4.3 Sets

Difference between a set and a list

Whereas a list can store the same item multiple times, a set stores just one copy of each item. Sets are standard mathematical objects with associated operations (union, intersection, difference, and symmetric difference). Python implements each of these standard operations that would take quite a few lines of code to implement from first principles.

Motivation for using Sets

Consider the problem of making a list of each student in either Mr. Thomas's English class or Mrs. Hernandez's Math class. Some students might be in both, but it doesn't make sense to list these students twice. This idea of taking two lists and merging them into one with one copy of any item that appears in either list is identical to the mathematical notion of taking the union between two sets, where each class is one set. Similarly, creating a list of students in both classes is the same as taking the intersection of the two sets. Making a list of each student in Mr. Thomas's class who ISN'T in Mrs. Hernandez's class is the set difference between the first class and the second class. Notice that order matters here, just as it does in regular subtraction. Finally, symmetric difference is the set of students who are in exactly one of the two classes. In different contexts it may be desirable to find the outcome of any of these four set operations, between two sets.

Initializing a set

We can create an empty set as follows:

```
class1 = set()
```

This creates class1 to be an empty set.

We can initialize a set with elements as follows:

```
class2 = set(["alex", "bobby", "dave", "emily"])
```

Adding an item to a set

We can add an item as follows:

```
class1.add("bobby")
class1.add("cheryl")
```

Note: Sets can store anything, not just strings. It's just easiest to illustrate the set methods using sets of strings.

Standard Set Operations

The following table shows the four key operators for sets. Assume that s and t are arbitrary sets.

Operation	Expression
Union	s | t
Intersection	s & t
Set Difference	s − t
Symmetric Difference	s ^ t

Using our two set variables from above, we can evaluate each of these operations as follows:

```
>>> class3 = class1 | class2
>>> class4 = class1 & class2
>>> class5 = class1 - class2
>>> class6 = class2 - class1
>>> class7 = class1 ^ class2
>>> print(class3)
{'cheryl', 'dave', 'alex', 'bobby', 'emily'}
>>> print(class4)
{'bobby'}
>>> print(class5)
{'cheryl'}
>>> print(class6)
{'dave', 'alex', 'emily'}
>>> print(class7)
{'cheryl', 'dave', 'alex', 'emily'}
```