
Xml 1.1

September 9, 2024



Contents

Xml	2
Load xml	2
Load XML from url	2
Load XML from JSON string	3
XmlDoc	3
XPath	3
Xml to JSON	3
XmlDoc to XML string	4
Modify an XML document	4

Xml

The Xml module enables parsing information stored in local or remote xml files.

Load xml

Parse the given string as xml and return an XmlDocument object which can be queried or turned into JSON.

Parameters

- `xml` an xml formatted string to parse

Example

```
1 var d = Xml.load("<hello>world</hello>");
```

Load XML from url

Fetch a local or a remote file and parse as xml. Returns an XmlDocument object.

Parameters

- `url` is a local or remote path to an xml file

Example

```
1 // A remote file
2 var remote = Xml.loadFrom("http://somewhere/over/the/rainbow.xml");
3 // A local file
4 var local = Xml.loadFrom("c:\\somewhere\\over\\the\\rainbow.xml");
```

Load XML from JSON string

You can use the `loadJson` function to parse a JSON string into an `XmlDoc` object. It should have the same structure as the output of the `Xml.toJson` function.

```
1 var json = JSON.stringify({ a: { b: 2 }, c: 3 });
2 var doc = Xml.loadJson(json);
```

XmlDoc

An `XmlDoc` is an object that wraps an xml document and which has a few functions for querying the underlying document.

XPath

Execute an XPath query and return the results. The result is a list of objects, each object represents the matching xml node.

Parameters

- `xpath` a well-formed XPath expression

Example

```
1 var doc = Xml.load("<hello>world</hello>");
2 var allHellos = doc.xpath("//hello");
```

Xml to JSON

Returns a JSON/JavaScript version of the document which can then be inspected in the flow.

Example

```
1 var doc = Xml.load("<hello>world</hello>");
2 var docObject = doc.json();
```

XmlDoc to XML string

Use the `toString` function to return the xml document as a string.

```
1 var doc = Xml.load("<hello>world</hello>");
2 var xmlString = doc.toString(); // "<hello>world</hello>"
```

Modify an XML document

The `walk` function implements a visitor-pattern for the underlying xml document. It can be used to modify the document. You supply it with a function that may either return a new value as an object or xml string or a boolean value where **false** indicates that the node should be removed. The function will get called for each node in the document.

```
1 var doc = Xml.load("<root><hello>world</hello></root>");
2 doc.walk(function(node) {
3     if (node.hello) {
4         return "<hello>universe</hello>";
5     }
6 });
7 var greeting = doc.toString(); // "<root><hello>universe</hello></root>"
```