

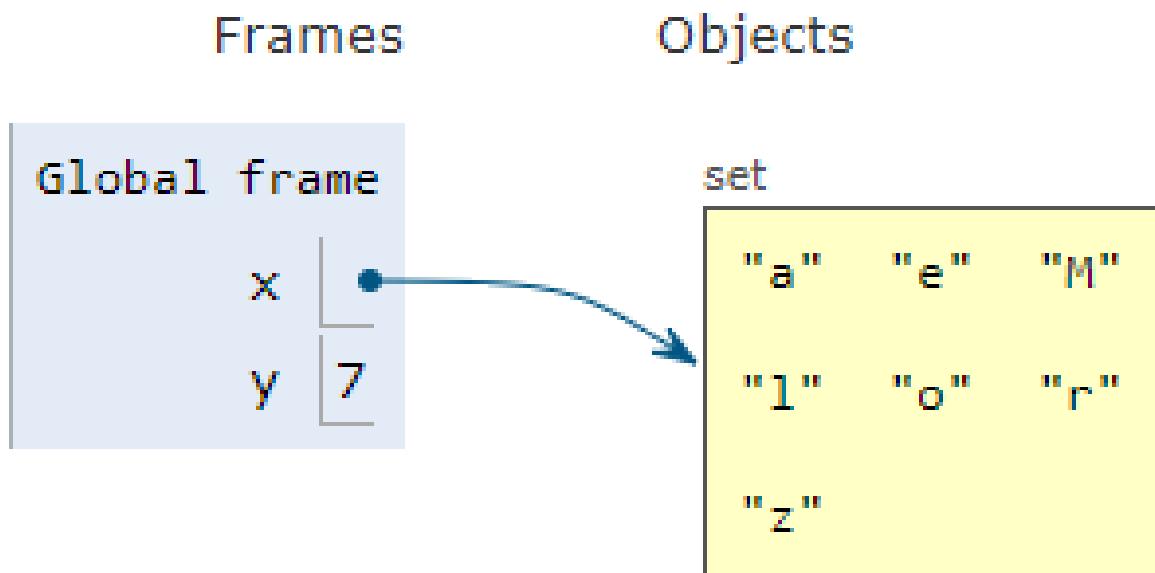
# Python

# Sets

# Python Sets

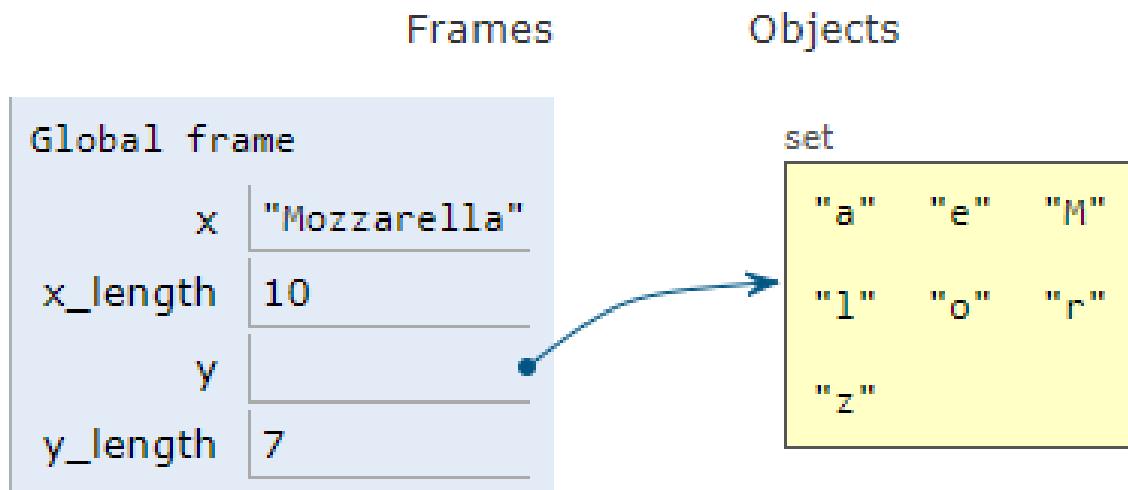
- unordered collection of unique elements
- every element is immutable
- set is mutable (can add/delete)

```
>>> x = {'M','o','z','z','a','r','e','l','l','a'}  
>>> #   x = set('Mozarella')  
>>> y = len(x)
```



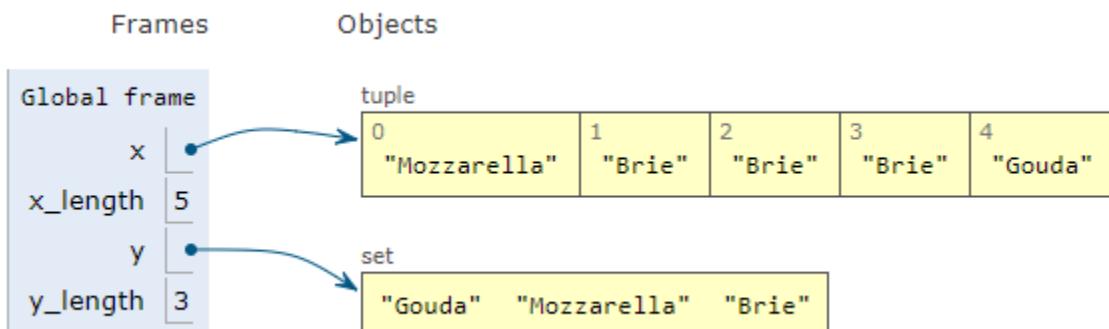
# Creating a Set from a String

```
>>> x = 'Mozzarella'  
>>> x_length = len(x)  
>>> y = set(x)  
>>> y_length = len(y)  
>>> y  
set(['a', 'e', 'M', 'l', 'o', 'r', 'z'])  
>>> type(y)  
<type 'set'>
```



