

6.101 – Today's Agenda

- Image Processing Art Show
- Bacon numbers lab – efficient data structures
 - sets
 - dictionaries
 - summary of data structures in Python

Sets

- `fruit1.add('apple')`
- `fruit1.discard('grape')` # *no exception if element not in set*
- `fruit1.remove('apple')` # *exception if element not in set*

- `fruit1 & fruit2` # *intersection*
- `fruit1 | fruit2` # *union*
- `fruit1 - fruit2` # *difference*
- `fruit1 ^ fruit2` # *symmetric difference*

- `fruit1.issubset(fruit2)` # *subset?*
- `fruit1.isdisjoint(fruit2)` # *disjoint?*
- `fruit1.issuperset(fruit2)` # *superset?*

Dictionaries

- `table = {}` # create empty dictionary
- `table[27] = 'foo'`

- for key in table:
- for val in table.values():
- for key,val in table.items():

- `table.get(32, [])`
- `table[32].append(1)`
- `table[32].setdefault(32, [])`

Summary of the main python data structures

	List	Tuple	Set	Frozenset	Dict
Ordered?	Yes	Yes	No	No	Yes**
Mutable?	Yes	No	Yes	No	Yes
Hashable?	No	Maybe*	No	Maybe*	No
O(1) access?	Yes	Yes	N/A	N/A	Yes
O(1) containment?	No	No	Yes	Yes	Yes***

* only if all elements contained therein are also hashable

** key/value pairs are kept in insertion order, based on when a given key was added to the dictionary (and removing a key and re-adding it puts it at the end).

*** for keys, not for values

Summary of the main python data structures

	List	Tuple	Set	Frozenset	Dict
Ordered?	Yes	Yes	No	No	Yes**
Mutable?	Yes	No	Yes	No	Yes
Hashable?	No	Maybe*	No	Maybe*	No
O(1) access?	Yes	Yes	N/A	N/A	Yes
O(1) containment?	No	No	Yes	Yes	Yes***
Create multiple	[1, 2, 3, 4]	(1, 2, 3, 4)	{1, 2, 3, 4}	frozenset(..)	{1:'a', 2:'b'}
Create single	[1]	(1,)	{1}	frozenset(..)	{1:'a'}
Create empty	[]	()	set()	frozenset()	{}
Add element	append() insert()		add()		setdefault() <i>update()</i> ****
Remove element	remove() pop()		remove() discard()		del pop() popitem()

Takes in a dictionary