

# TCL Commands

xxxxxxxxxx

1

GETTING A KNOB'S VALUE OF A SPECIFIC NODE:

2

3

#First frame of current read/write:

4

[value Read1.first]

5

6

#Getting a knob's value of current node:

7

[value this.first\_frame]

8

9

#Return label value of the input node:

10

[value this.size]

11

12

#Name of the input node:

13

[value this.input0.label]

14

15

#Name of the node before the group (Outside):

16

[value this.input.name]

17

18

#Return 1 if the node is on error otherwise 0:

19

[value this.parent.input.name]

20

21

#Get the bounding Box from the input of the node:

22

[value error]

23

24

25

#Here some expression for the Format

26

format.x

27

format.y

28

width

29

height

30

bbox.x

31

bbox.y

32

bbox.w

33

bbox.h

34

35

#Get the format from the input of the node:

36

#left boundary

37

[value input.bbox.x]

38

#right boundary

39

[value input.bbox.r]

40

41

#Get the format from the input of the node:

42

#width

43

[value input.format.r]

44

#height

45

[value input.format.t]

46

47

#Get the x position of the point #3 of the Bezier1 of the Roto1 node:

48

[value Roto1.curves.Bezier1.curve\_points.3.main.x]

49

50

#Return sample pixel value of the node Add1 reading in the red at position of knob Center:

51

[sample Add1 Red Center.x Center.y]

52

53

#Get the value of the channel of a node, at a specific pixelcoordinates (e.g.: 10,10):

54

[sample [node input] red 10 10]

55

56

#-----

57

#SET VALUES

58

59

#Setting a knob's value of a specific node:

60

```
[knob Read1.first 10]
```

61

62

```
#Setting a variable, without returning that (useful in a textnode):
```

63

```
[set seq [value Read1.file]; return]
```

64

65

```
#-----
```

66

```
#STRING
```

67

68

```
#Replace string in current node file knob with regex (string "proj" to "projects" in all occurrences):
```

69

```
[regsub -all "proj" [value [node this].file] "projects"]
```

70

71

```
#String map (replace multiple stringpairs) (this returns: xxxfffxyy):
```

72

```
[string map {"aa" "xx" "bb" "yy"} "aaffffaabb" ]
```

73

74

```
#Compare strings:
```

75

```
[string equal [value Text1.message] "bla"]
```

76

77

```
#Regex matching:
```

78

```
[regex -inline "_v\[0-9\]{3}" [value Read2.file]]
```

79

80

```
#Evaluating string
```

81

```
[python os.getenv('rotate') == 'xavierb']
```

82

83

```
#-----
```

84

```
#IF CONDITION
```

85

86

```
[if {condition} {expr1} else {expr2}]
```

87

88

```
#Example:
```

89

```
[if {[value blackpoint]==1} {return 2} {return 3}]
```

90

```
[if {[value blackpoint]==1} {return True} {return False}]
```

91

```
[if {[value blackpoint]==1} {return blackpoint} {return whitepoint}]
```

92

```
[if {[value filter]=="gaussian"} {return filter} {return False}]
```

93

94

```
#OTHER METHOD
```

95

```
condition ? then : else
```

96

97

```
#Example:
```

98

```
#if (r==1)? return 0: else (return r*2)
```

99

```
r ==1 ? 0 : r*2
```

10

0

10

1

```
#-----
```

10

2

```
#PATH MANIPULATIONS:
```

10

3



10

4

#Filepath without extension:

10

5

[file rootname [value [topnode].file]]

10

6

10

7

#Filename only:

10

8

[basename [value [topnode].file ]]

10

9

11

0

#Filename only without extension:

11

1

[basename[file rootname [value [topnode].file]]]

11

2

11

3

#-----

11

4

#RELATIVE PATH

11

5

#In the Read node you can use the relative path for your footage/Obj

11

6

11

7

#In your Read Node in the knob "file", use this:

11

8

11

9

#Read file "render.exe" in the same folder of your file nuke

12

0

[python {nuke.script\_directory()}]/render.exr

12

1

12

2

#Read file "render.exe" in the subfolder where your file nuke is

12

3

[python {nuke.script\_directory()}]/folder/render.exr

12

4

12

5

---

#Read file "render.exe" in the subfolder of your .nuke folder

---

12

6

---

[python {" /Users/gere/.nuke } ]/folder/render.exr

---



---

Revision #1

Created 20 September 2021 09:22:30 by Anthony

Updated 20 September 2021 09:22:46 by Anthony