In [11]:
```
word = "OCLA"
for letter in word:
    if letter == 'L':
        break
    print(letter)
```

```
O
C
```

In [10]:
```
word = "OCLA"
for letter in word:
    if letter == 'P':
        print("Letter P is bad")
        break
else:
    print("No P found!")
```

```
No P found!
```

Things to do:

6.1 Use a for loop to print the values of the list [3, 2, 1, 0].

In [18]:
```
for val in range(3,-1,-1):
    print(list(val))
#the reverse countdown is not working. It doesn't give any error too.
```

In [73]:
```
list_val=[]
for val in range(0,4,1):
    list_val.append(val)
print(list_val[::-1]) #reversed!
```

```
[3, 2, 1, 0]
```

6.2 Assign the value 7 to the variable guess_me, and the value 1 to the variable number. Write a while loop that compares number with guess_me. Print 'too low' if number is less than guess me. If number equals guess_me, print 'found it!' and then exit the loop. If number is greater than guess_me, print 'oops' and then exit the loop. Increment number at the end of the loop.

In [65]:
```
guess_me = 7
number = 1

while guess_me >= number - 1:
    if number < guess_me:
        print("Too Low")
    elif number == guess_me:
        print("Found it!")
    else:
        print ("Oops!")
    number += 1
```

```
Too Low
Too Low
Too Low
Too Low
Too Low
Too Low
Found it!
oops!
```

6.3 Assign the value 5 to the variable guess_me.Use a for loop to iteratea variable called number over range(10).If number is less than guess_me, print 'too low'.If it equals guess_me, print found it! and then breakout of the for loop.If number is greater than guess_me.print 'oops' and then exit the loop.

In [66]:
```
guess_me = 5
for number in range(10):
    if number < guess_me:
        print("Too Low!")
    elif number == guess_me:
        print("Found it!")
    else:
        print("Oops!")
```

```
Too Low!
Too Low!
Too Low!
Too Low!
Too Low!
Found it!
Oops!
Oops!
Oops!
Oops!
```

Chapter 7

Things to do:

7.1 Create a list called years_list, starting with the year of your birth, and each year thereafter until the year of your fifth birthday.For example, if you were born in 1980, the list would be: years_list = [1980, 1981, 1982, 1983, 1984, 1985]. If you're less than five years old and reading this book, I don't know what to tell you.

In [78]:
```
bday_years =[]
for years in range(1992,1998):
    bday_years.append(years)

print(bday_years)
```

```
[1992, 1993, 1994, 1995, 1996, 1997]
```

7.2 In which year in years_list was your third birthday? Remember, you were 0 years of age for your first year.

In [81]:
```
print("My third Birthday was in: ", bday_years[3])
```

```
My third Birthday was in:  1995
```

7.3 In which year in years_list were you the oldest?

In [82]:
```
print("I was oldest in the year: ", bday_years[5])
```

```
I was oldest in the year:  1997
```

7.4 Make a list called things with these three strings as elements: "mozzarella", "cinderella", "salmonella".

In [107]:
```
things = ["mozzarella", "cinderella", "salmonella"]
print(things)
```

```
['mozzarella', 'cinderella', 'salmonella']
```

7.5 Capitalize the element in things that refers to a person and then print the list. Did it change the element in the list?

In [109]:
```
print(things[1].capitalize())
print(things)

# No the list is unchanged.
```

```
Cinderella
['mozzarella', 'cinderella', 'salmonella']
```

7.6 Make the cheesy element of thingsall uppercase and then print the list.

In [109]:
```
print(things[0].upper())
print(things)

# No the list is unchanged.
```

```
MOZZARELLA
['mozzarella', 'cinderella', 'salmonella']
```

In [110]:
```
del things[2]

print(things)
```

```
['mozzarella', 'cinderella']
```

7.8 Create a list called surprise with the elements "Groucho", "Chico", and "Harpo".

In [111]:
```
surprise = ["Groucho", "Chico", "Harpo"]
```

7.9 Lowercase the last element of the surprise list, reverse it, and then capitalize it.

In [122]:
```
lower = surprise[2].lower()
print(lower)
print(lower[::-1])
print(rev)
print(rev.capitalize())
```

```
harpo
oprah
Oprah
```

7.10 Use a list comprehension to make a list called even of the even numbers in range(10).

In [124]:
```
even_list = [num for num in range(10) if num % 2 == 0]
print(even_list)
```

```
[0, 2, 4, 6, 8]
```

7.11 Let's create a jump rope rhyme maker. You'll print a series of two-line rhymes. Start with this program fragment:

```
start1 = ["fee", "fie", "foe"]
rhymes = [
    ("flop", "get a mop"),
    ("fope", "turn the rope"),
    ("fa", "get your ma"),
    ("fudge", "call the judge"),
    ("fat", "pet the cat"),
    ("fog", "walk the dog"),
    ("fun", "say we're done"),
    ]
start2 = "Someone better"
```

For each tuple (first, second) in rhymes:

For the first line:

Print each string in start1, capitalized and followed by an exclamation point and a space.

Print first, also capitalized and followed by an exclamation point.

For the second line:

Print start2 and a space.

Print second and a period.

In [128]:
```
start1 = ["fee", "fie", "foe"]
rhymes = [
    ("flop", "get a mop"),
    ("fope", "turn the rope"),
    ("fa", "get your ma"),
    ("fudge", "call the judge"),
    ("fat", "pet the cat"),
    ("fog", "walk the dog"),
    ("fun", "say we're done"),
    ]
start2 = "Someone better"

for first,second in rhymes:
    if first == start1:
        print(first)
```

Out[131]:
```
('flop', 'get a mop'),
 ('fope', 'turn the rope'),
 ('fa', 'get your ma'),
 ('fudge', 'call the judge'),
 ('fat', 'pet the cat'),
 ('fog', 'walk the dog'),
 ('fun', 'say we're done')]
```

Chapter: 8

Things to do

8.1 Make an English-to-French dictionary called e2f and print it. Here are your starter words: dog is chien, cat is chat, and walrus is morse.

In [146]:
```
e2f = {"dog":"chien",
       "cat":"chat",
       "walrus":"morse"}
print(e2f)
```

```
{'dog': 'chien', 'cat': 'chat', 'walrus': 'morse'}
```

8.2 Using your three-word dictionary e2f, print the French word for walrus.

In [172]:
```
print("French equivalent of 'walrus' is: ", e2f["walrus"])
```

```
French equivalent of 'walrus' is:  morse
```

8.3 Make a French-to-English dictionary called f2e from e2f. Use the items method.

In [147]:
```
f2e = {}
for keys,values in e2f.items():
    f2e.update({values:keys})
print(f2e)
```

```
{'chien': 'dog', 'chat': 'cat', 'morse': 'walrus'}
```

8.4 Print the English equivalent of the French word chien.

In [171]:
```
print("English equivalent of 'chien' is: ", f2e['chien'])
```

```
English equivalent of 'chien' is:  dog
```

8.5 Print the set of English words from e2f.

In [175]:
```
print("Set of english words from 'English-to-French dictionary':\n\n", list(e2f.keys()))
```

```
Set of english words from 'English-to-French dictionary':

 ['dog', 'cat', 'walrus']
```

8.6 Make a multilevel dictionary called life. Use these strings for the topmost keys: 'animals', 'plants', and 'other'. Make the 'animals' key refer to another dictionary with the keys 'cats', 'octopi', and 'emus'. Make the 'cats' key refer to a list of strings with the values 'Henri', 'Grumpy', and 'Lucy'. Make all the other keys refer to empty dictionaries.

In [184]:
```
life = {"animals":{"cats":['Henri',"Grumpy","Lucy"],"octopi":{},"emus":{}}, "plants":{},"others":{}}
print(life)
```

```
{'animals': {'cats': ['Henri', 'Grumpy', 'Lucy'], 'octopi': {}, 'emus': {}}, 'plants': {}, 'others': {}}
```

8.7 Print the top-level keys of life.

In [191]:
```
print("Top-level keys of life dictionary:\n\n", list(life.keys()))
```

```
Top-level keys of life dictionary:

 ['animals', 'plants', 'others']
```

8.8 Print the keys for life['animals'].

In [199]:
```
print("Keys for animals from the life dictionary:\n\n", list(life['animals'].keys()))
```

```
Keys for animals from the life dictionary:

 ['cats', 'octopi', 'emus']
```

8.9 Print the values for life['animals'] ['cats'].

In [215]:
```
print("Values for cats within animals from the life dictionary:\n\n", list(life['animals']['cats']))
```

```
Values for cats within animals from the life dictionary:

 ['Henri', 'Grumpy', 'Lucy']
```

8.10 Use a dictionary comprehension to create the dictionary squares. Use range(10) to return the keys, and use the square of each key as its value.

In [219]:
```
dict_sqr = {num:num*num for num in range(10)}
print(dict_sqr)
```

```
{0: 0, 1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81}
```

8.11 Use a set comprehension to create the set odd from the odd numbers in range(10).

In [225]:
```
odd_set = {num for num in range(10) if num%2!=0}
print("Odd numbers in the range 1-10\n\n", odd_set)
```

```
Odd numbers in the range 1-10

 {1, 3, 5, 7, 9}
```

8.12 Use a generator comprehension to return the string 'Got ' and a number for the numbers in range(10). Iterate through this by using a for loop.

In [239]:
```
a="Got "
got_num = (a,num) for num in range(10))
print(got_num)

[('Got ', 0), ('Got ', 1), ('Got ', 2), ('Got ', 3), ('Got ', 4), ('Got ', 5), ('Got ', 6), ('Got ', 7), ('Got ', 8), ('Got ', 9)]
```

8.13 Use zip() to make a dictionary from the key tuple ('optimist', 'pessimist', 'troll') and the values tuple ('The glass is half full', 'The glass is half empty', 'How did you get a glass?').

In [254]:
```
key_tuple=('Optimist', 'Pessimist', 'Troll')
value_tuple=('The glass is half full', 'The glass is half empty', 'How did you get a glass?')
print("Now zipped tuple based dictionary:")
dict(zip(key_tuple,value_tuple))
```

```
Now zipped tuple based dictionary:
```

Out[254]:
```
{'Optimist': 'The glass is half full',
 'Pessimist': 'The glass is half empty',
 'Troll': 'How did you get a glass?'}
```

8.14 Use zip() to make a dictionary called movies that pairs these lists: titles = ['Creature of Habit', 'Crewel Fate', 'Sharks On a Plane'] and plots = ['A nun turns into a monster', 'A haunted yarn shop', 'Check your exits']

In [257]:
```
titles = ['Creature of Habit', 'Crewel Fate', 'Sharks On a Plane']
plots = ['A nun turns into a monster', 'A haunted yarn shop', 'Check your exits']
movies = dict(zip(titles,plots))
print("Now zipped list based dictionary:")
movies
```

```
Now zipped list based dictionary:
```

Out[257]:
```
{'Creature of Habit': 'A nun turns into a monster',
 'Crewel Fate': 'A haunted yarn shop',
 'Sharks On a Plane': 'Check your exits'}
```