05-if-else-for-loop

July 23, 2024

1 IF , ELSE.... FOR LOOP

print this is hour in current minute is odd

[2]:
```
from datetime import datetime

# Get the current time
now = datetime.now()

# Get the current minute
minute = now.minute

# Check if the minute is odd or even and print the appropriate message
if minute % 2 != 0:
    print(f"minute : {minute}")
    print("This is Odd minute")
else:
    print(f"minute : {minute}")
    print("This is Even minute")
```

```
minute : 12
This is Even minute
```

if today is satarday then print 'Party !!!' or if sunday print take rest other wise print work , work, work

[4]:
```
#CODE HERE
day_of_week = "Sunday"  # Manually set for demonstration

if day_of_week == "Saturday":
    print("Party !!!")
elif day_of_week == "Sunday":
    print("Today is : Sunday")
    print("Take Rest.")
else:
    print("Work, work, work.")
```

```
Today is : Sunday
Take Rest.
```

Write a program which will find all such numbers which are divisible by 7 but are not a multiple of 5, between 2000 and 3200 (both included). The numbers obtained should be printed in a comma-separated sequence on a single line.

[ ]:
```
#CODE HERE
```

[5]:
```
#CODE HERE
# Initialize an empty list to store the numbers
numbers = []

# Iterate through the range from 2000 to 3200 (both included)
for i in range(2000, 3201):
    # Check if the number is divisible by 7 but not a multiple of 5
    if i % 7 == 0 and i % 5 != 0:
        numbers.append(str(i))

# Print the numbers in a comma-separated sequence
print(",".join(numbers))
```

```
2002,2009,2016,2023,2037,2044,2051,2058,2072,2079,2086,2093,2107,2114,2121,2128,
2142,2149,2156,2163,2177,2184,2191,2198,2212,2219,2226,2233,2247,2254,2261,2268,
2282,2289,2296,2303,2317,2324,2331,2338,2352,2359,2366,2373,2387,2394,2401,2408,
2422,2429,2436,2443,2457,2464,2471,2478,2492,2499,2506,2513,2527,2534,2541,2548,
2562,2569,2576,2583,2597,2604,2611,2618,2632,2639,2646,2653,2667,2674,2681,2688,
2702,2709,2716,2723,2737,2744,2751,2758,2772,2779,2786,2793,2807,2814,2821,2828,
2842,2849,2856,2863,2877,2884,2891,2898,2912,2919,2926,2933,2947,2954,2961,2968,
2982,2989,2996,3003,3017,3024,3031,3038,3052,3059,3066,3073,3087,3094,3101,3108,
3122,3129,3136,3143,3157,3164,3171,3178,3192,3199
```

Write a program which can compute the factorial of a given numbers. The results should be printed in a comma-separated sequence on a single line. Suppose the following input is supplied to the program: 8 Then, the output should be: 40320

[8]:
```
#CODE HERE
def factorial(n):
    if n == 0:
        return 1
    else:
        return n * factorial(n-1)

# Get input from the user
number = int(input("Enter the value to compute the factorial: "))

# Compute the factorial
result = factorial(number)

# Print the result in the specified format
print(f"{number}! = {result}")
```

```
Enter the value to compute the factorial: 8
8! = 40320
```

With a given integral number n, write a program to generate a dictionary that contains (i, i*i) such that is an integral number between 1 and n (both included). and then the program should print the dictionary. Suppose the following input is supplied to the program: 8 Then, the output should be: {1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64}

Hint : use dict()

[9]:
```
#CODE HERE
# Get input from the user
n = int(input("Enter the value: "))

# Generate the dictionary using a dictionary comprehension
squares_dict = {i: i * i for i in range(1, n + 1)}

# Print the dictionary
print(squares_dict)
```

```
Enter the value: 8
{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64}
```

create user defined fuction for - addition “add” - subtraction “sub” - multiplication “mul” - division “div” - power “pow”

[10]:
```
#CODE HERE
def add(a, b):
    return a + b

def sub(a, b):
    return a - b

def mul(a, b):
    return a * b

def div(a, b):
    if b != 0:
        return a / b
    else:
        return "Division by zero is not allowed"

def pow(a, b):
    return a ** b

# Get inputs from the user
a = int(input("Please enter the value : "))
b = int(input("Please enter the value : "))
```

```
# Get the operation keyword from the user
operation = input("Enter the keyword {add, sub, mul, div, pow}:")

# Perform the operation based on the keyword
if operation == "add":
    result = add(a, b)
    print(f"Addition of {a} and {b} is : {result}")
elif operation == "sub":
    result = sub(a, b)
    print(f"Subtraction of {a} from {b} is : {result}")
elif operation == "mul":
    result = mul(a, b)
    print(f"Multiplication of {a} and {b} is : {result}")
elif operation == "div":
    result = div(a, b)
    print(f"Division of {a} by {b} is : {result}")
elif operation == "pow":
    result = pow(a, b)
    print(f"Power of {a} by {b} is : {result}")
else:
    print("Invalid operation keyword")
```

```
Please enter the value : 8
Please enter the value : 6
Enter the keyword {add, sub, mul, div, pow}:pow
Power of 8 by 6 is : 262144
```

