



19 Editors - Drivers Editor

Table of content

Drivers Editor.....	2
Time cursor.....	3
F-Curves.....	3
Handles.....	4
Viewport Navigation.....	4
Viewport navigation.....	4
F-Curve Context Menu.....	4
Copy.....	4
Paste.....	4
Paste Flipped.....	4
Last operator Paste Keyframes / Flipped.....	4
Offset.....	5
Type.....	5
Flipped.....	5
Handle Type.....	5
Last Operator Set Keyframe Handle Type.....	5
Type.....	5
Interpolation Mode.....	5
Last Operator Set Keyframe Interpolation.....	6
Type.....	6
Easing Mode.....	6
Last Operator Set Keyframe Easing Type.....	6
Type.....	6
Insert Keyframes.....	6
Duplicate.....	6
Last Operator Duplicate.....	6
Mode.....	6
Values X / Y.....	6
Axis.....	6
Orientation.....	6
Proportional editing.....	6
Proportional Falloff.....	6
Proportional Size.....	7
Connected.....	7
Projected(2D).....	7
Delete Keyframes.....	7
Mirror.....	7
Last Operator Mirror Keys.....	7
Type.....	7
Snap.....	7
Last Operator Snap Keys.....	7
Type.....	7
Slider snapping.....	7
Quick Favorites menu.....	8
Editing the expression from the property.....	8
Simple Expressions.....	8
Variable Names.....	8

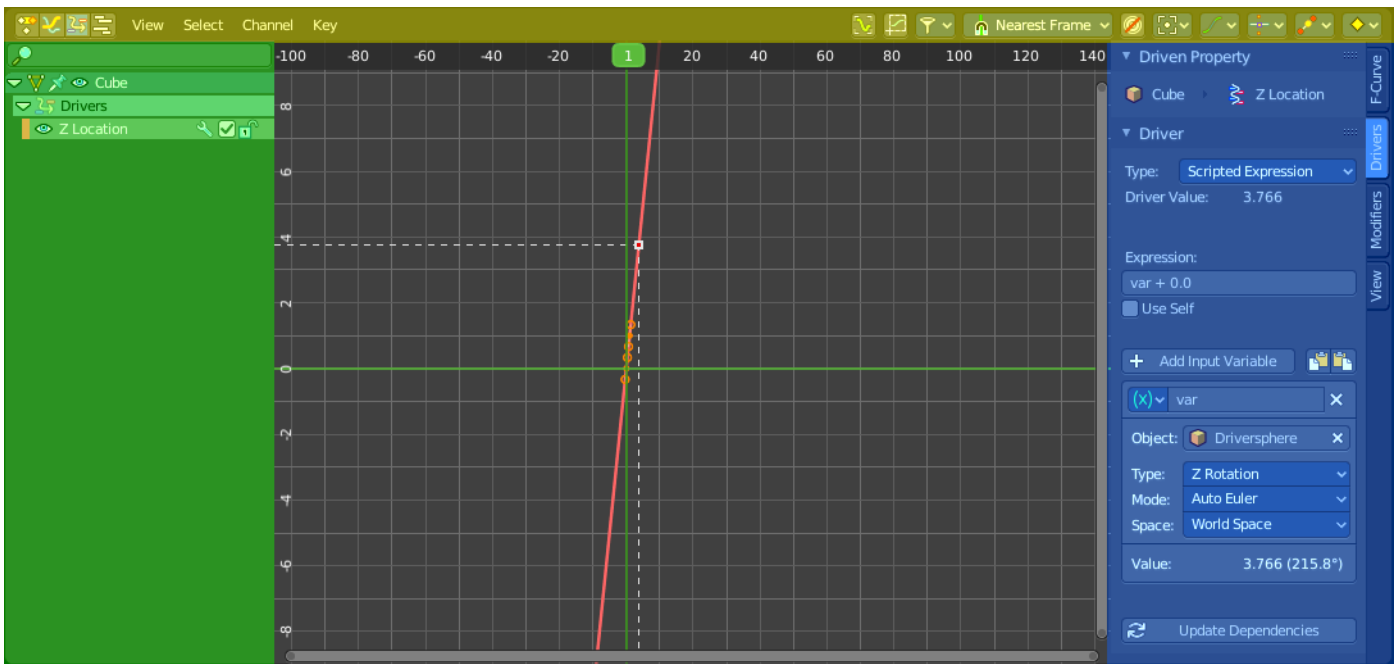
Literals.....	8
Globals.....	8
Constants.....	8
Operators.....	9
Functions.....	9
Drivers setup example.....	9
Control movement of a cube by rotation of a sphere.....	9
Add input variable.....	11

