*NoteheadHandler* Source Code.

import abjad

class NoteheadHandler:

def \_\_init\_\_(

self,

notehead\_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.notehead\_list = notehead\_list

self.continuous = continuous

self.\_cyc\_noteheads = cyc(notehead\_list)

self.\_count = 0

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

return self.add\_noteheads(selections)

def add\_noteheads(self, selections):

if self.notehead\_list != None:

head = self.\_cyc\_noteheads

for tie in abjad.select(selections).logical\_ties(pitched=True):

head\_name = next(head)

string = str(r"""\once \override Staff.NoteHead.style = #'""")

full\_string = string + head\_name

style = abjad.LilyPondLiteral(full\_string, format\_slot='before',)

for leaf in abjad.select(tie).leaves(pitched=True):

abjad.attach(style, leaf)

return selections