Adventures in Iterations

Vikrant Patil

Fun With Iterations

- Doing something repetatively
- Simple operations like shuffling of a deck
- Series calculations
- Pressing cos key on calculator repetatively
- Doing some tricks with digits of a number
- Doing some gemetric changes in a figure

Riffle Shuffle

```
deck       part1  part2     out-shuffle  in-shuffle

A1          A1                 A1          B1
A2                  B1         B1          A1
A3          A2                 A2          B2
A4                  B2    ===> B2          A2
B1          A3                 A3          B3
B2                  B3         B3          A3
B3          A4                 A4          B4
B4                  B4         B4          A4
```

Four Number Game

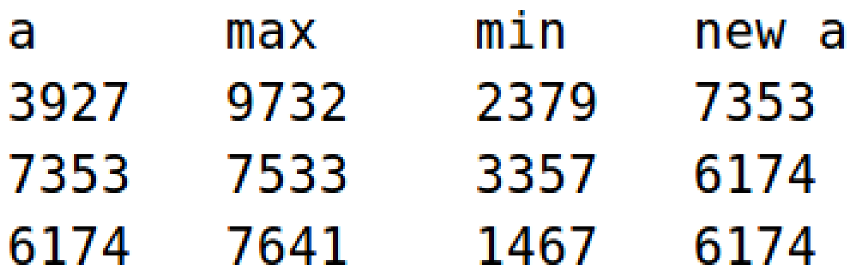
- Lets begin with a = 10, b=25, c=15, d=5
- Iteratively do following operation
  - a => |a-b| = |10-25| = 10
  - b => |b-c| = |25-15| = 10
  - c => |c-d| = |15-5| = 10
  - d => |d-a| = |5-10| = 5
- Obeserve what happens to a,b,c,d

Sum Of Squares(SSQ)

- Begin with a number a = 1234
- Find sum of squares of its digits (1 + 4 + 9 +16 = 30)
- Let this SSQ be new vale of a
- Repeat above procedure observe what happens to a

Kaprekar Iteration

Rearrage digits of a four digit number to form max and min number. Difference of max and min becomes new number

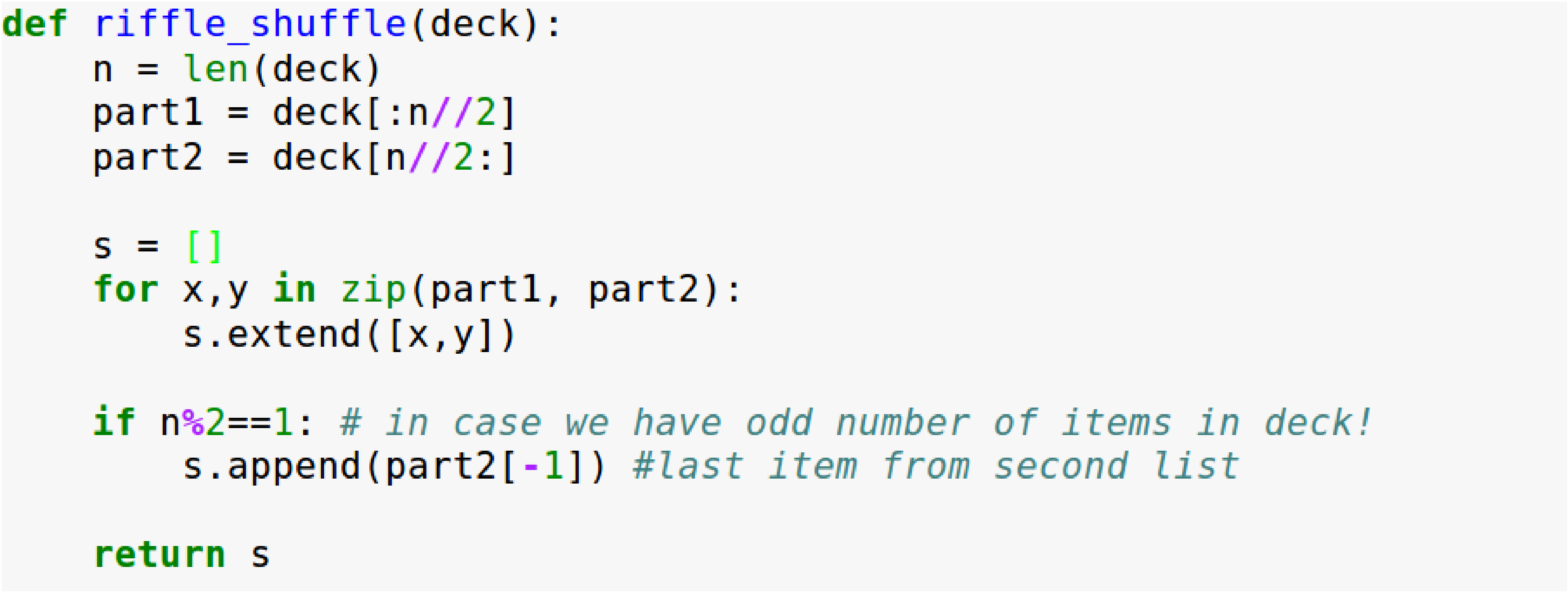


```
a       max      min      new a
3927    9732     2379     7353
7353    7533     3357     6174
6174    7641     1467     6174
```