# Creating a Set from a Tuple

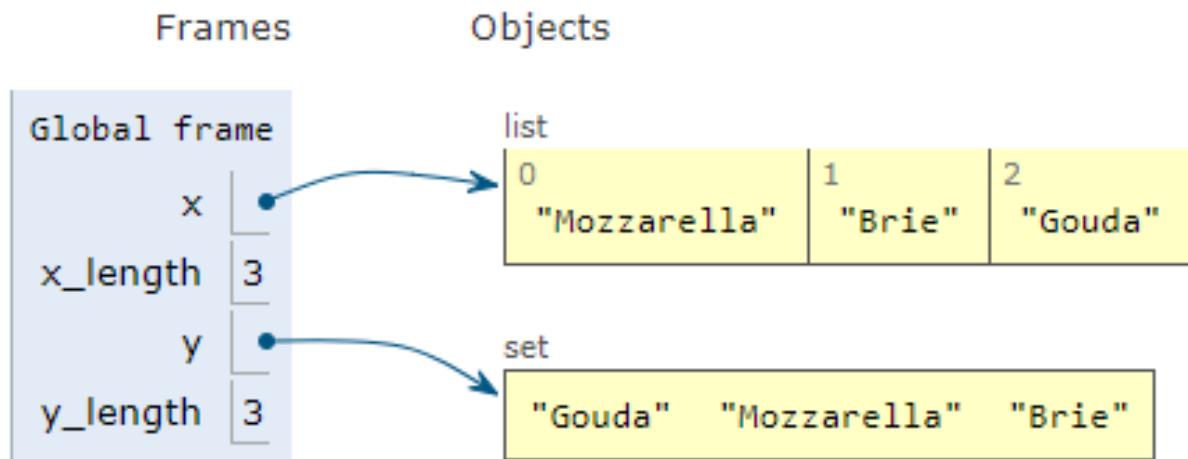
```
>>> x = ('Mozzarella', 'Brie', 'Brie', 'Brie',
        Gouda')
>>> x_length = len(x)
>>> y = set(x)
>>> y_length = len(y)
>>> y
set(['Gouda', 'Mozzarella', 'Brie'])
>>> type(y)
<type 'set'>
```



- no duplicates, no ordering

# Creating a Set from a List

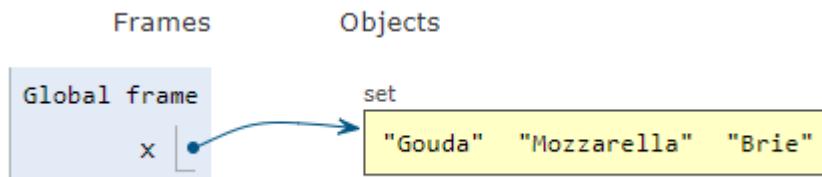
```
>>> x = [ 'Mozzarella', 'Brie', 'Gouda']
>>> x_length = len(x)
>>> y = set(x)
>>> y_length = len(y)
>>> y
set(['Gouda', 'Mozzarella', 'Brie'])
>>> type(y)
<type 'set'>
```



# Sets and Objects Mutability

- can contain immutable objects

```
>>> x = {'Mozzarella', 'Brie', 'Gouda'}
```



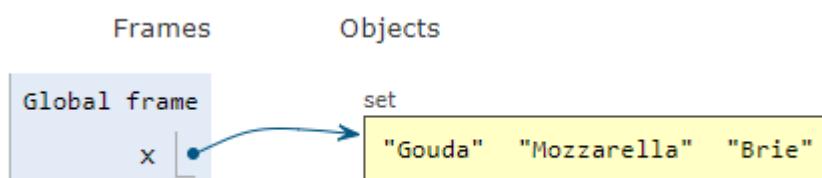
- cannot contain mutable objects

```
>>> x = {[ 'Mozzarella' ], [ 'Brie', 'Gouda' ]}
```

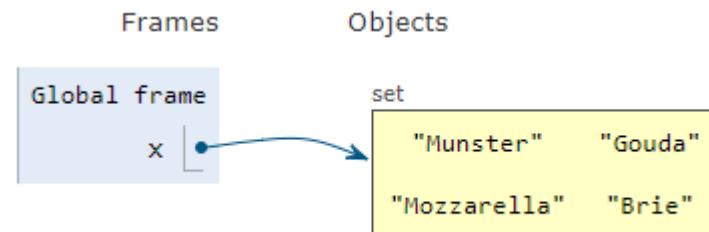
# TypeError: unhashable type: 'list'

