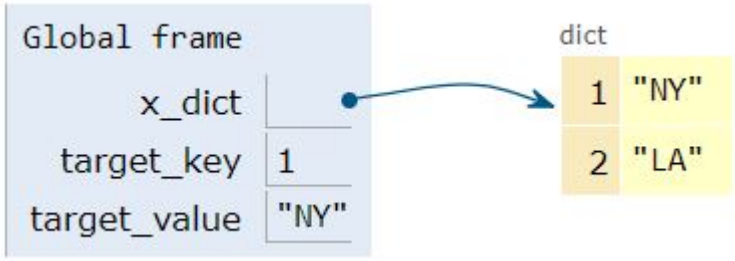
DICTIONARIES

A Python Dictionary

```
x_dict = { 1: 'NY', 2: 'LA' }
target_key = 1
target_value = x_dict[target_key]
```

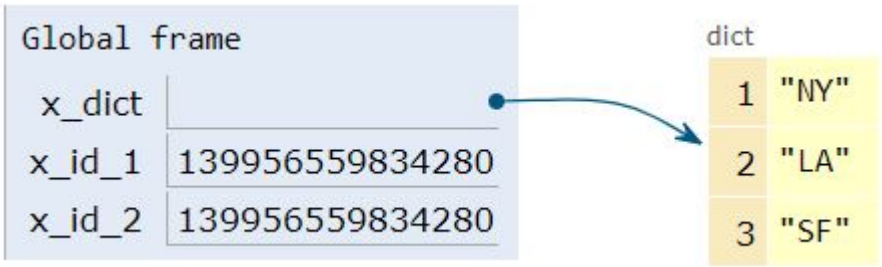


- collection of (key, value) pairs
- such pairs are called *items*
- built-in functions for keys, values and items

Mutability

```
x_dict = {1: 'NY', 2: 'LA'}
x_id_1 = id(x_dict)
```

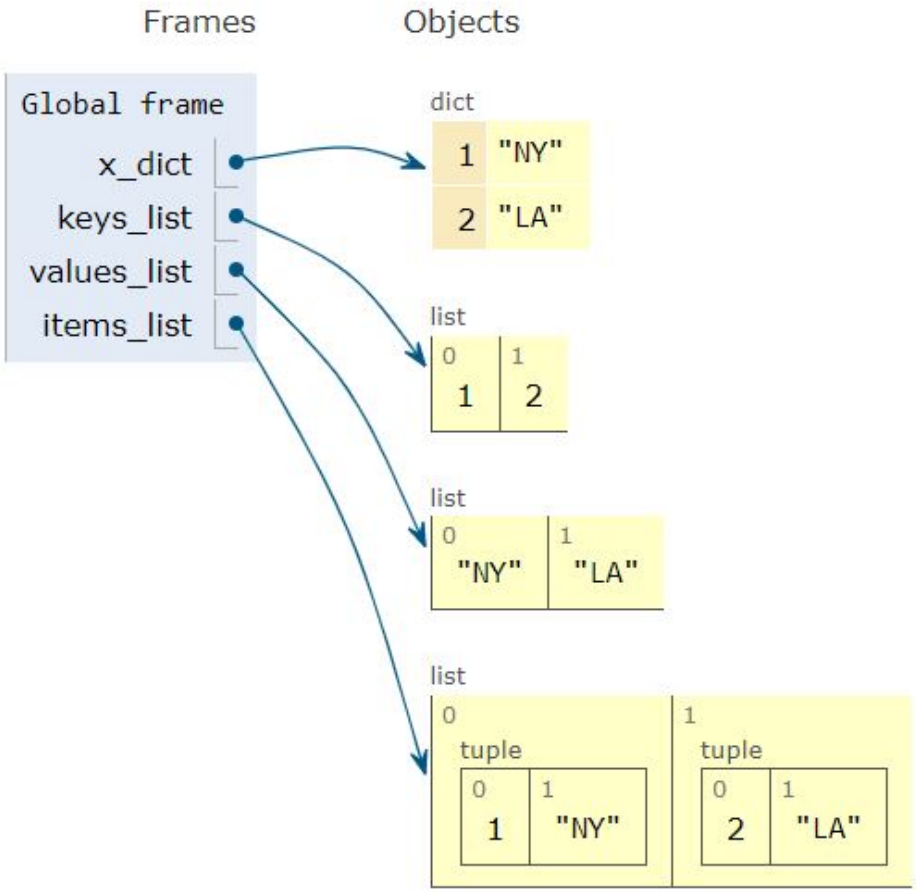

```
x_dict[3] = 'SF'
x_id_2 = id(x_dict)
```



- dictionaries are mutable

Keys, Values, Items

```
x_dict      = { 1: 'NY', 2: 'LA' }
keys_list   = list(x_dict.keys())
values_list = list(x_dict.values())
items_list  = list(x_dict.items())
```