Iterations in Python

- Lists – indexing, slicing
- Iteration Protocol – next()
- for loops
- List comprehensions
- Iteration patterns

Lets riffle-shuffle!



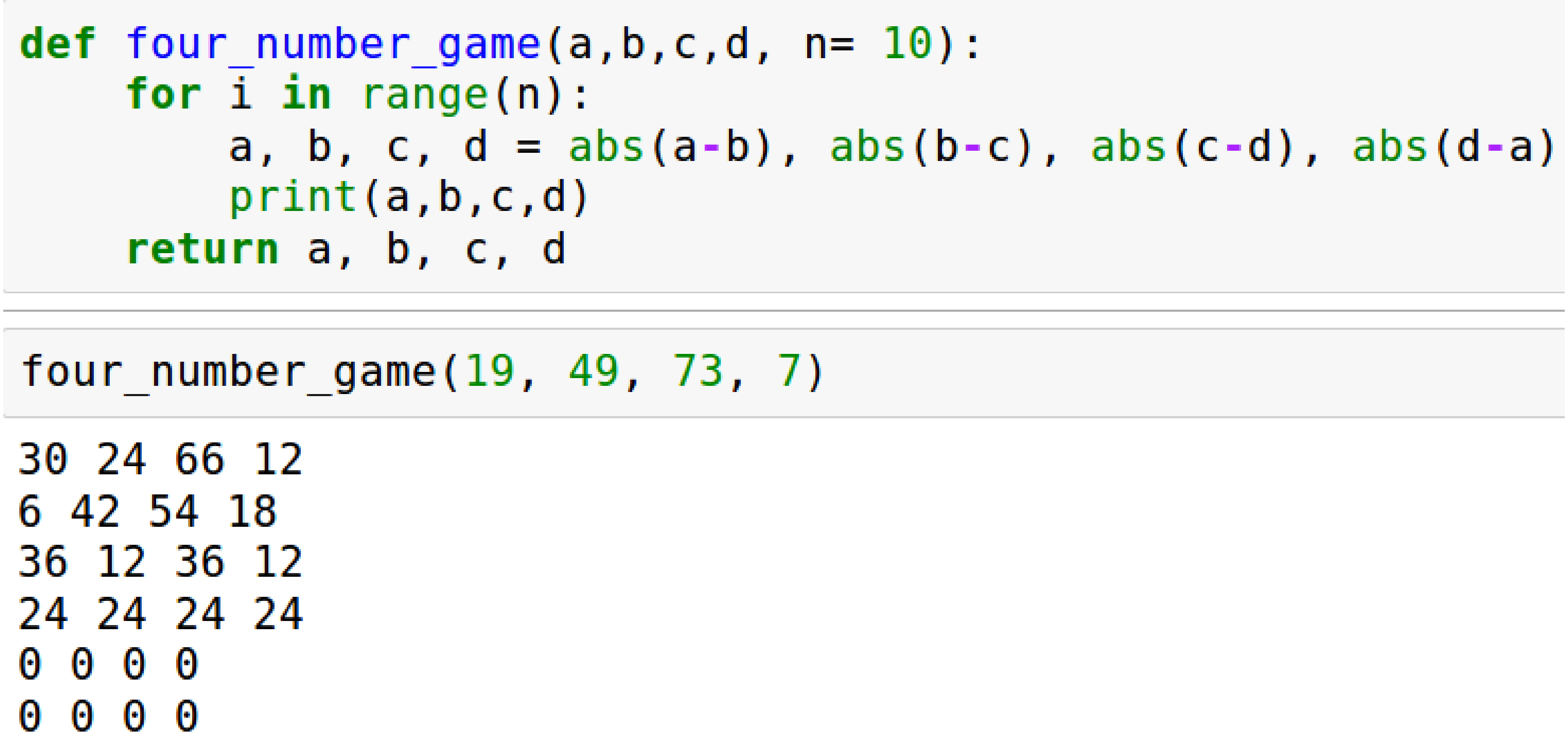
```
def riffle_shuffle(deck):
    n = len(deck)
    part1 = deck[:n//2]
    part2 = deck[n//2:]

    s = []
    for x,y in zip(part1, part2):
        s.extend([x,y])

    if n%2==1: # in case we have odd number of items in deck!
        s.append(part2[-1]) #last item from second list

    return s
```

