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

**Ans:** 60 \* 60 = 3600

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

**Ans:** seconds\_per\_hour= 60 \* 60

Print (seconds\_per\_hour)

1. How many seconds do you think there are in a day? Make use of the variable’s seconds per hour and minutes per hour.

**Ans: a.** Seconds per hour

seconds\_per\_hour = 60 \* 60

Print (seconds\_per\_hour)

**b.** Minutes per hour

minutes\_per\_hour = 60 \* 1

Print (minutes\_per\_hour)

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

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

Print (seconds\_per\_day)

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

**Ans:** x = seconds\_per\_day (86400) / seconds\_per\_hour (3600)

Print (x)

1. 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?

**Ans:** x = seconds\_per\_day (86400) // seconds\_per\_hour (3600)

Print (x)

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

**Ans:**

|  |
| --- |
| def genPrimes(): |
|  | primes = [2] |
|  | yield primes[0] |
|  | guess = 3 |
|  | while True: |
|  | if all(guess%x != 0 for x in primes): |
|  | primes.append(guess) |
|  | if guess == primes[-1]: |
|  | yield primes[-1] |
|  | guess += 2 |