

# An XML Interchange Format for Wiki Creole 1.0

MARTIN JUNGHANS, DIRK RIEHLE, AND UMIT YALCINALP

SAP Research, SAP Labs LLC  
3475 Deer Creek Rd  
Palo Alto, CA 94304 U.S.A.

---

Wikis have become an important application on the web and in the enterprise, yet there are no interoperability standards between different wiki engines. We present the first complete XML representation format of Wiki Creole 1.0. Wiki Creole is a community standard for wiki markup, the language used to write wiki pages. This report presents the complete XML representation format using a validating XML schema. In addition we present XSLT definitions for transforming the XML representations to XHTML on the one hand and for transforming the XML representations to Wiki Creole markup on the other hand. Our work shows how using XML technologies we can make wiki interchange, wiki upgrading, and wiki conversion independent from a specific wiki engine implementation.

---

## 1. INTRODUCTION

Wikis were invented in 1995 [2]. They have become a widely used tool on the web and in the enterprise since then [1]. In the form of Wikipedia, wikis are having a significant impact on society [8]. Many different wiki engines have been implemented since the first wiki was created. All of these wiki engines integrate the core page rendering engine, its storage backend, the processing tools, and the page editor in one software package.

Wiki pages are written in wiki markup. Almost all wiki engines define their own markup language. Different software components of the wiki engine like the page rendering part are tied to that particular markup language. In addition, users have to learn different wiki markup languages if they want to work with different wiki engines. Also, corporate IT departments have to implement custom migration tools if they want to switch from one wiki engine to another.

In a related article, we present an EBNF grammar [4] [5] for Wiki Creole [7], the first (and so far only) wiki markup standard. An agreed-upon wiki markup standard has many benefits; most notably, it allows users to switch between different wiki engines more easily. (Users don't have to learn different wiki markup syntaxes each time they start using a new wiki software.)

However, it should not only be easy for humans to work with different wiki engines, it should also be easy for machines to connect different wiki engines. Voelkel and Oren's work on a wiki interchange format discusses the problem of data islands that are not interoperable and that are closed for machine processing [9].

Our work tries to remedy the situation by bringing XML technologies to wikis. In contrast to prior work [9] [10], the work presented in this article is the first that

- is complete,
- is based on a community standard, and
- is based on a formal specification of the underlying wiki markup.

To the best of our knowledge, earlier work was never completed, was not based on any agreed-upon standard, and had no underpinning in a precise language definition.

## 2. BENEFITS OF A WIKI INTERCHANGE STANDARD

An interchange representation of wiki markup lets software developers decouple components of a wiki engine like the rendering engine, storage and export, and the editing component. These different components can then be developed independently and at different speeds, making it easier to innovate.

An interchange format is necessary to support the following use cases and features of wiki engine software:

- *Interchange.* The interchange format as an intermediate representation for wiki markup simplifies the process of exchanging wiki contents between wiki engines. Two steps are needed to exchange wiki contents: In the first step, the source engine translates the wiki contents to the interchange format and exports it. In the second step, the target engine imports the wiki contents and converts the intermediate representation to its internal representation. The interchange format acts as the conversion hub. Rather than having to consider different formats of other wiki engines, wiki engine implementers have to deal with only one format.
- *Migration.* Similar to the interchange scenario, an intermediate representation format for wiki contents makes it easier to migrate from a source markup standard to an enhanced markup standard, for example, to migrate from Wiki Creole 0.6 to 1.0. Wiki engines can either directly work with the interchange format as their internal format or they can export-then-import wiki contents to migrate the contents from one wiki markup to another.

