

## 1. Introduction

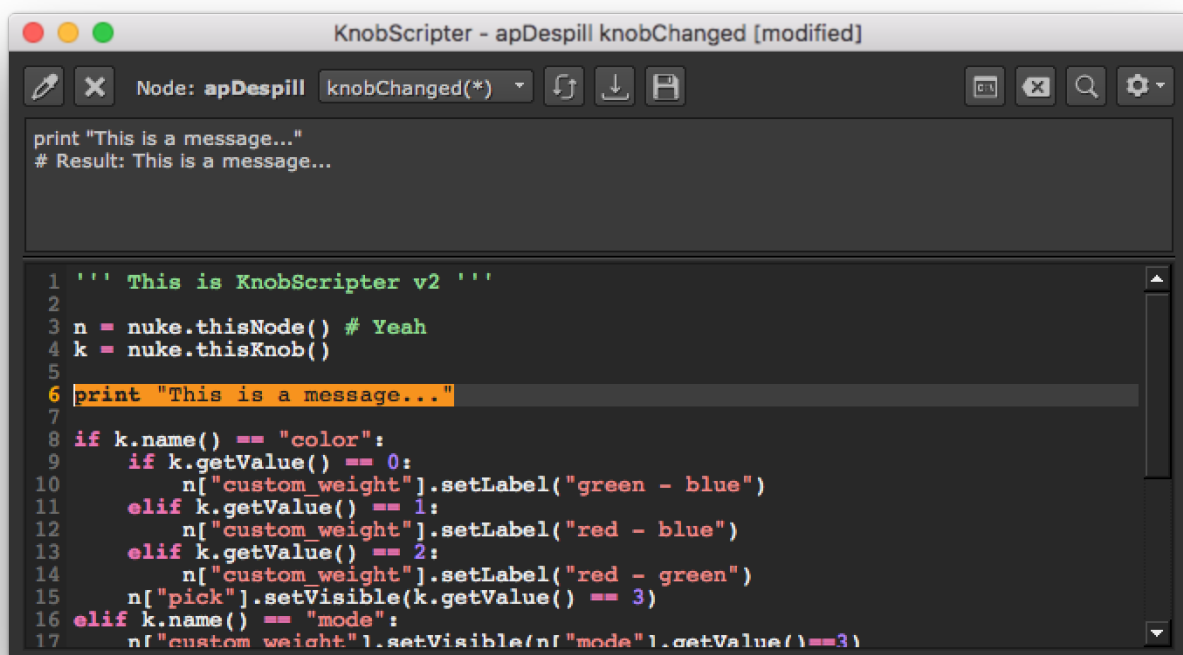
Thank you for downloading **KnobScripter 2**. I hope it'll help you script nicer in Nuke.

You can also find a **video** of the tool here: <https://vimeo.com/adrianpueyo/knobscripter2>

***KnobScripter 2** 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.

I recently started to work on expanding its functionality, and I found more and more cool ideas to implement. After quite some months of improving and testing it, **KnobScripter 2** is **finally in a good enough state!**



The screenshot shows the KnobScripter application window titled "KnobScripter - apDespill knobChanged [modified]". The interface includes a toolbar with icons for editing, running, and saving. The main text area displays a Python script for a Nuke knob named "apDespill" with the knob type "knobChanged(\*)". The script includes a docstring, variable declarations, and logic to change the label and visibility of a custom weight knob based on the selected mode.

