In [1]:
```
arr = [2, 6, 8, 10]
val = iter(arr)

i1 = next(val)
print(i1)

i2 = next(val)
print(i2)

i3 = next(val)
print(i3)
```

```
2
6
8
```

In [2]:
```
arr = [1, 3, 6, 9]
val = iter(arr)

while(True):
    try:
        item = next(val)
        print(item)
    except StopIteration:
        break
```

```
1
3
6
9
```

In [3]:
```
class arr_Even:
    def __init__(self, max):
        self.n = 2
        self.max = max

    def __iter__(self):
        return self

    def __next__(self):
        if(self.n <= self.max):
            result = self.n
            self.n +=2
            return result
        else:
            raise StopIteration
num = arr_Even(8)
print(next(num))
print(next(num))
print(next(num))
```

```
2
4
6
```

In [4]:
```
def even_gen():
    n=0

    n+=2
    yield n

    n+=2
    yield n

    n+=2
    yield n
num = even_gen()
print(next(num))
print(next(num))
print(next(num))
print(next(num))
```

```
2
4
6

---------------------------------------------------------------------------
StopIteration                             Traceback (most recent call last)
C:\Users\PRATIK~1\AppData\Local\Temp/ipykernel_17304/2705593751.py in <module>
     14 print(next(num))
     15 print(next(num))
---> 16 print(next(num))

StopIteration:
```

In [6]:
```
def even_gen(max):
    n = 2
    while(n<=max):
```

```
        yield n
        n+=2
num = even_gen(6)
print(next(num))
print(next(num))
print(next(num))
```

```
2
4
6
```

In [7]:
```
def gen_fib():
    n1 = 0
    n2 = 1
    while(True):
        yield n1
        n1, n2 = n2, n1+n2
seq = gen_fib()
print(next(seq))
print(next(seq))
print(next(seq))
print(next(seq))
print(next(seq))
print(next(seq))
```

```
0
1
1
2
3
5
```

In [ ]: