Assignment-15

1. How many seconds are in an hour? Use the interactive interpreter to calculate.

Ans. 60 * 60 # 60 seconds per minute * 60 minutes per hour

Result: 3600 seconds in an hour.

2. Assign the result from the previous task to a variable called `seconds_per_hour`:

Ans. 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. To find how many seconds there are in a day, we are instructed to use two variables: seconds_per_hour and minutes_per_hour.

Step-by-step analysis:

i) Seconds in an hour: You know there are 60 seconds in a minute and 60 minutes in an hour.So, the total number of seconds in one hour is:

seconds_per_hour = 60×60 = 3600 seconds

ii) Minutes in an hour: The question also mentions using minutes_per_hour, which is always 60 minutes per hour.

However, to calculate the total number of seconds in a day, you don't actually need to directly use minutes_per_hour, as you already have seconds_per_hour derived from it.

iii). Hours in a day: There are 24 hours in a day.

iv). Seconds in a day: Multiply the seconds_per_hour by the number of hours in a day:

seconds_per_day = seconds_per_hour × 24 = 3600 × 24 = 86400 seconds

The total number of seconds in a day is **86400**. The variable minutes_per_hour (60) indirectly contributes to this calculation because seconds_per_hour already includes it (60 seconds per minute × 60 minutes per hour = 3600 seconds per hour).

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

Ans.

```
seconds_per_hour = 60 * 60  # 3600 seconds in an hour
```

seconds_per_day = seconds_per_hour * 24 # 24 hours in a day

print(seconds_per_day)

Output: 86400

So, `seconds_per_day = 86400` seconds.

5. Divide `seconds_per_day` by `seconds_per_hour` using floating-point division.

Ans. seconds_per_day / seconds_per_hour # 86400 / 3600

Result: 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. To divide `seconds_per_day` by `seconds_per_hour` using integer division (`//`), here's the code:

seconds_per_hour = 60 * 60 # 3600 seconds in an hour

seconds_per_day = seconds_per_hour * 24 # 86400 seconds in a day

Integer division

result_integer_division = seconds_per_day // seconds_per_hour

print(result_integer_division) # Output: 24

The result of the integer division is 24.

Comparison with floating-point division:

In floating-point division (`/`), the result would be:

result_floating_point_division = seconds_per_day / seconds_per_hour

print(result_floating_point_division) # Output: 24.0

The result of the floating-point division is 24.0, which is essentially the same as the integer division result (24), except for the `.0`.

Yes, the result of integer division (24) agrees with the floating-point value (24.0), aside from the final `.0`. Both represent the same value.

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 = []  # List to hold the prime numbers
    num = 2  # Start from the first prime number
    while True:
        is_prime = True
        for p in primes:
            if num % p == 0:
                is_prime = False
                break
        if is_prime:
            primes.append(num)
            yield num
        num += 1


# Example of usage:
prime_generator = genPrimes()
print(next(prime_generator))  # 2
print(next(prime_generator))  # 3
print(next(prime_generator))  # 5
print(next(prime_generator))  # 7
```