- sets are mutable

```
>>> x = {'Mozzarella', 'Brie', 'Gouda'}
```

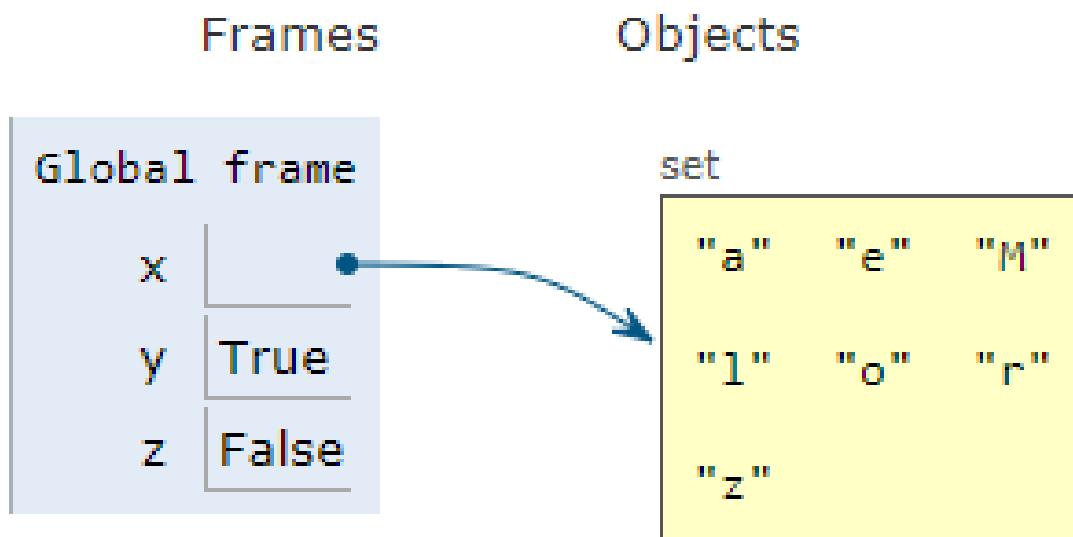


```
>>> x.add('Munster')
```



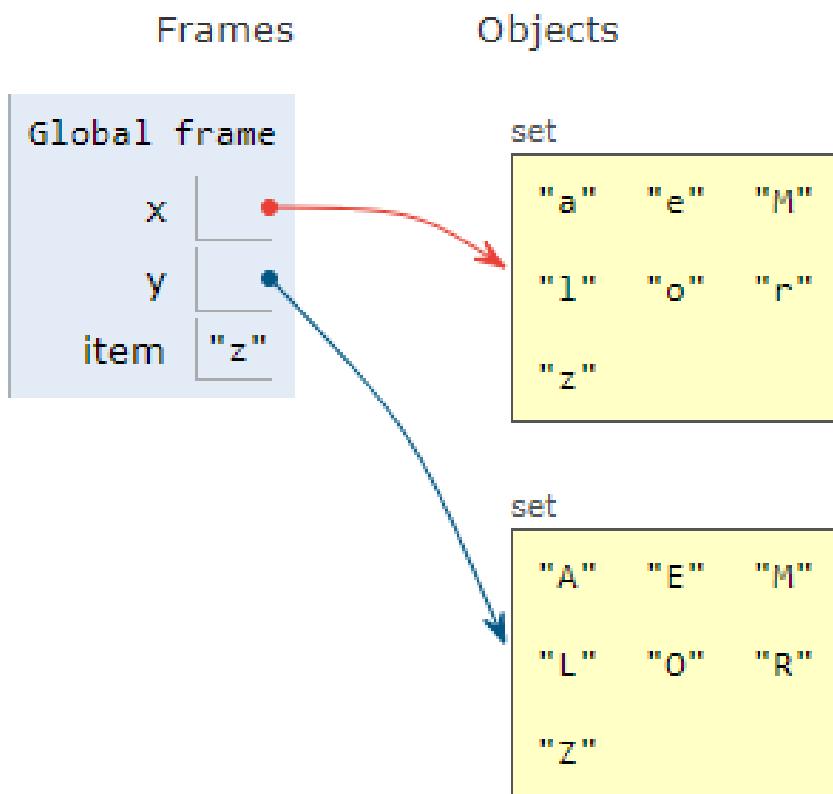
# Set Membership

```
>>> x = {'M','o','z','z','a','r','e','l','l','a'}  
>>> y = 'r' in x  
>>> z = 1 in x
```



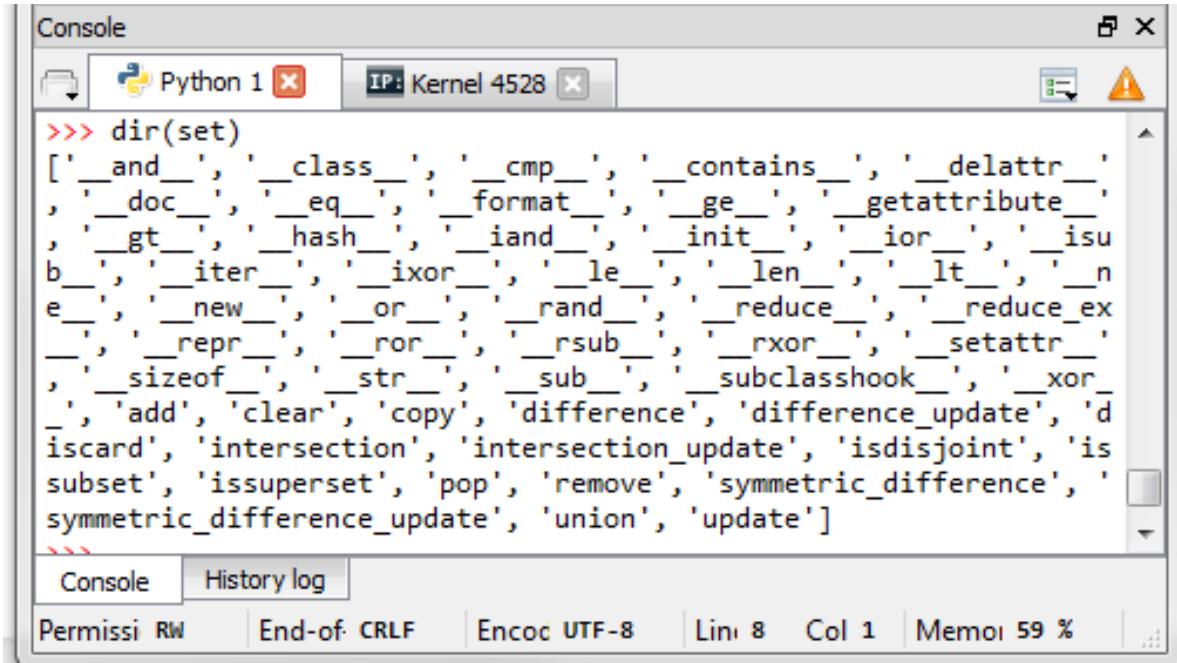
# Set Iteration

```
>>> # capitalize all letters in set x  
>>> x = {'M','o','z','z','a','r','e','l','l','a'}  
>>> y = set() # y={} is a dictionary!  
>>> for item in x:  
>>>     y.add(item.upper())
```



# *set* Type Methods

>>> *dir(set)*



>>> {'B', 'r', 'i', 'e'}.*\_\_contains\_\_*('r')

True

>>> {'B', 'r', 'i', 'e'}.*\_\_len\_\_*()

