

# 1 Snippets

## 1.1 Find all image textures path

```
for img in bpy.data.images:  
    print(img.name, ":", img.filepath)
```

## 1.2 Manipulate selected objects

```
for obj in bpy.context.selected_objects:  
    obj.rotation_euler.x += 1.5708 # Rotate 90 degrees (/2) on X-axis
```

# 2 Shortcuts

## 2.1 Selection & Navigation

Shortcut	Effect
Ctrl + `	Hide/Show gizmos
Tab	Toggle between Object Mode and Edit Mode
A	Select all
Alt + A	Deselect all
L	Select linked geometry (hover over a part and press L)
Ctrl + L	Select all linked geometry (based on selection)
B	Box select
C	Circle select
Shift + G	Select similar (choose criteria like area, shape, or material)
Alt + RMB	Loop select
Shift + RMB	Ring select

## 2.2 Transformations

Shortcut	Effect
Ctrl + `	Hide/Show gizmos
Tab	Toggle between Object Mode and Edit Mode
A	Select all
Alt + A	Deselect all
L	Select linked geometry (hover over a part and press L)
Ctrl + L	Select all linked geometry (based on selection)
B	Box select
C	Circle select
Shift + G	Select similar (choose criteria like area, shape, or material)
Alt + RMB	Loop select
Shift + RMB	Ring select
G	Grab (move)
R	Rotate
S	Scale
X / Y / Z	Constrain movement to an axis (e.g., G + X moves along the X-axis)
Shift + X / Y / Z	Move along the other two axes (exclude one axis)
Ctrl + A	Apply transformations (use in Object Mode)
Ctrl + Tab (or 1, 2, 3 in Blender 2.8+)	Switch between Vertex, Edge, and Face selection
Ctrl + E	Edge menu (Bevel, Mark Seam, etc.)
Ctrl + B	Bevel (works for edges and vertices)
Shift + Ctrl + B	Vertex bevel
F	Fill (creates a face between selected vertices/edges)
Alt + Left Click	Select edge loop
Shift + Alt + Left Click	Select multiple edge loops

Shortcut	Effect
<b>Ctrl + R</b>	Loop cut (scroll mouse wheel to increase cuts)
<b>K</b>	Knife tool (click to cut, Enter to confirm)
<b>Shift + R</b>	Repeat last action
<b>Ctrl + Shift + B</b>	Chamfer/Bevel vertices

## 2.3 Extrude, Inset & Merge

Shortcut	Effect
<b>E</b>	Extrude
<b>I</b>	Inset faces
<b>M</b>	Merge vertices (choose options like "At Center" or "At Last")
<b>Alt + M</b>	Older version of merge (pre-2.8)

## 2.4 Proportional Editing & Smoothing

Shortcut	Effect
<b>O</b>	Toggle Proportional Editing
<b>Shift + O</b>	Change proportional falloff type
<b>Ctrl + Shift + B</b>	Bevel vertices
<b>Shift + S</b>	Snap menu (snap selection to grid, cursor, etc.)
<b>U</b>	Unwrap UV (when in UV Editing)
<b>Ctrl + T</b>	Triangulate faces
<b>Alt + J</b>	Convert tris to quads

## 2.5 Miscellaneous

Shortcut	Effect
<b>H</b>	Hide selection
<b>Alt + H</b>	Unhide all
<b>Shift + H</b>	Hide everything except selection
<b>P</b>	Separate selection into a new object
<b>Ctrl + J</b>	Join selected objects

- Edit mode UV tools: press U
- Edge slide tool: in edit mode, with a vertex selected, press Grab (G) twice
- Triplanar projection: [https://www.youtube.com/watch?v=KV\\_hgeQdCXk](https://www.youtube.com/watch?v=KV_hgeQdCXk)
- Baking: [https://www.youtube.com/watch?v=sOvRr\\_D8ZpU](https://www.youtube.com/watch?v=sOvRr_D8ZpU)

## 2.6 Aside: Index Of Refraction for Metals

Blender's Principled BSDF shader expects a single float for the Index of Refraction (IOR), which is suitable for dielectrics but not ideal for metals, which use complex IOR values (with real and imaginary components) or RGB-based IOR values.

### Handling RGB IOR in Blender's Principled BSDF

Since Blender's Principled BSDF does not support complex or RGB IOR directly, you have a few options:

1. *Use the Specular Tint and Metallic Inputs*
  - Set Metallic = 1 to define the material as a metal.
  - Use the Base Color to approximate the reflection color.
  - Adjust Specular Tint to shift the highlights towards the desired reflectance color.
2. *Convert RGB IOR to a Specular Value Approximation*
  - If you have an RGB IOR value (e.g., from [refractiveindex.info](http://refractiveindex.info)), you can approximate the reflection using the Fresnel equations. For metals, the reflectance  $\bar{R}$  at normal incidence is

$$R = \frac{(n - 1)^2 + k^2}{(n + 1)^2 + k^2}$$

where

$n$  real part of the Index of Refraction (influences refraction and reflection)

$k$  complex part of the Index of Refraction (influences absorption)

This reflectance value is computed for each color channel, and the result is the *Base Color*

3. *Use a Shader Mix with Fresnel Inputs*: Instead of using the Principled BSDF, you can create a custom Shader Mix using:
  - A Glossy BSDF with color based on the RGB IOR.
  - A Fresnel node or Layer Weight node to drive the blending.
4. *Use a Node Group That Supports Complex IOR*: Some custom node setups, like "Complex IOR" node groups, allow you to input both  $n$  and  $k$  values and compute proper Fresnel reflectance. These setups typically use Math nodes to handle the complex Fresnel equations.

Video Reference: <https://www.youtube.com/watch?v=yX1CJZswHPs>

### 3 Pivot to Cursor

Press **Shift + Right Click** to place the 3D Cursor manually. Or use **Shift + S** → "Cursor to Selected" to place it at the selection.

Instead, to change the pivot point to cursor, do the following:

- In *Object Mode*, go to the top-center of the viewport (next to the selection mode dropdown) where the pivot point options are.
- Click on the Pivot Point dropdown (it's an icon that usually shows a circle with a dot in the center).
- Select 3D Cursor from the list of pivot options.

Alternatively:

- Period key (.) to open the pivot point menu and choose 3D Cursor.
- Perm
- Object - Set Origin

### 4 Sharing: Asset Browser

- Video Reference: <https://www.youtube.com/watch?v=cbzBt60dhY8>
- Blender docs: [https://docs.blender.org/manual/en/4.3/editors/asset\\_browser.html](https://docs.blender.org/manual/en/4.3/editors/asset_browser.html)

The *Asset Browser* is the main interface to organize Models, Materials, Geometry Nodes, ... **Poses** Across blend files.

To store our own assets, the first step is to choose the **Assets Folder**

#### Where are my Assets

1. Go To Edit/Preferences/File Paths/. Under Assets Libraries, add the folder for your assets (and possibly, remove the preexisting one).

**Library Paths** do not support relative paths of the form `//AssetBrowser/`<sup>1</sup>

Once registered, you can select the asset library in the Asset Browser. All the blend-files in the asset library will be scanned for assets, and all those assets will be shown in the Asset Browser.

2. Focus on the **Import Method** Option. This is the import method which will be used if in the *Asset Browser Editor* the *Import Method* Is set to *Follow Preferences*. Either way, the possible options are
  - **Link** Same as bringing the content of another blend file with *File/Link*, Meaning the assets will be linked and read only. Changes to the original will be reflected in the linked references. To introduce variations, we need to use *Library Overrides*
  - **Append** Same as **Copying** the content of another blend file with *File/Append*
  - **Append (Reuse Data)** Same as Append, but if the same asset is appended multiple times, Blender will try to reuse as much data as possible (if you introduce) variations on one instance of the asset, blender will need to duplicate the data though

Assets in a asset library can be organized into virtual folders called **Asset Catalogues**

#### Asset Types

Assets can be broadly divided into two types: primitive and preset assets. Which is which depends on the Data-Blocks type.

- *Primitive assets* are data-blocks that are either linked or appended to the current file. Examples are objects, materials, and worlds. These can be dragged from the Asset Browser into the scene (objects and worlds), or onto existing objects (materials).

---

<sup>1</sup>Even if there is a relative path checkbox, it's broken as of blender 4.3



Figure 1: Overrideable linked data blocks

- *Preset assets* are data-blocks that are loaded and then applied to something or activated. An example is a pose asset. When applying the pose, the data-block is loaded from its blend-file, and then the pose is applied to the active armature. Brush assets are an example of an asset type that is activated. They get loaded into the current file and activated for painting or sculpting, but don't get saved in the file.

## Asset Creation

To create a **Primitive Asset**

1. Create a `.blend` file into the Asset Folder (configured in the "user" blend file) (also a subfolder of the asset library works)
2. Create your mesh (or whatever else)
3. From the **Outliner**, right click on the object and *Mark As Asset* (the generated preview can be replaced with an arbitrary image, eg a render)

As you can see, the default generated preview image doesn't render materials for a mesh asset, and that can be confusing. While you can create a quick render, add a fake user to it, pack resources, and set that as a preview manually. **Quicker way (if a color is enough):**

1. Select the material(s) assigned to the mesh asset
2. Copy the base color (*Properties Editor*, Materials Tab with the selected material)
3. Scroll down to the *Viewport display* tab, paste the base color into the *Color* property

Note: If you imported a primitive asset with the **Link** method (which is what you should use), you cannot edit anything about it, as all properties are taken from the original source

To add some modifications (which will be stored in a non destructive, additive way) to a linked imported asset, you need to add some **Library Overrides**.

1. Select the data blocks you want to override (they are recognizable from the *Outliner* because they have a chain icon (figure 1))
2. Right click, *Library Override/Make/Selected*. Example: If you override only the object data block, you can freely transform the imported object
3. As a side note, you can create many variants of the same asset in this way
4. **Note:** Linked assets cannot be renamed, even if you override them. Solution: Put them inside a singleton collection

Final Notes on Primitive Assets:

- If you make a Primitive asset out of a **Skinned Mesh**, whenever you import it, its armature will come along with it automatically, because its *Armature Modifier* references the armature object.
- An *Action* can be marked as an asset by right clicking it from the *Dope Sheet Editor*, Action Editor, or from the *Outliner*, Blender File display mode, and selecting **Mask as Asset**

**Preset Assets** each have their own dedicated button. We'll focus on **Pose Assets**. To use them, we need to first and foremost **enable the add-on** called **Animation: Pose Library**. Basics covered here, more on Posing (subsection 5.5.4). Usually, pose assets are exported alongside a primary skinned mesh asset in its rest position.

## Pose Library

In the *Asset Browser*, Select from the topbar *Asset/Create Pose Asset*, and will create an asset with the current pose of the selected skinned mesh

# 5 Animations: Docs Summary

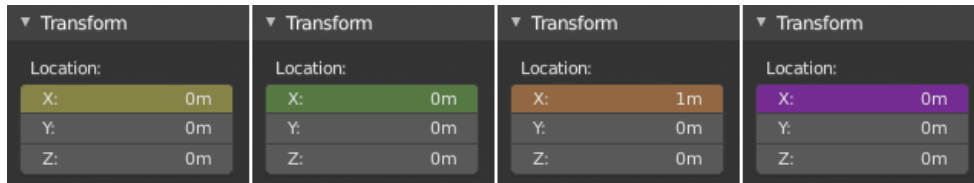
These notes about animations in Blender are made by summarizing the Blender 4.3 Docs.

## 5.1 Introduction

Animation = Transforming an object or changing its shape over time. More generally, any property about a blender object can be animated.

Animation is typically achieved by employing *Keyframes* (more on that later)

Any property in the *Properties Editor* has a *State Color*



Color	Meaning
Gray	Not animated
Yellow	Changed from the current frame
Green	Keyframed on a different frame
Orange	Changed from the keyframed value
Purple	Controlled by a <i>Driver</i>

### 5.1.1 Rigging

Reference Rig: <https://www.blendswap.com/blend/23220>

Rigging = adding controls handles to animate an object. Blender offers the following feature to rig a model

Rigging Method	Brief
<b>Armatures</b>	A Hierarchy of Joints associated with a mesh. Each joint has a <i>*weight*</i> [0.0, 1.0] for each vertex of the aforementioned mesh (can be painted). Transforming a joint will influence all vertices whose weight for that particular joint is greater than 0. This technique is called <i>*Skeletal Animation*</i> (more later).
<b>Constraints</b>	Control the kind of motion the rig is allowed to perform. They are found under <i>Properties Editor</i> , tab "Constraints" (more later).
<b>Object Modifiers</b>	Mesh deformation through modifiers. We are interested in Deformations and Physics (more later).
<b>Shape Keys</b>	Commonly called textitblendshape, meaning having different copies of a mesh (same topology, same UV, same <i>*everything*</i> ). Example: different facial expressions that blend over time with <i>Keyframing</i> (more later).
<b>Drivers</b>	Mechanisms to control multiple properties at once and make some properties automatically update when others change (more later).

## 5.2 Keyframes

### 5.2.1 Relevant Shortcuts

With a property/object Selected:

Shortcut	Effect
<b>I</b>	Insert Keyframe (brings up keyframe menu)
<b>Alt + I</b>	Delete Keyframe
<b>Shift + I</b>	Insert Keyframe for all properties
<b>Ctrl + I</b>	Add keyframe to active keying set
<b>Alt + S</b>	Reset Scale (useful when animating transforms)
<b>Alt + R</b>	Reset Rotation
<b>Alt + G</b>	Reset Location

Inside *Graph Editor* or *Dope Sheet Editor*

Shortcut	Effect
<b>G</b>	Move keyframe(s)
<b>S</b>	Scale keyframe(s)
<b>R</b>	Rotate keyframe handle (in Graph Editor)
<b>Shift + D</b>	Duplicate keyframe(s)
<b>X</b> or <b>Delete</b>	Delete keyframe(s)
<b>E</b>	Extrapolate (Graph Editor)
<b>T</b>	Set Keyframe Interpolation (Linear, Bezier, Constant, etc.)

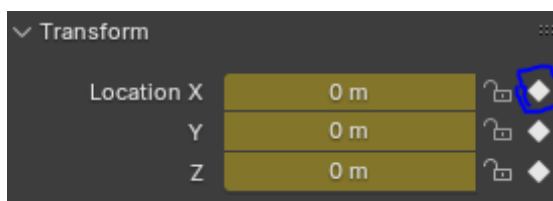
Shortcut	Effect
<b>V</b>	Set Keyframe Handle Type (Vector, Aligned, etc.)
<b>Ctrl + C</b>	Copy Keyframe
<b>Ctrl + V</b>	Paste Keyframe

## Playback Shortcuts

Shortcut	Effect
<b>Spacebar</b>	Play/Pause animation
<b>Shift + Left Arrow</b>	Jump to <b>start frame</b>
<b>Shift + Right Arrow</b>	Jump to <b>end frame</b>
<b>Left Arrow</b>	Move <b>one frame backward</b>
<b>Right Arrow</b>	Move <b>one frame forward</b>
<b>Up Arrow</b>	Move to <b>next keyframe</b>
<b>Down Arrow</b>	Move to <b>previous keyframe</b>
<b>Shift + Ctrl + Spacebar</b>	Play animation in <b>reverse</b>
<b>Home</b>	Zoom to fit all keyframes in <b>Graph Editor/Dope Sheet</b>
<b>Ctrl + Middle Mouse Scroll</b>	Zoom in/out in Timeline/Graph Editor

When you set a keyframe on a simple static mesh, like a cube.  
(in the Viewport, object mode, Ctrl + A -> Mesh -> Cube). If

- You press **I**, then all the transform properties are saved in the current frame as a keyframe (see in *Dope Sheet Editor*)
- If you want only a part of the default properties to be saved, then you can set them manually by clicking the *Animate Property* handle to the right of the property in the *Properties Editor*

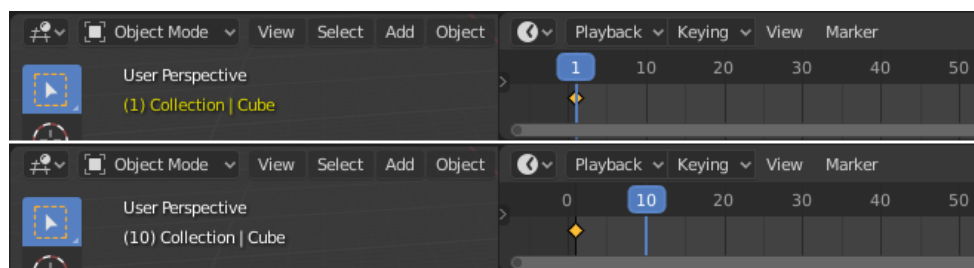


### 5.2.2 Introduction

A *Keyframe* is a marker of time which stores the value of the selected property.