Drivers Editor

Drivers are a way to control values or properties by other values or properties. This is done by expressions. As an example, you can use the Y rotation of a sphere to control the X movement of a cube. The rotation of the sphere drives the cube then.

The Drivers Editor is the place to adjust the settings for drivers.

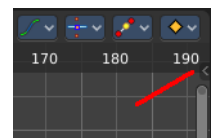
The Drivers editor has several areas.



Header (Yellow)

Channel list (Green).

Sidebar (Blue). The sidebar needs to be revealed, which can be done by clicking at the small triangle button up right.



Viewport (no color)

The header is divided into two parts. Left tools and menus. Right Options.



Menus (Green)

Options (Yellow)

Time cursor

The Time Cursor is the green slider at the top. It is used to set and display the current time frame.



F-Curves

The timeline displays the function curves for the driver. This curve is controlled by the expression. But this curve can also be manipulated in various ways like any other F-Curve. Note that manipulating the curve does not manipulate the expression. It adds on top of it. And there is no way to reset it but create the driver from scratch. So be careful with manipulations.



Handles

Every curve point has handles assigned. The curve can be manipulated by dragging these handlers. You can also change the handle type in the Keyframe Handle Type menu in the header. To make the curve sharp at this keyframe for example.

Viewport Navigation

Navigation in the viewport happens by mouse or hotkeys. Some of them does not have a menu entry. And needs to be explained here.

Viewport navigation

Clicking left at the number bar moves the frame marker.

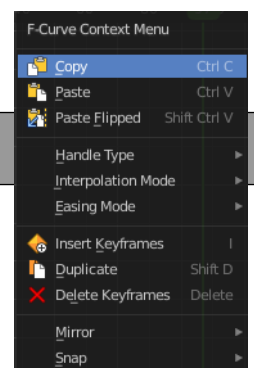
Middle mouse button pans the view.

Holding ctrl + middle mouse button zooms the view.

Scroll Wheel zooms the view.

To manipulate a curve point, grab one of its handlers and drag.

F-Curve Context Menu



When you double right click into the viewport then you will call the F-Curve context menu.

Copy

Copies the currently selected curve point(s).

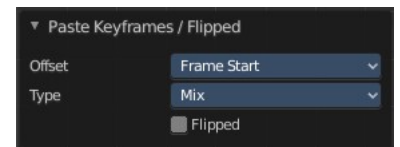
Paste

Pastes copied curve point(s)

Paste Flipped

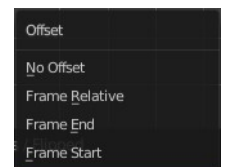
Pastes copied curve point(s), but flipped.

Last operator Paste Keyframes / Flipped



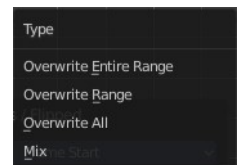
Offset

Define a time offset to paste the keys.



Type

The paste method.

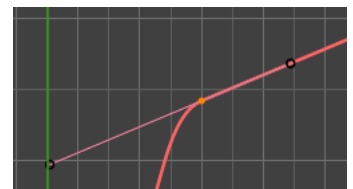
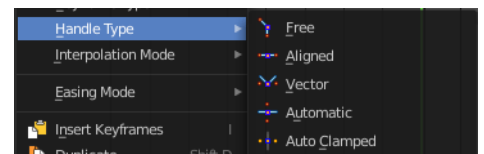


Flipped

Paste copied curve point(s) flipped.

Handle Type

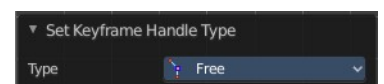
Set the handle type for the currently selected curve point.



