Day 5:

```
Lists:
```

How to Create a List
Accessing List Items Using Positive Indexing
Accessing List Items Using Negative Indexing
Unpacking List Items
Slicing Items from a List
Modifying Lists
Checking Items in a List
Adding Items to a List
Inserting Items into a List
Removing Items from a List
Removing Items Using Pop
Removing Items Using Del
Clearing List Items
Copying a List
Joining Lists
Counting Items in a List
Finding Index of an Item
Reversing a List
Sorting List Items
Exercises: Day 5

Lists

There are four collection data types in python :

List: is a collection which is ordered and changeable(modifiable). Allows duplicate members.
Tuple: is a collection which is ordered and unchangeable or unmodifiable(immutable). Allows duplicate members.
Set: is a collection which is unordered, unindexed and unmodifiable, but you can add new items. No duplicate members.
Dictionary: is a collection which is unordered, changeable(modifiable) and indexed. No duplicate members.
A list is collection of different data types which is ordered and modifiable(mutable). A list can be empty or it may have
different data type items or items

How to Create a List

In python we can create lists in two ways:

Using list built-in function
# syntax
lst = list()
empty_list = list() # this is an empty list, no item in the list
print(len(empty_list)) # 0
Using square brackets, []
# syntax
lst = []
empty_list = [] # this is an empty list, no item in the list
print(len(empty_list)) # 0

In [144]:

```
#Lists with initial values. We use len() to find the length of a list.

fruits = ['banana', 'orange', 'mango', 'lemon']                     # list of fruits
vegetables = ['Tomato', 'Potato', 'Cabbage','Onion', 'Carrot']      # list of vegetable
s
animal_products = ['milk', 'meat', 'butter', 'yoghurt']             # list of animal pr
oducts
web_techs = ['HTML', 'CSS', 'JS', 'React','Redux', 'Node', 'MongDB'] # list of web tech
nologies
countries = ['Finland', 'Estonia', 'Denmark', 'Sweden', 'Norway']

# Print the lists and its length
print('Fruits:', fruits)
print('Number of fruits:', len(fruits))
print('Vegetables:', vegetables)
print('Number of vegetables:', len(vegetables))
print('Animal products:',animal_products)
print('Number of animal products:', len(animal_products))
print('Web technologies:', web_techs)
print('Number of web technologies:', len(web_techs))
print('Countries:', countries)
print('Number of countries:', len(countries))
```

```
Fruits: ['banana', 'orange', 'mango', 'lemon']
Number of fruits: 4
Vegetables: ['Tomato', 'Potato', 'Cabbage', 'Onion', 'Carrot']
Number of vegetables: 5
Animal products: ['milk', 'meat', 'butter', 'yoghurt']
Number of animal products: 4
Web technologies: ['HTML', 'CSS', 'JS', 'React', 'Redux', 'Node', 'MongD
B']
Number of web technologies: 7
Countries: ['Finland', 'Estonia', 'Denmark', 'Sweden', 'Norway']
Number of countries: 5
```

In [145]:

```
# Lists can have items of different data types
lst = ['Asabeneh', 250, True, {'country':'Finland', 'city':'Helsinki'}] # list containi
ng different data types
```

Accessing List Items Using Positive Indexing

We access each item in a list using their index. A list index starts from 0.
[0,1,2,3]

In [146]:

```
fruits = ['banana', 'orange', 'mango', 'lemon']
first_fruit = fruits[0] # we are accessing the first item using its index
print(first_fruit)      # banana
second_fruit = fruits[1]
print(second_fruit)     # orange
last_fruit = fruits[3]
print(last_fruit) # lemon
# Last index
last_index = len(fruits) - 1
last_fruit = fruits[last_index]
```

```
banana
orange
lemon
```

In [147]:

```
#Accessing List Items Using Negative Indexing
#Negative indexing means beginning from the end, -1 refers to the last item, -2 refers
 to the second last item.
fruits = ['banana', 'orange', 'mango', 'lemon']
first_fruit = fruits[-4]
last_fruit = fruits[-1]
second_last = fruits[-2]
print(first_fruit)      # banana
print(last_fruit)       # lemon
print(second_last)      # mango
```

```
banana
lemon
mango
```

In [148]:

```
lst = ['item','item2','item3', 'item4', 'item5']
first_item, second_item, third_item, *rest = lst
print(first_item)     # item1
print(second_item)    # item2
print(third_item)     # item3
print(rest)           # ['item4', 'item5']
```

```
item
item2
item3
['item4', 'item5']
```

