*Python Dictionaries Study Sheet Last Update 5 Jul ‘23*

Making a dictionary:

dict1 = {key1 : val1, key2 : val2 ... } or dict1 = dict()

Key properties: can’t have duplicate key values

keys must be immutable (int, float, str, bool, tuple)

Keys can’t be: mutable (lists, dicts)

How does one...

retrieve a value: dict1[‘k’]

return all keys .keys() #returns a dict\_keys object

return all values .values() #returns a dict\_values object

return all key:val pairs .items() #returns a dict\_items object that contains (key, val) tuples that can be turned into a list

add a new entry dict1[‘newkey’] = ‘newval’

check if key exists if *key* in *dict* or if *key* in *dict*.keys()

update an entry dict1.update({‘key’: val})

or dict1[‘key’] = updated\_val

remove an entry .pop(*key*) or del(dict1[*key*])

remove last inserted item .popitem()

does this clone a dictionary? *dict2 = dict1*

no, this only creates a shallow copy; both vars point to the same dict

cloning: dict2 = dict1.copy() or dict2 = dict(dict1)