1. Introduction

Thank you for downloading *KnobScripter v2.4.* I hope it'll help you script nicer in Nuke.

You can also find a **video** of the tool here: ***https://vimeo.com/adrianpueyo/knobscripter2***
Also, one for the **v2.1 updates:** ***https://vimeo.com/adrianpueyo/knobscripter2-v1***

> **KnobScripter 2.4** *is a full Python script editor for Nuke that can script on .py files and python knobs, with all the functionality from the default script editor plus syntax helpers.*

KnobScripter started in 2016 as a little python programming widget for nuke, and an exercise for me to learn PySide. Its goal was to provide proper **syntax highlighting**, line **numbers** and auto-indenting, and the ability to program on node **knobs** directly.

Two years later I started working on expanding its functionality, and I found more and more cool ideas to implement. After months of improving and testing it, it turned into a simple but fully functional python script editor which I released as *KnobScripter 2*. The following updates introduced new color styles, font selection, full auto-completer and more, as well as compatibility for the latest Nuke versions and performance enhancements.
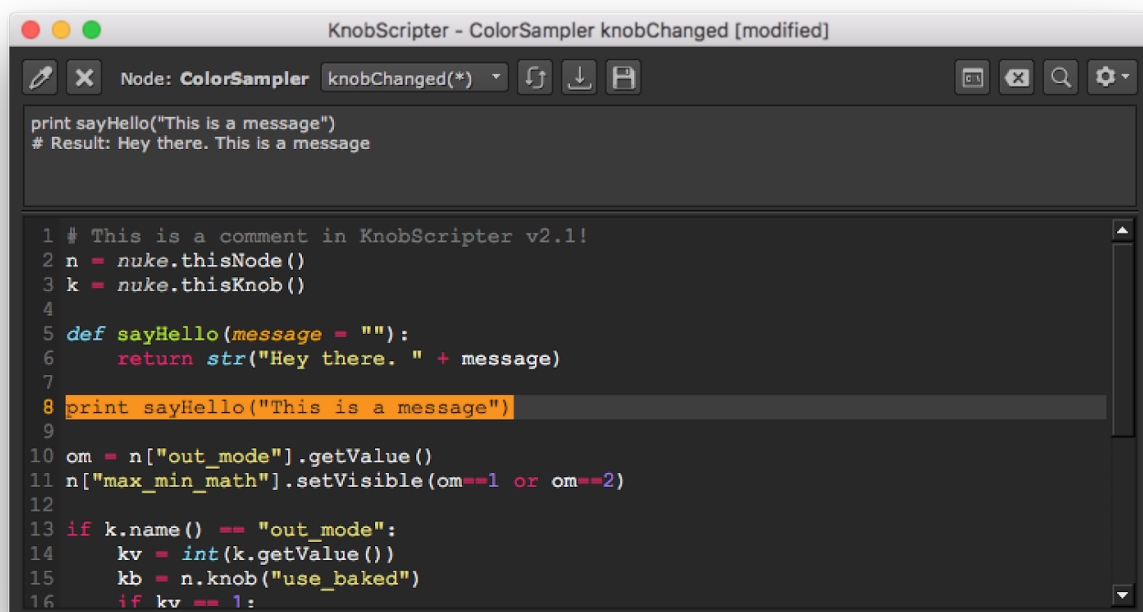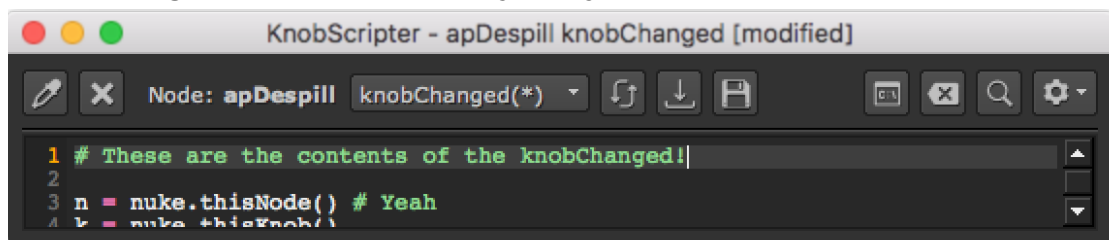
2. Features

