**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).**

**sol. 60**

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

**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.**

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

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

**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?**

**7. Write a generator, genPrimes, that returns the sequence of prime numbers on successive calls to**

**its next() method: 2, 3, 5, 7, 11, ...**

***SOLUTIONS***

1. *There are 60 seconds in a minute.*
2. *seconds\_per\_hour = 60*
3. *There are 24 hours in a day, so there would be 24 \* seconds\_per\_hour = 1440 seconds in a day.*
4. *seconds\_per\_day = 24 \* seconds\_per\_hour*
5. *Using floating-point division, seconds\_per\_day / seconds\_per\_hour = 24.0*
6. *Using integer division, seconds\_per\_day // seconds\_per\_hour = 24*
7. *Here is the code for the generator function genPrimes:*

***def genPrimes():***

***primes = []***

***last = 1***

***while True:***

***last += 1***

***for p in primes:***

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

***break***

***else:***

***primes.append(last)***

*yield last*