The purpose of a keyframe is to save the value of a property in a given instance of "time" (on a rendered frame. Physical elapsed time depends on the FPS of the animation).

An overview of all the existing keyframe in your animation can be seen in the *Playback Editor*. To get the full information about existing keyframes (ie. to which object they refer to and which property they alter/set, use the *Dope Sheet Editor*).



## Quick Experiment: Keyframe Visualization

File: 01\_Keyframes\_Intro-Moving\_Cube.blend

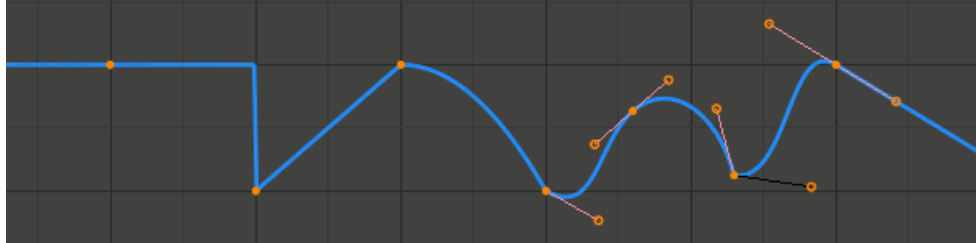
1. Create an empty Blender scene with a cube and check FPS (*Properties Editor* → Output → Frame Rate).
2. Select its position from the *Properties Editor* and set a keyframe (**I** to keyframe the transform).
3. Move to another keyframe inside *Dope Sheet Editor* or *Playback Editor*.
4. Freely transform your cube and then set a keyframe.

**Note:** It's imperative that you first move to another place in the timeline and then manipulate your object, otherwise it won't work!

- Go back to the start of the timeline (**Shift + Left Arrow**) and play the animation (**Spacebar**).  
Keep this example for the next section on Interpolation.

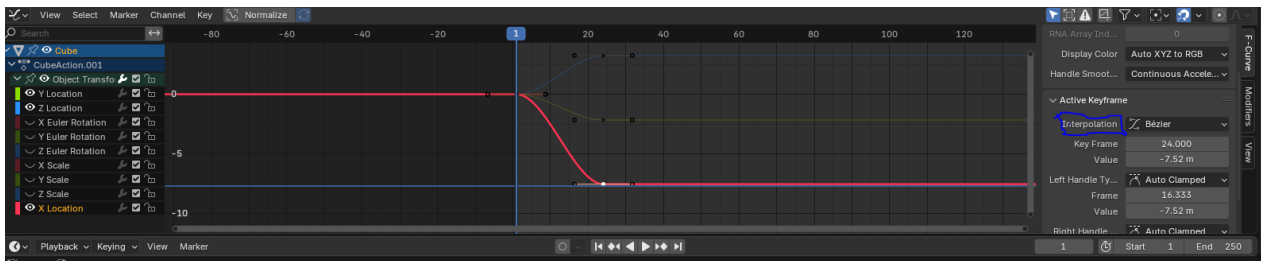
## Interpolation

When you set two keyframes on the same property, its value changes over the span of frames inbetween the two keyframes with *Interpolated Values*, ie values computed using a mathematical formula. In particular, such formula is defined by an *F-Curve*, manipulated in the *Graph Editor*.



There is 1 curve for each animated property in the *Dope Sheet Editor*. The main setting is the *Interpolation Type*, which appears in the *Graph Editor* inside the *F-Curve* Tab. **Interpolation Modes:**<sup>2</sup>

- Bezier Curve
- Linear
- Constant



While what happens during the transition between a keyframe and the next one is defined by the *Interpolation Mode*, What happens outside the "Keyframed Range" (before the first keyframe and after the last keyframe) is defined by the *Extrapolation Mode*.

*Extrapolation Mode* is found under "*Graph Editor/Channel/Extrapolation Mode*" or with shortcut **Shift + E** (*Graph Editor* Selected) The following are the available **Extrapolation Modes:**<sup>3</sup>

- Constant: Continue in a straight horizontal line
- Linear: Continue in a straight line keeping the slope
- Make Cyclic: Repeat the curve
- Clear Cyclic: Removes Cycles Modifier

The settings to manipulate Curve Handles (placed on the F-Curve on the keyframe positions) depend on the Interpolation Type. A common setting among them all is the *Auto Handle Smoothing*, which can be either *None* or *Continuous Acceleration*.

When not *None*, edits to a handle are propagated in the near handles (similar to proportional editing) to keep the F-Curve as smooth as possible.

## Quick Experiment: Interpolation and Extrapolation: "Cyclic Overshoot"

File: 02\_Keyframes\_Interpolation-Moving\_Cube\_Custom\_Interpolation.blend

- Open the cube example you produced from the previous experiment
- Open the *Graph Editor* and select a "Location" Curve (the one with the bigger displacement in the Vertical axis)
- Play around with the 2 Handles freely. Example: Use the last one as "Bezier" Interpolation and create an "overshoot"
- Change the *Extrapolation mode* to Linear and then to Make Cyclic
- Check in the "*Graph Editor/Modifiers*" (Tab) that The Cycles Modifier has been added to the *F-Curve*
- Go back to the start of the timeline (**Shift+Left Arrow**) and play the animation (**Spacebar**)

<sup>2</sup>All the settings inside the *F-Curve* Tab affect the keyframes selected

<sup>3</sup>Extrapolation Mode affects all *F-Curves* selected



Figure 2: **To the left**, keyframe types. **To the right**, bezier interpolation handles type

*Keep this example for the next section on Interpolation.*

## Keyframe Types

Blender has different keyframe types. Such type is determined by the keyframe's source (eg set manually or automatically generated), as well as the effect they achieve.

To change a keyframe type

1. In the *Dope Sheet Editor*, select the keyframes you want to change
2. Click the **Menu Key** (to the left of the right CTRL key) to display the popup context menu, therefore click "KeyFrame Type" and select your keyframe type

Note: **This has no functional purpose**, it is merely a visual tool to help us navigate into complicated animations.

Another visual tool the *Dope Sheet Editor* and *Graph Editor* give us are *Markers*, which can assign a name to a keyframe. To create a marker:<sup>4</sup>

1. select the keyframe you want to label
2. press **M**
3. then select the marker and, from the top menu of the editor, click "Marker/Rename Marker"

Here's the list of the available Keyframe types, their visual appearance, and their meaning

Keyframe Type	Appearance (Not Selected / Selected)	Meaning
<i>Keyframe</i>	white / yellow diamond	Normal keyframe.
<i>Breakdown</i>	small cyan diamond	Breakdown state. e.g. for transitions between key poses.
<i>Moving Hold</i>	dark gray / orange diamond	A keyframe that adds a small amount of motion around a holding pose. In the Dope Sheet it will also display a bar between them.
<i>Extreme</i>	big pink diamond	An "extreme" state, or some other purpose as needed.
<i>Jitter</i>	tiny green diamond	A filler or baked keyframe for keying on ones, or some other purpose as needed.
<i>Generated</i>	dark diamond	A key generated by some tool, for example Copy Global Transform: Fix to Camera. This keyframe type indicates to Blender and add-ons that it is safe to remove and re-generate them, so be careful when manually marking your hand-made animation with this type.

