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**

**Result - 3600**

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

**seconds\_per\_hour = 60\*60**

**res - 3600**

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

**Minutes\_per\_hour = 60**

**Day1 = 24**

**Day1\* seconds\_per\_hour**

**86400**

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

**Minutes\_per\_hour = 60**

**Day1 = 24**

**Seconds\_per\_day = Day1\* seconds\_per\_hour**

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

**Seconds\_per\_day / seconds\_per\_hour**

**24.0**

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**

**Seconds\_per\_day / seconds\_per\_hour**

**24**

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():**

**primes = [] # primes generated so far**

**last = 1 # last number tried**

**while True:**

**last += 1**

**for p in primes:**

**if last % p == 0:**

**break**

**else:**

**primes.append(last)**

**yield last**

**p = genPrimes()**