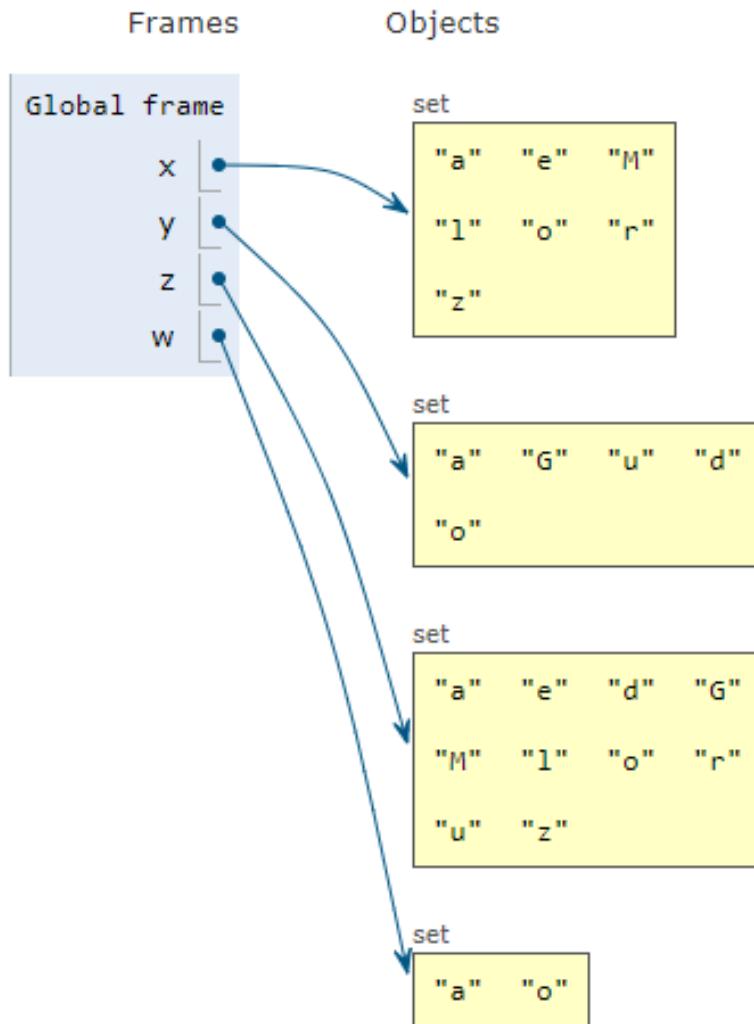
4

>>> {'B', 'r', 'r', 'r', 'r', 'r', 'i', 'e'}.*\_\_len\_\_*()

4

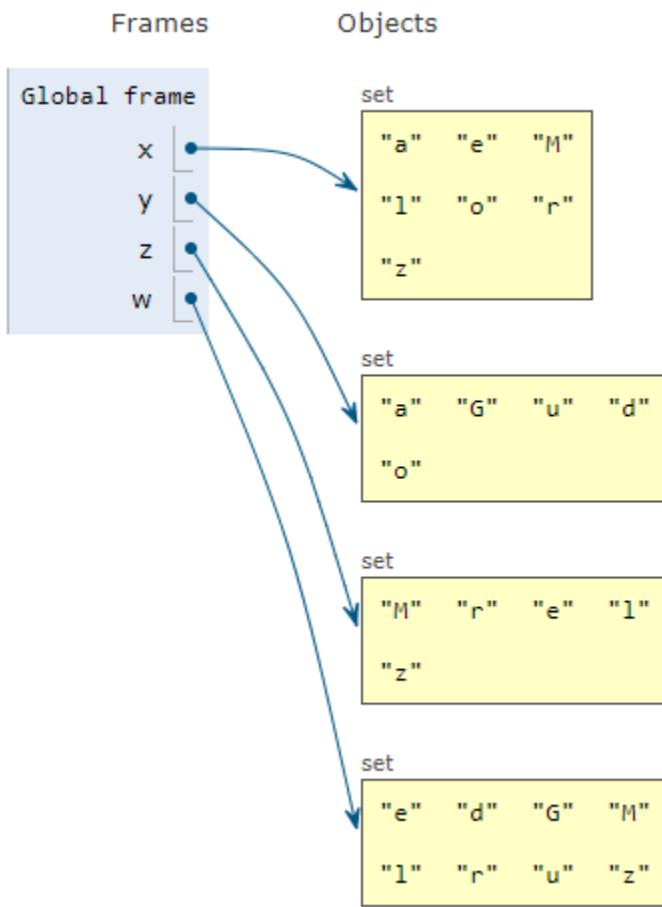
**Sets:** *union()*, *intersection()*

```
>>> x = {'M','o','z','z','a','r','e','l','l','a'}
>>> y = {'G', 'o', 'u', 'd', 'a'}
>>> z = x.union(y)          # z = x | y
>>> w = x.intersection(y)  # z = x & y
```



# Sets: *difference()*, *symmetric\_difference()*

```
>>> x = {'M', 'o', 'z', 'z', 'a', 'r', 'e', 'l', 'l', 'a'}
>>> y = {'G', 'o', 'u', 'd', 'a'}
>>> z = x.difference(y)      # z = x - y
>>> w = x.symmetric_difference(y)
>>> # w = x ^ y
```