- *Extensibility.* Enterprises frequently need to integrate data from backend databases into wiki pages dynamically. To bring in foreign data into a wiki, the data from a backend is best translated to the intermediate format from which it is imported into the wiki page or the wiki's database. Similar to the migration scenario, such an extension can deliver data to other wiki engines as well because it returns data independently from the wiki markup language.
- *Performance.* Most wiki engines today still store and work with markup rather than more structured data. Using an interchange format as the internal representation format (rather than markup) is likely to lead to performance gains as the parsing of markup can be dropped or significantly reduced, and more time remains for the actual rendering of the wiki page. With an interchange format, such rendering doesn't even have to take place any longer on the server but can be pushed out to a client like a browser.
- *Usability.* Editing a wiki page will be easier for users if they could choose a markup language that they already know. With a standardized interchange format, wiki engines can become agnostic of a specific markup and leave it to a client-side editor to display a wiki page using the user's markup of choice.

No wiki interchange format will be able to comprise all the possible features that different markup syntaxes allow users to express. Hence, we also need to allow for extension with unknown markup elements that can be passed on without being mangled.

#### **4. XML TECHNOLOGIES AND WIKI CREOLE 1.0**

We choose XML as the representation format for wiki pages, because of the availability of mature related technologies like XSLT and schema validation [11]. Also, we chose Wiki Creole as the markup standard to base our work on because it is the only existing standard. Wiki Creole is also the only standard with a well-defined grammar.

- Appendix A shows an XML schema definition that defines the structure and layout of XML documents representing wiki pages. The capabilities are based on our formalization of Wiki Creole 1.0 in [5].
- Appendix B shows an XSLT stylesheet that can be used by an XSLT processor to transform Wiki Creole 1.0 XML documents into XHTML for visual display in a browser or similar device.

- Appendix C shows an XSLT stylesheet that can be used by an XSLT processor to transform Wiki Creole 1.0 XML documents into Wiki Creole 1.0 markup.

These specifications were experimentally validated through appropriate implementations.

We defined context-free grammars for Wiki Creole in version 0.6 and 1.0 [5] and generated a scanner and parser for Wiki Creole markup using ANTLR [6], a Java parser generator. The parser checks the validity of some markup towards the specification and creates a parse tree for each wiki page. This parse tree is traversed and the proper XML document is generated.

The parse tree returned by the ANTLR parser is a Java data structure. We implemented a tree walker that performs a depth first search and visits all nodes. The leaves of the tree represent tokens, either wiki markup or regular text content. The markup is translated to the according representation of XML tags. In most cases the translation is straight forward, for instance opening and closing bold markup are translated to opening and closing XML tags for bold text formatting. The remaining regular text content is applied without changes.

The structure and contents of the XML documents are defined by an XML schema. It comprises the Wiki Creole standard markup specification in version 0.6 and 1.0. The XML schema is dependent on the Wiki Creole specification, so future releases of Wiki Creole are likely to lead to new releases of the XML schema presented here.

The XML schema in Appendix A describes a wiki as a sequence of wiki pages. A wiki page has the attribute “name” and is a sequence of paragraphs, i.e. an arbitrary sequence of elements like text, tables, lists, nowiki blocks, headings. The `<xsd:choice>` element does not restrict the order or the amount of elements by setting the `minOccurs` and `maxOccurs` attributes to zero and unbounded respectively.

Once an XML interchange representation has been generated, we can use XSLT stylesheets to transform those wiki pages to XHTML for rendering and to other wiki markup languages for migration.

When transforming more powerful to less powerful markup languages, care needs to be taken not to loose information, if features are not supported in the target markup language. The interchange format is responsible to store all data, even if some target markup languages are not able to handle all features. The XSLT stylesheets have to consider capabilities of the target language, for example, by using meta-data of documents or omitting these parts in the target markup.

### 3. SUMMARY AND CONCLUSIONS

This report presents an XML schema that defines how valid XML documents have to look like if they are to represent Wiki Creole 1.0 markup. Thus, for the first time, an XML representation format for wiki contents has been defined that is complete and based on a wiki standard. The schema is based on our earlier work on a context-free grammar for the Wiki Creole markup language. Thus, the XML representation is also the first specification based on a formally defined language. In addition to the core XML format we present XSL transformations that let users transform the XML either into XHTML for visual display in a browser or into the original Wiki Creole 1.0 markup that users can edit. Using implementations and test cases we have validated these specifications to show that they fulfill on the purposes of wiki interchange, migration, and extension.

