

# Dictionaries

# Data Collections

- Frequently several individual pieces of data are related
- We can collect them together in one object
- Examples:
  - ▣ A **list** or **tuple** contains an ordered sequence of items
  - ▣ A **string** contains an ordered sequence of characters
  - ▣ A **custom object**. Example from zellegraphics: A **Line object** contains two endpoints, a color, and the window in which it is drawn
  - ▣ A **dictionary** (defined soon) contains key-value pairs

# List - review

- An ordered collection of items
- Usually homogeneous (all items of the same type), but Python does not require this
- Access is **by position** (index) in the list

```
>>> animals = ['dog', 'cat', 'cow']
```

```
>>> animals[1]
```

```
'cat'
```

```
>>> animals[1:3]
```

```
['cat', 'cow']
```

- Lists can be **mutated**: elements changed, deleted, added

```
>>> animals[1] = 'pig'
```

```
>>> animals
```

```
['dog', 'pig', 'cow']
```

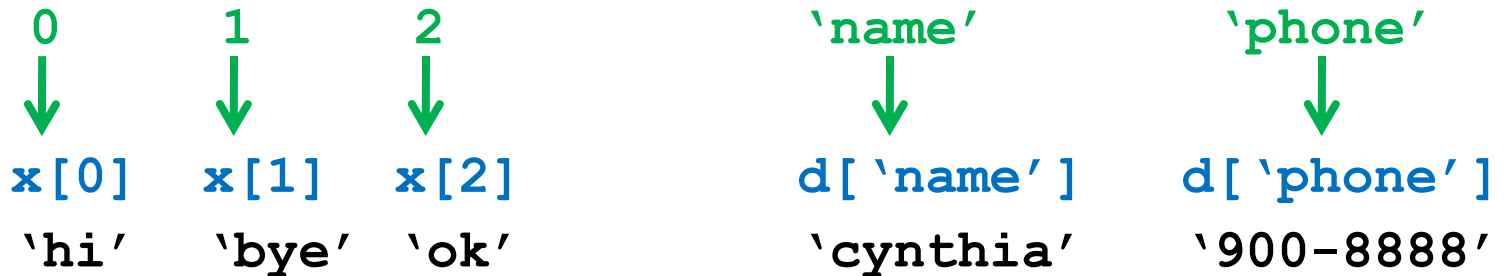
```
>>> animals.append('horse')
```

```
>>> animals
```

```
['dog', 'pig', 'cow', 'horse'] Q1
```

# Dictionary

- A collections object in which each item is a **key-value** pair
- No two items may have the same key
  - ▣ So a dictionary is a function (in the mathematical sense)



- Items are not stored in any particular order
- Typically all keys are same type (not required)
- Keys must be immutable (i.e., string, number, tuple of strings and/or numbers)
- Access to items is by key
  - ▣ key's purpose is similar to list's index
  - ▣ syntax also similar

# Dictionary operations

- Create a new dictionary

```
d = {}
```

```
d = { 'name' : 'bob' , 'age' : 19, 'gpa' : 3.1 }
```

- Access a dictionary value associated with a given key

```
d[ 'age' ]
```

- Modify a dictionary value associated with a given key

```
d[ 'age' ] = 20
```

- Add a new key/value pair to the dictionary

```
d[ 'major' ] = 'be'
```

- Delete a key/value pair from the dictionary

```
del d[ 'gpa' ]
```

- Determine whether a given key is in the dictionary

```
'year' in d
```

# Two main dictionary uses

- The dictionary is associated with a *single* object and stores attributes of the object, all under the single name of the dictionary
  - ▣ Example: The *student* dictionary from the quiz. It is associated with a particular student and stores that student's name, age, weight, and so forth.
  - ▣ Often we store a *list* of these dictionaries, e.g. a list of *student* dictionaries, one for each student in the class.
- A *collection* of similar objects
  - ▣ Designed for fast lookup by key
  - ▣ Examples:
    - A movie database in which we use the title as the key and look up the director.
    - A phone database in which we use the person's name as the key and look up the phone number.

# Exercises on dictionaries and lists

- Checkout 22-Dictionaries
- Begin the TODO's in dictionariesUse1.py
  - ▣ You'll finish them for homework
- Do the TODO's in dictionariesUse2.py

Rest of class: examine the Exam 2 topics document (from Schedule page, session 22). Ask questions. Start working problems. Make a crib sheet.