

Python Basic

Lab Assignment 1



Name: Japman Singh

Roll number: 04913202717, CSE1

GitHub Repository: https://github.com/japmansingh/ML_College/

Q1. Create a program that asks the user to enter their name and their age. Print out a message addressed to them that tells them the year that they will turn 100 years old. Extras:

1. Add on to the previous program by asking the user for another number and printing out that many copies of the previous message.
2. Print out that many copies of the previous message on separate lines. (Hint: the string `"\n"` is the same as pressing the ENTER button)

Pseudocode:

```
input: name, age, repeat
output: year for age 100 years
initialise name, age
initialise year to now.year + 100 - age
initialise result to name + 100 years old year
print result
initialise repeat
print result repeat number of times
```

Program Screenshot and Output:

[illegible]

Q2. Take a list, say for example this one: a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89] and write a program that prints out all the elements of the list that are less than 5.

Extras:

1. Instead of printing the elements one by one, make a new list that has all the elements less than 5 from this list in it and print out this new list.
2. Write this in one line of Python.
3. Ask the user for a number and return a list that contains only elements from the original list a that are smaller than that number given by the user.

Pseudocode:

input: list

output: elements greater than 5

initialise a as [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]

for each item in a ,

 if item is smaller than 5:

 print element

initialise new_list to item

for item in a

if item is smaller than 5

print new_list

initialise num

print item

for each item in a

if item is smaller than num

Program screenshot and Output:

Q2. Take a list, say for example this one: a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89] and write a program that prints out all the elements of the list that are less than 5. Extras: 1. Instead of printing the elements one by one, make a new list that has all the elements less than 5 from this list in it and print out this new list. 2. Write this in one line of Python. 3. Ask the user for a number and return a list that contains only elements from the original list a that are smaller than that number given by the user.

[24]

▶  MI

```
a = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]
```

```
for item in a:
    if(int(item) < 5 ):
        print(item, "\n")
```

```
nlist = [ item for item in a if item < 5 ]
print('The new list is - ' + str(nlist))
```

```
number = input('Enter a number to compare and print smaller numbers')
print([ item for item in a if item < int(number) ])
```

1

1

2

3

The new list is - [1, 1, 2, 3]

Enter a number to compare and print smaller numbers5

[1, 1, 2, 3]

Q3. Write a program that asks the user how many Fibonacci numbers to generate and then generates them. Take this opportunity to think about how you can use functions. Make sure to ask the user to enter the number of numbers in the sequence to generate.(Hint: The Fibonacci sequence is a sequence of numbers where the next number in the sequence is the sum of the previous two numbers in the sequence. The sequence looks like this: 1, 1, 2, 3, 5, 8, 13, ...).

Pseudocode:

input: limit

for fibonacci series,
output: fibonacci sequence upto n terms

function fibonacci(n_terms)

initialise n1 to 0
initialise n2 to 1
initialise count to 0

if n
n_terms is greater than or equal to 0
print enter an integer

elif n_terms is equal 1
print Fibonacci sequence upto n_terms print n1

else: print Fibonacci sequence while count is smaller than n_terms
print n1

initialise nth to n1 + n2
set n1 to n2
set n2 to nth
set count += 1
initialise length
print fibonacci(n_terms)

Program screenshot and Output:

Q3. Write a program that asks the user how many Fibonacci numbers to generate and then generates them. Take this opportunity to think about how you can use functions. Make sure to ask the user to enter the number of numbers in the sequence to generate. (Hint: The Fibonacci sequence is a sequence of numbers where the next number in the sequence is the sum of the previous two numbers in the sequence. The sequence looks like this: 1, 1, 2, 3, 5, 8, 13, ...)

```
[28] ▶ ▶≡ MI
def fibo(n_terms):
    n1, n2 = 0, 1
    count = 0
    if n_terms <= 0:
        print("Please enter a positive integer")
    elif n_terms == 1:
        print("Fibonacci sequence upto",n_terms,":")
        print(n1)
    else:
        print("Fibonacci sequence:")
        while count < n_terms:
            print(n1)
            nth = n1 + n2
            n1 = n2
            n2 = nth
            count += 1
    length = input('Enter the length of Fibonacci series: ')
    fibo(int(length))
```

```
Enter the length of Fibonacci series: 5
Fibonacci sequence:
0
1
1
2
3
```

Q4. Write a program (function!) that takes a list and returns a new list that contains all the elements of the first list minus all the duplicates.

Extras:

- Write two different functions to do this - one using a loop and constructing a list, and another using sets.

Pseudocode:

```
function duplicate_f1(list)
  initialise new_list
  for each i in list
    if i not in new_list
      new_list.append(i)
  return new_list
```

```
function duplicate_f2(list)
  return list(set(x))
  initialise user_list
  initialise limit
  for each i in range limit
    initialise elem
    append elem in user_list
  print user_list
  print Function using loop
  duplicate_f1(user_list)
  print Function
  set duplicate_f2(user_list)
```

Program screenshot and Output:

Q4. Write a program (function!) that takes a list and returns a new list that contains all the elements of the first list minus all the duplicates. Extras:

- Write two different functions to do this - one using a loop and constructing a list, and another using sets.

[1]

▶ ▶≡ MI

```
def duplicate_f1(x):
    new_list = []
    for i in x:
        if i not in new_list:
            new_list.append(i)
    return new_list

def duplicate_f2(x):
    return list(set(x))

user_list=[]
limit = int(input('Enter the length of the list: '))
for i in range(limit):
    ele = int(input('Enter '+str(i)+'th element:'))
    user_list.append(ele)

print(user_list)
print ('Function using loop: '+str(duplicate_f1(user_list)))
print ('Function using set: '+str(duplicate_f2(user_list)))
```

[2, 5, 6, 8, 9]

Function using loop: [2, 5, 6, 8, 9]

Function using set: [2, 5, 6, 8, 9]

Q5. Ask the user for a number and determine whether the number is prime or not. (For those who have forgotten, a prime number is a number that has no divisors.). Use functions.

Pseudocode:

```
input: number
output: number is prime or not
function prime(number)
  initialise flag to false
  if number is greater than 1
    for each i from 2 to number
      if i modulus number is 0
        set flag to true
        break
  if flag is true
    print number is a prime number
  else print number is not a prime number
```

Program screenshot and Output:

Q5. Ask the user for a number and determine whether the number is prime or not. (For those who have forgotten, a prime number is a number that has no divisors.). Use functions

[20]



```
def prime_check():
    num = int(input("Enter Number you want to check \t"))
    flag = 1
    for i in range(2, int(num/2)):
        if(num%i == 0):
            flag = 0
            break
    if(flag==0):
        print("Number is not Prime")
    else:
        print("Number is Prime")
    prime_check()
```

```
Enter Number you want to check 11
Number is Prime
```