

JS Lib

Return Types

[Array](#) [Boolean](#) [Date](#) [Math](#) [Namespace](#) [Number](#) [Object](#) [QName](#) [RegExp](#) [Special Operators](#) [Statements](#) [String](#) [XML](#) [XMLList](#)

Property Summery

[Number](#) [#Infinity](#)
Numeric value representing infinity.

[Number](#) [#NaN](#)
Value representing Not-a-Number.

[Object](#) [#undefined](#)
The value undefined.

Method Summery

[String](#) [#decodeURI\(\)](#)
Decodes a URI previously encoded with encodeURI or another similar routine.

[String](#) [#decodeURIComponent\(encodedURI\)](#)

[String](#) [#encodeURI\(\)](#)
Encodes a URI by replacing certain characters with escape sequences.

[String](#) [#encodeURIComponent\(\)](#)

[Object](#) [#eval\(\)](#)
Evaluates JavaScript code passed as a string.

[Boolean](#) [#isFinite\(\)](#)
Returns true if the given number is a finite number.

[void](#) [#isNaN\(\)](#)
The NaN property indicates that a value is 'Not a Number'.

[Boolean](#) [#isXMLName\(\)](#)
Returns true if the given name can be used as a valid name for an XML element or attribute.

[void](#) [#parseFloat\(\)](#)
Makes a floating point number from the starting numbers in a given string.

[void](#) [#parseInt\(\)](#)
Makes a integer from the starting numbers in a given string in the base specified.

[String](#) [#uneval\(\)](#)
Returns the string representation behind a given object.

Property Details

Infinity

Numeric value representing infinity.

Returns

[Number](#)

Sample

```
Infinity
```

NaN

Value representing Not-a-Number.

Returns

[Number](#)

Sample

```
NaN
```

undefined

The value undefined.

Returns

[Object](#)

Sample

```
undefined
```

Method Details

decodeURI

[String](#) **decodeURI()**

Decodes a URI previously encoded with encodeURI or another similar routine.

Returns

[String](#)

Sample

decodeURIComponent

[String](#) **decodeURIComponent(encodedURI)**

Replace with description

Parameters

{[String](#)} encodedURI

Returns

[String](#)

Sample

encodeURI

[String](#) **encodeURI()**

Encodes a URI by replacing certain characters with escape sequences.

Returns

[String](#)

Sample

encodeURIComponent

[String](#) **encodeURIComponent()**

Replace with description

Returns

[String](#)

Sample

eval

Object eval()

Evaluates JavaScript code passed as a string. Returns the value returned by the evaluated code.

Returns

Object

Sample

```
eval("var x = 2 + 3;");  
application.output(x); // prints: 5.0
```

isFinite

Boolean isFinite()

Returns true if the given number is a finite number.

Returns

Boolean

Sample

```
application.output(isFinite(1)); // prints: true  
application.output(isFinite(Infinity)); // prints: false  
application.output(isFinite(isNaN)); // prints: false
```

isNaN

void **isNaN()**

The NaN property indicates that a value is 'Not a Number'.

Returns

void

Sample

```
isNaN( value )
```

isXMLName

Boolean isXMLName()

Returns true if the given name can be used as a valid name for an XML element or attribute.

Returns

Boolean

Sample

```
application.output(isXMLName("good_name")); // prints: true  
application.output(isXMLName("bad name")); // because of the space, prints: false
```

parseFloat

void **parseFloat()**

Makes a floating point number from the starting numbers in a given string.

Returns

void

Sample

```
parseFloat('string')
```

parseInt

void **parseInt()**

Makes an integer from the starting numbers in a given string in the base specified.

Returns

void

Sample

```
parseInt( 'string' [, base] )
```

uneval

String uneval()

Returns the string representation behind a given object.

Returns

String

Sample

```
application.output(uneval(isNaN)); // prints something like: function isNaN() { [native code for isNaN,
arity=1] }
```