

```
In [1]: for i in range(0, 5) :  
        print i
```

```
0  
1  
2  
3  
4
```

```
In [3]: r = range(0, 5)  
        print r, type(r)
```

```
[0, 1, 2, 3, 4] <type 'list'>
```

```
In [4]: class myRange :  
        def __init__(self, lo, hi, step=1) :  
            self.x = lo  
            self.hi = hi  
            self.step = step  
  
        def __iter__(self) :  
            return self  
  
        def next(self) :  
            if self.x < self.hi :  
                ret = self.x  
                self.x += self.step  
                return ret  
            else :  
                raise StopIteration()
```

```
In [5]: r1 = myRange(0, 5)  
        print type(r1)
```

```
<type 'instance'>
```

```
In [6]: for i in r1 :  
        print i
```

```
0  
1  
2  
3  
4
```

```
In [7]: r2 = xrange(0, 5)
print type(r2)
for i in r2 :
    print i
```

```
<type 'xrange'>
0
1
2
3
4
```

```
In [8]: def yieldRange(lo, hi, step=1) :
        x = lo
        while (x < hi) :
            yield x
            x += step
        return

for i in yieldRange(0, 5) :
    print i
```

```
0
1
2
3
4
```

```
In [23]: class findDist :
        def __init__(self, tstr, char) :
            self.string = tstr
            self.char = char
            self.pos = 0

        def __iter__(self) :
            return self

        def next(self) :
            delta = 0
            if (self.pos == len(self.string)) :
                raise StopIteration()
            while (self.string[self.pos] != self.char) :
                delta += 1
                self.pos += 1
            self.pos += 1
            return delta
```

```
In [24]: for i in findDist("abracadabra", 'a') :  
        print i
```

```
0  
2  
1  
1  
2
```

```
In [25]: def findDistYield(string, char) :  
        delta = 0  
        for c in string :  
            if c == char :  
                yield delta  
                delta = 0  
            else :  
                delta += 1
```

```
In [26]: for i in findDistYield("abracadabra", 'a') :  
        print i
```

```
0  
2  
1  
1  
2
```