out/in

```
 0 Python Conference 2018
 1 Peyrtehnocne  C2o0n1f8
 2 Pee y rCt2eoh0nno1cfn8
 3 Poehe0 nyn or1Cctf2ne8
 4 Poore1hCec0t fn2ynne 8
 5 Pto ofrne21yhnCneec 08
 6 Python Conference 2018
 7 Peyrtehnocne  C2o0n1f8
 8 Pee y rCt2eoh0nno1cfn8
 9 Poehe0 nyn or1Cctf2ne8
10 Poore1hCec0t fn2ynne 8
```

```
 1 ePryetnhcoen  2C0o1n8f
 2 ne P r2yCe0ton1hnc8ofe
 3 tnoen 1Ph nrc28yoCfee0
 4 rtcn2o8eyno C1fPehe 0n
 5  rCt1cfnP2eoh8ee y0nno
 6 o hr8Cete1 cyf0nnPn2oe
 7 coy fh0rn8nCPent2eo1e
 8 CcPoeyn tf2he0or1ne8 n
 9 hCec0Poore1ynne 8t fn2
10 yhnCneec 08Pto ofrne21
11 Python Conference 2018
12 ePryetnhcoen  2C0o1n8f
```

Four Number Game

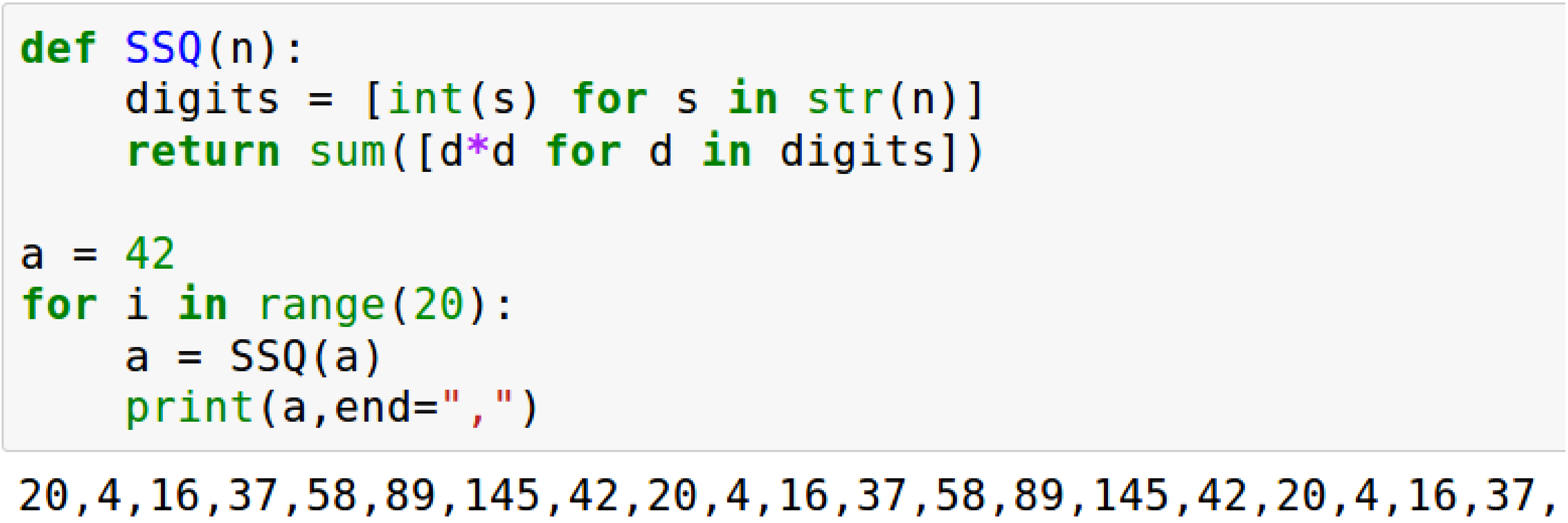


```
def four_number_game(a,b,c,d, n= 10):
    for i in range(n):
        a, b, c, d = abs(a-b), abs(b-c), abs(c-d), abs(d-a)
        print(a,b,c,d)
    return a, b, c, d
```

```
four_number_game(19, 49, 73, 7)
```

```
30 24 66 12
6 42 54 18
36 12 36 12
24 24 24 24
0 0 0 0
0 0 0 0
```

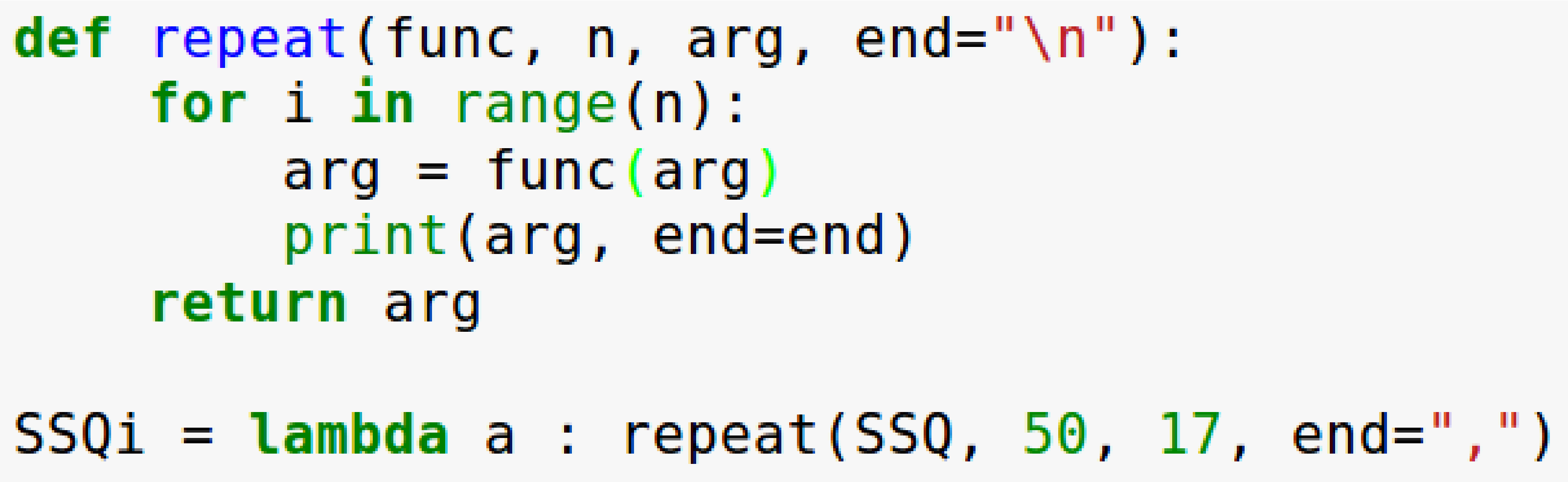
SSQ



```
def SSQ(n):
    digits = [int(s) for s in str(n)]
    return sum([d*d for d in digits])

a = 42
for i in range(20):
    a = SSQ(a)
    print(a,end=",")
```