In [149]:

```
# First Example
fruits = ['banana', 'orange', 'mango', 'lemon','lime','apple']
first_fruit, second_fruit, third_fruit, *rest = lst
print(first_fruit)     # banana
print(second_fruit)    # orange
print(third_fruit)     # mango
print(rest)           # ['lemon','lime','apple']
# Second Example about unpacking list
first, second, third,*rest, tenth = [1,2,3,4,5,6,7,8,9,10]
print(first)          # 1
print(second)         # 2
print(third)          # 3
print(rest)           # [4,5,6,7,8,9]
print(tenth)          # 10
# Third Example about unpacking list
countries = ['Germany', 'France','Belgium','Sweden','Denmark','Finland','Norway','Icela
nd','Estonia']
gr, fr, bg, sw, *scandic, es = countries
print(gr)
print(fr)
print(bg)
print(sw)
print(scandic)
print(es)
```

```
item
item2
item3
['item4', 'item5']
1
2
3
[4, 5, 6, 7, 8, 9]
10
Germany
France
Belgium
Sweden
['Denmark', 'Finland', 'Norway', 'Iceland']
Estonia
```

In [150]:

```
#Slicing Items from a List

#Positive Indexing: We can specify a range of positive indexes by specifying the start, end and step, the return value will be #a new list. (default values for start = 0, end = len(lst) - 1 (last item), step = 1)

fruits = ['banana', 'orange', 'mango', 'lemon']
all_fruits = fruits[0:4] # it returns all the fruits
# this will also give the same result as the one above
all_fruits = fruits[0:] # if we don't set where to stop it takes all the rest
orange_and_mango = fruits[1:3] # it does not include the first index
orange_mango_lemon = fruits[1:]
orange_and_lemon = fruits[::2] # here we used a 3rd argument, step. It will take every 2cnd item - ['orange', 'lemon']

#Negative Indexing: We can specify a range of negative indexes by specifying the start, end and step, the return value will be #a new list.

fruits = ['banana', 'orange', 'mango', 'lemon']
all_fruits = fruits[-4:] # it returns all the fruits
orange_and_mango = fruits[-3:-1] # it does not include the last index
orange_mango_lemon = fruits[-3:] # this will give the same result as the one above
reverse_fruits = fruits[::-1] # a negative step will take the list in reverse order
```

In [152]:

```
#Modifying Lists
#List is a mutable or modifiable ordered collection of items. Lets modify the fruit list.

fruits = ['banana', 'orange', 'mango', 'lemon']
fruits[0] = 'avocado'
print(fruits)       #  ['avocado', 'orange', 'mango', 'lemon']
fruits[1] = 'apple'
print(fruits)       #  ['avocado', 'apple', 'mango', 'lemon']
last_index = len(fruits)
```

```
['avocado', 'orange', 'mango', 'lemon']
['avocado', 'apple', 'mango', 'lemon']
```

In [153]:

```
#Checking Items in a List

fruits = ['banana', 'orange', 'mango', 'lemon']
does_exist = 'banana' in fruits
print(does_exist)  # True
does_exist = 'lime' in fruits
print(does_exist)  # False
```

```
True
False
```

In [162]:

```
#Adding Items to a List
#To add item to the end of an existing list we use the method

# syntax
lst = list()

fruits = ['banana', 'orange', 'mango', 'lemon']
fruits.append('apple')
print(fruits)           # ['banana', 'orange', 'mango', 'lemon', 'apple']
fruits.append('lime')   # ['banana', 'orange', 'mango', 'lemon', 'apple', 'lime']
print(fruits)
lst.append(fruits)
```

```
['banana', 'orange', 'mango', 'lemon', 'apple']
['banana', 'orange', 'mango', 'lemon', 'apple', 'lime']
```

In [164]:

```
#Inserting Items into a List
#Use insert() method to insert a single item at a specified index in a list. Note that
 other items are shifted to the right.

# syntax
lst = ['item1', 'item2']
#lst.insert(index, item)
fruits = ['banana', 'orange', 'mango', 'lemon']
fruits.insert(2, 'apple') # insert apple between orange and mango
print(fruits)           # ['banana', 'orange', 'apple', 'mango', 'lemon']
fruits.insert(3, 'lime')   # ['banana', 'orange', 'apple', 'lime', 'mango', 'lemon']
print(fruits)
```

```
['banana', 'orange', 'apple', 'mango', 'lemon']
['banana', 'orange', 'apple', 'lime', 'mango', 'lemon']
```

