

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

- `60*60`
- `>>> 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`

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

- `# No. of seconds per day...`
- `seconds_per_hour*24`
- `>>> 86400`
- `# There are 86,400 secs per day`

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

- `seconds_per_day = 24 * 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?

- `seconds_per_day//seconds_per_hour`
- `>>> 24`
- `seconds_per_day//seconds_per_hour == seconds_per_day/seconds_per_hour`
- `>>> True`
- `# Yes, both the numbers result in the same value with different datatypes.`

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():  
    n = 2  
    primes = []  
    while True:  
        is_prime = True  
        for p in primes:  
            if p*p > n:  
                break  
            if n % p == 0:  
                is_prime = False  
                break  
        if is_prime:  
            primes.append(n)  
            yield n  
        n += 1
```