

29

ANSWERS TO EVEN-NUMBERED EXERCISES

2. Write and run a Python program that you store in a file. The program should demonstrate how to prompt the user for input and display the string the user entered.

```
$ cat mypy.py
#!/usr/bin/python

inp = raw_input('Enter the name of a month: ')
print 'You entered ' + inp

$ python mypy.py
Enter the name of a month: June
You entered June
```

or

```
$ chmod 755 mypy.py
$ ./mypy.py
```

4. Using the Python interactive shell, use a **for** control structure to iterate through the elements of the list you instantiated in exercise 3 and display each abbreviated name followed by a period on a line by itself. (*Hint:* The period is a string.)

```
>>> for nam in mon:
...     print nam + '.'
...
Jan.
Feb.
Mar.
Apr.
May.
Jun.
```

6. Instantiate a dictionary in which the keys are the months in the third quarter of the year and the values are the number of days in the corresponding month. Display the dictionary, the keys, and the values. Add the tenth month of the year to the dictionary and display the value of that month only.

```
>>> days = {'July': 31, 'August': 31, 'September': 30}
>>> days
{'September': 30, 'July': 31, 'August': 31}
>>> days.keys()
['September', 'July', 'August']
>>> days.values()
[30, 31, 31]
>>> days['October'] = 31
>>> days['October']
31
```

8. Write and demonstrate a Lambda function named **stg()** that appends **.txt** to its argument. What happens when you call the function with an integer?

```
>>> stg = lambda b: b + '.txt'
>>> stg('aaa')
'aaa.txt'
```

When called with an integer, the function returns an error when it tries to concatenate an integer and a string:

```
TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

10. Define a function named **cents2** that returns its argument divided by 100 exactly (and includes decimal places if necessary). Make sure your function does not truncate the answer. For example:

```
>>> cents2(12345)
123.45

>>> def cents2(val):
...     return(val / 100.)
```

12. Why does the following assignment statement generate an error?

```
>>> x.y = 5
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'x' is not defined
```

The **x.y** is not a valid name; Python parses it as “apply method **y** to object **x**.” Python reports that **x** is not defined.

14. Use a list comprehension to display the numbers from 1 through 30 inclusive that are divisible by 3.

```
>>> [n for n in range(1,31) if n % 3 == 0]
[3, 6, 9, 12, 15, 18, 21, 24, 27, 30]
```

16. Rewrite exercise 15 to call the function with a random number between 0 and 10 inclusive. (*Hint*: The **randint** function in the **random** library returns a random number between its two arguments inclusive.)

```
$ cat number_guess2.py
#!/usr/bin/python
from random import randint

def guessANumber(val):
    guess = int(input('Enter your guess: '))
    if guess > val:
        print 'Too high.'
        return 1
    elif guess < val:
        print 'Too low.'
        return -1
    else:
        print 'Got it!'
        return 0

val = randint(0,10)
while (guessANumber(val) != 0):
    print 'Guess again.'
```

18. Write a function that counts the vowels (**aeiou**) in a string the user inputs. Make sure it counts upper- and lowercase vowels. Then write a routine that calls the function and displays the following output.

```
$ ./count_vowels.py
Enter some words: Go East young man!
The string "Go East young man!" has 6 vowels in it.

$ cat count_vowels.py
#!/usr/bin/python
def countVowels(my_string):
    count = 0
    for s in my_string:
        if s.lower() in 'aeiou':
            count += 1
    return count

stg = raw_input('Enter some words: ')
print 'The string \'' + stg + '\'' + ' has ',
print countVowels(stg),
print 'vowels in it.'
```

Following is an alternative function that performs the same task:

```
$ cat count_vowels2.py
#!/usr/bin/python
def countVowels(my_string):
    my_string = my_string.lower()
    return len(my_string) - len(my_string.translate(None,'aeiou'))

stg = raw_input('Enter some words: ')
print 'The string \'' + stg + '\'' + ' has ',
print countVowels(stg),
print 'vowels in it.'
```