Lists and dictionaries

Presenter: Steve Baskauf
steve.baskauf@vanderbilt.edu

DiSC DIGITAL SCHOLARSHIP AND COMMUNICATIONS

Jean & Alexander Heard LIBRARIES

CodeGraf landing page

- vanderbi.lt/codegraf

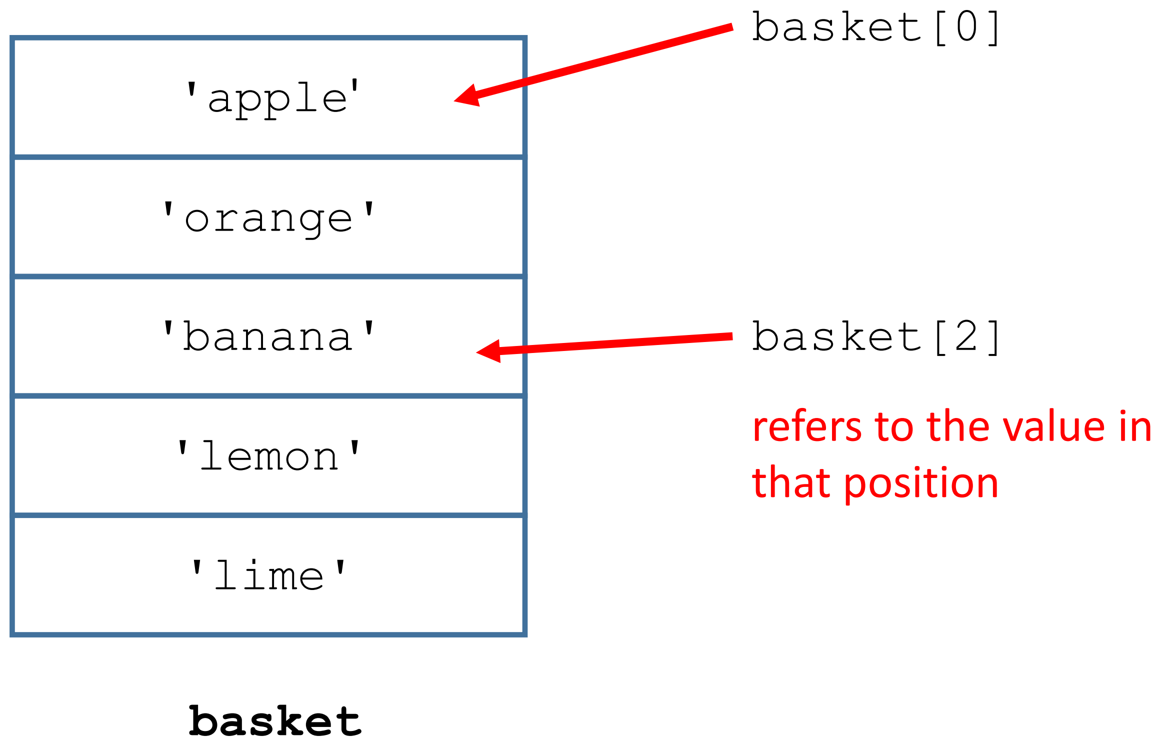
List objects

DIGITAL SCHOLARSHIP AND COMMUNICATIONS

Jean & Alexander Heard LIBRARIES

List objects

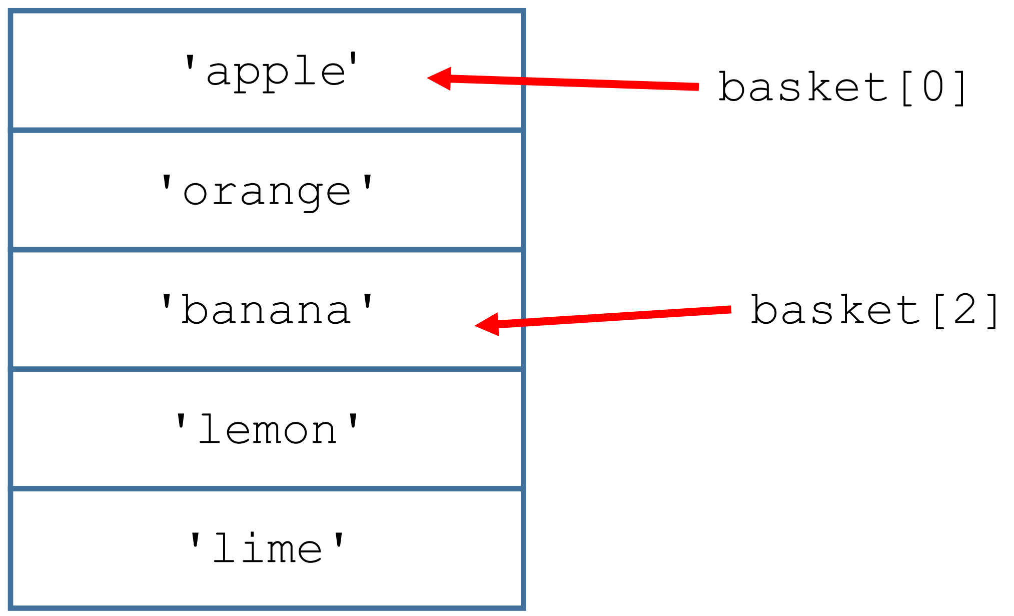
- A list object is a one-dimensional data structure
- Lists can hold any other kind of object.
- The items in a list are referred to by an index number (0-based)



