

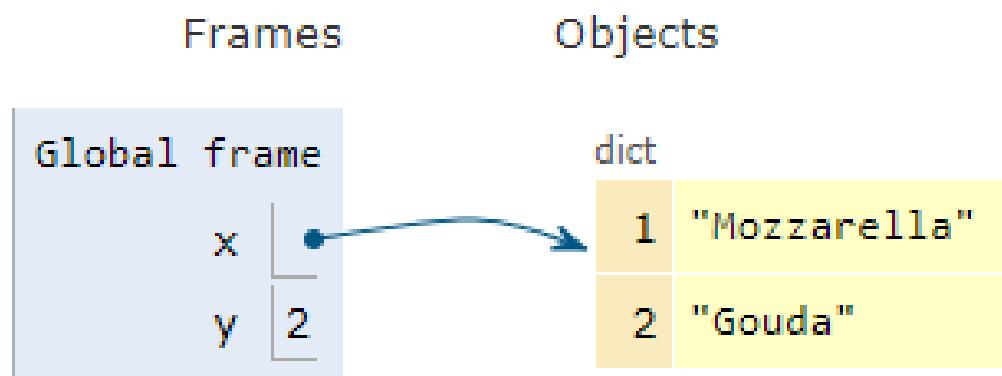
# Python

# Dictionary

# Python Dictionary

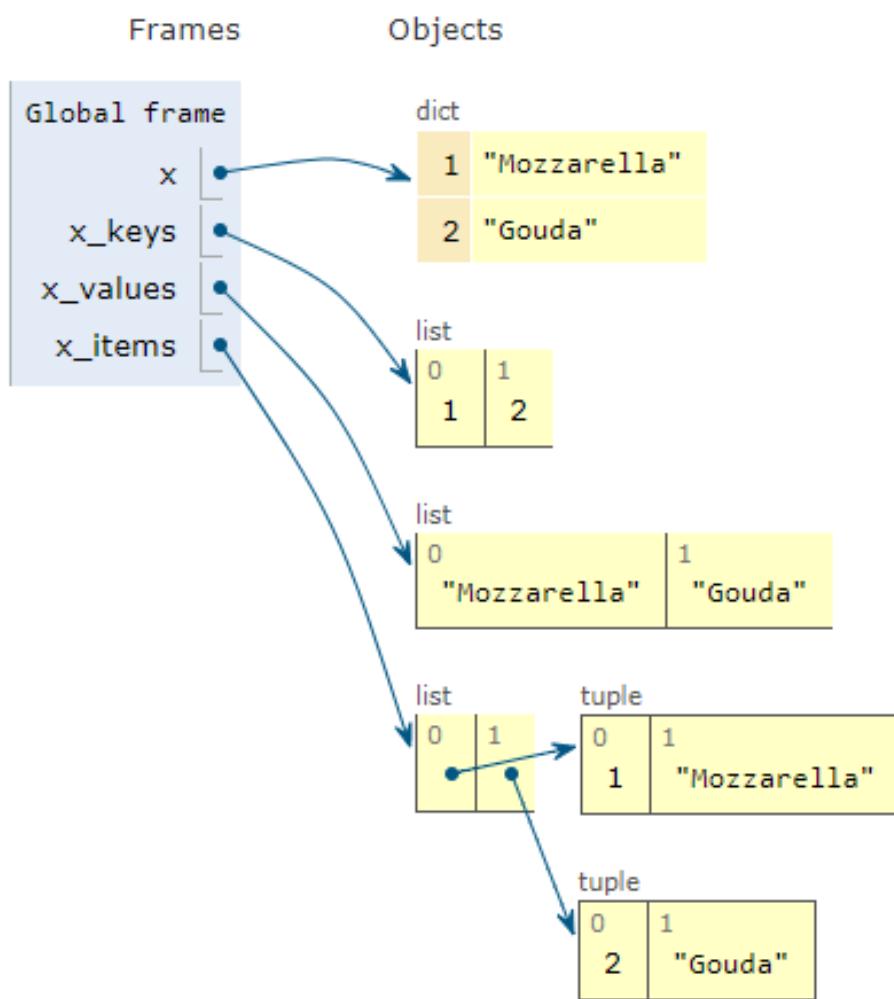
- unordered collection of items
- compound data type: {key: value}
- keys are unique (values may be not)
- keys are immutable type (strings, tuples, numbers)
- optimized for data retrieval by key

```
>>> x = dict() # for empty dictionary  
>>> x = {1: 'Mozzarella', 2: 'Gouda'}  
>>> y = len(x)
```