## Interpolation Mode Handles

Similarly to *Keyframe Types*, the *Dope Sheet Editor* shows Keyframe interpolation handles with *Bezier* mode differently based on the handle type given in the F-Curve tab, giving us some visual feedback without having to open the *Graph Editor*

- *Circle*: Auto Clamped (default)
- *Circle with Dot*: Automatic
- *Square*: Vector
- *Clipped Diamond*: Aligned
- *Diamond*: Free

This holds only if the left handle and right handle have the same type, otherwise, it's always displayed as a Diamond.

### 5.2.3 Keying Sets

Reference Video: <https://www.youtube.com/watch?v=5g7EbDtlKGM>

Whenever you press **I** to create a keyframe, the newly created keyframw will automatically Store and therefore animate

<sup>4</sup>the file 02\_Keyframes\_Interpolation-Moving\_Cube\_Custom\_Interpolation.blend contains an Extreme keyframe and a marker



the whole *Object Transform*, and animated properties appear grouped under the "Object Transform" group, or better, under the *Keying Set* "CubeAction", .



If we want to animate only certain properties of our objects, we would need to spend time finding them in the appropriate tab inside the *Properties Editor*, right click and "Insert Keyframe".

We can speed up the Keyframe creation process by specifying which properties we want to animate, and we can do that by creating a *Keying Set*, found inside the *Properties Editor*, under "Scene/Keying Sets"<sup>5</sup>.

Once you create a keying set, you can add properties to it by

1. Right Clicking the property (eg. from the *Properties Editor*, Object Tab)
2. Click "Add Single to Keying Set" to add the property to the active keying set

Once you added it, go back to the Scene/Keying Sets/Active Keying Set panel and make sure that

- The *Target ID Block*, ie to which object the keyframe should be applied has no object selected. This means that, regardless of which object you have selected, the keyframe will be created from the object you specify here
- You can modify the *Index* of the property, ie the order in which they appear inside the *Dope Sheet Editor*
- You can modify how the animated properties are organized into a folder hierarchy by changing the *F-Curve Grouping* (details in video)

Furthermore, if you want to export and import keying sets from a blender scene to another, you can click the *Export to File* button to generate a Python script, which, executed on a new scene, will create the keying set you exported.

<sup>6</sup> Usage Limited on a specific scenario, for example

- You have a predefined set of objects you want to animate. Therefore, you can create a Keying set which encompasses all the properties across all the different objects you need
- You can select a frame, modify the properties of *all the objects* to their desired value, and then press **I** once (instead of setting all the properties from the context menu or pressing **I** for each object)

Instead of going to the *Properties Editor*, "Scene/Keying Sets", you can set the Active Keying set also from the *Timeline Editor*, the Keying dropdown menu.

## 5.3 Actions

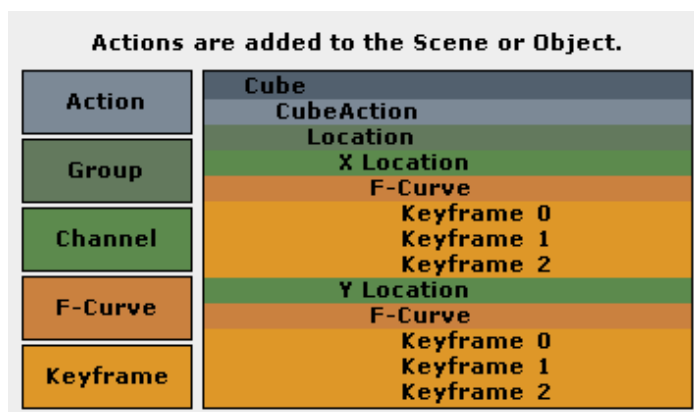
Video Reference:

- Part1: <https://www.youtube.com/watch?v=x28RWgsIu8Y7>
- Part2: <https://www.youtube.com/watch?v=LXMiHvwIgDs>

Blender saves everything as a *Data Block*, and animations are no exceptions. In fact, whenever we create a new keyframe, blender actually creates for us a new *Action*, with a default name equal to "(Object Name)Action". To visualize all Actions in your scene<sup>8</sup>

- In the *Outliner Editor*, open the *Data API* Display Mode, and click under the item "Actions"

The structure of the Action Data block is the following



Succinctly, **Actions are reusable animation segments.**

Actions can be managed from 2 places

- *Dope Sheet Editor*, with Display Mode *Action Editor* (and not the default *Dope Sheet*, which gives an overview of the current action)
- *NLA Editor* sidebar (See later *Non Linear Animations*)

This allow us to

<sup>5</sup>Once you create a keying set, the default one won't be active anymore, and you need to add at least one property to the active (selected/highlighted)

<sup>6</sup>Maybe modifying that script you can also use the `bpy.context.active_object` instead of fixing the keying set property object?

<sup>7</sup>This rig is also available for download!

<sup>8</sup>Useful if you want to remove some unused data which is still present in the .blend file

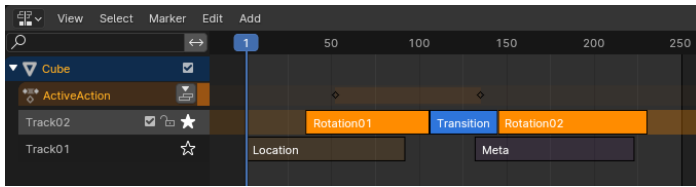


Figure 3: Example of *NLA Editor* Tracks

- Apply the same Action to different objects (as shown in the second video, in which the animation is first applied to a stand in, then applied to the final model)
- Store different animations applied to the same object (example running, walking, throwing a punch, ...), all starting from zero. Of course, we then need to **Combine** them together (Example: We want our character to run to a given position, and then throw a punch) (used with NLA)

In the *Dope Sheet Editor/Action Editor*, the Action Sidebar shows 2 checkboxes

- *Manual Frame Range*: Instead of using the range specified by the *Timeline Editor*, uses a manually specified frame range (used with NLA)
- *Cyclic Animation* The animation is Cyclic over the specified range (used with the previous checkbox)

### Non Linear Animations

- Video Reference: <https://www.youtube.com/watch?v=Hz1TwvSNsrA> (**Really Important**)
- Blender Ref: <https://docs.blender.org/manual/en/4.3/editors/nla/introduction.html>

To combine different actions, we need to use the *NLA Editor*. Brief description of the content of this editor

- Left Panel containing a dropdown menu for each object containing at least 1 action. Each dropdown menu is a **Stack of Actions**
- The Main Region displays a Stack of *Track*, in which each Track is made up of *Strips*. The top of the stack, highlighted in orange, which contains no strips, is the object's *Active Action*
- Right Panel, non empty only for the active object's action, having options *Extrapolation* (already seen), *Blending*, *Influence*. Brief:
  - *Blending*<sup>9</sup> how to combine this action's property values with the ones from the track below (similar to layer modes in Photoshop). Formulas used by each mode in the footnote link
  - *Influence* how much the actions contributes to the NLA Stack

Each Strip of an NLA Track has 2 fundamental settings

- *Extrapolation* should we hold the pose once the strip finishes?
- *Blending* how to mix with NLA tracks underneath
- *Blend In* and *Blend Out* Fade effects for when the strip starts and finishes playing

## Quick Experiment: Actions and Non Linear Animation: "Enlarge and Shoot"

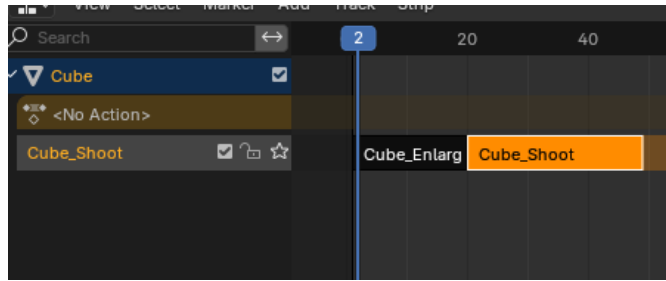
File: 03\_Actions\_NLA-Cube\_Path.blend

1. Create a Cube, and in the *Dope Sheet Editor/Action Editor* create a new Action, call it "Cube\_Enlarge"
2. Open the *NLA Editor* alongside *Dope Sheet Editor/Action Editor*, and *Push Down* (either from NLA Editor or from Dope Sheet) the current action.

