**Beautiful Idiomatic approach to solve programming problems**

**1)**

**for i in [0, 1, 2, 3, 4, 5]:  
print i\*\*2  
for i in range(6):  
print i\*\*2**

**for i in xrange(6):  
print i\*\*2  
2)  
colors = ['red', 'green', 'blue', 'yellow']  
for i in range(len(colors)):  
print colors[i]**

**for color in colors:  
print color  
3)  
colors = ['red', 'green', 'blue', 'yellow']  
for i in range(len(colors)-1, -1, -1):  
print colors[i]**

**for color in reversed(colors):  
print color  
4)  
colors = ['red', 'green', 'blue', 'yellow']  
for i in range(len(colors)):  
print i, '--->', colors[i]**

**for i, color in enumerate(colors):  
print i, '--->', color  
5)  
names = ['raymond', 'rachel', 'matthew']  
colors = ['red', 'green', 'blue', 'yellow']  
n = min(len(names), len(colors))  
for i in range(n):  
print names[i], '--->', colors[i]  
for name, color in zip(names, colors):  
print name, '--->', color**

**for name, color in izip(names, colors):  
print name, '--->', color  
6)  
colors = ['red', 'green', 'blue', 'yellow']  
def compare\_length(c1, c2):  
if len(c1) < len(c2): return -1  
if len(c1) > len(c2): return 1  
return 0  
print sorted(colors, cmp=compare\_length)**

**print sorted(colors, key=len)  
7)  
blocks = []  
while True:  
block = f.read(32)  
if block == '':  
break  
blocks.append(block)**

**blocks = []  
for block in iter(partial(f.read, 32), ''):  
blocks.append(block)  
8(  
def find(seq, target):  
found = False  
for i, value in enumerate(seq):  
if value == target:  
found = True  
break  
if not found:  
return -1  
return i**

**def find(seq, target):  
for i, value in enumerate(seq):  
if value == target:  
break  
else:  
return -1  
return i  
9)  
# Not very fast, has to re-hash every key and do a lookup  
for k in d:  
print k, '--->', d[k]  
# Makes a big huge list  
for k, v in d.items():  
print k, '--->', v**

**for k, v in d.iteritems():  
print k, '--->', v  
10)  
colors = ['red', 'green', 'red', 'blue', 'green', 'red']  
# Simple, basic way to count. A good start for beginners.  
d = {}  
for color in colors:  
if color not in d:  
d[color] = 0  
d[color] += 1  
# {'blue': 1, 'green': 2, 'red': 3}**

**for color in colors:  
d[color] = d.get(color, 0) + 1  
# Slightly more modern but has several caveats, better for advanced users  
# who understand the intricacies  
d = defaultdict(int)  
for color in colors:  
d[color] += 1**