*GlissandoHandler* Source Code.

import abjad

class GlissandoHandler:

def \_\_init\_\_(

self,

glissando\_style=None,

line\_style=None,

):

self.glissando\_style = glissando\_style

self.line\_style = line\_style

def \_\_call\_\_(self, selections):

return self.add\_glissando(selections)

def add\_glissando(self, selections):

runs = abjad.select(selections).runs()

if self.glissando\_style == 'hide\_middle\_note\_heads':

if self.line\_style != None:

for run in runs:

if len(run) > 1:

abjad.glissando(run[:], abjad.tweak(self.line\_style).style, hide\_middle\_note\_heads=True, )

else:

for run in runs:

if len(run) > 1:

abjad.glissando(run[:], hide\_middle\_note\_heads=True, )

elif self.glissando\_style == 'hide\_middle\_stems':

if self.line\_style != None:

for run in runs:

if len(run) > 1:

abjad.glissando(run[:], abjad.tweak(self.line\_style).style, hide\_middle\_note\_heads=True, hide\_middle\_stems=True, )

else:

for run in runs:

if len(run) > 1:

abjad.glissando(run[:], hide\_middle\_note\_heads=True, hide\_middle\_stems=True, )

else:

if self.line\_style != None:

for run in runs:

if len(run) > 1:

abjad.glissando(run[:], abjad.tweak(self.line\_style).style, allow\_repeats=True, allow\_ties=True, )

else:

for run in runs:

if len(run) > 1:

abjad.glissando(run[:], allow\_repeats=True, allow\_ties=True, )

return selections