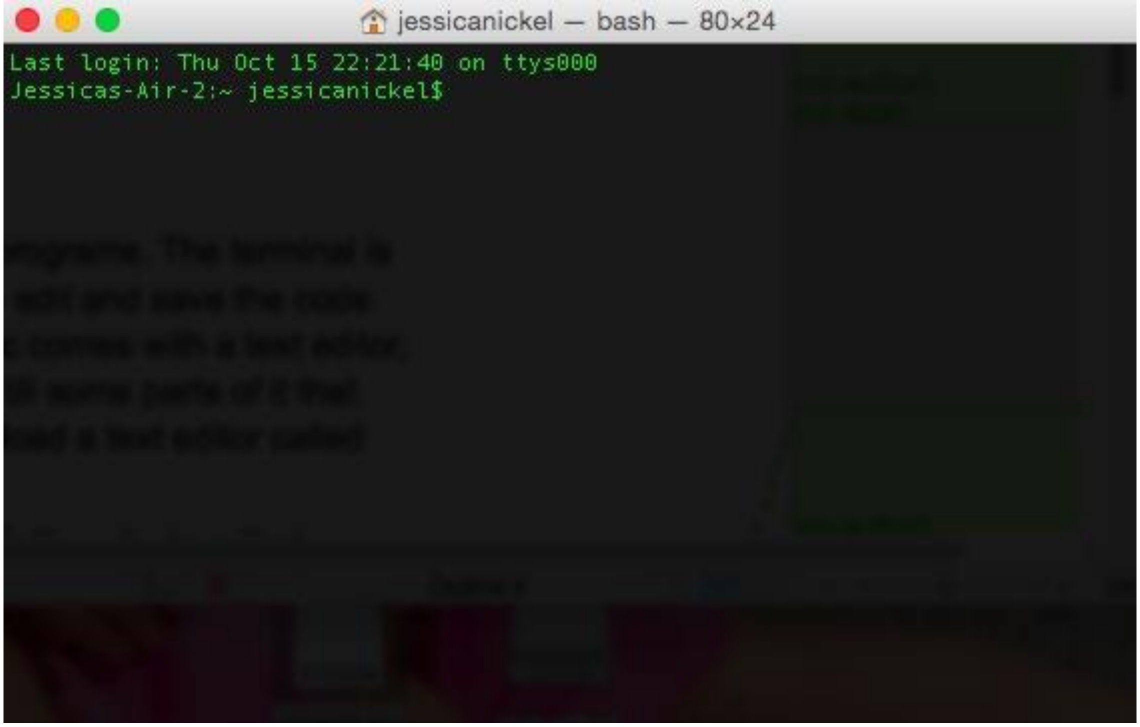
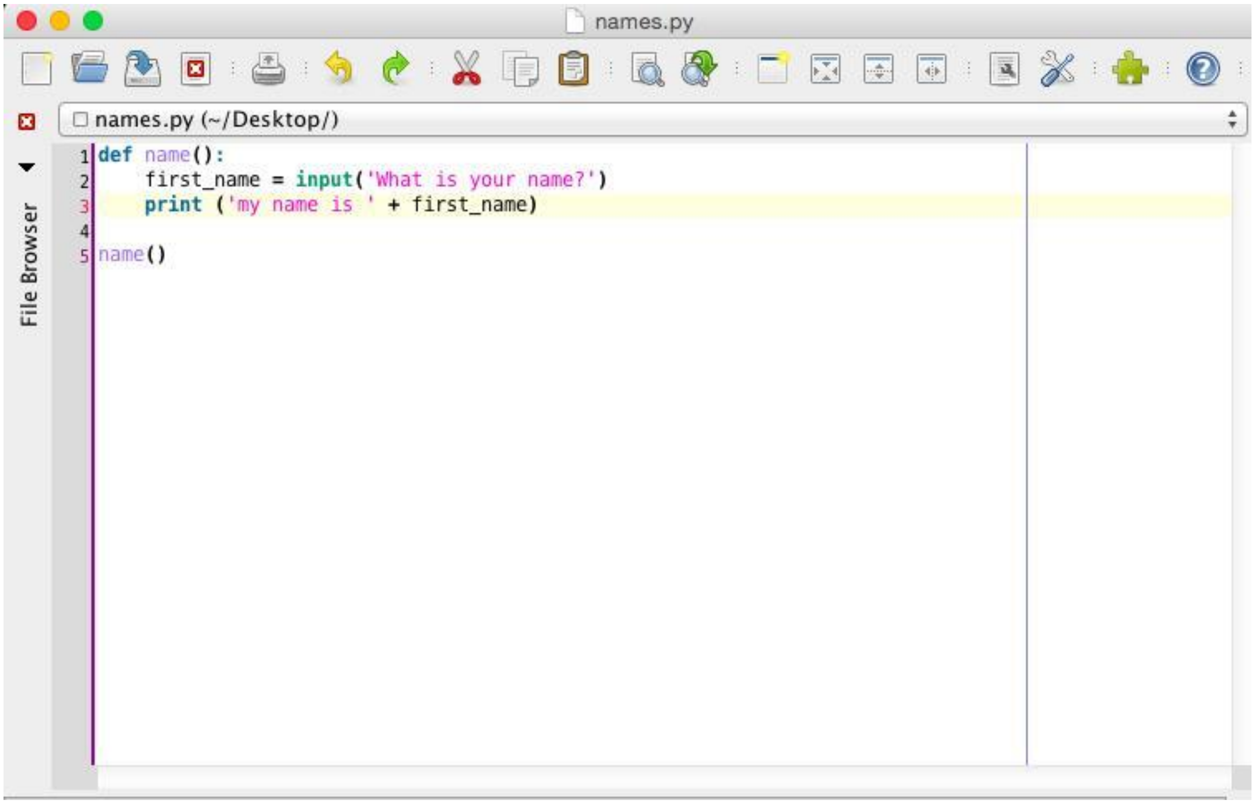
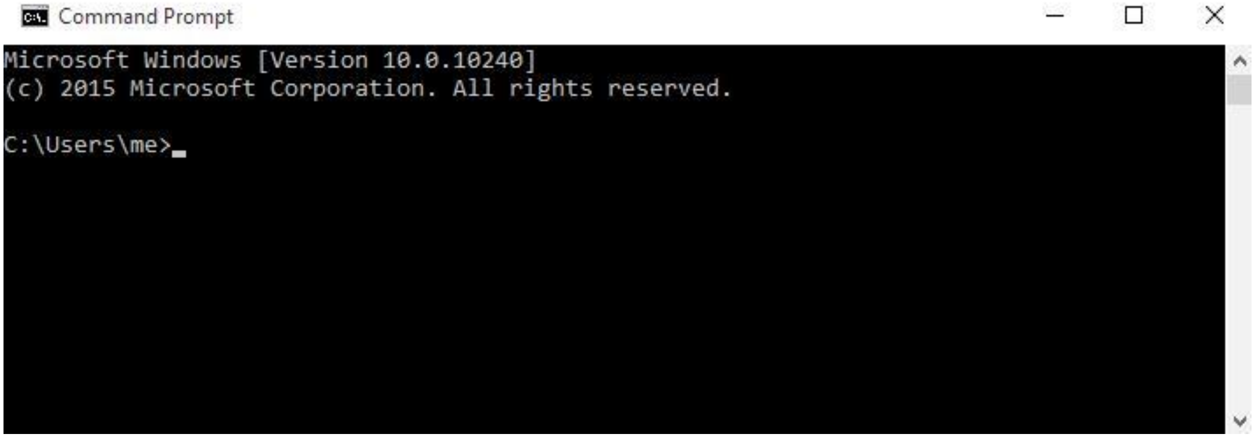
Chapter 1: Welcome! Let's Get Started



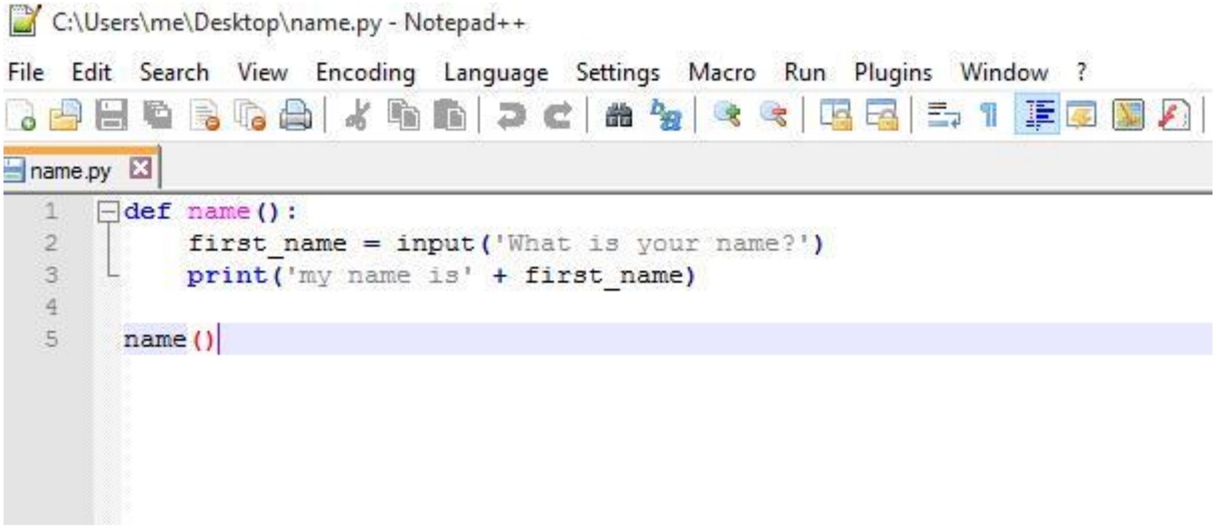
```
Last login: Thu Oct 15 22:21:40 on ttys000
Jessicas-Air-2:~ jessicanickel$
```



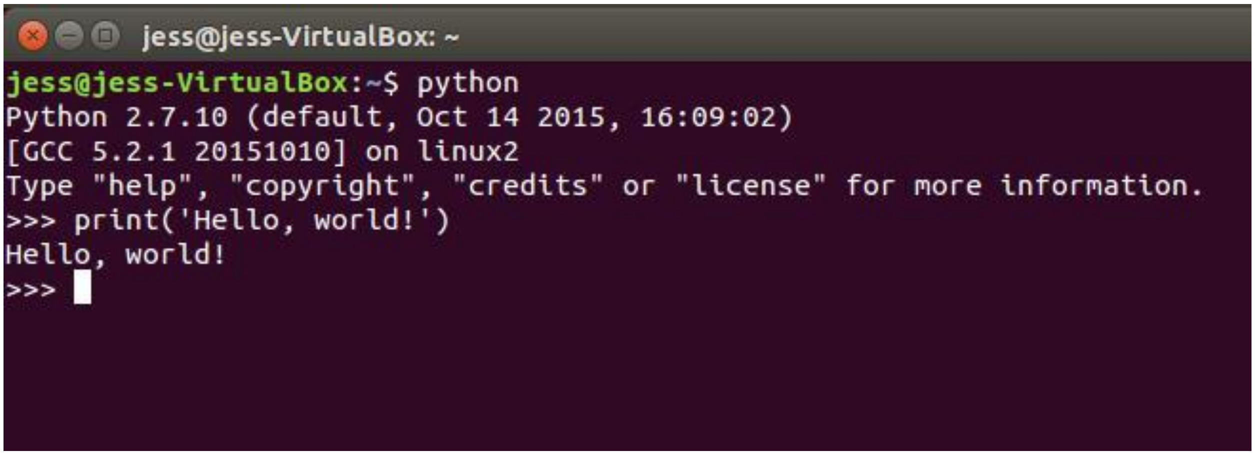
```
def name():
    first_name = input('What is your name?')
    print ('my name is ' + first_name)

name()
```



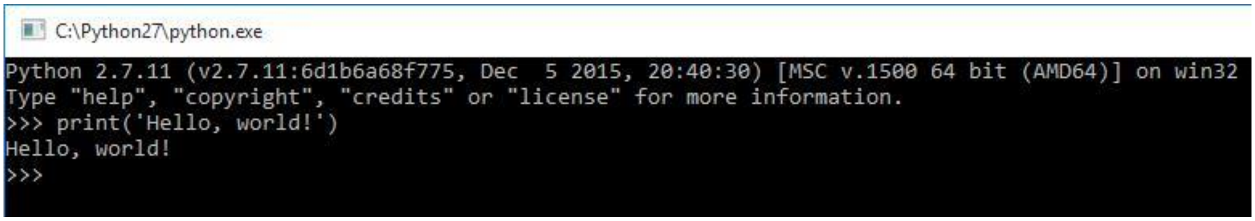
```
Microsoft Windows [Version 10.0.10240]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\me>
```



```
def name():
    first_name = input('What is your name?')
    print('my name is' + first_name)

name()
```

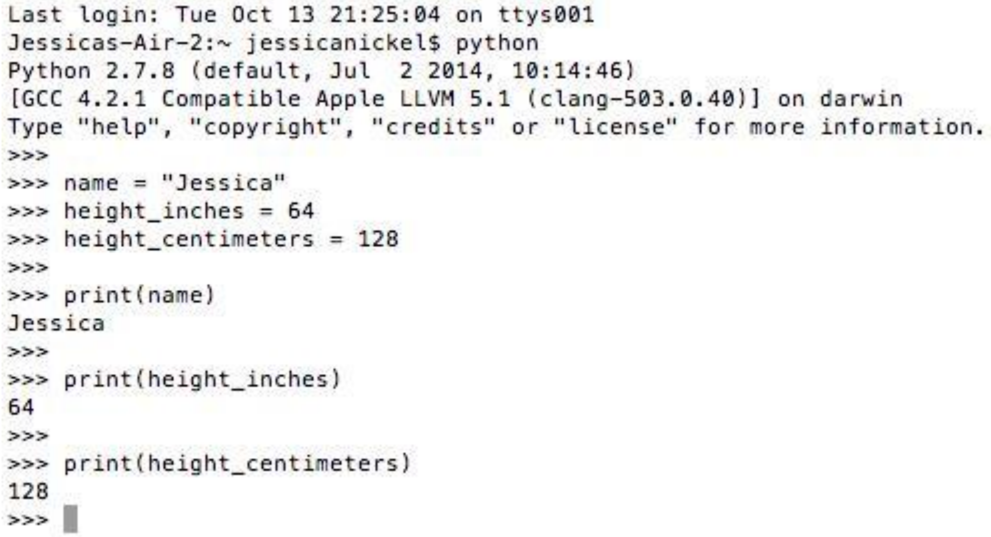


```
jess@jess-VirtualBox:~$ python
Python 2.7.10 (default, Oct 14 2015, 16:09:02)
[GCC 5.2.1 20151010] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> print('Hello, world!')
Hello, world!
>>> 
```

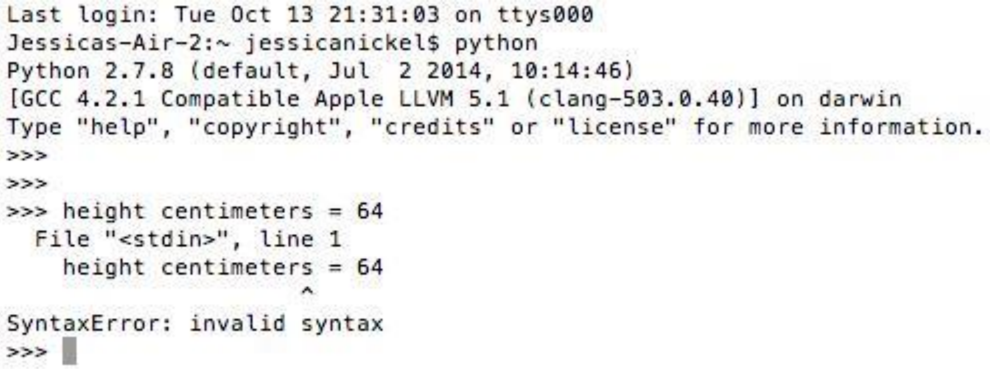


```
Python 2.7.11 (v2.7.11:6d1b6a68f775, Dec  5 2015, 20:40:30) [MSC v.1500 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print('Hello, world!')
Hello, world!
>>>
```

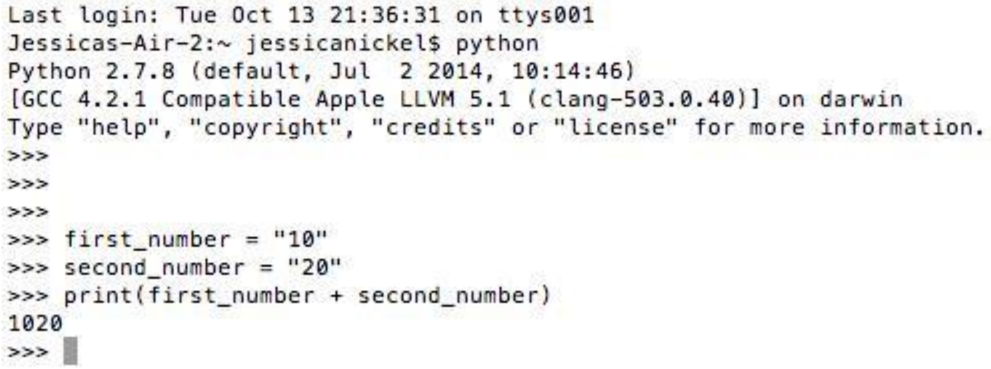
Chapter 2: Variables, Functions, and Users