### REFERENCES

- [1] GOOGLE TRENDS. *Trends on Wiki Keyword Search*. See <http://www.google.com/trends?q=wiki>
- [2] LEUF, B., AND CUNNINGHAM, W., 1999. *The Wiki Way: Quick Collaboration on the Web*. Addison-Wesley.
- [3] JUNGHANS, M., AND RIEHLE, D. 2007. *Wiki Creole 1.0 EBNF Grammar*. See <http://www.wikimarkup.org>. Web-published.
- [4] JUNGHANS, M., RIEHLE, D., GURRAM, R., KAISER, M., LOPES, M., AND YALCINALP, U. 2008. A Grammar for Standardized Wiki Markup. In submission. Please contact the authors for a copy.
- [5] JUNGHANS, M., RIEHLE, D., GURRAM, R., KAISER, M., LOPES, M., AND YALCINALP, U. 2007. "An EBNF Grammar for Wiki Creole 1.0." *SIGWEB Newsletter Winter 2007*. ACM, 2007.
- [6] PARR, T. 2007. *ANTLR 3 Grammar Syntax*. See <http://www.antlr.org/wiki/display/ANTLR3/Grammars>. Web-published.
- [7] WIKI CREOLE 2007. *Wiki Creole 1.0 Specification*. See <http://www.wikicreole.org/wiki/Creole1.0>. Web-published.
- [8] WIKIPEDIA, ENGLISH. See <http://en.wikipedia.org>. Web-published.
- [9] VOELKEL, M. AND OREN, E. 2006. *Towards a Wiki Interchange Format*. In Proceedings of the First Workshop on Semantic Wikis. Web-published.
- [10] VOSS, J. 2007. *Wikipedia DTD*. See [http://meta.wikimedia.org/w/index.php?title=Wikipedia\\_DTD](http://meta.wikimedia.org/w/index.php?title=Wikipedia_DTD). Web-published, 2007.
- [11] W3C 1999. XSL Transformations. See <http://www.w3.org/TR/xslt>. Web-published, 1999.

---

Martin Junghans is an M.S. student at Technical University of Cottbus, Germany. The work presented in this report was performed during an internship at SAP Labs LLC in Palo Alto, California, under the supervision of Dirk Riehle.

Dirk Riehle is the principal investigator of the open source and Web 2.0 applications research group at SAP Research, SAP Labs LLC, in Palo Alto, California. He is the founder of the *International Symposium on Wikis* conference series.

Umit Yalcinalp is a researcher at SAP Research, SAP Labs LLC, in Palo Alto, California as well, and a W3C standards expert.

## APPENDIX A: XML SCHEMA FOR THE REPRESENTATION OF WIKI CRE- OLE 1.0 MARKUP AS XML DOCUMENTS

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified" attributeFormDefault="unqualified">

  <xsd:element name="wiki">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:choice minOccurs="0" maxOccurs="unbounded">
          <xsd:element name="wikipage">
            <xsd:complexType>
              <xsd:choice minOccurs="0" maxOccurs="unbounded">
                <xsd:element name="textparagraph" type="textType"/>
                <xsd:element name="table" type="tableType"/>
                <xsd:element name="unorderedlist" type="unorderedlistType"/>
                <xsd:element name="orderedlist" type="orderedlistType"/>
                <xsd:element name="nowikiblock" type="xsd:string"/>
                <xsd:element name="horizontalrule"/>
                <xsd:element name="heading">
                  <xsd:complexType mixed="true">
                    <xsd:attribute name="level" use="required"/>
                  </xsd:complexType>
                </xsd:element>
              </xsd:choice>
            <xsd:attribute name="name" use="required"/>
          </xsd:complexType>
        </xsd:element>
      </xsd:choice>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