- **Full scripting mode for .py files.**
  You can create, browse, modify or toggle between python files and folders.
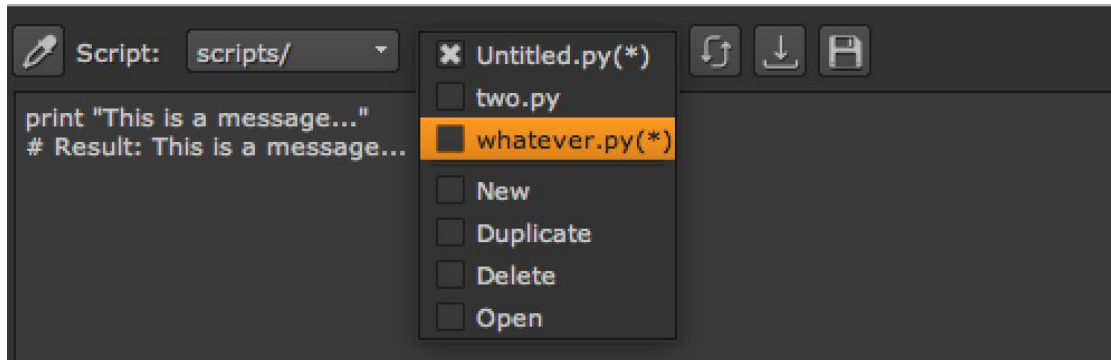
- **Node editing mode, to script directly on python buttons or callback knobs.**
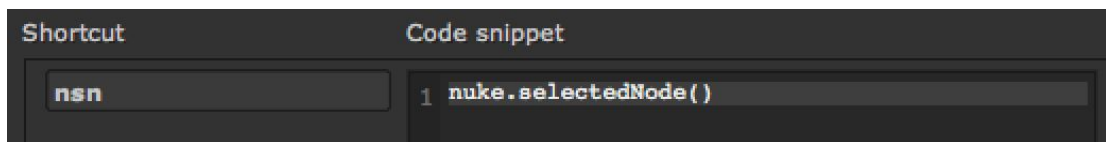

- **Python output console.**
  Same as the one from Nuke's default script editor, where you can execute any code.

- **Find-Replace.**
  A proper find-replace widget as you'd expect in a python editor :D

- **Snippets!**
  They are short codes you can assign to longer pieces of codes, so that by writing the short code and pressing tab you'll get the long code.

- **Python syntax highlighting, line numbers, auto-intending, auto-completer (v2.1)**

- **Syntax helpers, multi-line commenting, moving/duplicating lines, and more!**
  Helps you open and close parenthesis, comments, etc. Multilin

- **Free!**

3. Basics

A. Panel Types

In Nuke, you can open the *KnobScripter* both as a floating window or as a dockable pane.
- To open the KnobScripter as a **floating window**, simply press Alt+Z on the Node Graph.
- In order to bring the **dockable pane** you need to do the following:
  Right click on the pane selection bar, and go to:
  ```
  Windows -> Custom -> KnobScripter.
  ```
  Then, a KnobScripter pane will get created. Now you can even save the workspace, so the KnobScripter will be created by default when you open nuke.
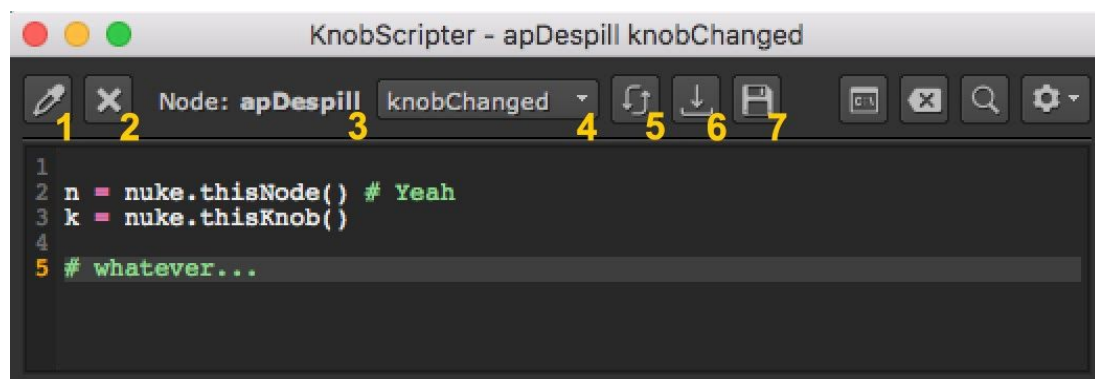
B. KnobScripter Modes

