

## Chapter 4

## Random Functions

Many times, we require using random numbers in our application to produce simulations or games and other applications that require random events. For example, in a game of dice, without having random events, we will have the same side popping up every time we throw the dice. In the physical environment, we can have random events generated but it is not possible when it comes to computers.

We will learn how to generate random numbers using some functions such as `randint()`, `randrange()`, `uniform()` in the `random` module.

### 1. Generating Random Integer Numbers

To generate a random number, you can use the `randint(a, b)` function. This function returns a random integer `i` such that  $a \leq i \leq b$ .

**Example 01:** random integer between 0 and 9

```
import random

# Generate random numbers
number = random.randint(0, 9)

# Display the number
print(number)
```

#### Output

3

Suppose you want to develop a program to help a first grader practice addition. The program randomly generates two single-digit integers, `number1` and `number2`, and displays to the student a question such as **What is 1 + 7?** After the student types the answer, the program displays a message to indicate whether it is correct or incorrect.

**Example 02:** An Addition Quiz program for grade one students. The program prompts the user to enter an answer for a question on addition.

```
import random

# Generate random numbers
number1 = random.randint(0, 9)
number2 = random.randint(0, 9)

# Prompt the user to enter an answer
answer = eval(input("What is " + str(number1) + " + " + str(number2) + "? "))

# Display result
print(number1, "+", number2, "=", answer, "is", number1 + number2 == answer)
```

#### Output

What is 1 + 7? 8  Enter  
1 + 7 = 8 is True

What is 4 + 8? 9  Enter  
4 + 8 = 9 is False

## Practice

1. Write a program to help a user practice multiplication. The program randomly generates two integers between 10 to 20, **number1** and **number2**, and displays to the user a question such as **What is 15 \* 18?** After the user types the answer, the program displays a message to indicate whether it is "Correct" or "Incorrect".
2. Modify the practice 1 to allow the program to generate five questions. At the end, display how many correct answers the user gets out of five.

You can use `randrange(start, stop, step)` function to randomly select an integer *i* such that  $start \leq i < stop$ .

- start: An integer specifying at which position to start.
- stop: An integer specifying at which position to end.
- step: An integer specifying the incrementation. Default 1

**Example 03:** Generate a random number between the numbers between [0, 100[ in steps of five,

```
import random

number = random.randrange(0, 100, 5)

print(number)
```

## Output

30

## Practice

1. Write a program that randomly generate three positive numbers that are less than 100 and are divisible by 3.
2. Ask the user how many positive numbers that are less than 100 and are divisible by 3 that the user would like the program to generate. Randomly generate the numbers, store them in the list, and display the list.

## 2. Generating Random Floating-point Numbers

You can use `uniform(a, b)` to generate a random floating point number *f* such that  $a \leq f \leq b$  for  $a \leq b$ .

**Example 04:** Generate a random floating-point number between 1.5 to 2.4.

```
import random

number = random.uniform(1.5, 2.4)

print(number)
```

## Output:

2.2484429532193806

### 3. Randomly Selecting Items from a List

The random module also provide a function called `choice()` that randomly chooses a **single item** from a list, dictionary, tuple and set.

**Example 05:** Picks a random color from a list containing red, black and green.

```
import random

color = random.choice(["Red", "Black", "Green"])

print(color)
```

#### Output

Black

The `choice()` function only returns a single item from a list. If you want to randomly select **multiple items** from a list, you can use `sample()` or `choices()` instead.

**Example 06:** Randomly select four items from a list.

```
import random

list1 = [20, 30, 40, 50, 60, 70, 80]

sampling = random.choices(list1, k = 4)

print(sampling)
```

#### Output:

[60, 20, 30, 60]

Also, as you can see in the output, we got a few repeated numbers. `choices()` can repeat elements. Use the `sample()` function when you want to choose multiple random items from a list without repetition or duplicates.

**Example 07:** Randomly select four items from a list without duplicates.

```
import random

list1 = [20, 30, 40, 50, 60, 70, 80]

sampling = random.sample(list1, k = 4)

print(sampling)
```

#### Output:

[60, 20, 30, 80]

**Note:** The number of items to be selected from the list `k` must be an integer. In `choices()`, `k` can be greater than the length of the list since duplicates are allowed. In `sample()`, `k` must be less than or equal to the length of the list.

