PYTHON DICTIONARIES:

```
thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
```

Dictionary

Dictionaries are used to store data values in key:value pairs.

A dictionary is a collection which is ordered*, changeable and do not allow duplicates.

Dictionaries are written with curly brackets, and have keys and values:

ExampleGet your own Python Server

Create and print a dictionary:

```
thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
print(thisdict)
```

Dictionary Items

Dictionary items are ordered, changeable, and does not allow duplicates.

Dictionary items are presented in key:value pairs, and can be referred to by using the key name.

Example

Print the "brand" value of the dictionary:

```
thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
print(thisdict["brand"])
```

Ordered or Unordered?

As of Python version 3.7, dictionaries are ordered. In Python 3.6 and earlier, dictionaries are unordered.

When we say that dictionaries are ordered, it means that the items have a defined order, and that order will not change.

Unordered means that the items does not have a defined order, you cannot refer to an item by using an index.

Changeable

Dictionaries are changeable, meaning that we can change, add or remove items after the dictionary has been created.

Duplicates Not Allowed

Dictionaries cannot have two items with the same key:

Example

Duplicate values will overwrite existing values:

```
thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964,
  "year": 2020
}
print(thisdict)
```

Dictionary Length

To determine how many items a dictionary has, use the len() function:

Example

Print the number of items in the dictionary:

```
print(len(thisdict))
```

Dictionary Items - Data Types

The values in dictionary items can be of any data type:

Example

String, int, boolean, and list data types:

```
thisdict = {
  "brand": "Ford",
  "electric": False,
  "year": 1964,
  "colors": ["red", "white", "blue"]
}
```

type()

From Python's perspective, dictionaries are defined as objects with the data type 'dict':

```
<class 'dict'>
```

Example

Print the data type of a dictionary:

```
thisdict = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
print(type(thisdict))
```

The dict() Constructor

It is also possible to use the dict() constructor to make a dictionary.

Example

Using the dict() method to make a dictionary:

```
thisdict = dict(name = "John", age = 36, country = "Norway")
print(thisdict)
```

DICTIONARY METHODS:

1)Python Dictionary clear() Method

ExampleGet your own Python Server

Remove all elements from the car list:

```
car = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}

car.clear()

print(car)
```

Definition and Usage

The clear() method removes all the elements from a dictionary.

2)Python Dictionary copy() Method

ExampleGet your own Python Server

Copy the car dictionary:

```
car = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}

x = car.copy()

print(x)
```

Definition and Usage

The copy() method returns a copy of the specified dictionary.

3)Python Dictionary fromkeys() Method

ExampleGet your own Python Server

Create a dictionary with 3 keys, all with the value 0:

```
x = ('key1', 'key2', 'key3')
y = 0

thisdict = dict.fromkeys(x, y)
```

print(thisdict)

Definition and Usage

The fromkeys() method returns a dictionary with the specified keys and the specified value.

4)Python Dictionary get() Method

ExampleGet your own Python Server

Get the value of the "model" item:

```
car = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}

x = car.get("model")

print(x)
```

Definition and Usage

The get() method returns the value of the item with the specified key.

5)Python Dictionary items() Method

ExampleGet your own Python Server

Return the dictionary's key-value pairs:

```
car = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}

x = car.items()

print(x)
```

Definition and Usage

The items() method returns a view object. The view object contains the key-value pairs of the dictionary, as tuples in a list.

The view object will reflect any changes done to the dictionary, see example below.

6)Python Dictionary keys() Method

ExampleGet your own Python Server

Return the keys:

```
car = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
```

```
x = car.keys()


print(x)
```

Definition and Usage

The keys() method returns a view object. The view object contains the keys of the dictionary, as a list.

The view object will reflect any changes done to the dictionary, see example below.

7)Python Dictionary pop() Method

ExampleGet your own Python Server

Remove "model" from the dictionary:

```
car = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}


car.pop("model")


print(car)
```

Definition and Usage

The pop() method removes the specified item from the dictionary.

The value of the removed item is the return value of the pop() method, see example below.

8)Python Dictionary popitem() Method

ExampleGet your own Python Server

Remove the last item from the dictionary:

```
car = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}

car.popitem()

print(car)
```

Definition and Usage

The popitem() method removes the item that was last inserted into the dictionary. In versions before 3.7, the popitem() method removes a random item.

The removed item is the return value of the popitem() method, as a tuple, see example below.

9)Python Dictionary setdefault() Method

Get the value of the "model" item:

```
car = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}

x = car.setdefault("model", "Bronco")

print(x)
```

Definition and Usage

The setdefault() method returns the value of the item with the specified key.

If the key does not exist, insert the key, with the specified value, see example below

10)Python Dictionary update() Method

ExampleGet your own Python Server

Insert an item to the dictionary:

```
car = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}
```

```
car.update({"color": "White"})


print(car)
```

Definition and Usage

The update() method inserts the specified items to the dictionary.

The specified items can be a dictionary, or an iterable object with key value pairs.

11)Python Dictionary values() Method

ExampleGet your own Python Server

Return the values:

```
car = {
  "brand": "Ford",
  "model": "Mustang",
  "year": 1964
}


x = car.values()


print(x)
```

Definition and Usage

The values() method returns a view object. The view object contains the values of the dictionary, as a list.

The view object will reflect any changes done to the dictionary, see example below.