CISS450: Artificial Intelligence
Lecture 8: Dictionaries

Yihsiang Liow

Agenda

- Study dictionaries which are mapping types

Dictionaries

- Dictionaries are mapping types. Right now there is only one mapping type in Python.
- Dictionaries are hash tables
- Dictionary values are mutable
- Dictionary keys are immutable
- You can think of dictionaries as arrays but they are accessed by immutable objects and the entries are not ordered based on the natural ordering of the keys but by their hashed values.

First Examples

- Try:

```
gpa = {}
print(gpa, type(gpa))
gpa["John"] = 3.5
gpa["Bill"] = 2.9
print(gpa, type(gpa))

gpa = {"Mary":3.3, "Susan":4.5}
print(gpa, type(gpa))
```

- We say `"Mary"` is a ***key*** with ***value*** `3.3`

Mutability

- You can change the value of a (key,value) pair in the dictionary. Try:

```
dict = {}
dict[1.23] = 5              # adding key-value
print(dict, type(dict))
dict[1.23] = "Hi"           # modifying value
print(dict, type(dict))
```

Keys

- You can check for existence of keys and generate a list of keys. Try:

```
dict = {}
dict[1] = "one"
dict[2] = "two"
print(1 in dict)
print(3 in dict)
keys = dict.keys()
print(keys, type(keys))
```

- Keys must be immutable. Try:

```
dict = {}
dict[[1,2]] = 3 # BAD! [1,2] is mutable
```

Keys

- Keys must be immutable. Try:

```
dict = {}
dict[[1,2]] = 3 # BAD! [1,2] is mutable
dict[(1,2)] = 3 # OK
```

Values

- You can generate a list of values from dictionaries:

```
dict = {1:"one", 2:"two"}
values = dict.values()
print(values, type(values))
```

Deletion

- You can delete key-value entries
- Try:

```
dict = {1:"one", 2:"two"}
print(dict, type(dict))
del dict[2]
print(dict, type(dict))
```

Tuples Conversion

- You can convert a dictionary to a list of key-value tuple pairs
- Try:

```
dict =  {1:"one", 2:"two"}
key_values = dict.items()
print(key_values, type(key_values))

for key, value in dict.items():
    print(key, value)
```

Getting Values with Default

- If you retrieve a value via a non-existing key, you can an error
- You can give default value if the key does not exist

```
dict = {1:"one", 2:"two"}
val = dict.get(1, None)
print(val, type(val))
val = dict.get(3, None)
print(val, type(val))
```

Combining Dictionaries

- You can add the key-values of a dictionary into another
- Try:

```
dict = {1:"one", 2:"two"}
print(dict)
dict1 = {3:"three", 4:"four"}
dict.update(dict1)
print(dict, type(dict))
```

Python Documentation

- Make sure you check Python's documentation on dictionaries
- There are some other methods listed in the documentation that is not presented in these notes. But the ones in these notes are the most useful.