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).

```
60*60
```

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.

```
one_day  = 24
second_in_a_day = 24 * seconds_per_hour
second_in_a_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
seconds_per_day
```

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

```
seconds_per_day/seconds_per_hour
```

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 this number agrees
seconds_per_day // seconds_per_hour
```

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:
        for p in primes:
            if n % p == 0:
                break
        else:
            primes.append(n)
            yield n
        n += 1
```