

<< KnotTheory`

KnotTheory` Naming`

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

This subpackage should be loaded after the `Data subpackage.

It is concerned mostly with names of knots, and provides several functions translating names of knots, such as "K11a33" into more useful internal formats. It's been added on late in the development of the KnotTheory` package, so some code that ought to live here is buried elsewhere in the package.

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

```
TorusKnots::usage =
  "TorusKnots[n_] returns a list of all torus knots with up to n crossings.";
```

```
NameString::usage =
  "NameString[K_] returns the 'standard' string name for the knot K. These names
  are used throughout the Knot Atlas, and can be reinterpreted simply
  using the function Knot. Thus NameString[Knot[7,2]] returns \"7_2\",
  and NameString[Knot[10,NonAlternating,124]] returns \"K10n124\".";
```

```
NextKnot::usage = PreviousKnot::usage =
  "Use NextKnot and PreviousKnot to traverse lists of knots. These functions
  mostly exist to generate navigation links for the Knot Atlas.";
```

```
KnotNumber::usage =
  "For a knot K from the tables, KnotNumber[K] returns its number in the
  appropriate sequence. Thus KnotNumber[Knot[8,19]] returns
  19, while KnotNumber[Link[10,NonAlternating,5]] returns 5.";
```

```
Begin["`Naming`"]
```

KnotTheory`Naming`

```
TorusKnots[Xmax_] := Module[{res},
  res = Flatten[Table[Cases[Range[2, Min[Floor[1 + Xmax / m], m - 1]],
    n_ /; GCD[m, n] == 1 -> TorusKnot[m, n]], {m, 3, Xmax}]];
  Last /@ Sort[{Crossings[#, #] & /@ res}
]
```

```

KnotNumber[Knot[_ , k_]] := k
KnotNumber[Knot[_ , _ , k_]] := k
KnotNumber[Link[_ , _ , k_]] := k

```

NameString

```

NameString[Knot[n_Integer? (# ≤ 10 &), k_Integer]] /; (k ≤ NumberOfKnots[n]) :=
ToString[n] <> "_" <> ToString[k]

```

```

NameString[Knot[n_Integer? (# ≥ 11 &), Alternating, k_Integer]] /;
(k ≤ NumberOfKnots[n, Alternating]) := "K" <> ToString[n] <> "a" <> ToString[k]

```

```

NameString[Knot[n_Integer? (# ≥ 11 &), NonAlternating, k_Integer]] /;
(k ≤ NumberOfKnots[n, NonAlternating]) := "K" <> ToString[n] <> "n" <> ToString[k]

```

```

NameString[Link[n_Integer, Alternating, k_Integer]] /;
(k ≤ NumberOfLinks[n, Alternating]) := "L" <> ToString[n] <> "a" <> ToString[k]

```

```

NameString[Link[n_Integer, NonAlternating, k_Integer]] /;
(k ≤ NumberOfLinks[n, NonAlternating]) := "L" <> ToString[n] <> "n" <> ToString[k]

```

```

NameString[TorusKnot[m_Integer, n_Integer]] :=
"T(" <> ToString[m] <> ", " <> ToString[n] <> ")"

```

Recognise knot string names

```

Knot[S_String? (StringMatchQ[#,
StringExpression[DigitCharacter .., "_" | " ", DigitCharacter ..] &)] /;
((#[[1]] ≤ 10 ∧ #[[2]] ≤ NumberOfKnots[#[[1]])] & [ToExpression /@
StringSplit[S, "_" | " "]] :=
Knot @@ (ToExpression /@ StringSplit[S, "_" | " "])

```

```

Knot[S_String? (StringMatchQ[#,
StringExpression["K", DigitCharacter .., "a", DigitCharacter ..] &)] /;
((#[[1]] ≥ 11 ∧ #[[2]] ≤ NumberOfKnots[#[[1]], Alternating]) & [
ToExpression /@ StringSplit[S, {"K", "a"}]]) :=
Knot[#[[1]], Alternating, #[[2]]] & [(ToExpression /@ StringSplit[S, {"K", "a"}])]

```

```

Knot[S_String? (StringMatchQ[#,
  StringExpression["K", DigitCharacter .., "n", DigitCharacter ..] &)] /;
((#[1] ≥ 11 ∧ #[2] ≤ NumberOfKnots[#[1], NonAlternating]) &[
  ToExpression /@ StringSplit[S, {"K", "n"}]]) :=
Knot[#[1], NonAlternating, #[2]] &[(ToExpression /@ StringSplit[S, {"K", "n"}])]

```

```

Knot[S_String? (StringMatchQ[#,
  StringExpression["L", DigitCharacter .., "a", DigitCharacter ..] &)] /;
((1 ≤ #[2] ≤ NumberOfLinks[#[1], Alternating]) &[
  ToExpression /@ StringSplit[S, {"L", "a"}]]) :=
Link[#[1], Alternating, #[2]] &[(ToExpression /@ StringSplit[S, {"L", "a"}])]

```

```

Knot[S_String? (StringMatchQ[#,
  StringExpression["L", DigitCharacter .., "n", DigitCharacter ..] &)] /;
((1 ≤ #[2] ≤ NumberOfLinks[#[1], NonAlternating]) &[
  ToExpression /@ StringSplit[S, {"L", "n"}]]) :=
Link[#[1], NonAlternating, #[2]] &[(ToExpression /@ StringSplit[S, {"L", "n"}])]

