In [ ]:

```
# List Comprehension
# newlist=(expression(x) for x in givenlist if condition)
```

In [128…

```
# Normal trational way with for loop
nlist=list()
for i in l1:
    nlist.append(i*i)
print(nlist)
```

```
[9, 25, 36, 49]
```

In [129…

```
nlist=list(i**2 for i in l1)
print(l1)
```

```
[3, 5, 6, 7]
```

In [133…

```
# Show length of each string in given list
l1=["fdhf","Xfh","erjgd","Setrusf"]
nlist=list(len(i) for i in l1)
print(nlist)
```

```
[4, 3, 5, 7]
```

In [134…

```
# Show all numbers divisible by 5 from first 50 numbers
nlist=list(i for i in range(0,51) if i%5==0)
print(nlist)
```

```
[0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50]
```

In [27]:

```
# Show sum of all digits in the string Eg,"24th Jan 2022"
str1="24th Jan 2022"
sum=0
for x in str1:
    if x.isdigit():
        sum=sum+int(x)
print(sum)
```

```
12
```

In [152…

```
# Dictionary Comprehension
# ndict={k:v for(k,v) in iterable}

citylist=["Pune","Mumbai","Pune","Nashik"]
d1={k:len(k) for k in citylist}
print(d1)
cityoccurance={k:citylist.count(k) for k in citylist}
print(cityoccurance)
```

```
{'Pune': 4, 'Mumbai': 6, 'Nashik': 6}
{'Pune': 2, 'Mumbai': 1, 'Nashik': 1}
```

In [145…

```
dict1={"Book":400,"Pen":80,"Pencil":10,"Scale":5}
dict2={k:v for (k,v) in zip(dict1.keys(),dict1.values())}
print(dict2)
```

```
{'Book': 400, 'Pen': 80, 'Pencil': 10, 'Scale': 5}
```

In [2]:

```
citylist=["Pune","Mumbai","Pune","Nashik"]
cityoccurance1={k:v for (k,v) in zip(citylist,list(citylist.count(i) for i in citylist ))}
print(cityoccurance1)
```

```
{'Pune': 2, 'Mumbai': 1, 'Nashik': 1}
```

In [156…

```
import math as m
```

In [161…

```
print(m.cos(35.87))
```

```
-0.25545231974812554
```

In [164…

```
print(m.factorial(4))
```

```
24
```

In [165…

```
print(m.floor(56.78))
```

```
56
```

In [166…

```
print(m.ceil(56.78))
```

```
57
```

In [167…

```
import datetime as d
```

In [181…

```
print(d.date(2022,1,24))
```

```
2022-01-24
```

In [185…

```
d1=d.date(2022,1,24)
d1.day
```

Out[185…

```
24
```

In [186…

```
d1.month
```

Out[186…

```
1
```

In [187…

```
d1.year
```

Out[187…

```
2022
```

In [194…

```
import datetime as dt
print(dt.datetime.now())
print(dt.datetime.today())
```

```
2022-01-24 18:54:47.076222
2022-01-24 18:54:47.077215
```

In [9]:

```
# Assignment
```

In [6]:

```
#1 Find all of the numbers from 1-1000 that are divisible by 7
nlist=list(i for i in range(1,1001) if i%7==0)
print("The numbers from 1-1000 that are divisible by 7 :",nlist)
```

```
The numbers from 1-1000 that are divisible by 7 : [7, 14, 21, 28, 35, 42, 49, 56, 63, 70, 77, 84, 91, 98, 105, 112, 119, 126, 133, 140, 147, 154, 161, 168, 17
5, 182, 189, 196, 203, 210, 217, 224, 231, 238, 245, 252, 259, 266, 273, 280, 287, 294, 301, 308, 315, 322, 329, 336, 343, 350, 357, 364, 371, 378, 385, 392,
399, 406, 413, 420, 427, 434, 441, 448, 455, 462, 469, 476, 483, 490, 497, 504, 511, 518, 525, 532, 539, 546, 553, 560, 567, 574, 581, 588, 595, 602, 609, 61
6, 623, 630, 637, 644, 651, 658, 665, 672, 679, 686, 693, 700, 707, 714, 721, 728, 735, 742, 749, 756, 763, 770, 777, 784, 791, 798, 805, 812, 819, 826, 833,
840, 847, 854, 861, 868, 875, 882, 889, 896, 903, 910, 917, 924, 931, 938, 945, 952, 959, 966, 973, 980, 987, 994]
```

