

Exercise 1 –

Description:

- Write a Python program that check if a string only contains numbers.
- If it does, print True. Else, print False.

Expected Output:

String	Output
"Hello"	False
"4567"	True
"Hello59"	True
""	False

Exercise 2 –

Description:

- Write a Python Program that prints the reversed version of a string.
- The program must preserve uppercase and lowercase letters.
- If the string is empty, print it intact.

Expected Output:

Input	Output
""Hello"	"olleH"
""Wo"	"oW"
""	""

Exercise 3 –

Description:

- Write a Python program that prints the corresponding season based on the value of the variable `season_num`.
- The possible values of `season_num` are: **1** for Spring, **2** for Summer, **3** for Fall, **4** for Winter.
- If the value of `season_num` is neither one of these values, print "Please enter a valid number".

Expected Output:

season_num	Output
1	"Spring"
2	"Summer"
3	"Fall"
4	"Winter"

Exercise 4 –

Description:

- Write a Python program that prints "Equal " if three numbers `a`, `b`, and `c` are equal.
- If at least one number is different, the program should print "Not Equal".

Expected Output:

a	b	c	Output
3	3	3	"Equal"
3	4	3	"Not Equal"
3	4	4	"Not Equal"
1	2	3	"Not Equal"

Exercise 5 –

Description:

- Write a Python program that prints the number of days in a given month.
- The value of the variable month is the name of the month **with the first letter capitalized**.
- Do **not** consider leap years for the number of days in February.
- You can add a customized message. For example: "<month> has: <num_days> days."

Expected Output:

Month	Output
January	31
February	28
March	31
April	30
May	31
June	30
July	31
August	31
September	30
October	31
November	30
December	31

Hints:

- Remember that February has 28 days.

Exercise 6 –

Description:

- Write a Python program that prints the first 100 even numbers (from 2 to 200 inclusive).

Expected Output:

```
2
4
6
8
10
12
14
16
18
20
22
24
26
28
30
32
34
36
38
40
42
44
46
48
50
52
54
56
58
60
62
64
66
68
70
72
74
76
78
80
82
84
86
88
90
92
94
96
98
100
102
104
106
108
110
112
114
```

116
118
120
122
124
126
128
130
132
134
136
138
140
142
144
146
148
150
152
154
156
158
160
162
164
166
168
170
172
174
176
178
180
182
184
186
188
190
192
194
196
198
200

◆ **Hints:**

- You can check if a number is even or odd with `num % 2 == 0`. If it's true, the number is even. Else, it's odd.
- The `range()` function can take a third parameter to customize the step (difference) between the integers in the sequence.

Exercise 6 –

Description:

- Write a Python program that prints if a given year was (or will) be a leap year.

Tip: A leap year is "a year, occurring once every four years, that has 366 days including February 29 as an intercalary day." (Definition by Oxford Languages).

This is how you can determine if a year is a leap year or not:

- **if** (*year* is not divisible by 4) **then** (it is a common year).
- **else if** (*year* is not divisible by 100) **then** (it is a leap year)
- **else if** (*year* is not divisible by 400) **then** (it is a common year)
- **else** (it is a leap year)

To learn more about leap years and how to determine if a year is a leap year or not, I recommend reading [this resource](#).

Expected Output:

Year	Output
2025	No
2033	No
1836	Yes
1912	Yes

You can add a custom message to the output. For example: "<year> is not a leap year"

Exercise 7 -

Description:

- Write a Python program that simulates the "Rock, Paper, Scissors" game.
- The game should ask the user to enter an option (either "Rock", "Paper", or "Scissors").
- The player should play against the computer, which will select a random option.
- The computer's selection will be compared against the player's selection to determine who wins.
- A descriptive message should be displayed indicating if the player won, lost, or if the game ended in a tie.

Basic Game Rules:

- Paper beats Rock
- Rock beats Scissors
- Scissors beat Paper.

Expected Output:

Sample Game 1:

```
===== Welcome to the game =====  
Please enter Rock, Paper, or Scissors below:  
Rock  
It's a tie! Try again.
```

Sample Game 2:

```
===== Welcome to the game =====  
Please enter Rock, Paper, or Scissors below:  
Paper  
You lose! Your opponent chose 'Scissors'
```

Sample Game 3:

```
===== Welcome to the game =====  
Please enter Rock, Paper, or Scissors below:  
Rock  
You win! Your opponent chose 'Scissors'
```

Hints:

- You will need to use nested conditionals (conditionals within conditionals) to implement this game.
- To generate a random choice for the computer player, you may use the random module and the randint function.

Exercise 8 -

Description:

- Write a Python program that removes duplicate elements from a list, only keeping one occurrence of each element in the list.
- The original list should be mutated (modified).
- The program must print the final version of the list.

Expected Output:

List	Output
[1, 1, 2, 3, 4, 4]	[1, 2, 3, 4]
["a", "a", "b", "a"]	["a", "b"]
[1, 2, 3]	[1, 2, 3]
[]	[]

Hints:

- **Sets** are commonly used to remove duplicates from lists and tuples in Python.

Exercise 9 -

Description:

- Write a Python program that converts seconds to minutes and hours.
- Present the minutes as an **integer** and the hours as a **decimal** value.

Expected Output:

Seconds	Minutes	Hours
5400	90	1.5
7200	120	2

You can customize the output like this:

5400 seconds is equivalent to:

90 Minutes

1.5 Hours

Hints:

- Remember that 1 hour has 60 minutes and 1 minute has 60 seconds.

Exercise 10 -

Description:

- Write a Python program that calculates body mass index.
- The formula to calculate body mass index is $BMI = \text{kg}/\text{m}^2$ where kg is a person's weight in **kilograms** and m² is their height in **meters squared**.
- The user should be able to enter his or her **height in centimeters** and **weight in kilograms**.
- You may assume that the height and weight entered will be positive integers.
- The program must print a message with the value of the Body Mass Index (BMI) rounded to two decimals and the category:

Underweight = less than 18.5

Normal = from 18.5 to 24.9

Overweight = from 25 to 29.9

Obesity = 30 or greater

Note: The classification of the result of the Body Mass Index is based on the ranges provided by [this source](#).

Expected Output:

Height	Weight	BMI	Output
150	52	23.1	"Normal"
175	98	32	"Obesity"
190	50	13.8	"Underweight"

Hints:

- The formula uses **meters squared** but the user input will be in **centimeters**. You will need to convert the input to the appropriate units.