# Jacard's Similarity Measure

- defined as  $|A \cap B| / |A \cup B|$

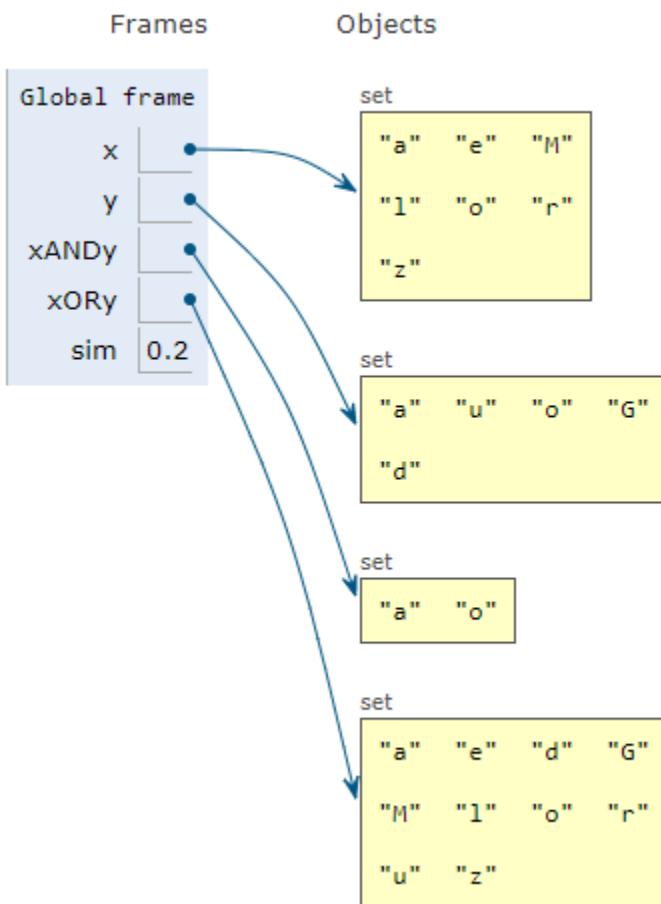
```
>>> x = set("Mozzarella")
```

```
>>> y = set("Gouda")
```

```
>>> xANDy = x.intersection(y)
```

```
>>> xORy = x.union(y)
```

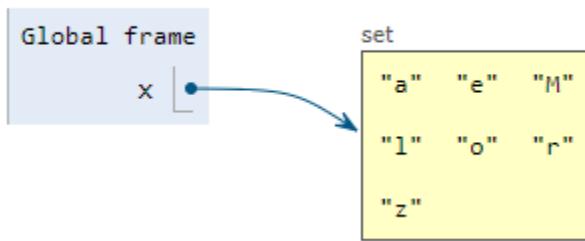
```
>>> jacard = float(len(xANDy))/len(xORy)
```



# Sets: *add()*, *remove()*

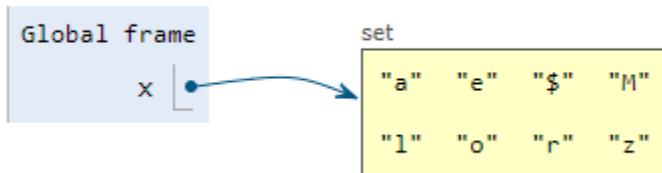
```
>>> x = {'M', 'o', 'z', 'z', 'a', 'r', 'e', 'l', 'l', 'a'}
```

Frames                    Objects



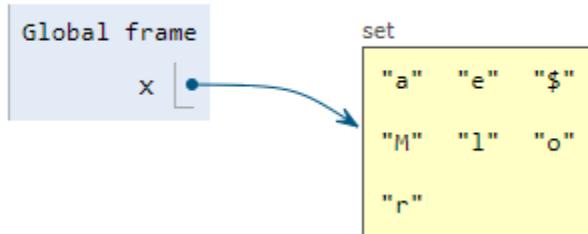
```
>>> x.add('$')
```

Frames                    Objects



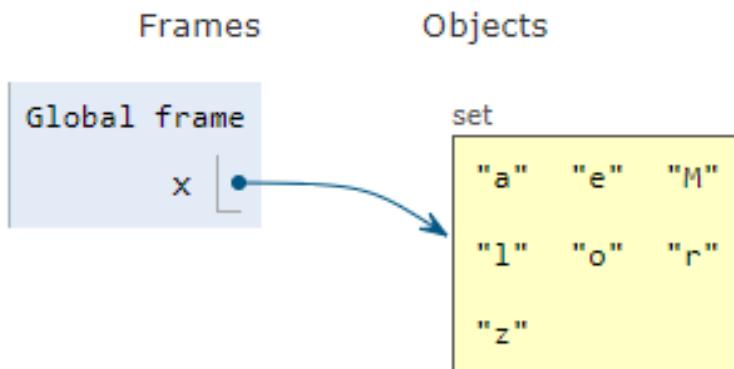
```
>>> x.remove('z') # or x.discard('z')
```