Instantiating a list

- A list can be constructed directly by listing its contents.
- The type of the list is different from the type of items the list contains.
- List items don't have to all be of the same type (but often are).

```
basket = ['apple', 'orange', 'banana', 'lemon', 'lime']
```



Finding the length of a list

- The **`len()`** function will return the number of items in a list
- Example:

```
basket = ['apple', 'orange', 'banana', 'lemon', 'lime']
print(len(basket))
```

- Item indices range from 0 to 4
- Length is 5 (the actual count)
- In many ways, a string is like a list of characters; **`len()`** works for it

Other ways to make a list

DIGITAL SCHOLARSHIP AND COMMUNICATIONS

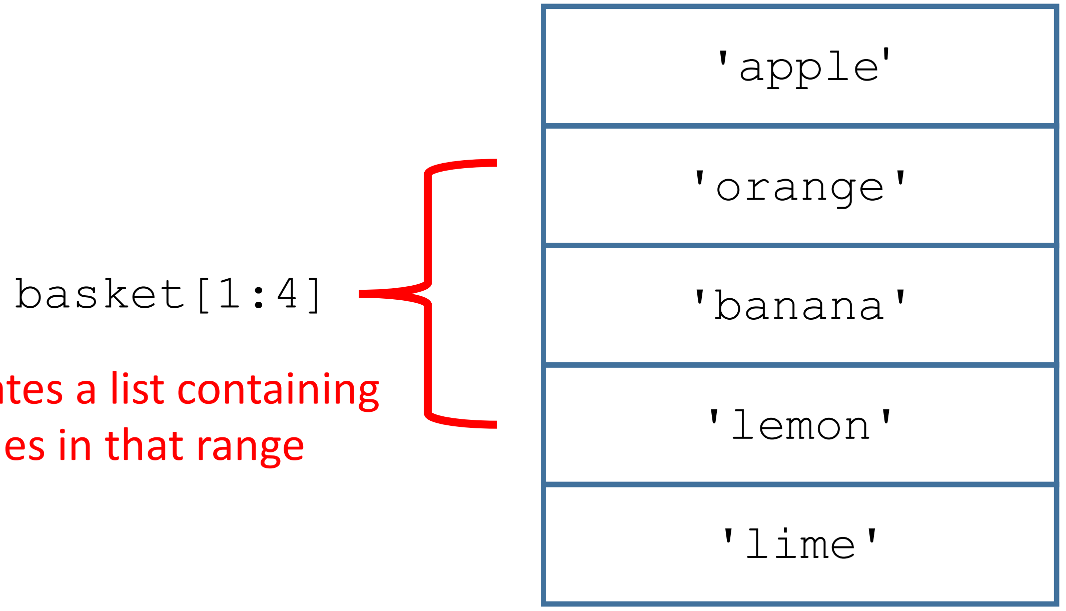
Jean & Alexander Heard LIBRARIES

Output of functions or methods

- The output of a function or method may be a list:
  - **`os.listdir()`** function
  - **`random.sample()`** function
  - **`.split()`** string method

Slicing a list

- A range is given instead of a single index
- Start of range is zero-based.
- End of range is one less than ending index.
- Slicing generates another list



Aside: slicing a string

- Since a string is like a list of characters, we can slice it in the same way
- Example:

```
a_word = 'Mississippi'
word_piece = a_word[1:4]
```

- Range is from 1 to 4
- Slice goes from letters 1 to 3 (start counting with 0)
- Answer: `'iss'`

Useful things to do with lists

- Randomize a list
  - **`random.shuffle()`** function
- Sort a list
  - **`.sort()`** list method
- Pick an item from a list
  - **`random.choice()`** function