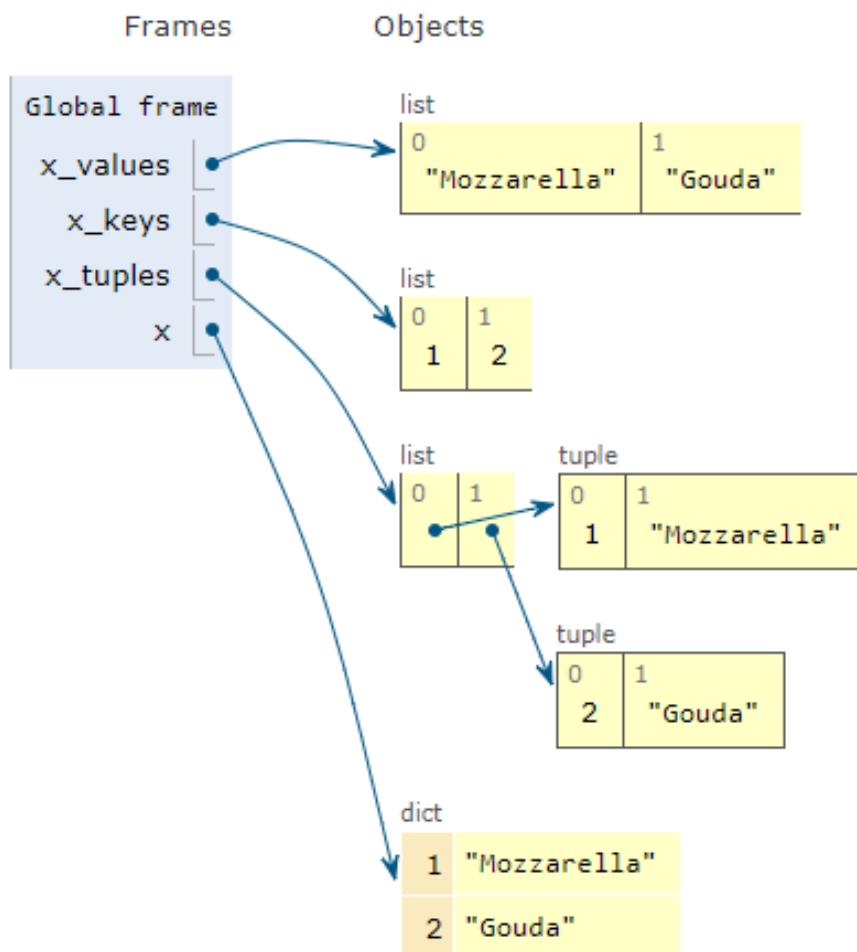
# Dictionary *keys()*, *values()*, *items()*

```
>>> x = {1: 'Mozzarella', 2: 'Gouda'}
>>> x_keys = x.keys()
>>> x_values = x.values()
>>> x_items = x.items()
```



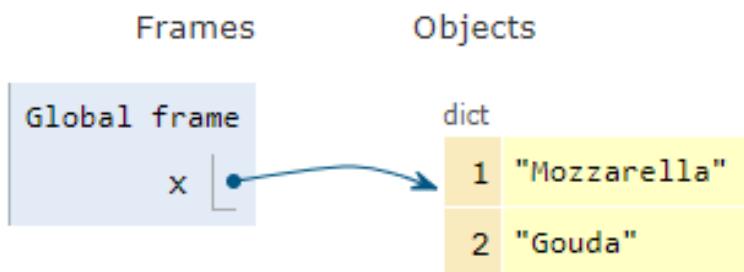
# Creating Dictionary from Keys and Values

```
>>> x_values = ['Mozzarella', 'Gouda']
>>> x_keys = [1, 2]
>>> x_tuples = zip(x_keys, x_values)
>>> x = dict(x_tuples)
```

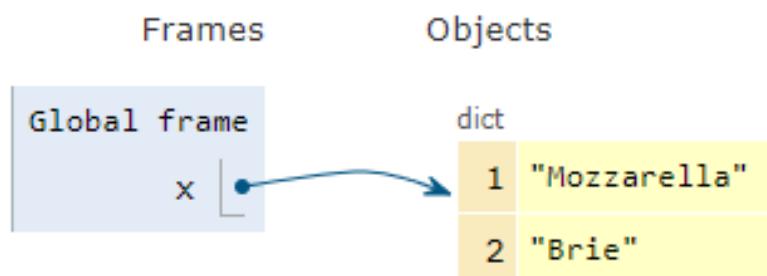


# Dictionary Updating

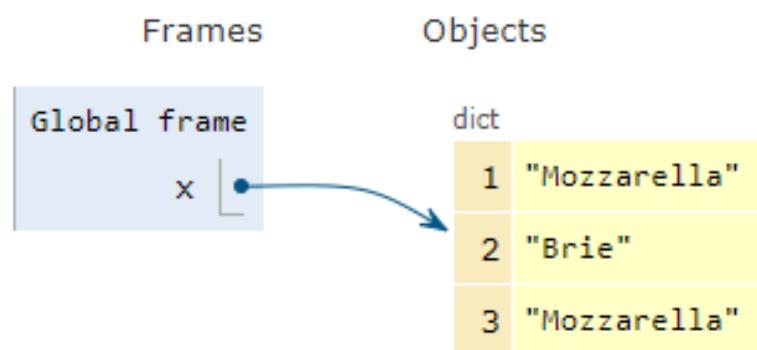
```
>>> x = {1: 'Mozzarella', 2: 'Gouda']}
```



```
>>> x[2] = 'Brie'          # existing entry
```

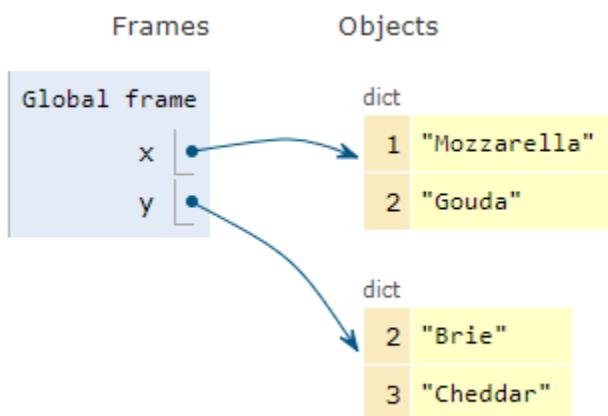


```
>>> x[3] = 'Mozzarella' # many-to-one
```

