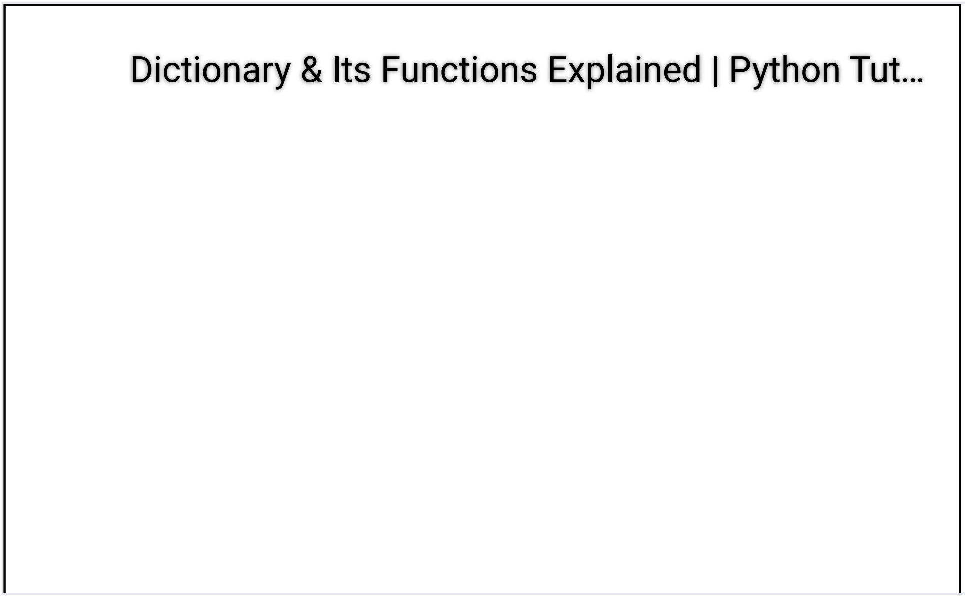

Dictionary & Its Functions Explained | Python Tut...

Show Course Contents ⊕

Overview Q&A Downloads Announcements

Dictionary & Its Functions Explained

Before going through the actual content, i.e., the implementation of a dictionary, it is essential to know some fundamental theories so that we can understand what we are going to learn and why we are spending our precious time learning it.

Let's start with the basic definition of a Python Dictionary:

“Python dictionary is an unordered collection of items. Each item of the dictionary has a key and value pair/ key-value pair.”

Now coming to the more formal approach:

Every programming language has its distinct features, commonly known as its key

Here are a few important features of a python dictionary:

- It is unordered (no sequence is required - data or entries have no order)
- It is mutable (values can be changed even after its formation, or new data/information can be added to the already existing dictionary, we can also pop/remove an entry completely)
- It is indexed (Dictionary contains key-value pairs, and indexing is done with keys. Also, after the Python 3.7th update, the compiler stores the entries in the order they are created)
- No duplication of data (each key is unique; no two keys can have the same name, so there is no chance for a data being overridden)

If we talk a little about how it works, its syntax comprises of key and values separated by colons in curly brackets, where the key is used as a keyword, as we see in real life dictionaries, and the values are like the explanation of the key or what the key holds (the value). And for the successful retrieval of the data, we must know the key so that we can access its value, just like in a regular oxford dictionary where if we don't know the word or its spelling, we cannot obtain its definition. Let's look into the syntax of a Python dictionary:

```
 a = {'key', 'value', 'cow':'mooh'}
print(a['cow'])
#will print "mooh" on the screen
```

With the help of dictionaries, we do not have to do most of our work manually through code like in C or C++. I mean that Python provides us with a long list of already defined methods for dictionaries that can help us do our work in a shorter span of time with a minimal amount of code. Some of these methods are, clear(), copy(), popitem(), etc. The best part about them is that no extra effort is required to be put in order to learn the functionality as their names explain their functions (in most of the cases), such as clear() will clear all the data from the dictionary, making it empty, copy() will make a copy of the dictionary, etc.

the dictionary.

This was pretty much about dictionaries in Python. You will get further details along with explanations and implementation in the video tutorial.

I hope you are enjoying it. Do not forget to subscribe to **CodeWithHarry.**

Code file as described in the video

Copy

```
 # Dictionary is nothing but key value pairs
d1 = {}
# print(type(d1))
d2 = {"Harry":"Burger",
      "Rohan":"Fish",
      "SkillF":"Roti",
      "Shubham":{"B":"maggie", "L":"roti", "D":"Chicken"}}
# d2["Ankit"] = "Junk Food"
# d2[420] = "Kebabs"
# print(d2)
# del d2[420]
# print(d2["Shubham"])
# d3 = d2.copy()
# del d3["Harry"]
# d2.update({"Leena":"Toffee"})
# print(d2.keys())
# print(d2.items())
```

Previous Next

CodeWithHarry