In [165]:

```
#Removing Items from a List
#The remove method removes a specified item from a list

# syntax
lst = ['item1', 'item2']
#lst.remove(item)
fruits = ['banana', 'orange', 'mango', 'lemon', 'banana']
fruits.remove('banana')
print(fruits)  # ['orange', 'mango', 'lemon', 'banana'] - this method removes the first
occurence of the item in the list
fruits.remove('lemon')
print(fruits)  # ['orange', 'mango', 'banana']
```

```
['orange', 'mango', 'lemon', 'banana']
['orange', 'mango', 'banana']
```

In [166]:

```
#Removing Items Using Pop
#The pop() method removes the specified index, (or the last item if index is not specif
ied):

# syntax
lst = ['item1', 'item2']
lst.pop()       # last item
#lst.pop(index)
fruits = ['banana', 'orange', 'mango', 'lemon']
fruits.pop()
print(fruits)       # ['banana', 'orange', 'mango']

fruits.pop(0)
print(fruits)       # ['orange', 'mango']
```

```
['banana', 'orange', 'mango']
['orange', 'mango']
```

In [167]:

```
#Removing Items Using Del
#The del keyword removes the specified index and it can also be used to delete items wi
thin index range. It can also delete the #list completely

# syntax
lst = ['item1', 'item2']
#del lst[index] # only a single item
del lst        # to delete the list completely
fruits = ['banana', 'orange', 'mango', 'lemon', 'kiwi', 'lime']
del fruits[0]
print(fruits)       # ['orange', 'mango', 'lemon', 'kiwi', 'lime']
del fruits[1]
print(fruits)       # ['orange', 'lemon', 'kiwi', 'lime']
del fruits[1:3]     # this deletes items between given indexes, so it does not delete t
he item with index 3!
print(fruits)       # ['orange', 'lime']
del fruits
print(fruits)       # This should give: NameError: name 'fruits' is not defined
```

```
['orange', 'mango', 'lemon', 'kiwi', 'lime']
['orange', 'lemon', 'kiwi', 'lime']
['orange', 'lime']

---------------------------------------------------------------------------
-
NameError                                 Traceback (most recent call las
t)
<ipython-input-167-a444a38c3a7a> in <module>
     14 print(fruits)       # ['orange', 'lime']
     15 del fruits
---> 16 print(fruits)       # This should give: NameError: name 'fruits' i
s not defined

NameError: name 'fruits' is not defined
```

In [168]:

```
#Clearing List Items
#The clear() method empties the list:

# syntax
lst = ['item1', 'item2']
lst.clear()
fruits = ['banana', 'orange', 'mango', 'lemon']
fruits.clear()
print(fruits)       # []
```

```
[]
```

In [169]:

```
#Copying a List
#It is possible to copy a list by reassigning it to a new variable in the following wa
y: list2 = list1. Now, list2 is a #reference of list1, any changes we make in list2 wil
l also modify the original, list2. But there are lots of case in which we #do not like
 to modify the original instead we like to have a different copy. One of way avoid the
 problem above is using #copy().

# syntax
lst = ['item1', 'item2']
lst_copy = lst.copy()
fruits = ['banana', 'orange', 'mango', 'lemon']
fruits_copy = fruits.copy()
print(fruits_copy)       # ['banana', 'orange', 'mango', 'lemon']
```

```
['banana', 'orange', 'mango', 'lemon']
```

In [171]:

```
#Joining Lists
#There are several ways to join, or concatenate, two or more lists in Python.

#Plus Operator (+)
# syntax
#list3 = list1 + list2
positive_numbers = [1, 2, 3, 4, 5]
zero = [0]
negative_numbers = [-5,-4,-3,-2,-1]
integers = negative_numbers + zero + positive_numbers
print(integers)
fruits = ['banana', 'orange', 'mango', 'lemon']
vegetables = ['Tomato', 'Potato', 'Cabbage', 'Onion', 'Carrot']
fruits_and_vegetables = fruits + vegetables
print(fruits_and_vegetables )
```

```
[-5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5]
['banana', 'orange', 'mango', 'lemon', 'Tomato', 'Potato', 'Cabbage', 'Oni
on', 'Carrot']
```