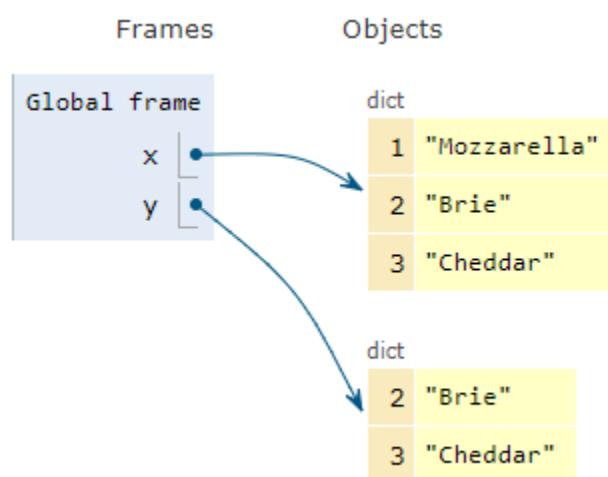


# Dictionary Updating with Another Dictionary

```
>>> x = {1: 'Mozzarella', 2: 'Gouda'}  
>>> y = {2: 'Brie', 3: 'Cheddar'}
```

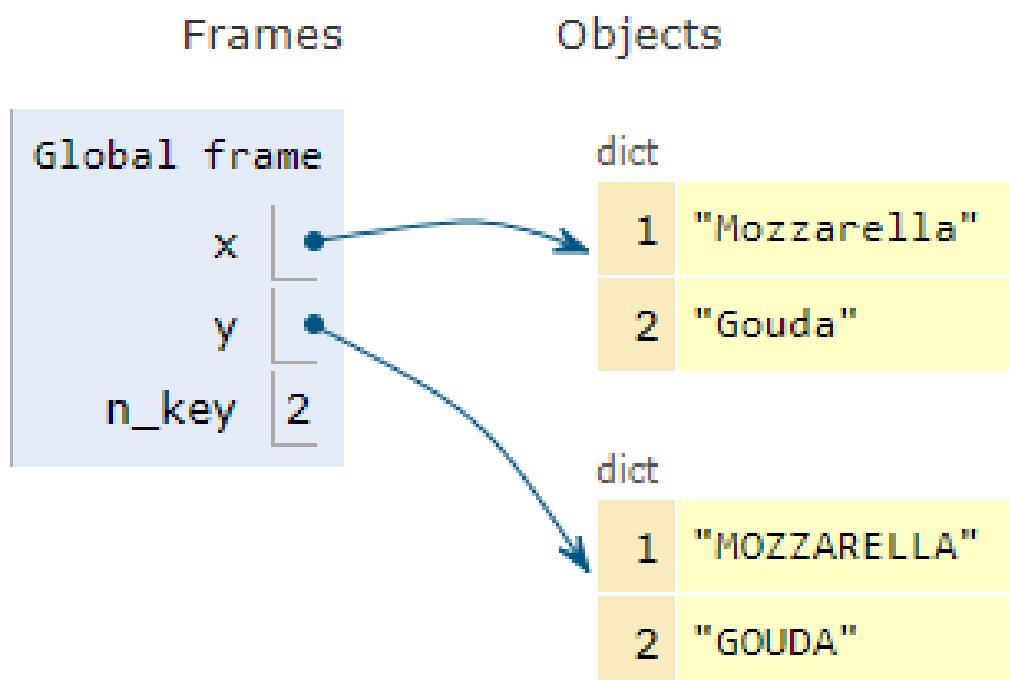


```
>>> x.update(y)
```



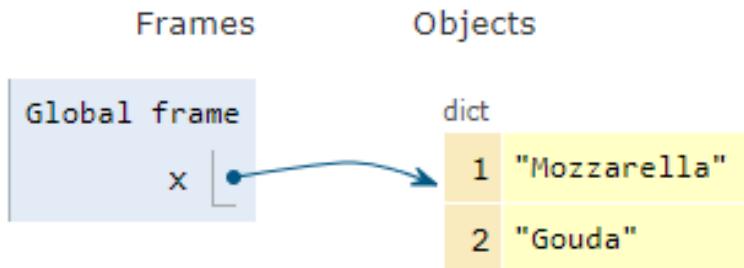
# Dictionary Iteration

```
>>> # capitalize all dictionary values  
>>> x = {1: 'Mozzarella', 2: 'Gouda'}  
>>> y = dict()  
>>> for n_key in x.keys():  
>>>     y[n_key] = x[n_key].upper()
```

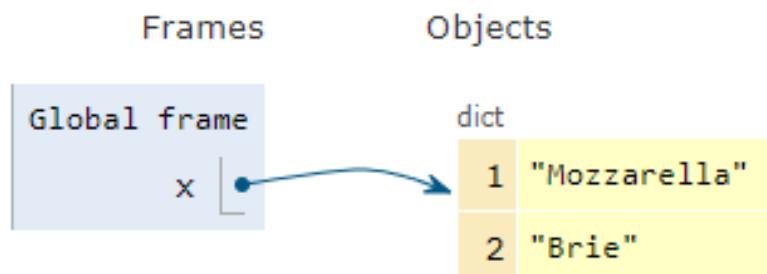


# Dictionary Additions

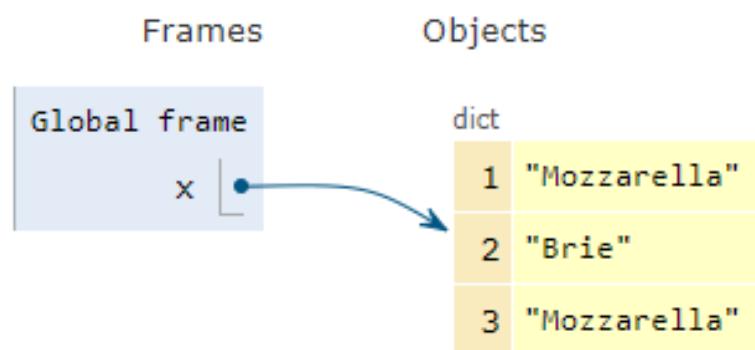
```
>>> x = {1: 'Mozzarella', 2: 'Gouda'}
```



```
>>> x[2] = 'Brie' # removes old value
```



```
>>> x[3] = 'Mozzarella' # many-to-one
```



# *dict* Type Methods

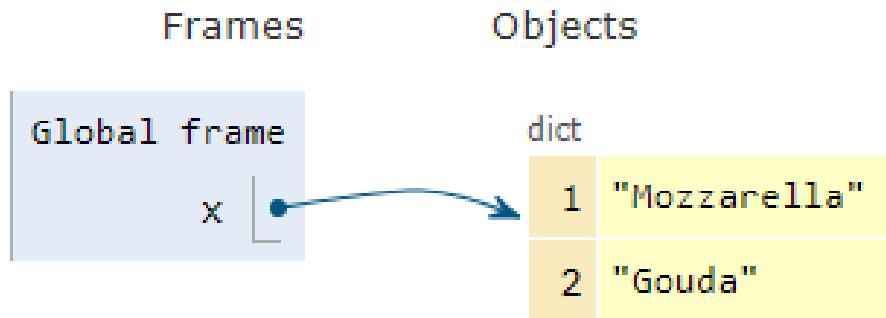
>>> **dir(set)**

The screenshot shows a Jupyter Notebook interface with a Python 1 kernel. The console tab is active, displaying the output of the command `dir(dict)`. The output lists numerous methods and attributes of the `dict` type, including `__class__`, `__cmp__`, `__contains__`, `__delattr__`, `__delitem__`, `__doc__`, `__eq__`, `__format__`, `__ge__`, `__getattribute__`, `__getitem__`, `__gt__`, `__hash__`, `__init__`, `__iter__`, `__le__`, `__len__`, `__lt__`, `__ne__`, `__new__`, `__reduce__`, `__reduce_ex__`, `__repr__`, `__setattr__`, `__setitem__`, `__sizeof__`, `__str__`, `__subclasshook__`, and various methods for clearing, copying, and iterating over the dictionary.

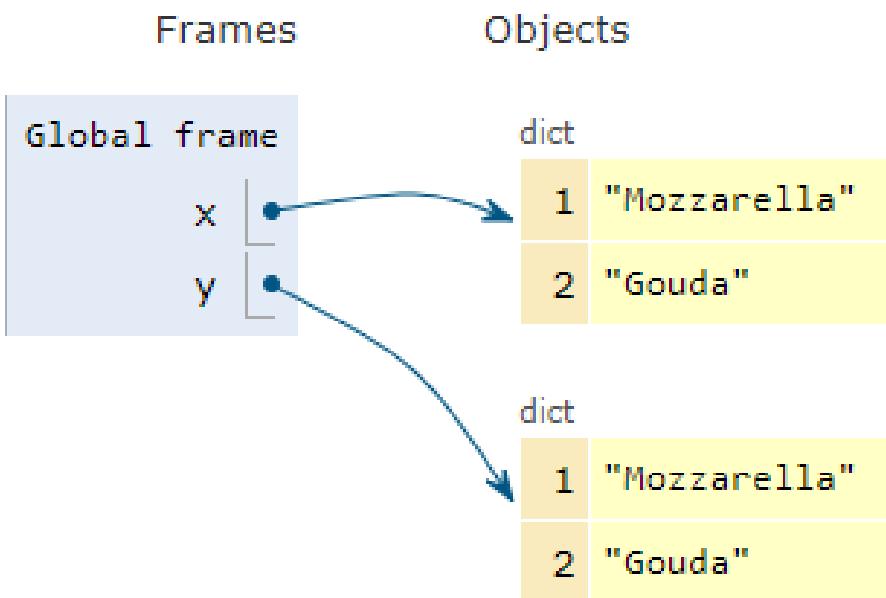
>>> **{1: 'Mozzarella', 2: 'Brie'}.keys()**  
**[1, 2]**  
>>> **{1: 'Mozzarella', 2: 'Brie'}.viewvalues()**  
**dict\_values(['Mozzarella', 'Brie'])**

# Dictionary: *copy()*

```
>>> x = {1: 'Mozzarella', 2: 'Gouda'}
```

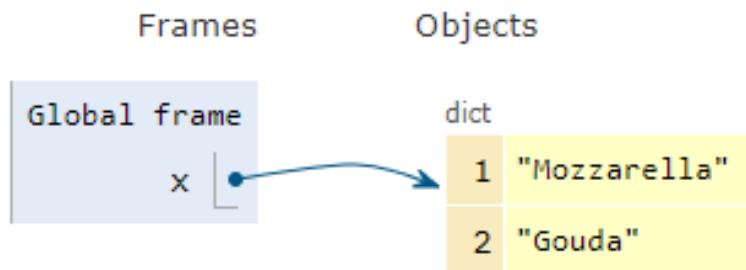


```
>>> y = x.copy()
```

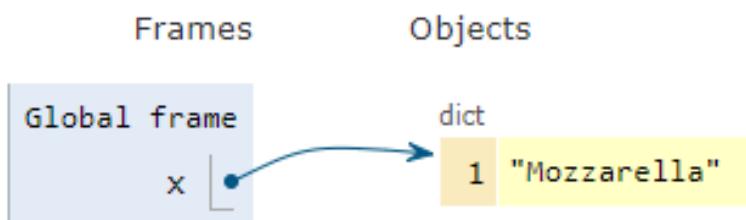


# Dictionary: *del()*, *clear()*

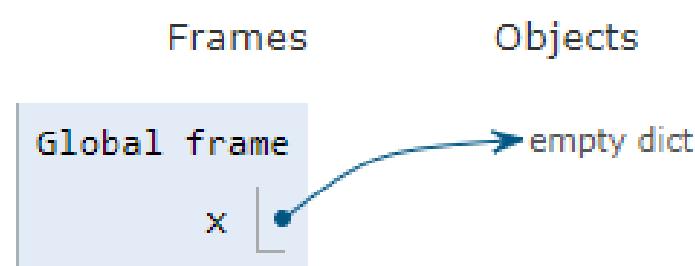
```
>>> x = {1: 'Mozzarella', 2: 'Gouda'}
```



```
>>> del x[2]
```



```
>>> x.clear() # remove all values
```

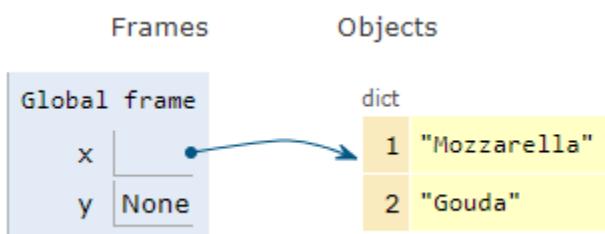


# Dictionary: *get()*

- alternative to square brackets
- returns *None* instead of *KeyError*

```
>>> x = {1: 'Mozzarella', 2: 'Gouda'}
```

```
>>> y = x.get(3)
```



```
>>> # need to check key membership
```

```
>>> y = x[3]
```

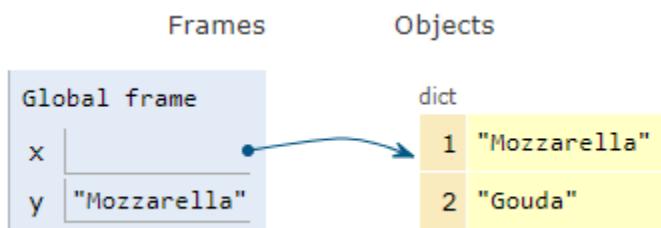
```
Console
Python 1 IP: Kernel 1
>>>
>>>
>>> x = {1: 'Mozzarella', 2: 'Gouda'}
>>> y=x[3]
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
KeyError: 3
>>>
```

Encoding: UTF-8    Line: 1    Column: 1    Memory: 44 %

# Accessing Dictionary Values

- use square brackets

```
>>> x = {1: 'Mozzarella', 2: 'Gouda'}  
>>> y = x[1]
```



```
>>> # need to check key membership  
>>> y = x[3]
```

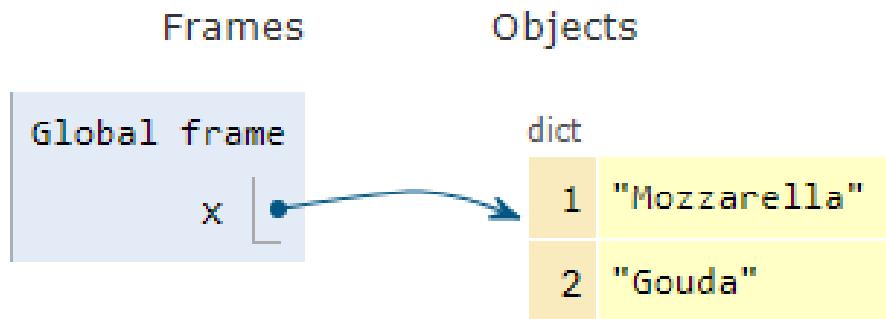
```
Console  
Python 1 IP: Kernel 1  
>>>  
>>>  
>>> x = {1: 'Mozzarella', 2: 'Gouda'}  
>>> y=x[3]  
Traceback (most recent call last):  
  File "<stdin>", line 1, in <module>  
KeyError: 3  
>>>
```

The screenshot shows a Jupyter Notebook console window. The user has defined a dictionary x with keys 1 and 2. When they try to access x[3], a KeyError is raised because there is no key 3 in the dictionary. The console also shows the current kernel status as 'Kernel 1'.

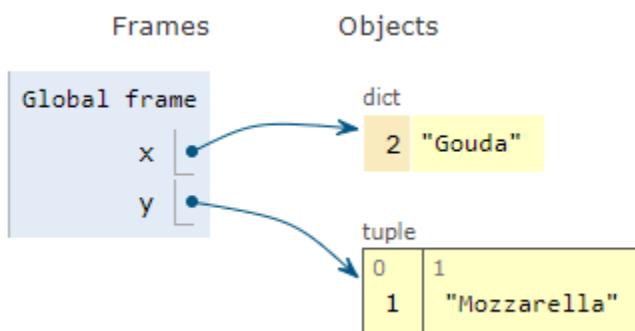
# Dictionary: *popitem()*

- removes an arbitrary *key:value* pair
- use *pop()* for a specific pair

```
>>> x = {1: 'Mozzarella', 2: 'Gouda']}
```



```
>>> y = x.popitem()
```



# Review Problems

# Interview Problem

- when are two dictionaries  $A$  and  $B$  considered equal?

