**1. How many seconds are in an hour? Use the interactive interpreter as a calculator and multiply the number of seconds in a minute (60) by the number of minutes in an hour (also 60).**

60\*60

**2. Assign the result from the previous task (seconds in an hour) to a variable called seconds\_per\_hour.**

seconds\_per\_hour = 60 \* 60

seconds\_per\_hour

**3. How many seconds do you think there are in a day? Make use of the variables seconds per hour and minutes per hour.**

one\_day = 24

second\_in\_a\_day = 24 \* seconds\_per\_hour

second\_in\_a\_day

**4. Calculate seconds per day again, but this time save the result in a variable called seconds\_per\_day**

seconds\_per\_day = 24 \* seconds\_per\_hour

seconds\_per\_day

**5. Divide seconds\_per\_day by seconds\_per\_hour. Use floating-point (/) division.**

seconds\_per\_day/seconds\_per\_hour

**6. Divide seconds\_per\_day by seconds\_per\_hour, using integer (//) division. Did this number agree with the floating-point value from the previous question, aside from the final .0?**

YES this number agrees

seconds\_per\_day // seconds\_per\_hour

**7. Write a generator, genPrimes, that returns the sequence of prime numbers on successive calls to its next() method: 2, 3, 5, 7, 11, ...**

def genPrimes():

n = 2

primes = []

while True:

for p in primes:

if n % p == 0:

break

else:

primes.append(n)

yield n

n += 1