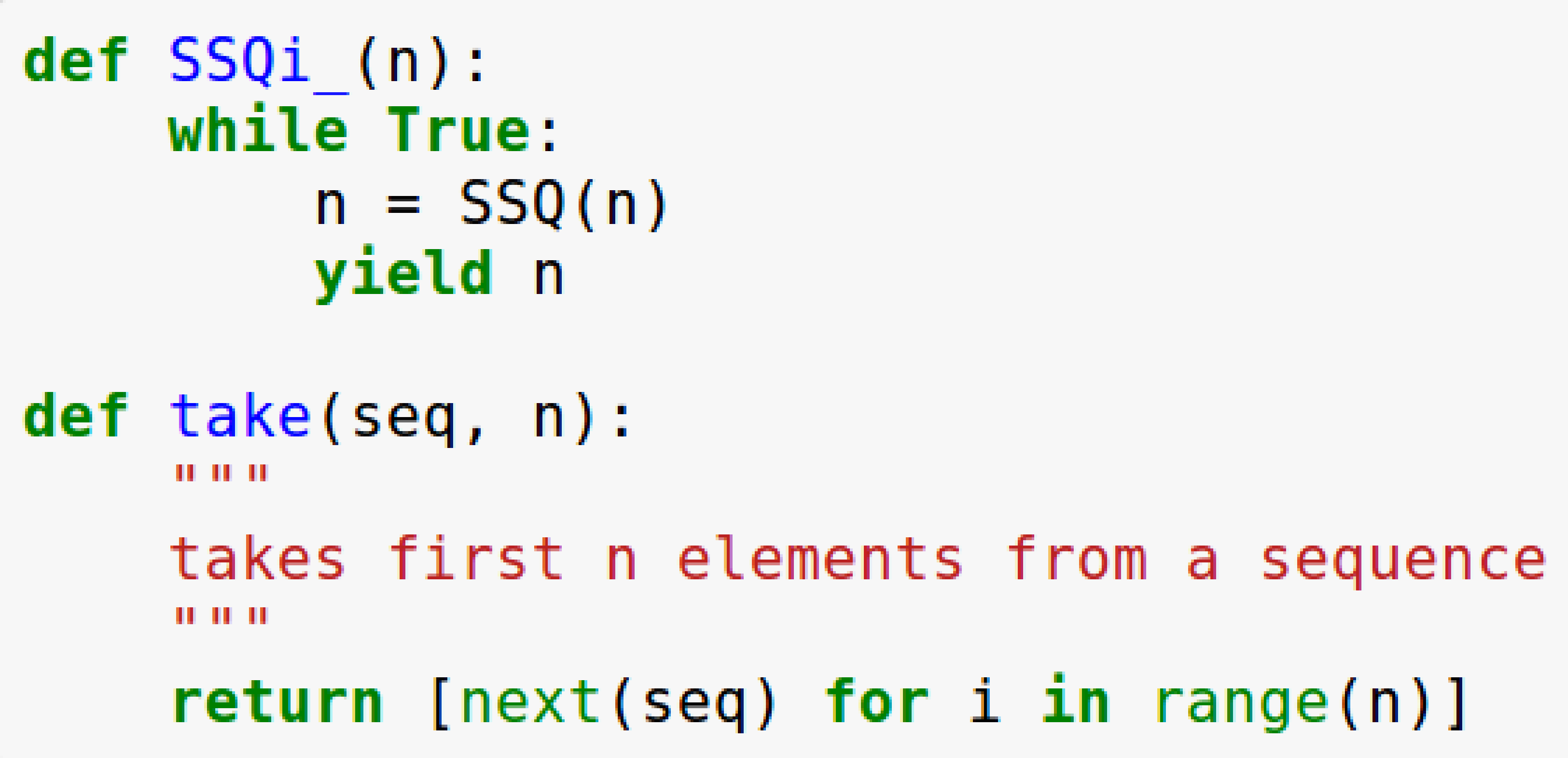
```
20,4,16,37,58,89,145,42,20,4,16,37,58,89,145,42,20,4,16,37,
```