Changing a list

DIGITAL SCHOLARSHIP AND COMMUNICATIONS

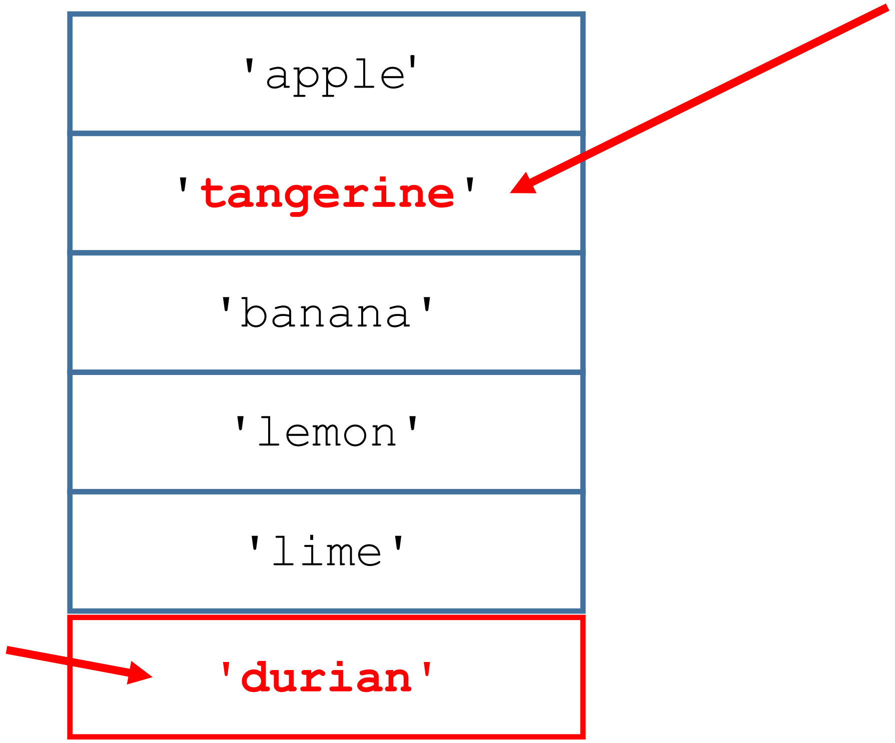
Jean & Alexander Heard LIBRARIES

Editing lists

```
basket = ['apple', 'orange', 'banana', 'lemon', 'lime']
```

```
basket[1] = 'tangerine'
```

We can assign a new value to any list item.



```
basket.append('durian')
```

The `.append()` method does not return a value – it changes the list.

More commands for editing lists

- An **empty list** can be created using

```
basket = []
```

- `.remove()` can be used to remove a **particular value** from the list.
- `del basket[3]` can be used to remove an **item by position**

- The **+** operator appends the items in the second list to the end of the first list.

