attribute全体を返します。 右例) filenameが「test.text.csv」の時、「test」を返します。	<code>\${filename:substringBefore('.')}</code>

Apache NiFi 式言語一覧

No	種別	頻度	関数	引数	戻り値	例
20	抽出	☆	substringBeforeLast()	検索したい文字	Attributeの文字を検索し、引数の最後に出現する直前の文字まで返します。引数がAttributeに存在しない場合は、Attribute全体を返します。 右例) filenameが「test.text.csv」の時、「test.text」を返します。	<code>\${filename:substringBeforeLast('.')}</code>
21	抽出	☆	substringAfter()	検索したい文字	Attributeの文字を検索し、引数が最初に出現する直後の文字から返します。引数がAttributeに存在しない場合は、Attribute全体を返します。 右例) filenameが「test.text.csv」の時、「text.csv」を返します。	<code>\${filename:substringAfter('.')}</code>
22	抽出	☆	substringAfterLast ()	検索したい文字	Attributeの文字を検索し、引数が最後に出現する直後の文字から返します。引数がAttributeに存在しない場合は、Attribute全体を返します。 右例) filenameが「test.text.csv」の時、「csv」を返します。	<code>\${filename:substringAfterLast('.')}</code>
23	抽出		getDelimitedField ()	index, delimiter, quoteChar, escapeChar, stripChars	Attributeを区切られたテキスト行として解析し、1つのフィールドのみを返します。 index : 返されるフィールドのインデックスを指定します。値が1の場合は、1番目のフィールドが返され、値が2の場合は、2番目のフィールドのように～番目を指定します。 delimiter : フィールドを区切り文字として使用する文字を指定します。指定しない場合は、「,」が使用します。 quoteChar : フィールドを引用符で囲むために使用できる文字を指定します。指定しない場合は、「"」が使用します。 escapeChar : フィールド内の引用符または区切り文字をエスケープするために使用できる文字を指定します。指定しない場合、「¥」が使用します。 stripChars : 引用符とエスケープを使用するかtrueまたはfalseで	<code>\${filename:getDelimitedField(2,',')}</code>
24	文字の追加	☆	append()	Attributeの末尾に追加したい文字	Attribute + 引数を返します。 右例) filenameが「test.csv」の時、「test.csv.gz」を返します。	<code>\${filename:append('.gz')}</code>
25	文字の追加		prepend()	Attributeの先頭に追加したい文字	引数 + Attributeを返します。 右例) filenameが「test.csv」の時、「20240513test.csv」を返します。	<code>\${filename:prepend('20240513')}</code>
26	文字置換		replace()	検索文字、置換文字	Attributeの文字を検索し見つければ置き換えます。 右例) filenameが「test.csv」の時、「test_csv」を返します。	<code>\${filename:replace('.', '_')}</code>
27	文字置換		replaceFirst()	検索文字、置換文字	Attribute内で最初に見つかった文字を置き換えます。 右例) tt2が「A,B,C,D,E」の時、「A?B,C,D,E」を返します。	<code>\${tt2:replaceFirst(',', '?')}</code>
28	文字置換		replaceAll()	検索文字（正規表現）、置換文字	正規表現に一致するAttributeの文字をすべて置き換えます。 右例) filenameが「test.csv」の時、「test」を返します。	<code>\${filename:replaceAll('.*', '')}</code>
29	文字補完		padLeft()	埋め込み整数値、埋め込み文字（指定なしは"_"）	埋め込み文字になるようにAttributeの先頭に文字を追加します。埋め込み文字が埋め込み整数値の数を超えている場合は、トリミングされます。 右例) tt2が「A,B,C,D,E」の時、「----- 'A,B,C,D,E」を返します。 右例 2) tt2が「A,B,C,D,E」の時、 「asdfghjklqwertyyuii'A B C D E」を返します。	<code>\${tt2:padLeft(30, "-")}</code> <code>\${tt2:padLeft(30, "asdfghjklqwertyyuiiopzxcvbnm")}</code>

Apache NiFi 式言語一覧

No	種別	頻度	関数	引数	戻り値	例