```

```

Knot[S_String? (StringMatchQ[#,
  StringExpression["T(", DigitCharacter .., ",", DigitCharacter .., ")"] &)] :=
TorusKnot[#[1], #[2]] &[(ToExpression /@ StringSplit[S, {"T(", ",", ")"}])]

```

```

Link[S_String] := Knot[S]

```

Recognise Livingston's naming system.

```

Knot[S_String? (StringMatchQ[#,
  StringExpression[DigitCharacter .., "a_", DigitCharacter ..] &)] /;
((#[1] ≥ 11 ∧ #[2] ≤ NumberOfKnots[#[1], Alternating]) &[
  ToExpression /@ StringSplit[S, {"a_"}]]) :=
Knot[#[1], Alternating, #[2]] &[(ToExpression /@ StringSplit[S, {"a_"}])]

```

```

Knot[S_String? (StringMatchQ[#,
  StringExpression[DigitCharacter .., "n_", DigitCharacter ..] &)] /;
((#[1] ≥ 11 ∧ #[2] ≤ NumberOfKnots[#[1], NonAlternating]) &[
  ToExpression /@ StringSplit[S, {"n_"}]]) :=
Knot[#[1], NonAlternating, #[2]] &[(ToExpression /@ StringSplit[S, {"n_"}])]

```

NextKnot and PreviousKnot

```
NextKnot[Knot[0, 1]] = Knot[3, 1];
NextKnot[Knot[n_Integer? (# ≤ 10 &), k_Integer]] /; (k < NumberOfKnots[n]) :=
  Knot[n, k + 1]
NextKnot[Knot[n_Integer? (# ≤ 9 &), k_Integer]] /; (k == NumberOfKnots[n]) :=
  Knot[n + 1, 1]
NextKnot[Knot[10, k_Integer]] /; (k == NumberOfKnots[10]) := Knot[11, Alternating, 1]
```

```
NextKnot[Knot[n_Integer? (# ≥ 11 &), t_, k_Integer]] /;
  (k < NumberOfKnots[n, t]) := Knot[n, t, k + 1]
NextKnot[Knot[n_Integer? (# ≥ 11 &), Alternating, k_Integer]] /;
  (k == NumberOfKnots[n, Alternating]) := Knot[n, NonAlternating, 1]
NextKnot[Knot[n_Integer? (# ≥ 11 &), NonAlternating, k_Integer]] /;
  (k == NumberOfKnots[n, NonAlternating]) := Knot[n + 1, Alternating, 1]
```

```
PreviousKnot[Knot[0, 1]] = Knot[0, 1];
PreviousKnot[Knot[3, 1]] = Knot[0, 1];
PreviousKnot[Knot[n_Integer? (# ≤ 10 &), 1]] := Knot[n - 1, NumberOfKnots[n - 1]]
PreviousKnot[Knot[n_Integer? (# ≤ 10 &), k_Integer]] := Knot[n, k - 1]
```

```
PreviousKnot[Knot[11, Alternating, 1]] = Knot[10, NumberOfKnots[10]];
PreviousKnot[Knot[n_Integer? (# ≥ 12 &), Alternating, 1]] :=
  Knot[n - 1, NonAlternating, NumberOfKnots[n - 1, NonAlternating]]
PreviousKnot[Knot[n_Integer? (# ≥ 11 &), NonAlternating, 1]] :=
  Knot[n, Alternating, NumberOfKnots[n, Alternating]]
PreviousKnot[Knot[n_Integer? (# ≥ 11 &), t_, k_Integer]] := Knot[n, t, k - 1]
```

```
NextKnot[Last[AllLinks[]]] = Last[AllLinks[]];
PreviousKnot[Link[2, Alternating, 1]] := Link[2, Alternating, 1];
NextKnot[L_Link] := With[{all = AllLinks[]}, all[[Position[all, L][[1, 1]] + 1]]]
PreviousKnot[L_Link] := With[{all = AllLinks[]}, all[[Position[all, L][[1, 1]] - 1]]]
```

```
PreviousKnot[TorusKnot[3, 2]] = TorusKnot[3, 2];
```

```
TorusKnotPosition[TorusKnot[m_, n_]] := Module[{l = 36},
  While[! MemberQ[TorusKnots[l], TorusKnot[m, n]], l += 36];
  Position[TorusKnots[l], TorusKnot[m, n]][[1, 1]]
]
```

```
PreviousKnot[T_TorusKnot] := TorusKnots[Crossings[T]][[TorusKnotPosition[T] - 1]]
```

```
NextKnot[T_TorusKnot] := Module[{p = TorusKnotPosition[T] + 1, n = 36},  
  While[Length[TorusKnots[n]] < p, n += 36];  
  TorusKnots[n][[p]]  
]
```

EndPackage

```
End[]
```

KnotTheory`Naming`

```
EndPackage[]
```

```
Equal[  
  Knot /@ {"3_1", "3_1", "K11a1", "K11n1", "L11a1", "L11n1", "11a_1", "11n_1"},  
  {Knot[3, 1], Knot[3, 1], Knot[11, Alternating, 1],  
   Knot[11, NonAlternating, 1], Link[11, Alternating, 1], Link[11, NonAlternating, 1],  
   Knot[11, Alternating, 1], Knot[11, NonAlternating, 1]}  
]  
True
```