In [9]:

```
#2 Find all of the numbers from 1-1000 that have a 3 in them
list1 = [n for n in range(0,1000) if '3' in str(n)]
print("The numbers from 1-1000 that have a 3 in them :",list1)
```

```
The numbers from 1-1000 that have a 3 in them : [3, 13, 23, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 43, 53, 63, 73, 83, 93, 103, 113, 123, 130, 131, 132, 133,
134, 135, 136, 137, 138, 139, 143, 153, 163, 173, 183, 193, 203, 213, 223, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 243, 253, 263, 273, 283, 293, 30
0, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331,
332, 333, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362, 36
3, 364, 365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375, 376, 377, 378, 379, 380, 381, 382, 383, 384, 385, 386, 387, 388, 389, 390, 391, 392, 393, 394,
395, 396, 397, 398, 399, 403, 413, 423, 430, 431, 432, 433, 434, 435, 436, 437, 438, 439, 443, 453, 463, 473, 483, 493, 503, 513, 523, 530, 531, 532, 533, 53
4, 535, 536, 537, 538, 539, 543, 553, 563, 573, 583, 593, 603, 613, 623, 630, 631, 632, 633, 634, 635, 636, 637, 638, 639, 643, 653, 663, 673, 683, 693, 703,
713, 723, 730, 731, 732, 733, 734, 735, 736, 737, 738, 739, 743, 753, 763, 773, 783, 793, 803, 813, 823, 830, 831, 832, 833, 834, 835, 836, 837, 838, 839, 84
3, 853, 863, 873, 883, 893, 903, 913, 923, 930, 931, 932, 933, 934, 935, 936, 937, 938, 939, 943, 953, 963, 973, 983, 993]
```

In [12]:

```
#3 Count the number of spaces in a string
str1 = "Have a nice day"
print("String :",str1)
spaces = [s for s in str1 if s == ' ']
print("The number of spaces in a string :",len(spaces))
```

```
String : Have a nice day
The number of spaces in a string : 3
```

In [17]:

```
#4 Create a list of all the consonants in the string “Yellow Yaks like yelling and yawning and yesturday they
# yodled while eating yuky yams”
string = "Yellow Yaks like yelling and yawning and yesturday they yodled while eating yuky yams"
print("Given string :",string)
list1 = [i for i in string if i not in 'a,e,i,o,u, " "']
print("All the consonants in the string :",list1)
```

```
Given string : Yellow Yaks like yelling and yawning and yesturday they yodled while eating yuky yams
All the consonants in the string : ['Y', 'l', 'l', 'w', 'Y', 'k', 's', 'l', 'k', 'y', 'l', 'l', 'n', 'g', 'n', 'd', 'y', 'w', 'n', 'n', 'g', 'n', 'd', 'y',
's', 't', 'r', 'd', 'y', 't', 'h', 'y', 'y', 'd', 'l', 'd', 'w', 'h', 'l', 't', 'n', 'g', 'y', 'k', 'y', 'y', 'm', 's']
```

In [41]:

```
#5 Get the index and the value as a tuple for items in the list “hi”, 4, 8.99, ‘apple’, (‘t,b’,’n’).
# Result would look like (index, value), (index, value)
list1 = ["hi", 4, 8.99, 'apple', ('t,b','n')]
print("Given list :",list1)
list2 = [(index, list1) for index, list1 in enumerate(list1)]
print("Index and value as a tuple :",list2)
```

```
Given list : ['hi', 4, 8.99, 'apple', ('t,b', 'n')]
Index and value as a tuple : [(0, 'hi'), (1, 4), (2, 8.99), (3, 'apple'), (4, ('t,b', 'n'))]
```

