"""

checks all doors and sets door hinge side.

this is the pyRevit documentation help text.

"""

import clr

clr.AddReference("RevitAPI")

from Autodesk.Revit.DB import BuiltInCategory as Bic

from Autodesk.Revit.DB import FilteredElementCollector as Fec

from Autodesk.Revit.DB import Transaction

# DONE loop over all doors (loops)

# DONE ask each door for its type (access family type)

# DONE get preset default\_hinges\_side door side (how to read parameter)

# DONE ask if door is mirrored or not (how to read property)

# DONE perform some logic to find the actual hinges side (conditionals)

# DONE write result to door instance (how to write to parameter)

# \_\_window\_\_.set\_font\_sizes(19) # set bigger font in RPS

# reference the current open revit model to work with:

doc = \_\_revit\_\_.ActiveUIDocument.Document

# parameter names to work with:

# these are just the names not the actual parameters or their values

default\_hinges\_side = "Aufschlagrichtung\_Family"

din = "Aufschlagrichtung\_DIN"

# connect to Revit model elements via FilteredElementCollector

# collect all the doors (works the same way with other categories: e.g: walls: Bic.OST\_Walls)

doors = Fec(doc).OfCategory(Bic.OST\_Doors).WhereElementIsNotElementType().ToElements()

# entering a transaction to the modify revit model database

# Start transaction

tx = Transaction(doc, 'set door hinges side')

tx.Start()

# create main logic here..:

for door in doors:

print(15\*"-")

print(door.Id)

# ask each door for its type

door\_type = door.Symbol

# get preset default\_hinges\_side door side

hinges\_default\_param = door\_type.LookupParameter(default\_hinges\_side)

if hinges\_default\_param:

door\_type\_default\_side = hinges\_default\_param.AsString()

print("door default side: ", door\_type\_default\_side)

# ask if door is mirrored or not

is\_mirrored = door.Mirrored

# perform some logic to find the actual hinges side

if not is\_mirrored:

door.LookupParameter(din).Set(door\_type\_default\_side)

print("door is not mirrored.")

print("instance is:", door\_type\_default\_side)

elif is\_mirrored:

print("door is mirrored.")

if door\_type\_default\_side == "L":

door.LookupParameter(din).Set("R")

print("instance is: R")

elif door\_type\_default\_side == "R":

door.LookupParameter(din).Set("L")

print("instance is: L")

else:

print("parameter missing")

# commit the changes to the model database

# End transaction

tx.Commit()

print("successfully changed model")