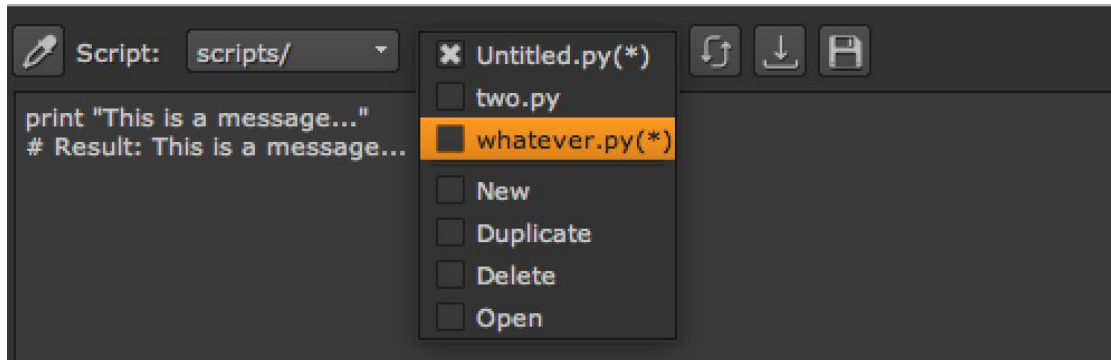
```
print "This is a message..."
# Result: This is a message...

1 ''' This is KnobScripter v2 '''
2
3 n = nuke.thisNode() # Yeah
4 k = nuke.thisKnob()
5
6 print "This is a message..."
7
8 if k.name() == "color":
9     if k.getValue() == 0:
10         n["custom_weight"].setLabel("green - blue")
11     elif k.getValue() == 1:
12         n["custom_weight"].setLabel("red - blue")
13     elif k.getValue() == 2:
14         n["custom_weight"].setLabel("red - green")
15     n["pick"].setVisible(k.getValue() == 3)
16 elif k.name() == "mode":
17     n["custom_weight"].setVisible(n["mode"].getValue() == 3)
```

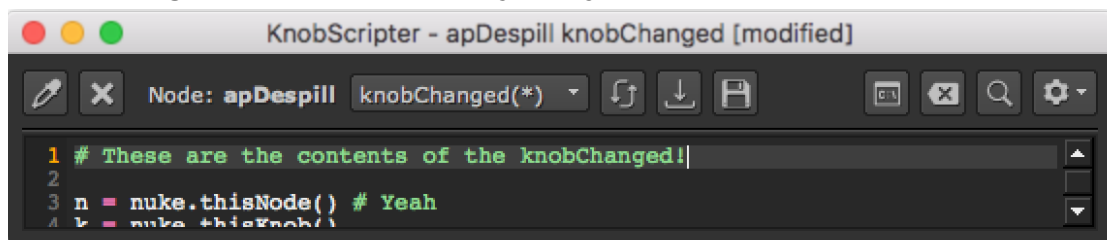
## 2. Features

- **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.**

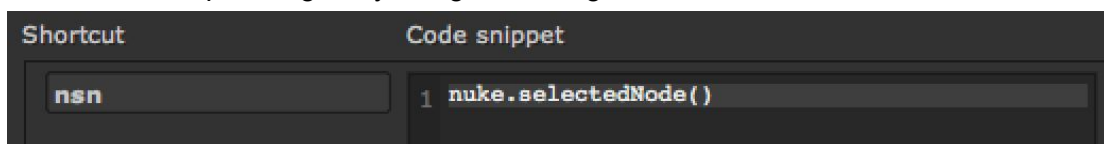
Same as the one from Nuke's default script editor, where you can execute any code.

- **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.**

- **Syntax helpers, multi-line commenting, moving/duplicating lines, and more!**

Helps you open and close parenthesis, comments, etc. Multilin

- **Free! (of course)**

### 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.

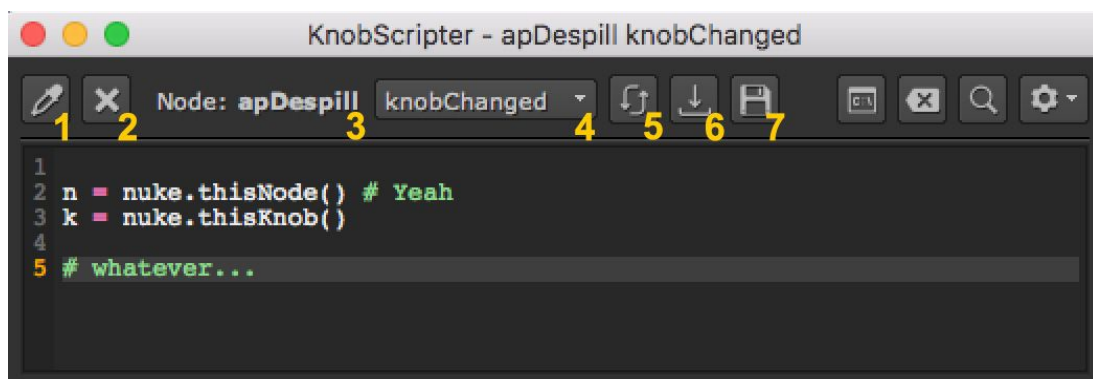
#### B. KnobScripter Modes

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.

