

# 5.1. Iteration with `range`

- Using `for` with `range`
- Summation
- Computing Functions and for-loops
- Open-Ended repetition
- The `while` statement
- `Example` : Fibonacci Numbers

# for-loop Mechanics with range

```
for k in range(n):
```



Loop Body

Let  $k = 0$  and then execute the loop body.

Let  $k = 1$  and then execute the loop body.

Let  $k = 2$  and then execute the loop body.

:

Let  $k = n-1$  and then execute the loop body.

# Iterating Through a Range

```
n = 4
for k in range(n):
    Print(k)
```

Output:

0  
1  
2  
3

# Iterating Through a String

```
s = 'abcd'  
for k in s:  
    Print(k)
```

Output:

```
a  
b  
c  
d
```

In this example, the “for-loop” variable is `k`. One at a time, it takes on the value of each character in `s`.

# Note the Similarities

```
n = 4
for k in range(n):
    Print(k)
```

Output:

0  
1  
2  
3

```
s = 'abcd'
for k in s:
    Print(k)
```

Output:

a  
b  
c  
d

# Summation is a Good Example

```
n = 4
s = 0
for k in range(n):
    s = s + (2**k)
print s
```

$$s = \sum_{k=0}^3 2^k$$

Output:

15

$$1 + 2 + 4 + 8 = 15$$

We are repeating the purple box 4 times

# More General:

$$1 + 2 + 4 + \dots + 2^{n-1}$$

```
n = int(input('input anynumber:'))  
s = 0  
for k in range(n):  
    s = s + (2**k)  
print(s)
```

$$s = \sum_{k=0}^{n-1} 2^k$$

Counting:  
A Special Type of Summation



How Many Integers Less than (100) are there that are divisible by 5?

```
N = 0
for k in range(101):
    if k%5==0:
        N = N+1
print N
```

---

Output: 

# "Counting from Here to There"

```
Here = 20  
There = 24  
for k in range(Here, There) :  
    print k
```

Output:

```
20  
21  
22  
23
```

# "Counting Down"

```
Here = 20
There = 24
for k in range(There, Here, -1) :
    print k
```

Output:

```
24
23
22
21
```

## 5.2. Open-Ended Iteration

So far, we have only addressed iterative problems in which we know (in advance) the required number of repetitions. **Not** all iteration problems are like that. Some iteration problems are open-ended.

**As example,** keep tossing a coin until the number of heads and the number of tails differs by 10

# The While Loop

We introduce an alternative to the for-loop called the **while-loop**.

The while loop is more flexible and is essential for "open ended" iteration.

# While-loop Mechanics

**k = 0**

**While k < n :**

Loop Body

**k = k+1**

Let  $k = 0$  and then execute the loop body.

Let  $k = 1$  and then execute the loop body.

Let  $k = 2$  and then execute the loop body.

:

Let  $k = n-1$  and then execute the loop body.

# How Does a While-Loop Work?

A simple example:

Sum the first 5 whole numbers and display the summation process.

# Two Solutions

```
k = 1
s = 0
while k < 6:
    s = s + k
    k = k + 1
    print (k, s)
```

```
S=0
for k in range(1, 6):
    s = s + k
    print k, s
```



# The While-Loop Solution

```
k = 0
s = 0
while k < 5:
    k = k + 1
    s = s + k
    print(k, s)
```

1	1
2	3
3	6
4	10
5	15

Observation: `k` is used for counting, `s` is used for the running sum, and the `while` is used to control the repetition of the indented code.

Example

Fibonacci Numbers

# Fibonacci Numbers

Here are the first 12 Fibonacci Numbers

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144



Starting here, each one is the sum of its two predecessors

# Generating Fibonacci Numbers

```
x = 0
print x
y = 1
print y
for k in range(6):
    z = x+y
    x = y
    y = z
    print z
```

```
x = 0
print x
y = 1
print y
k = 0
while k<6:
    z = x+y
    x = y
    y = z
    print z
    k = k+1
```