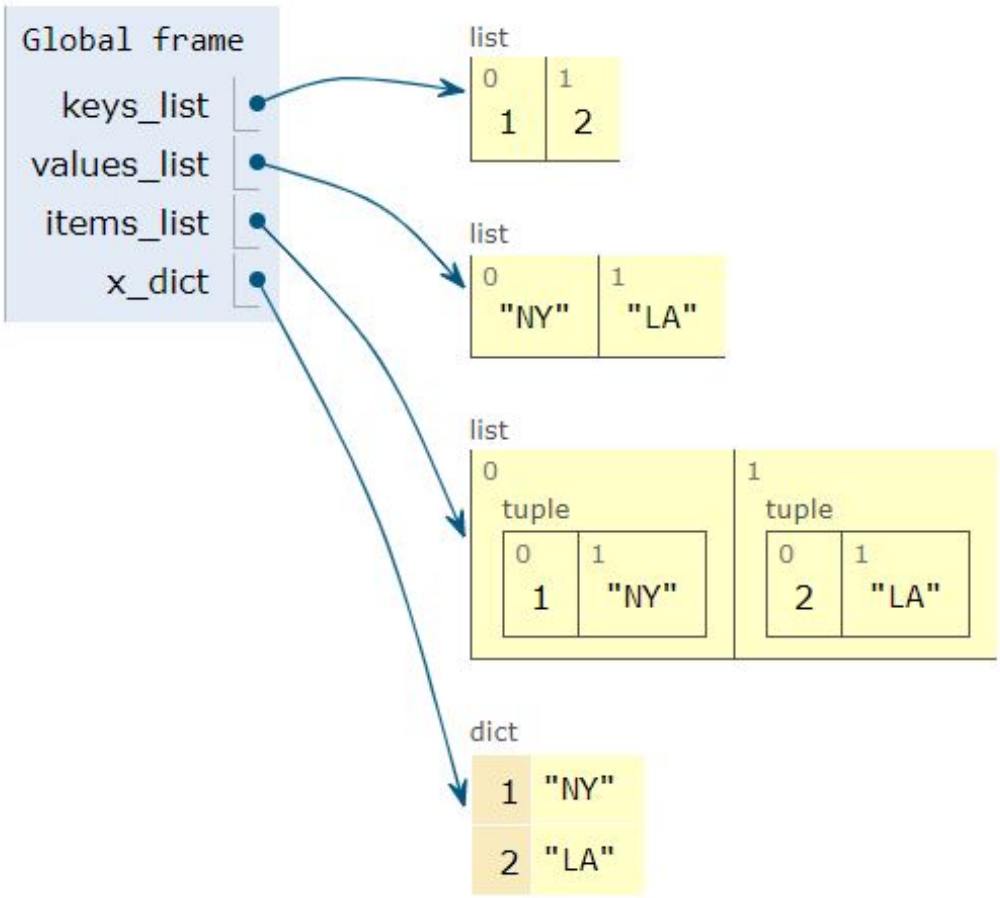
Exercise(s):

- print keys, values and items from *x_dict*:

```
x_dict={"tiger" : "cat",
        "lion"  : "cat",
        "salmon": "fish"}
```

Construction with *zip*()

```
keys_list   = [1, 2]
values_list = ['NY', 'LA']
items_list  = list(zip(keys_list, values_list))
x_dict      = dict(zip(keys_list, values_list))
```



Exercise(s):

- use *zip*() to construct the following *x_list*:

```
x_list=[("tiger","cat"),
        ("lion","cat"),
        ("salmon","fish")]
```

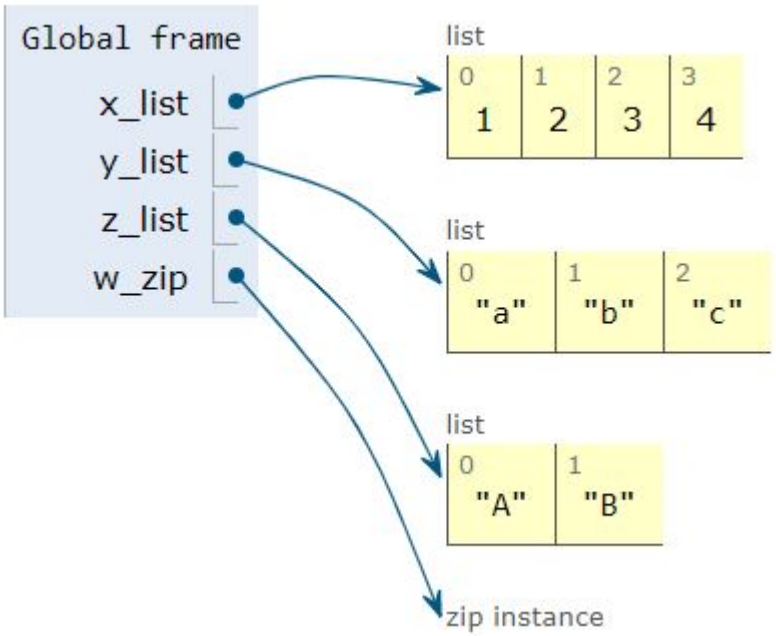
Exercise(s):

- use *zip*() to construct the following *x_dict*:

```
x_dict = {"tiger": "cat",
          "lion" : "cat",
         "salmon": "fish"}
```

