

BIS 420 PROGRAMMING FOR DATA SCIENCE
PRAJAKTA POHARE
CHAPTER 10 EXERCISE 11.9
ILLINOIS STATE UNIVERSITY

If you did Exercise 10.8, you already have a function named `has_duplicates` that takes a list as a parameter and returns `True` if there is any object that appears more than once in the list. Use a dictionary to write a faster, simpler version of `has_duplicates`. Solution: http://thinkpython.com/code/has_duplicates.py.

```
from __future__ import print_function, division
```

```
def has_duplicates(t):
```

```
    d = {}
```

```
    for x in t:
```

```
        if x in d:
```

```
            return True
```

```
        d[x] = True
```

```
    return False
```

```
def has_duplicates2(t):
```

```
    return len(set(t)) < len(t)
```

```
if __name__ == '__main__':
```

```
    t = [1, 2, 3]
```

```
    print(has_duplicates(t))
```

```
t.append(1)
```

```
print(has_duplicates(t))
```

```
t = [1, 2, 3]
```

```
print(has_duplicates2(t))
```

```
t.append(1)
```

```
print(has_duplicates2(t))
```

```
Users > prajaktapohare > Library > CloudStorage > OneDrive-ILStateUniversity > BIS420 > We
1  from __future__ import print_function, division
2
3  def has_duplicates(t):
4      d = {}
5      for x in t:
6          if x in d:
7              return True
8          d[x] = True
9      return False
10
11 def has_duplicates2(t):
12     return len(set(t)) < len(t)
13
14 if __name__ == '__main__':
15     t = [1, 2, 3]
16     print(has_duplicates(t))
17     t.append(1)
18     print(has_duplicates(t))
19
20     t = [1, 2, 3]
21     print(has_duplicates2(t))
22     t.append(1)
23     print(has_duplicates2(t))
```