1. List the basic data types in Python and classify them as mutable or/and ordered. Also, specify the corresponding class.

Data Type	Ordered	Mutable	Class
String	✅	❌	str
Integer	❌	❌	int
Float	❌	❌	float
Dictionary	✅	✅	dict
List	✅	✅	list
Set	❌	✅	set
Tuple	✅	❌	tuple
Boolean	❌	❌	bool

2. Write a Python program to create a new list of only the first and last elements of the given list.

In [1]:
```
list1 = ['The', 'quick', 'brown', 'fox', 'jumps', 'over', 'the', 'lazy', 'dog']
list2 = list1[0::len(list1)-1]
list2
```

Out[1]:
```
['The', 'dog']
```

3. Write a Python program to create a new list of alternate elements of the given list.

In [2]:
```
list1 = ['The', 'quick', 'brown', 'fox', 'jumps', 'over', 'the', 'lazy', 'dog']
list2 = list1[0::2]
list2
```

Out[2]:
```
['The', 'brown', 'jumps', 'the', 'dog']
```

4. You are given a string and your task is to swap cases. In other words, convert all lowercase letters to uppercase letters and vice versa.

In [3]:
```
string1 = 'McDonald's'
string2 = string1.swapcase()
string2
```

Out[3]:
```
'mCdONALD'S'
```

5. You are given a string. Split the string on a " " (space) delimiter and join using a -hyphen.

In [4]:
```
string1 = 'The quick brown fox jumps over the lazy dog'
my_list = string1.split(' ')
'-'.join(my_list)
```

Out[4]:
```
'The-quick-brown-fox-jumps-over-the-lazy-dog'
```

In [5]:
```
# Alternate solution
string1.replace(' ', '-')
```

Out[5]:
```
'The-quick-brown-fox-jumps-over-the-lazy-dog'
```

6. Write a simple Python program to remove the duplicate elements from a list and return the output as a list.

In [6]:
```
list1 = ['You', 'cannot', 'end', 'a', 'sentence', 'with', 'because', 'because', 'because', 'is', 'a', 'conjunction.']
set1 = set(list1)
list2 = list(set1)
list2
```

Out[6]:
```
['a',
 'because',
 'cannot',
 'You',
 'conjunction.',
 'sentence',
 'end',
 'with',
 'is']
```

In [7]:
```
# Alternate solution
mylist = list(dict.fromkeys(list1))
mylist
```

Out[7]:
```
['You',
 'cannot',
 'end',
 'a',
 'sentence',
 'with',
 'because',
 'is',
 'conjunction.']
```

7. Write a Python program to concatenate two dictionaries into one.

In [8]:
```
dict1 = {'Python':'DS-542','Data Mining':'DS-600','Algorithms I':'DS-590'}
dict2 = {'Algorithms II':'DS-690'}
```

In [9]:
```
dict1.update(dict2)
dict1
```

Out[9]:
```
{'Python': 'DS-542',
 'Data Mining': 'DS-600',
 'Algorithms I': 'DS-590',
 'Algorithms II': 'DS-690'}
```

8. Write a Python program to create set difference.

In [10]:
```
dict1 = {'apple', 'mango'}
dict2 = {'mango', 'orange'}
dict1.difference(dict2)
```

Out[10]:
```
{'apple'}
```

In [11]:
```
# Alternate solution
dict1 - dict2
```

Out[11]:
```
{'apple'}
```

9. Using keys and indexing, grab the 'hello' from the following dictionary :

{'k1':[{'k2':['this',['hello']]}]}

In [12]:
```
dict1 = {'k1':[{'k2':['this',['hello']]}]}
dict1['k1'][0]['k2'][1][0]
```

Out[12]:
```
'hello'
```

10. Explain with example which values are considered as True & False in Python.

Statement	Output	Reason
print(bool(()))	False	Empty tuple is equivalent to False
print(bool(False))	False	False is equivalent to False
print(bool(None))	False	Null or None is equivalent to False
print(bool(0))	False	0 is equivalent to False
print(bool(""))	False	Empty string is equivalent to False
print(bool([]))	False	Empty list is equivalent to False
print(bool({}))	False	Empty set is equivalent to False
print(bool(dict()))	False	Empty dictionary is equivalent to False
print(bool(list))	False	Empty list is equivalent to False
print(bool(set))	False	Empty set is equivalent to False
print(bool(tuple()))	False	Empty tuple is equivalent to False
print(2==2.0)	True	2 is equal to 2.0
print(2>1)	True	2 is greater than 1
print(2<1)	False	2 is not lesss than 1
print(2==1)	False	2 is not equal to 1
print(bool("Hello"))	True	Present value is equivalent to True
print(bool(15))	True	Present value is equivalent to True
print(bool(1.5))	True	Present value is equivalent to True
print(isinstance(15, int))	True	15 is an integer
print(isinstance(1.5, float))	True	1.5 is a float
print(isinstance("1.5", str))	True	"1.5" is a string
print(isinstance("1.5", float))	True	"1.5" is not a float

In [13]:
```
print(bool(()))
```

```
False
```

In [14]:
```
print(2==2.0)
```

```
True
```

In [15]:
```
print(2>1)
```

```
True
```

In [16]:
```
print(2<1)
```

```
False
```

In [17]:
```
print(2==1)
```

```
False
```

In [18]:
```
print(bool("Hello"))
```

```
True
```

In [19]:
```
print(bool(15))
```

```
True
```

In [20]:
```
print(bool(1.5))
```

```
True
```

In [21]:
```
print(bool(False))
```

```
False
```

In [22]:
```
print(bool(None))
```

```
False
```

In [23]:
```
print(bool(0))
```

```
False
```

In [24]:
```
print(bool(""))
```

```
False
```

In [25]:
```
print(bool(()))
```

```
False
```

In [26]:
```
print(bool([]))
```

```
False
```

In [27]:
```
print(bool({}))
```

```
False
```

In [28]:
```
print(bool(dict()))
```

```
False
```

In [29]:
```
print(bool(list()))
```

```
False
```

In [30]:
```
print(bool(set()))
```

```
False
```

In [31]:
```
print(bool(tuple()))
```

```
False
```

In [32]:
```
print(isinstance(15, int))
```

```
True
```

In [33]:
```
print(isinstance(1.5, float))
```

```
True
```

In [34]:
```
print(isinstance("1.5", str))
```

```
True
```

In [35]:
```
print(isinstance("1.5", float))
```

```
False
```