

```
In [2]: import numpy as np
import pandas as pd
```

## Find all of the numbers from 1–1000 that have a 6 in them

```
In [3]: l=[i for i in range(1,1000) if str(i).find('6')>=0]
print(l)
```

```
[6, 16, 26, 36, 46, 56, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 76, 86, 96, 106, 116, 126, 136, 146, 156, 160, 16
1, 162, 163, 164, 165, 166, 167, 168, 169, 176, 186, 196, 206, 216, 226, 236, 246, 256, 260, 261, 262, 263, 264,
265, 266, 267, 268, 269, 276, 286, 296, 306, 316, 326, 336, 346, 356, 360, 361, 362, 363, 364, 365, 366, 367, 36
8, 369, 376, 386, 396, 406, 416, 426, 436, 446, 456, 460, 461, 462, 463, 464, 465, 466, 467, 468, 469, 476, 486,
496, 506, 516, 526, 536, 546, 556, 560, 561, 562, 563, 564, 565, 566, 567, 568, 569, 576, 586, 596, 600, 601, 60
2, 603, 604, 605, 606, 607, 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, 64
7, 648, 649, 650, 651, 652, 653, 654, 655, 656, 657, 658, 659, 660, 661, 662, 663, 664, 665, 666, 667, 668, 669,
670, 671, 672, 673, 674, 675, 676, 677, 678, 679, 680, 681, 682, 683, 684, 685, 686, 687, 688, 689, 690, 691, 69
2, 693, 694, 695, 696, 697, 698, 699, 706, 716, 726, 736, 746, 756, 760, 761, 762, 763, 764, 765, 766, 767, 768,
769, 776, 786, 796, 806, 816, 826, 836, 846, 856, 860, 861, 862, 863, 864, 865, 866, 867, 868, 869, 876, 886, 89
6, 906, 916, 926, 936, 946, 956, 960, 961, 962, 963, 964, 965, 966, 967, 968, 969, 976, 986, 996]
```

## Find all of the numbers from 1–1000 that are divisible by 8

```
In [5]: l = [i for i in range(1,1000) if i%8 == 0]
print(l)
```

```
[8, 16, 24, 32, 40, 48, 56, 64, 72, 80, 88, 96, 104, 112, 120, 128, 136, 144, 152, 160, 168, 176, 184, 192, 200,
208, 216, 224, 232, 240, 248, 256, 264, 272, 280, 288, 296, 304, 312, 320, 328, 336, 344, 352, 360, 368, 376, 38
4, 392, 400, 408, 416, 424, 432, 440, 448, 456, 464, 472, 480, 488, 496, 504, 512, 520, 528, 536, 544, 552, 560,
568, 576, 584, 592, 600, 608, 616, 624, 632, 640, 648, 656, 664, 672, 680, 688, 696, 704, 712, 720, 728, 736, 74
4, 752, 760, 768, 776, 784, 792, 800, 808, 816, 824, 832, 840, 848, 856, 864, 872, 880, 888, 896, 904, 912, 920,
928, 936, 944, 952, 960, 968, 976, 984, 992]
```

## Count the number of spaces in a string

```
In [9]: s = 'Practice Problems to Drill List Comprehension in Your Head.'
l = [i for i in s if i==' ']
len(l)
```

Out[9]: 8

## Remove all of the vowels in a string

```
In [12]: s = 'Practice Problems to Drill List Comprehension in Your Head.'
l = [i for i in s if i not in list('aeiou')]
''.join(l)
```

Out[12]: 'Prctc Prblms t Drll Lst Cmprhnsn n Yr Hd.'

## Find all of the words in a string that are less than 5 letters

```
In [16]: s = 'Practice Problems to Drill List Comprehension in Your Head.'
l = [i for i in s.split() if len(i)<5]
print(l)
```

```
['to', 'List', 'in', 'Your']
```

## Use a dictionary comprehension to count the length of each word in a sentence

In [19]:

```
s = 'Practice Problems to Drill List Comprehension in Your Head.'
d = {i:len(i) for i in s.split()}
print(d)
```

```
{'Practice': 8, 'Problems': 8, 'to': 2, 'Drill': 5, 'List': 4, 'Comprehension': 13, 'in': 2, 'Your': 4, 'Head.': 5}
```

