Dictionaries in Python

A dictionary is one of the four basic collection types. A dictionary consists of key:value pairs, they are ordered and changeable, no duplicates allowed.

```
# dictionary =  a collection of {key:value} pairs
#               ordered and changeable. No duplicates

capitals = {"USA": "Washington D.C.",
            "India": "New Delhi",
            "China": "Beijing",
            "Russia": "Moscow"}

# print(dir(capitals)) # <-- ALL attributes and methods
# print(help(capitals))
# print(capitals.get("USA"))   # <-- key exists so returns
value
# print(capitals.get("Japan")) # <-- key doesn't exist,
returns none

if capitals.get("Japan"):
    print("That capital exists")
else:
    print("That capital doesn't exist")

capitals.update({"Germany": "Berlin"}) # <-- adds "k:v" pair
# capitals.update({"USA": "Detroit"}) # <-- updates existing
# capitals.pop("China")         # <-- removes a "k:v" pair
# capitals.popitem()            # <-- removes latest "k:" pair
# capitals.clear()              # <-- clear entire dictionary

keys = capitals.keys() # <-- collects just "keys" into a list
for key in capitals.keys():
    print(key)

# values = capitals.values()
# for value in capitals.values():
# print(value)

# items = capitals.items()
# print(items) # <-- output resembles a 2D list of tuples

# for key, value in capitals.items():
#    print(f"{key}: {value}")
```