Functions, First Class Objects



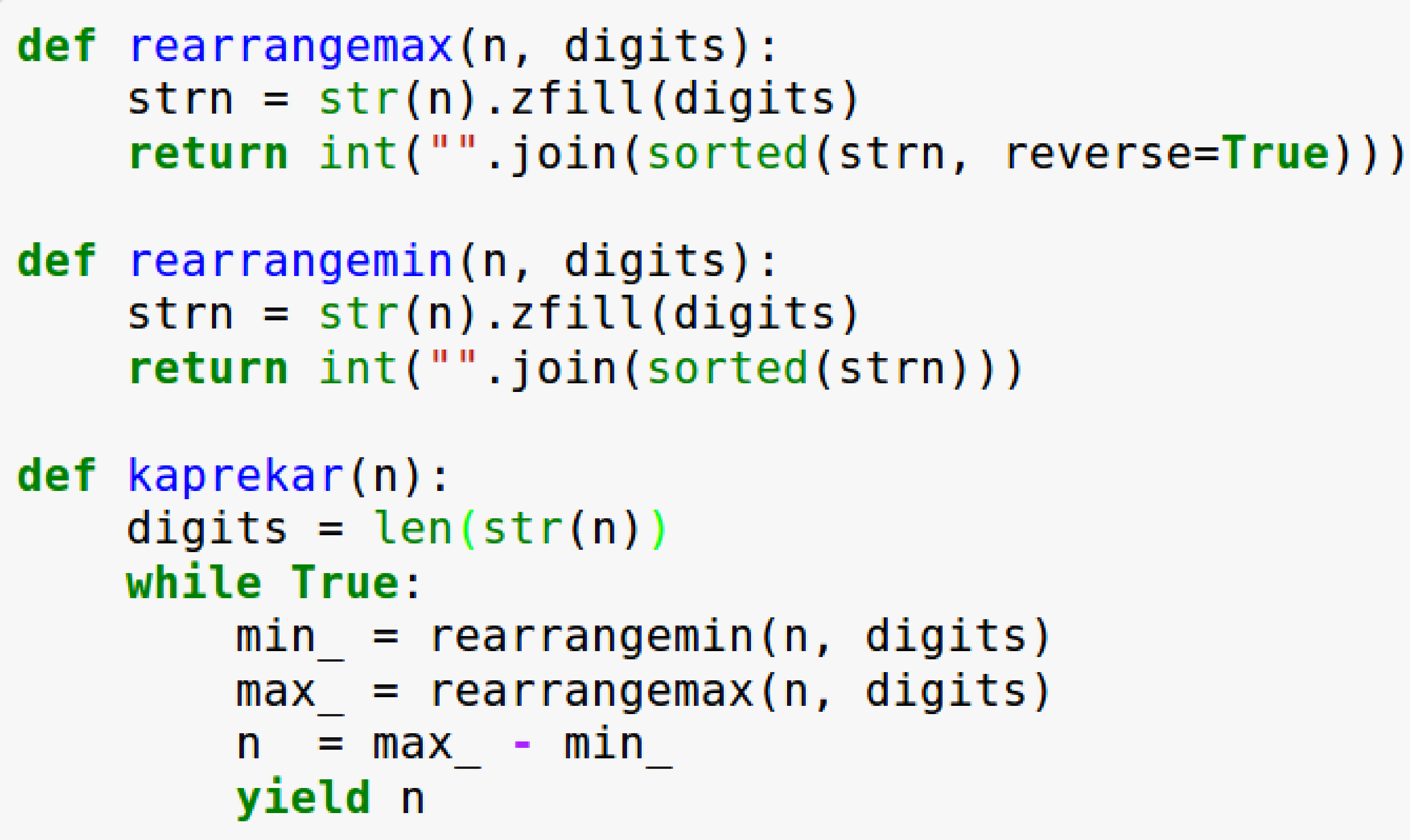
```
def repeat(func, n, arg, end="\n"):
    for i in range(n):
        arg = func(arg)
        print(arg, end=end)
    return arg

SSQi = lambda a : repeat(SSQ, 50, 17, end=",")
```

Sequences



```python
def SSQi_(n):
    while True:
        n = SSQ(n)
        yield n

def take(seq, n):
    """
    takes first n elements from a sequence
    """
    return [next(seq) for i in range(n)]
```

Kaprekar Iteration



```python
def rearrangemax(n, digits):
    strn = str(n).zfill(digits)
    return int("".join(sorted(strn, reverse=True)))

def rearrangemin(n, digits):
    strn = str(n).zfill(digits)
    return int("".join(sorted(strn)))

def kaprekar(n):
    digits = len(str(n))
    while True:
        min_ = rearrangemin(n, digits)
        max_ = rearrangemax(n, digits)
        n  = max_ - min_
        yield n
```