Frames                    Objects

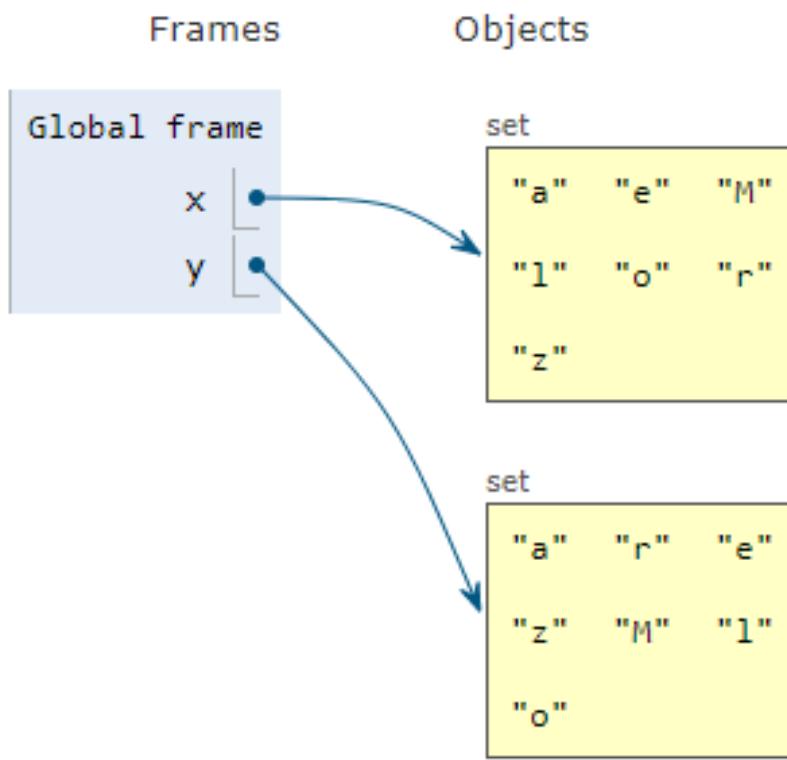


# Sets: *copy()*

```
>>> x = {'M','o','z','z','a','r','e','T','T','a'}
```

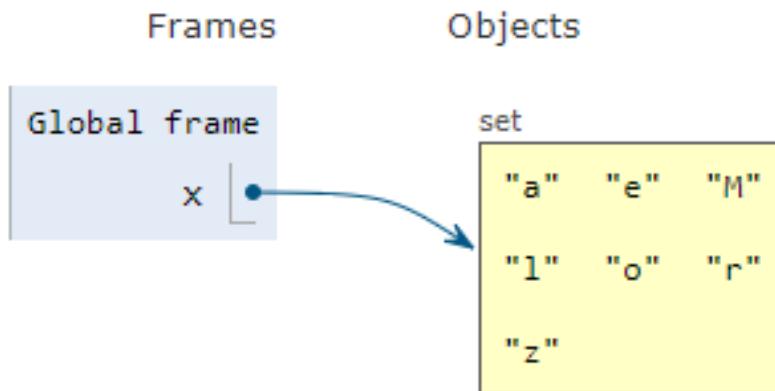


```
>>> y = x.copy()
```

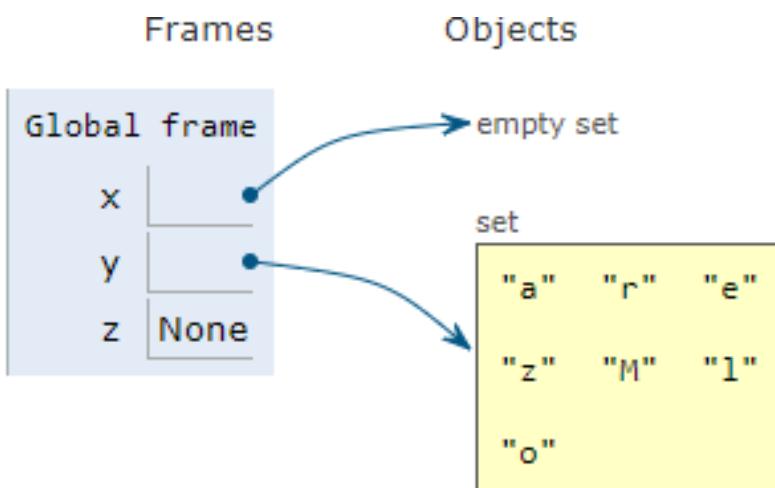


# Sets: *clear()*

```
>>> x = {'M','o','z','z','a','r','e','l','l','a'}
```

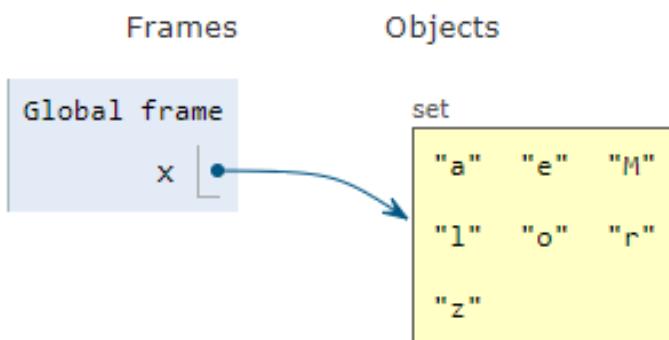


```
>>> z = x.clear()
```

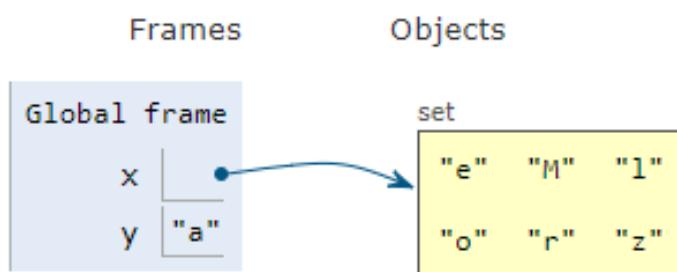


**Sets:** *pop()*, *discard()*

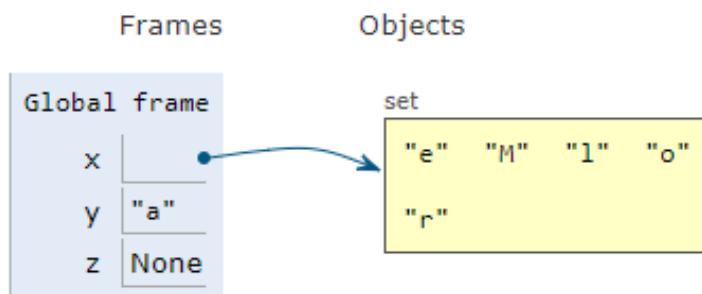
```
>>> x = {'M','o','z','z','a','r','e','T','T','a'}
```



```
>>> y = x.pop() # remove random item
```