# Interview Problem

- what are list and dictionary comprehensions?

# Interview Problem

- how to create a dictionary which can preserve the order of pairs?

# Interview Problem

- when is a dictionary used instead of a list?

# Interview Problem

- explain the difference between *list* and *tuple* in terms of:
  - (1) syntax
  - (2) mutability
  - (3) size
  - (4) performance
  - (5) usage

# Interview Problem

- explain the difference between *list* and *dictionary* in terms of:
  - (1) syntax
  - (2) referencing (indexing)
  - (3) ordering
  - (4) hashing

# Interview Problem

- what datatype would you use to store first and last names of employees

# Interview Problem

- iterate over a list of words and compute the frequency of each word

# Programming Exercise

The screenshot shows a software interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' on the left, 'Workbench' in the center, and a help icon ('?') and settings icon ('⚙️') on the right. Below the title, it says 'Exercise 51265 —'. There are two tabs: 'WORK AREA' (which is selected) and 'SOLUTIONS'. A green button labeled 'Content Support' is visible. The main area contains the following text:

Given a dictionary `d`, create a new dictionary that reverses the keys and values of `d`. Thus, the keys of `d` become the values of the new dictionary and the values of `d` become the keys of the new dictionary. You may assume `d` contains no duplicate values (that is, no two keys map to the same values.) Associate the new dictionary with the variable inverse.

# **Programming Exercise**

## **Worksheet**

**this page is intentionally left blank**

# Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' in green rounded rectangles, followed by a 'Workbench' button in blue and red, and a help icon with a question mark and a gear icon. Below these are two tabs: 'WORK AREA' and 'SOLUTIONS'. A 'Content Support' button is located above a large text area. The main text area contains the following problem statement:

Given the dictionary, `d`, find the largest key in the dictionary and associate the corresponding value with the variable `val_of_max`. For example, given the dictionary `{5:3, 4:1, 12:2}`, 2 would be associated with `val_of_max`. Assume `d` is not empty.

# **Programming Exercise**

## **Worksheet**

**this page is intentionally left blank**

# Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' on the left, 'Workbench' in the center, and a question mark icon and a gear icon on the right. Below these is the title 'Exercise 51282'. Underneath the title are two tabs: 'WORK AREA' and 'SOLUTIONS', with 'WORK AREA' being the active tab. In the 'WORK AREA' section, there is a green button labeled 'Content Support'. The main content area contains the following text:

The *inverse* of a map is a new map where the values of the original map become the keys of the new map and the keys become the values. For example, given the map `{1:2, 3:4}`, the inverse is the map `{2:1, 4:3}`. Given the map, `d`, create the inverse of `d`. Associate the new map with the variable `inverse`. You may assume that there are no duplicate values in `d` (that is no two keys in `d` map to the same value).

# **Programming Exercise**

## **Worksheet**

**this page is intentionally left blank**

# Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' on the left, 'Workbench' in the center, and a help icon ('?') and settings icon ('⚙️') on the right. Below the navigation is the title 'Exercise 51299 —'. Underneath the title are two tabs: 'WORK AREA' (which is selected) and 'SOLUTIONS'. In the bottom right corner of the main area, there is a button labeled 'Content Support'. The main content area contains the following text:

Given dictionaries,  $d_1$  and  $d_2$ , create a new dictionary with the following property: for each entry  $(a, b)$  in  $d_1$ , if there is an entry  $(b, c)$  in  $d_2$ , then the entry  $(a, c)$  should be added to the new dictionary. For example, if  $d_1$  is  $\{2:3, 8:19, 6:4, 5:12\}$  and  $d_2$  is  $\{2:5, 4:3, 3:9\}$ , then the new dictionary should be  $\{2:9, 6:3\}$ . Associate the new dictionary with the variable  $d_3$ .

# **Programming Exercise**

## **Worksheet**

**this page is intentionally left blank**

# Programming Exercise

The screenshot shows a software interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT'. To the right of these are the words 'Workbench' and icons for help (?) and settings (gear). Below the title, it says 'Exercise 51300 —'. There are two tabs: 'WORK AREA' (which is selected) and 'SOLUTIONS'. In the 'WORK AREA' section, there is a green button labeled 'Content Support'. The main content area contains the following text:

Given dictionaries,  $d_1$  and  $d_2$ , create a new dictionary with the following property: for each entry  $(a, b)$  in  $d_1$ , if  $a$  is not a key of  $d_2$  (i.e., not  $a \in d_2$ ) then add  $(a,b)$  to the new dictionary for each entry  $(a, b)$  in  $d_2$ , if  $a$  is not a key of  $d_1$  (i.e., not  $a \in d_1$ ) then add  $(a,b)$  to the new dictionary. For example, if  $d_1$  is  $\{2:3, 8:19, 6:4, 5:12\}$  and  $d_2$  is  $\{2:5, 4:3, 3:9\}$ , then the new dictionary should be  $\{8:19, 6:4, 5:12, 4:3, 3:9\}$ . Associate the new dictionary with the variable d3.

