Day 6

Date 12 June 2024

Daily Report

Today's topic was based on Python Fuction, Lambda Fuction, Recursive function, functions all categories of function arguments, list comprehension & dictionary comprehension

---

Today's topic

Functions

In Python, a function is used for specific task and is reusable.

```
def function_name(arguments):
  #code
```

function calling

```
variable = function_name(actual argument)
```

Type of Function:-

1. Built-in-function:- These are predefined Functions in python.
2. User-defined function:- these are functions created by programmer to perform specific task. They follow a defined structure given above.
3. Anonymous Function(Lambda Function):- These are function without name,often used for short, simple operations.
4. Recursive Functions:-These functions call themselves within their definition, useful for task that can be divided into similar subtasks.

Argument in Functinos

Arguments are values passed to a function when it is called.

1. Positonal Arguments:- These are arguments that are passed to a function in a specific order.
2. Keyword Arguments:- These are arguments passed to a function by explicitly naming each parameter and its corresponding value.
3. Default Argument:- These are arguments that are assume a default value if no value is provided during the function call.
4. Variable-Length Argument:- *args(Non-keyword arguments) allows a function to accept any number of positional arguments.
5. kwargs(keyword argument) allows a function to accept any number of keyword arguments.

List Comprehension

List comprehension provides an elegent way a create new lists.

```
[ expression for item in list]
```

Dictionary Comprehension

```
{key:value for var in iterable}
```

Some Question for practice:-

```
#Reverse the string
def reverse_string():
  st = input("Enter string : ")
  return st[::-1]
reverse_string()
```

```
Enter string : hello
'olleh'
```

```
#Create a list of numbers. Write a function that finds and returns the maximum value in the list without using the built-in max() function.
def max(n):
  c = n[0]
  for x in n:
    if x>c:
      c = x
  return c
num = [3,5,2,235,73,3,5]
print(max(num))
```

```
235
```

```
#Define a function that accepts a list of integers and returns a new list containing only the even numbers from the original list.
def even_number(num):
  num1 = []
  for x in num:
    if x%2==0:
      num1.append(x)
  return num1
num = [23,56,34,9,12,2,78,44]
print(even_number(num))
```

```
[56, 34, 12, 2, 78, 44]
```

```
#Implement a Python function to check if a given word is a palindrome (reads the same backward as forward).
def palindrom(st):
  rev = st[::-1]
  if rev == st:
    return "palindrom"
  else:
    return "not palindrom"
st = input("Enter a string : ")
print(palindrom(st))
```

```
Enter a string : teacher
not palindrom
```

```
#Create a dictionary with student names as keys and their corresponding ages as values.
#Write a function to find and print the names of students who are above a certain age.
def student(**n):
  for k,v in n.items():
    if v>20:
      print(k)
student(john = 34,harry = 13, wilson = 25)
```

```
john
wilson
```

```
#Develop a Python function that calculates the sum of squares for a given range of numbers.
def square(a):
  c = 0
  for x in range(0,a+1):
    c +=x*x
  return c
print(square(10))
```

```
385
```

```
#Use a lambda function to filter a list of integers and return a new list containing only the numbers greater than 10.
num = lambda x:x>10
n = [23,4,6,12,9,56,3]
for x in n:
  if num(x) == True:
    print(x,end = " ")
```

```
23 12 56
```

```
#Create a Python function that accepts a string and counts the occurrences of each character. Return the result as a dictionary.
def count(st):
  res = {k:st.count(k) for k in st}
  return res
st = "paradise"
print(count(st))
```

```
{'p': 1, 'a': 2, 'r': 1, 'd': 1, 'i': 1, 's': 1, 'e': 1}
```

```
#Implement a function to calculate the average of a list of numbers without using the built-in sum() and len() functions.
def average(num):
  c,i = 0,0
  for x in num:
    c +=x
    i+=1
  average = c/i
  return average
num = [12,2,3,4,5,6,7,8]
print(average(num)
)
```

```
5.875
```

```
#.Write a function that checks if a given year is a leap year. A leap year is divisible by 4 but not divisible by 100 unless it is divisible
def leap_year(year):
  if year%4 == 0 and year%100!=0:
    return "leap year"
  else:
    return "Not leap year"
year = int(input("Enter Year : "))
print(leap_year(year))
```

```
Enter Year : 2024
leap year
```

```
#Design a function that takes a list of strings and returns a new list with only the strings that have more than 2 characters.
def string_cal(lis):
  res = [k for k in lis if len(k)>2]
  return res
lis = ["res",'hi',"yes",'hr']
print(string_cal(lis))
```

```
['res', 'yes']
```

Start coding or generate with AI.