1. What is the number of seconds in an hour? Multiply the amount of seconds in a minute (60) by the number of minutes in an hour using the interactive interpreter as a calculator (also 60).

Ans:- 3600 is the number of seconds in an hour.

>>> 60 \* 60

3600

2.Create a variable called seconds per hour with the result of the

previous task (seconds per hour).

Ans:- >>> seconds\_per\_hour = 60 \* 60

>>> seconds\_per\_hour

3600

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

Ans-: 86400 is the number of seconds in a day.

>>>seconds\_per\_hour = 60 \* 60

>>>minutes\_per\_hour = 60

>>>seconds\_per\_day = 60 \* 60 \* 24

>>>seconds\_per\_day

86400

1. Calculate seconds per day once more, but this time save the result in the seconds per day variable.

Ans-: >>>seconds\_per\_hour = 60 \* 60

>>>minutes\_per\_hour = 60

>>>seconds\_per\_day = 60 \* 60 \* 24

>>>seconds\_per\_day

86400

1. Subtract seconds per hour from seconds per day. Floating-point (/) division is the way to go.

Ans:- >>>seconds\_per\_hour = 60 \* 60

>>>minutes\_per\_hour = 60

>>>seconds\_per\_day = 60 \* 60 \* 24

>>>seconds\_per\_day

86400

>>>seconds\_per\_day - seconds\_per\_hour = 86400-3600

82800.0

>>>seconds\_per\_day /seconds\_per\_hour

>>> 24.0

6. Using integer (//) division, divide seconds per day by seconds per hour. Apart from the final, did this number agree with the floating-point value from the previous question ?

>>>seconds\_per\_day // seconds\_per\_hour

>>>seconds\_per\_hour \* 24 //seconds\_per\_hour

>>> 24

7. Create genPrimes, a generator that returns a sequence of prime numbers after each call to its next() method: 2, 3, 5, 7, 11,...

Ans:-

|  |
| --- |
| def genPrimes(): |
|  |

|  |
| --- |
| primes = [] |
|  |

|  |
| --- |
| n = 2 |
|  |

|  |
| --- |
| last = n |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| while True: |
|  |

|  |
| --- |
| for i in primes: |
|  |

|  |
| --- |
| if n % i == 0: |
|  |

|  |
| --- |
| n += 1 |
|  |

|  |
| --- |
| break |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| else: |
|  |

|  |
| --- |
| primes.append(n) |
|  |

|  |
| --- |
| last = n |
|  |

|  |
| --- |
| n += 1 |
|  |

|  |
| --- |
| yield last |