1.0.1 Write a Python program that accepts a string and calculate the number of digits and letters.

- Input
  - Hello321Bye360
- Output
  - Digit - 6
  - Letter - 8

[11]:
```
# CODE HERE
word = input("ENter the text: ")

# Initialize counters for digits and letters
digit_count = 0
letter_count = 0

# Iterate through each character in the input string
for char in word:
```

```
    if char.isdigit():
        digit_count += 1
    elif char.isalpha():
        letter_count += 1

# Print the counts
print(f"Digit - {digit_count}")
print(f"Letter - {letter_count}")
```

```
ENter the text: innomatics1231239746
Digit - 10
Letter - 10
```

1.0.2 With a given tuple (1,2,3,4,5,6,7,8,9,10), write a program to print the first half values in one line and the last half values in one line.

[12]:
```
# METHOD - 1
# Given tuple
tpl = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)

# Calculate the midpoint
mid = len(tpl) // 2

# Split the tuple into two halves
first_half = tpl[:mid]
second_half = tpl[mid:]

# Print the first half values in one line
print(" ".join(map(str, first_half)))

# Print the second half values in one line
print(" ".join(map(str, second_half)))
```

```
1 2 3 4 5
6 7 8 9 10
```

[13]:
```
# METHOD - 2
# Given tuple
tpl = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)

# Calculate the midpoint
mid = len(tpl) // 2

# Split the tuple into two halves
first_half = tpl[:mid]
second_half = tpl[mid:]
```

```
# Print the halves as tuples
print(first_half, second_half)
```

```
(1, 2, 3, 4, 5) (6, 7, 8, 9, 10)
```

[14]:
```
# METHOD - 3
# Given tuple
tpl = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)

# Calculate the midpoint
mid = len(tpl) // 2

# Split the tuple into two halves and convert them to lists
first_half = list(tpl[:mid])
second_half = list(tpl[mid:])

# Print the halves as lists
print(first_half)
print(second_half)
```

```
[1, 2, 3, 4, 5]
[6, 7, 8, 9, 10]
```

1.0.3 Write a program to generate and print another tuple whose values are even numbers in the given tuple (1,2,3,4,5,6,7,8,9,10)

[15]:
```
# Given tuple
tpl = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10)

# Generate a tuple with only the even numbers
even_numbers = tuple(x for x in tpl if x % 2 == 0)

# Print the resulting tuple
print(even_numbers)
```

```
(2, 4, 6, 8, 10)
```

1.0.4 Write a program which accepts a string as input to print "Yes" if the string is "yes" or "YES" or "Yes", otherwise print "No"

[16]:
```
text = input("Please type something. --> ")

# Check if the input string matches "yes", "YES", or "Yes"
if text in ["yes", "YES", "Yes"]:
    print("Yes")
else:
    print("No")
```

```
Please type something. --> yes
Yes
```

1.0.5 Please write a program to generate a list with 5 random numbers between 100 and 200 inclusive.

[17]:
```python
import random

# Generate a list with 5 random numbers between 100 and 200 inclusive
random_numbers = [random.randint(100, 200) for _ in range(5)]

# Print the list
print(random_numbers)
```

```
[155, 130, 153, 197, 143]
```

1.0.6 Please generate a random float where the value is between 5 and 95 using Python module.

[20]:
```python
import random

# Generate a random float between 5 and 95
random_float = random.uniform(5, 95)

# Print the random float
print(random_float)
```

```
70.80449749608682
```

1.0.7 [1,1,1,1,3,3,4,4,4,5,6,7,8,8,8]

1. from the above list print the uniques values

2. count how many times each number exist

3. Print the number which is occured more number of times

[21]:
```python
from collections import Counter

# Given list
numbers = [1, 1, 1, 1, 3, 3, 4, 4, 4, 5, 6, 7, 8, 8, 8]

# 1. Print the unique values
unique_values = set(numbers)
print("Unique values:", unique_values)
```

```
Unique values: {1, 3, 4, 5, 6, 7, 8}
```

[22]:
```
# 2. Count how many times each number exists
count = Counter(numbers)
print("Count of each number:", count)
```

```
Count of each number: Counter({1: 4, 4: 3, 8: 3, 3: 2, 5: 1, 6: 1, 7: 1})
```

[23]:
```
# 3. Print the number which occurred the most times
most_common = count.most_common(1)[0][0]  # Get the most common number
print("Number that occurred the most times:", most_common)
```

```
Number that occurred the most times: 1
```

2 Innomatics Research Labs

www.innomatics.in

[ ]: