

Solution Review: Find Duplicates in a List

This lesson will help find duplicates in a list.

WE'LL COVER THE FOLLOWING ^

- Solution: Use a Nested **for** Loop

Solution: Use a Nested **for** Loop

Use a nested for loop where you have to check for duplicates on the rest of the list for each value.

- The outer loop selects one element and checks for duplicates over the list—through the inner-loop by comparing values.
- If a duplicate is found, it sets the flag to true.

	0	1	2	3
<code>list</code>	1	2	3	3

1 of 9

	0	1	2	3
<code>list</code>	1	2	3	3

```
list[0]==list[1]  
flag=0
```

2 of 9

	0	1	2	3
list	1	2	3	3

```
list[0]==list[2]
```

```
flag=0
```

3 of 9

	0	1	2	3
list	1	2	3	3

```
list[0]==list[3]
```

```
flag=0
```

4 of 9

	0	1	2	3
list	1	2	3	3

```
list[1]==list[2]
```

```
flag=0
```

5 of 9

	0	1	2	3
list	1	2	3	3

```
list[1]==list[3]
```

```
flag=0
```

6 of 9

list

0	1	2	3
1	2	3	3

```
list[2]==list[3]  
flag=1
```

7 of 9

list

0	1	2	3
1	2	3	3

```
list[2]==list[3]  
flag=1
```

8 of 9

```
flag=1  
List contains duplicates
```

9 of 9



The following python code helps to check if the list has duplicate values.

```
def has_duplicates(list):  
    flag = 0  
    for i in range (len(list)):  
        for j in range (i+1,len(list)):  
            if (list[i] == list[j]):  
                flag = 1  
    if (flag == 1):  
        return True  
    else:  
        return False
```



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

```
l=[1, 2, 3, 3, 4]  
print(has_duplicates(l))
```



Let's move on to the next problem.