# **Programming Exercise**

## **Worksheet**

**this page is intentionally left blank**

# Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top is a toolbar with 'PREV' and 'NEXT' buttons, a 'Workbench' tab, and a help icon. Below the toolbar is the title 'Exercise 51302 —'. Underneath the title are two tabs: 'WORK AREA' and 'SOLUTIONS'. A 'Content Support' button is located below the tabs. The main content area contains the following text:

Given a dictionary `d` and a list `lst`, remove all elements from the dictionary whose key is an element of `lst`. For example, given the dictionary `{1:2, 3:4, 5:6, 7:8}` and the list `[1, 7]`, the resulting dictionary would be `{3:4, 5:6}`. Assume every element of the list is a key in the dictionary.

# **Programming Exercise**

## **Worksheet**

**this page is intentionally left blank**

# Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' on the left, 'Workbench' in the center, and a question mark icon and gear icon on the right. Below these is the title 'Exercise 51815 —'. Underneath the title is a toolbar with two tabs: 'WORK AREA' (which is selected) and 'SOLUTIONS'. A green button labeled 'Content Support' is located above the main content area. The main content area contains the following text: 'Write a **statement** that associates **d** with an empty dictionary.'

# **Programming Exercise**

## **Worksheet**

**this page is intentionally left blank**

# Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' on the left, 'Workbench' in the center, and a question mark icon and gear icon on the right. Below these is the title 'Exercise 51816 —'. Underneath the title are two tabs: 'WORK AREA' and 'SOLUTIONS', with 'WORK AREA' being the active tab. A green button labeled 'Content Support' is located above the main content area. The main content area contains the following text:

Write a **statement** that associates **d** with a one-entry dictionary that maps the **str** 'answer' to the **int** value 42.

# **Programming Exercise**

## **Worksheet**

**this page is intentionally left blank**

# Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' in green rounded rectangles, followed by a 'Workbench' button in blue, and a help icon ('?') and settings icon ('⚙️'). Below these are two tabs: 'WORK AREA' and 'SOLUTIONS'. A 'Content Support' button is located in the top right corner of the main content area. The main content area contains the following text:

Given that `d` refers to a dictionary,  
write an expression that is the  
value to which the dictionary maps  
the key 'answer'.

# **Programming Exercise**

## **Worksheet**

**this page is intentionally left blank**

# Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' on the left, 'Workbench' in the center, and a help icon (?) and settings gear icon on the right. Below these is the title 'Exercise 51818 —'. Underneath the title are two tabs: 'WORK AREA' and 'SOLUTIONS'. A green button labeled 'Content Support' is located below the tabs. The main area contains the following text:

Given that `d` refers to a dictionary,  
change the value mapped to by the  
key 'Monty' to 'Python'.

# **Programming Exercise**

## **Worksheet**

**this page is intentionally left blank**

# Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' on the left, 'Workbench' in the center, and a question mark icon and a gear icon on the right. Below these are two tabs: 'WORK AREA' and 'SOLUTIONS'. A green button labeled 'Content Support' is located near the bottom of the tabs. The main area contains the following text:

Given that `d` refers to a dictionary and that `x` has been defined, delete the dictionary entry whose key is equal to the value associated with

# **Programming Exercise**

## **Worksheet**

**this page is intentionally left blank**

# Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' on the left, 'Workbench' in the center, and a help icon with a gear symbol on the right. Below these are two tabs: 'WORK AREA' (highlighted in dark blue) and 'SOLUTIONS' (in light blue). A green button labeled 'Content Support' is located below the tabs. The main area contains the following text:

Assume there is a variable,  
nobel\_peace\_prizes, that is associated with a  
dictionary that maps years to winners of the  
Nobel Peace Prize for that year and assume it  
is up-to-date through the year 2005. Write a  
statement that adds an entry that maps the  
key 2006 to "Muhammad Yunus and  
Grameen Bank".

# **Programming Exercise**

## **Worksheet**

**this page is intentionally left blank**

# Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' in green rounded rectangles, followed by a 'Workbench' button in blue and orange, and a help/gear icon. Below these are two tabs: 'WORK AREA' and 'SOLUTIONS'. A 'Content Support' button is located at the top right of the main content area. The main content area contains the following text:

Assume there is a variable,  
album\_artists, that is associated with  
a dictionary that maps albums to  
performing artists. Write a statement  
that inserts the key/value pair: "Live  
It Out"/"Metric".

# **Programming Exercise**

## **Worksheet**

**this page is intentionally left blank**

# Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' on the left, 'Workbench' in the center, and a help icon ('?') and settings icon ('⚙️') on the right. Below these is the title 'Exercise 51822 —'. Underneath the title are two tabs: 'WORK AREA' (highlighted in blue) and 'SOLUTIONS'. A green button labeled 'Content Support' is located below the tabs. The main content area contains the following text:

The variable `most_recent_novel` is associated with a dictionary that maps the names of novelists to their most recently published novels. Write a statement that replaces the value "Harry Potter and the Half-Blood Prince" with the value "Harry Potter and the Deathly Hallows" for the key "J.K. Rawling".

Note: we realize that we have unfortunately misspelled the author's name. We will correct this error at a future date.

# **Programming Exercise**

## **Worksheet**

**this page is intentionally left blank**

# Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' on the left, 'Workbench' in the center, and a question mark icon and a gear icon on the right. Below these is the title 'Exercise 51823 —'. Underneath the title are two tabs: 'WORK AREA' and 'SOLUTIONS', with 'WORK AREA' being the active tab. To the right of the tabs is a green button labeled 'Content Support'. The main area contains the following text:

Given a variable, `us_cabinet`, that is associated with a dictionary that maps department names to department heads, replace the value "Gonzalez" with "Mukasey" for the key "Justice Department".

