DekiScript Functions and Variables

DekiScript Developer Documentation

Home Beginner's Guide Tutorials/Samples/FAQ Reference

Add Function/Variable

DekiScript Runtime Function & Variables

The following functions and variables are part the DekiScript runtime environment and are always available.

1. DekiScript Runtime Function & Variables
   1.1. Culture Functions
   1.2. Date Functions
   1.3. List Functions
   1.4. Map Functions
   1.5. Meta Functions
   1.6. Number Functions
   1.7. String Functions
   1.8. Uri Functions
   1.9. Web Functions
   1.10. Xml Functions
   1.11. Misc. Functions
   1.12. Global Variables

Culture Functions

The culture functions enable conversion of the culture/language information into other representations.

Function	Description
Culture.EnglishName	Get culture name in English.
Culture.Iso2Code	Get ISO 639-1 two-letter code for culture.
Culture.Iso3Code	Get ISO 639-2 three-letter code for culture.
Culture.NativeName	Get culture name in its own language.

Date Functions

Date functions enable manipulation of date-time strings, such simple arithmetic, formatting, and extracting individual components.

Function	Description
Date.AddDays	Adds the specified number of days to a date and returns the result. The number of days can be positive or negative.
Date.AddHours	Adds the specified number of hours to a date and returns the result. The number of hours can be positive or negative.
Date.AddMinutes	Adds the specified number of minutes to a date and returns the result. The number of minutes can be positive or negative.
Date.AddMonths	Adds the specified number of months to a date and returns the result. The number of months can be positive or negative.
Date.AddSeconds	Adds the specified number of seconds to a date and returns the result. The number of seconds can be positive or negative.
Date.AddWeeks	Adds the specified number of weeks to a date and returns the result. The number of weeks can be positive or negative.
Date.AddYears	Adds the specified number of years to a date and returns the result. The number of years can be positive or negative.
Date.ChangeTimezone	Change time-zone of date-time.
Date.Compare	Compare the first date-time to the second date-time.
Date.Date	Display the day, month, and year portion of a date.
Date.Day	Display the day of month from a specified date.
Date.DayName	Get name of day from a specified date.
Date.DayOfWeek	Display the day of week from a specified date.
Date.DayOfYear	Display the day of year from a specified date.
Date.DaysInMonth	Get the number of days in date-time month.
Date.DiffDays	Compute the difference between the first and second date in days.
Date.DiffHours	Compute the difference between the first and second date in hours.
Date.DiffMinutes	Compute the difference between the first and second date in minutes.

Date.DiffMonths	Compute the difference between the first and second date in months.
Date.DiffSeconds	Compute the difference between the first and second date in seconds.
Date.Format	Formats a date according to the specified format string. See .Net documentation for formatting date-time for more information on format strings.
Date.Hours	Get the hours component of date-time.
Date.InRange	Check if date-time occurs between lower and upper date-time, inclusive.
Date.IsAfter	Evaluates two dates and returns true if the first date is after the second date.
Date.IsBefore	Evaluates two dates and returns true if the first date is before the second date.
Date.ISOWeek	Takes a standard date string and returns the numeric values of an ISO 8601 week-date as a map.
Date.IsSameDay	Evaluates two dates and returns true if the first date and the second date fall on the same day.
Date.IsSameMonth	Check if first date-time and the second date-time fall on the same month.
Date.IsSameWeek	Check if first date-time and the second date-time fall on the same week.
Date.IsValid	Check if the string is a valid date-time.
Date.Minutes	Get the minutes component of date-time.
Date.Month	Display the month from a specified date.
Date.MonthName	Get name of month from a specified date.
Date.New	Create a new date-time value.
Date.Now	Retrieves the current date and time in GMT format.
Date.Parse	Parse a custom date-time string.
Date.ParseISOWeek	Takes a full ISO 8601 week-date (yyyy-Www-d) and translates it back to a standard date string.
Date.Parts	Decompose a date-time into its components parts.
Date.Seconds	Get the seconds component of date-time.
Date.StartOfDay	Compute the date corresponding to the beginning of day (i.e. midnight).

Date.StartOfMonth	Compute the date corresponding to the beginning of month.
Date.StartOfWeek	Compute the date corresponding to the beginning of week (i.e. Sunday).
Date.StartOfYear	Compute the date corresponding to the beginning of year.
Date.Time	Display the time portion of a specified date.
Date.Timezone	Get the time-zone component of date-time.
Date.TimezoneNow	Current date-time in time-zone format.
Date.Week	Display the week from a specified date.
Date.Year	Display the year from a specified date.

List Functions

List functions enable manipulation of lists, such as applying a code fragment to each value or sorting values in ascending order.

Function	Description
List.Apply	Create a new list by applying the expression to each item.
List.Average	Get average of values in list.
List.Collect	Collect all values from each map inside the list.
List.Contains	Check if list contains the given value.
List.GroupBy	Group items in list by common expression value.
List.IndexesOf	Find all index position of the given value.
List.IndexOf	Find the index position of the given value.
List.LastIndexOf	Find the last index position of the given value.
List.Max	Get largest value in list.
List.Min	Get smallest value in list.
List.New	Create a list of a given size.
List.Orderby	Sort items in list of maps using key names.
List.Random	Get a random item from the list.