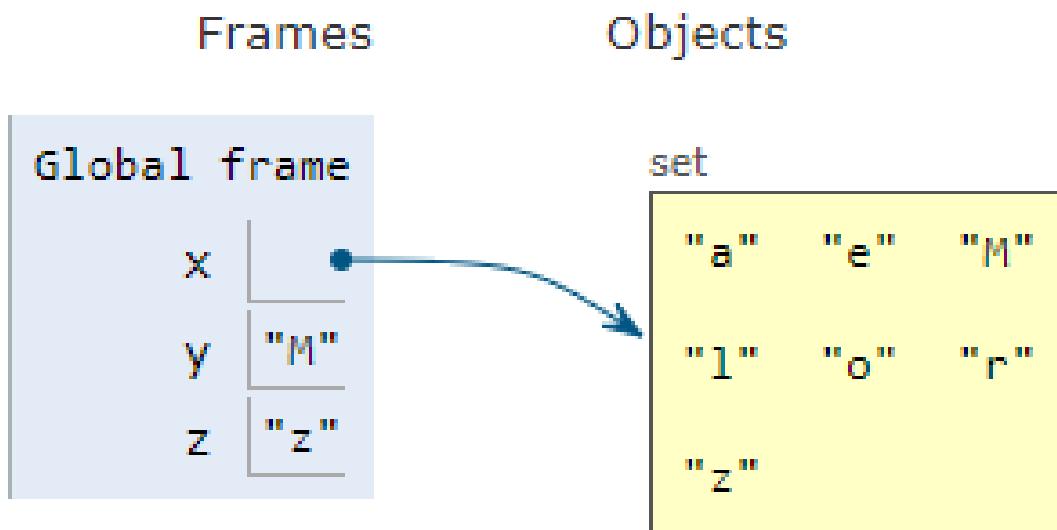
```
>>> z = x.discard('z')
```



# Sets: *min()*, *max()*, *sum()*

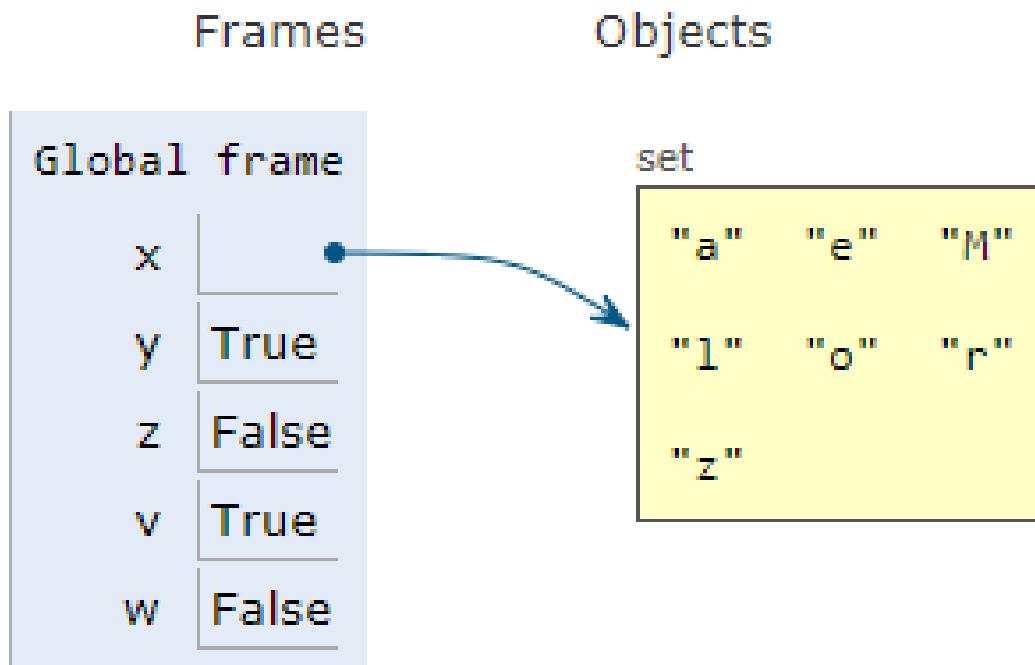
- *sum()* is defined for numeric elements

```
>>> x = {'M','o','z','z','a','r','e','l','l','a'}  
>>> y = min(x)  
>>> z = max(x)
```



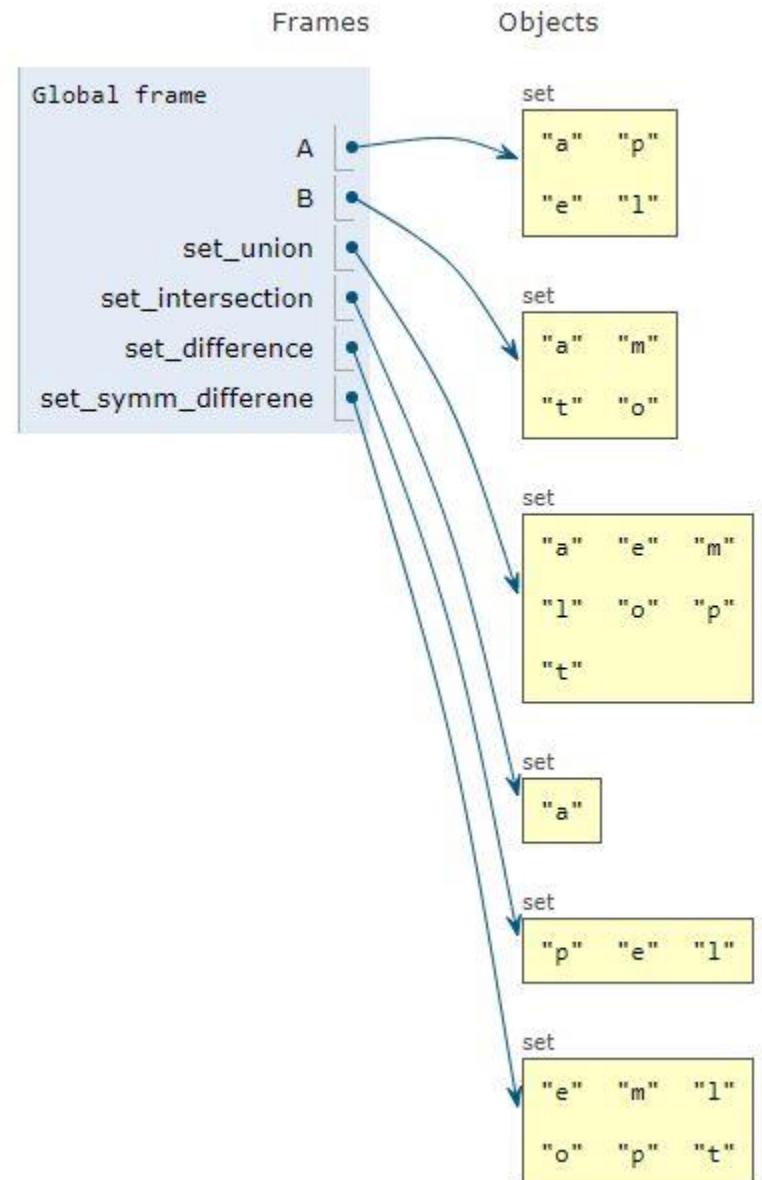
**Sets:** *issubset()*, *issuperset()*

```
>>> x = {'M','o','z','z','a','r','e','T','T','a'}
>>> y = x.issuperset({'o', 'z'})
>>> z = x.issuperset({1, 2, 3})
>>> v = x.issubset({'a', 'e', 'M', 'T', 'o', 'r',
'z', 1, 2, 3})
>>> w = x.issubset({100, 200})
```