# **Programming Exercise**

## **Worksheet**

**this page is intentionally left blank**

# Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' on the left, 'Workbench' in the center, and a question mark icon and a gear icon on the right. Below these is the title 'Exercise 51824 —'. Underneath the title are two tabs: 'WORK AREA' and 'SOLUTIONS'. A green button labeled 'Content Support' is located below the tabs. The main content area contains the following text:

The variable `planet_distances` is associated with a dictionary that maps planet names to planetary distances from the sun. Write a statement that deletes the entry for the planet name "Pluto".

# **Programming Exercise**

## **Worksheet**

**this page is intentionally left blank**

# Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' on the left, 'Workbench' in the center, and a help icon ('?') and settings icon ('gear') on the right. Below these are two tabs: 'WORK AREA' and 'SOLUTIONS'. A green button labeled 'Content Support' is located above the main content area. The main content area contains the following text:

Given a variable,  
unproved\_conjectures, that is  
associated with a dictionary that  
maps the common names of  
mathematical conjectures to the  
years when the conjectures were  
made, write a statement that  
deletes the entry for "Fermat's Last  
Theorem".

# **Programming Exercise**

## **Worksheet**

**this page is intentionally left blank**

# Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' on the left, 'Workbench' in the center, and a help icon with a gear symbol on the right. Below these are two tabs: 'WORK AREA' (highlighted in blue) and 'SOLUTIONS'. A 'Content Support' button is located above a large text area. The main text area contains the following challenge:

You are given a variable, `wwII_battle_winners`, that is associated with a dictionary that maps the names of World War II battles to the victors of WWII battles. Write some code that associates a sorted list of the battle names that appear as keys in this dictionary with the variable `battles`.

# **Programming Exercise**

## **Worksheet**

**this page is intentionally left blank**

# Programming Exercise

The screenshot shows a programming exercise interface. At the top, there are navigation buttons: 'PREV' and 'NEXT' in green rounded rectangles, followed by a 'Workbench' button in blue and orange, and a help/gear icon. Below these are two tabs: 'WORK AREA' and 'SOLUTIONS'. A 'Content Support' button is located above a scrollable text area. The text area contains the following instruction:

Assume there is a variable,  
mp\_affiliation, that is associated  
with a dictionary that maps the  
names of parliament members to  
party affiliations, associate with the  
variable party\_size a dictionary that  
maps party names to the number  
of members they have.

# **Programming Exercise**

## **Worksheet**

**this page is intentionally left blank**

# Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' on the left, 'Workbench' in the center, and a help icon ('?') and settings icon ('⚙️') on the right. Below these is the title 'Exercise 51828 —'. Underneath the title are two tabs: 'WORK AREA' and 'SOLUTIONS', with 'WORK AREA' being the active tab. A green button labeled 'Content Support' is located above the main content area. The main content area contains the following text:

Given a variable, `province_premier`, that is associated with a dictionary that maps the province names to the names of province premiers, associate with `premier_province` a dictionary that is the inverse of `province_premier`, i.e. one that maps names of premiers to province names.

# **Programming Exercise**

## **Worksheet**

**this page is intentionally left blank**

# Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' in green rounded rectangles, a 'Workbench' button in blue, and a help/gear icon. Below these is a title 'Exercise 51829 —'. Underneath the title are two tabs: 'WORK AREA' (highlighted in blue) and 'SOLUTIONS'. A 'Content Support' button is located above a large text area. The main text area contains the following instruction:

Given a variable, `state_capitals`, that refers to a dictionary that maps U.S. states to their capitals, and another dictionary, `provincial_capitals`, that maps Canadian provinces to their capitals, associate a dictionary that maps states or provinces to their respective capitals with a variable, `regional_capitals`.

# **Programming Exercise**

## **Worksheet**

**this page is intentionally left blank**

# Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there is a toolbar with buttons for 'PREV' and 'NEXT', a 'Workbench' button, and a help icon. Below the toolbar, the title 'Exercise 51830 —' is displayed. Underneath the title are two tabs: 'WORK AREA' and 'SOLUTIONS'. A 'Content Support' button is located above the main content area. The main content area contains the following text:

Given three dictionaries, associated with the variables, `canadian_capitals`, `mexican_capitals`, and `us_capitals`, that map provinces or states to their respective capitals, create a new dictionary that combines these three dictionaries, and associate it with a variable, `nafta_capitals`.

# **Programming Exercise**

## **Worksheet**

**this page is intentionally left blank**

# Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' on the left, 'Workbench' in the center, and a question mark icon and gear icon on the right. Below these is the title 'Exercise 51831 –'. Underneath the title are two tabs: 'WORK AREA' and 'SOLUTIONS', with 'WORK AREA' being the active tab. A green button labeled 'Content Support' is located above the main content area. The main content area contains the following text:

Given a variable, `polygon_sides`,  
that is associated with a dictionary  
that maps names of polygons to  
number of sides, create a new  
dictionary that maps number of  
sides to polygon names, and  
associate it with a variable  
`n_polygons`.

# **Programming Exercise**

## **Worksheet**

**this page is intentionally left blank**

# Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are green navigation buttons for 'PREV' and 'NEXT'. To the right of these are the words 'Workbench' and icons for help (?) and settings (gear). Below this, the title 'Exercise 51832 –' is displayed. Underneath the title are two tabs: 'WORK AREA' and 'SOLUTIONS'. A green button labeled 'Content Support' is located below the tabs. The main content area contains the following text:

Given a variable, `election_results`, that is associated with a dictionary that maps candidate names to votes received, associate the name of the candidate with the most votes with the variable `winner`.

# **Programming Exercise**

## **Worksheet**

**this page is intentionally left blank**