Dictionaries vs. Lists

Dictionaries are similar to lists, but there are a few differences:

Both dictionaries and lists:

- are used to organize elements into collections;
- are used to initialize a new dictionary or list, use empty brackets;
- can iterate through the items or elements in the collection; and
- can use a variety of methods and operations to create and change the collections, like removing and inserting items or elements.

Dictionaries only:

- are unordered sets;
- have keys that can be a variety of data types, including strings, integers, floats, tuples;.
- can access dictionary values by keys;
- use square brackets inside curly brackets { [ ] };
- use colons between the key and the value(s);
- use commas to separate each key group and each value within a key group;
- make it quicker and easier for a Python interpreter to find specific elements, as compared to a list.

```python
pet_dictionary = {"dogs": ["Yorkie", "Collie", "Bulldog"], "cats": ["Persian", "Scottish Fold", "Siberian"], "rabbits": ["Angora", "Holland Lop", "Harlequin"]}


print(pet_dictionary.get("dogs", 0))
# Should print ['Yorkie', 'Collie', 'Bulldog']
```

Lists only:

- are ordered sets;
- access list elements by index positions;
- require that these indices be integers;
- use square brackets [ ];
- use commas to separate each list element.

```
pet_list  = ["Yorkie", "Collie", "Bulldog", "Persian", "Scottish Fold", "Siberian", "Angora", "Holland Lop",
"Harlequin"]



print(pet_list[0:3])
# Should print ['Yorkie', 'Collie', 'Bulldog']
```