There are two main modes on the *KnobScripter*:

1. **Node Mode**
   Lets you script on python buttons, python custom or callback knobs, on any node.

   To enter the node mode, click on the **node picker** [icon] or simply open a floating *KnobScripter* (alt+Z) with a node selected. Then, you'll see the following:

1	Node picker. To select a different node. If no node selected, lets you type the name of any node in your script. Including `root` and `preferences`.
2	Close the node mode.
3	Currently selected node, in which we are scripting.
4	Dropdown of available knobs on the node (python and callback knobs). v2.1: You can display knob labels and names, or just names.
5	Refresh the dropdown...
6	Reload the saved state of the knob. Will revert any changes on the editor.
7	Save the editor contents into the knob. Till you save, nothing changes.

2. **Script Mode**

Lets you script on .py files, that you can create, modify, duplicate, and even create different folders for them. Additionally, it lets you point to any folder in your computer.
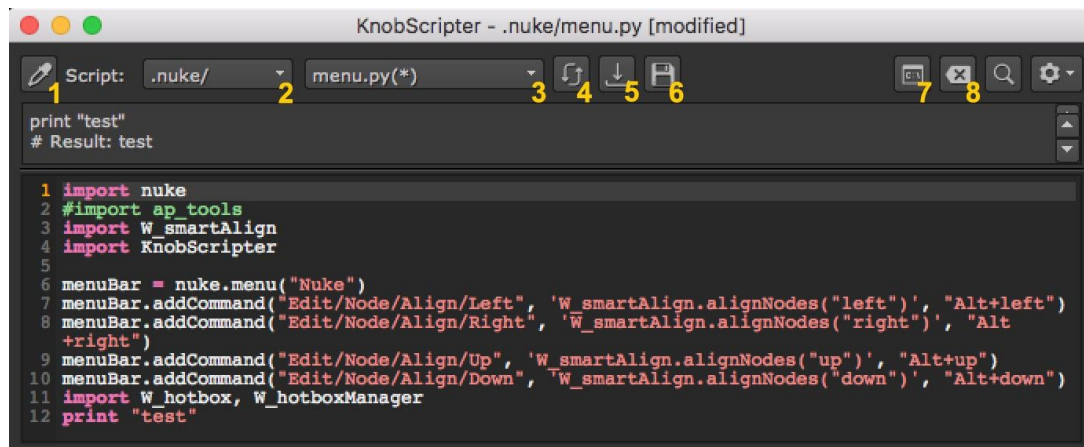

1	Node picker. To select a node, and enter the node mode.
2	Dropdown with available folders as well as custom added ones (like .nuke) As well as: New, Open (browse), and Add custom path.
3	Dropdown with all .py scripts in the selected folder. (menu.py selected) As well as: New, Duplicate, Delete, and Open with external app.
4	Refresh the dropdowns.
5	Reload the save state of the script. Will revert any changes on the editor.
6	Save the script into the .py file. A .py.autosave will be created otherwise.
7	Execute the selected code (or all code if no selection) on the output panel.
8	Clear the output panel (Top part, with the lighter grey background).

C. Common Features

The buttons on the right are common to both modes. There's two we haven't mentioned yet:

 Show or hide the **Find-Replace** widget. This is also activated through Ctrl+F.
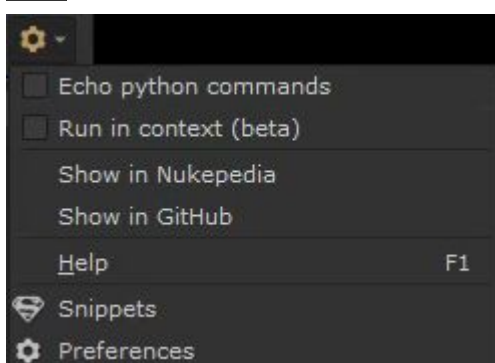
 The last button corresponds to the Settings.
Echo python commands toggles Nuke's echo.
Run in context enables the context mode, explained in section 5.
Show... and Help link to sites on your browser.
Snippets opens the snippet editor (see next page).

`Preferences` opens the preferences panel, with the scripting options, looks and defaults.

4. Snippets

Snippets are a convenient way to make your programming faster, especially for the lines of code that you tend to write over and over again. A **snippet** is a short text that you can define, and associate a longer code to it, so that you can add the longer code to your script by simply typing the short text and pressing `Tab`.

You can edit the snippets on the **Snippet Editor**, under the **Settings** dropdown button.
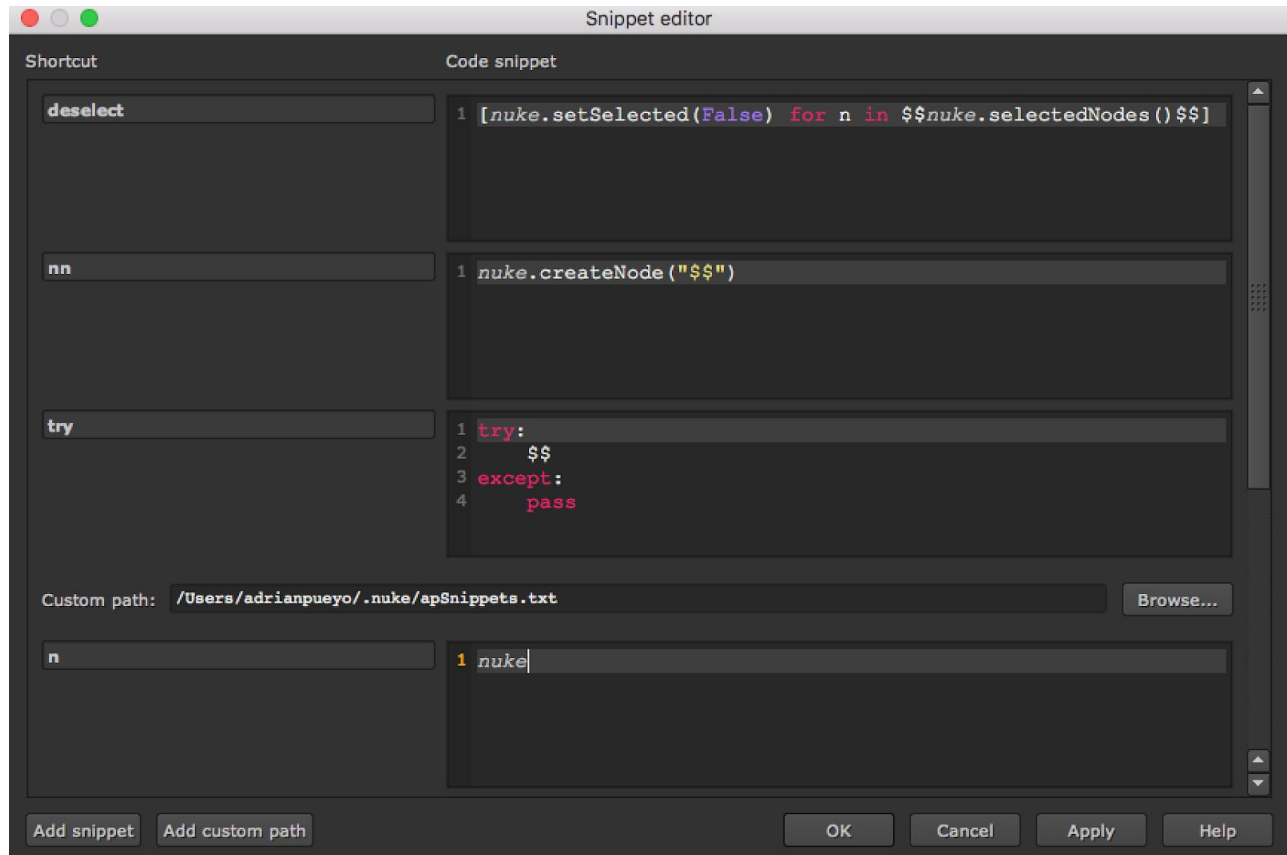
A. Create, Delete and Modify Snippets
To make a new snippet, click on `Add Snippet`, and an empty row will be created.
To remove a snippet, leave its fields empty (and then click `OK` or `Apply`).
Snippets are saved on `.nuke/KnobScripter_Snippets.txt.` You can edit that file too!

