*ArticulationHandler* Source Code.

import abjad

class ArticulationHandler:

def \_\_init\_\_(

self,

articulation\_list=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.articulation\_list = articulation\_list

self.continuous = continuous

self.\_cyc\_articulations = cyc(articulation\_list)

self.\_count = 0

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

return self.add\_articulations(selections)

def add\_articulations(self, selections):

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

for tie in ties:

if len(tie) == 1:

if self.articulation\_list != None:

articulation = self.\_cyc\_articulations

abjad.attach(abjad.Articulation(next(articulation)), tie[0])

return selections