



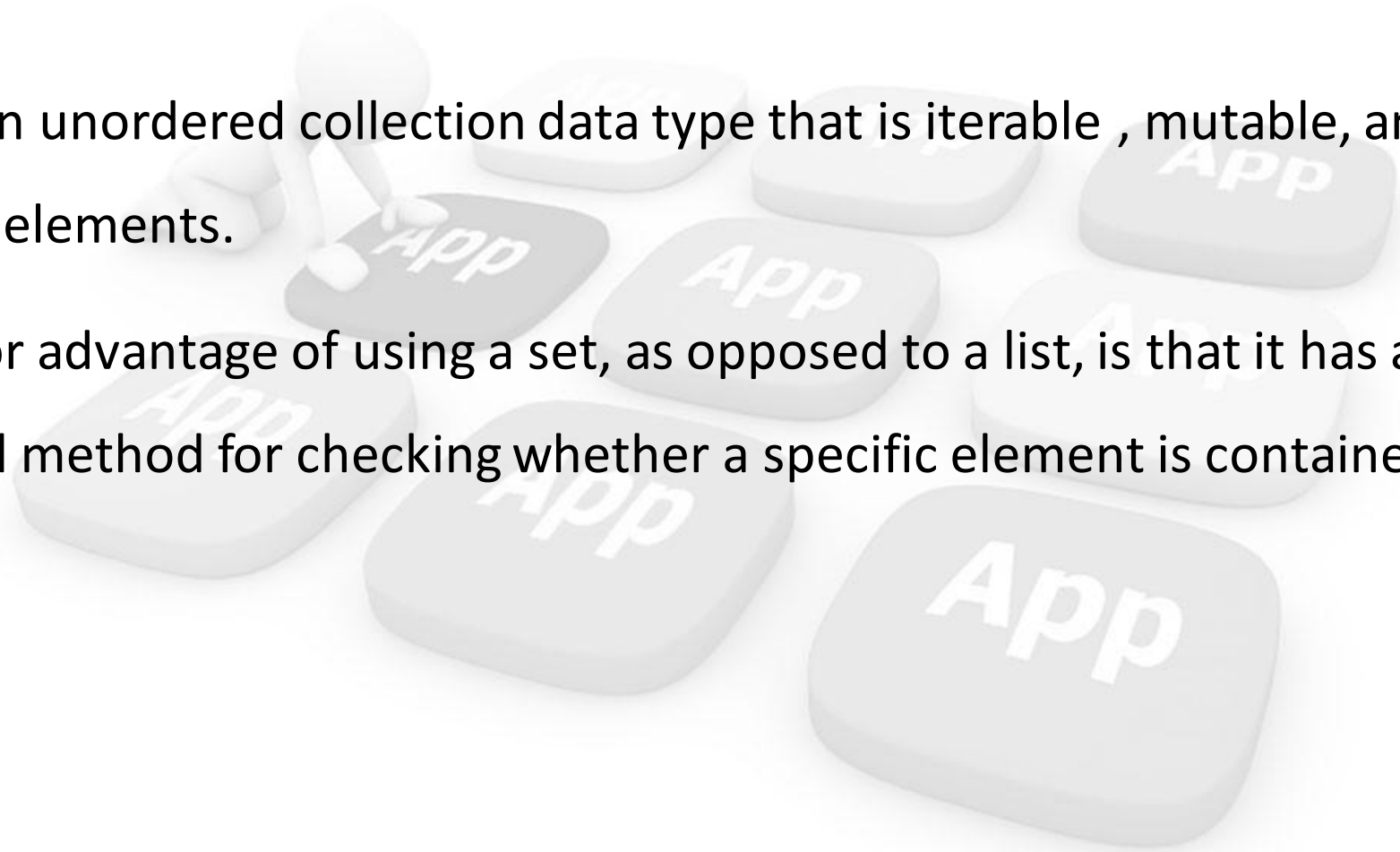
Set



Set

SET

- A Set is an unordered collection data type that is iterable , mutable, and has no duplicate elements.
- The major advantage of using a set, as opposed to a list, is that it has a highly optimized method for checking whether a specific element is contained in the set.



Creation Of Set

- A set is created by using the set() function
- For e.g.

```
words=set(["abc","cde","efg","hij","klm","nop","qrst"])  
alpha={"a","b","c"}  
print(words)  
print(alpha)
```


Adding and Removing of elements from set

```
alpha={"a","b","c"}  
alpha.add('t')  
print(alpha)
```

```
alpha.discard('t')  
print(alpha)
```

Discard()

Removes the element from the set if and only if the element is present in the set. If the element is not present in the set, then no error will be raised and the original set is printed.



METHOD	DESCRIPTION
add()	Adds the item to set if it is not already present in the set.
union()	Returns a union of two set. Using the ' ' operator union is done.
intersect()	Returns an intersection of two sets . The '&' operator comes can also be used in this case.
difference()	Returns a set containing all the elements of invoking set but not of the second set. We can use '-' operator here.
clear()	Empties the whole set.

UNION

The union operation on two sets produces a new set containing all the distinct elements from both the sets. Union() and |

```
words={'abc','cde','efg','hij','klm','nop','qrst'}
```

```
alpha={"a","b","c"}
```

```
s=words | alpha
```

```
print(s)
```

INTERSECTION

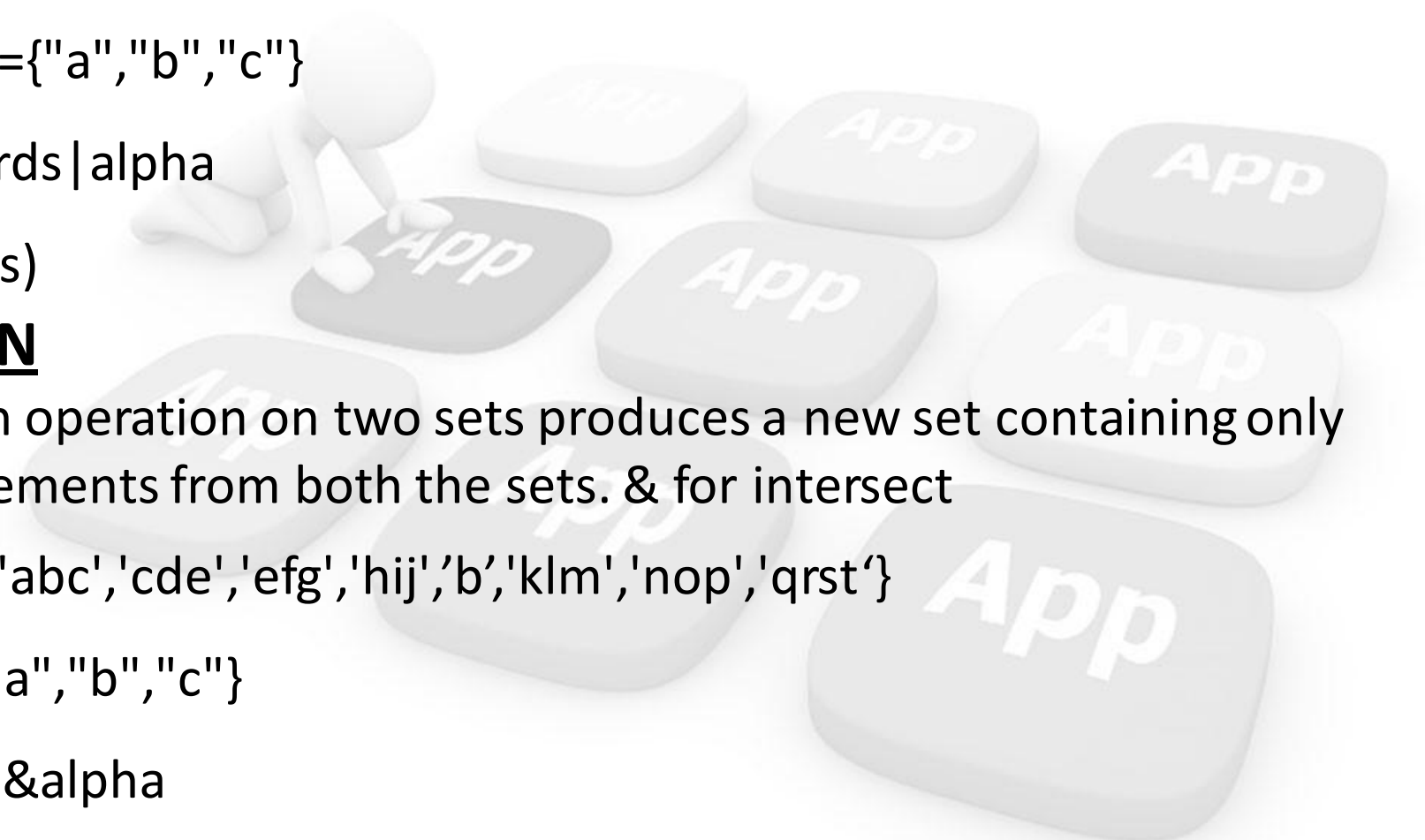
The intersection operation on two sets produces a new set containing only the common elements from both the sets. & for intersect

```
words={'abc','cde','efg','hij','b','klm','nop','qrst'}
```

```
alpha={"a","b","c"}
```

```
s=words&alpha
```

```
print(s)
```



DIFFERENCE OF SETS

The difference operation on two sets produces a new set containing only the elements from the first set and none from the second set. Although it won't print the common elements that are present in both the set objects

```
words={'abc','cde','efg','hij','b','klm','nop','qrst'}  
alpha={"a","b","c"}  
s=words-alpha  
print(s)
```

COMPARE SETS

We can check if a given set is a subset or superset of another set. The result is True or False depending on the elements present in the sets.

```
words={1,2,3,4,5,6,7,8,9}  
alpha={1,2,3}  
s=words<=alpha  
d=words>=alpha  
print(d)  
print(s)
```