## 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)

In [22]:

```
l = [i for i in range(1, 1000) for j in range(2, 10) if i%j==0]
print(set(l))
```

```
{2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 14, 15, 16, 18, 20, 21, 22, 24, 25, 26, 27, 28, 30, 32, 33, 34, 35, 36, 38, 39, 40, 42, 44, 45, 46, 48, 49, 50, 51, 52, 54, 55, 56, 57, 58, 60, 62, 63, 64, 65, 66, 68, 69, 70, 72, 74, 75, 76, 77, 78, 80, 81, 82, 84, 85, 86, 87, 88, 90, 91, 92, 93, 94, 95, 96, 98, 99, 100, 102, 104, 105, 106, 108, 110, 111, 112, 114, 115, 116, 117, 118, 119, 120, 122, 123, 124, 125, 126, 128, 129, 130, 132, 133, 134, 135, 136, 138, 140, 141, 142, 144, 145, 146, 147, 148, 150, 152, 153, 154, 155, 156, 158, 159, 160, 161, 162, 164, 165, 166, 168, 170, 171, 172, 174, 175, 176, 177, 178, 180, 182, 183, 184, 185, 186, 188, 189, 190, 192, 194, 195, 196, 198, 200, 201, 202, 203, 204, 205, 206, 207, 208, 210, 212, 213, 214, 215, 216, 217, 218, 219, 220, 222, 224, 225, 226, 228, 230, 231, 232, 234, 235, 236, 237, 238, 240, 242, 243, 244, 245, 246, 248, 249, 250, 252, 254, 255, 256, 258, 259, 260, 261, 262, 264, 265, 266, 267, 268, 270, 272, 273, 274, 275, 276, 278, 279, 280, 282, 284, 285, 288, 290, 291, 292, 294, 295, 296, 297, 298, 300, 301, 302, 303, 304, 305, 306, 308, 309, 310, 312, 314, 315, 316, 318, 320, 321, 322, 324, 325, 326, 327, 328, 329, 330, 332, 333, 334, 335, 336, 338, 339, 340, 342, 344, 345, 346, 348, 350, 351, 352, 354, 355, 356, 357, 358, 360, 362, 363, 364, 365, 366, 368, 369, 370, 371, 372, 374, 375, 376, 378, 380, 381, 382, 384, 385, 386, 387, 388, 390, 392, 393, 394, 395, 396, 398, 399, 400, 402, 404, 405, 406, 408, 410, 411, 412, 413, 414, 415, 416, 417, 418, 420, 422, 423, 424, 425, 426, 427, 428, 429, 430, 432, 434, 435, 436, 438, 440, 441, 442, 444, 445, 446, 447, 448, 450, 452, 453, 454, 455, 456, 458, 459, 460, 462, 464, 465, 466, 468, 469, 470, 471, 472, 474, 475, 476, 477, 478, 480, 482, 483, 484, 485, 486, 488, 489, 490, 492, 494, 495, 496, 497, 498, 500, 501, 502, 504, 505, 506, 507, 508, 510, 511, 512, 513, 514, 515, 516, 518, 519, 520, 522, 524, 525, 526, 528, 530, 531, 532, 534, 535, 536, 537, 538, 539, 540, 542, 543, 544, 545, 546, 548, 549, 550, 552, 553, 554, 555, 556, 558, 560, 561, 562, 564, 565, 566, 567, 568, 570, 572, 573, 574, 575, 576, 578, 579, 580, 581, 582, 584, 585, 586, 588, 590, 591, 592, 594, 595, 596, 597, 598, 600, 602, 603, 604, 605, 606, 608, 609, 610, 612, 614, 615, 616, 618, 620, 621, 622, 623, 624, 625, 626, 627, 628, 630, 632, 633, 634, 636, 637, 638, 639, 640, 642, 644, 645, 646, 648, 650, 651, 652, 654, 655, 656, 657, 658, 660, 662, 663, 664, 665, 666, 668, 669, 670, 672, 674, 675, 676, 678, 679, 680, 681, 682, 684, 685, 686, 687, 688, 690, 692, 693, 694, 695, 696, 698, 699, 700, 702, 704, 705, 706, 707, 708, 710, 711, 712, 714, 715, 716, 717, 718, 720, 721, 722, 723, 724, 725, 726, 728, 729, 730, 732, 734, 735, 736, 738, 740, 741, 742, 744, 745, 746, 747, 748, 749, 750, 752, 753, 754, 755, 756, 758, 759, 760, 762, 763, 764, 765, 766, 768, 770, 771, 772, 774, 775, 776, 777, 778, 780, 782, 783, 784, 785, 786, 788, 789, 790, 791, 792, 794, 795, 796, 798, 800, 801, 802, 804, 805, 806, 807, 808, 810, 812, 813, 814, 815, 816, 818, 819, 820, 822, 824, 825, 826, 828, 830, 831, 832, 833, 834, 835, 836, 837, 838, 840, 842, 843, 844, 845, 846, 847, 848, 849, 850, 852, 854, 855, 856, 858, 860, 861, 862, 864, 865, 866, 867, 868, 870, 872, 873, 874, 875, 876, 878, 879, 880, 882, 884, 885, 886, 888, 889, 890, 891, 892, 894, 895, 896, 897, 898, 899, 900, 902, 903, 904, 905, 906, 908, 909, 910, 912, 914, 915, 916, 917, 918, 920, 921, 922, 924, 925, 926, 928, 929, 930, 931, 932, 933, 934, 935, 936, 938, 939, 940, 942, 944, 945, 946, 948, 950, 951, 952, 954, 955, 956, 957, 958, 959, 960, 962, 963, 964, 965, 966, 968, 969, 970, 972, 973, 974, 975, 976, 978, 980, 981, 982, 984, 985, 986, 987, 988, 990, 992, 993, 994, 995, 996, 998, 999}
```

## To get list of car names whose price less than 5000 from given dict

In [26]:

```
Dict={"Sedan": 1500, "SUV": 2000, "Pickup": 2500, "Minivan": 1600, "Van": 2400, "Semi": 13600, "Bicycle": 7, "Motorcycle": 1200}

d = [i for i in Dict if Dict[i]<5000 ]
print(d)
```

```
['Sedan', 'SUV', 'Pickup', 'Minivan', 'Van', 'Bicycle', 'Motorcycle']
```