

KnotTheory`WikiForm`

This file is a subpackage of the KnotTheory` package, whose home is at <http://katlas.math.toronto.edu/>

It is concerned mostly with formatting of strings for use on the wiki Knot Atlas, and as such is extremely boring. This is the place to be if automatically uploaded data on the Knot Atlas is being formatted incorrectly, but otherwise you should stay well away.

```
BeginPackage["KnotTheory`"];
```

```
{Reversible, FullyAmphicheiral, NegativeAmphicheiral, Chiral};
```

```
WikiForm::usage =
  "ToString[expression_,WikiForm] attempts to format expression in a manner suitable
  for a MediaWiki wiki. This is a strange kludge of html and pseudo-latex,
  particularly for long polynomials. It's not perfect, but not a disaster either.";
```

```
Begin["`WikiForm`"]
```

WikiForm

It's important not to wrap integers in `<math>` tags, so they can be spliced into URLs, etc.

```
WikiForm /: ToString[a_Integer, WikiForm] := ToString[a]
```

```
WikiForm /: ToString[a_?NumberQ, WikiForm] := ToString[a]
```

```
WikiForm /: ToString["", WikiForm] := ""
```

```
WikiForm /: ToString[WikiForm[S_String], WikiForm] := S
```

```
WikiTextQ[S_String] :=
  (! (StringFreeQ[S, {"<table", "<tr", "<td", "{|", "|-", "|+", "|}",
    "{" ~ StringExpression ~ __ ~ StringExpression ~ "}"},
    "[" ~ StringExpression ~ __ ~ StringExpression ~ "]", "http://"}])) ||
  StringMatchQ[S, "<nowiki>" ~ StringExpression ~ __ ~ StringExpression ~ "</nowiki>"]
```

```
WikiForm /: ToString[s_String, WikiForm] := If[WikiTextQ[s], s,
StringReplace[
  "<nowiki>" <> s <> "</nowiki>",
  {"|" → "&#124;"}
]
]
```

```
WikiForm /: ToString[K_Knot, WikiForm] := NameString[K]
WikiForm /: ToString[L_Link, WikiForm] := NameString[L]
WikiForm /: ToString[T_TorusKnot, WikiForm] := NameString[T]
```

```
WikiForm /: ToString[Null, WikiForm] = "";
```

```
MathTags[s_String] := "<math>" <> s <> "</math>"
```

Presentations

```
listToString[{}, s_String] := ""
```

```
listToString[x_List, s_String] :=
StringJoin[Drop[Flatten[Transpose[{ToString /@ x, Table[s, {Length[x]}]}]], -1]]
```

```
WikiForm /: ToString[gc_GaussCode, WikiForm] := listToString[List @@ gc, ", "
```

```
WikiForm /: ToString[dtc_DTCode, WikiForm] :=
If[Length[dtc] == 0, "", listToString[List @@ dtc, " "]]
```

```
WikiForm /: ToString[NotAvailable, WikiForm] = "";
WikiForm /: ToString[_NotAvailable, WikiForm] = "";
```

```
WikiForm /: ToString[X[i_, j_, k_, L_], WikiForm] :=
Module[{i1 = ToString[i], j1 = ToString[j], k1 = ToString[k], l1 = ToString[L]},
If[{1, 1, 1, 1} == StringLength /@ {i1, j1, k1, l1},
ToString[StringForm["X<sub>`" <> i1 <> j1 <> k1 <> l1 <> "`", i1, j1, k1, l1]],
ToString[StringForm["X<sub>`", i1, j1, k1, l1 <> "`", i1, j1, k1, l1]]]]
```

```
WikiForm /: ToString[pd_PD, WikiForm] :=
StringJoin @@ Table[ToString[pd[[i]], WikiForm] <> " ", {i, Length[pd]}]
```

SymmetryType values

```
SymmetryType["Reversible"] = Reversible;
SymmetryType["Fully amphicheiral"] = FullyAmphicheiral;
SymmetryType["Negative amphicheiral"] = NegativeAmphicheiral;
SymmetryType["Chiral"] = Chiral;
```

```
WikiForm /: ToString[Reversible, WikiForm] = "Reversible";
WikiForm /: ToString[FullyAmphicheiral, WikiForm] = "Fully amphicheiral";
WikiForm /: ToString[NegativeAmphicheiral, WikiForm] = "Negative amphicheiral";
WikiForm /: ToString[Chiral, WikiForm] = "Chiral";
```

```
WikiForm /: ToString[_SymmetryType, WikiForm] = "";
WikiForm /: ToString[_UnknottingNumber, WikiForm] = "";
WikiForm /: ToString[_ThreeGenus, WikiForm] = "";
WikiForm /: ToString[_BridgeIndex, WikiForm] = "";
WikiForm /: ToString[_SuperBridgeIndex, WikiForm] = "";
WikiForm /: ToString[_NakanishiIndex, WikiForm] = "";
```

```
WikiForm /: ToString[NotHyperbolic, WikiForm] = "Not hyperbolic";
```

Polynomials

```
WikiForm /: ToString[poly_?LaurentPolynomialQ, WikiForm] :=
  MathTags[StringReplace[ToString[poly, TeXForm], LaurentPolynomialTeXReplacementRule]]
```

Defaults for everything else.

```
WikiTeXForm /: ToString[a_, WikiTeXForm] :=
  StringReplace[ToString[a, TeXForm], "\\text{" → "\\textrm{"}
```

```
WikiForm /: ToString[a_, WikiForm] := MathTags[ToString[a, WikiTeXForm]]
```

Laurent polynomials

```
PowerQ[_Integer] := True;
PowerQ[_Integer] = True;
PowerQ[_Symbol] = True;
PowerQ[_] = False;
```

General::spell1: Possible spelling error: new symbol name "PowerQ" is similar to existing symbol "Power". More...

```
MonomialQ[x_Times] := And @@ (PowerQ /@ List @@ x)
```

```
MonomialQ[x_] := PowerQ[x]
```

```
SplitMonomial[x_?MonomialQ] := If[MatchQ[x, _Times], List @@ x, {x}]
```

```
MonomialStringQ[x_String] :=  
  MonomialQ[ToExpression[StringReplace[x, {"{" → "(", "}" → ")"}], InputForm]]
```

```
MonomialStringQ[_] := False
```

```
PowerToString[x_?PowerQ] := x /. {k_Integer => ToString[k] <> " ",  
  z_^n_ => ToString[z] <> "^{" <> ToString[n] <> "}" , z_Symbol => ToString[z]}
```

```
InvertMonomialString[x_?MonomialStringQ] :=  
  StringJoin @@ (PowerToString /@ (#^-1 &) /@ SplitMonomial[  
    ToExpression[StringReplace[x, {"{" → "(", "}" → ")"}], InputForm]])
```

```
LaurentPolynomialQ[x_?MonomialQ] := True  
LaurentPolynomialQ[x_Plus] := And @@ (MonomialQ /@ List @@ x)
```

```
IfNotOne["1"] = "";  
IfNotOne[x_String] := x
```

```
LaurentPolynomialTeXReplacementRule =  
  "\\frac{" ~ StringExpression ~ (numerator : ShortestMatch[___]) ~  
    StringExpression ~ "}" ~ StringExpression ~ (denominator : ShortestMatch[___]) ~  
    StringExpression ~ "}" ~ StringExpression ~ (rest : ("+" | "-" | EndOfString)) =>  
  IfNotOne[numerator] ~ StringExpression ~ " " ~ StringExpression ~  
    InvertMonomialString[denominator] ~ StringExpression ~ rest;
```

General::spell1: Possible spelling error: new symbol name "numerator" is similar to existing symbol "Numerator". More...

General::spell1: Possible spelling error: new symbol name "denominator" is similar to existing symbol "Denominator". More...

General::spell: Possible spelling error: new symbol name "rest" is similar to existing symbols {res, Rest}. More...

EndPackage

```
End[]
```

```
EndPackage[]
```