In [22]:

```
#6 Find the common numbers in two lists (without using a tuple or set) list_a = 1, 2, 3, 4, list_b = 2, 3, 4, 5
list_a = [1, 2, 3, 4]
list_b = [2, 3, 4, 5]
common = [i for i in list_a if i in list_b]
print("Common numbers in two lists :",common)
```

```
Common numbers in two lists : [2, 3, 4]
```

In [27]:

```
#7 Get only the numbers in a sentence like ‘In 1984 there were 13 instances of a protest with over 1000 people attending’
sentence = "In 1984 there were 13 instances of a protest with over 1000 people attending"
words = sentence.split()
nlist = [i for i in words if not i.isalpha()]
print("Only the numbers in a sentence :",nlist)
```

```
Only the numbers in a sentence : ['1984', '13', '1000']
```

In [42]:

```
#8 Given numbers = range(20), produce a list containing the word ‘even’ if a number in the numbers is even, and the word
#‘odd’ if the number is odd. Result would look like ‘odd’,’odd’, ‘even’
list1 = ['even' if i%2 == 0 else 'odd' for i in range(20)]
print("List containing the word even and odd :",list1)
```

```
List containing the word even and odd : ['even', 'odd', 'even', 'odd', 'even', 'odd', 'even', 'odd', 'even', 'odd', 'even', 'odd', 'even', 'odd', 'even', 'od
d', 'even', 'odd', 'even', 'odd']
```

In [40]:

```
#9 Produce a list of tuples consisting of only the matching numbers in these lists list_a = 1, 2, 3,4,5,6,7,8,9,
# list_b = 2, 7, 1, 12. Result would look like (4,4), (12,12)
list_a = [1, 2, 3, 4, 5, 6, 7, 8, 9]
list_b = [2, 7, 1, 12]
listn = [(a, b) for a in list_a for b in list_b if a == b]
print("Matching numbers :",listn)
```

```
Matching numbers : [(1, 1), (2, 2), (7, 7)]
```

In [2]:

```
#10 Find all of the words in a string that are less than 4 letters
string = "Find all of the words in a string that are less than 4 letters"
e = string.split()
list1 = [i for i in e if len(i)>=4]
print("String that are less than 4 letters :",list1)
```

```
String that are less than 4 letters : ['Find', 'words', 'string', 'that', 'less', 'than', 'letters']
```

