NodeCalculator cheat sheet

Mischa Kolbe
www.mischakolbe.com
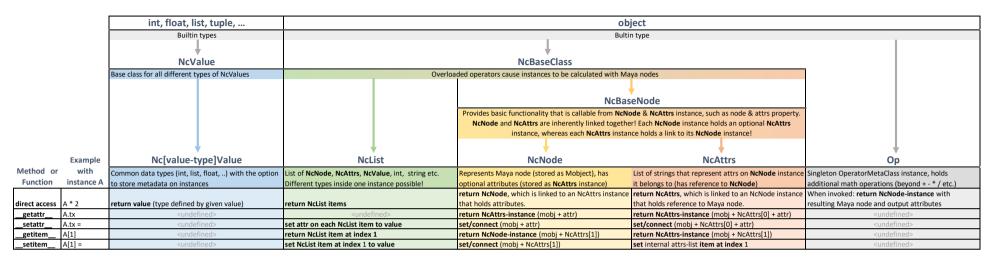
This chart shows class inheritance (underneath each class-type is a short description what the class is and what makes this class necessary). At the bottom are common forms of accessing instances; their behavior or return types.

		int, float, list, tuple, …		object			
		Builtin types		Bultin type			
		↓		↓			
		NcValue		**NcBaseClass**			
		Base class for all different types of NcValues	Overloaded operators cause instances to be calculated with Maya nodes				
				NcBaseNode			
				Provides basic functionality that is callable from **NcNode** & **NcAttrs** instance, such as node & attrs property. **NcNode** and **NcAttrs** are inherently linked together! Each **NcNode** instance holds an optional **NcAttrs** instance, whereas each **NcAttrs** instance holds a link to its **NcNode** instance!			
Method or Function	**Example with instance A**	**Nc[value-type]Value**	**NcList**	**NcNode**	**NcAttrs**	**Op**
		Common data types (int, list, float, ..) with the option to store metadata on instances	List of **NcNode**, **NcAttrs**, **NcValue**, int, string etc. Different types inside one instance possible!	Represents Maya node (stored as Mobject), has optional attributes (stored as **NcAttrs** instance)	List of strings that represent attrs on **NcNode** instance it belongs to (has reference to **NcNode**)	Singleton OperatorMetaClass instance, holds additional math operations (beyond + - * / etc.)
direct access	A * 2	**return value** (type defined by given value)	**return NcList items**	**return NcNode**, which is linked to an NcAttrs instance that holds attributes.	**return NcAttrs**, which is linked to an NcNode instance that holds reference to Maya node.	When invoked: **return NcNode-instance** with resulting Maya node and output attributes
__getattr__	A.tx	<undefined>	<undefined>	**return NcAttrs-instance** (mobj + attr)	**return NcAttrs-instance** (mobj + NcAttrs[0] + attr)	<undefined>
__setattr__	A.tx =	<undefined>	**set** attr on each NcList item to value	**set/connect** (mobj + attr)	**set/connect** (mobj + NcAttrs[0] + attr)	<undefined>
__getitem__	A[1]	<undefined>	**return NcList item at index 1**	**return NcNode-instance** (mobj + NcAttrs[1])	**return NcAttrs-instance** (mobj + NcAttrs[1])	<undefined>
__setitem__	A[1] =	<undefined>	**set NcList item at index 1 to value**	**set/connect** (mobj + NcAttrs[1])	**set** internal attrs-list **item at index** 1	<undefined>

Keywords

Keywords are special properties on your NodeCalculator Nodes. Use them like any other attribute on an NcList/NcNode/NcAttrs-instance:
nc_node = noca.Node("pCube1")
nc_node.node

node	<unavailable>	<unavailable>	**string** : Name of Maya node	**string** : Name of Maya node	<unavailable>
nodes	<unavailable>	**list of strings** : Maya nodes inside this NcList	**list of string** : Name of Maya node	**list of string** : Name of Maya node	<unavailable>
attrs	<unavailable>	<unavailable>	**NcAttrs** : Instance connected to this NcNode	**NcAttrs** : self	<unavailable>
attrs_list	<unavailable>	<unavailable>	**list of strings** : Stored attrs	**list of strings** : Stored attrs	<unavailable>
plugs	<unavailable>	<unavailable>	**list of strings** : Stored plugs	**list of strings** : Stored plugs	<unavailable>

Glossary

Word	Description	Examples
node	Name of Maya node (dagPath if not unique).	pCube1, |groupA|pSphere1, namespace1:pTorus1
attr/attribute	Attribute on a Maya node in the scene.	tx, translateX, v, visibility, input3D[0].input3Dx
plug	Combination of node and attribute; node.attr	pCube1.tx, |groupA|pSphere1.visibility

array plug	A plug that allows any number of connections.	"input3D" is the array plug of the plugs "input3D[i]".
array plug element	A specific plug of an array plug.	"input3D[7]" is an array plug element of "input3D".
parent plug	A plug that can be split into child plugs.	"translate" is the parent plug of ["tx", "ty", "ty"].
child plug	A plug that makes up part of a parent plug.	"translateX" is a child plug of "translate".