

Iterator classes are a superset.

For example, one subset technique is the range function:

```
def generate_range(start, end, step):  
    while start < end:  
        yield start  
        start += step
```

This simple function yields every value from start to end incrementing by step. This could be translated to an iterator class like:

```
class Range:  
    def __init__(self, start, end, step):  
        self.start = start  
        self.end = end  
        self.step = step  
    def __iter__(self):  
        return self  
  
    def __next__(self):  
        if self.start < self.end:  
            result = self.start  
            self.start += self.step  
            return result  
        else:  
            raise StopIteration
```

In both of these techniques, one would need to create a generator/iterator by setting a variable as either the method or the range, and then one would need to continuously call next() on the object.