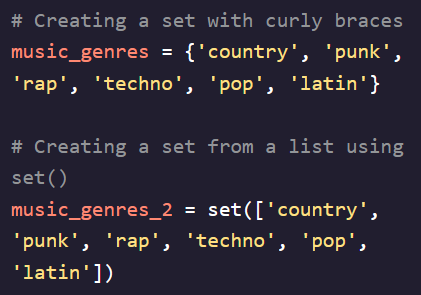
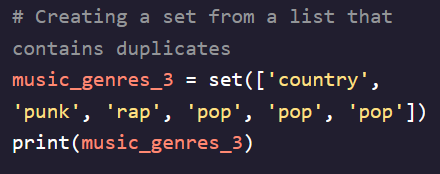
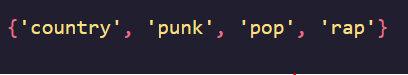
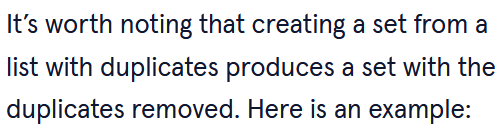
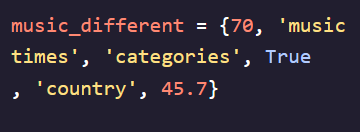
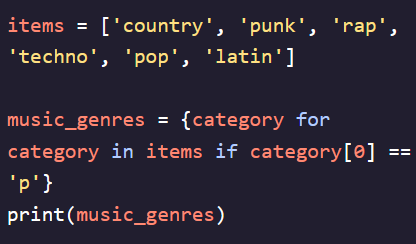
**Sets:**

- A ***set*** is a group of elements that are unordered and do not contain duplicates, useful for organizing items and performing set mathematics  
- Using set mathematics we can find matching items, differences combine sets based on different parameters and more – very useful when combing through very large datasets  
- A ***frozenset*** is an immutable version of a ***set*** that behaves the same way but does not include any methods that can modify it in any way (tuple of sets)

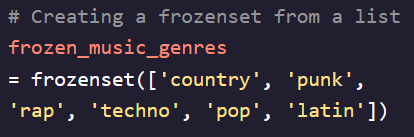
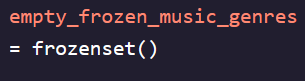
**Creating a Set:**

- A set can be created by passing an iterable object into its constructor, using curly braces, or using a set comprehension  
- They can contain any combination of data types as long as they are unique values  


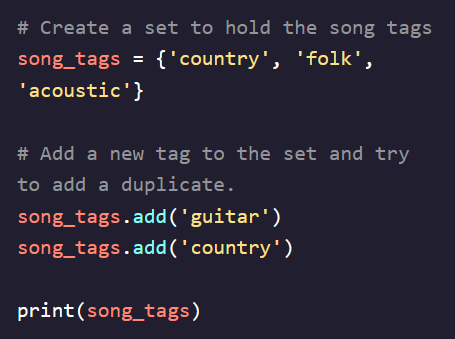
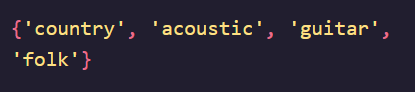
- Can also create sets using a *set comprehension*, similar to list or generator comprehensions  


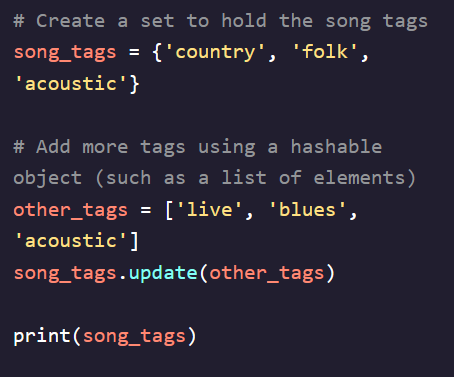
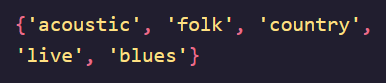
**Creating a frozenset:**

