def \_find\_no\_duplicates(self, name, domain=None, path=None):

toReturn = None

for cookie in iter(self):

if cookie.name == name:

if domain is None or cookie.domain == domain:

if path is None or cookie.path == path:

if toReturn is not None:

raise CookieConflictError('There are , %r' % (name))

toReturn = cookie.value

if toReturn:

return toReturn

raise KeyError(name, domain, path)

def take(seq, n):

for i in range(n):

yield next(seq)

if not hasattr(seq, 'next'):

seq = iter(seq)

while True:

x = list(take(seq, size))

if x:

yield x

else:

break

def calc\_median\_bbox\_pivots(positions):

median = None

bbox = [None, None]

n = 0

for pos in positions:

extend\_bbox(bbox, pos)

try:

median += pos

except:

median = pos.copy()

n += 1

median = median / n

bbox\_center = (Vector(bbox[0]) + Vector(bbox[1])) \* 0.5

return median, bbox\_center