



/TheVampi



LAB 01

Conjuntos en Python

Estructura de
Datos

Martinez Rivera Luis Fernando
Ruiz Olvera Alfonso Emilio





Descripción

Instrucciones:

1) Resuelve en Python la representación de las siguientes operaciones de conjuntos.

Entrada sean los conjuntos

$U = \{1,2,3,4,5,6,7,8,9,10,11,12,13,14\}$

$A = \{2,4,6,8,10,12,14\}$

$B = \{1,2,3,5,8,13\}$

$C = \{1,2,4,6,7,10,11,13\}$

Determinar:

$A \cup B$

$B \cap C$

C'

$B - A$

$C - B'$

Archivo completo disponible en el siguiente repositorio:

<https://github.com/TheVampi/DataStructures-TecNM/blob/main/Python/conjuntosLAB01.py>

Código fuente

```
U = {1,2,3,4,5,6,7,8,9,10,11,12,13,14}
A = {2,4,6,8,10,12,14}
B = {1,2,3,5,8,13}
C = {1,2,4,6,7,10,11,13}
```

```
#USANDO LIBRERIAS Y FUNCIONES
#A union B
print(A | B)
#B intersection C
print (B & C)
#Complement C
print(U-C)
#B union C
print(B | C)
#A intersection C
print(A & C)
#Complement A
print(U - A)
#B - A
print(B - A)
```





```
# C - Complement B
print(C-(U-B))

#USANDO CICLOS
print("Los resultados de abajo son los mismos, pero usando ciclos for")
# A union B
A_union_B = set()
for a in A:
    A_union_B.add(a)
for b in B:
    A_union_B.add(b)
print(A_union_B)

# B intersección C
B_interseccion_C = set()
for b in B:
    if b in C:
        B_interseccion_C.add(b)
print(B_interseccion_C)

# Complemento C
complemento_C = set()
for u in U:
    if u not in C:
        complemento_C.add(u)
print(complemento_C)

# B union C
B_union_C = set()
for b in B:
    B_union_C.add(b)
for c in C:
    B_union_C.add(c)
print(B_union_C)

# A intersección C
A_interseccion_C = set()
for a in A:
    if a in C:
        A_interseccion_C.add(a)
print(A_interseccion_C)

# Complemento A
complemento_A = set()
for u in U:
```





```
    if u not in A:
        complemento_A.add(u)
print(complemento_A)

# B - A
B_menos_A = set()
for b in B:
    if b not in A:
        B_menos_A.add(b)
print(B_menos_A)

# C - Complemento B
complemento_B = set()
for u in U:
    if u not in B:
        complemento_B.add(u)
C_menos_complemento_B = set()
for c in C:
    if c not in complemento_B:
        C_menos_complemento_B.add(c)
print(C_menos_complemento_B)
```

Capturas de ejecución

```
[Running] python -u "c:\Users\luisi\Desktop\DataStructures-TecNM\Python\conjuntosLAB01.py"
{1, 2, 3, 4, 5, 6, 8, 10, 12, 13, 14}
{1, 2, 13}
{3, 5, 8, 9, 12, 14}
{1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 13}
{2, 10, 4, 6}
{1, 3, 5, 7, 9, 11, 13}
{1, 13, 3, 5}
{1, 2, 13}
Los resultados de abajo son los mismos, pero usando ciclos for
{1, 2, 3, 4, 5, 6, 8, 10, 12, 13, 14}
{1, 2, 13}
{3, 5, 8, 9, 12, 14}
{1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 13}
{2, 10, 4, 6}
{1, 3, 5, 7, 9, 11, 13}
{1, 13, 3, 5}
{1, 2, 13}

[Done] exited with code=0 in 0.172 seconds
```