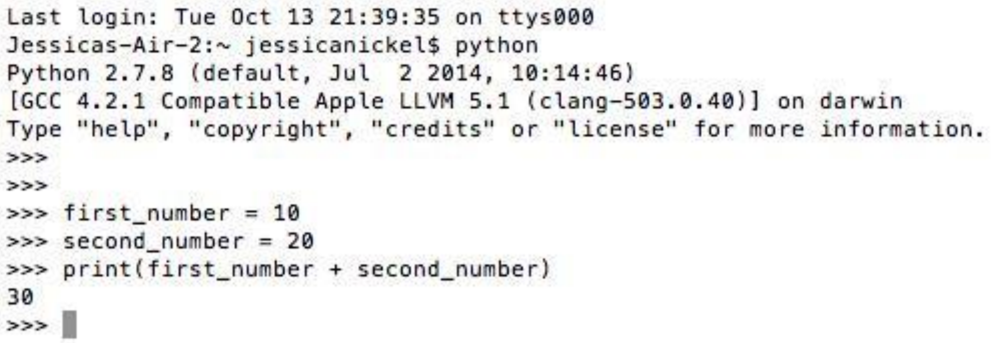
```
Last login: Tue Oct 13 21:25:04 on ttys001
Jessicas-Air-2:~ jessicanickel$ python
Python 2.7.8 (default, Jul  2 2014, 10:14:46)
[GCC 4.2.1 Compatible Apple LLVM 5.1 (clang-503.0.40)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>> name = "Jessica"
>>> height_inches = 64
>>> height_centimeters = 128
>>>
>>> print(name)
Jessica
>>>
>>> print(height_inches)
64
>>>
>>> print(height_centimeters)
128
>>>
```



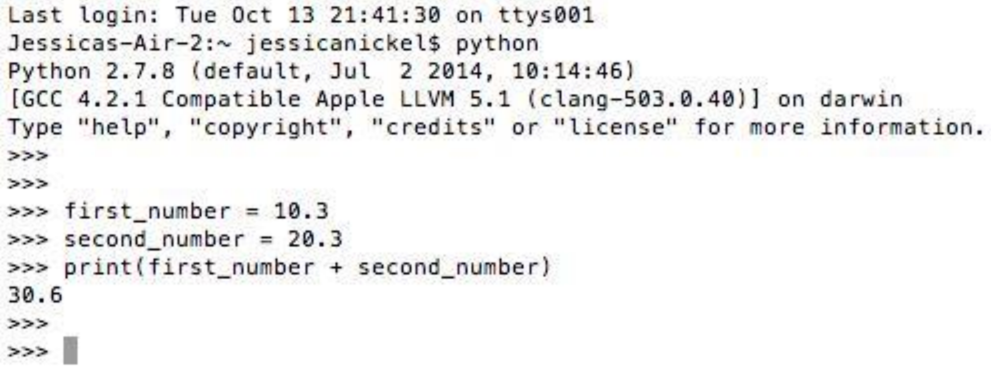
```
Last login: Tue Oct 13 21:31:03 on ttys000
Jessicas-Air-2:~ jessicanickel$ python
Python 2.7.8 (default, Jul  2 2014, 10:14:46)
[GCC 4.2.1 Compatible Apple LLVM 5.1 (clang-503.0.40)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>>
>>> height centimeters = 64
  File "<stdin>", line 1
    height centimeters = 64
                     ^
SyntaxError: invalid syntax
>>>
```



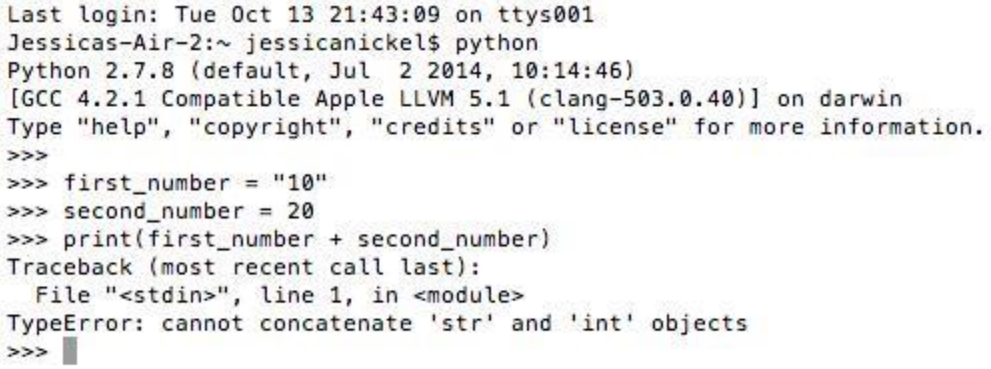
```
Last login: Tue Oct 13 21:36:31 on ttys001
Jessicas-Air-2:~ jessicanickel$ python
Python 2.7.8 (default, Jul  2 2014, 10:14:46)
[GCC 4.2.1 Compatible Apple LLVM 5.1 (clang-503.0.40)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>>
>>>
>>> first_number = "10"
>>> second_number = "20"
>>> print(first_number + second_number)
1020
>>>
```



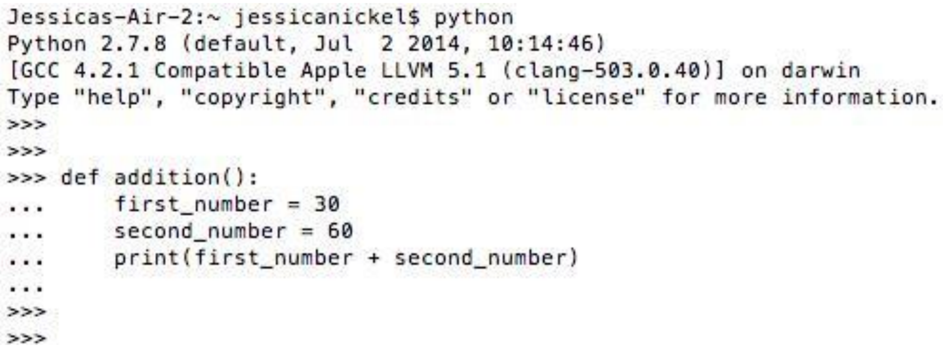
```
Last login: Tue Oct 13 21:39:35 on ttys000
Jessicas-Air-2:~ jessicanickel$ python
Python 2.7.8 (default, Jul  2 2014, 10:14:46)
[GCC 4.2.1 Compatible Apple LLVM 5.1 (clang-503.0.40)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>>
>>> first_number = 10
>>> second_number = 20
>>> print(first_number + second_number)
30
>>>
```



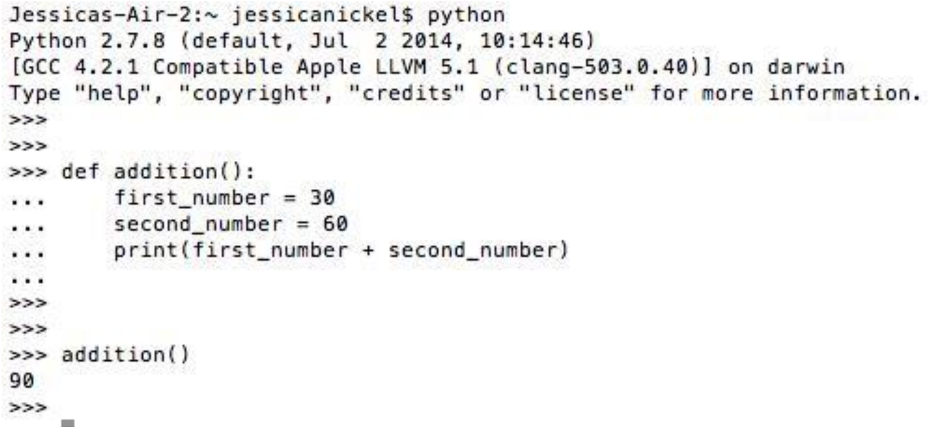
```
Last login: Tue Oct 13 21:41:30 on ttys001
Jessicas-Air-2:~ jessicanickel$ python
Python 2.7.8 (default, Jul  2 2014, 10:14:46)
[GCC 4.2.1 Compatible Apple LLVM 5.1 (clang-503.0.40)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>>
>>> first_number = 10.3
>>> second_number = 20.3
>>> print(first_number + second_number)
30.6
>>>
>>>
```



