Tuples and Dictionaries

GIS 5653 – Spatial Programming and GIS

COLLEGE OF ATMOSPHERIC AND GEOGRAPHIC SCIENCES
DEPARTMENT OF GEOGRAPHY
AND ENVIRONMENTAL SUSTAINABILITY
The UNIVERSITY of OKLAHOMA

Tuples

- Sequence of immutable objects
- Set of comma separated items

`tuples` use round brackets ***()*** *→ `lists` use brackets* ***[]***

Create tuples:

```
t1 = "Test", 100, 7.0 #  create a tuple
print(type(t1))
print(t1)

t2 = (500, 'Jenny', 4.0) # create a tuple
print(type(t2))
print(t2)
```

Immutability

Tuples are immutable

- Other immutable data structure in Python?
- Tuples can be indexed:

```
t1 = 5.7, 8.9, 7.0
print(t1)
print(t1[1])
```

Use when immutability is desirable

→ e.g., point coordinates

Dictionaries versus Lists

- Dictionaries are like list, but more general

Lists:

- Index – integers specify position in list

```
num_list = ['one', 'two', 'three']
```

Dictionaries:

- Unique identifies – can have different data types (immutable)
- Not ordered → key (not index)
- **key-value pairs (items)**

```
num_dict = {'one':'eins', 'two':'zwei', 'three':'drei'}
```

→ Differences?

Create empty …

… list

```
word_list = []
print(word_list)

word_list = list()
print(word_list)
```

… dictionary

```
e2d_dict = {}
print(e2d_dict)

e2d_dict = dict()
print(e2d_dict)
```

Add Elements

List

```
word_list = []
print(word_list)
word_list[0] = 'one'
word_list[1] = 'two'
word_list[2] = 'three'
print(word_list)
```

Dictionary

```
e2d_dict = {}
print(e2d_dict)
e2d_dict['one'] = 'eins'
e2d_dict['two'] = 'zwei'
e2d_dict['three'] = 'drei'
print(e2d_dict)
```

Data Types

Lists

```
cities = ['Berlin', 'London']
pops = [3.5, 8.3]
city_pop_list = [cities, pops]
print(city_pop_list)
[['Berlin', 'London'], [3.5, 8.3]]
```

Dictionary

```
city_pop_dict = dict()
city_pop_dict['Berlin'] = 3.5
city_pop_dict['London'] = 8.3
print(city_pop_dict)
{'London':8.3, 'Berlin':3.5}
```

Working with Dictionaries

- Use *key* to look up corresponding *value*:

```
print(city_pop_dict['London'])
```

- `len()` function works with dictionaries:

```
print(len(city_pop_dict))
```

- `in` operator

```
print('Berlin' in city_pop_dict)
print('Norman' in city_pop_dict)
print(3.5 in city_pop_dict)
```

Explanation?

keys() and values()

List keys

- Lists the keys in the dictionary

```
print(city_pop_dict.keys())
```

List values

- Lists the values in the dictionary

```
print(city_pop_dict.values())
```

List items

- Lists the items in the dictionary as key-value pairs

```
print(city_pop_dict.items())
```

→ use `in` operator with values

```
values = city_pop_dict.values()
print(3.5 in values)
```

Modify Dictionary

Update or add item:

dictionary_name[key] = value # update or add item

```
city_pop_dict['Berlin'] = 1.0 # update existing item
city_pop_dict['Barcelona']  = 1.621 # add new item
print(len(city_pop_dict))
print(city_pop_dict)
```

Delete item:

del dictionary_name[key] # delete the item with the key

```
del city_pop_dict['Berlin']
del city_pop_dict['Barcelona']
print(len(city_pop_dict))
print(city_pop_dict)
```

Modify Dictionary

Update or add mutable item

```
land_use_dict = dict()
land_use_dict['urban'] = [10, 23, 45]
land_use_dict['natural'] = [75, 84, 65]
print(land_use_dict)

land_use_dict['urban'] = [23, 23, 45, 75, 98]
land_use_dict['natural'] = 'N/A'
print(land_use_dict)

land_use_dict['urban'].append(100)
print(land_use_dict)
```

Iterate over Dictionary

- Listing methods can be used to loop through dictionaries
- Iterate through keys or values:

```
for i in city_pop_dict.keys():
    print(i)

for j in city_pop_dict.values():
    print(j)
```

- Use `items()` to loop through dictionaries:

```
for k,v in land_use_dict.items():
    print(k, '\t\t', v)
```

Pick Data Structure

- **Use a list when:**
  - Data has natural order
  - You will need to update or alter the data during the program
  - The primary purpose of the data structure is iteration
- **Use a tuple when:**
  - Data has natural order
  - You will **NOT** need to update or alter the data during the program
  - The primary purpose of the data structure is iteration
- **Use a dictionary when:**
  - The data is unordered, or the order does not matter
  - You will need to update or alter the data during the program
  - The primary purpose of the data structure is looking up values

COLLEGE OF ATMOSPHERIC AND GEOGRAPHIC SCIENCES
DEPARTMENT OF GEOGRAPHY
AND ENVIRONMENTAL SUSTAINABILITY
The UNIVERSITY *of* OKLAHOMA