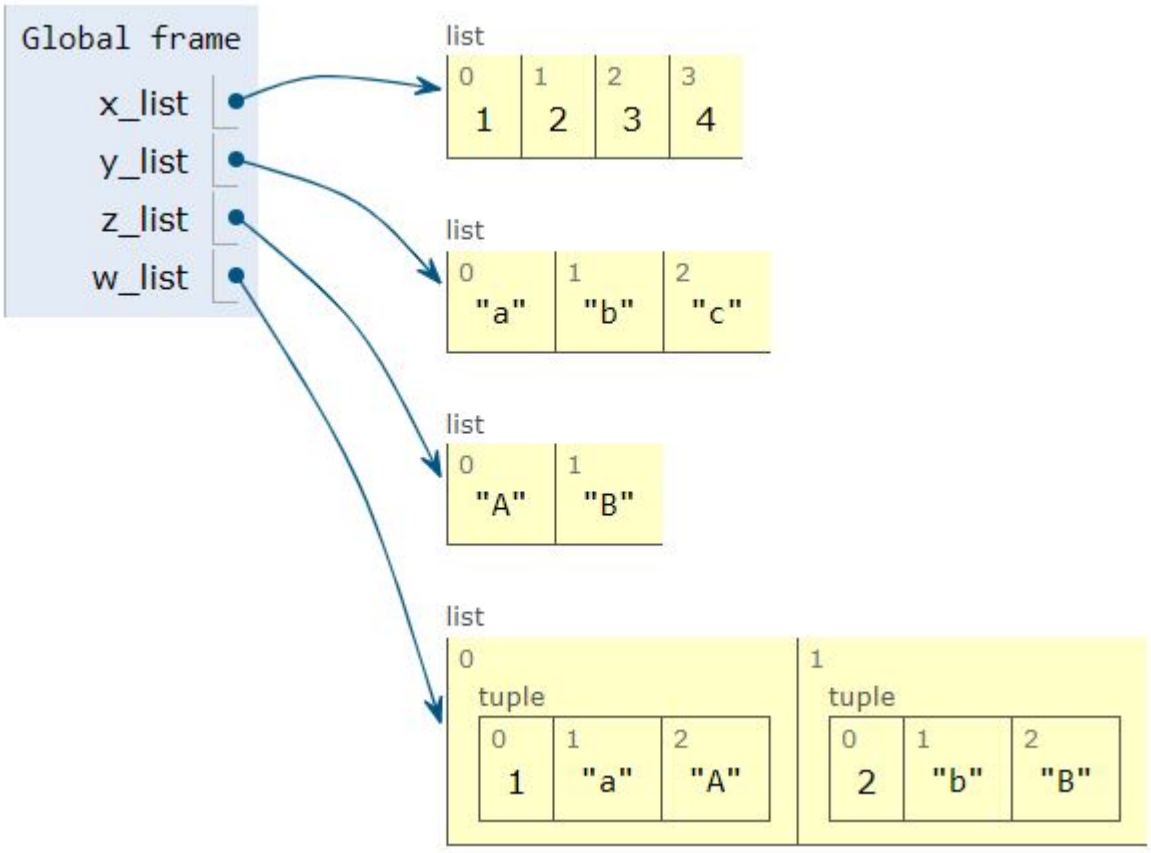
Multiple Lists with *zip*()

```
x_list = [1,2,3,4]
y_list = list('abc')
z_list = list('AB')
w_zip  = zip(x_list, y_list, z_list)
```



Multiple Lists with *zip*()

```
w_list = list(zip(x_list, y_list, z_list))
```



Exercise(s):

- use *zip*() to construct the following *x_list*:

```
x_list=[("tiger",("cat","forest")),
        ("lion", ("cat", "plains")),
        ("salmon",("fish","water"))]
```

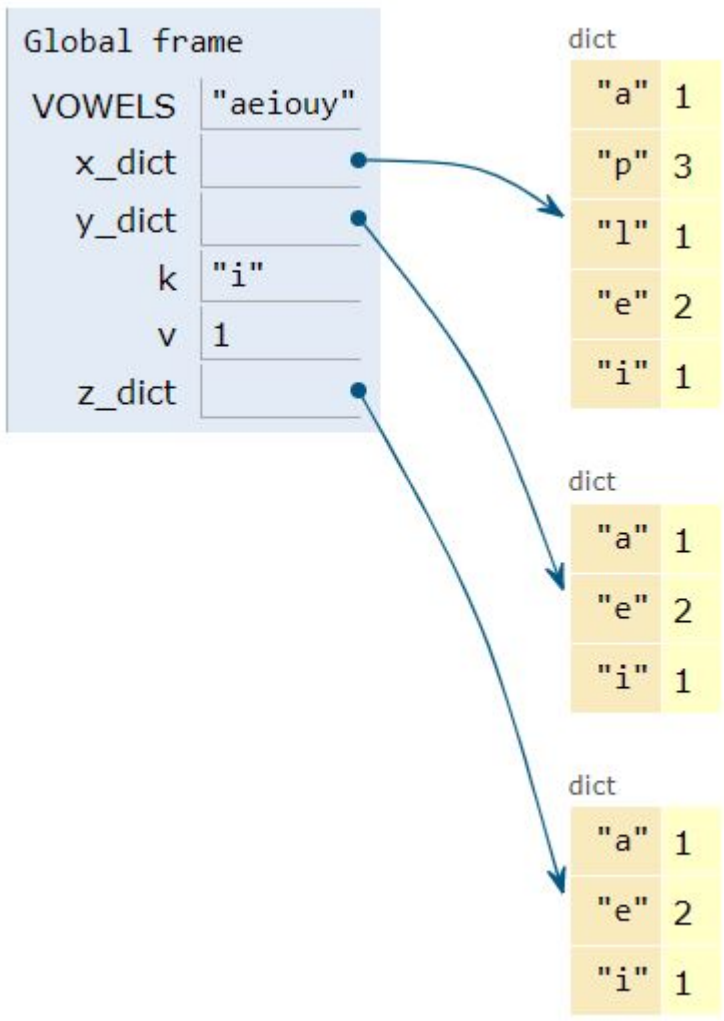
Comprehension

```
VOWELS = 'aeiouy'
x_dict = {'a':1, 'p':3, 'l' : 1, 'e': 2, 'i':1}

y_dict = dict()
for k, v in x_dict.items():
    if k in VOWELS:
        y_dict[k] = v

z_dict = {k:v for k,v in x_dict.items()
                          if k in VOWELS}
```