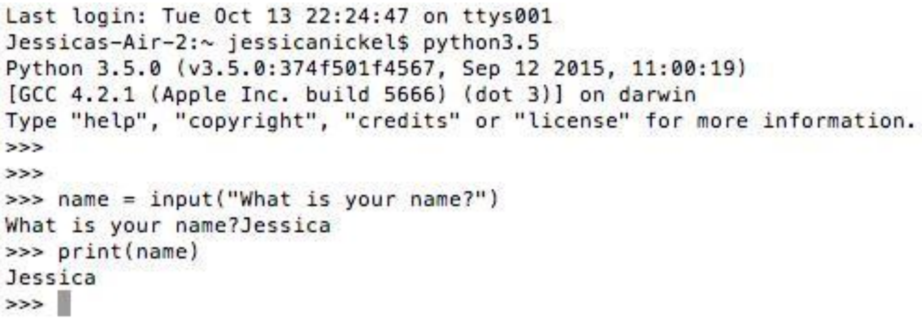
```
Last login: Tue Oct 13 21:43:09 on ttys001
Jessicas-Air-2:~ jessicanickel$ python
Python 2.7.8 (default, Jul  2 2014, 10:14:46)
[GCC 4.2.1 Compatible Apple LLVM 5.1 (clang-503.0.40)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>> first_number = "10"
>>> second_number = 20
>>> print(first_number + second_number)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: cannot concatenate 'str' and 'int' objects
>>>
```



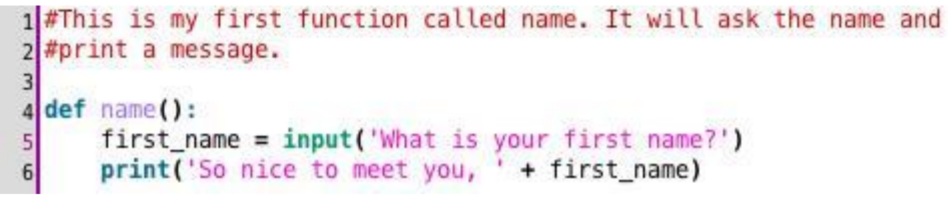
```
Jessicas-Air-2:~ jessicanickel$ python
Python 2.7.8 (default, Jul  2 2014, 10:14:46)
[GCC 4.2.1 Compatible Apple LLVM 5.1 (clang-503.0.40)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>>
>>> def addition():
...     first_number = 30
...     second_number = 60
...     print(first_number + second_number)
...
>>>
>>>
```



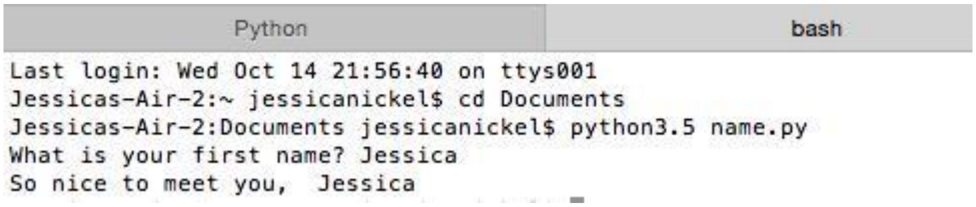
```
Jessicas-Air-2:~ jessicanickel$ python
Python 2.7.8 (default, Jul  2 2014, 10:14:46)
[GCC 4.2.1 Compatible Apple LLVM 5.1 (clang-503.0.40)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>>
>>> def addition():
...     first_number = 30
...     second_number = 60
...     print(first_number + second_number)
...
>>>
>>>
>>> addition()
90
>>>
```



```
Last login: Tue Oct 13 22:24:47 on ttys001
Jessicas-Air-2:~ jessicanickel$ python3.5
Python 3.5.0 (v3.5.0:374f501f4567, Sep 12 2015, 11:00:19)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>>
>>> name = input("What is your name?")
What is your name?Jessica
>>> print(name)
Jessica
>>>
```

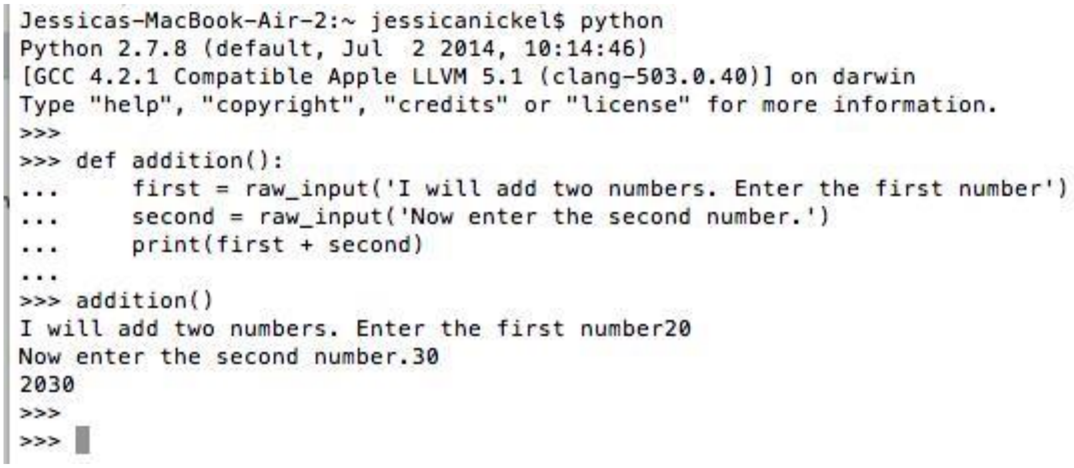


```
#This is my first function called name. It will ask the name and
#print a message.

def name():
    first_name = input('What is your first name?')
    print('So nice to meet you, ' + first_name)
```

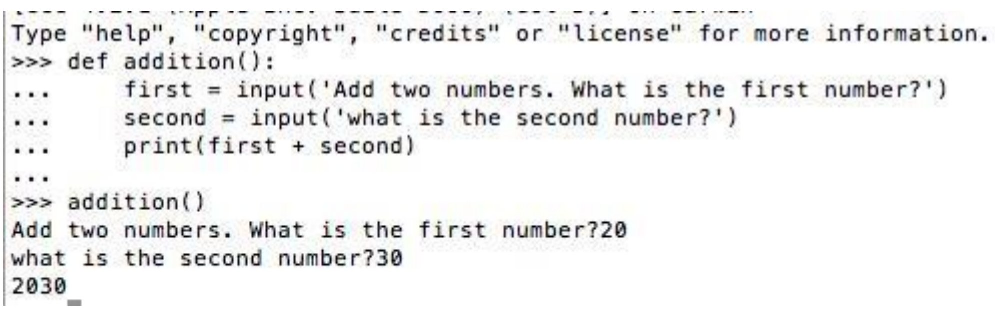


```
Last login: Wed Oct 14 21:56:40 on ttys001
Jessicas-Air-2:~ jessicanickel$ cd Documents
Jessicas-Air-2:Documents jessicanickel$ python3.5 name.py
What is your first name? Jessica
So nice to meet you,  Jessica
```

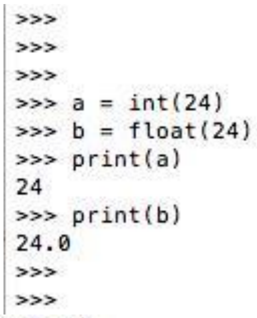
Chapter 3: Calculate This!



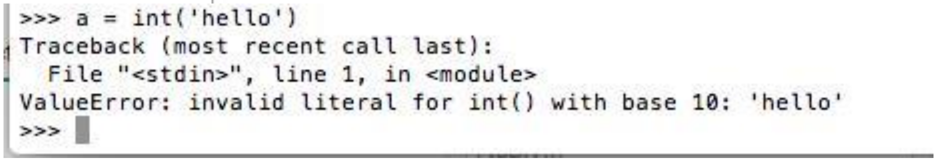
```
Jessicas-MacBook-Air-2:~ jessicanickel$ python
Python 2.7.8 (default, Jul  2 2014, 10:14:46)
[GCC 4.2.1 Compatible Apple LLVM 5.1 (clang-503.0.40)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>> def addition():
...     first = raw_input('I will add two numbers. Enter the first number')
...     second = raw_input('Now enter the second number.')
...     print(first + second)
...
>>> addition()
I will add two numbers. Enter the first number20
Now enter the second number.30
2030
>>>
>>> 
```



