

1.用生成器完成自定义range方法，可以完成系统range的所有功能

```
def my_range(start, end=None, step=1):
    if end is None: # 一个参数时
        start, end = 0, start
    count = start
    while True:
        b1 = count >= end and step > 0 # 正向取值
        b2 = count <= end and step < 0 # 反向取值
        if b1 or b2: # 满足正向反向
            break
        yield count
        count += step

for v in my_range(30, 20, -2):
    print(v)
```

2.用生成器完成自定义enumerate方法，也可以为可迭代对象提供索引支持

不判断，和系统一样，传非迭代对象抛异常

```
def my_enumerate(iterable, start=0):
    for v in iterable:
        yield (start, v)
        start += 1

# 判断，那么传入b1, b2以外的情况也不抛异常
def my_enumerate(iterable, start=0):
    # str, list, tuple, set, dict 五个基础
    b1 = isinstance(iterable, (str, list, tuple, set, dict))
    # 处理字典的keys, values, items
    b2 = type(iterable) in [type({}.keys()), type({}.values()), type({}.items())]
    if b1 or b2:
        for v in iterable:
            yield (start, v)
            start += 1
    else:
        yield '暂不能处理'

for v in my_enumerate({'a': 1}.items(), 100):
    print(v)
```

3.用生成器完成获取阶乘得方法，第一次next得到1的阶乘，第二次next得到2的阶乘，依次类推，直接for循环，可以依次得到指定范围内得所有阶乘，eg: factorial(10)，可以得到1~10之间的10个阶乘

```
def factorial(num):  
    pro = 1  
    for i in range(1, num + 1): # 完成累积  
        pro *= i  
        yield pro  
  
for v in factorial(5):  
    print(v)
```