  <xsd:complexType name="unorderedlistType" mixed="true">
    <xsd:choice minOccurs="1" maxOccurs="unbounded">
      <xsd:element name="listelem" type="textType"/>
      <xsd:element name="unorderedlist" type="unorderedlistType"/>
      <xsd:element name="orderedlist" type="orderedlistType"/>
    </xsd:choice>
  </xsd:complexType>

  <xsd:complexType name="orderedlistType" mixed="true">
    <xsd:choice minOccurs="1" maxOccurs="unbounded">
      <xsd:element name="listelem" type="textType"/>
      <xsd:element name="unorderedlist" type="unorderedlistType"/>
      <xsd:element name="orderedlist" type="orderedlistType"/>
    </xsd:choice>
  </xsd:complexType>

  <xsd:complexType name="tableType">
    <xsd:sequence>
      <xsd:element name="tablerow" minOccurs="1" maxOccurs="unbounded">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="tablecell" minOccurs="1" maxOccurs="unbounded">
              <xsd:complexType mixed="true">
                <xsd:choice minOccurs="0" maxOccurs="unbounded">
                  <xsd:element name="bold" type="boldtextType"/>
                  <xsd:element name="italic" type="italictextType"/>
                  <xsd:element name="image" type="imageType"/>
                  <xsd:element name="link" type="linkType"/>
                  <xsd:element name="nowiki" type="xsd:string"/>
                  <xsd:element name="textlinebreak"/>
                </xsd:choice>
                <xsd:attribute name="header" type="xsd:boolean" use="required"/>
              </xsd:complexType>
            </xsd:element>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:sequence>
</xsd:complexType>
```

```

<xsd:complexType name="boldtextType" mixed="true">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="italic" type="bolditalictextType"/>
    <xsd:element name="link" type="linkType"/>
    <xsd:element name="nowiki" type="xsd:string"/>
    <xsd:element name="image" type="imageType"/>
    <xsd:element name="textlinebreak"/>
    <xsd:element name="extension" type="extensionType"/>
  </xsd:choice>
</xsd:complexType>

<xsd:complexType name="italictextType" mixed="true">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="bold" type="bolditalictextType"/>
    <xsd:element name="link" type="linkType"/>
    <xsd:element name="nowiki" type="xsd:string"/>
    <xsd:element name="image" type="imageType"/>
    <xsd:element name="textlinebreak"/>
    <xsd:element name="extension" type="extensionType"/>
  </xsd:choice>
</xsd:complexType>

<xsd:complexType name="bolditalictextType" mixed="true">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="link" type="linkType"/>
    <xsd:element name="nowiki" type="xsd:string"/>
    <xsd:element name="image" type="imageType"/>
    <xsd:element name="textlinebreak"/>
    <xsd:element name="extension" type="extensionType"/>
  </xsd:choice>
</xsd:complexType>

<xsd:complexType name="linkType">
  <xsd:sequence>
    <xsd:element name="uri" type="xsd:string"/>
    <xsd:element name="description" type="simpletextType" minOccurs="0" maxOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="imageType">
  <xsd:sequence>
    <xsd:element name="uri" type="xsd:string"/>
    <xsd:element name="alternative" type="simpletextType" minOccurs="0" maxOccurs="1"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="textType" mixed="true">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="bold" type="boldtextType"/>
    <xsd:element name="italic" type="italictextType"/>
    <xsd:element name="textlinebreak"/>
    <xsd:element name="link" type="linkType"/>
    <xsd:element name="nowiki" type="xsd:string"/>
    <xsd:element name="image" type="imageType"/>
    <xsd:element name="extension" type="extensionType"/>
  </xsd:choice>
</xsd:complexType>

<xsd:complexType name="simpletextType" mixed="true">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="bold" type="boldtextType"/>
    <xsd:element name="italic" type="italictextType"/>
    <xsd:element name="image" type="imageType"/>
    <xsd:element name="textlinebreak"/>
  </xsd:choice>
</xsd:complexType>
</xsd:schema>

```

## APPENDIX B: XSLT FOR TRANSFORMING WIKI CREOLE XML DOCUMENTS TO XHTML

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:transform version="2.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="html" encoding="UTF-8" indent="yes"/>

  <xsl:template match="wikipage">
    <xsl:variable name="wpName" select="@name"/>
    <xsl:result-document href="{ $wpName }.html">
      <html>
        <head>
          <title><xsl:value-of select="$wpName"/></title>
        </head>
        <body><xsl:apply-templates/></body>
      </html>
    </xsl:result-document>
  </xsl:template>

  <xsl:template match="textparagraph">
    <p><xsl:apply-templates/></p>
  </xsl:template>

  <xsl:template match="table">
    <table border="1"><xsl:apply-templates/></table>
  </xsl:template>
  <xsl:template match="tablerow">
    <tr><xsl:apply-templates/></tr>
  </xsl:template>
  <xsl:template match="tablecell">
    <xsl:variable name="header" select="@header"/>
    <xsl:if test="$header = boolean(1)"><th><xsl:apply-templates/></th></xsl:if>
    <xsl:if test="$header=boolean(0)"><td><xsl:apply-templates/></td></xsl:if>
  </xsl:template>

  <xsl:template match="unorderedlist">
    <ul><xsl:apply-templates/></ul>
  </xsl:template>
  <xsl:template match="orderedlist">
    <ol><xsl:apply-templates/></ol>
  </xsl:template>
  <xsl:template match="listelem">
    <li><xsl:apply-templates/></li>
  </xsl:template>

  <xsl:template match="nowikiblock">
    <pre><xsl:apply-templates/></pre>
  </xsl:template>

  <xsl:template match="horizontalrule">
    <hr />
  </xsl:template>

  <xsl:template match="heading">
    <xsl:variable name="level" select="@level"/>
    <xsl:text disable-output-escaping="yes">&lt;h</xsl:text>
    <xsl:if test="$level > 6"><xsl:text>6</xsl:text></xsl:if>
    <xsl:if test="$level &lt;= 6"><xsl:value-of select="$level"/></xsl:if>
    <xsl:text disable-output-escaping="yes">&gt;</xsl:text>
    <xsl:apply-templates/>
    <xsl:text disable-output-escaping="yes">&lt;/h</xsl:text>
    <xsl:if test="$level > 6"><xsl:text>6</xsl:text></xsl:if>
    <xsl:if test="$level &lt;= 6"><xsl:value-of select="$level"/></xsl:if>
    <xsl:text disable-output-escaping="yes">&gt;</xsl:text>
  </xsl:template>

  <xsl:template match="bold">
    <strong><xsl:apply-templates/></strong>
  </xsl:template>
  <xsl:template match="italic">
    <em><xsl:apply-templates/></em>
  </xsl:template>

  <xsl:template match="nowiki">
    <tt><xsl:apply-templates/></tt>
  </xsl:template>

  <xsl:template match="textlinebreak">
    <xsl:text disable-output-escaping="yes">&lt;br /&gt;</xsl:text>
  </xsl:template>

```



```
<xsl:template match="link[description]">
  <xsl:text disable-output-escaping="yes">&lt;a href=&quot;</xsl:text>
  <xsl:value-of select="uri"/>
  <xsl:text disable-output-escaping="yes">&quot;&gt;</xsl:text>
  <xsl:apply-templates select="description"/>
  <xsl:text disable-output-escaping="yes">&lt;/a&gt;</xsl:text>
</xsl:template>

<xsl:template match="link">
  <xsl:text disable-output-escaping="yes">&lt;a href=&quot;</xsl:text>
  <xsl:value-of select="uri"/>.html
  <xsl:text disable-output-escaping="yes">&quot;&gt;</xsl:text>
  <xsl:value-of select="uri"/>
  <xsl:text disable-output-escaping="yes">&lt;/a&gt;</xsl:text>
</xsl:template>

<xsl:template match="image[alternative]">
  <xsl:text disable-output-escaping="yes">&lt;img src=&quot;</xsl:text>
  <xsl:value-of select="uri"/>
  <xsl:text disable-output-escaping="yes">&quot; alt=&quot;</xsl:text>
  <xsl:apply-templates select="alternative"/>
  <xsl:text disable-output-escaping="yes">&quot;/&gt;</xsl:text>
</xsl:template>

<xsl:template match="image">
  <xsl:text disable-output-escaping="yes">&lt;img src=&quot;</xsl:text>
  <xsl:value-of select="uri"/>
  <xsl:text disable-output-escaping="yes">&quot;/&gt;</xsl:text>
</xsl:template>
</xsl:transform>
```

## APPENDIX C: XSLT FOR TRANSFORMING WIKI CREOLE XML DOCUMENTS TO WIKI CREOLE MARKUP

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:transform version="2.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:WikiCreole="http://www.wikicreole.org/ns/creole">
  <xsl:output method="text" encoding="UTF-8" indent="yes"/>

  <xsl:template match="WikiCreole:wikipage">
    <xsl:variable name="wpName" select="@name"/>
    <xsl:result-document href="{ $wpName }">
      <xsl:apply-templates/>
    </xsl:result-document>
  </xsl:template>

  <xsl:template match="WikiCreole:textparagraph">
    <xsl:apply-templates/>
    <!-- one blank line (two linefeeds) after paragraph as paragraph separator -->
    <xsl:text>

  </xsl:text>
</xsl:template>

  <xsl:template match="WikiCreole:bold">
    <xsl:text>**</xsl:text>
    <xsl:apply-templates/>
    <xsl:text>**</xsl:text>
  </xsl:template>

  <xsl:template match="WikiCreole:italic">
    <xsl:text>//</xsl:text>
    <xsl:apply-templates/>
    <xsl:text>//</xsl:text>
  </xsl:template>

  <xsl:template match="WikiCreole:nowiki">
    <xsl:text>{{</xsl:text>
    <xsl:apply-templates/>
    <xsl:text>}}</xsl:text>
  </xsl:template>

  <xsl:template match="WikiCreole:textlinebreak">
    <xsl:text>\\</xsl:text>
  </xsl:template>

  <xsl:template match="WikiCreole:heading">
    <xsl:variable name="level" select="@level"/>
    <xsl:call-template name="heading_markup">
      <xsl:with-param name="level" select="$level"/>
    </xsl:call-template>
    <xsl:apply-templates/>
    <xsl:call-template name="heading_markup">
      <xsl:with-param name="level" select="$level"/>
    </xsl:call-template>
    <xsl:text>

  </xsl:text>
</xsl:template>

  <xsl:template name="heading_markup">
    <xsl:param name="level" select="1"/>
    <xsl:if test="$level > 0">
      <xsl:text>=</xsl:text>
      <xsl:call-template name="heading_markup">
        <xsl:with-param name="level" select="$level - 1"/>
      </xsl:call-template>
    </xsl:if>
  </xsl:template>

  <xsl:template match="WikiCreole:orderedlist">
    <xsl:call-template name="oList">
      <xsl:with-param name="listlevel" select="string('#')"/>
    </xsl:call-template>
    <xsl:text>

  </xsl:text>
</xsl:template>
```

```

<xsl:template match="WikiCreole:unorderedlist">
  <xsl:call-template name="uList">
    <xsl:with-param name="listlevel" select="string('*')"/>
  </xsl:call-template>
  <xsl:text>
</xsl:text>
</xsl:template>