30	文字補完		padRight()	埋め込み整数値、埋め込み文字（指定なしは“_”）	埋め込み文字になるようにAttributeの末尾に文字を追加します。 埋め込み文字が埋め込み整数値の数を超過している場合は、トリミングされます。 右例) tt2が「A,B,C,D,E」の時、「A,B,C,D,E'----- ---」を返します。 右例 2) tt2が「A,B,C,D,E」の時、 「A,B,C,D,E'asdfghiklqwertyyui」を返します。	<code>\${tt2:padRight(30,"-")}</code> <code>\${tt2:padRight(30,"asdfghiklqwertyyuiiopzxcvbnm")}</code>
31	文字置換		replaceAll()	置換文字(nullの時に返す文字)	Attributeがnullの場合に引数を返します。 nullでない場合は、Attributeを返します。	<code>\${filename:replaceAll('abc')}</code>
32	文字置換		replaceEmpty()	置換文字(Null、空白の時に返す文字)	Attributeがnullまたは空白の場合に引数を返します。 nullまたは空白でない場合は、Attributeを返します。	<code>\${filename:replaceEmpty('abc')}</code>
33	抽出		length()	なし	Attributeの文字数を返します。	<code>\${filename:length()}</code>
34	判定		evaluateELString()	なし	Attributeの値を評価します。	<code>\${query:evaluateELString()}</code>
35	文字補完		repeat()	最小リピート回数、最大リピート回数	最小リピート回数と最大リピート回数の間にランダムな回数だけ繰り返しAttributeの文字を返します。 右例) filenameが「test.csv」の時、「test.csvtest.csv」を返します。	<code>\${filename:repeat(2)}</code>
36	文字エンコード		escapeJson()	なし	JSONドキュメント用のエスケープ文字を追加します。	<code>\${message:escapeJson()}</code>
37	文字エンコード		escapeXml()	なし	Xmlドキュメント用のエスケープ文字を追加します。	<code>\${message:escapeXml()}</code>
38	文字エンコード		escapeCsv()	なし	Csvドキュメント用のエスケープ文字を追加します。	<code>\${message:escapeCsv()}</code>
39	文字エンコード		escapeHtml3()	なし	Html3ドキュメント用のエスケープ文字を追加します。	<code>\${message:escapeHtml3()}</code>
40	文字エンコード		escapeHtml4()	なし	Html4ドキュメント用のエスケープ文字を追加します。	<code>\${message:escapeHtml4()}</code>
41	文字エンコード		unescapeJson()	なし	JSONリテラルのエスケープ文字を削除します。	<code>\${message:unescapeJson()}</code>
42	文字エンコード		unescapeXml()	なし	Xmlリテラルのエスケープ文字を削除します。	<code>\${message:unescapeXml()}</code>
43	文字エンコード		unescapeCsv()	なし	Csvリテラルのエスケープ文字を削除します。	<code>\${message:unescapeCsv()}</code>
44	文字エンコード		unescapeHtml3()	なし	Html3リテラルのエスケープ文字を削除します。	<code>\${message:unescapeHtml3()}</code>
45	文字エンコード		unescapeHtml4()	なし	Html4リテラルのエスケープ文字を削除します。	<code>\${message:unescapeHtml4()}</code>
46	文字エンコード		urlencode()	なし	AttributeをURLエンコードします。	<code>\${url:urlEncode()}</code>
47	文字エンコード		urldecode()	なし	AttributeをURLデコードします。	<code>\${url:urlDecode()}</code>
48	文字エンコード		base64Encode()	なし	AttributeをBase64エンコードします。	<code>\${payload:base64Encode()}</code>
49	文字エンコード		base64Decode()	なし	AttributeをBase64デコードします。	<code>\${payload:base64Decode()}</code>
50	文字エンコード		UUID3()	UUID識別子	タイプ 3 (MD5/ハッシュ) のUUIDを返します。	<code>\${attr:UUID3('b9e81de3-7047-4b5e-a822-8fff5b49f808')}</code>
51	文字エンコード		UUID5()	UUID識別子	タイプ 5 (SHA-1/ハッシュ) のUUIDを返します。	<code>\${attr:UUID5('245b55a8-397d-4480-a41e-16603c8cf9ad')}</code>