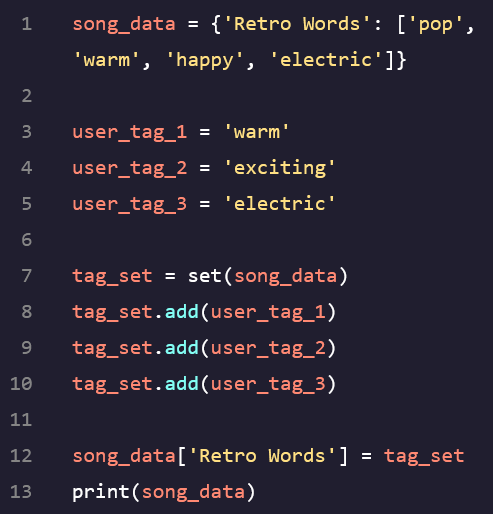
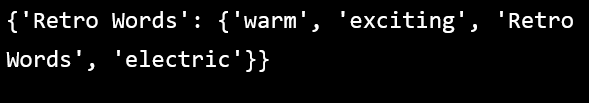
- Can only create a ***frozenset*** using its constructor  
 

**Adding to a Set:**

- There are two different ways to add to a set:

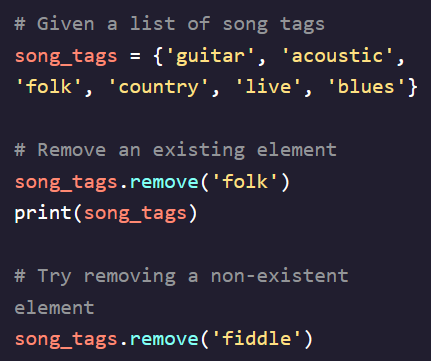
1. Using the ***.add()*** method – Adds individual elements  
 

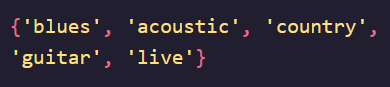
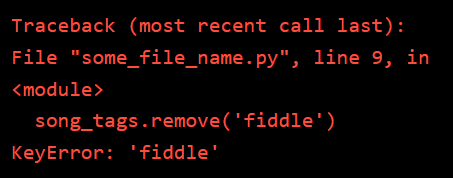
2. Using the ***.update()*** method adds multiple elements  
 

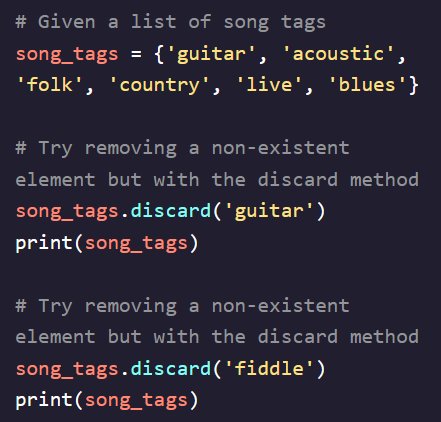
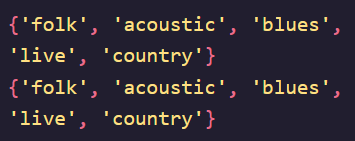
- Neither of these methods will add a duplicate item to a set  
- ***frozensets*** cannot have any items added to them  
- When printed, neither ***set*** nor ***frozenset*** print items in the same order because they are unorded lists  
 

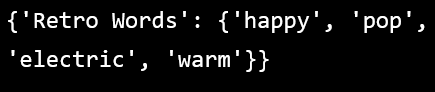
**Removing from a set:**

- There are two methods to remove from a set:

