**Activities: Loops**

The activities on this page will allow you to demonstrate your ability to:

* Create a Python script that uses a for loop to repeat an activity until a specific criterion is met.
* Create a Python script that uses a while loop to repeat an activity as long as a specific criterion holds true.

**Activity 1**

Given the [list](https://academy.engagelms.com/mod/page/view.php?id=185968) fruit\_list, write a script that iterates through the [list](https://academy.engagelms.com/mod/page/view.php?id=185968) and prints each item on a separate line.

|  |
| --- |
| fruit\_list = ["apple", "banana", "cherry", "gooseberry", "kumquat", "orange", "pineapple"]    # Your code goes here  for fruit in fruit\_list:      print(fruit) |

**Activity 2**

Write a Python script that asks the user for a string and displays the characters of the string to the user, with each character on a new line.

For example, if the input is *Hello*, the output should be:

H

|  |
| --- |
| # Ask the user for input  user\_input = input("Please enter a string: ")  # Iterate over each character in the input string  for char in user\_input:      print(char) |

**Activity 3**

Write a Python script that computes the length of a string without using the len() function.

|  |
| --- |
| input\_string = input("Please enter a string: ")  length = 0  for char in input\_string:      length += 1  print("Length of the string '{}' is: {}".format(input\_string, length)) |

**Activity 4**

Create a program that starts with a [list](https://academy.engagelms.com/mod/page/view.php?id=185968) of strings, identifies all the strings with more than two characters, stores the results in another [list](https://academy.engagelms.com/mod/page/view.php?id=185968), and displays the new [list](https://academy.engagelms.com/mod/page/view.php?id=185968).

For example:

a = ["a", "bc", "rye", "hello", "c", ""]

Output:

["rye", "hello"]

|  |
| --- |
| a = ["a", "bc", "rye", "hello", "c", ""]  # Using list comprehension to filter strings with more than two characters  result\_list = [string for string in a if len(string) > 2]  # Display the new list containing strings with more than two characters  print("Strings with more than two characters:", result\_list) |

**Activity 5**

Write two scripts, each of which displays all numbers divisible by 50 between 100 and 1000 (inclusive).

* Use the range function with for in one script.
* Use while without range in the other script.

Both scripts should have identical outputs.

|  |
| --- |
| # Part 1  print("Using for loop with range() function:")  for num in range(100, 1001, 50):      print(num)  # Part 2  print("Using while loop without range():")  num = 100  while num <= 1000:      if num % 50 == 0:          print(num)      num += 50 |

**Activity 6**

Create a script that computes the sum of all numbers between 0 and 100.

|  |
| --- |
| # Initialize a variable to store the sum  total\_sum = 0  # Iterate through numbers from 0 to 100 (inclusive)  for num in range(101):      total\_sum += num  # Print the total sum  print("The sum of all numbers between 0 and 100 is:", total\_sum) |

**Activity 7**

Create a script that computes the [factorial](https://en.wikipedia.org/wiki/Factorial) of any number input by the user.

|  |
| --- |
| # Function to compute factorial  def compute\_factorial(n):      factorial = 1      for i in range(2, n + 1):          factorial \*= i      return factorial  # Get user input  num = int(input("Enter a number to compute its factorial: "))  # Calculate and display factorial  if num < 0:      print("Factorial is not defined for negative numbers.")  else:      print(f"The factorial of {num} is: {compute\_factorial(num)}") |

**Activity 8**

Starting with the defined fruit\_list in the code block below, update the script to perform the following tasks.

1. Prompt the user to enter the name of a fruit.
2. If the fruit is in fruit\_list, display an appropriate message to the user and tell the user its index value in the [list](https://academy.engagelms.com/mod/page/view.php?id=185968).
3. If the fruit is not in fruit\_list, display an appropriate message to the user and prompt them to try again.
4. The script should repeat itself until the user enters a stop word at the prompt.

|  |
| --- |
| fruit\_list = ["apple", "banana", "cherry", "gooseberry", "kumquat", "orange", "pineapple"]  while True:      user\_input = input("Enter the name of a fruit (or 'stop' to end): ").strip().lower()        if user\_input == 'stop':          print("Exiting the program...")          break        try:          index = fruit\_list.index(user\_input)          print(user\_input + " is in the list at index " + str(index) + ".")      except ValueError:          print(user\_input + " is not in the list. Please try again.") |

**Tip**

It's always a good idea to tell the user how to end a loop!

