- 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).

In [1]:
```
60*60
```

Out[1]:
```
3600
```

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

In [2]:
```
seconds_per_hour = 60 * 60
seconds_per_hour
```

Out[2]:
```
3600
```

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

In [3]:
```
one_day = 24
second_in_a_day = 24 * seconds_per_hour
second_in_a_day
```

Out[3]:
```
86400
```

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

In [4]:
```
seconds_per_day = 24 * seconds_per_hour
seconds_per_day
```

Out[4]:
```
86400
```

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

In [6]:
```
seconds_per_day / seconds_per_hour
```

Out[6]:
```
24.0
```

- 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?

In [8]:
```
# YES it agrees
seconds_per_day // seconds_per_hour
```

Out[8]:
```
24
```

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

In [9]:

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

In [10]:

```
genPrimes()
```

Out[10]:

```
<generator object genPrimes at 0x0000017FB4A52350>
```

In [ ]:

```

```