Sorting
Outline

• myList.sort()

• sorted(myList)
How to learn a built-in function?

• Name: fname()
• Arguments: fname(Arg1, Arg2,......)
• Return value: None or a list or a number?
• For example,

```python
myList = [9, 8, 1, 3, 16, 27]
resultList = list(filter(lambda x:x%2==1, myList))
listSum = sum(myList)
```
Sorting Basics -- sorted(myList)

Example:

```python
myList = [5, 1, 4, 3, 2]
resultList = sorted(myList)
print(resultList)
```

Running result:

```
[1, 2, 3, 4, 5]
```
Example:

```python
myList = [5, 1, 4, 3, 2]
result = myList.sort()
print(myList)
print(result)
```

Running result:

```
[1, 2, 3, 4, 5]
None
```
Sorting Basics

• Differences between `sorted(myList)` and `myList.sort()`:
  1. `myList.sort()` method is only defined for lists. In contrast, the `sorted(myList)` function accepts any iterable type (a sequence)

  2. `myList.sort()` will change the “myList”, but `sorted(myList)` won’t change “myList”

  3. Type of returning value:
     `myList.sort()`: None
     `sorted(myList)`: an ordered list
Key function

- The value of the key parameter should be a function that takes a single argument and returns a key to use for sorting purposes. The first entry is used as the sorting key by default.
- Example:

  ```python
  grade_tuples = ( ('Math505', 'Fall2012', B),
                  ('CS130R', 'Spring2013', A),
                  ('BIOS710', 'Fall2013', C))
  sorted(grade_tuples, key=lambda x: x[2])
  ```

Running steps:
1. read input arguments
2. Identify the third entry of each element: B A C
3. sort all elements using their third entry

Running result: (list type)

```python
[('CS130R', 'Spring2013', A), ('Math505', 'Fall2012', B), ('BIOS710', 'Fall2013', C)]
```
Key function

• What’s the running result of

grade_tuples = ( ('Math505', 'Fall2012', B),
    ('CS130R', 'Spring2013', A),
    ('BIOS710', 'Fall2013', C))
result1 = sorted(grade_tuples, key=lambda x: x[2])
result 2= list(grade_tuples).sorted(grade_tuples, key=lambda x: x[1])

print(result1)
print(grade_tuples)
print(result2)
print(type(result1))
print(type(result2))
Ascending and Descending

- Both `myList.sort()` and `sorted(myList)` accept a `reverse` parameter with a `boolean` value.
- This is used to flag descending sorts.
- Example:

  ```python
  grade_tuples = ( ('Math505', 'Fall2012', B),
                  ('CS130R', 'Spring2013', A),
                  ('BIOS710', 'Fall2013', C))
  sorted(grade_tuples, key = lambda x: x[2], reverse = True)
  ```

Question: what’s the running result of

```python
grade_tuples.sort(key = lambda x:x[1], reverse = False)
```

Or is there an error in it?
In-class Exercise

1. Given a tuple, like
   
   \[
   \text{tuples} = (('Math','A'), ('Calculus','C'), ('Physics','B'))
   \]
   
   How to sort the above tuple according to the score of each course, with the lower score before higher score?

2. Given a list with n numbers, like
   
   \[
   \text{numList} = [5, 89, 132, 67, 31, 89, 97]
   \]
   
   How to sort the list with larger number before smaller number?