Universidad Peruana Cayetano Heredia
Facultad de Ciencias e Ingeniería
Departamento de Ciencias Exactas

Programación Avanzada 2025

Lab. 3.1. Conjuntos en Python

Septiembre 03, 2025

Cree una carpeta (folder) en el disco D, nómbrela con su apellido paterno seguido de su código. Ejemplo: LOPEZ12345

Un **conjunto** es una colección desordenada de ítems (elementos). Cada elemento es único (no duplicados) y debe de ser inmutable (que no puede cambiar).

Sin embargo, el mismo conjunto es mutable. Podemos adicionar o remover un ítem de él.

Si vuestra aplicación no necesita que los elementos estén colocados en algún orden particular, usar un conjunto para almacenar elementos es más eficiente que usar una lista.

Un conjunto puede tener cualquier cantidad de ítems y pueden ser de diferentes tipos (integer, float, tuple, string, etc.). Sin embargo, no puede tener elementos mutables, como listas, conjuntos, diccionarios.

Creando conjuntos

Los conjuntos se crean colocando sus elementos entre llaves (**{}**). Los elementos están separados por comas.

Ejemplos de creación de conjuntos:

s1 = set() # Crea un conjunto vacío # s1 = set(<iter>)

s2 = {**1**, **3**, **5**} # Crea un conjunto con tres elementos

s3 = {1.0, "Hello", (1, 2, 3)} # Crea un conjunto con elementos de distintos tipos

s4 = set([**1**, **3**, **5**]) # Crea un conjunto a partir de una lista

s5= set(("hello", "world", "of", "words", "of", "world")) # Crea un conjunto a partir de una tupla. Los elementos duplicados se eliminan

De la misma manera se puede crear una lista o una tupla a partir de un conjunto usando la sintaxis **list(set)** o **tuple(set)**.

También se puede crear un conjunto a partir de un string. Cada carácter del string se convierte en un elemento del conjunto. Por ejemplo:
Crea un conjunto a partir de un string

s6 = set(**"abac"**) # s6 is {'a', 'b', 'c'}

Recorrido de un conjunto

Se puede recorrer por cada uno de los elementos de un conjunto con la ayuda del bucle `for`, sin embargo, no podemos usar índices, ya que los elementos no siguen un orden específico en el conjunto (en la memoria).

```
for letter in set("Hola"):
      print(letter)
```

Para ejercicios del 1 al 11, verificar la salida con la ayuda del interpretador de python

1. ¿Cómo crear un conjunto vacío?
2. ¿Puede una lista, conjunto o tupla tener elementos de diferentes tipos?
3. ¿Cuáles de los siguientes conjuntos están creados correctamente?
```
s = {1, 3, 4}
s = {{1, 2}, {4, 5}}
s = {[1, 2], [4, 5]}
s = {(1, 2), (4, 5)}
```

4. ¿Cuál es la diferencia entre una lista y un conjunto?, ¿cómo se crea un conjunto desde una lista?, ¿cómo se crea una tupla desde un conjunto?

5. Muestre lo que imprime el siguiente código:
```
students = {"peter", "john"}
print(students)
students.add("john")
print(students)
students.add("peterson")
print(students)
students.remove("peter")
print(students)
```

6. ¿Tendrá el siguiente código un error de tiempo de ejecución?
```
students = {"peter", "john"}
students.remove("johnson")
print(students)
```

7. Muestre lo que imprime el siguiente código:
```
student1 = {"peter", "john", "tim"}
student2 = {"peter", "johnson", "tim"}
print(student1.issuperset({"john"}))
print(student1.issubset(student2))
print({1, 2, 3} > {1, 2, 4})
print({1, 2, 3} < {1, 2, 4})
print({1, 2} < {1, 2, 4})
print({1, 2} <= {1, 2, 4})
```

8. Muestre lo que imprime el siguiente código:
```
numbers = {1, 4, 5, 6}
print(len(numbers))
print(max(numbers))
print(min(numbers))
print(sum(numbers))
```

9. Muestre lo que imprime el siguiente código:
```
s1 = {1, 4, 5, 6}
s2 = {1, 3, 6, 7}
print(s1.union(s2))
print(s1 | s2)
print(s1.intersection(s2))
print(s1 & s2)
print(s1.difference(s2))
print(s1 - s2)
print(s1.symmetric_difference(s2))
print(s1 ^ s2)
```

10. Muestre lo que imprime el siguiente código:
```
set1 = {2, 3, 7, 11}
print(4 in set1)
print(3 in set1)
print(len(set1))
print(max(set1))
```

```
print(min(set1))
print(sum(set1))
print(set1.issubset({2, 3, 6, 7, 11}))
print(set1.issuperset({2,3, 7, 11}))
```

11. Muestre lo que imprime el siguiente código:

```
set1 = {1, 2, 3}
set2 = {3, 4, 5}
set3 = set1 | set2
print(set1, set2, set3)
set3 = set1 - set2
print(set1, set2, set3)
set3 = set1 & set2
print(set1, set2, set3)
set3 = set1 ^ set2
print(set1, set2, set3)
```

Para los siguientes ejercicios, elabore un programa en Python. Verifique los programas ejecutando y probando con distintas entradas.

1. Escriba una función que reciba una cantidad variable de strings como argumento y que imprima los caracteres únicos de cada string en forma de lista.

2. Escriba una función que reciba una lista de números enteros y devuelva un conjunto con los números que se repiten.

   En el programa principal generar una lista de números enteros de longitud N, los valores de la lista pueden ser ingresados del teclado o generados con `random`.

   ```
   Ingrese el valor de N: 10
   Lista de números: [6, 1, 8, 1, 6, 6, 6, 1, 9, 9]
   Valores duplicados: {1, 6, 9}
   ```

3. El formato PDB es usado con frecuencia para almacenar información sobre moléculas. Un archivo PDB puede contener 0 o más líneas que inician con la palabra AUTHOR (el cual puede estar en mayúsculas, minúsculas, o mezclado), seguido por espacios o tabulaciones, seguido por el nombre de las personas que crearon el archivo. Escriba una función que reciba como argumento una lista de nombres de archivos y devuelva un conjunto de nombres de autores que aparecen en los archivos.

4. Los objetos conjuntos de Python tienen un método llamado `pop` que remueve y devuelve un elemento arbitrario del conjunto. Si el conjunto `gatos` contiene cinco animales mimosos, por ejemplo, invocando `gatos.pop()` cinco veces, retornará uno a uno los animales, dejando al conjunto vacío al final. Usando ése método, escriba una función llamado `pareja` que reciba dos conjuntos de igual tamaño llamados `machos` y `hembras` y devuelva un conjunto de pares, cada par debe de ser una tupla que contenga un macho y una hembra. (Los elementos de `machos` y `hembras` son nombres de los gatos)
   En el programa principal:
   - Ingrese del teclado el tamaño de las listas de nombres.
   - Genere dos listas de nombres (macho y hembra), las listas deben de ser del mismo tamaño.
   - Convierta las listas a conjuntos (machos y hembras)
   - Asegure que los conjuntos sean del mismo tamaño
   - Invoque a la función e imprima las parejas de nombres

Guarde todos vuestros programas en una carpeta con el nombre su **Apellido** paterno seguido de vuestro **DNI**, luego comprima esta carpeta. Envíe este archivo a:
Katherine Navarro katherine.navarro@upch.pe especificando como asunto **Lab3.1**.