Exercise:

1. using for loop to print the number from 1 to 10

for i in range(1,11,1):

print(i)

1. using for loop to print number from 2 to 20 by 2. Ex: 2, 4, 6,8

for i in range(2,21,2):

print(i)

1. a = [‘cat’, ‘dog’, ‘zebra’] b = [1, 2, 3] print each combination. Ex: (cat,1), (cat, 2), (cat,3), (dog ,1)…

for animal in a:

for num in b:

print(a,b)

**ADVANCED FOR LOOP – LIST COMPREHENSION**

**Syntax**

The list comprehension starts with a '**[' and ']'**, to help you remember that the

result is going to be a list.

The basic syntax is

[ expression **for** item **in** list **if** condition ]

This is equivalent to:

**for** item **in** list:

**if** conditional:

expression

**Old Way:**

years\_of\_birth **=** [1990, 1991, 1990, 1990, 1992, 1991]

ages **=** []

**for** year **in** years\_of\_birth:

ages**.**append(2018 **-** year)

**List Comprehension:**

ages = [2018-year for year in years\_of\_birth]

**Example:**

a = [1, 4, 9, 16, 25, 36, 49, 64, 81, 100]. Write one line of Python that takes this list a and makes a new list that has only the even elements of this list in it.

**Old Way:**

even = []

for num in a:

if num%2==0:

even.append(num)

**List Comprehension:**

even = [num for num in a if num%2 ==0 ]

**Nested Loop using List comprehension**

a = [‘cat’, ‘dog’, ‘zebra’] b = [1, 2, 3] print each combination. Ex: (cat,1), (cat, 2), (cat,3), (dog ,1)…

**old way**

a = [‘cat’, ‘dog’, ‘zebra’]

b = [1, 2, 3]

for animal in a:

for num in b:

print((animal,num))

**list comprehension**

**[**result **for** outter loop **for** inner loop**]**

[(animal, num) for animal in a for num in b]

**FUNCTIONS**

**Creating a Function**

In Python a function is defined using the def keyword:

**Example:**

def my\_function():

print("Hello from a function")

**Calling a Function**

To call a function, use the function name followed by parenthesis:

**Example:**

def my\_function():  
  print("Hello from a function")  
  
my\_function()

**Parameters**

Information can be passed to functions as parameter.

Parameters are specified after the function name, inside the parentheses. You can add as many parameters as you want, just separate them with a comma.

**Example**

def my\_function(name):

print('your name is "'+name+ '"')

my\_function("Emily")

**Example:**

def my\_function(fname, lname):

print(‘your name is “ ’+fname,lname+ ‘”’)

my\_function(‘Emliy’, ‘Blunt’)

**Default Parameter Value**

The following example shows how to use a default parameter value.

If we call the function without parameter, it uses the default value:

**Example**

def my\_function(country = "China"):  
  print("I am from " + country)  
  
my\_function("Sweden")  
my\_function("India")  
my\_function()  
my\_function("Brazil")

**Return Values**

To let a function return a value, use the return statement:

**Example**

def my\_function(x):  
  return 5 \* x

print(my\_function(3))  
print(my\_function(5))  
print(my\_function(9))

**difference between print and return in function**

def function\_that\_prints():

print("I printed")

def function\_that\_returns():

return("I returned")

f1 = function\_that\_prints()

f2 = function\_that\_returns()

print "Now let us see what the values of f1 and f2 are"

print f1

print f2

**Exercise:**

1. Create a function named my\_function.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:

print("Hello from a function")

1. Call (execute) a function named my\_function.
2. Inside a function with two parameters, print the first parameter.

def my\_function(fname, lname):

print(\_\_\_\_\_)

1. Let the function return the x parameter + 5.
2. Define a function that can receive two integral numbers in string form and compute their sum and then print it in console.

def printValue(s1,s2):

print int(s1)+int(s2)

printValue("3","4")

1. Write a method which can calculate square value of number

def square(num):

return num \*\* 2

print square(2)

print square(3)

1. Write a Python function to sum all the numbers in a list.

def sum(numbers):

total = 0

for x in numbers:

total += x

return total

print(sum((8, 2, 3, 0, 7)))

1. Write a Python function to multiply all the numbers in a list.

def multiply(numbers):

total = 1

for x in numbers:

total \*= x

return total

print(multiply([8, 2, 3, -1, 7]))

1. Write a Python function to check if the number is odd or even

def odd\_or\_even(num):

if num%2 > 0:

return(‘odd’)

else:

return(‘even’)

1. Define a function which can generate and print a list where the values are square of numbers between 1 and 20 (both included).

def printList():

li=list()

for i in range(1,21):

li.append(i\*\*2)

return (li)

printList()

**list comprehensions**

def printList():

li = [i\*\*2 for i in range(1,21)]

return(li)

**self-define starting and ending point**

def printList(start, end):

li = [i\*\*2 for i in range(start, end)]

return(li)

**Pretend we have two lists, one is students’ names and the other one is their scores:**

Names = ['Tom', 'Jimmy', 'Alice','Alex','Ben','Tanya', 'Howard','Edward','Jack',"Rose"]

Score = [98,77,65,77,85,78,83,92,100,99]

**How do we create a dictionary and convert the score into letter grade?**

**First, we need to create a function that**

def grade(score):

if score >=90:

letter = 'A'

elif score >=80:

letter = 'B'

elif score >= 70:

letter = 'C'

elif score >= 60:

letter = 'D'

else:

letter = "F"

return(letter)

**Secondly, we use a for loop to convert the scores into grades:**

**We use list comprehension we mentioned earlier to convert the score into grades**

new\_score = [grade(s) for s in score]

**Check if the loop ran properly**

new\_score

**Lastly, we define a function that can combine to lists into dictionary**

def combine\_list(name, grade):

new\_lis = zip(name,grade)

new\_dict = dict(new\_lis)

return(new\_dict)

**Then we pass in the two lists and call the function**

Combine\_list(names, new\_score)

**New Guessing Word Game:**

def guess\_word(word):

word = word.upper()

guessed = "\_" \* len(word)

word = list(word)

guessed = list(guessed)

lstGuessed = []

letter = input("guess letter: ")

while True:

if letter.upper() in lstGuessed:

letter = ''

print("Already guessed!!")

elif letter.upper() in word:

index = word.index(letter.upper())

guessed[index] = letter.upper()

word[index] = '\_'

else:

print(''.join(guessed))

if letter is not '':

lstGuessed.append(letter.upper())

letter = input("guess letter: ")

if '\_' not in guessed:

print(''.join(guessed))

print("You won!!")

break