B. Add custom path
`Add custom path` lets you point to an external snippets .txt file (recursively), so you can share some snippets with colleagues or all read from a central snippets file. Your local Snippets will always be given preference in case two snippets share the same name.

C. Cursor Placeholder
Type $$ anywhere on the Snippet code to create a cursor placeholder.
Then, the cursor will appear there after you press Tab. Check the `try` snippet above and you'll understand.
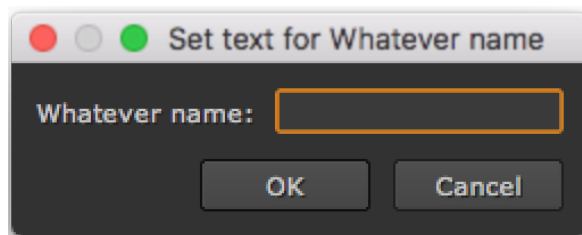
6

D. Selected text Placeholder

From v2.1 you can also add a selection placeholder, instead of just the cursor. It works the following way: If you create a Snippet with the code **`hello $$world$$,`** it will create the text **`hello world`**, with the word **`world`** selected straightaway. This will let you change it quickly afterwards or leave it as it is.

E. Variable Placeholders

Another cool thing you can do through the Snippets, is define variables. I explain it in detail in the tutorial video, but in short:

```
for n in $Whatever name$:
    print n
```

^ If you link this code to the shortcode "for", for example, when you type `for` and press `Tab`, it will prompt you with this panel:

Then, after filling in the name, the snippet code will be added to your script editor, with the word you just typed in the `$Whatever name$` placeholder location.

This is especially useful when you want that variable to **appear several times** on the same snippet.

You can put **as many variable placeholders as you want**, or combine it with the cursor placeholder `$$` or selected text placeholder `$$text$$`.

5. Preferences

The **Preferences panel** lets you modify the default behavior of the KnobScripter:

`Font`: choose any font! (new in v2.1)
`Font size`: size of the editor font (pixels)

`Width/height`: default dimensions of the floating KnobScripter.
`Grab current dimensions`: sets width/height to the current window size.

`Tab spaces`: number of spaces that will be created when indenting with tab.

`Run in context`: explained below.

`Show labels`: on node mode, display knob label and name on the dropdown or just name (new in v2.1)

`Color scheme`: explained below

Color scheme (new in v2.1)
Lets you choose between two syntax highlighters for your python code. Nuke is similar to Nuke's script editor default one. Subl resembles the look of Sublime's default colors.

Run in context (new in v2.4)
If you set `Run in context` on the Preferences, that will let you execute code you're writing in a knob and have it behave as if it were executed from within its node.

For example, if you're coding inside the knobChanged of a Blur node, and you have the code `print nuke.thisNode().name()`, before it would print `Root` (as if you ran the code on Nuke's Script editor), but with `Run in Context` enabled it will print something like `Blur1`.

