Dictionaries - common operations in VBA and Python

Operation	VBA	Python	Notes
Create a dictionary	Dim my_dict As New Scripting.Dictionary	my_dict = {}	VBA requires referencing Microsoft Scripting Runtime for early binding.
Early vs Late binding	Dim my_dict As Object Set my_dict = CreateObject("Scripting.Dictionary")		VBA allows late binding for flexibility, but Python does not require this due to dynamic typing.
Add a new entry	my_dict.Add "key", "value"	my_dict["key"] = "value"	
Remove an entry	my_dict.Remove "key"	del my_dict["key"]	
Retrieve value for key	value = my_dict("key")	value = my_dict["key"]	Accessing a non-existent key throws an error in both VBA and Python.
Check if key exists	If my_dict.Exists("key") Then ...	if "key" in my_dict: ...	
Get all keys	allKeys = my_dict.Keys	all_keys = list(my_dict.keys())	In VBA, Keys returns a collection. In Python, keys() returns an iterable.
Get all values	allValues = my_dict.Items	all_values = list(my_dict.values())	In VBA, Items returns a collection. In Python, values() returns an iterable.
Get dictionary size	size = my_dict.Count	size = len(my_dict)	
Clear the dictionary	my_dict.RemoveAll	my_dict.clear()	
Loop through dictionary	For Each key In my_dict.Keys:	for key, value in my_dict.items():	
Copy a dictionary	Set dict_copy = my_dict	dict_copy = my_dict.copy()	In VBA, Set creates a reference. Copying manually would require looping.
Shallow copy	Manual looping required	dict_copy = my_dict.copy()	Python’s copy() creates a shallow copy.

Deep copy	Manual looping required	import copy
		dict_copy = dict.deepcopy(my_dict)
			In Python, deepcopy() ensures nested dictionaries are also copied.

Default value if key missing	If my_dict.Exists("key") Then value = my_dict("key") Else value = "default_value" End If	value = my_dict.get("key", "default_value")	Python’s get() method retrieves a value or returns a default if the key doesn’t exist.
Nested dictionaries	Manually manage nested Scripting.Dictionary objects	my_dict = {"nested_dict": {"key": "value"}}	Python handles nested dictionaries more naturally. In VBA, it requires deeper nesting logic.
Dictionary iteration order	Not guaranteed	Guaranteed since Python 3.7+	Since Python 3.7, dictionaries maintain insertion order. VBA dictionaries do not.

Owen Price, 2024-10-07