Comprehension



Comprehension

```
VOWELS = 'aeiouy'
x_list = ['a','p','p','l','e']
z_list = [ e for e in x_list if e in VOWELS ]

x_set  = {'a','p','p','l','e'}
z_set  = { e for e in x_set  if e in VOWELS }

x_dict = {'a':1, 'p':3, 'l' : 1, 'e': 2, 'i':1}
z_dict = {k:v for k,v in x_dict.items()
                         if k in VOWELS}
```

- mutable collections only!!!

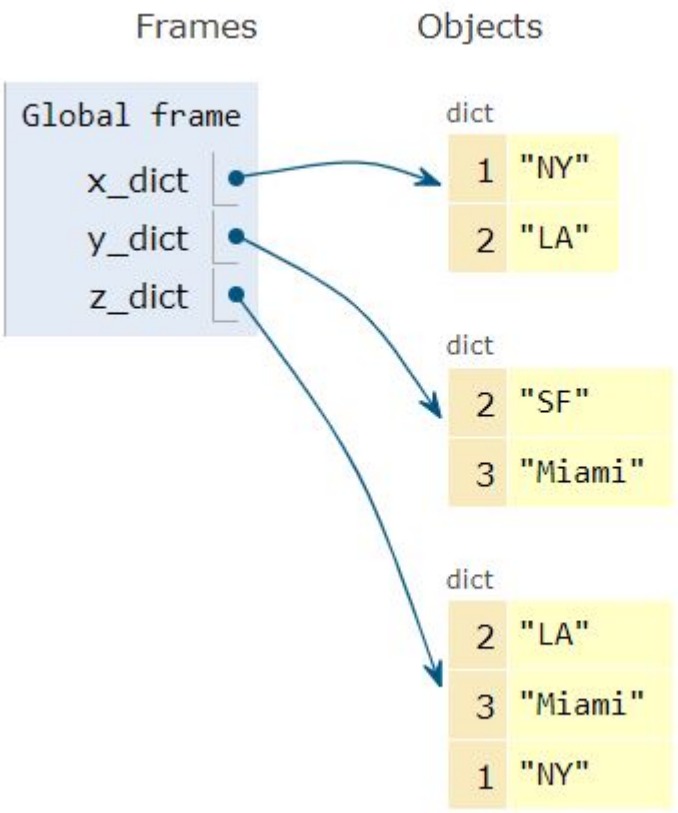
Exercise(s):

- use comprehension to construct dictionary *y_dict* from *x_dict* containg value *"cat"*:

```
x_dict = {"tiger": "cat",
          "lion": "cat",
         "salmon": "fish"}
y_dict = {"tiger": "cat",
          "lion": "cat"}
```

Examples of Methods

```
x_dict = { 1: 'NY', 2: 'LA'     }
y_dict = { 2: 'SF', 3: 'Miami' }
z_dict = { 2: 'SF', 3: 'Miami' }
z_dict.update(x_dict)   # merge two dictionaries
```



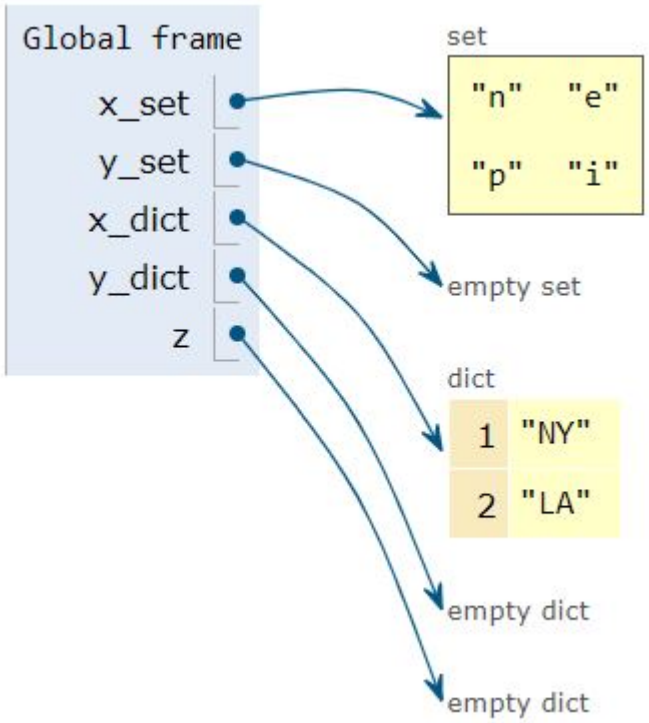
Exercise(s):

- show two ways to add item *("eagle", "bird")* to *x_dict*:

```
x_dict = {"tiger": "cat",
          "lion": "cat",
         "salmon": "fish"}
```

Empty Set & Dictionary

```
x_set  = {'p', 'i', 'n', 'e'}
y_set  = set()  # no ambiguity
x_dict = { 1: 'NY', 2: 'LA' }
y_dict = dict() # no ambiguity
z = {}          # dictionary of a set?
```

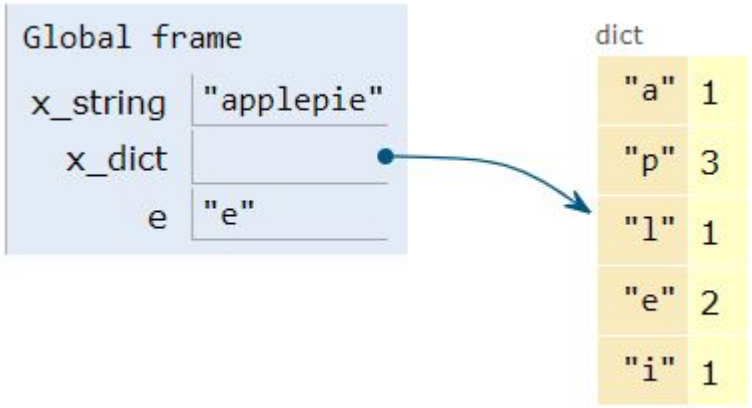


- when in doubt: set() or dict()

Example

```
# compute frequencies of letters
x_string = 'applepie'
x_dict   = dict()

for e in x_string:
    if e not in x_dict.keys():
        x_dict[e] = 1
    else:
        x_dict[e] = x_dict[e] + 1
```



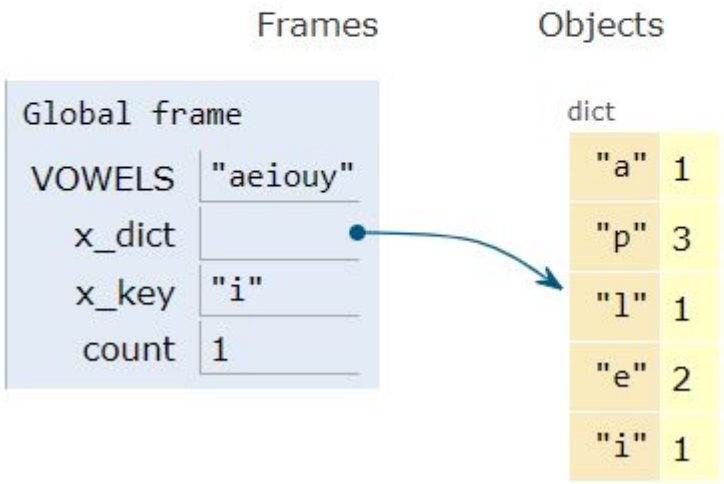
Iteration Comparison

```
VOWELS = 'aeiouy'
x_dict = {'a':1, 'p':3, 'l' : 1, 'e': 2, 'i':1}

for x_key in x_dict.keys():
    if x_key in VOWELS:
        count = x_dict[x_key]
        print(x_key, count)
```

Print output (drag lower right corner to resize)

```
a 1
e 2
i 1
```



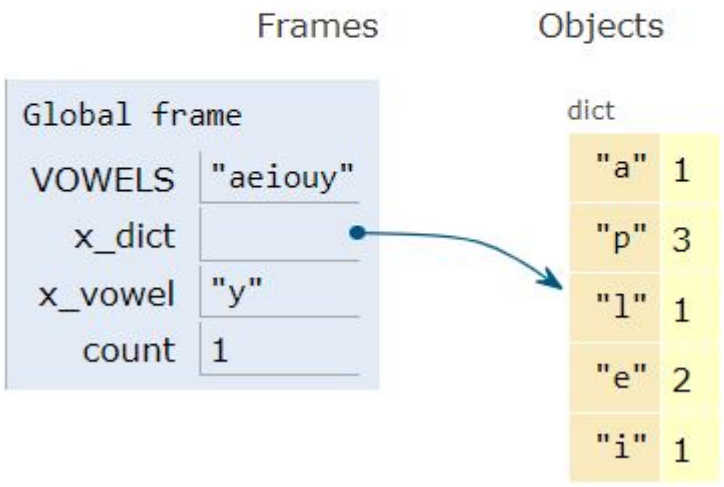
Iteration Comparison

```
VOWELS = 'aeiouy'
x_dict = {'a':1, 'p':3, 'l':1, 'e':2, 'i':1}

for x_vowel in VOWELS:
    if x_vowel in x_dict.keys():
        count = x_dict[x_vowel]
        print(x_vowel, count)
```

Print output (drag lower right corner to resize)

```
a 1
e 2
i 1
```



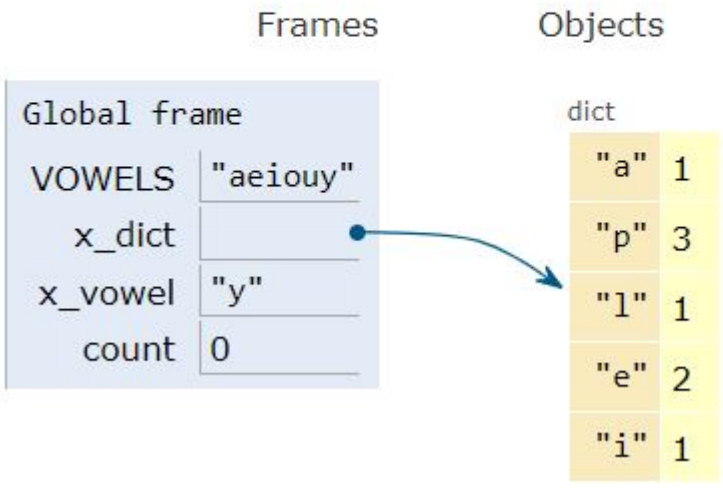
Iteration Comparison

```
VOWELS = 'aeiouy'
x_dict = {'a':1, 'p':3, 'l':1, 'e':2, 'i':1}

for x_vowel in VOWELS:
    count = x_dict.get(x_vowel, 0) # default 0
    if count > 0:
        print(x_vowel, count)
```

Print output (drag lower right corner to resize)

```
a 1
e 2
i 1
```



Code Comparison

```
VOWELS = 'aeiouy'
x_dict = {'a':1, 'p':3, 'l':1, 'e':2, 'i':1}

for x_key in x_dict.keys():          # Method 1
    if x_key in VOWELS:
        count = x_dict[x_key]
        print(x_key, count)

for x_vowel in VOWELS:               # Method 2
    if x_vowel in x_dict.keys():
        count = x_dict[x_vowel]
        print(x_vowel, count)

for x_vowel in VOWELS:               # Method 3
    count = x_dict.get(x_vowel, 0)
    if count > 0:
        print(x_vowel, count)
```

- what are the trade-offs?

Exercise(s):

- show 2 ways to print values for key *"lion"* in *x_dict*:

```
x_dict = {"tiger": "cat",
          "lion": "cat",
         "salmon": "fish"}
```

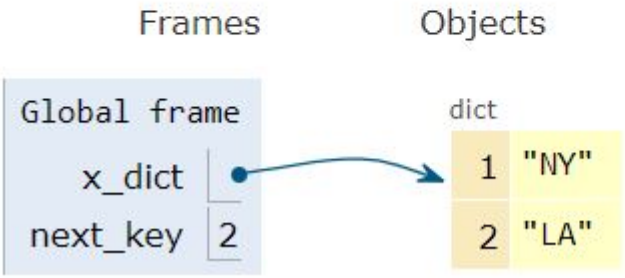
Iteration

```
# print dictionary keys and values
x_dict = {1: 'NT', 2: 'LA'}

for next_key in x_dict.keys():
    print(next_key, x_dict[next_key])
```

Print output (drag lower right corner to resize)

```
1 NY
2 LA
```



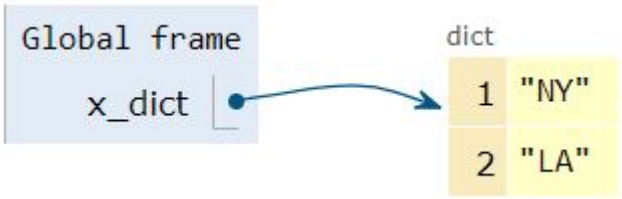
Exercise(s):

- print all keys in *x_dict* for value *"cat"*:

```
x_dict = {"tiger": "cat",
          "lion": "cat",
         "salmon": "fish"}
```

Dictionary Methods

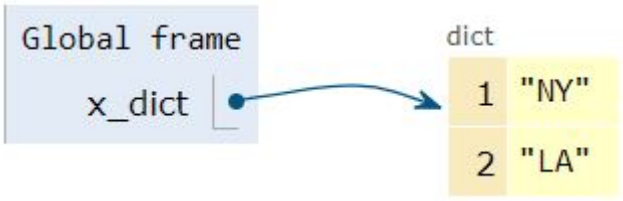
```
x_dict = {1: 'NY', 2: 'LA'}
```



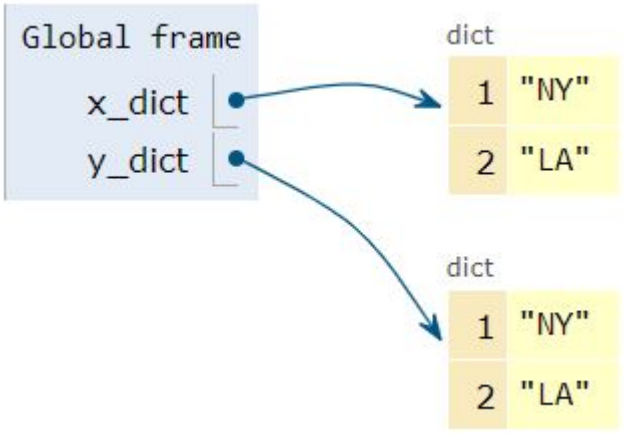

```
x_dict.clear()
```

Dictionary Methods

```
x_dict = {1: 'NY', 2: 'LA'}
```

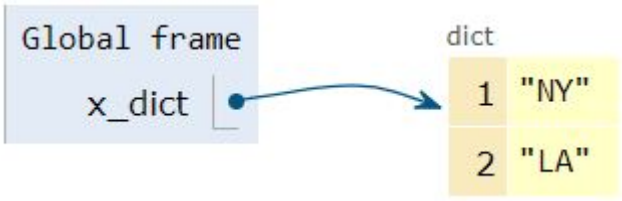



```
y_dict = x_dict.copy()      # shallow copy
```



