

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).

Ans:

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

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

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

Ans:

```
In [2]: seconds_per_hour = 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.?

Ans:

```
: seconds_per_hour*24
```

```
: 86400
```

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

Ans:

```
seconds_per_day = seconds_per_hour*24  
seconds_per_day
```

```
86400
```

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

Ans:

```
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?

Ans:

```
seconds_per_day // seconds_per_hour
```

24

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

Ans:

```
: def genPrimes():  
  
    primes = [ 2, 3, 5, 7, 11 ]  
  
    def isPrimeNumber(n):  
        if n in primes:  
            return True  
  
        for elem in primes:  
            if n % elem == 0:  
                return False  
  
        primes.append(n)  
        return True  
    num = 1  
    while True:  
        num += 1  
        if isPrimeNumber(num):  
            next = num  
            yield next  
            num = next  
    primeNumber = genPrimes()  
  
    for i in range(189):  
        print(primeNumber.__next__())
```

2
3
5
7
11
13
17
19
23
29