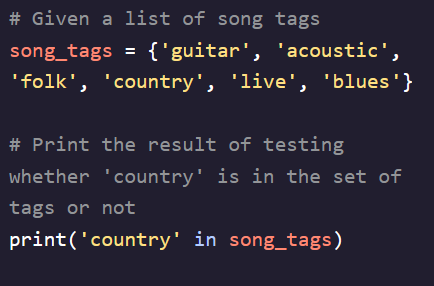
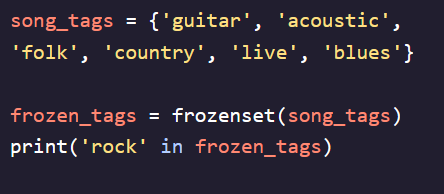
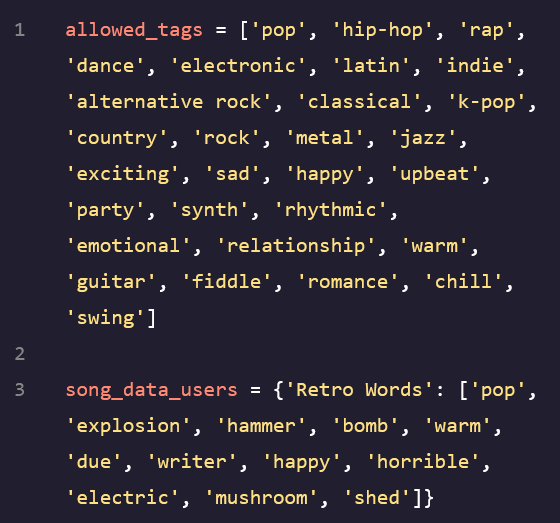
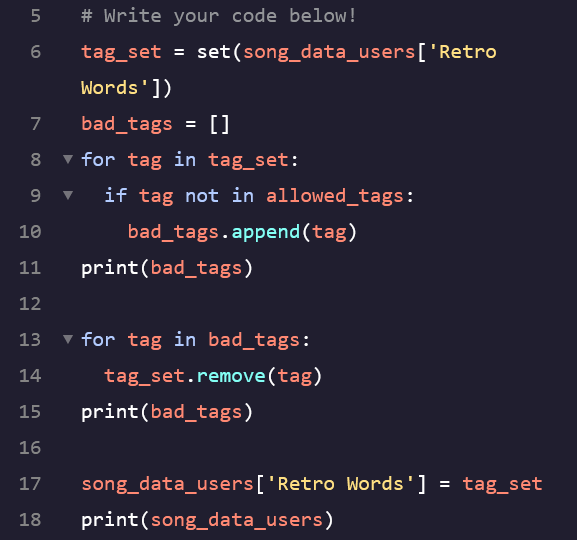
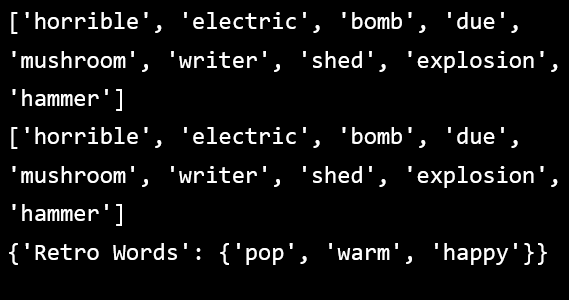
1. Using the ***.remove()*** method – searches for an element and removes it or throws a KeyError if it doesn’t exist  


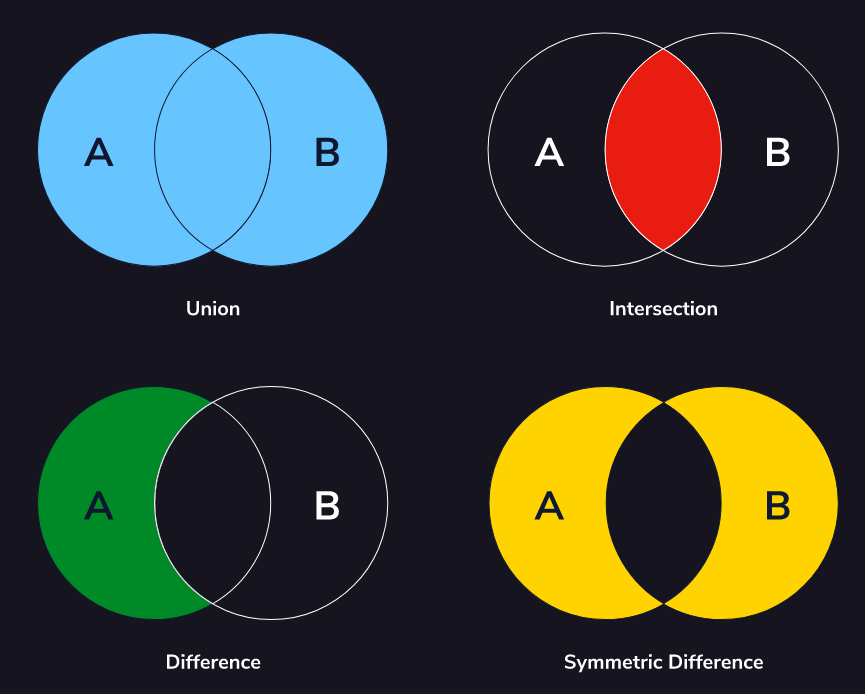
  


2. Using the ***.discard()*** method – Works the same way but does not throw an exception if element doesn’t exist  
 

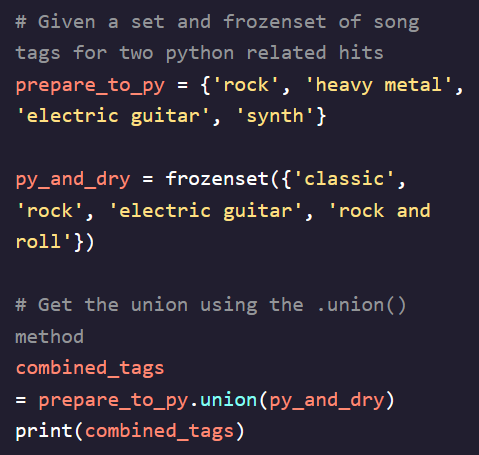
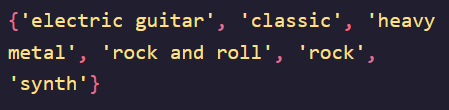
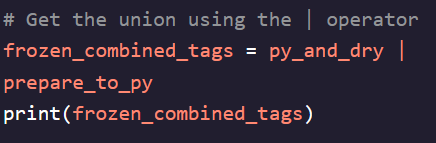
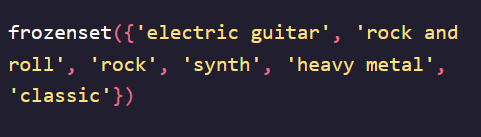
 

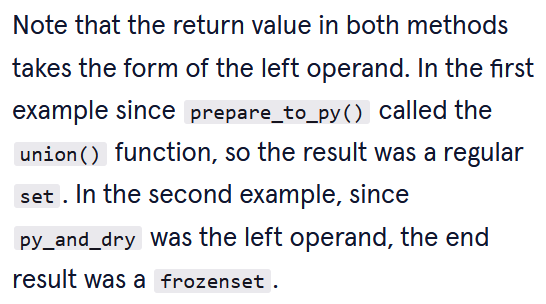
**Finding Elements in a Set:**

- ***set*** and ***frozenset*** cannot be accessed by a specific index because they are unordered and have no indices  
- Can use the ***in*** keyword to determine if an element exists within either one  
   
   


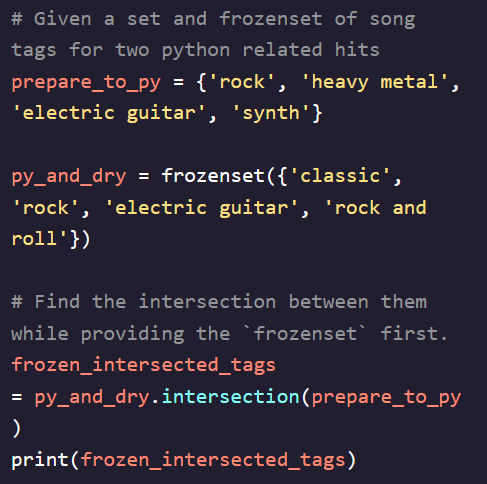
**Set Operations:**

**Set Union:**

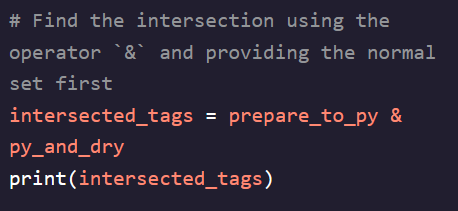
- One of the most common operations we can perform is a merge that will return a new ***set/frozenset*** containing all elements from both sets without duplicates  
- Executed using the ***.union()*** method or with the **|** operator  
   
 



**Set Intersection:**

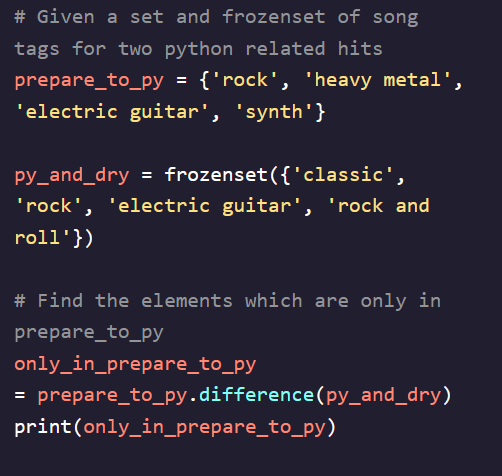
- Used to create a new set with items that two sets have in common  
- Executed using the ***.intersection()*** method or with the **&** operator  




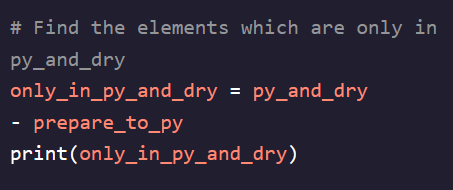
  


- Can also use a method called .***intersection\_update()*** to update the original set instead of creating a new one

**Set Difference:**

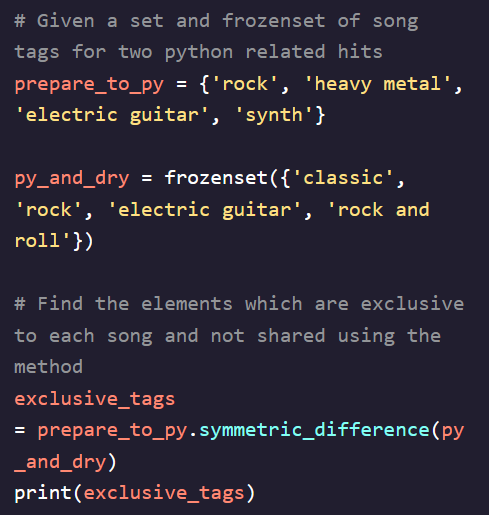
- Used to create a new set with items that are unique elements between two sets (elements they **don’t** have in common) – returns a new set that contains only the elements from the first set that aren’t in the second set  
- Executed using the ***.difference()*** method or with the **–** operator  


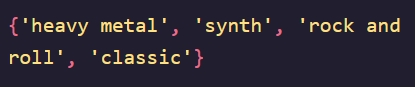


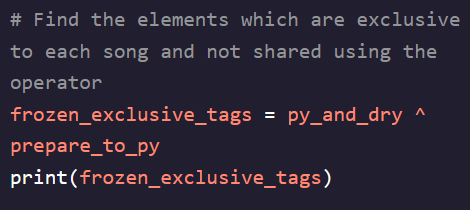
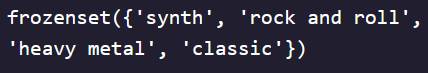
  


- Can also use a method called ***.difference\_update()*** to update the original set instead of creating a new one

**Symmetric Difference:**

- Used to create a new set with items that are in one but not the other or in neither – opposite of the intersection method  
- Executed using the ***.symmetric\_difference()*** method or with the **^** operator  




- Can also use a method called ***.symmetric\_difference\_update()*** to update the original set instead of creating a new one