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).

In [2]:

```
print(60*60)
```

```
3600
```

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

In [3]:

```
seconds_per_hour = 60*60
seconds_per_hour
```

Out[3]:

```
3600
```

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

In [8]:

```
minute_per_hour=60
second_per_hour=3600
hour_per_day=24
seconds_per_day=hour_per_day*second_per_hour
seconds_per_day
```

Out[8]:

```
86400
```

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

In [10]:

```
seconds_per_day=hour_per_day*second_per_hour
seconds_per_day
```

Out[10]:

```
86400
```

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

In [11]:

```
seconds_per_day/second_per_hour
```

Out[11]:

```
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?

In [21]:

```
seconds_per_day/second_per_hour
```

Out[21]:

```
24.0
```

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

In [22]:

```
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 [26]:

```
genPrimes()
```

Out[26]:

```
<generator object genPrimes at 0x000001CF8B0E7BA0>
```

In [ ]:

```

```