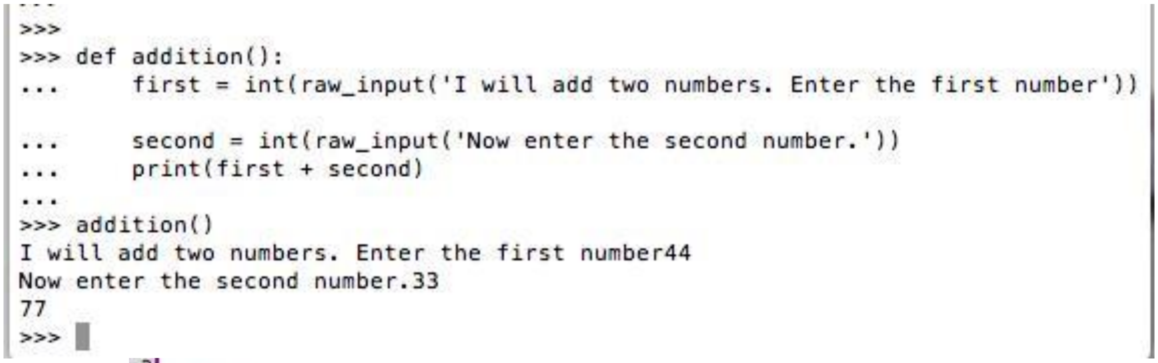
```
Type "help", "copyright", "credits" or "license" for more information.
>>> def addition():
...     first = input('Add two numbers. What is the first number?')
...     second = input('what is the second number?')
...     print(first + second)
...
>>> addition()
Add two numbers. What is the first number?20
what is the second number?30
2030
```



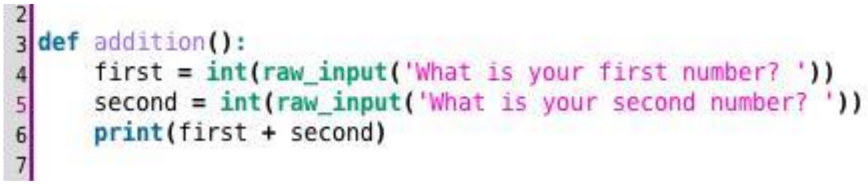
```
>>>
>>>
>>>
>>> a = int(24)
>>> b = float(24)
>>> print(a)
24
>>> print(b)
24.0
>>>
>>>
```



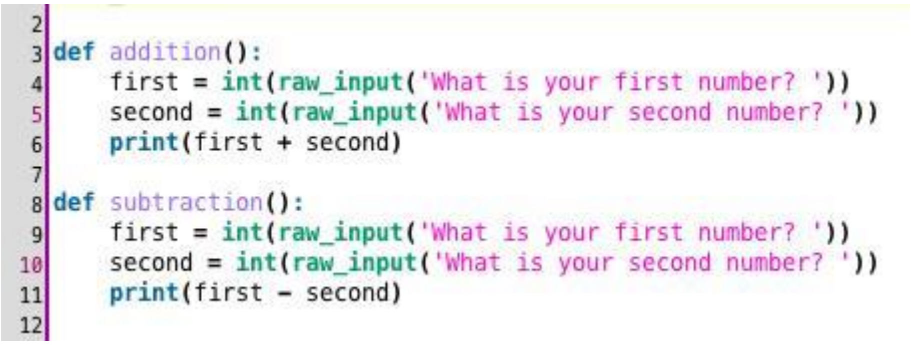
```
>>> a = int('hello')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ValueError: invalid literal for int() with base 10: 'hello'
>>> 
```



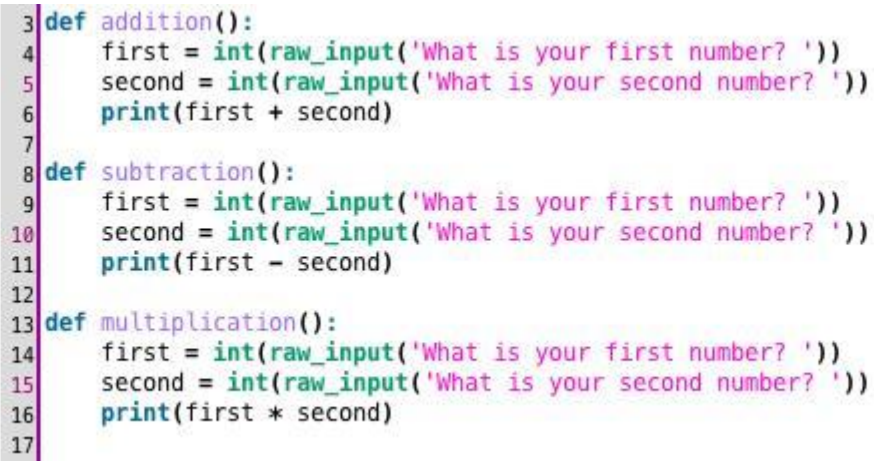
```
...
>>>
>>> def addition():
...     first = int(raw_input('I will add two numbers. Enter the first number'))

...     second = int(raw_input('Now enter the second number.'))
...     print(first + second)
...
>>> addition()
I will add two numbers. Enter the first number44
Now enter the second number.33
77
>>> 
```



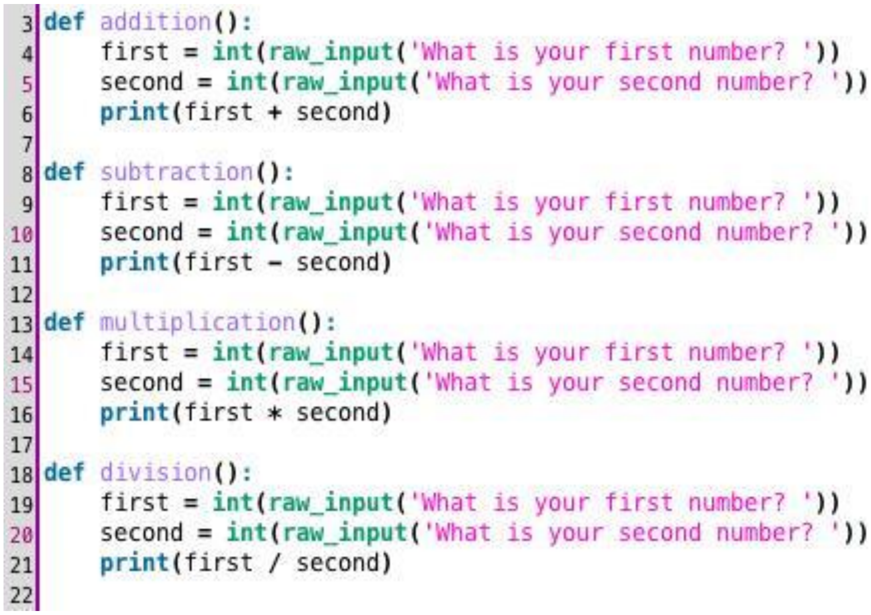
```python

def addition():
    first = int(raw_input('What is your first number? '))
    second = int(raw_input('What is your second number? '))
    print(first + second)

```



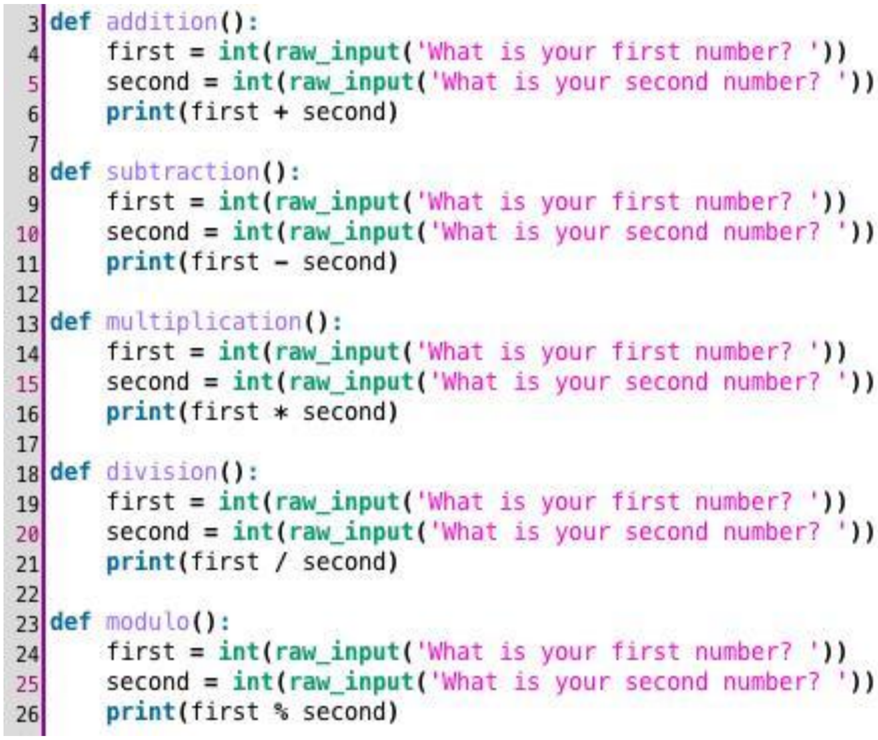
```python

def addition():
    first = int(raw_input('What is your first number? '))
    second = int(raw_input('What is your second number? '))
    print(first + second)

def subtraction():
    first = int(raw_input('What is your first number? '))
    second = int(raw_input('What is your second number? '))
    print(first - second)

```



```python
def addition():
    first = int(raw_input('What is your first number? '))
    second = int(raw_input('What is your second number? '))
    print(first + second)

def subtraction():
    first = int(raw_input('What is your first number? '))
    second = int(raw_input('What is your second number? '))
    print(first - second)

def multiplication():
    first = int(raw_input('What is your first number? '))
    second = int(raw_input('What is your second number? '))
    print(first * second)

```

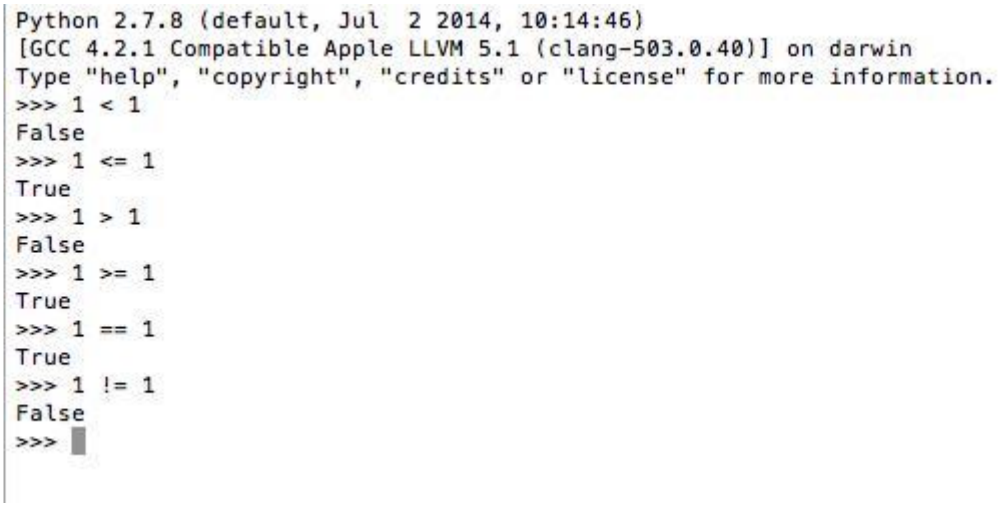


```python
def addition():
    first = int(raw_input('What is your first number? '))
    second = int(raw_input('What is your second number? '))
    print(first + second)

def subtraction():
    first = int(raw_input('What is your first number? '))
    second = int(raw_input('What is your second number? '))
    print(first - second)

def multiplication():
    first = int(raw_input('What is your first number? '))
    second = int(raw_input('What is your second number? '))
    print(first * second)

def division():
    first = int(raw_input('What is your first number? '))
    second = int(raw_input('What is your second number? '))
    print(first / second)

```

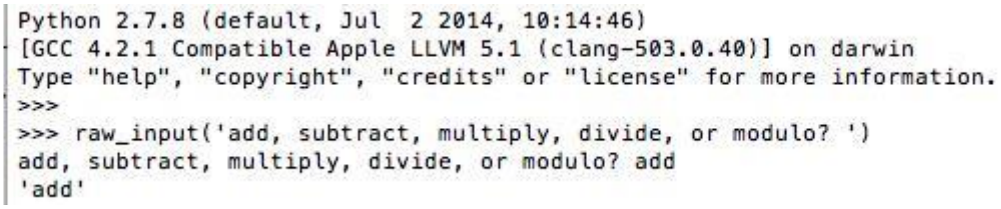


```python
def addition():
    first = int(raw_input('What is your first number? '))
    second = int(raw_input('What is your second number? '))
    print(first + second)

def subtraction():
    first = int(raw_input('What is your first number? '))
    second = int(raw_input('What is your second number? '))
    print(first - second)

def multiplication():
    first = int(raw_input('What is your first number? '))
    second = int(raw_input('What is your second number? '))
    print(first * second)

def division():
    first = int(raw_input('What is your first number? '))
    second = int(raw_input('What is your second number? '))
    print(first / second)

def modulo():
    first = int(raw_input('What is your first number? '))
    second = int(raw_input('What is your second number? '))
    print(first % second)
```

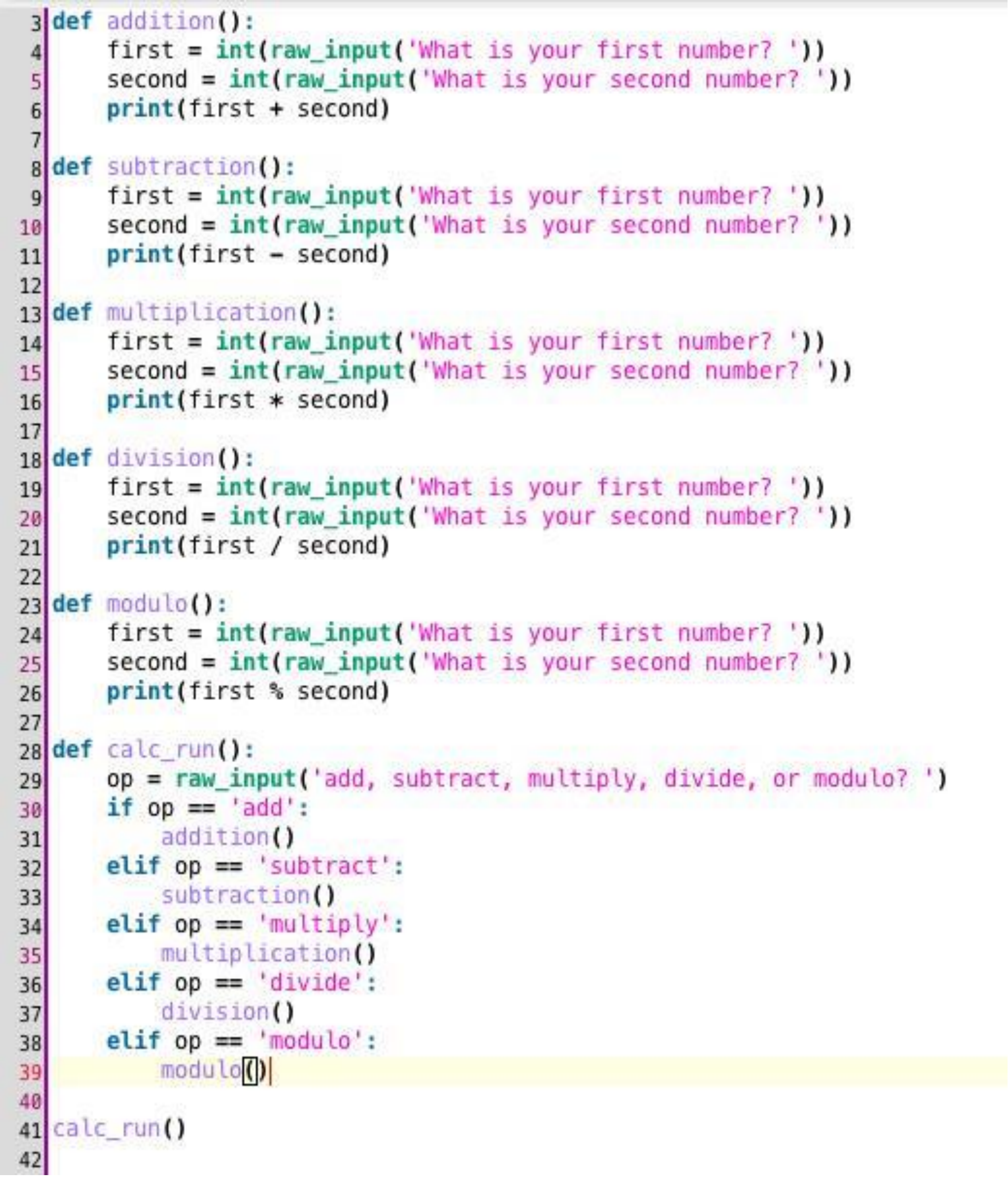
Chapter 4: Making Decisions – Python Control Flows



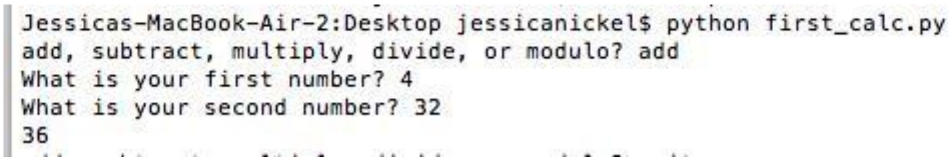
```
Python 2.7.8 (default, Jul  2 2014, 10:14:46)
[GCC 4.2.1 Compatible Apple LLVM 5.1 (clang-503.0.40)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> 1 < 1
False
>>> 1 <= 1
True
>>> 1 > 1
False
>>> 1 >= 1
True
>>> 1 == 1
True
>>> 1 != 1
False
>>> 
```