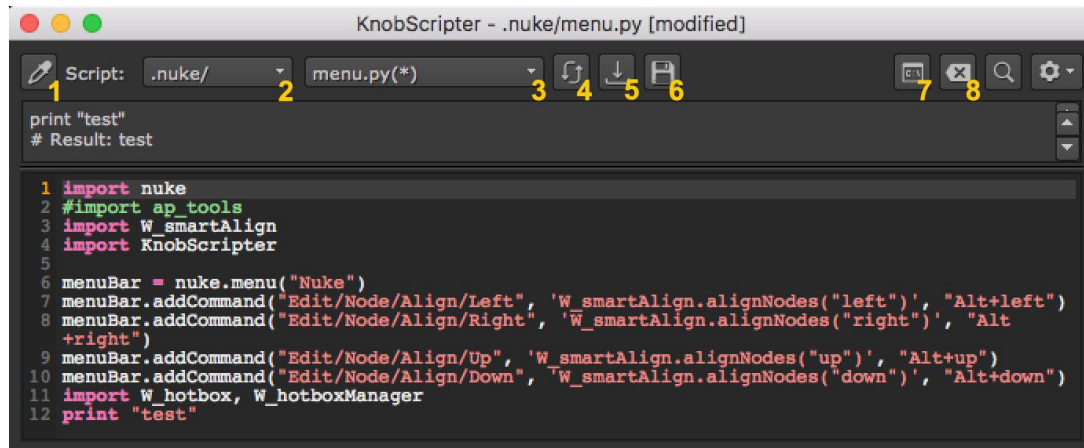
To enter the node mode, click on the **node picker**  or simply open a floating *KnobScripter* (alt+Z) with a node selected. Then, you'll see the following:



1	Node picker. To select a different node. If no node selected, lets you type the name of any node in your script. Including <code>root</code> and <code>preferences</code> .
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).
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.

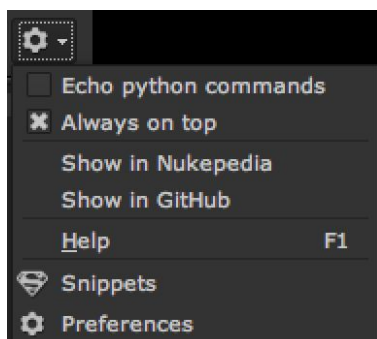


1	Node picker. To select a node, and enter the node mode.
2	Dropdown with available folders as well as custom added ones (like .nuke)
3	Dropdown with all .py scripts in the selected folder. (menu.py selected)
4	Refresh the dropdowns.
5	Reload the save state of the script. Will revert any changes on the 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 right buttons 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 the nuke echo.

Always on top changes the behavior of the floating panel.

Show... and Help are links to pages on your web browser.

