Learn Explore Blog Help Center Forum DOWNLOAD

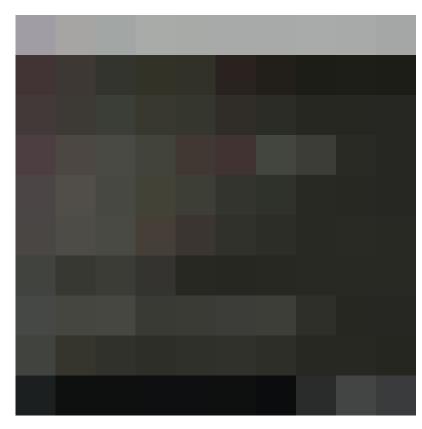
# Flip Work Plane Equivalent

Daniel\_Woodcock1

Nov '16

Hi @Dynamonkey,

If you want to get jiggie with python I think the FamilyInstance method flipFacing() might be what you want.



Although, i would use **@Mostafa\_El\_Ayoubi** 's solution. Easier.

Cheers, Dan

**Dynamonkey** 

Nov '16

Hey @Daniel\_Woodcock1 & @Mostafa\_El\_Ayoubi

I will take a look again but yes as mentioned in my post, I've tried those flip face nodes but unfortunately had no success. I have a feeling python is the way to go. The API example I was referring to calls for:

famInst.IsWorkPlaneFlipped = true

is there any equivalent for this in Python?

Daniel\_Woodcock1

Nov '16

Hi @Dynamonkey,

Instead of elem.flipFacing() Just use elem.lsWorkPlaneFlipped(b) where "b" is a bool value.

There are several flip methods (although the one you mention is a property).

## Family Instance Members (Flip)

		· —
•	flipFacing	The orientation of family instance facing will be flipped. If it can not be flipped, return false, otherwise return true.
	FlipFromToRoom	Flips the settings of "From Room" and "To Room" for the door or window instance.
	flinHand	The orientation of family instance hand will be flipped. If it can not be flipped, return false, otherwise return true.

# Family Instance Properties (Flip)

i i	CanFlipFacing	Property to test whether the orientation of family instance facing can be flipped.
3	CanFlipHand	Property to test whether the orientation of family instance hand can be flipped.
7	CanFlipWorkPlane	Identifies if the instance can its its work plane.
7	CanRotate	Property to test whether the family instance can be rotated by 180 degrees.
3	Category	Retrieves a Category object that represents the category or sub category in which the element reside (Inhented from Element.)
3	CreatedPhaseId	Id of a Phase at which the Element was created. (Inherited from <u>Element</u> .)
3	DemolishedPhaseId	Id of a Phase at which the Element was demolished. (Inherited from <u>Element</u> .)
3	DesignOption	Returns the design option to which the element belongs.  (Inherited from <u>Element</u> .)
3	Document	Returns the Document in which the Element resides.  (Inherited from <u>Element</u> .)
3	ExtensionUtility	Property to check whether the instance can be extended and return the interface for extension operat
<del>}</del>	EacingFlipped	Property to test whether the orientation of family instance facing is signed.
*	FacingOrientation	Property to get the orientation of family instance facing.
2	FromRoom	The "From Room" set for the door or window in the last phase of the project.
2	FromRoom(Phase)	
7	Geometry	Retrieves the geometric representation of the element. (Inherited from <u>Element</u> .)
3	GroupId	The id of the group to which an element belongs. (Inherited from <u>Element</u> .)
3	HandFlipped	Property to test whether the orientation of family instance hand is figured.
3	HandOrientation	Property to get the orientation of family instance hand.
2	<u>HasSpatialElementCalculationPoint</u>	Identifies if this instance has a single SpatialElementCalculationPoint used as the search point for Revi room or space.
3	HasSpatialElementFromToCalculationPoints	Identifies if this instance has a pair of SpatialElementCalculationPoints used as the search points for R between up to two rooms or spaces.
3	Host	If the instance is contained within another element, this property returns the containing element. An in the element containing the face.
7	Hostface	Property to get the reference to the host face of family instance.
3	HostParameter	If the instance is hosted by a wall, this property returns the parameter value of the insertion point of t curve, as long as the family of the instance isn't work plane based.
7	Īd	A unique identifier for an Element in an Autodesk Revit project. (Inherited from <u>Element</u> .)
7	Invisible	Property to test whether the family instance is invisible.
8	IsSlantedColumn	Indicates if the family instance is a slanted column.
7	<u>IsValidObiect</u>	Specifies whether the .NET object represents a valid Revit entity. (Inherited from <u>Element</u> .)
*	IsWorkPlaneFlipped	Identifies if the instance's work plane is flipped.

Dynamonkey Nov '16

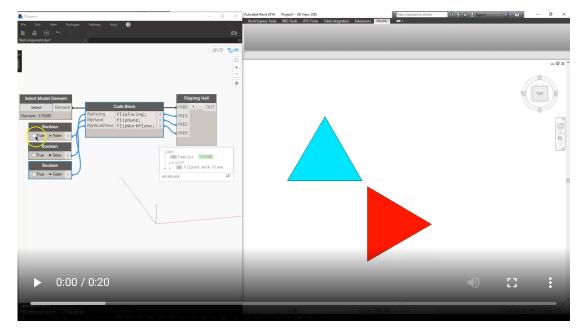
I think we are getting closer. I did a quick test literally adding onto the python code in the FlipFacing node by adding IN[1] to receive a Boolean toggle value and switching item.FlipFacing() to item.IsWorkPlaneFlipped(b) where b = IN[1]. Output is empty and family instance(s) remains unchanged.

Daniel\_Woodcock1 Nov '16

Hi @Dynamonkey,

Sorry, it was my fault, I made a mistake with how I told you to set the property (rookie mistake). Properties need to be set with an = so elem.IsWorkPlaneFlipped = b and not elem.IsWorkPlaneFlipped(b).

It seems to work fine here, see below (script attached also)...



Cheers, Dan

Dynamonkey Nov '16

Now that's what I call teamwork. Glad we got this one sorted out.

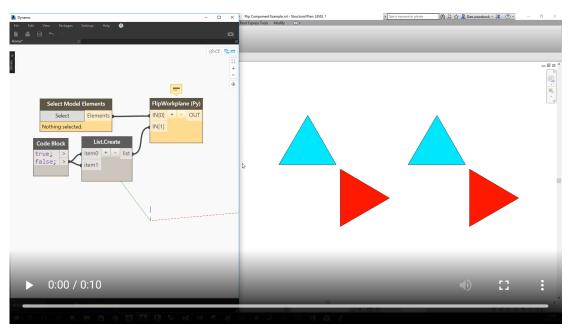
Viraj\_Voditel1 Jan '17

Is there any way we can apply the same principle to do it for multiple elements? I tried replacing the node with the one for multiple elements but it wouldn't work.

Daniel\_Woodcock1 Jan '17

Hi @Viraj\_Voditel1,

Yes, this is pretty simple. The following demostrates how to do this...



```
import clr
clr.AddReference("RevitServices")
import RevitServices
from RevitServices.Persistence import DocumentManager
from RevitServices.Transactions import TransactionManager
doc = DocumentManager.Instance.CurrentDBDocument
clr.AddReference("RevitNodes")
import Revit
clr.ImportExtensions(Revit.Elements)
clr.ImportExtensions(Revit.GeometryConversion)
clr.AddReference("RevitAPI")
from Autodesk.Revit.DB import *
def tolist(obj1):
    if hasattr(obj1,"__iter__"): return obj1
    else: return [obj1]
elems = tolist(UnwrapElement(IN[0]))
bools = tolist(IN[1])
```

```
TransactionManager.Instance.EnsureInTransaction(doc)
for elem,b in zip(elems, bools):
    elem.IsWorkPlaneFlipped = b
TransactionManager.Instance.TransactionTaskDone()
```

Hope this helps.

Cheers,

Dan

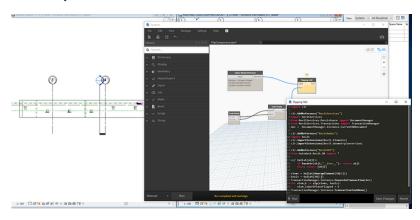
#### **𝚱** Stick the element on the wall

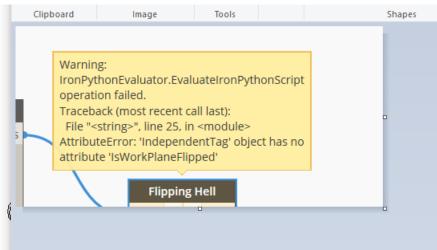
Jade Jun '18

Hi @Daniel\_Woodcock1

I've tried to run your script with multiple but it will only slip one still? I also have an error I don't know how to fix in the python script. Are you able to provide some help or spot the one thing I am missing?

## Thanks if you can!





Daniel\_Woodcock1 Jun '18

Hi @Jade,

Yes, two problems I can see.

- 1. The Boolean list needs to have the same number of bools as you do elements. It's a 1:1 operation as the elements/bools are zipped.
- 2. You are trying to flip a tag element and this doesn't support flipping (the yellow balloon warning message) since they are not hosted to planes as they are 2d elements. I don't think tags actually support flipping at all, only switching between orientation (horizontal/vertical) or changing their Family Type.

Hope this solves your problem.

Cheers, Dan

Jade Jun '18

Hi @Daniel\_Woodcock1

Thanks for responding. Where do I place 1. Do I have that in the code block and adjust the list or do I need to place the value in the python script?

2. is working fine, it flips tagged or untagged, it just only flips one instead of all. Which ^ should fix hopefully!

Kind regards

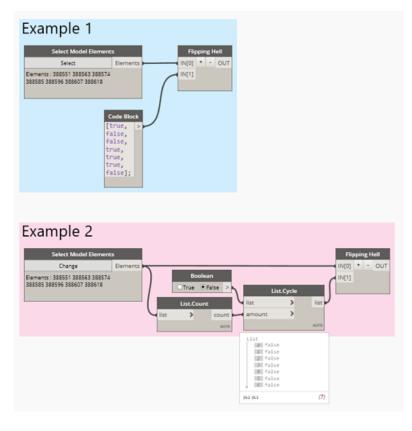
**ℰ** Change ref plane direction, or flip work plane of lighting fixture

Daniel\_Woodcock1 Jun '18

HI @Jade,

Yeah, you have 2 boolean values in the code block, but the number of these needs to match the number of elements you want to flip. So, in this instance, you need 7 as you have 7 Elements in the Select Model Elements node.

if you only need True or only need false, use the List.Cycle node. See examples below...



Jade Jun '18

Hi @Daniel\_Woodcock1 thank you for responding I am sure this will work now when I test run! Thank you again for being so helpful much appreciated!

n.dull	Jan '19			
Hi Daniel_Woodcock1,				
Where is that image you posted of the properties from?				
Thanks,				
ammar11khalil	May '19			

Thank you for your help <a>@Daniel\_Woodcock1</a> . Much Appreciated.