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. 3600

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

Sol.

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

Now, the variable seconds\_per\_hour contains the value 3600, which is the number of seconds in an 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.

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

**print(seconds\_per\_day)**

So there are 86,400 seconds in a day (assuming a 24-hour day).

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

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

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

**seconds\_per\_day / seconds\_per\_hour**  
Answer is 24

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, it should give the same value as the previous question, except it will be an integer without the decimal point. Here's the code to verify it:

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

This should output 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, ...

the genPrimes generator that returns the sequence of prime numbers on successive calls to its next() method:

**def genPrimes():**

**primes = [2]** # start with the first prime number

**yield 2**

**num = 3**

**while True:**

**for p in primes:**

**if num % p == 0:**

**break**

**else:**

**primes.append(num)**

**yield num**

**num += 2**

This generator works by maintaining a list of prime numbers found so far (starting with 2), and checking whether each successive odd number is divisible by any of the primes in the list. If not, that number is added to the list and yielded to the caller as the next prime number.