Last Operator Set Keyframe Handle Type

Type

Set the handle type for the currently selected curve point.



Interpolation Mode

The Interpolation mode defines how the curve acts from keyframe to keyframe. You can have a linear curve between two keyframes instead of a bent one for example.

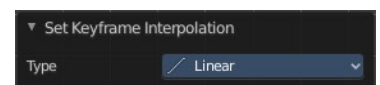


The easing methods here in the interpolation mode menu are for the easing shape. There is also an easing menu where you can choose a easing method.

Last Operator Set Keyframe Interpolation

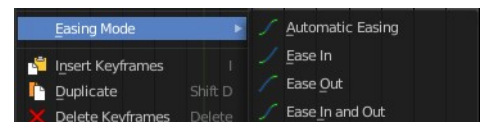
Type

Set the interpolation mode.



Easing Mode

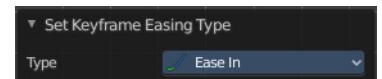
The easing methods in the interpolation mode menu are for the easing shape. This menu allows you to choose an easing method.



Last Operator Set Keyframe Easing Type

Type

Set the easing type.



Insert Keyframes

Insert a keyframe at the current position. This functionality does not work from the Drivers editor.

Duplicate

Duplicate the selected curve point(s).

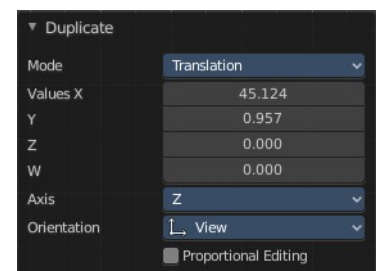
Last Operator Duplicate

Mode

Values X / Y

The x and y values for the pasted keyframes. Note that these values starts at the position of the original copied keyframe. These values are relative.

Values Z and W have no effect here.



Axis

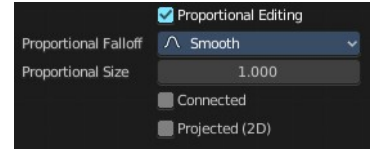
These values have no effect.

Orientation

These values have no effect.

Proportional editing

Enables proportional editing. Activating proportional editing reveals further settings.



Proportional Falloff

Adjust the falloff methods.

Proportional Size

See and adjust the falloff radius.

Connected

The proportional falloff gets calculated for connected parts only.

Projected(2D)

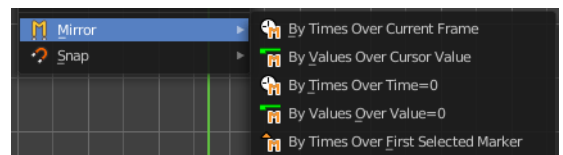
The proportional falloff gets calculated in the screen space. Depth doesn't play a role. When it's in the radius, then it gets calculated.

Delete Keyframes

Delete the selected curve point(s).

Mirror

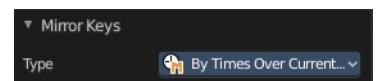
Mirrors the animation by the given method.



Last Operator Mirror Keys

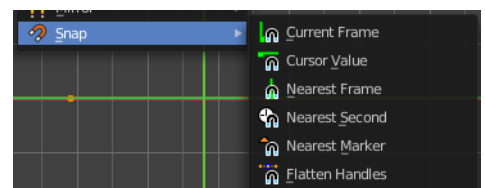
Type

Flips the selected keyframes over the current frame position by the chosen method.



Snap

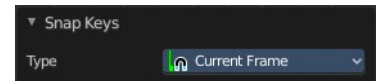
Snaps the selected keyframes by the given method.



Last Operator Snap Keys

Type

Snaps the selected keyframes by the chosen method.



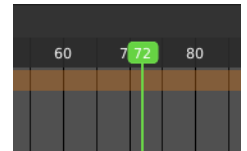
Slider snapping

Snapping also works at sliders. Hover with the mouse over the slider, start to slide, and holding down **ctrl** will snap the sliders in incremental steps.



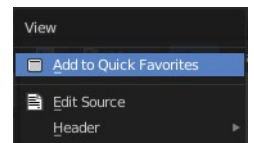
When it's a default value between 0 and 1 then it usually snaps in 0.1 steps. When it's a default value over 1 then it usually snaps in steps of 10.

The increment snapping also works at the frame slider. Here the incremental snapping happens by the frame rate that you have defined. With a frame rate of 24 it will snap in steps of 24 frames when holding down ctrl.

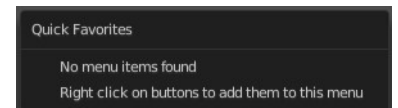


Quick Favorites menu

When you right click at a menu or a button, then a right click menu will open. Tools have usually a Add to Quick Favorites menu entry.



The Quick Menu is empty by default. With Add to Quick favorites you can add this menu to the Quick menu.

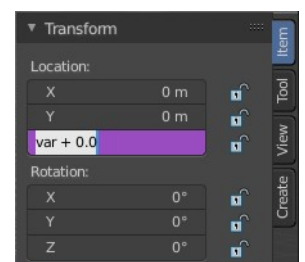
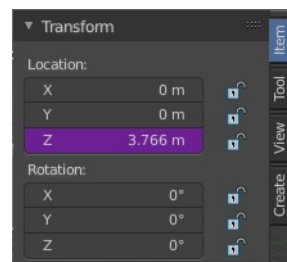


In the 3D view we have a menu called Quick in the header, which shows this content then. In the Dope Sheet Editor you can just call it with its hotkey. Q. It has no regular menu entry here.

Editing the expression from the property

A property with a driver attached will turn pink.

When you click into the edit box, then the expression for the driver will appear. And you can edit this expression now without to edit the driver again.



Simple Expressions

There are some advanced expression methods, called Simple Expressions.

Variable Names

Use only ASCII characters.

Literals

Floating point and decimal integer.

Globals

frame

Constants

pi, True, False

Operators

+, -, *, /, ==, !=, <, <=, >, >=, and, or, not, conditional operator/ ternary if

Functions

min, max, radians, degrees, abs, fabs, floor, ceil, trunc, int, sin, cos, tan, asin, acos, atan, atan2, exp, log, sqrt, pow, fmod

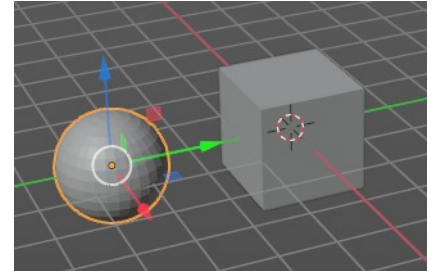
Simple expressions are evaluated even when Python script execution is disabled.

When an expression outside of this subset is used, Blender displays a “Slow Python expression” warning. However, as long as the majority of drivers use simple expressions, using a complex expression in select few is OK.

Drivers setup example

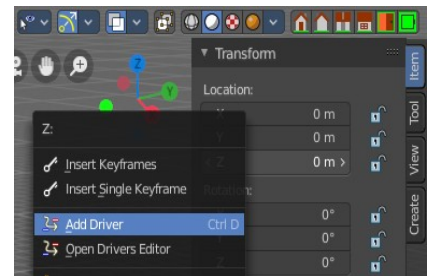
Control movement of a cube by rotation of a sphere

Create a cube and a sphere, and place the sphere a bit off. We want to be able to select and modify it.



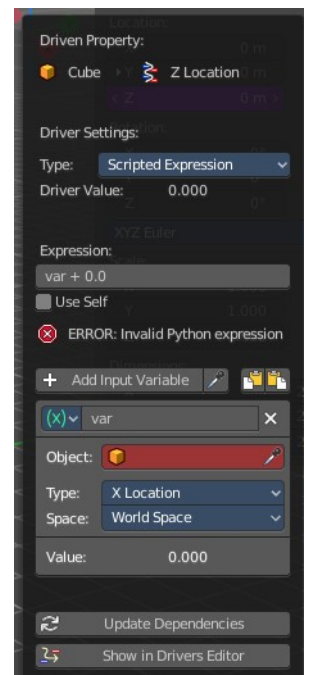
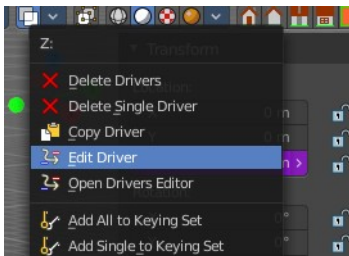
Select the cube.

