

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

**Answer:** 3600

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

**Answer:** `sec_per_hour=60*60`

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

**Answer:** `sec_per_hour*24`

86400

4. Calculate seconds per day again, but this time save the result in a variable called `seconds_per_day`

**Answer:** `sec_per_day=sec_per_hour*24`

5. Divide `seconds_per_day` by `seconds_per_hour`. Use floating-point (`/`) division.

**Answer:** `sec_per_day/sec_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?

**Answer:** `sec_per_day//sec_per_hour`

24. Yes the result is the same, except data type is integer

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

**Answer:**

```
def genPrimes():
    primes=[]
    num = 2
    while True:
        for p in primes:
            if num % p == 0:
                primes.append(num)
            else:
                num += 1
                yield num

p = genPrimes()
```