## Exercises

1. Randomly generate a number between 1 and 6. Ask the user to pick a number. If the user guess correctly, display the message “Well done”, otherwise, display “Incorrect”, and allow the user to enter a second guess. If the user guess correctly on the second guess, display “Correct”, otherwise display “You lose. The correct number is [random\_number]”.

Here are sample runs (for example, the random\_number is 3):

```
Pick a number between 1 and 6: 3 Enter
Well done
```

```
Pick a number between 1 and 6: 5 Enter
Incorrect
Enter your second guess: 1 Enter
You lose. The correct number is 3
```

2. (*Game: heads or tails*) Write a program that lets the user guess whether a flipped coin displays the head or the tail. The program randomly generates head or tail. The program prompts the user to enter a guess and reports whether the guess is correct or incorrect. Make it be case insensitive. Here are sample runs (for example, the random coin side is Head):

```
Enter your guess (Head or Tail): Head Enter
Correct
```

```
Enter your guess (Head or Tail): Tail Enter
Incorrect
```

3. (*Simulation: heads or tails*) Write a program that simulates flipping a coin one million times and displays the number of heads and tails. Here is a sample run:

```
The number of Head is: 504321
The number of Tail is: 495679
```

4. (*Game: scissor, rock, paper*) Write a program that plays the popular scissor-rock-paper game. (A scissor can cut a paper, a rock can knock a scissor, and a paper can wrap a rock.) The program randomly generates scissor, rock, or paper. The program asks the user to enter scissor, rock, or paper and displays a message indicating whether the user wins, loses, or draws. Here are sample runs:

```
scissor, rock or paper? rock Enter
The computer is scissor. You are rock. You won.
```

```
scissor, rock or paper? paper Enter
The computer is paper. You are paper too. It is a draw.
```

```
scissor, rock or paper? scissor Enter
The computer is rock. You are scissor. You lose.
```

5. Write a program to randomly generate ten numbers between 1 and 10, and store them in a list. Displays all the missing number(s) from the list. Here is a sample run:

```
Random list: 2 10 8 2 3 4 1 8 9 5
The missing number(s) is/are: 6 7
```

6. (*Repeat addition quiz*) Write a quiz program to randomly create a question and alert the user if an answer is needed to be entered again. Here is a sample run:

```
What is 5 + 9? 12 Enter
Wrong answer. Try again. What is 5 + 9? 34 Enter
Wrong answer. Try again. What is 5 + 9? 12 Enter
You already entered 12
Wrong answer. Try again. What is 5 + 9? 14 Enter
You got it!
```

7. (*Math Quiz*) Make another quiz program that asks five questions by randomly generating two integer numbers between 10 to 20 to make the question (What is [num1] + [num2]?). There must be four operations: +, -, \* and /. If the result is floating-point number, make it only two decimal places. Ask the user to enter the answer. If they get it right add a point to their score. At the end of the quiz, tell them how many they got correct out of five and the details. Here is a sample run:

```
Question 1: What is 15 + 10? 25 Enter
Question 2: What is 12 - 20? 5 Enter
Question 3: What is 14 / 19? 0.74 Enter
Question 4: What is 17 * 11? 108 Enter
Question 5: What is 20 / 15? 8 Enter

You get 2 correct answers out of 5. The detail is below:
Question 1: 15 + 10 = 25 Correct
Question 2: 12 - 20 = 5 Incorrect. The correct answer is - 8
Question 3: 14 / 19 = 0.74 Correct
Question 4: 17 * 11 = 108 Incorrect. The correct answer is 181
Question 5: 20 / 15 = 1.8 Incorrect. The correct answer is 1.33
```

8. (*Simulation picking a pearl*) A screen flashes in front of you and explains you the game. There are six pearls (three are white and three are black) and two empty bowls. You are asked to divide all pearls into the two bowls in whatever way you like as long as each bowl has at least one pearl. Once you are done, the room will turn pitch black. The bowls will move and shuffle around. In the dark where you can see nothing, you have to pick up one pearl from any bowl. If the pearl you have in your hand is white, you will be allowed to live, but if the pearl you picked is black, the room will be filled with poisonous gas and you will die.

How would you divide the pearls to increase your chances of survival? To answer this question, write a program that simulates you picking a pearl one hundred times for each of every possible choice you can make. Display the number of your survival in each of your choice. Here is a sample run:

```
Choice 1: [ ] and [ ], the number of survivals is ____ / 100 times
Choice 2: [ ] and [ ], the number of survivals is ____ / 100 times
...
```