In the Transform panel in the sidebar right click at the Z value and choose Add Driver. We will add the driver to the Z axis property of the cube.



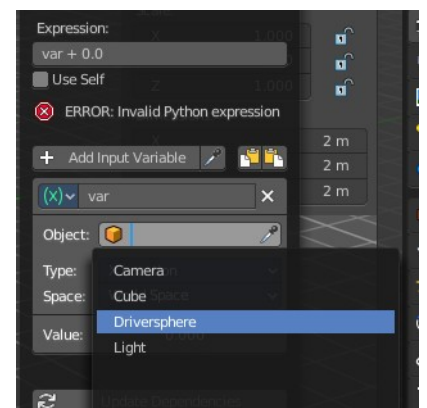
When you add a driver a panel will pop up where you can do the drivers setup.

In case you miss this panel by too fast clicking, it can be called again by right clicking at the property again and choose Edit Drivers from the menu.



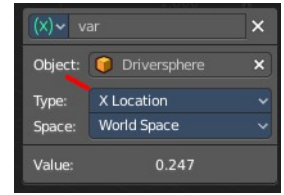
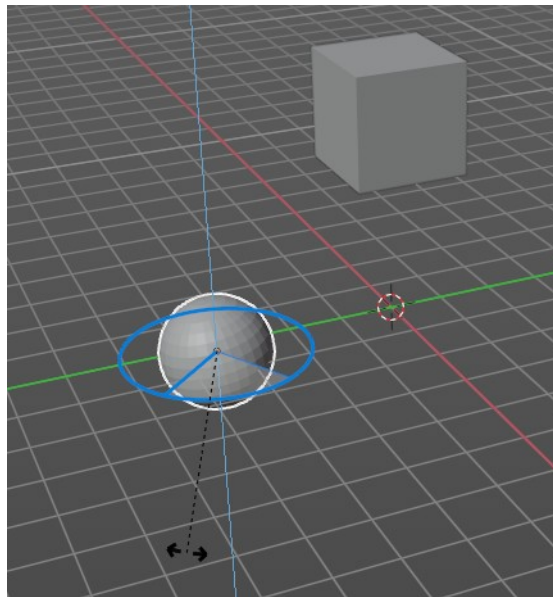
Don't worry about the ERROR: invalid Python expression at this point. we are going to change this now.

In the red field with the Object label choose the sphere.



