

Assignment_15

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).

There are 3600 seconds in an hour.

```
>>> 60 * 60
3600
```

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

Assigning the value of 3600 to the variable 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.

There is 86400 seconds in a day.

```
seconds_per_hour = 60 * 60
hours_per_day = 24
seconds_per_day = seconds_per_hour * hours_per_day
print(seconds_per_day)
```

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

```
seconds_per_hour = 60 * 60
hours_per_day = 24
seconds_per_day = seconds_per_hour * hours_per_day
print(seconds_per_day)
```

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

```
float_seconds_per_day = 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`?

```
integer_seconds_per_day = seconds_per_day //  
seconds_per_hour
```

we can see, the output of the code is the number of hours in a day, which is 24. The integer division agrees with the floating-point division, aside from the final `.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, ...

```
def genPrimes():  
    primes = []  
    current = 2  
    while True:  
        if all(current % prime != 0 for prime in primes):  
            primes.append(current)  
            yield current  
            current += 1  
  
for prime in genPrimes():  
    if prime <= 10:  
        print(prime)  
    else:  
        break
```