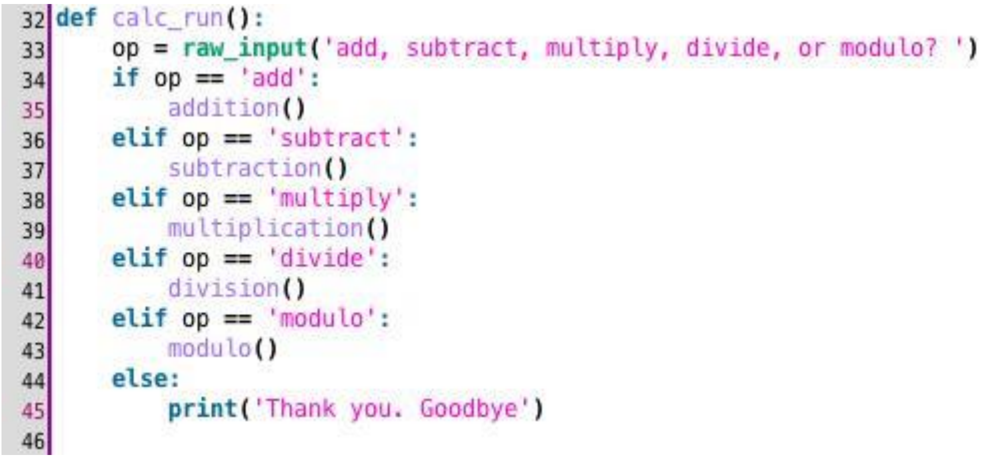
```
Python 2.7.8 (default, Jul  2 2014, 10:14:46)
[GCC 4.2.1 Compatible Apple LLVM 5.1 (clang-503.0.40)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>> raw_input('add, subtract, multiply, divide, or modulo? ')
add, subtract, multiply, divide, or modulo? add
'add'
```



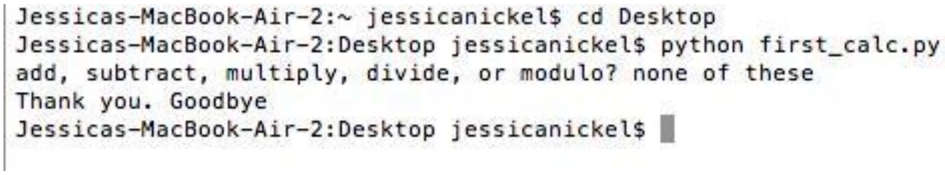
```
def addition():
    first = int(raw_input('What is your first number? '))
    second = int(raw_input('What is your second number? '))
    print(first + second)

def subtraction():
    first = int(raw_input('What is your first number? '))
    second = int(raw_input('What is your second number? '))
    print(first - second)

def multiplication():
    first = int(raw_input('What is your first number? '))
    second = int(raw_input('What is your second number? '))
    print(first * second)

def division():
    first = int(raw_input('What is your first number? '))
    second = int(raw_input('What is your second number? '))
    print(first / second)

def modulo():
    first = int(raw_input('What is your first number? '))
    second = int(raw_input('What is your second number? '))
    print(first % second)

def calc_run():
    op = raw_input('add, subtract, multiply, divide, or modulo? ')
    if op == 'add':
        addition()
    elif op == 'subtract':
        subtraction()
    elif op == 'multiply':
        multiplication()
    elif op == 'divide':
        division()
    elif op == 'modulo':
        modulo()

calc_run()
```



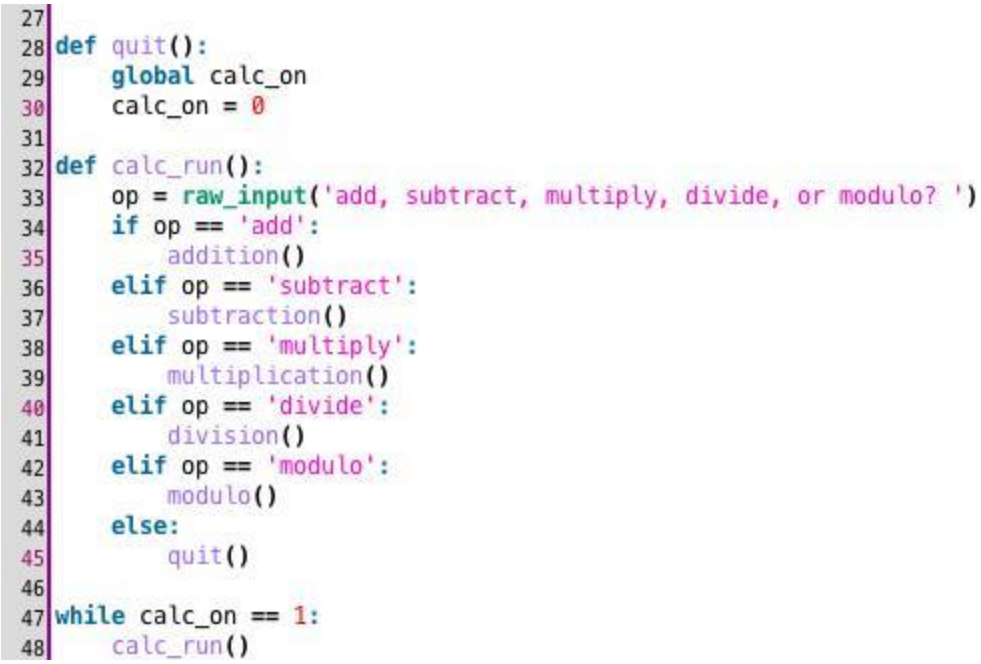
```
Jessicas-MacBook-Air-2:Desktop jessicanickel$ python first_calc.py
add, subtract, multiply, divide, or modulo? add
What is your first number? 4
What is your second number? 32
36
```



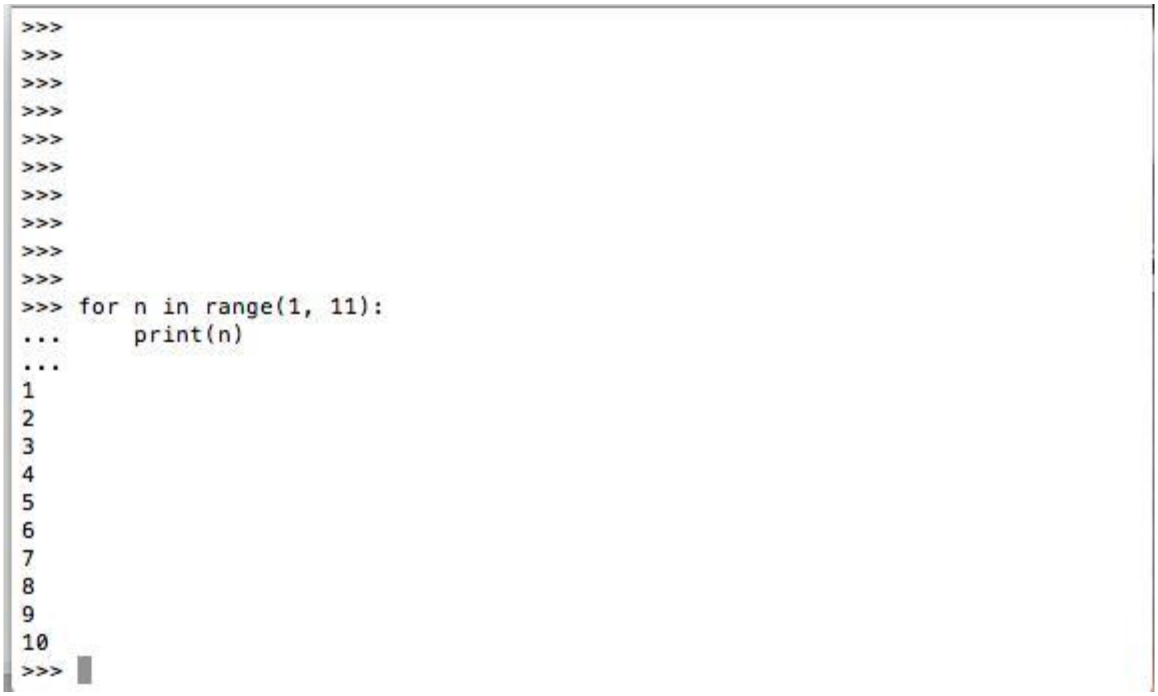
```python
def calc_run():
    op = raw_input('add, subtract, multiply, divide, or modulo? ')
    if op == 'add':
        addition()
    elif op == 'subtract':
        subtraction()
    elif op == 'multiply':
        multiplication()
    elif op == 'divide':
        division()
    elif op == 'modulo':
        modulo()
    else:
        print('Thank you. Goodbye')