52	文字エンコード		hash()	ハッシュ関数 例 : SHA-384, SHA-224, SHA-256, MD2, SHA, SHA-512, MD5	指定されたハッシュ関数を使用して、16進数でエンコードされた文字を返します。	<code>\${payload:hash('SHA-256')}</code>
53	判定		startsWith()	検索したい文字	Attributeが引数で始まる : true それ以外 : false	<code>\${filename:startsWith('test')}</code>
54	判定		endsWith()	検索したい文字	Attributeが引数で終わる : true それ以外 : false	<code>\${filename:endsWith('csv')}</code>
55	判定		contains()	検索したい文字	Attributeの値のどこかに引数の値が含まれる : true それ以外 : false	<code>\${filename:contains('new')}</code>
56	判定		in()	検索したい文字（1 番目の検索文字）、検索したい文字（N番目の検索文字）	Attributeが引数のいずれかと等しい : true それ以外 : false	<code>\${myEnum:in("PAUL", "JOHN", "MIKE")}</code>
57	判定		find()	検索したい文字（正規表現）	Attributeが引数の正規表現と部分一致 : true それ以外 : false	<code>\${filename:find('brand')}</code>
58	判定		matches()	検索したい文字（正規表現）	Attributeが引数の正規表現に完全一致 : true それ以外 : false	<code>\${filename:matches('a.*txt')}</code>
59	判定		indexOf()	検索したい文字	引数と一致するAttributeの最初の文字のインデックス数を返します。 右例) filenameが「test.csv」の場合、5を返します。	<code>\${filename:indexOf('csv')}</code>

Apache NiFi 式言語一覧

No	種別	頻度	関数	引数	戻り値	例
60	判定		lastIndexOf()	検索したい文字	引数と一致するAttributeの最後の文字のインデックス数を返します。 右例) filenameが「test.csv.csv」の場合、9 を返します。	<code>\${filename:lastIndexOf('csv')}</code>
61	JSON		jsonPath()	JSONパス式	AttributeをJSONパス式として扱い、引数から検索し、文字を返します。 ただし、Attributeが空またはAttributeに無効なJSONパス式が含まれていると、例外通知が表示されます。 右例) myJsonに「firstName»: "John"とパスが定義されている場合、「John」を返します。	<code>\${myJson:jsonPath('\$.firstName')}</code>
62	JSON		jsonPathDelete()	JSONパス式	AttributeをJSONパス式として扱い、引数に一致する文字を削除した後、JSONパス式を返します。 ただし、Attributeが空またはAttributeに無効なJSONパス式が含まれていると、例外通知が表示されます。 右例) myJsonに「firstName»: "John"、"lastName": "Smith"とパスが定義されている場合、「lastName»: "Smith"」を返します。	<code>\${myJson:jsonPathDelete('\$.firstName')}</code>
63	JSON		jsonPathAdd()	JSONパス式、追加したい値	AttributeをJSONパス式として扱い、引数からJSONパス式に一致する値を追加し、返します。 ただし、Attributeが空またはAttributeに無効なJSONパス式が含まれていると、例外通知が表示されます。 右例) myJsonに「firstName»: "John"、"nicknames": []とパスが定義されている場合、「firstName»: "John"、"nicknames": ["Jimmy"]」を返します。	<code>\${myJson:jsonPathAdd('\$.nicknames', 'Jimmy')}</code>
64	JSON		jsonPathSet()	JSONパス式、更新したい値	AttributeをJSONパス式として扱い、引数からJSONパス式を更新し、返します。 ただし、Attributeが空またはAttributeに無効なJSONパス式が含まれていると、例外通知が表示されます。 右例) myJsonに「firstName»: "John"とパスが定義されている場合、「firstName»: "James"」を返します。	<code>\${myJson:jsonPathSet('\$.firstName', 'James')}</code>
65	JSON		jsonPathPut()	JSONパス式、追加したいパス、追加したい値	AttributeをJSONパス式として扱い、引数からJSONパス式を追加し、返します。 ただし、Attributeが空またはAttributeに無効なJSONパス式が含まれていると、例外通知が表示されます。 右例) myJsonに「firstName»: "John"とパスが定義されている場合、「firstName»: "John"、"middlename": "Turon"」を返します。	<code>\${myJson:jsonPathPut('\$.middlename', 'Turon')}</code>
66	計算		plus()	加算したい値	Attributeに引数を加算します。 ただし、オーバーフローの処理を行いません。	<code>\${fileSize:plus(1000)}</code>
67	計算		minus()	減算したい値	Attributeに引数を減算します。 ただし、オーバーフローの処理を行いません。	<code>\${fileSize:minus(100)}</code>
68	計算		multiply()	乗算したい値	Attributeに引数を乗算します。 ただし、オーバーフローの処理を行いません。	<code>\${fileSize:multiply(1024)}</code>
69	計算		divide()	除算したい値	Attributeに引数を除算した商を返します。	<code>\${fileSize:divide(12)}</code>
70	計算		mod()	除算したい値	Attributeに引数を除算した余りを返します。	<code>\${fileSize:mod(12)}</code>
71	文字置換		toRadix()	基数（2 から 36 までの数値）、変換後に出力する最小文字数（ゼロ埋め）	Attributeを、10進整数から引数に変換する。	<code>\${fileSize:toRadix(11)}</code>
72	文字置換		fromRadix()	基数（2 から 36 までの数値）	Attributeを、引数から10進整数に変換する。	<code>\${fileSize:fromRadix(11)}</code>
73	抽出		random()	なし	0 ～ 2 ⁶³ -1 の間のランダムな整数を返します。 右例) 1 から 10 までの乱数を返します。	<code>\${random():mod(10):plus(1)}</code>
74	抽出		math()	実行するJava Mathメソッドの名前、メソッドの2番目のパラメーター(省略可能)	この式は、上級ユーザーのみが使用するように設計されています。 Java Reflectionを利用して、Java Mathメソッドを実行します。	<code>\${math("random")}</code>

Apache NiFi 式言語一覧

No	種別	頻度	関数	引数	戻り値	例
75	日付操作		format()	Java DateTimeFormatter構文で使用する形式、タイムゾーンを指定する。(Javaタイムゾーン構文内)	引数で指定された形式に従って、数値を日付/時刻として書式設定します。 指定しない場合はローカルタイムゾーンになります。	<code>\${att1:format("yyyy/MM/dd HH:mm:ss.SSS'Z'", "Asia/Tokyo")}</code>
76	日付操作		formatInstant()	Java DateTimeFormatter構文で使用する形式、タイムゾーンを指定する。(Javaタイムゾーン構文内)	引数で指定された形式に従って、インスタント、数値、または文字を返します。 指定しない場合はローカルタイムゾーンになります。	<code>\$(timeIsoInstant:formatInstant("yyyy/MM/dd", "Asia/Tokyo"))</code>
77	日付操作		toDate()	Java DateTimeFormatter構文で使用する形式、Attributeを解析するときに使用するタイムゾーンをJava TimeZone構文で指定する。	引数で指定された形式に基づいて、文字を日付データ型に変換します。 右例) att1が「05-13-2024」の時、「2024/05/13」と返します。	<code>\$(att1:toDate('MM-dd-yyyy')):format('yyyy/MM/dd')</code>
78	日付操作	☆	toInstant()	Java DateTimeFormatter構文で使用する形式、Attributeを解析するときに使用するタイムゾーンをJava TimeZone構文で指定する。	文字または数値を、引数で指定された形式に基づいてインスタントデータ型に変換します。	<code>\$(time:toInstant("yyyy/MM/dd HH:mm:ss.SSS'Z'", "GMT"))</code>
79	日付操作		toMicros()	なし	Attributeをマイクロ秒の精度で数値に変換します。	<code>\$(dateTime:toInstant('yyyy/MM/dd HH:mm:ss.SSSSSSSS', 'America/New_York'):toMicros())</code>
80	日付操作		toNanos()	なし	Attributeをナノ秒の精度で数値に変換します。	<code>\$(dateTime:toInstant('yyyy/MM/dd HH:mm:ss.SSSSSSSS', 'America/New_York'):toNanos())</code>
