

## Set comprehensions

```
S = { expression for item in iterable }
```

# set

```
S = set()
```

# method of creating an empty set

```
S1 = { x for x in range(10) }
```

# it will take values from 0 to 9

```
S2 = { x**2 for x in [-2, -1, 0, 1, 2] }
```

# it gives the square of the elements

```
S3 = { x for x in (10, 5, 7, 8, 12, 3) if x%2==0 }
```

# it contains a condition

```
S4 = { x.upper() for x in 'philippines' }
```

Unhashable type set

### IDLE:

```
s={}
type(s)
<class 'dict'>
s=set()
type(s)
<class 'set'>
for x in range(1, 11):
    s.add(x)
```

```
s
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

Same thing can be done using set comprehensions

```
s1={x for x in range(1, 11)}
s1
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

```
s2={x**2 for x in [-2, -1, 0, 1, 2]}
s2
{0, 1, 4}
```

# it is giving the square of the element

$$(-2)^2 = 4$$

$$(-1)^2 = 1$$

$$0^2 = 0$$

$$1^2 = 1$$

$$2^2 = 4$$

So, if we notice 4 repeating two times so, it is just taking single element.

```
s3={x for x in (10,5,7,8,12,3)}.  
if(x%2==0)
```

# this is having a condition also. It means just pickup even number from the tuple.

```
s4={x.upper() for x in 'phillipines'}
```

# unhashable type set is not allowed because mutable are not washable.

A set have all mutable types and also list cannot be a member of set