```
"""
sets_lab.py
@Author: ITP 150 Student
Date Created: Sept. 30, 2024
This script shows how to create sets, and uses the intersection &, union |
difference -, operators on sets. We see how we can use a set to remove
duplicate items from a list by making it into a set and then saving it back
to a list.
A set is a collection which is unordered, unchangeable, and unindexed. Set
items are immutable, but you can remove items and add new items.
"""


print('Task 1. Intersection & gives items common to both sets.')
setx = set(['green', 'blue'])
print(setx)
sety = set(['blue', 'yellow'])
print(sety)
setz = setx & sety  # Note the & intersection operator
print(setz)

print('Task 2. Union | which gives items from both sets but does not allow \
a duplicate')
seta = setx | sety  # note the union | operator
print(seta)
print(id(seta))

print('Task 3. Add an element to the set.')
seta.add('red')
print(seta)
print(id(seta))

print('Task 4. Difference - which gives values not common in both sets.')
setb = seta - sety  # note the - difference operator
print(setb)

print('Task 5. Take a list with duplicate items and make a set from it in \
order to remove the duplicate items and then assign again as a list with the \
same name as the original list. But because we used the list() function on \
set1 to "recreate" list1, we have a different object ID')
list1 = [1, 2, 2, 3, 4, 4, 5, 6, 6]
print('list1 contents', list1)
print('list1 ID', id(list1))
set1 = set(list1)
print('set1 contents', set1)
print('set1 ID', id(set1))
list1 = list(set1)
print('list1 contents', list1)
print('list1 ID', id(list1))

```