As you can see, this will create a new *Track* containing a single *Strip* which contains the action we just pushed.

- Now the object has no active action, and in fact the dope sheet is empty...
  - ... But if you press **Shift + Left Arrow** and **Spacebar** (go to beginning of timeline and play), the animation still plays out, because the final animation is the **composition of the whole NLA Stack**
  - On the newly created *Track*, try to grab the *Strip* and move it along the Track's timeline. A strip can be placed wherever we want in the frame range
  - To the left of each *Track* there is a visibility checkbox. If deactivated, the *Track*'s animation won't be played out
3. Create another action to move the cube, with an "explosive" F-Curve, ie most of the displacement is done earlier in the F-Curve (therefore you need to open it in the *Graph Editor*) (it is recommended to turn off all NLA Tracks while working on another action in the *Dope Sheet*)
  4. Push down the newly created Action in the NLA Stack. This will create a new NLA Track. We don't want that, so click and drag the *Strip* in the new *Track* onto the existing track, delete the now empty track, and place the Shoot Action after the Enlarge action. The result should be something like this:

<sup>9</sup>Link: [https://docs.blender.org/manual/en/4.3/editors/nla/sidebar.html?utm\\_source=blender-4.3.2#bpy-types-animdata-action-blend-type](https://docs.blender.org/manual/en/4.3/editors/nla/sidebar.html?utm_source=blender-4.3.2#bpy-types-animdata-action-blend-type)



5. Note: **Having more strips that modify different properties will reset all the work done by the previous strip, even if Extrapolation is set to Hold in the right side panel, strip menu**

As you can see playing out the animation, after the Enlarge Action is done, the cube snaps back to its original scale and starts translating. Fix:

- Have *Strips* which modify different properties in different *Tracks*. Therefore, from the topbar of the Editor, with a track selected, Click Track/Add and move one of the two strips in the newly created Track<sup>a</sup>

Quick Tip: In your Animation Layout, use two bottom panels, the bottom one *NLA Editor*, and the top one will switch between *Dope Sheet Editor/Action Editor* and *Graph Editor* with the shortcuts **Shift + F12** and **Shift + F6**

<sup>a</sup>Remember to rename the tracks in a "serious" project

**Important Note: Unassigned Actions are LOST when you quit blender if they are not used by anybody, therefore you need to protect them with a fake user.**

As a remainder on how to add a fake user, you can do it in different ways, one of them is by clicking the "Shield" icon from the *Outliner Editor/Blender File Display* mode, under Actions.

For more detail, eg how to *Repeat* (Under NLA Editor/Strip/Action Clip/Repeat) a Strip, check out the Reference video.

## 5.4 Armatures

- Video Workshop: <https://www.youtube.com/watch?v=iZBLtooU2Cs> (This is a follow along type of tutorial. from beginning to 58:51 is about armature creation, then follows skinning, then in 1:22:01 armature keyframed animation)
- Blender Docs: <https://docs.blender.org/manual/en/4.3/animation/armatures/introduction.html>
- Reference Rig: <https://blendswap.com/blend/25094>
- Note: Use Quaternion as rotation mode

An *Armature* is a hierarchical structure of *Bones* (an Acyclic Directed Graph, to be precise).

To create a bone in a fresh blender scene, use Add/Armature. Observe in the *Outliner Editor, Blender File Display* mode, what a bone created for us.

1. Expand the *Armatures* tab and *Objects* tab.
2. As you can see, we created an *Armature*, and an Object containing that armature.
3. The armature has inside it a composite structure
  - *Pose* The object itself has an additional data property, which is the Pose (in fact, there is a *Armature* tab in the *Properties Editor*)
  - *Bone Hierarchy* and *Bone Collection* inside the armature
  - Go to edit mode and extrude the bone. Now the bone hierarchy has an additional bone

Once you select an armature object, you can see from the viewport that it has only 3 modes: Object Mode, Edit Mode, and *Pose Mode*

- Edit Mode: Create More bones
- Pose Mode: The Armature is done, and you want to create some key poses for your armature

To have X-Ray on the bones, click, inside the Data tab in the *Properties Editor* the option *Viewport Display/In Front*.

To Rename your selected bone, use **F2**.

If the armature is supposed to be symmetric

- Create one side (eg right arm, right leg) and the middle too
- Click **F3** and type "Armature Symmetrize" (See figure ??)
- **Symmetrization works only if you follow the blender naming conventions with blender.** Renaming bones therefore becomes mandatory

There are different variants, basically you have to use the format "*Thing\_R*" for right and "*Thing\_L*" for left, and no suffix for bones in the middle.

Link: <https://docs.blender.org/manual/en/4.3/animation/armatures/bones/editing/naming.html#armature-editing-naming-conventions>

To visualize the hierarchy of bones without using the Outliner, you can use the *Properties Editor*, in the *Bones Data* tab, under "Relations", in which you can see the direct parent of the selected bone.

To create a new bone, you can Extrude it with **E** in Edit mode, but you can also use the *Subdivide* command from the edit mode context menu (with a bone selected).

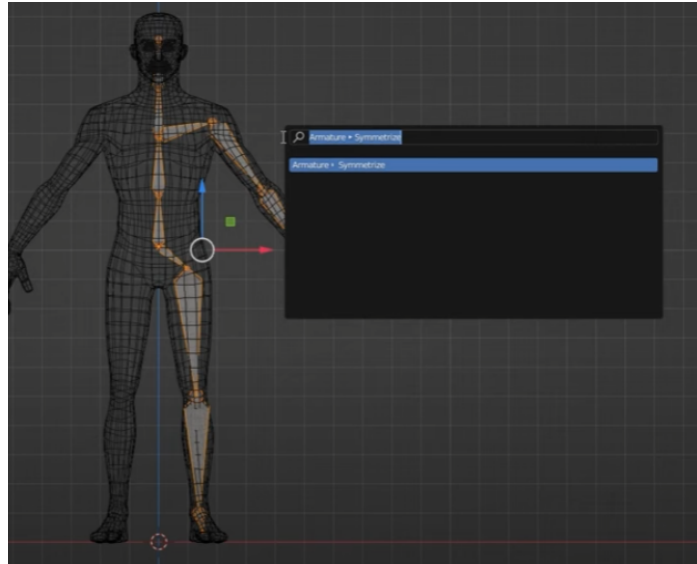


Figure 4: Symmetrizing an Armature

## Forward Kinematic vs Inverse Kinematic

If you create a bone hierarchy, go to *Pose Mode*, and then try to move one of its bones (example: a hand of a humanoid character), you'll notice that you can only rotate the selected bone subtree. By default, by moving a bone, you **move only its subtree**, ie a **child cannot move a parent**.

What if we want to move the hand of our humanoid and we want the elbow to follow it? *Inverse Kinematic*

<sup>10</sup> To create an *IK Rig* (Workshop video, 26:16), we need

- **create *IK Target***, which is a duplicate bone (**Shift + D**) of the bone that should trigger inverse kinematic (eg hand, foot)<sup>11</sup>
- with the duplicate bone selected, go to *Properties Editor*, Bones Data Tab, "Relations" and **remove the parent**
- Press **N** (Edit Mode) and adjust the Length property of this bone, such that it is easier to select
- Before adding any IK constraints, once your basic bones are already placed, **reset all bones roll**: Armature/Bone Roll/Recalculate (from a character-frontal orthographic view)
- Go back to *Pose Mode*, select the bones you need to affect and go to the *Bone Constraint* tab, add the constraint *Inverse Kinematic*. Under its options
  - Select as target: the armature itself
  - select as bone: IK Bone<sup>12</sup>
  - Select the bone with the constraint and change the *Chain Length* property (0 = infinite, 1 = affect only 1 adjacent bone)<sup>13</sup>
  - select the original bone, and add a bone constraint *Copy Rotation*, and set the Target Property: armature, and Bone: IK Bone<sup>14</sup>
  - with the IK Bone selected, go to the bone data tab and check off the tab *Deform*.<sup>15</sup>

After you've done that, say you want to hide the original hand bone (since you'll be manipulating the IK one), you can do that by moving the hand bone into a different *Bone Layer*. Bone layer options can be found under the *Armature* data tab in *Properties Editor*, "Skeletor/Layers", or you can select the bone and press **M**. Then, you can turn on/off bone layers as you need.

## IK Pull Target

(Video Workshop: 35:16)

Say you want to realize an elbow for a humanoid character. When animating it, you don't want to manipulate the forearm directly, you want to control a handle which should be the direction into which the elbow tip should point to, ie, a *Pull Target*.<sup>16</sup>

- With the elbow **Joint**<sup>17</sup> selected, extrude a new bone strictly backward.<sup>18</sup> (call that ElbowIK\_L)
- Remove the parent of the newly selected bone (Relations tab under the bone data tab in the *Properties Editor*)

<sup>10</sup>Side: To clear the pose in *Pose Mode*, use "Pose/Clear Transform/All"

<sup>11</sup>Naming convention: OriginalBoneName + IK + \_L/\_R

<sup>12</sup>At this point, IK works, but the IK target moves the whole armature

<sup>13</sup>At this point, the original bone doesn't follow the orientation of the IK bone

<sup>14</sup>The IK Bone can still move freely, don't make it go too far

<sup>15</sup>we don't want the IK bones to pull vertices of the mesh once skinned

<sup>16</sup>process used for knee, elbow, ...

<sup>17</sup>joint = little sphere, bone = connects two joints

<sup>18</sup>to a position in which the elbow cannot reach

- Select the forearm bone (the one whose base joint contains is elbow), go in *Pose Mode*, which should already have an IK constraint (if you configured the hand), select as *Pull Target* the armature itself, and as *Bone* for the pull the elbow IK bone.
- Change the *Pole Angle* in the IK Constraint such that it matches the desired direction (it's a trial and error process, so move around the forearm)

## Controller Bone

A Bone to move the whole character:

- move the 3D Cursor where the bone controller should be (eg. world origin)
- Create a new armature object. The newly created bone will be the new parent of the other armature object
- Turn off deform for this controller bone (Bone Data tab inside the *Properties Editor*)
- therefore select first then the original armature the controller and press **Ctrl + P** (Control Parent) and click *Keep Offset*<sup>19</sup>
- select all IK Bones (ie bones which are not in the hierarchy), then select the controller bone and **Ctrl + P**

## Non Limb Bones

We are referring to bones which should animate part of a model which is separated (on a different mesh object) (example: Hen Welding Helmet). While you can modify its pivot and keyframe its rotation, having a bone with some predefined poses makes the whole process much easier.

These bones, **even if they affect a different mesh the armature is a single hierarchy**, so use **Ctrl + P** after placing your bone. Furthermore, all of these bones should have *Deform* (from the bone data tab in the *Properties Editor*) turned off, such that they do not influence the main body<sup>20</sup>. That's because the non-limb parts just need to rotate following the bones, not deform.

When skinning with non limb bones, use parenting with *Bone*, and not *Armature Deform*, **With the correct bone selected**

## Bones Viewport display

Changes the way bones are visualized (some can be manipulated in edit mode by scaling bones (remember to reset transform)). Viewport Display is found in the *Armature Data Tab* in the *Properties Editor*

## 5.5 Skinning and Topology

Here are some links:

- Topology Guide: <https://topologyguides.com/>
- Retopology in blender: <https://www.youtube.com/watch?v=1myOZaxtHes>
- Topology for animation video: <https://www.youtube.com/watch?v=qfSRiE-6FtA>

### 5.5.1 Bones

As we've seen, bones can be classified into 2 basic categories, *Deforming Bones* and *Control Bones*

- *Deforming Bones*: Bones which, when transformed, will transform the vertices they are associated with (See Section 5.5.3)
- *Control Bones*: They control other bones and have no direct effect on vertices. Examples of this are *IK Target Bones* and *Main Controller Bone*

## Bone Collections

<sup>21</sup> Similarly to how you can group objects into Collections in the *Outliner Editor*, you can group bones into *Bones Collections*. **This won't affect the bones hierarchy**. Bone Collections can be created and managed from the *Armature* and *Bone* Property Panels in the *Property Editor*.

<sup>22</sup> There are 2 primary usages for Bone Collections:

- **Organization and visibility**: You can turn off each layer group individually
- **Library Overrides**<sup>23</sup>: When you try to *Link* an armature<sup>24</sup>, you can modify the imported bone collections by
  1. Link The Armature, which should be inside a collection. *File/Link*.
  2. Select the linked armature and go to *Object/Relations/Make Library Override* (**Ctrl + Shift + O**)
  3. In *Pose Mode*, open the *Bone Collections* panel in the Armature Properties tab
  4. Click the *Override* Button (Outliner, Right click on the component with the chain icon, select *ID Data/Make Library Override*)
  5. Change the link mode for the bone collections from data to object (see video footnote, minute 5:00 - 5:15)

<sup>19</sup>now the IK Bones are not following the controller

<sup>20</sup>How to skin and influence different parts later

<sup>21</sup>Added From Blender 4.0

<sup>22</sup>Note: Bone Collections are a blender 4.0 replacement for armature layers and bone groups

<sup>24</sup>Meaning The Armature Itself + Armature Object

6. (Maybe?) **Ctrl + L** and check the appropriate link option

Bone Collections Example:

- FK Controls
- IK Controls
- Face Controls
- Face Detail Controls

## Bone Selection Shortcuts

Reference: <https://docs.blender.org/manual/en/4.3/animation/armatures/bones/selecting.html>

Shortcut	Brief
[, ]	Select Parent, Select Child
Shift + [, Shift + ]	Extend selection with Parent, Extend Selection with Child

## Bone Editing

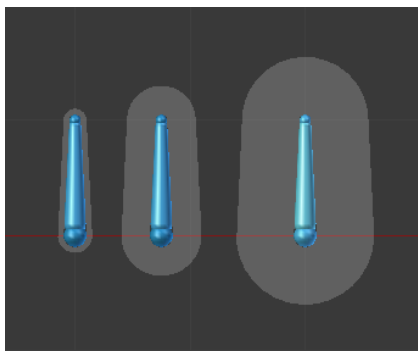
Very similar to Mesh Editing. We've already seen how *Extrude*, *Subdivide*, *Grab*, *Symmetrize*, *Parenting*, *Naming* work in the first paragraph of this subsection. you can find additional commands, such as *Split (P)*, in the documentation <https://docs.blender.org/manual/en/4.3/animation/armatures/bones/editing/index.html>.

An important editing command, which should be performed before skinning, is *Bone Roll Recalculation (Armature/Bone Roll/I)* or **Shift + N**)

## Bone Properties

25

- **Bendy Bones** Bones which can be curved, as they are represented as a Bezier Curve. Perfect for modeling chains of small ligaments/bones such as the Human spine column or facial bones. In Edit Mode, we can control the Bendy Bone as a Bezier Curve, and its properties editor, under the bendi bone panel, has your typical Bezier Curve Controls, such as number of *Segments*, Reference: [https://docs.blender.org/manual/en/4.3/animation/armatures/bones/properties/bendy\\_bones.html](https://docs.blender.org/manual/en/4.3/animation/armatures/bones/properties/bendy_bones.html)
- **Inverse Kinematics** Controls how the bone behaves when linked in a **Inverse Kinematic Chain**<sup>26</sup>. More details on this in the Posing SubSection, Bone Constraints Paragraph
- **Deform** Checks whether the bone should deform the parented mesh<sup>27</sup> or not. The bone defines a volume of space inside which vertices are affected. Such volume is called *Envelope*. We can control its Size with *Envelope Distance*, Head and Tail Radius. **Envelopes are considered only if you skin with Deform/With Envelope Weights**



Furthermore, for parts of the geometry shared by more than 1 bone, we can use the *Envelope Weight* option on these neighboring bones to given an importance to each.