**Activity 9**

Create a script that asks the user for a variable number of values and displays the sum of those values to the user. The program should prompt the user for values until the user enters the word "quit" (uppercase or lowercase), display the values used in the calculation, and then display the total of those values.

|  |
| --- |
| # Initialize variables  values = []  # Continuously prompt for values until "quit" is entered  while True:      user\_input = input("Enter a value (or 'quit' to finish): ").strip().lower()      if user\_input == 'quit':          break      try:          values.append(float(user\_input))      except ValueError:          print("Invalid input. Please enter a numeric value or 'quit'.")  # Display the values used in the calculation and total sum  print("Values used in calculation:", values)  print("Total sum:", sum(values)) |

**Activity 10**

Write a script that asks the user for an integer value and then displays the multiplication table of that input number from 1 through the integer squared.

|  |
| --- |
| # Prompt user for an integer input  while True:      try:          num = int(input("Enter an integer value: "))          break      except ValueError:          print("Invalid input. Please enter an integer.")  # Display multiplication table from 1 through num^2  print("Multiplication table for", num, ":")  for i in range(1, num + 1):      for j in range(1, num + 1):          print(i, "x", j, "=", i \* j) |

**Activity 11**

Create a script that identifies all prime numbers between 0 and 100.

|  |
| --- |
| # Function to check if a number is prime  def is\_prime(num):      if num <= 1:          return False      if num <= 3:          return True  # 2 and 3 are prime numbers        # Check for factors from 2 to the square root of num      for i in range(2, int(num\*\*0.5) + 1):          if num % i == 0:              return False        return True  # Find and print all prime numbers between 0 and 100  print("Prime numbers between 0 and 100:")  for number in range(101):      if is\_prime(number):          print(number, end=" ") |

**Activity 12**

Write a script that calculates the greatest common denominator between two numbers.

For example, given the numbers 18 and 27, the greatest common denominator is 9.

|  |
| --- |
| # Function to calculate the greatest common divisor  def gcd(a, b):      while b != 0:          a, b = b, a % b      return a  # Prompt the user for two integers  while True:      num1\_str = input("Enter the first integer (or 'quit' to exit): ").strip().lower()      if num1\_str == 'quit':          print("Exiting the program...")          break        num2\_str = input("Enter the second integer: ").strip().lower()        # Validate input      try:          num1 = int(num1\_str)          num2 = int(num2\_str)            # Calculate GCD          result\_gcd = gcd(num1, num2)            # Print result          print(f"The greatest common denominator of {num1} and {num2} is: {result\_gcd}")      except ValueError:          print("Invalid input. Please enter integers.") |

**Activity 13**

Write a Python script that computes the frequency of each digit in a given integer.

For example, if the input number is 334, the output should be:

3 occurs 2 times

4 occurs 1 time

|  |
| --- |
| # Prompt user for an integer input  while True:      try:          num = int(input("Enter an integer number: "))          break      except ValueError:          print("Invalid input. Please enter an integer.")  # Compute digit frequencies using dictionary comprehension  frequency = {digit: str(num).count(digit) for digit in set(str(num)) if digit.isdigit()}  # Print the results  print("Digit frequencies:")  for digit, count in frequency.items():      print(f"{digit} occurs {count} {'time' if count == 1 else 'times'}") |

**Activity 14**

Write a script that calculates the lowest common multiple of two given integers.

For example, given the values 4 and 6, the lowest common multiple is 12.

|  |
| --- |
| # Function to calculate the greatest common divisor using Euclidean algorithm  def gcd(a, b):      while b != 0:          a, b = b, a % b      return a  # Function to calculate the lowest common multiple (LCM)  def lcm(a, b):      return abs(a \* b) // gcd(a, b)  # Prompt the user for two integers  while True:      try:          num1 = int(input("Enter the first integer: "))          num2 = int(input("Enter the second integer: "))          break      except ValueError:          print("Invalid input. Please enter integers.")  # Calculate GCD  def gcd(a,b):      while b:       a,b=b,a% |

**Activity 15**

Write a Python script that determines if an input number can be expressed as the sum of two prime numbers.

For example, the number 10 can be expressed as the sum of two prime numbers:

* 10 = 3 + 7 : both prime numbers
* 10 = 5 + 5 : both prime numbers

However, the number 11 cannot be:

* 11 = 1 + 10 : neither 1 nor 10 are prime numbers
* 11 = 2 + 9 : 9 is not a prime number
* 11 = 3 + 8 : 8 is not a prime number
* 11 = 4 + 7 : 4 is not a prime number
* 11 = 5 + 6 : 6 is not a prime number

|  |
| --- |
| # Prompt user for an integer input  while True:      try:          number = int(input("Enter an integer (greater than 1): "))          if number > 1:              break          else:              print("Please enter a number greater than 1.")      except ValueError:          print("Invalid input. Please enter an integer.")  # Check if the number can be expressed as the sum of two primes  found = False  for p in range(2, number // 2 + 1):      # Check if p and (number - p) are prime      is\_p\_prime = True      is\_number\_minus\_p\_prime = True        for i in range(2, int(p \*\* 0.5) + 1):          if p % i == 0:              is\_p\_prime = False              break        if is\_p\_prime:          for i in range(2, int((number - p) \*\* 0.5) + 1):              if (number - p) % i == 0:                  is\_number\_minus\_p\_prime = False                  break            if is\_number\_minus\_p\_prime:              print(f"{number} = {p} + {number - p}")              found = True              break  if not found:      print(f"{number} cannot be expressed as the sum of two prime numbers.") |