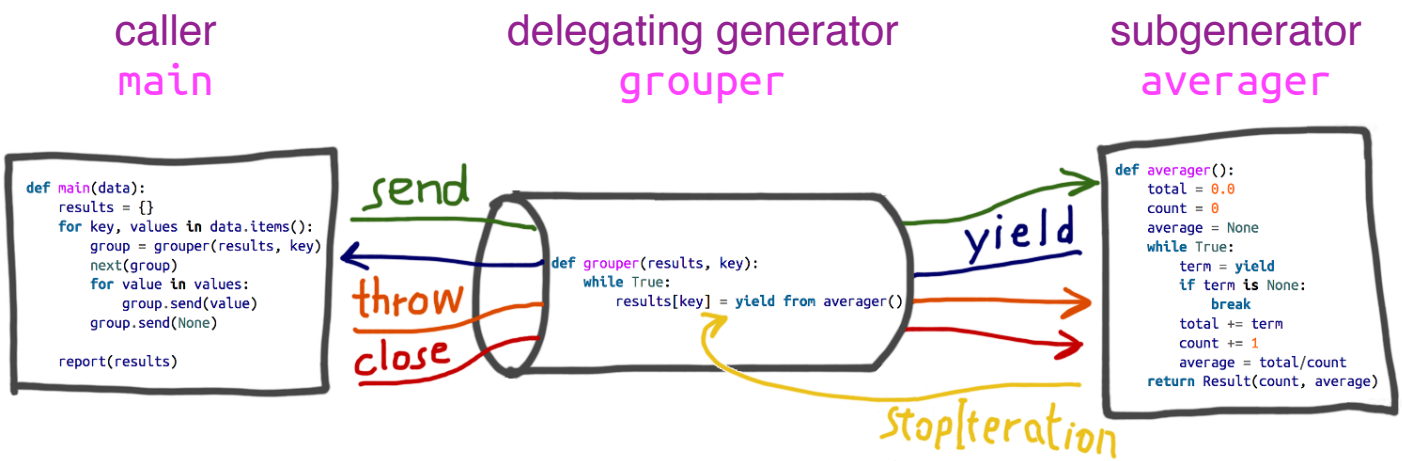
caller
main
delegating generator
grouper
subgenerator
averager
def main(data):
results = {}
for key, values in data.items():
group = grouper(results, key)
next(group)
for value in values:
group.send(value)
group.send(None)
report(results)
send
throw
close
def grouper(results, key):
while True:
results[key] = yield from averager()
yield
StopIteration
def averager():
total = 0.0
count = 0
average = None
while True:
term = yield
if term is None:
break
total += term
count += 1
average = total/count
return Result(count, average)