In [172]:

```
#Joining using extend() method

# syntax
list1 = ['item1', 'item2']
list2 = ['item3', 'item4', 'item5']
list1.extend(list2)
num1 = [0, 1, 2, 3]
num2= [4, 5,6]
num1.extend(num2)
print('Numbers:', num1)
negative_numbers = [-5,-4,-3,-2,-1]
positive_numbers = [1, 2, 3,4,5]
zero = [0]

negative_numbers.extend(zero)
negative_numbers.extend(positive_numbers)
print('Integers:', negative_numbers)
fruits = ['banana', 'orange', 'mango', 'lemon']
vegetables = ['Tomato', 'Potato', 'Cabbage', 'Onion', 'Carrot']
fruits.extend(vegetables)
print('Fruits and vegetables:', fruits )
```

```
Numbers: [0, 1, 2, 3, 4, 5, 6]
Integers: [-5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5]
Fruits and vegetables: ['banana', 'orange', 'mango', 'lemon', 'Tomato', 'P
otato', 'Cabbage', 'Onion', 'Carrot']
```

In [174]:

```
#Counting Items in a List
#The count() method returns the number of times an item appears in a list:

# syntax
lst = ['item1', 'item2']
#lst.count(item)
fruits = ['banana', 'orange', 'mango', 'lemon']
print(fruits.count('orange'))   # 1
ages = [22, 19, 24, 25, 26, 24, 25, 24]
print(ages.count(24))           # 3
```

```
1
3
```

In [175]:

```
#Finding Index of an Item
#The index() method returns the index of an item in the list:

# syntax
lst = ['item1', 'item2']
#lst.index(item)
fruits = ['banana', 'orange', 'mango', 'lemon']
print(fruits.index('orange'))   # 1
ages = [22, 19, 24, 25, 26, 24, 25, 24]
print(ages.index(24))           # 2, the first occurrence
```

```
1
2
```

In [176]:

```
#Reversing a List
#The reverse() method reverses the order of a list.

# syntax
lst = ['item1', 'item2']
lst.reverse()
fruits = ['banana', 'orange', 'mango', 'lemon']
fruits.reverse()
print(fruits.reverse())
ages = [22, 19, 24, 25, 26, 24, 25, 24]
ages.reverse()
print(ages.reverse())
['lemon', 'mango', 'orange', 'banana']
[24, 25, 24, 26, 25, 24, 19, 22]
```

None
None

Out[176]:

```
[24, 25, 24, 26, 25, 24, 19, 22]
```

In [177]:

```
#Sorting List Items
#To sort lists we can use sort() method or sorted() built-in functions. The sort() meth
od reorders the list items in ascending #order and modifies the original list. If an ar
gument of sort() method reverse is equal to true, it will arrange the list in #descendi
ng order.

#sort(): this method modifies the original list

# syntax
lst = ['item1', 'item2']
lst.sort()                # ascending
lst.sort(reverse=True)    # descending
```

In [ ]:

```
#Example:

fruits = ['banana', 'orange', 'mango', 'lemon']
fruits.sort()
print(fruits)             # sorted in alphabetical order
fruits.sort(reverse=True)
print(fruits)
ages = [22, 19, 24, 25, 26, 24, 25, 24]
ages.sort()
print(ages)
ages.sort(reverse=True)
print(ages)
['banana', 'lemon', 'mango', 'orange']
['orange', 'mango', 'lemon', 'banana']
[19, 22, 24, 24, 24, 25, 25, 26]
[26, 25, 25, 24, 24, 24, 22, 19]
```

In [178]:

```
#sorted(): returns the ordered list without modifying the original Example:

fruits = ['banana', 'orange', 'mango', 'lemon']
print(sorted(fruits))     # ['banana', 'lemon', 'mango', 'orange']
# Reverse order
fruits = ['banana', 'orange', 'mango', 'lemon']
fruits = sorted(fruits,reverse=True)
print(fruits)     # ['orange', 'mango', 'lemon', 'banana']
```

```
['banana', 'lemon', 'mango', 'orange']
['orange', 'mango', 'lemon', 'banana']
```

Declare an empty list

Declare a list with more than 5 items

Find the length of your list

Get the first item, the middle item and the last item of the list

Declare a list called mixed_data_types, put your(name, age, height, marital status, address)

Declare a list variable named it_companies and assign initial values Facebook, Google, Microsoft, Apple, IBM, Oracle and
Amazon.

Print the list using print()

Print the number of companies in the list

Print the first, middle and last company

Print the list after modifying one of the companies

Add an IT company to it_companies

Insert an IT company in the middle of the companies list

Change one of the it_companies names to uppercase (IBM excluded!)

Join the it_companies with a string '#; '

Check if a certain company exists in the it_companies list.

