*DynamicHandler* Source Code.

import abjad

class DynamicHandler:

def \_\_init\_\_(

self,

starting\_dynamic=None,

ending\_dynamic=None,

hairpin=None,

continuous=False,

):

def cyc(lst):

if self.continuous == False:

self.\_count = 0

while True:

yield lst[self.\_count % len(lst)]

self.\_count += 1

self.starting\_dynamic = starting\_dynamic

self.ending\_dynamic = ending\_dynamic

self.hairpin = hairpin

self.continuous = continuous

self.\_cyc\_dynamics = cyc([starting\_dynamic, ending\_dynamic])

self.\_count = 0

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

return self.add\_dynamics(selections)

def add\_dynamics(self, selections):

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

ties = abjad.select(selections).logical\_ties(pitched=True)

for run in runs:

if len(run) > 1:

leaves = abjad.select(run).leaves()

if self.starting\_dynamic != None:

abjad.attach(abjad.Dynamic(self.starting\_dynamic), leaves[0])

if self.hairpin != None:

abjad.attach(abjad.StartHairpin(self.hairpin), leaves[0])

if self.ending\_dynamic != None:

abjad.attach(abjad.Dynamic(self.ending\_dynamic), leaves[-1])

abjad.attach(abjad.StartHairpin('--'), leaves[-1]) #makes ending with a logical tie weird. If problematic: reduce indentation by 1

else:

leaves = abjad.select(run).leaves()

dynamic = next(self.\_cyc\_dynamics)

if self.starting\_dynamic != None:

if self.ending\_dynamic != None:

abjad.attach(abjad.Dynamic(dynamic), leaves[0])

else:

abjad.attach(abjad.Dynamic(self.starting\_dynamic), leaves[0])

if self.starting\_dynamic == None:

if self.ending\_dynamic != None:

abjad.attach(abjad.Dynamic(self.ending\_dynamic), leaves[0])

abjad.attach(abjad.StartHairpin('--'), leaves[0])

return selections