

XmlNode

Method Summary

`String[]` [#getAttributeNames\(\)](#)
Return all the attribute names of the current node.

`String` [#getAttributeValue\(\)](#)
Return the value of the named attribute.

`XmlNode[]` [#getChildNodes\(\)](#)
Return the child nodes of the current node.

`String` [#getName\(\)](#)
Return the name of the XML node element.

`String` [#getTextValue\(\)](#)
Return the text-value of the XML node element.

`String` [#getType\(\)](#)
Return the type of the XML node element.

Method Details

[getAttributeNames](#)

`String[]` **[getAttributeNames\(\)](#)**

Return all the attribute names of the current node.

Returns

`String[]`

Sample

```
nodes = plugins.XmlReader.readXmlDocumentFromString("<root attr1='value1' attr2='value2'/>")
rootNode = nodes[0];
attributes = rootNode.getAttributeNames();
application.output(attributes[0])
application.output(attributes[1])
vall = rootNode.getAttributeValue('attr1');
application.output(vall)
```

[getAttributeValue](#)

`String` **[getAttributeValue\(\)](#)**

Return the value of the named attribute.

Returns

`String`

Sample

```
nodes = plugins.XmlReader.readXmlDocumentFromString("<root attr1='value1' attr2='value2'/>")
rootNode = nodes[0];
attributes = rootNode.getAttributeNames();
application.output(attributes[0])
application.output(attributes[1])
vall = rootNode.getAttributeValue('attr1');
application.output(vall)
```

[getChildNodes](#)

`XmlNode[]` **[getChildNodes\(\)](#)**

Return the child nodes of the current node.

Returns

`XmlNode[]`

Sample

```
nodes = plugins.XmlReader.readXmlDocumentFromString('<father><child1>John</child1><child2>Mary</child2></father>');
application.output(nodes[0].getName())
application.output(nodes[0].getTextValue())
application.output(nodes[0].getType())
childs = nodes[0].getChildNodes()
application.output(childs[0].getName())
application.output(childs[0].getTextValue())
application.output(childs[0].getType())
subChilds = childs[0].getChildNodes()
application.output(subChilds[0].getName())
application.output(subChilds[0].getTextValue())
application.output(subChilds[0].getType())
```

getName

String getName()

Return the name of the XML node element.

Returns

String

Sample

```
nodes = plugins.XmlReader.readXmlDocumentFromString('<father><child1>John</child1><child2>Mary</child2></father>');
application.output(nodes[0].getName())
application.output(nodes[0].getTextValue())
application.output(nodes[0].getType())
childs = nodes[0].getChildNodes()
application.output(childs[0].getName())
application.output(childs[0].getTextValue())
application.output(childs[0].getType())
subChilds = childs[0].getChildNodes()
application.output(subChilds[0].getName())
application.output(subChilds[0].getTextValue())
application.output(subChilds[0].getType())
```

getTextValue

String getTextValue()

Return the text-value of the XML node element.

Returns

String

Sample

```
nodes = plugins.XmlReader.readXmlDocumentFromString('<father><child1>John</child1><child2>Mary</child2></father>');
application.output(nodes[0].getName())
application.output(nodes[0].getTextValue())
application.output(nodes[0].getType())
childs = nodes[0].getChildNodes()
application.output(childs[0].getName())
application.output(childs[0].getTextValue())
application.output(childs[0].getType())
subChilds = childs[0].getChildNodes()
application.output(subChilds[0].getName())
application.output(subChilds[0].getTextValue())
application.output(subChilds[0].getType())
```

getType

String getType()

Return the type of the XML node element.

Returns

String

Sample

```
nodes = plugins.XmlReader.readXmlDocumentFromString('<father><child1>John</child1><child2>Mary</child2></father>');
application.output(nodes[0].getName())
application.output(nodes[0].getTextValue())
application.output(nodes[0].getType())
childs = nodes[0].getChildNodes()
application.output(childs[0].getName())
application.output(childs[0].getTextValue())
application.output(childs[0].getType())
subChilds = childs[0].getChildNodes()
application.output(subChilds[0].getName())
application.output(subChilds[0].getTextValue())
application.output(subChilds[0].getType())
```