By default, does the function `sorted` place the contents of a list in ascending or descending order? (Ascending would be increasing from left to right, while descending would be decreasing from left to right.)

- Ascending
- Descending

**Correct Answer:** Ascending

Consider a tuple of name, age, and weight data. An example tuple might read:

- ("Alice", 20, 120.0)

Also consider a list of these tuples:

```python
data = [("Alice", 20, 120.0), ..]
```

Which of the following lines of code will return a sorted list of the tuples by age in ascending order? Assume that the following line of code has already been executed:

```python
from operator import itemgetter
```

- `sorted(data, key=itemgetter(1))`
- `sorted(data, key=itemgetter(0))`
- `data.sort(key=itemgetter(1))`

**Correct Answer:** `sorted(data, key=itemgetter(1))`
sorted(data, key=itemgetter(1))

Feedback:
The second answer sorts by name and is not correct.
The third answer is not correct because although the variable data will contain a sorted list, but the line of code will return `None`.

2/2

Consider a tuple of name, age, and weight data. An example tuple might read:

("Alice", 20, 120.0)

Also consider a list of these tuples:

data = [("Alice", 20, 120.0), .. ]

In the box below, write a line of code that will return a sorted list of tuples in priority by age, then by weight, then by name.

```
sorted(data, key=itemgetter(1, 2, 0))
```

Correct

Answer:

Feedback:

sorted(data, key=itemgetter(1, 2, 0))

2/2

Consider the following code:

data = [-3.5, -2.5, -1.5, 0, 1, 2, 3]
sorted_data = sorted(data, key=abs)

In the box below, indicate the contents of the list sorted_data. For example, if the list contained the integers 39, 42, and 9001 in that order, then you would write:

```
[39, 42, 9001]
```

Correct

Answer:

Feedback:
[0, 1, -1.5, 2, -2.5, 3, -3.5]. This is because the sorting will use absolute value for comparison because the abs function is applied to all values for the purpose of sorting.

Questions or Comments?
Contact Ruth Anderson at rea@cs.washington.edu