

1

Dictionaries — a data object that allows you to connect pieces of related information.

— models real life problems more accurately

If I wanted to store information about my cat bongo:

key value.

```
bongo_info = { 'color': 'black', 'weight': 15, 'fav_food': 'pill packet' }
```

now we can access this information:

```
bongo_info['color'] ... prints: 'black'
bongo_info['weight'] ...      15
bongo_info['fav_food'] ... pill packet.
```

keys: "color", "weight", "fav_food"

→ let you access the data.

★ We can add data:

```
bongo_info['texture'] = 'really soft'
```

★ We could start with an empty dictionary

```
bongo_info = {} and then add piece by piece
```

★ Modifying elements ... what if Bongo gained weight?

```
bongo_info["weight"] = 16
print(f"Bongo now weighs {bongo_info['weight']}
      because he ate too many {bongo_info['fav_food']}")
```

★ Removing key-value Pairs.

```
del bongo_info['texture']
```

* Errors ...

if the key 'texture' is not in the dictionary then trying to access the values will give an error.

```
print(bongo_info['texture'])
```

Error...

the `get()` command can help.

```
print(bongo_info.get('Texture', "Can't find texture"))
```

↓
if it exists
it returns the value

↓
if not it
returns the
message.

* You can loop through a dictionary:

- `items()` — gives the key and the value

- `keys()` — just returns the keys

- `values()` — just returns the values.

```
for k, v in bongo_info.values():  
    print(f'Bongos {k} is {v}')
```

this will print: Bongos color is black
Bongos weight is 15
Bongos fav_food is pill pockets.

```
for k in bongo_info.keys():  
    print(f'we know Bongos {k}')
```

this will print: We know Bongos color
We know Bongos weight
We know Bongos fav_food.

```
for v in bongo_info.values():
    print(f'Bongo is {v}')
```

this will print

```
Bongo is black
Bongo is 15
Bongo is pill pockets.
```

* We can also check if values exist in

'color' in bongo_info.keys()	True
'black' in bongo_info.values()	True
'color' in bongo_info.items()	False
('color', 'black') in bongo_info.items()	True

* The sorted command will sort alphabetically by key.

```
sorted(bongo_info.keys())
```

but dictionaries do not have the .sort() function.

NESTING ... you can make a list of dictionaries
a dictionary of lists
a dictionary of dictionaries ...