9 Exercise Questions

1. Write a `for` loop to print the numbers from 1 to 5. Use the `range()` function.
2. Write a `while` loop that prints the numbers from 1 to 5.
3. Write a `for` loop to print each character in the string `"Hello DC"`.
4. Accept a number from the user using `input()` function. Then, use an `if` block to check if the number is positive or negative. Print `"Positive"` if the number is greater than zero; otherwise, print `"Not Positive"`.
5. Use a `for` loop to calculate the sum of all numbers from 1 to 10.
6. Generate 5 random integers between 1 and 10 using `random.randint()` inside a `for` loop. Import the `random` module first.
7. Write a `while` loop that keeps generating a random number between 0 and 1 until the number is greater than 0.8. Print each generated number.
8. Take a user's name using `input()` function. Use an `if . . . elif . . . else` block to check if a given string starts with `"A"`, `"B"`, or some other letter which in this case you should mention that the first letter of the name. Print a different message for each case.
9. Write a `for` loop to print the squares of numbers from 1 to 10. Use the `range()` function.
10. Write a program that takes a number as input and prints `"Even"` if it is even, and `"Odd"` if it is odd. Use the `%` (modulus) operator and an `if . . . else` block.
11. Write a `for` loop to print every third character of the string `"Python Programming is fun!"` starting from the first character.
12. Write a program that generates a random floating-point number between 0 and 10, rounds it to 2 decimal places, and prints the result. Use the `random` module and `round()` function.
13. Write a program that asks the user to enter a word, and then checks if the word contains the letter `"e"`. Print `"Contains 'e'"` if it does, otherwise print `"Does not contain 'e'"`. Use the `in` keyword.
14. Write a `while` loop that asks the user for a number and calculates the cumulative sum until the user enters `0`. Print the cumulative sum at the end.
15. Write a program that asks the user for a temperature in Celsius and converts it to Fahrenheit using the formula `F = C * 9/5 + 32`. Print the result using an `f-string`.
16. What is the difference between `=` and `==`?
17. Write a code that accepts 4 numbers from the user and prints *all equal* if they all are equal numbers and prints *not equal* if not all numbers are the same.
18. Import `random` module and generate 16 uniform random numbers between 0 and 1. You need to use `random.random()` function inside a `for` loop.
19. Import `random` module and generate 10,000 uniform random numbers between 0 and 1. You need to use `random.random()` function inside a `for` loop.
    (a) Find *sum* of these numbers.
    (b) Find *average* of these numbers. You need to find sum and divide it by 10,000.
    (c) Find *minimum* of these numbers? Did you get an exact 0 as the minimum?
    (d) Find *maximum* of these numbers? Did you get the exact 1 as the maximum?
20. Inside a `for` loop generate 5000 uniform random numbers between 0 and 1. Use an `if` block to print those numbers that are bigger than 0.5. How many numbers did you get?

21. There are several ways to generate standard normal random numbers. One method is to start with two uniform[0, 1] random numbers `U_1` and `U_2` and apply the following functions to get two standard normal random numbers:

$$Z_1 = \sqrt{-2\ln U_1}\cos(2\pi U_2)$$
$$Z_2 = \sqrt{-2\ln U_1}\sin(2\pi U_2)$$

    (a) Use `random` and `math` modules to generate two standard normal random numbers using the above formulas. You need to explore functions available to you in `math` module.
    (b) Using a `for` loop, generate 20 standard normal random numbers using the above method.
    (c) Using a `for` loop, generate 2000 standard normal random numbers using the above method and keep only the positive numbers. How many positive numbers did you get?
    (d) Inside a `while` loop generate enough random numbers to get the first number bigger than 3. How many numbers did you have to generate to get to the first number bigger than 3?
    (e) Inside a `while` loop generate enough random numbers to get the 5 numbers that either are less than -2.5 or more than 2.5. How many numbers did you have to generate to get this result?

22. We learned that `True` and `False` represent boolean values in Python. We know that computers store everything in memory using a **binary** format, which means only `0`s and `1`s. In this binary system, `True` is stored as `1`, and `False` is stored as `0`. Run this code to understand this behavior:

numerical values of `True` and `False`

```
print(f'True + True = {True + True}')
print(f'True + False = {True + False}')
print(f'False + False = {False + False}')
print(f'True * 5 = {True * 5}')
print(f'True > False = {True > False}')
```

    What do you observe? How do you think Python treats `True` and `False` when used in calculations, given that they are stored as `1` and `0` in memory?

23. It is very tempting to use `max`, `min`, `sum`, `type`, and `id` as variable names. This question is designed to show you why this is a terrible idea. Running the following code

trouble using built-in functions as variables

```
num_1 = 5
num_2 = 6
sum = num_1 + num_2
print(sum)
print(sum(1,3))
```

    will lead to `TypeError: 'int' object is not callable`. Why do you think that is the case?

24. We learned that we can compare numerical values using comparison operators such as `=`, `<`, `>`, `<=`, `>=`. This assignment will help you extend that to `str`. Run this code and try to understand how Python compares two `strings`

string comparisons

```
s_1 = "Adams"
s_2 = "James"
s_3 = "Zara"
print(f'{s_1} < {s_2} results in {s_1<s_2}')
print(f'{s_3} < {s_2} results in {s_3>s_2}')
```

How do you think Python compares two strings?

25. Using the `Decimal` data type, multiply `0.1` by `3` and display the exact result.
26. You are working on a financial application that requires precise interest rate calculations. Using the `Decimal` type, calculate the compounded value of an investment of $1,000 at an annual interest rate of 4.5%, compounded monthly, for 5 years. Use the formula:

$$A = P\left(1 + \frac{r}{n}\right)^{nt}$$

where:

- $P = 1000$ (the principal)
- $r = 0.045$ (annual interest rate)
- $n = 12$ (compounding periods per year)
- $t = 5$ (time in years)

Ensure precise calculation without any rounding errors.

10 Exercise Solutions

Solutions to these problems can be found on the following GitHub page:
https://tinyurl.com/3wekv7mt
You can also access the same link using the QR code below: