```
Start coding or generate with AI.
```

1. Create a set with the rst ve prime numbers.

```
A = {2,3,5,7,9,11}
print(A)
```

➔▾ {2, 3, 5, 7, 9, 11}

Double-click (or enter) to edit

2. Add the number 7 to the set.

```
A = {2,3,5,7,11}
A.add(8)
print(A)
```

➔▾ {2, 3, 5, 7, 8, 11}

4. Check if the set is a subset of {2, 3, 5, 7, 11}.

```
A = {2, 3, 5, 7, 11}
B = {2, 3, 5, 7}
print(A.issubset(B))
```

➔▾ False

5. Find the union of the set with {7, 11, 13}.

```
A = {2,3,5,7,11}
B = {7,11,13}
print(A.union (B))
```

➔▾ {2, 3, 5, 7, 11, 13}

6. Create a frozen set from the original set

```
s = frozenset({1,2,5,4,8,9})
print(s)
```

➔▾ frozenset({1, 2, 4, 5, 8, 9})

7. Check if the set has any common elements with {1, 4, 9}.

```
A = {2,3,5,7,11}
A.intersection({1,4,9})
```

➔▾ set()

8. Remove all elements from the set.

```
A = {2,4,6,8,7,9}
A.clear()
print(A)
```

➔▾ set()

9. Create a set of your favorite fruits and nd the intersection with { 'apple', 'banana', 'orange' }

```
fruits = {'apple', 'banana', 'orange' , 'mango'}
b = { 'apple', 'banana', 'orange' }
print(fruits.intersection(b))
```

⇥ {'orange', 'banana', 'apple'}

10. Use set comprehension to create a set of squares for numbers 1 to 10

```
sqares_set = {x**2 for x in range(1,11)}
print(sqares_set)
```

⇥ {64, 1, 4, 36, 100, 9, 16, 49, 81, 25}

LISTS

1. Create a list of integers from 1 to 5.

```
A = [1,2,3,4,5]
print(A)
```

⇥ [1, 2, 3, 4, 5]

2. Append the number 6 to the list.

```
A = [1,2,3,4,5]
A.append(6)
print(A)
```

⇥ [1, 2, 3, 4, 5, 6]

3. Extend the list with another list: [7, 8, 9]

```
A = [1,2,3,4,5]
B = [7,8,9]
A.extend(B)
print(A)
```

⇥ [1, 2, 3, 4, 5, 7, 8, 9]

4. Access and print the third element of the list.

```
A = [1,2,3,4,5]
print(A[2])
```

⇥ 3