### 5.5.2 Armature Properties

Basic ones:

- Pose tab to switch between the rest position of the armature and the current pose position
- Motion Paths (See Motion Path Animation Subsection)
- Inverse Kinematic: chooses the IK Solver (See Bone Constraints paragrah, Inverse kinematic)
- Bones Collection (Explained above)

<sup>24</sup>Link on Library overrides: <https://www.youtube.com/watch?v=nujaW-qNoRk>

<sup>25</sup>Excluding Transform, Viewport Display, Relations, which have been already covered

<sup>26</sup>Signaled with a yellow line

<sup>27</sup>Through the Armature Deform Tool with Automatic Weights

### 5.5.3 Skinning

Select the mesh, select the armature<sup>28</sup>, **Ctrl + P** -> Choose Armature Deform/With Automatic Weights. The mesh not has a *Armature* Modifier<sup>29</sup>

The mesh, skinned with automatic weights, never works properly. We need to use the *Weight Paint Mode* to manually adjust the weights of each bone.

To check the vertex weights for a given vertex

1. Select a skinned mesh and go to *Edit Mode*, select a vertex
2. press **N** to bring up the sidebar, select the *Item* tab and expand the *Vertex Weights* submenu. This will contain a list of bones which have a nonzero weight

In fact, once you skinned with automatic weights, blender created for you a *Vertex Group* for each bone (with the same name). Each *Vertex Group* contains the vertices that are affected by a given bone.

- with the skinned mesh selected, go to the Data panel of the *Properties Editor*, *Vertex Group* and select a vertex group for a bone. Pressing **A** with a vertex group selected will select all vertices in the vertex group

From the above considerations, we understand that blender has Two ways of Skinning

- **Parent/Constrain Objects to Bones**: When you transform the bones in *Pose Mode*, the "Child" mesh gets transform but **never** deformed
- **Armature Modifier on a mesh**<sup>30</sup>: Deform the geometry of the objects based on per bone weight association to some vertex groups

### Weight Paint Overview

Video Workshop: 1:14:25

Clicking **Ctrl + RMB**<sup>31</sup> lets you select the bone you want to edit the weights of.

By clicking into the brush icon (under the "Weight Paint" text) you can select the type of brush you want to use. By type we mean the math operation done to the weights. (eg. add, subtract).

In weight painting, having a **mirror modifier** saves you a lot of headache! Blender can recognize .R and .L bones with the mirror modifier!

In short, follow this simple 6 step guide: [https://docs.blender.org/manual/en/4.3/sculpt\\_paint/weight\\_paint/usage.html](https://docs.blender.org/manual/en/4.3/sculpt_paint/weight_paint/usage.html)

### 5.5.4 Posing

- Docs Reference: <https://docs.blender.org/manual/en/4.3/animation/armatures/posing/introduction.html>
- Video Reference:
- Pose Library: <https://www.youtube.com/watch?v=95rcqlpsMO4>
- Pose to Pose Workflow<sup>32</sup>: <https://www.youtube.com/watch?v=p8Bi7k60IS0>

Once an Armature is skinned to some objects and ready to go, we need to configure particular armature (or part of it) positions into *Poses* (Example: clenched fist Pose, which edits only the finger bones).

The main course here is to use *Pose Mode*. Its selection commands are the same of Edit Mode, plus *Select/Constraint Target* which selects the bones used as target by the currently selected bones.

If, during editing of a particularly heavy Skinned object, you experience some lag, you can enable, in the *Armature Panel*, the *Delay Deform* button, which will trigger the deformation only when you finish transforming a bone.

For a thorough overview of *Pose Mode*, let us exclude commands analogous to *Edit Mode* like *Apply* or *Clear Transform*.

- *Copy Paste* in Pose Mode, will actually copy the pose of the selected bones. Also, if you want to flip the pose along the X axis, you can use *Paste Flipped* (**Ctrl + Shift + V**)
- *Propagate* (**Alt + P**) the current pose to another Keyframe/keyframe range (specified in options like "To Next Keyframe")
- *In-Between Tools*: Tools to control interpolation between poses. All the following commands are in the *Pose/In-Betweens/* menu, and some have shortcuts
  - **Push Pose From Breakdown** (**Ctrl + E**) interpolates the current pose by making it closer to the next keyframed position. Video Reference: [https://www.youtube.com/watch?v=R0o\\_yHwSafE](https://www.youtube.com/watch?v=R0o_yHwSafE) (first 3 minutes). Similar is *Pose Breakdowner Tool*, which does the same thing, but it is interactive and therefore gives more control. For example, we can choose (like the Grab tool) which "axes" to affect
    - \* **G, R, S**: move, rotate, scale
    - \* **X, Y, Z**: to the corresponding axes
    - \* **B**: Bendy bones
  - **Relax Pose To Breakdown**<sup>33</sup> while the previous tools brings the pose closer to the next keyframe, this one pulls the pose closer to the previous keyframed position

<sup>28</sup>child then parent

<sup>29</sup>Be careful of where in the modifier list the armature modifier is. Example: It shouldn't be on below a subdivision modifier, such that it controls a simpler mesh. It shouldn't be above a mirror modifier, otherwise only the original mesh, and not the mirrored one, will be deformed.

**Modifiers are applied top to bottom**

<sup>30</sup>What the Armature Deform/With Automatic weights does

<sup>31</sup>I use right mouse button selection, otherwise left mouse button

<sup>32</sup>Watch This when you already know the basics

<sup>33</sup>There are also push/relax pose with respect to the rest pose



## Pose Library

To bring together Armatures with a Skinned Mesh, Actions, and the NLA Editor, the *Pose Library* comes into play. It is a simple .blend file which will store a catalogue of poses for a given object, each stored as a so-called *Pose Asset*<sup>34</sup>.

## Bone Constraints: Inverse Kinematics

Inverse Kinematics is a bone-based posing technique which allows **automatic positioning of a Chain of bones** by only manipulating the **Last bone in the chain**. (see example in introduction to armatures).

Such Constraints are *IK Solver* and *Spline IK*, which will be discussed when talking about Tracking in Subsection 5.6. In here, instead, we discuss the bone panel properties about *Inverse Kinematic*, for the standard solver

- *IK Stretch* Stretch influence to IK target.
- *Lock* Disallow movement around the axis.
- *Stiffness* Stiffness around the axis. Influence disabled if using Lock.
- *Limit* Limit movement around the axis.

While *IK Solver* tries to align a chain to point towards a Target and flex with respect to a Pull Target, the *Spline IK* is a constraint which aligns a chain of bones along a curve. Useful to model Tentacles, Tails, Ropes, ...

To setup Spline IK, it is necessary to have a chain of connected bones and a curve to constrain these bones to:

1. With the last bone in the chain selected, add a Spline IK Constraint from the Bone Constraints tab in the Properties.
2. Set the Chain Length setting to the number of bones in the chain (starting from and including the selected bone) that should be influenced by the curve.
3. Finally, set the Target field to the curve that should control the curve.

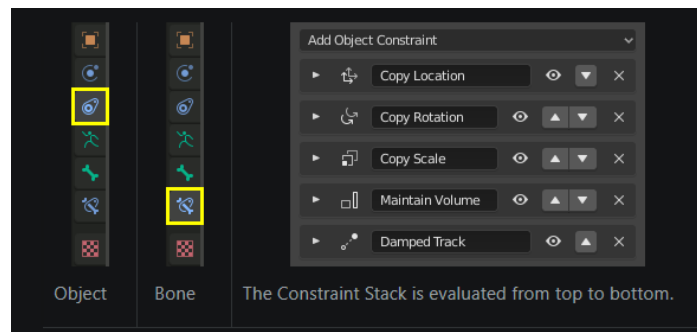
Tips:

- not too long bones and with same length
- Use few control points

Complete controls in [https://docs.blender.org/manual/en/4.3/animation/armatures/posing/bone\\_constraints/inverse\\_kinematics/spline\\_ik.html](https://docs.blender.org/manual/en/4.3/animation/armatures/posing/bone_constraints/inverse_kinematics/spline_ik.html)

Video reference in <https://www.youtube.com/watch?v=vZaNZhAoMts>

## 5.6 Constraints



## 5.7 Lattice

3D Grid created with the *Lattice Modifier*. It is basically a localized Proportional Editing used to deform a mesh, mainly to create *Shape Keys* (See subsection 5.9)

## 5.8 Drivers

1. Intro: <https://www.youtube.com/watch?v=rdHlRk6qvXM>

A driver is a relationship tool which controls the properties of some objects with properties of some other objects, as specified by a function<sup>35</sup> or mathematical expression.