Sort the list using sort() method

Reverse the list in descending order using reverse() method

Slice out the first 3 companies from the list

Slice out the last 3 companies from the list

Slice out the middle IT company or companies from the list

Remove the first IT company from the list

Remove the middle IT company or companies from the list

Remove the last IT company from the list

Remove all IT companies from the list

Destroy the IT companies list

Join the following lists:

front_end = ['HTML', 'CSS', 'JS', 'React', 'Redux']
back_end = ['Node','Express', 'MongoDB']
After joining the lists in question 26. Copy the joined list and assign it to a variable full_stack. Then insert Python and
SQL after Redux.

The following is a list of 10 students ages:

ages = [19, 22, 19, 24, 20, 25, 26, 24, 25, 24]
Sort the list and find the min and max age
Add the min age and the max age again to the list
Find the median age (one middle item or two middle items divided by two)
Find the average age (sum of all items divided by their number)
Find the range of the ages (max minus min)
Compare the value of (min - average) and (max - average), use abs() method
Find the middle country(ies) in the countries list
Divide the countries list into two equal lists if it is even if not one more country for the first half.
['China', 'Russia', 'USA', 'Finland', 'Sweden', 'Norway', 'Denmark']. Unpack the first three countries and the rest as
scandic countries

In [7]:

```
l=["sd",1,3.5,[1,2,3],(1,2,3),{"a":2}]
```

In [8]:

```
for i in l:
    print(type(i))
```

```
<class 'str'>
<class 'int'>
<class 'float'>
<class 'list'>
<class 'tuple'>
<class 'dict'>
```

In [12]:

```
a=[0,0,0,0,0,0,0,0,0,0]
for i in range(10):
    a[i]=i
print(a)
```

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

In [16]:

```
al=["sd",1,3.5,[1,2,3],(1,2,3),{"a":2}]
for i in range(len(al)):
    print(al[i])
```

```
sd
1
3.5
[1, 2, 3]
(1, 2, 3)
{'a': 2}
```

In [21]:

```
a="string"
a[-2] #negative indexing
a[2] #positive
```

Out[21]:

```
'r'
```

In [50]:

```
k=["orange","apple","mango"]
for i in range(len(k)):
    if k[i][-1]=="e":
        print(k[i])
```

```
orange
apple
```

In [34]:

```
al
```

Out[34]:

```
['sd', 1, 3.5, [1, 2, 3], (1, 2, 3), {'a': 2}]
```

In [51]:

```
c=al
```

In [68]:

```
a,b,c,*r=al
kl=(a,b,c)
```

```
---------------------------------------------------------------------------
-
TypeError                                 Traceback (most recent call las
t)
<ipython-input-68-7aeb86e1e348> in <module>
      1 a,b,c,*r=al
----> 2 kl=list(a,b,c)

TypeError: 'list' object is not callable
```

In [67]:

```
print(a,b,c,r,kl)
```

```
sd 1 3.5 [[1, 2, 3], (1, 2, 3), {'a': 2}] ('sd', 1, 3.5)
```

In [65]:

```
al
```

Out[65]:

```
['sd', 1, 3.5, [1, 2, 3], (1, 2, 3), {'a': 2}]
```

In [76]:

```
s="string"
s[::-1]
```

Out[76]:

```
'gnirts'
```

In [83]:

```
#slicing
al[1:len(al):2]
```

Out[83]:

```
[1, [1, 2, 3], {'a': 2}]
```

In [85]:

```
"123456789"
"1234(5678)9"
1+2+3+4+9
print(8765+19)
```

```
8784
```

In [118]:

```
a="123456789123"
b=[]
for i in a:
    b.append(int(i))
print(b)
k=sum(b[:a.find("5")])+sum(b[a.find("8")+1:])
c=a[a.find("5"):a.find("8")+1]
print(int(c[::-1])+k)
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3]
8790
```

In [100]:

```
a.find("5")
```

Out[100]:

```
4
```

In [107]:

```
print(b)
```

```
---------------------------------------------------------------------------
-
AttributeError                            Traceback (most recent call las
t)
<ipython-input-107-d845036cdb08> in <module>
----> 1 b=list(map(int,a[:4].sep()))
      2 print(b)

AttributeError: 'str' object has no attribute 'sep'
```

In [102]:

```
a.find("8")
```

Out[102]:

```
7
```

In [103]:

```
a[7+1:]
```

Out[103]:

```
'9123'
```

In [139]:

```
s="12345"
m=[]
for i in s:
    m.append(int(i))
print(m)
```

```
[1, 2, 3, 4, 5]
```

In [9]:

```
n=[1,2]
a=[5,6,7]
n.append([8])
n.extend(a)
n.insert(1,a)
```

In [10]:

```
n
```

Out[10]:

```
[1, [5, 6, 7], 2, [8], 5, 6, 7]
```

In [150]:

```
m
```

Out[150]:

```
[1, 2, 3, 4, 5, [1, 2], 8, 9, 0, 8, 8]
```

In [155]:

```
m.insert(len(m),9)
```

In [156]:

```
m
```

Out[156]:

```
[9, 1, 2, 3, 4, 5, [1, 2], 8, 9, 0, 0, 8, 8, 9]
```

In [17]:

```
n=[1,2,3,4]
n.pop()
#print(x)
print(n)
a=4
print(a)
a=5
print(a)
```

```
[1, 2, 3]
4
5
```

In [26]:

```
n=["{","[","("]
del n[1:]
print(n)
```

```
['{']
```

In [27]:

```
n=[1,2,3]
if 1 in n:
    print("True")
else:
    print("False")
```

```
True
```

In [33]:

```
n=["a","b","c"]
n.remove("d")
print(n)
```

```
---------------------------------------------------------------------------
-
ValueError                                Traceback (most recent call las
t)
<ipython-input-33-4aeb990d02de> in <module>
      1 n=["a","b","c"]
----> 2 n.remove("d")
      3 print(n)

ValueError: list.remove(x): x not in list
```

In [47]:

```
n=[1,2,3]
m=[1,2,3]
p=n
print(id(n))
print(id(m))
print(id(p))
p[0]=5
print(n)
print(m)
c=n.copy()
print(c)
print(id(c))
p[0]=6
print(c)
print(p,n,c)
```

```
2309648053832
2309649724040
2309648053832
[5, 2, 3]
[1, 2, 3]
[5, 2, 3]
2309648567560
[5, 2, 3]
[6, 2, 3] [6, 2, 3] [5, 2, 3]
```

In [53]:

```
v=n+m+[3]
print(v)
```

```
[6, 2, 3, 1, 2, 3, 3]
```

In [54]:

```
v.count(3)
```

Out[54]:

```
3
```

In [58]:

```
max(n)
min(n)
```

Out[58]:

```
2
```

In [60]:

```
import itertools
n=[1,2,3,4,5]
m=0
mn=1000
n.remove(min(n))
print(n)
```

```
[2, 3, 4, 5]
```

In [68]:

```
n.reverse()
```

In [69]:

```
print(n)
```

```
[5, 4, 3, 2]
```

In [80]:

```
n.reverse()
```

In [81]:

```
n.sort()
```

Out[81]:

```
[5, 4, 3, 2]
```

In [84]:

```
c=sorted(n,reverse=False)
print(c)
print(n)
```

```
[2, 3, 4, 5]
[5, 4, 3, 2]
```

In [92]:

```
a=input()
print("{gfg} is a {0} science portal for {1}"
.format("computer", "geeks", "b",gfg=a))
```

```
1234
1234 is a computer science portal for geeks
```

In [100]:

```
print ("This site is {0:c}% securely {1}!!".
                    format(70, "encrypted"))
```

```
This site is F% securely encrypted!!
```

In [107]:

```
print("{0:50} was founded in {1:10}!"
    .format("GeeksforGeeks", 2009))
```

```
GeeksforGeeks                                      was founded in       20
09!
```

In [110]:

```
print("{0:^160} was founded in {1:<40}!"
        .format("GeeksforGeeks", 2009))
```

```
                                                                                                                G
eeksforGeeks
was founded in 2009                                    !
```

In [116]:

```
print("{:#>20s}".format("Geeks"))
```

```
################Geeks
```

In [121]:

```
l=[]
for i in range(10):
    l.append([i])
print(l)
```

```
[[0], [1], [2], [3], [4], [5], [6], [7], [8], [9]]
```

In [133]:

```
l=[[i,j] for i in range(10) for j in range(3)]
```

In [142]:

```
l=[{i:j} for i in range(10) for j in range(3) if i==j]
```

In [143]:

```
print(l)
```

```
[{0: 0}, {1: 1}, {2: 2}]
```

In [136]:

```
l=[]
for i in range(10):
    for j in range(3):
        if i==j:
            l.append([i,j])
```

In [137]:

```
print(l)
```

```
[[0, 0], [1, 1], [2, 2]]
```

In [ ]:

```

```