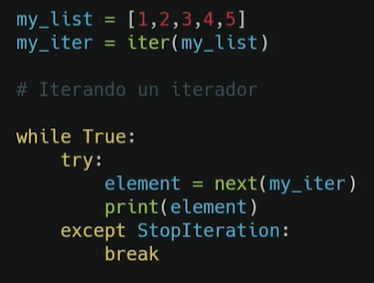
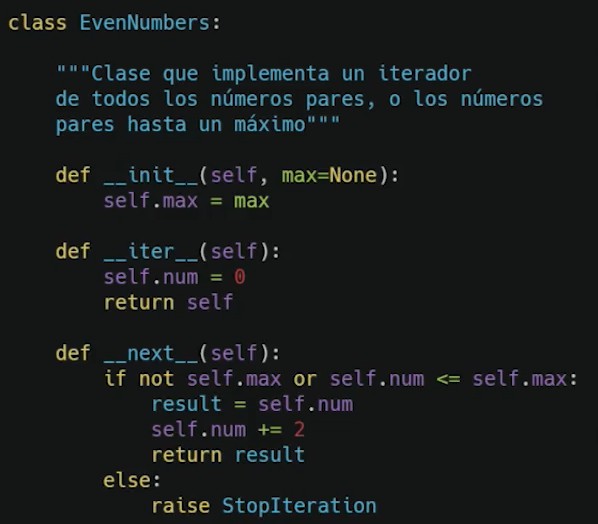
ITERACION

* POCA MEMORIA Y TRABAJAMOS MAS MEMORIA
* PUEDO GENERAR TODOS LOS NUMEROS PARES HASTA EL MAXIMO

for item in my\_list:

print(item)





FIBONACCI

0 1 1 2 3 5 8 13 21 34 55 89

import time

class FiboIter():

**def \_\_init\_\_(selft, max = None):**

if max == None:

selft.max = None

else:

selft.max = max

**def \_\_iter\_\_(selft):**

selft.n1 = 0

selft.n2 = 1

selft.counter = 0

return selft

**def \_\_next\_\_(selft):**

**def fibo():** #funcion fibonacci

selft.aux = selft.n1 + selft.n2

#selft.n1 = selft.n2

#selft.n2 = selft.aux

selft.n1, selft.n2 = selft.n2, selft.aux

selft.counter += 1

return selft.aux

if selft.counter == 0:

selft.counter += 1

return selft.n1

elif selft.counter == 1:

selft.counter += 1

return selft.n2

elif selft.max == None:

return fibo() #ejecucion infinita

elif selft.counter >= selft.max:

raise StopIteration

else:

return fibo()#ejecucion controlada

if \_\_name\_\_ == "\_\_main\_\_":

fibonacci = FiboIter()

i = 0

for element in fibonacci:

print(str(i) + " Vueltas")

print(element)

time.sleep(0.05)

i += 1