Dictionary Methods

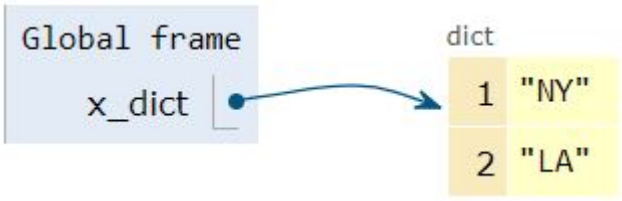
```
x_dict = {1: 'NY', 2: 'LA'}
```



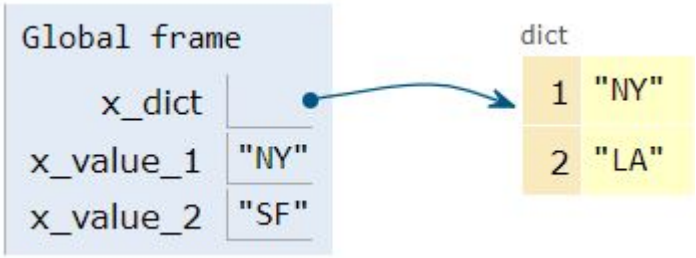

```
y_dict = dict.fromkeys([1, 2],'SF') # same value
```

Dictionary Methods

```
x_dict = {1: 'NY', 2: 'LA'}
```

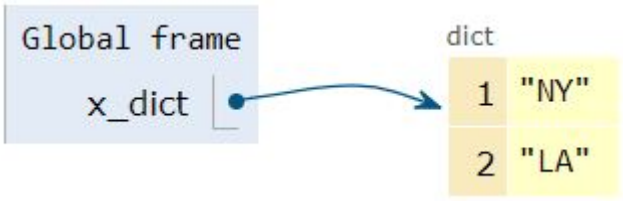



```
x_value_1 = x_dict.get(1, 'SF')   # existing key
x_value_2 = x_dict.get(3, 'SF')   # use default
```

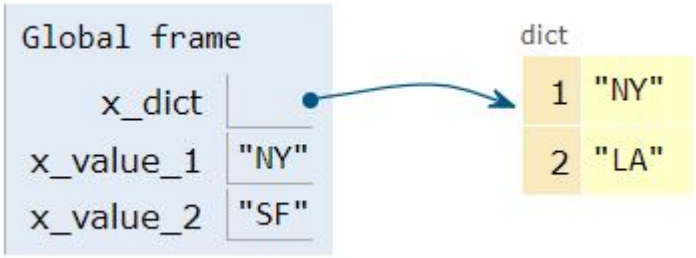


Dictionary Methods

```
x_dict = {1: 'NY', 2: 'LA'}
```

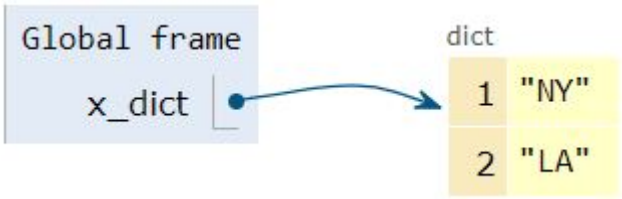



```
x_value_1 = x_dict.get(1, 'SF')   # existing key
x_value_2 = x_dict.get(3, 'SF')   # use default
```