# Common Set Operations

```
Python 2.7
1 A = set('apple')
2 B = set('tomato')
3 set_union = A | B
4 set_intersection = A & B
5 set_difference = A - B
6 set_symm_difference = A ^ B
```



# Review Problems

# 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, and 'Workbench' in the center. To the right of 'Workbench' are two green icons: a question mark and a gear. Below these buttons, the title 'Exercise 51298 —' is displayed. Underneath the title are two tabs: 'WORK AREA' and 'SOLUTIONS'. The 'WORK AREA' tab is highlighted with a blue border. In the main content area, there is a green button labeled 'Content Support'. Below it, a text box contains the following instruction: 'Remove the smallest element from the set, s. Assume the set 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 is a toolbar with four buttons: 'PREV' (left arrow), 'NEXT' (right arrow), 'Workbench' (text in blue and red), a question mark icon, and a gear icon. Below the toolbar, the text 'Exercise 51301 —' is displayed. Underneath this, there 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: 'Remove the smallest element from the set, s. If the set is empty, it remains empty.'

PREV NEXT Workbench ?

Exercise 51301 —

WORK AREA SOLUTIONS

Content Support

Remove the smallest element from the set, s. If the set is empty, it remains 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 51305 —'. 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 set, **weights**, and an integer **desired\_weight**, remove the element of the set that is closest to **desired\_weight** (the closest element can be less than, equal to OR GREATER THAN **desired\_weight**), and associate it with the variable **actual\_weight**. For example, if weights is (12, 19, 6, 14, 22, 7) and **desired\_weight** is 18, then the resulting set would be (12, 6, 14, 22, 7) and **actual\_weight** would be 19. If there is a tie, the element LESS THAN **desired\_weight** is to be chosen. Thus if the set is (2, 4, 6, 8, 10) and **desired\_weight** is 7, the value chosen would be 6, not 8. Assume there is at least one value in the set.

# **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/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 question:

Exercise 51800 —

Write a **statement** that associates **s** with the empty set.

# **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 are two tabs: 'WORK AREA' and 'SOLUTIONS'. A green button labeled 'Content Support' is located near the bottom of the interface. The main area contains the following text:

Write a **statement** that associates **s** with a set that contains the following elements: **23, 42, -11, 89.**

# **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 top right of the main area. The main content area contains the following text:

Write an **expression** whose value  
is a set that contains the following  
elements: -2, 1, 10001, 89, 63.

# **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 green button labeled 'Content Support' is positioned above the main content area. The main content area contains the following text:

Given that L has been defined to refer to a list, write an expression whose value is a set containing all the elements of L.

# **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, and 'Workbench' in the center. To the right of 'Workbench' are two icons: a question mark and a gear. Below these are two tabs: 'WORK AREA' and 'SOLUTIONS'. The 'WORK AREA' tab is currently selected, indicated by a grey background. In the main content area, there is a green button labeled 'Content Support'. The main text area contains the following instruction:

Given that `s` refers to a set, write a statement that adds the int value **42** to the set.

# **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. Below these are two tabs: 'WORK AREA' and 'SOLUTIONS'. A 'Content Support' button is located at the bottom right of the main area. The main content area contains the following text:

Given that `s` refers to a set, write a statement that removes the int value 5 from the set.

# **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 'Content Support' button is located above a large text area. The main text area contains the following question:

Given that `s` refers to a set, write a statement that attempts to remove the int value `11` from the set, but will do nothing if `11` is not in the set.

# **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 labeled "PREV" and "NEXT" with circular arrows, a title "Workbench", and a help icon. Below the title, the text "Exercise 51808" is displayed. A tab bar at the bottom has two tabs: "WORK AREA" (selected) and "SOLUTIONS". A "Content Support" button is located in the middle right of the main area. The main content area contains the following text:

Given that `s` has been defined, and that `the_set` that refers to a set, write an **expression** that whose value is True if and only if the value to which `s` refers is in `the_set`.

# **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, a 'Workbench' tab in the center, and a help icon ('?') and settings icon ('⚙️') on the right. Below the navigation is the title 'Exercise 51809 —'. Underneath the title 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 instruction:

Given that v has been defined, and that s that refers to a set, write a statement that adds the value referred to by v to set s.

# **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 right of the main area. The main content area contains the following text:

Given that `v` has been defined, and that `s` refers to a set, write a **statement** that removes the value associated with `v` from the set.

# **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'. To the right of these are links for 'Workbench', '?', and a gear icon. Below the navigation is the title 'Exercise 51811 —'. Underneath the title are two tabs: 'WORK AREA' (which is selected) and 'SOLUTIONS'. A green button labeled 'Content Support' is located below the tabs. The main content area contains the following text:

Given that `v` has been defined, and that `s` that refers to a set, write a **statement** that attempts to remove the value associated with `v` from set `s`, but **will do nothing if it is not in `s`.**

# **Programming Exercise**

## **Worksheet**

**this page is intentionally left blank**