



Loop and iteration (2)

Introduction to Programming and Problem Solving

Erdogan Dogdu

Computer Science Department / Angelo State University

range function

- **range(n)** generates a sequence of integers from 0 to $n-1$

```
>>> for i in range(5):  
...     print(i)  
...  
0  
1  
2  
3  
4
```

range function

- **range(x,y)** generates a sequence of integers from **x to y-1**
- **range(x,y,z)** generates a sequence of integers from **x to y-1** incremented by **z**

```
range(5, 10)
```

```
5, 6, 7, 8, 9
```

```
range(0, 10, 3)
```

```
0, 3, 6, 9
```

```
range(-10, -100, -30)
```

```
-10, -40, -70
```

range returns a function/object

```
>>> x = range(5)
```

```
>>> x
```

```
range(0, 5)
```

```
>>> list(x)
```

```
[0, 1, 2, 3, 4]
```

```
>>> y = list(x)
```

```
>>> y
```

```
[0, 1, 2, 3, 4]
```

```
>>> for i in x:
```

```
...     print(i)
```

```
...
```

```
0
```

```
1
```

```
2
```

```
3
```

```
4
```

list, sum, len built-in functions

```
>>> x = range(5)
```

```
>>> x
```

```
range(0, 5)
```

```
>>> list(x)
```

```
[0, 1, 2, 3, 4]
```

```
>>> y = list(x)
```

```
>>> y
```

```
[0, 1, 2, 3, 4]
```

```
>>> sum(x)
```

```
10
```

```
>>> sum(y)
```

```
10
```

```
>>> len(x)
```

```
5
```

```
>>> len(y)
```

```
5
```

Example: Finding prime numbers

```
for n in range(2, 10):  
    isPrime = True  
    for x in range(2, n):  
        if n % x == 0:  
            print(n, 'equals', x, '*', n//x)  
            isPrime = False  
            break  
    if isPrime:  
        print(n, 'is a prime number')
```

Example: Fibonacci numbers

```
>>> def fib(n):      # write Fibonacci series up to n
...     """Print a Fibonacci series up to n."""
...     a, b = 0, 1
...     while a < n:
...         print(a, end=' ')
...         a, b = b, a+b
...     print()
...
>>> # Now call the function we just defined:
... fib(2000)
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597
```

Random numbers

```
>>> import random
>>> random.random()
0.7288786316051834
>>> random.random()
0.6944688980009052
>>> random.random()*100
3.974298107543217
>>> random.random()*100
64.80664718794529
```

```
>>> int(random.random()*100)
99
>>> int(random.random()*100)
93
>>> int(random.random()*100)
60
```

Random numbers

```
import random
countbig = 0
countsmall = 0
for i in range(100):
    x = int(random.random() * 100)
    print(x, end=' ')
    if x >= 50:
        countbig += 1
    else:
        countsmall += 1
print()
print("Random numbers >= 50: ", countbig)
print("Random numbers < 50: ", countsmall)
```

3 92 56 64 89 18 80 74 3 84 59 23 78
15 62 10 0 82 21 43 43 68 28 67 86 0
92 94 60 0 26 56 37 67 13 62 25 83 75
28 27 86 43 76 57 95 48 8 51 36 80 65
84 97 24 10 50 63 80 65 8 9 44 42 25
88 19 73 30 40 29 70 90 78 59 47 29
69 28 34 2 97 67 90 49 45 93 43 83 68
52 29 11 68 3 44 22 24 56 97

Random numbers >= 50: 52

Random numbers < 50: 48

Shuffle numbers randomly?

```
>>> x = list(range(10))  
>>> x  
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]  
>>> random.shuffle(x)  
>>> x  
[2, 9, 6, 3, 1, 7, 8, 4, 5, 0]
```

Randomly choose elements from a list

```
>>> x = list(range(10))
>>> x
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
>>> random.choice(x)
5
>>> random.choice(x)
4
>>> random.choice(x)
8
```

Notice the assignment statement

```
>>> a,b = 2,3
```

```
>>> print(a,b)
```

```
2 3
```

```
>>> a,b = b,a
```

```
>>> print(a,b)
```

```
3 2
```

```
>>> a,b = b,a+b
```

```
>>> a
```

```
2
```

```
>>> b
```

```
5
```

Summary

- Built-in functions:
 - **range**: generates a sequence of numbers
 - **list**: creates a list from a sequence generated (in memory)
 - **sum**: returns the sum of a sequence/list of numbers
 - **len**: returns the number (count) of a sequence/list of numbers
 - **random.random()**: returns a random number between (0.0,1.0)
- $a, b = c, d$
 - double assignments, happens at the same time