In [1]:

```
#11 Use a nested list comprehension to find all of the numbers from 1-1000 that are divisible by any single digit
# besides 1 (2-9)
listn= [i for i in range(1,1001) if True in [True for x in range(1,10) if i % x == 0]]
print("All the numbers from 1-1000 that are divisible by any single digit :",listn)
```

```
All the numbers from 1-1000 that are divisible by any single digit : [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 2
5, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64,
65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 10
3, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134,
135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 16
6, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197,
198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 22
9, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260,
261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 29
2, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323,
324, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 35
5, 356, 357, 358, 359, 360, 361, 362, 363, 364, 365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375, 376, 377, 378, 379, 380, 381, 382, 383, 384, 385, 386,
387, 388, 389, 390, 391, 392, 393, 394, 395, 396, 397, 398, 399, 400, 401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 41
8, 419, 420, 421, 422, 423, 424, 425, 426, 427, 428, 429, 430, 431, 432, 433, 434, 435, 436, 437, 438, 439, 440, 441, 442, 443, 444, 445, 446, 447, 448, 449,
450, 451, 452, 453, 454, 455, 456, 457, 458, 459, 460, 461, 462, 463, 464, 465, 466, 467, 468, 469, 470, 471, 472, 473, 474, 475, 476, 477, 478, 479, 480, 48
1, 482, 483, 484, 485, 486, 487, 488, 489, 490, 491, 492, 493, 494, 495, 496, 497, 498, 499, 500, 501, 502, 503, 504, 505, 506, 507, 508, 509, 510, 511, 512,
513, 514, 515, 516, 517, 518, 519, 520, 521, 522, 523, 524, 525, 526, 527, 528, 529, 530, 531, 532, 533, 534, 535, 536, 537, 538, 539, 540, 541, 542, 543, 54
4, 545, 546, 547, 548, 549, 550, 551, 552, 553, 554, 555, 556, 557, 558, 559, 560, 561, 562, 563, 564, 565, 566, 567, 568, 569, 570, 571, 572, 573, 574, 575,
576, 577, 578, 579, 580, 581, 582, 583, 584, 585, 586, 587, 588, 589, 590, 591, 592, 593, 594, 595, 596, 597, 598, 599, 600, 601, 602, 603, 604, 605, 606, 60
7, 608, 609, 610, 611, 612, 613, 614, 615, 616, 617, 618, 619, 620, 621, 622, 623, 624, 625, 626, 627, 628, 629, 630, 631, 632, 633, 634, 635, 636, 637, 638,
639, 640, 641, 642, 643, 644, 645, 646, 647, 648, 649, 650, 651, 652, 653, 654, 655, 656, 657, 658, 659, 660, 661, 662, 663, 664, 665, 666, 667, 668, 669, 67
0, 671, 672, 673, 674, 675, 676, 677, 678, 679, 680, 681, 682, 683, 684, 685, 686, 687, 688, 689, 690, 691, 692, 693, 694, 695, 696, 697, 698, 699, 700, 701,
702, 703, 704, 705, 706, 707, 708, 709, 710, 711, 712, 713, 714, 715, 716, 717, 718, 719, 720, 721, 722, 723, 724, 725, 726, 727, 728, 729, 730, 731, 732, 73
3, 734, 735, 736, 737, 738, 739, 740, 741, 742, 743, 744, 745, 746, 747, 748, 749, 750, 751, 752, 753, 754, 755, 756, 757, 758, 759, 760, 761, 762, 763, 764,
765, 766, 767, 768, 769, 770, 771, 772, 773, 774, 775, 776, 777, 778, 779, 780, 781, 782, 783, 784, 785, 786, 787, 788, 789, 790, 791, 792, 793, 794, 795, 79
6, 797, 798, 799, 800, 801, 802, 803, 804, 805, 806, 807, 808, 809, 810, 811, 812, 813, 814, 815, 816, 817, 818, 819, 820, 821, 822, 823, 824, 825, 826, 827,
828, 829, 830, 831, 832, 833, 834, 835, 836, 837, 838, 839, 840, 841, 842, 843, 844, 845, 846, 847, 848, 849, 850, 851, 852, 853, 854, 855, 856, 857, 858, 85
9, 860, 861, 862, 863, 864, 865, 866, 867, 868, 869, 870, 871, 872, 873, 874, 875, 876, 877, 878, 879, 880, 881, 882, 883, 884, 885, 886, 887, 888, 889, 890,
891, 892, 893, 894, 895, 896, 897, 898, 899, 900, 901, 902, 903, 904, 905, 906, 907, 908, 909, 910, 911, 912, 913, 914, 915, 916, 917, 918, 919, 920, 921, 92
2, 923, 924, 925, 926, 927, 928, 929, 930, 931, 932, 933, 934, 935, 936, 937, 938, 939, 940, 941, 942, 943, 944, 945, 946, 947, 948, 949, 950, 951, 952, 953,
954, 955, 956, 957, 958, 959, 960, 961, 962, 963, 964, 965, 966, 967, 968, 969, 970, 971, 972, 973, 974, 975, 976, 977, 978, 979, 980, 981, 982, 983, 984, 98
5, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000]
```

In [ ]: