The list and dictionary object **types** are two of the most important and often used types in a Python program. X

Compare the functionalities of those two objects. X

What are some ways to insert, update, and remove elements from lists and dictionaries? This is your code example. X

Why would you choose one data type over another? X

Hello,

Lists are a sequence type whereas dictionaries are a mapping type (support key lookup). They map immutable values to arbitrary values.

A list is a good choice for storing a collection of changeable and potentially duplicate values and it's useful for implementing stacks and queues.

Dictionaries are optimal for storing a group of unique pairs of data that are not accessed by index but rather by key.

(see examples below)

```
cart = ['soap', ('pencil', 'pen', 'marker'), {'apples': 2, 'bananas': 3}, ['cups', 'plates']]
print(cart)
```

```
['soap', ('pencil', 'pen', 'marker'), {'apples': 2, 'bananas': 3}, ['cups', 'plates']]
```

```
cart.insert(1, 'shampoo') # insert at a given index
print(cart)
```

```
['soap', 'shampoo', ('pencil', 'pen', 'marker'), {'apples': 2, 'bananas': 3}, ['cups', 'plates']]
```

```
cart.append('mints') # insert at end
print(cart)
```

```
['soap', 'shampoo', ('pencil', 'pen', 'marker'), {'apples': 2, 'bananas': 3}, ['cups', 'plates'], 'mints']
```

```
cart.extend(['eraser', 'glue']) # insert at end
print(cart)
```

```
['soap', 'shampoo', ('pencil', 'pen', 'marker'), {'apples': 2, 'bananas': 3}, ['cups', 'plates'], 'mints', 'eraser', 'glue']
```

```
cart[-1] = 'tylenol' # update last value
cart[0:2] = ["toothpaste", "toothbrush"] # update multiple values by indices
print(cart)
```

```
['toothpaste', 'toothbrush', ('pencil', 'pen', 'marker'), {'apples': 2, 'bananas': 3}, ['cups', 'plates'], 'mints', 'eraser', 'tylenol']
```

```
cart[-4] = [dinnerware.capitalize() for dinnerware in cart[-4]] # update list by index
print(cart)
```

```
['toothpaste', 'toothbrush', ('pencil', 'pen', 'marker'), {'apples': 2, 'bananas': 3}, ['Cups', 'Plates'], 'mints', 'eraser', 'tylenol']
```

```
cart.remove('toothbrush') # removes the first matching value
print(cart)
```

```
['toothpaste', ('pencil', 'pen', 'marker'), {'apples': 2, 'bananas': 3}, ['Cups', 'Plates'], 'mints', 'eraser', 'tylenol']
```

```
del cart[1] # remove by index
print(cart)
```

```
['toothpaste', {'apples': 2, 'bananas': 3}, ['Cups', 'Plates'], 'mints', 'eraser', 'tylenol']
```

```
cart.pop(1) # remove by index. pop() will return the removed value.
```

```
{'apples': 2, 'bananas': 3}
```

```
cart.pop() # remove last item
```

```
'tylenol'
```

```
contacts = {
    'Ayesha': '975-328-0147',
    'Emily': '814-387-0645',
    'Nozomi': '429-723-1039',
    'Oliver': '240-138-3897',
    'Li': '102-477-2543',
    'Ahmed': '391-242-7953',
    'Gabriel': '638-117-5674',
    'Lucia': '527-833-6518'
}
```

```
contacts['Elena'] = '280-743-1847' # insert new pair at end
contacts.keys()
```

```
dict_keys(['Ayesha', 'Emily', 'Nozomi', 'Oliver', 'Li', 'Ahmed', 'Gabriel', 'Lucia', 'Elena'])
```

```
new_contacts = {'Ray': '487-310-8463', 'Daniel': '612-436-0813'} # insert new pairs at end
contacts.update(new_contacts)
contacts.keys()
```

```
dict_keys(['Ayesha', 'Emily', 'Nozomi', 'Oliver', 'Li', 'Ahmed', 'Gabriel', 'Lucia', 'Elena', 'Ray', 'Daniel'])
```

```
contacts['Li'] = '538-907-8234' # update key value
print(contacts['Li'])
```

```
538-907-8234
```

```
contacts.update(Gabriel='329-567-7843', Ahmed='198-327-8436') # update multiple existing pairs
print(contacts['Gabriel'])
print(contacts['Ahmed'])
```

```
329-567-7843
198-327-8436
```

```
del contacts['Oliver'] # remove pair by key
'Oliver' in contacts
```

```
False
```

```
contacts.pop('Emily') # remove pair by key and return value
```

```
'814-387-0645'
```

```
contacts.popitem() # remove and return last added pair
```

```
('Daniel', '612-436-0813')
```

Thanks,
Lauren

References

The Python Language Reference. (n.d.). *Built-in Types: Lists*

https://docs.python.org/3/library/stdtypes.html#lists

The Python Language Reference. (n.d.). *Built-in Types: Mapping Types -- dict*

https://docs.python.org/3/library/stdtypes.html?highlight=dict#mapping-types-dict

---
Responses and more notes

Hi Padmaja,

You make a good note on performance. Most dictionary operations take constant time as opposed to list operations which often take linear time or longer. Nice work.

Note: https://bradfieldcs.com/algos/analysis/performance-of-python-types/

Dictionaries preserve insertion order.

Rizart:

In regards to your dictionary example, must new_contacts itself be a dictionary to add on to the contacts dictionary?

Is there another way to add on to a dictionary without using a dictionary itself?

Can you insert keys and values separately by other means, perhaps such as a list, into a dictionary?

Hi Rizart,

In my example I'm using update() which accepts either a dictionary or an iterable of key/value pairs as tuples or lists. Here's what inserting a list could look like:

```
new_contacts = [['Ray', '487-310-8463'], ['Daniel', '612-436-0813']]
contacts.update(new_contacts)
contacts.keys()
```

```
dict_keys(['Ayesha', 'Emily', 'Nozomi', 'Oliver', 'Li', 'Ahmed', 'Gabriel', 'Lucia', 'Elena', 'Ray', 'Daniel'])
```

Lauren