<xsl:template name="uList">
  <xsl:param name="listlevel"/>
  <xsl:for-each select="WikiCreole:listelem | WikiCreole:orderedlist | WikiCreole:unorderedlist">
    <xsl:variable name="type" select="local-name()"/>
    <xsl:if test="$type='listelem'">
      <xsl:value-of select="$listlevel"/>
      <xsl:apply-templates/>
      <xsl:text>
</xsl:text>
    </xsl:if>
    <xsl:if test="$type='orderedlist'">
      <xsl:call-template name="oList">
        <xsl:with-param name="listlevel" select="concat($listlevel, '#')"/>
      </xsl:call-template>
    </xsl:if>
    <xsl:if test="$type='unorderedlist'">
      <xsl:call-template name="uList">
        <xsl:with-param name="listlevel" select="concat($listlevel, '*')"/>
      </xsl:call-template>
    </xsl:if>
  </xsl:for-each>
</xsl:template>

<xsl:template name="oList">
  <xsl:param name="listlevel"/>
  <xsl:for-each select="WikiCreole:listelem | WikiCreole:orderedlist | WikiCreole:unorderedlist">
    <xsl:variable name="type" select="local-name()"/>
    <xsl:if test="$type='listelem'">
      <xsl:value-of select="$listlevel"/>
      <xsl:apply-templates/>
      <xsl:text>
</xsl:text>
    </xsl:if>
    <xsl:if test="$type='orderedlist'">
      <xsl:call-template name="oList">
        <xsl:with-param name="listlevel" select="concat($listlevel, '#')"/>
      </xsl:call-template>
    </xsl:if>
    <xsl:if test="$type='unorderedlist'">
      <xsl:call-template name="uList">
        <xsl:with-param name="listlevel" select="concat($listlevel, '*')"/>
      </xsl:call-template>
    </xsl:if>
  </xsl:for-each>
</xsl:template>

<xsl:template match="WikiCreole:listelem">
  <xsl:apply-templates/>
  <xsl:text>
</xsl:text>
</xsl:template>

<xsl:template match="WikiCreole:table">
  <xsl:apply-templates/>
  <xsl:text>
</xsl:text>
</xsl:template>

<xsl:template match="WikiCreole:tablerow">
  <xsl:apply-templates/>
  <xsl:text>|
</xsl:text>
</xsl:template>

```

```
<xsl:template match="WikiCreole:tablecell">
  <xsl:variable name="header" select="@header"/>
  <xsl:if test="$header = boolean(1)">
    <xsl:text>|</xsl:text>
    <xsl:apply-templates/>
  </xsl:if>
  <xsl:if test="$header = boolean(0)">
    <xsl:text>|</xsl:text>
    <xsl:apply-templates/>
  </xsl:if>
</xsl:template>

<xsl:template match="WikiCreole:nowikiblock">
  <xsl:text>{{{
</xsl:text>
  <xsl:apply-templates/>
  <xsl:text>
}}}</xsl:text>
</xsl:template>

<xsl:template match="WikiCreole:horizontalrule">
  <xsl:text>----</xsl:text>
</xsl:template>

<xsl:template match="WikiCreole:link">
  <xsl:variable name="uri" select="WikiCreole:uri"/>
  <xsl:variable name="description" select="WikiCreole:description"/>
  <xsl:text>[[</xsl:text>
  <xsl:value-of select="$uri"/>
  <xsl:if test="string-length($description) > 0">
    <xsl:text>|</xsl:text>
    <xsl:apply-templates select="$description"/>
  </xsl:if>
  <xsl:text>]]</xsl:text>
</xsl:template>

<xsl:template match="WikiCreole:image">
  <xsl:variable name="uri" select="WikiCreole:uri"/>
  <xsl:variable name="alternative" select="WikiCreole:alternative"/>
  <xsl:text>{{</xsl:text>
  <xsl:apply-templates select="$uri"/>
  <xsl:if test="string-length($alternative) > 0">
    <xsl:text>|</xsl:text>
    <xsl:apply-templates select="$alternative"/>
  </xsl:if>
  <xsl:text>}}</xsl:text>
</xsl:template>
</xsl:transform>
```