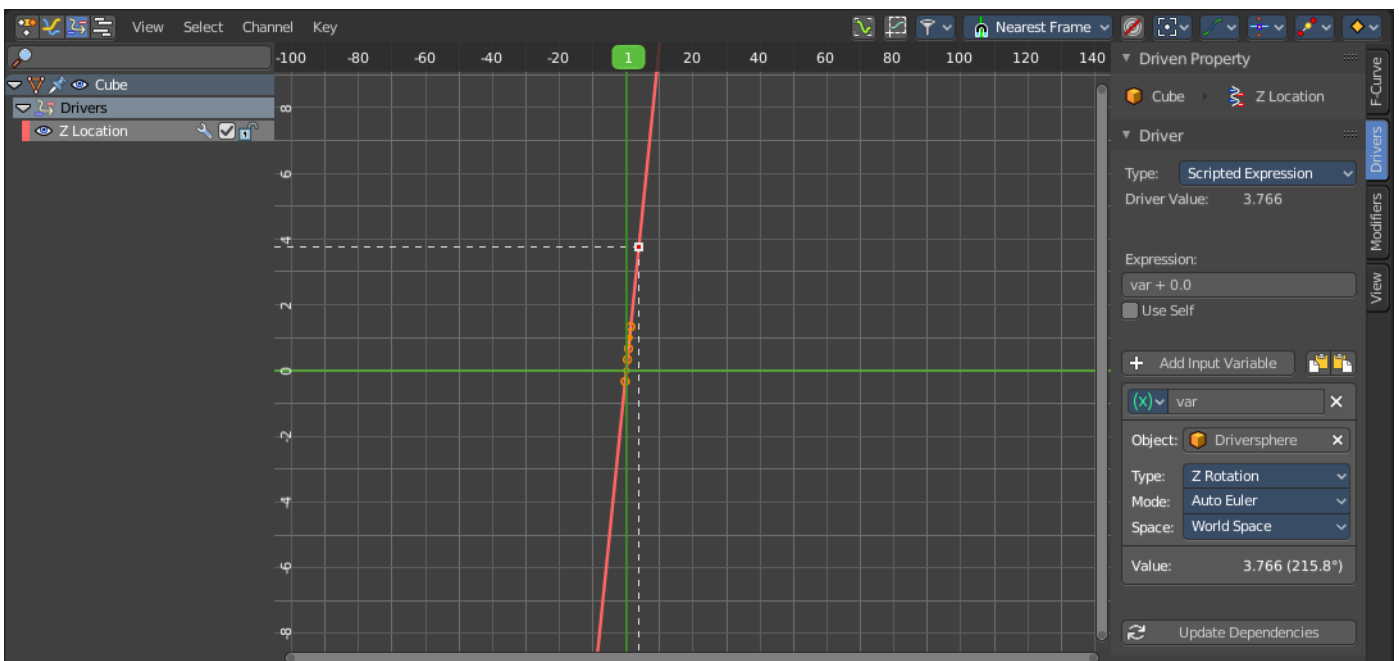
By default the X value influences now the Z position of the cube. So when you move the sphere in X direction, the cube will move in Z direction.

Let's change this to rotation around Z axis. Now the cube moves in Z axis when we rotate our sphere in Z axis.

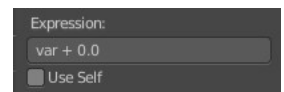


Let's have a look into our Drivers editor.

We have now content available- The channels list has an entry. And when you click at the Z Location channel in the channel list then the content and the driver settings panel in the sidebar will appear. It is the same than the one from the right click menu.



Let's have a closer look at how this all works. The Expression field is the place where the magic happens. When you change this expression then you can change the behavior of the driver. For example instead of `var + 0.0` you could add a multiplier here to our variable called `var`. Let's say `var *2`. Then the cube moves in double speed of the rotation of our sphere.

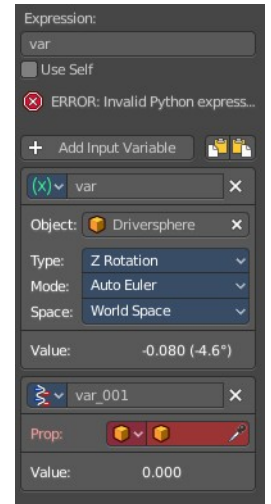


Add input variable

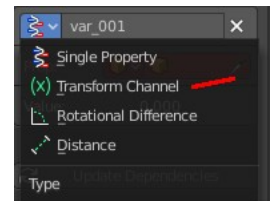
You could also add a second input value, and connect it with another object. Let's say a cone.

Add a cone object.

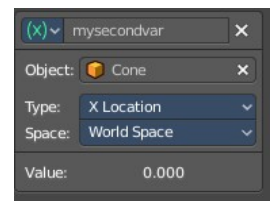
In the Driver add a new input variable.



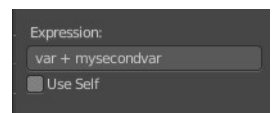
Change this input variable to type Transform Channel.



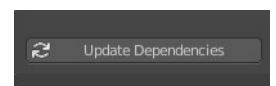
Now in the object field choose our cone object. And rename the input variable to something meaningful. Again set the type to your needs.



Now let's change the Expression to this: var + mysecondvar.



As a last step update the dependencies by clicking at the Update Dependencies button at the end, which will remove the Invalid Python Expression error.



And now the driver is controlled by the Z rotation of our sphere, which is defined in the variable var. PLUS the x location of the cone object, which is defined in the variable mysecondvar.