List.Reduce	Combine all values in list into a single value using the supplied expression.
List.Reverse	Reverse the entries in a list.
List.Select	Create a list that only contains values for which the condition succeeds.
List.Sort	Sort the entries in a list.
List.Splice	Remove the entries in a list and add from another list.
List.Sum	Get sum of values in list.

Map Functions

Map functions enable manipulation of maps, such as extracting the keys or values.

Function	Description
Map.Apply	Create a new map by applying the expression to each key value pair.
Map.Contains	Check if the key is contained in a map.
Map.Keys	Get the list of keys of a map.
Map.KeyValues	Create a list of the key-value pairs.
Map.Select	Create a map that only contains key-value pairs for which the condition succeeds.
Map.Values	Get the list of values of a map.

Meta Functions

Meta functions enable embedding of <meta> elements in the <head> section of the current document. Requires that the page has UNSAFECONTENT permission.

Function	Description
Meta.Author	Add the author's name to the page.
Meta.Copyright	Add a copyright statement to the page.
Meta.Custom	Add a custom meta tag to the page.
Meta.Description	Add a plain language description to the page.
Meta.Googlebot	Add control statements for Google's search engine to the page.
Meta.Keywords	Add keywords for search engines to the page.

Meta.Rating	Add simple content rating to the page.
Meta.Refresh	Add an automatic reload/redirect to the page.
Meta.Robots	Add control statements for search engines to the page.

Number Functions

Number functions enable manipulation of numeric values, such as trigonometric functions and formatting.

Function	Description
Num.Abs	Get absolute value of number.
Num.Acos	Get the angle whose cosine is the specified number.
Num.Asin	Get the angle whose sine is the specified number.
Num.Atan	Get the angle whose tangent is the specified number.
Num.Atan2	Get the angle whose tangent is the quotient of two specified numbers.
Num.Cast	This function takes in one argument and tries to cast it to a number. The function returns nil if it can't cast the value to a number.
Num.Ceiling	Get the smallest integer greater or equal to the specified number.
Num.Cos	Get the cosine of the specified angle.
Num.Cosh	Get the hyperbolic cosine of the specified angle.
Num.E	Natural logarithmic base.
Num.Epsilon	Smallest positive number that greater than zero.
Num.Exp	Get e raised to the specified number.
Num.Floor	Get the largest integer less or equal to the specified number.
Num.Format	Show number in custom format. See .Net documentation for formatting numbers.
Num.Int	Get the integral part of the specified number.
Num.IsInfinity	Check if number is infinite.
Num.IsNan	Check if number is not-a-number (NaN).
Num.IsNegativeInfinity	Check if number is negative infinity.

Num.IsPositiveInfinity	Check if number is positive infinity.
Num.Log	Get the logarithm of the specified number in the specified base.
Num.Max	Get the larger of two numbers.
Num.Min	Get the smaller of two numbers.
Num.Nan	Not a number value (NaN).
Num.NegativeInfinity	Negative infinity.
Num.Pi	Ratio of the circumference of a circle to its diameter.
Num.PositiveInfinity	Positive infinity.
Num.Pow	Get the specified number raised to the specified power.
Num.Round	Round to nearest number with given precision.
Num.Series	Create a series of numbers.
Num.Sign	Get a value indicating the sign of the number.
Num.Sin	Get the sine of the specified angle.
Num.Sinh	Get the hyperbolic sine of the specified angle.
Num.Sqrt	Get the square root of the specified number.
Num.Tan	Get the tangent of the specified angle.
Num.Tanh	Get the hyperbolic tangent of the specified angle.

String Functions

String function enable manipulation of string values, such as capitalization, checking for equality, and formatting.

Function	Description
String.Compare	Compare the first string to the second string.
String.Contains	Check if the first string contains the second string.
String.Deserialize	Convert a string into a value. This function should be used in conjunction with string.Serialize.
String.EndsWith	Check if the first string ends with the second string.

String.Escape	Escape special characters in string.
String.Eval	Evaluate a DekiScript expression.
String.Hash	Compute MD5 hash of string.
String.IndexesOf	Get all indexes of the second string in the first string.
String.IndexOf	Get first index of the second string in the first string. The index is zero-based.
String.Insert	Insert the second string into the first string at the specified index
String.IsControl	Check if first character in string is a control character.
String.IsDigit	Check if first character in string is a decimal digit.
String.IsLetter	Check if first character in string is an alphabetic character.
String.IsLetterOrDigit	Check if first character in string is a an alphabetic character or decimal digit.
String.IsLower	Check if first character in string is lowercase.
String.IsNumber	Check if first character in string is a number.
String.IsPunctuation	Check if first character in string is a punctuation mark.
String.IsSeparator	Check if first character in string is a separator character.
String.IsSymbol	Check if first character in string is a symbol character.
String.IsUpper	Check if first character in string is uppercase.
String.IsWhiteSpace	Check if first character in string is white space.
String.Join	Join strings in a list using a specified separator.
String.LastIndexOf	Get last index of the second string in the first string.
String.Length	Get the length of the string.
String.Match	Match a string against a .NET regular expression pattern and return the first capture.
String.Matches	Match a string against a .NET regular expression pattern and return all captured parts.
String.PadLeft	Pad string with characters on the left until it reaches the specified width.
String.PadRight	Pad string with characters on the right until it reaches the specified width.