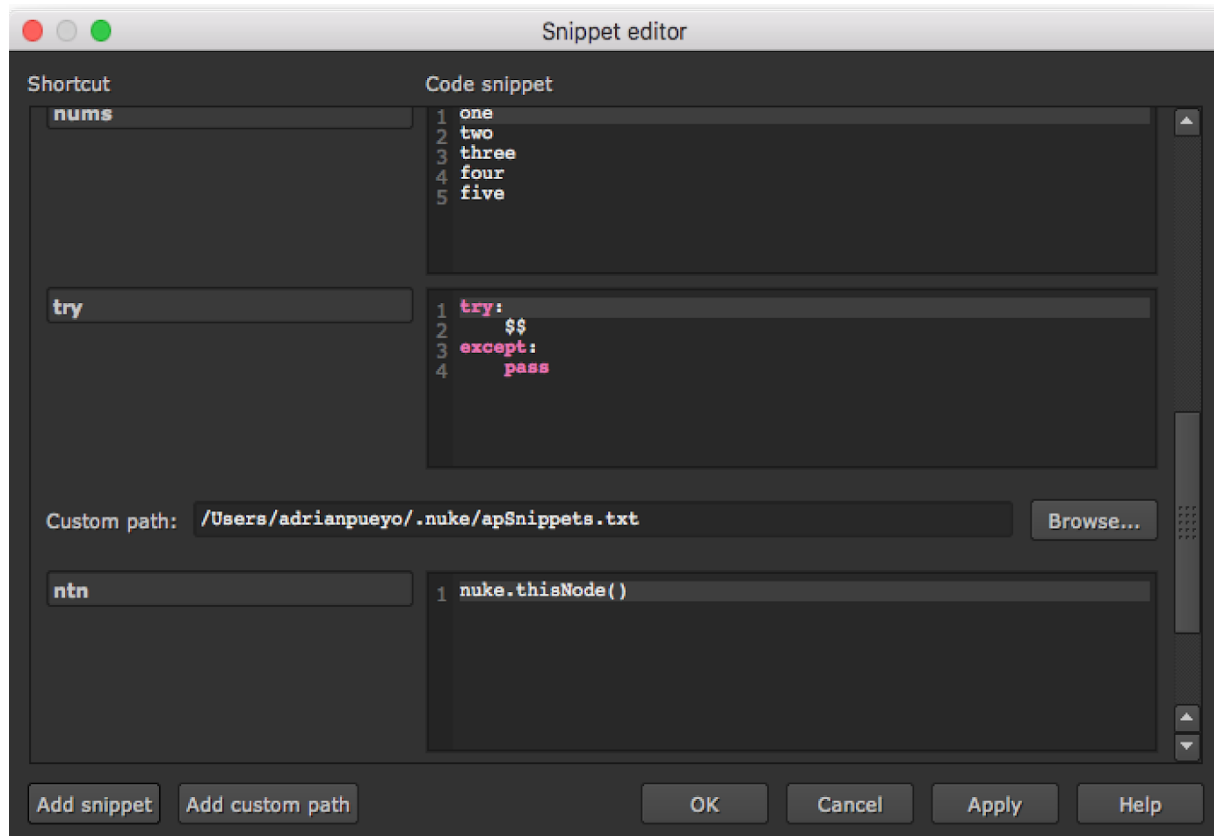
Snippets opens the snippet editor (check the next page).

Preferences opens some extra options.

## 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 whatever.

### C. Cursor Placeholder

Type `$$` anywhere on the Snippet code to create a cursor placeholder.

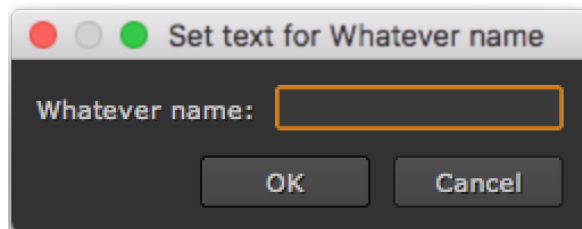
Then, the cursor will appear there after you press `Tab`. Check the `try` snippet above and you'll understand.

## D. 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:



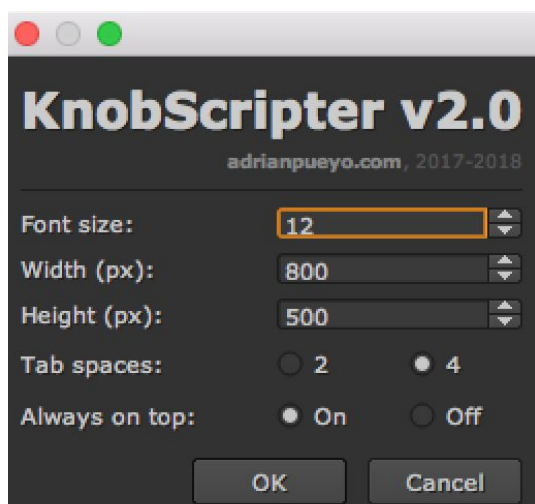
Then, after filling in the name, the snippet code will be added to your script editor, with the word you just typed in the `$Whatever name$` placeholder location.

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 `$$`.

## 5. Preferences

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



Font size: size of the editor font (pixels)

Width/height: default dimensions of the floating KnobScripter.

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

Always on top: behavior of the floating KnobScripter, in relation to the other windows.

## 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.

- **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.*
- And I'm probably forgetting. Will update this documentation at some point, 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.

## License



Copyright © 2016-2019, Adrian Pueyo  
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

**[www.adrianpueyo.com](http://www.adrianpueyo.com)**