81	日付操作		now()	なし	現在の日付と時刻を日付データ型として返します。	<code>\$(now())</code>
82	型変換		toString()	なし	Attributeを文字に変換します。	<code>\$(fileSize:toString())</code>
83	型変換		toNumber()	なし	Attributeを数値に変換します。	<code>\$(fileSize:toNumber())</code>
84	文字置換		toDecimal()	なし	AttributeをDecimal型（10進数）に変換します。	<code>\$(fileSize:toDecimal())</code>
85	抽出		ip()	なし	PCの IP アドレスを返します。	<code>\$(ip())</code>
86	抽出		hostname()	ホスト名を完全修飾するかを指定します。指定しない場合は、完全修飾されません。	PCのホスト名を返します。	<code>\$(hostname())</code>
87	抽出		UUID()	なし	ランダムに生成されたUUID4を返します。	<code>\$(UUID())</code>
88	抽出		nextInt()	なし	1つの値（0 から開始）を返します。値はプロセッサの実行に応じて増加します。	<code>\$(nextInt())</code>
89	抽出		literal()	リテラル文字、数値、または正誤値。	引数をリテラル文字値として返します。	<code>\$(literal(2):gt(1))</code>
90	抽出		getStateValue()	確認したい状態値	プロセッサの状態値をアクセスする引数を渡し、状態値を確認します。	<code>\$(getStateValue("count"))</code>
91	抽出		thread()	なし	式を評価するときにプロセッサによって使用されるスレッドの名前を返します。	<code>\$(thread())</code>
92	抽出		getUri()	スキーム、ユーザー情報、ホスト、ポート、パス、クエリ、フラグメント ※対応するパラメータにnullを渡すか、portパラメータの場合は-1を渡すことで、未定義のままにしておくことができます。	RFC2396に準拠したURIを返します。 右例) <code>https://nifi.apache.org:8443/docs.html</code>	<code>\$(getUri('https', null, 'nifi.apache.org', 8443, '/docs.html', null, null))</code>
93	判定		anyAttribute()	確認したいAttribute名	引数のいずれかが指定された条件に一致するかどうかを確認します。	<code>\$(anyAttribute("att1", "filename"):contains("-"))</code>
94	判定		allAttributes()	確認したいAttribute名	全ての引数に指定された条件に一致するかどうかをチェックします。	<code>\$(allAttributes("att1", "filename"):contains("-"))</code>
95	判定		anyMatchingAttribute()	検索文字（正規表現）	引数のいずれかが指定された条件に一致するかどうかを確認します。	<code>\$(anyMatchingAttribute(".*"):isNull())</code>
96	判定		allMatchingAttributes()	検索文字（正規表現）	全ての引数に指定された条件に一致するかどうかを確認します。	<code>\$(allMatchingAttributes(".*"):isNull())</code>
97	判定		anyDelineatedValue()	確認したい文字、引数を分割するために使用する値	引数の区切り文字に従って文字を分割し、文字を判定します。 いずれかの値が式を満たせばtrueを返します。	<code>\$(anyDelineatedValue("\${number_list}", ","):contains("5"))</code>
98	判定		allDelineatedValues()	確認したい文字、引数を分割するために使用する値	引数の区切り文字に従って文字を分割し、文字を判定します。 すべての値が式を満たせばtrueを返します。	<code>\$(allDelineatedValues("\${word_list}", ","):contains("o"))</code>
99	判定		join()	値を結合するときに使用する区切り文字	複数の値を連結する際に、引数を値の後に指定します。この関数は、allAttributes、allMatchingAttributes、およびallDelineatedValues 関数との組み合わせでのみ使用できます。	<code>\$(allAttributes("abc", "xyz"):join(" now"))</code>
100	判定		count()	なし	allAttributes、allMatchingAttributesおよびallDelineatedValuesによって返されるnull、false以外の値の数をカウントする集計関数。	<code>\$(allMatchingAttributes("\${ax}.*"):substringBefore(" ").count())</code>