String.Quote	Quote string and escape characters.
String.Remove	Remove characters from a string at the specified index.
String.Replace	Replaces all occurrences of a string with another string.
String.SearchEscape	Escape special characters in string for use as a in lucene query.
String.Serialize	Convert a value into a string. This function should be used in conjunction with string.Deserialize.
String.Split	Split string at each occurrence of the separator up to the specified limit.
String.SplitCsv	Convert string containing a comma separated list of values into a collection of rows and columns.
String.SqlEscape	Escape special SQL characters in string.
String.StartsWith	Check if the first string starts with the second string.
String.SubStr	Extract a substring.
String.ToCamelCase	Convert first charater of each word in string to uppercase.
String.ToLower	Convert string to lowercase characters.
String.ToLowerFirst	Convert first character in string to lowercase.
String.ToUpper	Convert string to uppercase characters.
String.ToUpperFirst	Convert first character in string to uppercase.
String.Trim	Remove whitespace from the beginning and end of the string.
String.TrimEnd	Remove whitespace from the end of the string.
String.TrimStart	Remove whitespace from the beginning of the string.

Uri Functions

Uri functions enable manipulation of uri values, such as appending query parameters and decoding strings string in uri format.

Function	Description
Uri.AppendPath	Append path segments to a URI.
Uri.AppendQuery	Append query parameters to a URI.

Uri.Build	Build a new URI with path and query parameters.
Uri.Decode	Decode text as a URI component.
Uri.Encode	Encode text as a URI component.
Uri.IsValid	Check if a value is a valid URI.
Uri.Parse	[OBSOLETE: Use Uri.Parts] Parse a URI into its component parts.
Uri.Parts	Decompose a URI into its component parts. (Obsoletes uri.parse)

Web Functions

Web functions enable manipulation of web entities, such as HTML fragments and images, as well as fetching information from web sources in text or XML format.

Function	Description
Web.CheckStyle	Check contents of style attribute for possible XSS vulnerabilities.
Web.CheckUri	Check URI for possible JavaScript code.
Web.Html	Convert source to HTML by optionally fetching and parsing it.
Web.IFrame	Embed an <iframe> element. (OBSOLETE: this function is no longer available)
Web.Image	Insert an image.
Web.Json	Parse JSON source into a DekiScript data-structure.
Web.Link	Insert a hyperlink.
Web.List	Convert source to a list of text values by optionally fetching and parsing it.
Web.Pre	Insert text as a <pre> element.
Web.Size	Convert numeric size to text.
Web.Style	Create text for a 'style' attribute.
Web.Text	Convert source to text by optionally fetching and parsing it.
Web.Toggle	Will create a section containing the content that can be hidden/unhidden by clicking on the title.
Web.UriEncode	Encode text as a URI component.
Web.Xml	Convert source to HTML by optionally fetching and parsing it.

Xml Functions

Xml functions enable manipulation of XML documents, such as extracting a text value or selecting a specifix XML node using an XPath expression.

Function	Description
Xml.Format	Convert the XML document to a formatted string. Xml.Format provides three formatting options: plain, xhtml, and indented.
Xml.Html	Extract the inner portion of xml nodes into a new xml document.
Xml.List	Get list of string values from an XML document using an XPath expression.
Xml.Name	Get the name of the root node in the XML document.
Xml.Num	Get a number from an XML document.
Xml.Select	Create an XML selection using an xpath.
Xml.Text	Get the the text value of an XML node. The XML node can be optionally selected using an XPath expression with XML namespaces. Xml.Text captures the text value of its first child node, which must be a text node. This means that if the XML node has mixed content (i.e. sub-elements and text nodes), then only the text up to the first sub-element is returned. Alternatively, Xml.Text can return the XML string representation of the selected node.

Misc. Functions

Function	Description
Json.Emit	Convert DekiScript value to JavaScript Object Notation (JSON) value.
Json.Format	Render the Json data.

Global Variables

Variable	Description
__count	Iteration counter used by the *foreach* statement and the <foreach> element. Only increments when the condition for the *foreach* body is successful. (8.05.1 or later)
__env	Entire environment accessible from DekiScript code.
__index	Iteration counter used by the *foreach* statement and the <foreach> element. Increments for each item in the *foreach* statement. (8.05.2 or later)
__request	Holds the "args" map of arguments passed to the page (i.e. - en/docs/DekiScript/Reference/DekiScript_Functions_and_Variables?hello=world will render {args:{hello:"world" } }). As of 9.08.1, it also holds the "referer" and "useragent" information sent in by the request.