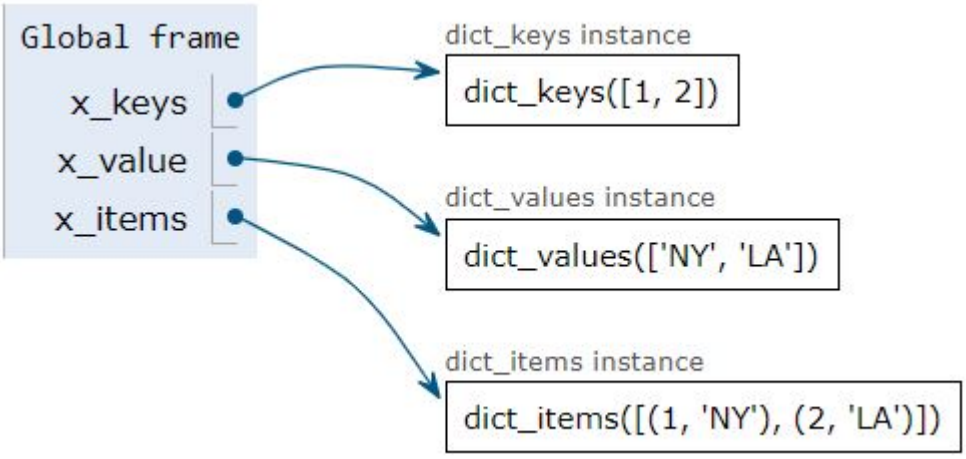


Dictionary Methods

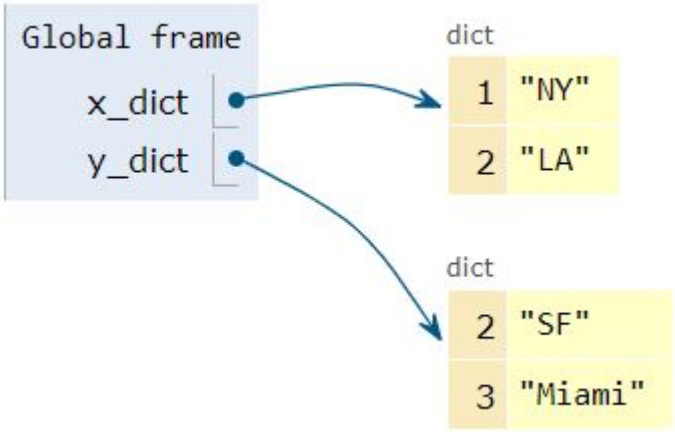
```
x_dict = {1: 'NY', 2: 'LA'}
```



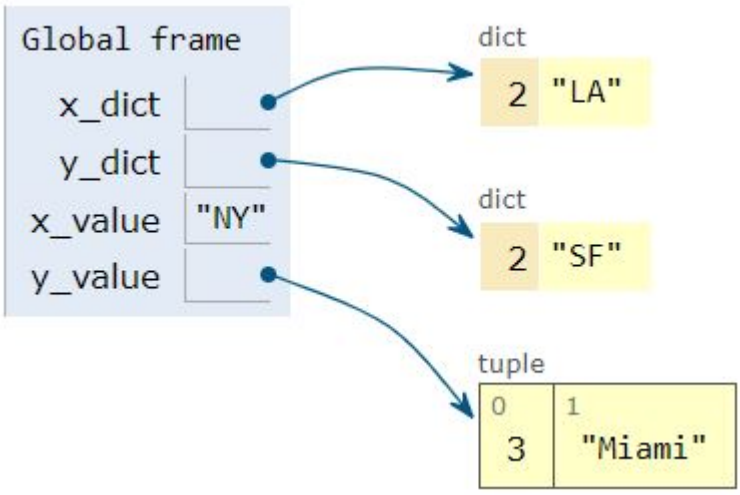

```
x_keys   = x_dict.keys()
x_values = x_dict.values()
x_items  = x_dict.items()
```



Dictionary Methods




```
x_value  = x_dict.pop(1)
y_value  = y_dict.popitem()    # last inserted
```



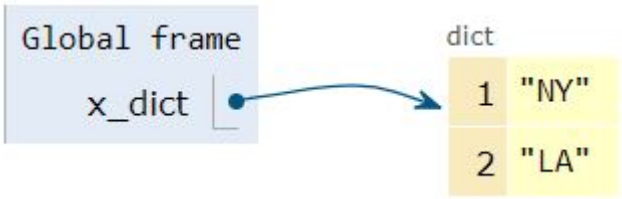
Exercise(s):

- remove item with key *"lion"* from *x_dict*:

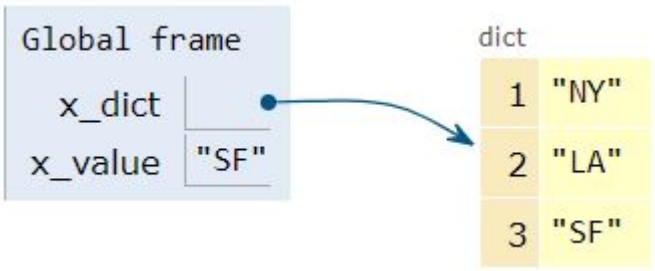
```
x_dict={"tiger" : "cat",
        "lion"  : "cat",
        "salmon": "fish"}
```

Dictionary Methods

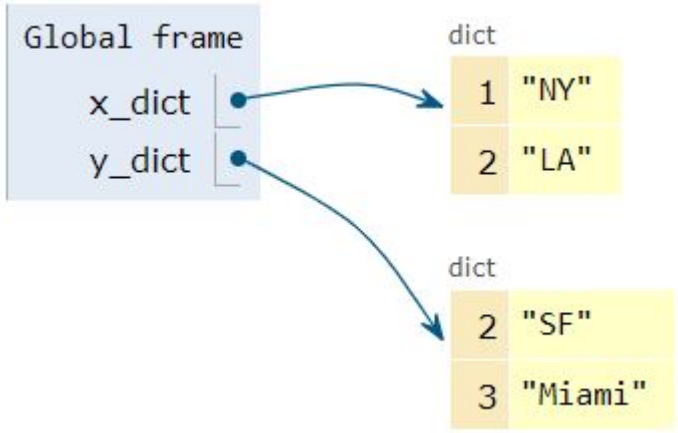
```
x_dict = {1: 'NY', 2: 'LA'}
```



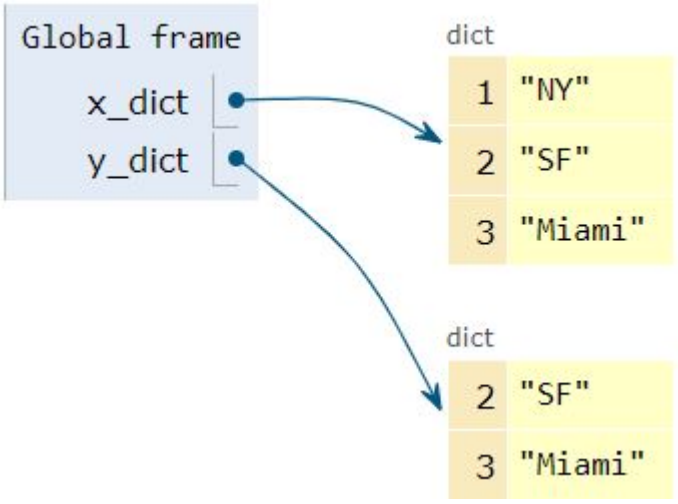

```
x_value = x_dict.setdefault(3, 'SF')
```



Dictionary Methods




```
x_dict.update(y_dict)
```



Summary:

- collection of (*key*, *value*) pairs
- iterable and mutable
- unique, immutable and hashable elements for keys
- no restrictions for values
- many methods