The languages the driver mathematical expressions use is a subset of the Python Programming language. Quick Reference: [https://docs.blender.org/manual/en/4.3/animation/drivers/drivers\\_panel.html#drivers-simple-expressions](https://docs.blender.org/manual/en/4.3/animation/drivers/drivers_panel.html#drivers-simple-expressions)

<sup>34</sup>For more information on the *Asset Browser*, look at Section 4

<sup>35</sup>Namely, an F-Curve



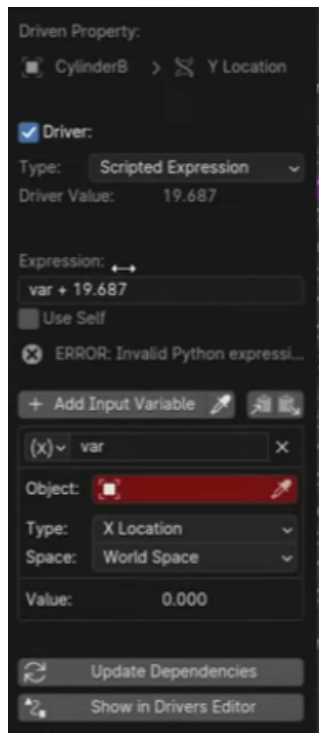


Figure 5: Scriptable Expression editor after *Add A Driver*

## Introduction Workflow from video

To create a Driver from a property, right click on it and *Copy as a new Driver*. This will put it into the clipboard, therefore you can go into another property of another object and use *Paste Driver*. **The driven property becomes purple.** Of course, once a driver is created, it is visible from the *Outliner* in *Blender File Display Mode*.

As a metaphor, Blender Drivers are very similiar to Excel Sheet's computed cell values.

Another button is: Right click to a property and *Add A Driver*, which will open an Expression editor. Once you close that popup, you need to open the *Driver Editor* and select the object with driven (purple) properties to get it back.

## 5.9 Shape Keys

- Video Reference <https://www.youtube.com/watch?v=7WGtmADZCzU>
- Blender Reference: [https://docs.blender.org/manual/en/4.3/animation/shape\\_keys/introduction.html](https://docs.blender.org/manual/en/4.3/animation/shape_keys/introduction.html)

*Shape Keys*<sup>a</sup>, authored in the *Shape Keys Panel* in the (Mesh) Data Tab of the Properties. This panel contains the *Influence Value* for each possible Shape Key defined. Whenever an object "shows" a shape key, it shows it with a certain percentage or **Influence**. This is usually paired with *Vertex Group* Assignment to some shape keys, such that some shape keys do not modify parts of your model they are not supposed to touch.<sup>b</sup>

<sup>a</sup>Also *Morph Targets*, *Blend Shapes*

<sup>b</sup>You can set the vertex group for a selected shape key from the data tab

### A Shape Key cannot modify the topology of the mesh<sup>36</sup>

The first shape key to be created is always the Base, ie the model in its "resting pose"

To **Add a new Shape Key**, from the *Shape Key Tab*, click the + button. If you add a new shape key with the command *Special/New Shape from Mix*, the new shape key will be created starting from the current vertices configuration.<sup>37</sup>

A mesh defines a **Stack** of Shape Keys, each of which can be either **Relative** or **Absolute**

*	Absolute	Relative
Usage	Deform objects into different shape over time	Muscles, linb joints, facial animation
Result	Each shape defines an evaluation time, and the result will be the interpolation between two shapes in two adjacent evaluation times	Each shape defines a weight, and the result will be the mix of all the relative offsets to the reference key
Value <sup>38</sup>	<i>Evaluation Time</i> <sub>(below)</sub> at which the shape key will be active	The Weight of the blend between the Shape key and the <i>reference key</i> (see below)
Basis	Name of the first (topmost) key in the stack, represents the original position	Name of the first (topmost) key in the stack, represents the original position. It is the reference key and is not weightable

To switch Shape keys between Absolute and Relative, use the checkbox in the Shape Keys Menu in the Data Tab

<sup>36</sup>As a consequence, Shape keys should be added **after** the topology is finalized

<sup>37</sup>meaning the new shape key will start as a mix of the current shape keys, each with the current influence

<sup>38</sup>Number which appears in the Shape Key list in the Data tab

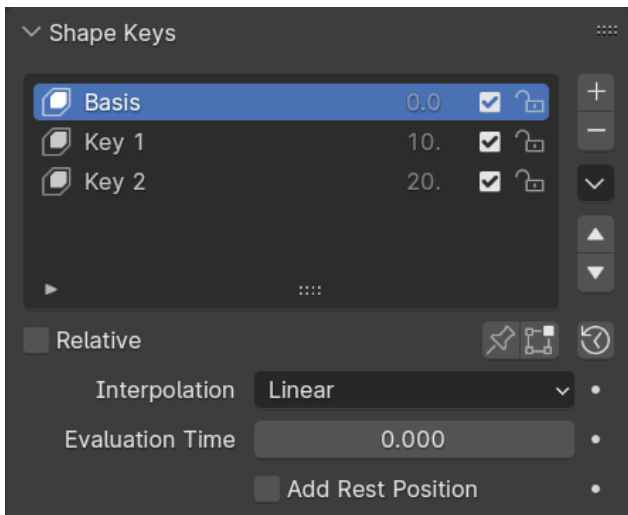


Figure 6: Absolute Shape Keys Panel

They each have their own settings. To the left, Relative Shape Keys settings, to the right, Absolute Shape Keys settings

- *Value*: The weight of the blend between the shape key and its reference key (usually the Basis shape). A value of 0.0 denotes 100% influence of the reference key and 1.0 of the shape key.
- *Range*: Minimum and maximum range for the influence value of the active shape key. Blender can extrapolate results when the Value goes lower than 0.0 or above 1.0.
- *Vertex Group*: Limit the active shape key deformation to a vertex group. Useful to break down a complex shape into components by assigning temporary vertex groups to the complex shape and copying the result into new simpler shapes.
- *Relative To*: Select the shape key to deform from. This is called the Reference Key for that shape.
- *Re-Time Shape Keys* (clock icon): Absolute shape keys are timed, by order in the list, at a constant interval. This button resets the timing for the keys. Useful if keys were removed or re-ordered.
- *Interpolation*: Controls the interpolation between shape keys. Linear, Cardinal, Catmull-Rom, B-Spline. The red line represents interpolated values between keys (black dots).
- *Evaluation Time*: Controls the shape key influence. Scrub to see the effect of the current configuration. Typically, this property is **keyed for animation or rigged with a driver**.

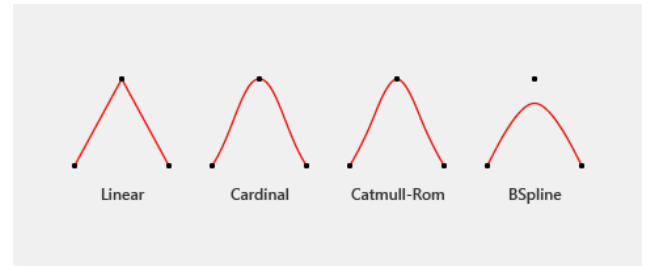


Figure 7: Interpolation Types for Absolute Shape Keys

## 5.10 Motion Paths

## 6 Animations Through Physics

<https://www.youtube.com/watch?v=69peFKU4XqI>

## 7 Compositing Essentials

## 8 Video Editing Essentials