6. Other features

- **Duplicating lines**
  You can duplicate a line of code by pressing:
  `Ctrl (or cmd) + Shift + D`
  If you have any code selected, that code will get duplicated.

- **Moving lines up and down**
  You can move a line of code up or down by pressing:
  `Ctrl (or cmd) + Shift + Up or Down arrow`
  If you select a code spanning across several lines, they will all be moved in block.

- **Auto indenting**
  When creating a new line, indenting will be added automatically.

- **Multi-line commenting and uncommenting**
  If you select several lines and press `#` (the hash key), a hash will be placed before each of the lines. If all of the lines already started with a hash, it will be removed.

- **Tab menu showing the available modules**
  If you press `Tab` after starting to type a word, if that word doesn't have a snippet associated to it, it will show Nuke's dropdown completer with the available modules and/or functions for the Script Editor.

- **Auto-completer and current script modules (new in v2.1)**
  Pressing `Tab` after starting to type a word will auto-complete it if a variable, function or class starting with those letters has been declared in the current script. If there's more than one match, the tab menu dropdown will appear and also include these words.

- **Syntax helpers**
  - Opening parenthesis will close it too. *Same with brackets, braces and quotes.*
  - Opening parenthesis with text selected will open and close them around the selection. *Same with brackets, braces and quotes.*
  - I'm probably forgetting some. Will keep updating this documentation, anyway. (Maybe)

7. Installation

A. Fresh install
1. Copy the *KnobScripter* folder and paste it inside your *.nuke* directory.
2. Open the file *menu.py* inside your .nuke folder with a text editor, or create it if it doesn't exist.
3. Add the following line: `import KnobScripter`
4. Restart nuke.

B. Updating KnobScripter
1. Replace the *KnobScripter* folder inside your *.nuke* directory.
2. Restart nuke.

8. Updates Log

KnobScripter v2.4 *- 7 December 2020*
New features:
1. Run in Context mode.
2. "Grab current dimensions" button.

Bug fixes and enhancements:
1. Improved modal window behavior (thanks Jed Smith!)
2. Ctrl+Backspace now clears output log.
3. Improved how Nuke's Script Editor widgets are found (thanks Jed Smith!)
4. Modified (*) property now shows properly on the knob dropdown.
5. Picker ("change to Node") crashing fixed.
6. Unicode encoding works with both .py scripts and nodes now.
7. Compatible with Nuke 12.2
8. Add snippet button scrolls to top.

KnobScripter v2.3 *- 24 December 2019*
Bug fixes:
1. Fixed bracket and brace closing behaviour.
2. Added QStringListModel compatibility on nuke 12.
3. Symlink error fixed, hidden on Windows.

KnobScripter v2.2 *- 12 August 2019*
Bug fixes:
1. Variable placeholders error fixed (bug introduced on v2.1).
2. PySide import switch including Qt.

KnobScripter v2.1 *- 10 August 2019*
New features:
1. New Sublime color style. You can now choose between sublime or Nuke style in the Preferences.
2. Font selector in the Preferences, where you can set any available font.
3. Option to display the knob labels too (and not only names) on the knob dropdown.
4. Now accepting special characters (like accents or symbols) through utf-8 encoding.
5. New Snippets functionality: A snippet containing "hello $$world$$" will write "hello world" and the word "world" will be selected straightaway.
6. Auto-completer (when pressing tab) now also includes the functions, variables, classes etc live from the current script you're writing.

Bug fixes and enhancements:
1. Python syntax highlighting improved.
2. Snippet editor now includes syntax highlighting.
3. Parenthesis/brackets auto-closing behavior improved when written inside each other.

4. Fixed error that prevented opening a new KnobScripter when a Blinkscript node properties were open too.
5. Tested on Windows OS (Nuke 11.3) and a few minor bugs fixed.
6. Improved auto-scroll to cursor behavior when duplicating or moving lines.
7. Unindent keeps scroll value.

Guide last updated: 7 December 2020

License

www.adrianpueyo.com