L7 / Diction... / Introducti...

Dictionaries:

- **Dictionary** is an **unordered** collection of `key` and `value` pairs.
- General usage of dictionaries is to store key-value pairs like Employees and their wages, Countries and their capitals, Commodities and their prices etc.
- In a dictionary, the **keys should be unique**, but the values can be similar.
- **Immutable data types** like number, string, tuple etc. are used for the **key** and any data type is used for the **value**.
- Dictionaries are optimized to retrieve values when the keys are known.
- Dictionaries are represented using key-value pairs separated by commas inside curly braces `{}`.
- The key-value pairs are represented as `key` : `value`

- [x] Dictionary is a **Python** data type to store multiple values.
- [ ] We use parenthesis **()** to define a dictionary.
- [ ] In dictionary we represent an element in the **{key-value}** format.
- [x] Keys of the dictionary cannot be changed.
- [ ] **dictionary()** function is used to create an empty dictionary.

Close Reset Submit

Validation OTP for CodeTantra | CodeTantra Edu | Download file | iLovePDF

csmssengg.codetantra.com/secure/topic-details1.jsp?cid=65969c70422c5e79de0c160e&tid=65969c71422c5e79de0c1659&bd=AMTc3M19jdF9jaA==

Q2

100%

L7 / Dictionaries / Understanding Dictionary

Let us create a dictionary called **dict1** with three **keys** named as `name`, `number` and `age`, **values** of dictionary are `Jay`, `514`, and `12` and then try to get values using respective keys.

1. **Creating a dictionary:**

```
dict1 = {"name":"Jay", "number":514, "age":12}
print(dict1)            #Result : {'name': 'Jay', 'number': 514, 'age': 12}
```

2. **Let us retrieve the values using their keys as index:**

```
print(dict1['age'])     # using key called 'age', we can get value 12
print(dict1['number'])  # using key called 'number', we can get value 514
```

3. **Let us find what happens when we try to retrieve an element (key) which is not present in the dictionary:**

```
print(dict1['place'])   #Results in KeyError as there is no key 'place' in
the dictionary
```

If the **key** is not there in the dictionary, we get a `KeyError`.

Dict3.py

Submit

```
1  mydict = {"game":"chess","dish":"chicken","place":"home"}
2  print(mydict['game'])
3  print(mydict['dish'])
4  print(mydict['place'])
5  mydict['game'] = "cricket" # change game chess to cricket using respective key
6  print(mydict['game'])
```

Close | Reset | Submit

Type here to search

39°C Sunny

ENG US 4:59 PM 07/05/2024

Validation OTP for CodeTantra | CodeTantra Edu | Download file | iLovePDF

csmssengg.codetantra.com/secure/topic-details1.jsp?cid=65969c70422c5e79de0c160e&tid=65969c71422c5e79de0c1659&bd=AMTc3M19jdF9jaA==

Q3

100%

L7 / Dictionaries / Accessing Elements of Dictionary

We cannot use a numerical index (as in lists, tuples and strings) to access the items/elements of the dictionaries as dictionaries are unordered(Imagine all the items of a dictionary are put in a bag and jumbled, so there is no order and we cannot retrieve the items using a sequential index.)

The elements of the dictionary can be **retrieved/accessed** in 2 ways.

1. Using the keys of the dictionary.

```
capitals = {"U.S.A": "Washington D.C", "India": "New Delhi", "Nepal":
"Kathmandu"}
print(capitals["India"])        # Printing the value with the key "India"
i.e. "New Delhi".
```

Trying to access an element in the dictionary using a key that is not present in the dictionary, results in a `Key Error`.

```
capitals = {"U.S.A": "Washington D.C", "India": "New Delhi", "Nepal":
"Kathmandu"}
print(capitals["Australia"])    # Since the key "Australia" is not present,
it results in a Key Error
```

Dict1.py

Submit

```
# Taking input from the user for keys and values
key1 = input("key1: ")
value1 = input("value1: ")

key2 = input("key2: ")
value2 = input("value2: ")

key3 = input("key3: ")
value3 = input("value3: ")


# Construct the dictionary with the given key and value

d={key1:value1, key2: value2, key3:value3}
print(d)


print(d.get("name"))


print(d.get("branch"))

print(d.get("place"))
# Print the values of dictionary using keys
```

Close Reset Submit

Type here to search

39°C Sunny ENG US 5:00 PM 07/05/2024

Validation OTP for CodeTantra | CodeTantra Edu | Download file | iLovePDF

csmssengg.codetantra.com/secure/topic-details1.jsp?cid=65969c70422c5e79de0c160e&tid=65969c71422c5e79de0c1659&bd=AMTc3M19jdF9jaA==

Q1

100%

L7 / Data Ty... / Data Type...

Converting data of one type to data of another type is called **Type Conversion**. Python defines various **type conversion functions**.

1. **int(x, base)**: This function converts **x** to an integer of the specified base, the default **x** value is **0**. If the base is not specified, it defaults to **10**.

The syntax of `int()` function is:

```
int(x=0, base=10)
```

Consider the following program to understand the working of `int()` function.

```
s = "0011"       # A binary string.
print(int(s, 2)) # Result: 3
```

DataType1.py

Submit

```
a = int(input("Enter a value: "))
b = input("Enter b value: ")

# Convert "a" to floating point value.
print(float(a))
#Convert "b" to floating point value.
print(float(b))
```

Close | Reset | Submit

Type here to search

39°C Sunny

ENG US 5:00 PM 07/05/2024

Validation OTP for CodeTantra | CodeTantra Edu | Download file | iLovePDF

csmssengg.codetantra.com/secure/topic-details1.jsp?cid=65969c70422c5e79de0c160e&tid=65969c71422c5e79de0c1659&bd=AMTc3M19jdF9jaA==

Q2

100%

L. / Data Type Conv... / Data Type Conversion - ord(),hex(),oct ...

3. **ord( )**: The `ord()` method returns an integer representing a Unicode code point for the given Unicode character.

The syntax of `ord()` function is:

```
ord(c)
```

Examples:

```
print(ord('A'))   # Result: 65
print(ord('a'))   # Result: 97
print(ord('AB'))  # Results TypeError
```

- If the length of the string is greater than 1, then `ord()` function results in a TypeError.

4. **hex(x)**: The `hex()` function converts an integer to its corresponding hexadecimal string.

The syntax of `hex()` function is:

DataType1.py

Submit

```
1  a = input("Enter character: ")
2  b = int(input("Enter an integer: "))
3
4  # Calculate and print the Unicode code point of 'a'
5
6  print(ord(a))
7  # Calculate and print the hexadecimal representation of 'b'
8  print(hex(b))
9  # Calculate and print the octal representation of 'b'
10 print(oct(b))
11 # Calculateand print the complex number representation of 'b'
12 print(complex(b))
13
14
15
16
17
```

Close | Reset | Submit

Type here to search

39°C Sunny

ENG US 5:00 PM 07/05/2024

Validation OTP for CodeTantra | CodeTantra Edu | Download file | iLovePDF

csmssengg.codetantra.com/secure/topic-details1.jsp?cid=65969c70422c5e79de0c160e&tid=65969c71422c5e79de0c1659&bd=AMTc3M19jdF9jaA==

Q3

100%

L7 / Data Ty... /

Magnifier 100%

7. **str(x)**: The `str()` function is used to convert x to a string representation.

The syntax of `str()` function is:

```
str(object='')
str(object=b'', encoding='utf-8', errors='strict')
If no object is given, it returns an empty string.
The behaviour of str() depends on whether encoding
or errors are specified.
```

- object - object whose informal representation is to be returned
- encoding - Defaults of UTF-8. Encoding of the given object
- errors - response when decoding fails.

DataType1.py

Submit

```
1  a = int(input("Enter an integer: "))
2
3  # Calculate and print the character representation of 'a'
4  print(chr(a))
5  # Calculate and print the string representation of 'a'
6  print(str(a))
7
```

Close Reset Submit

Type here to search

39°C Sunny

ENG US 5:01 PM 07/05/2024

L7 / Data Type Conversions / Data Type Conversions

10. **tuple( )** : This function is used to convert any data type to a tuple.

```
str = "python"
print(tuple(str)) # Result: ('p', 'y', 't', 'h', 'o', 'n')
```

11.**set( )** : This function is used to convert any data type to set.

```
str = "python"
print(set(str)) # Result: {'n', 't', 'p', 'h', 'y', 'o'}
```

12. **list( )** : This function is used to convert any data type to a list type.

```
str = "python"
print(list(str)) # Result: ['p', 'y', 't', 'h', 'o', 'n']
```

13.**dict(d)** : This function is used to create a dictionary, but `d` must be tuple of order **(key, value)**.

```
mytuple = ((1, 'a'), (2, 'b'))
print(dict((y, x) for x, y in mytuple)) # {'a': 1, 'b': 2}
```

Here we used **for loop** to iterate every element of tuple object and we use **dict( )** function to convert tuple elements into key-value pairs.

DataType2.py

Submit

```
1  tup1 = tuple(input().split(","))
2  tup2 = tuple(input().split(","))
3
4  # print tup1
5  print(tup1)
6  # print tup2
7  print(tup2)
8
9  list1 = list(tup1)# convert tup1 to list data type
10 list2 = list(tup2)# convert tup2 to list data type
11
12
13
14 print(list1)
15 print(list2)
16
17 dict1 =  zip(list1,list2) # create a dictionary with the elements of list1 as keys ar
18 dict=dict(dict1)
19 print(dict)
20 set1 = set(dict)# convert dict1 to set data type
21
22 print(sorted(set1))
```

Close Reset Submit