Dictionary objects

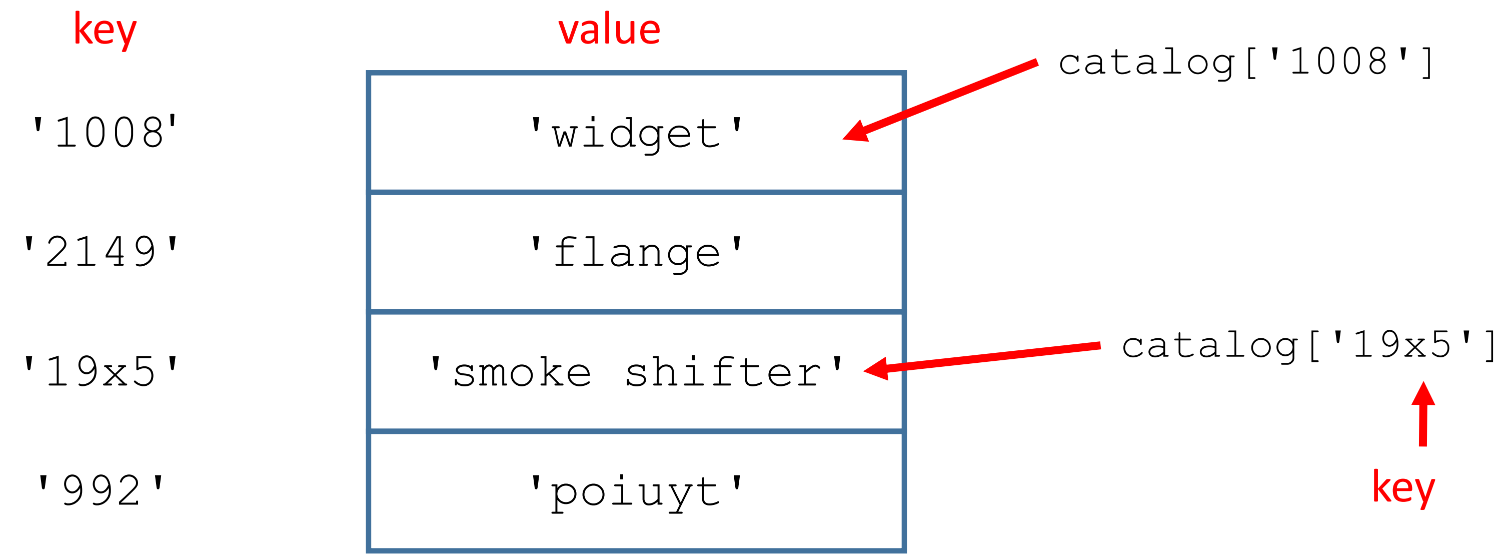
DIGITAL SCHOLARSHIP AND COMMUNICATIONS

Jean & Alexander Heard LIBRARIES

Dictionaries

- Dictionaries are an unordered data structure.
- They're defined using curly brackets: **{ }**
- Values are identified by keys. In this example, the keys are identifiers for the values
- We "look up" values in the dictionary using the keys.

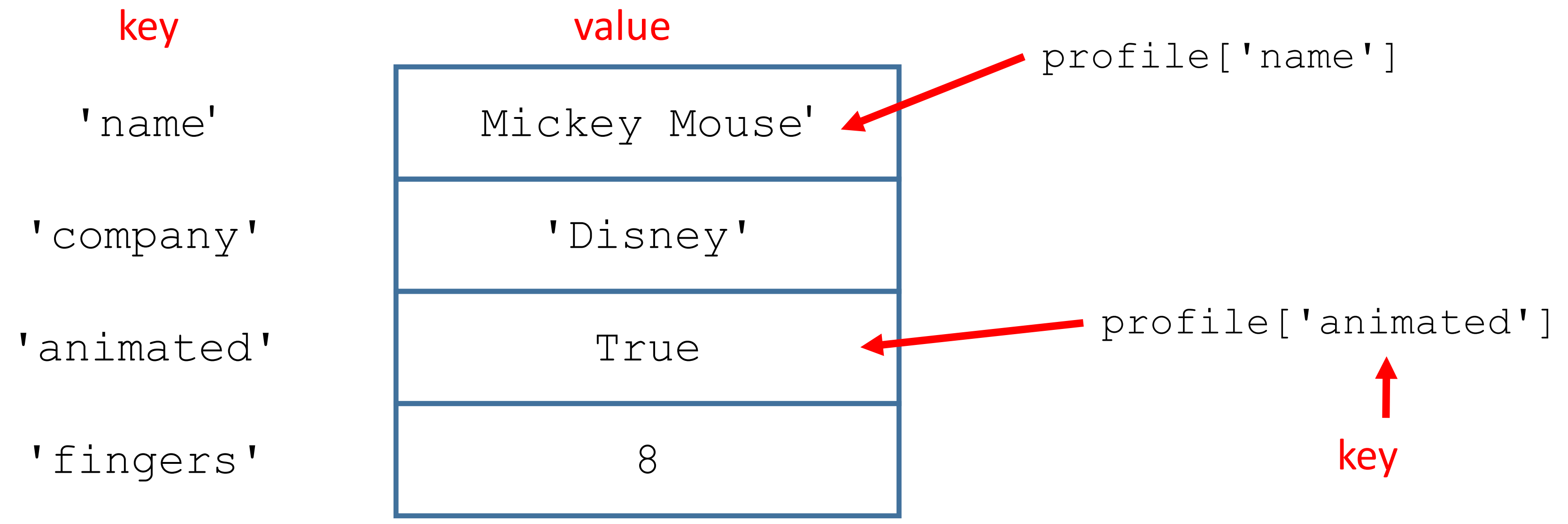
```
catalog = {'1008':'widget', '2149':'flange', '19x5':'smoke
shifter', '992':'poiuyt'}
```



Dictionaries

- Keys can also represent characteristics of an object
- Keys are always strings, values can be any object type
- "dict" is Python slang for "dictionary"

```
profile = {'name':'Mickey Mouse', 'company':'Disney', 'animated':True,
'fingers':8}
```



Commands for editing dictionaries

- An **empty dictionary** can be created using

```
traits = {}
```

- Both **creating** and **changing a value** in the dictionary are done by assigning a value by designated key

```
traits['height'] = 12
```

- An item can be **removed** using the **`del`** command

```
del traits['eye color']
```