```



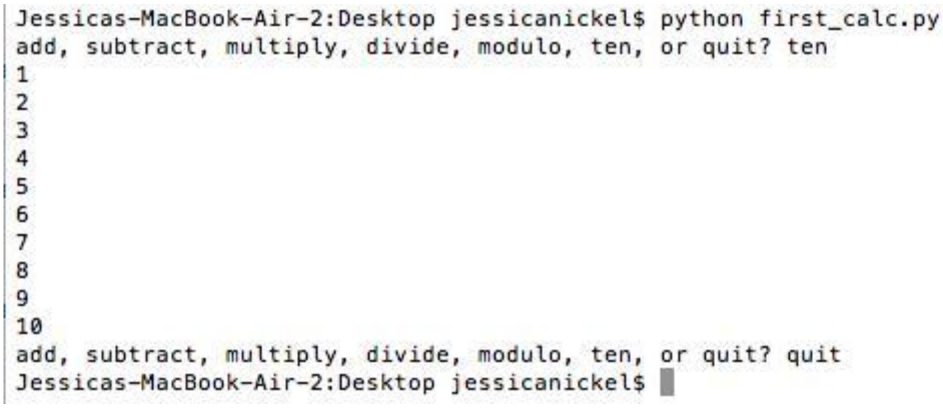
```
Jessicas-MacBook-Air-2:~ jessicanickel$ cd Desktop
Jessicas-MacBook-Air-2:Desktop jessicanickel$ python first_calc.py
add, subtract, multiply, divide, or modulo? none of these
Thank you. Goodbye
Jessicas-MacBook-Air-2:Desktop jessicanickel$ 
```



```python

def quit():
    global calc_on
    calc_on = 0

def calc_run():
    op = raw_input('add, subtract, multiply, divide, or modulo? ')
    if op == 'add':
        addition()
    elif op == 'subtract':
        subtraction()
    elif op == 'multiply':
        multiplication()
    elif op == 'divide':
        division()
    elif op == 'modulo':
        modulo()
    else:
        quit()

while calc_on == 1:
    calc_run()
```

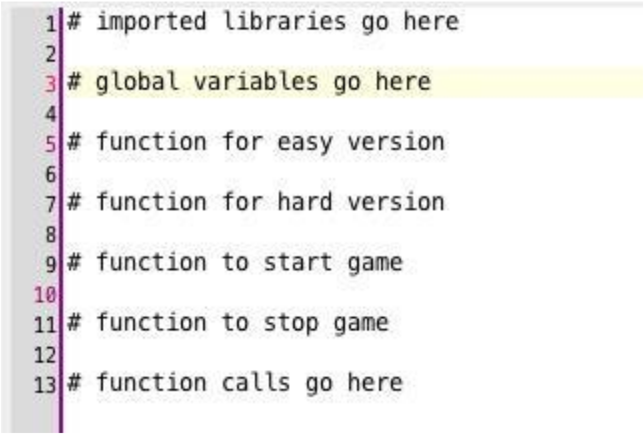


```
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>> for n in range(1, 11):
...     print(n)
...
1
2
3
4
5
6
7
8
9
10
>>> 
```

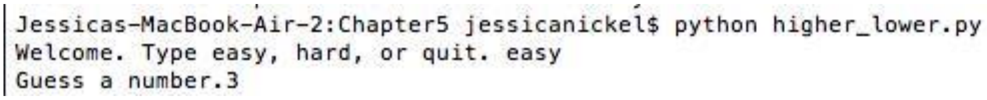


```
Jessicas-MacBook-Air-2:Desktop jessicanickel$ python first_calc.py
add, subtract, multiply, divide, modulo, ten, or quit? ten
1
2
3
4
5
6
7
8
9
10
add, subtract, multiply, divide, modulo, ten, or quit? quit
Jessicas-MacBook-Air-2:Desktop jessicanickel$ 
```

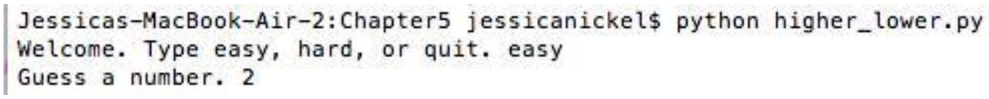
Chapter 5: Loops and Logic



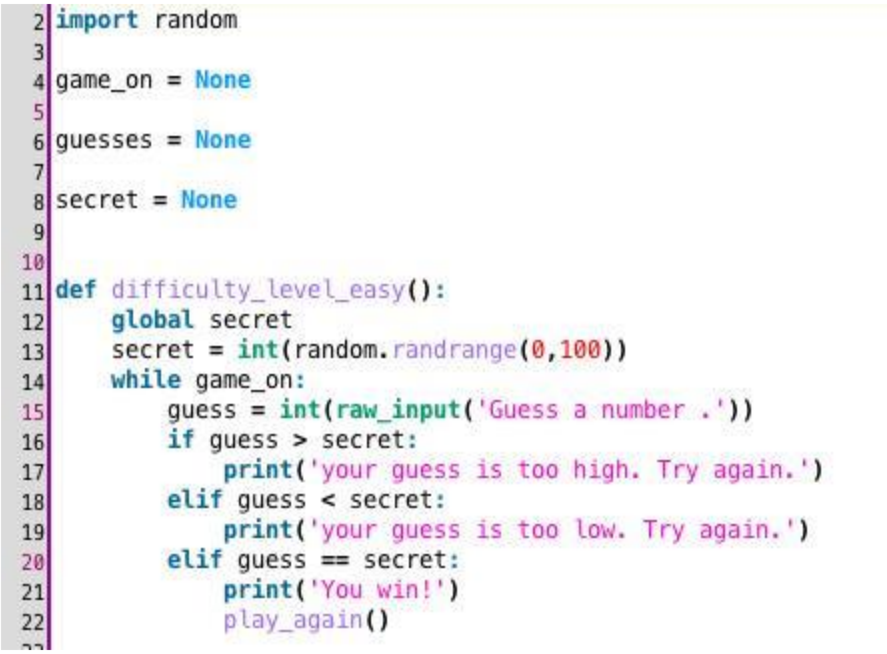
```
# imported libraries go here

# global variables go here

# function for easy version

# function for hard version

# function to start game

# function to stop game

# function calls go here
```



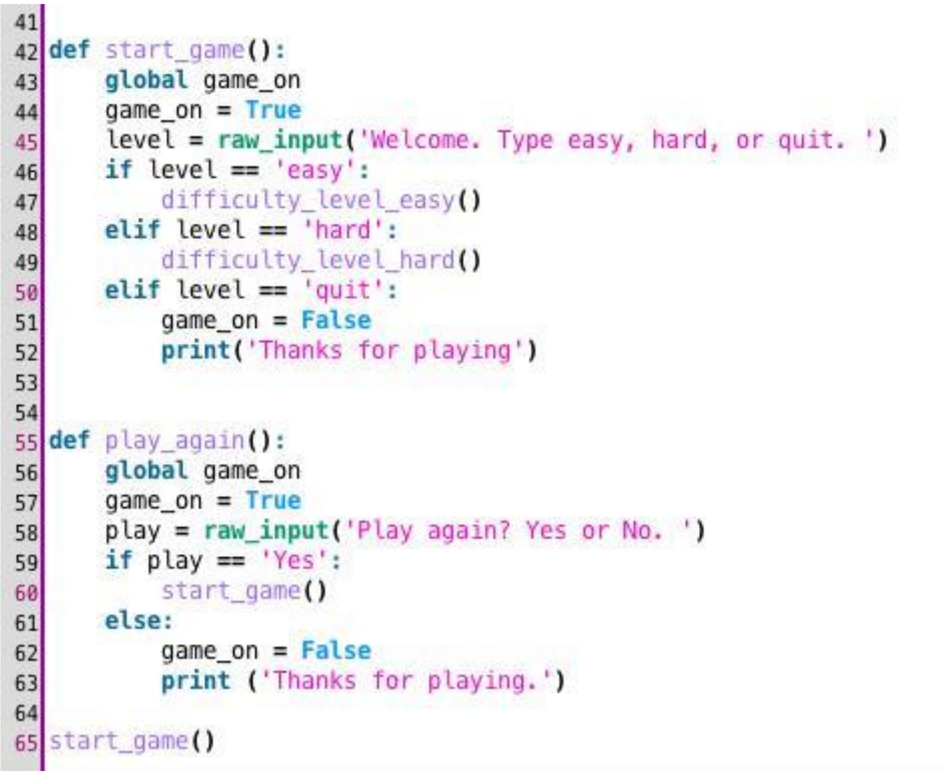
```
Jessicas-MacBook-Air-2:Chapter5 jessicanickel$ python higher_lower.py
Welcome. Type easy, hard, or quit. easy
Guess a number.3
```



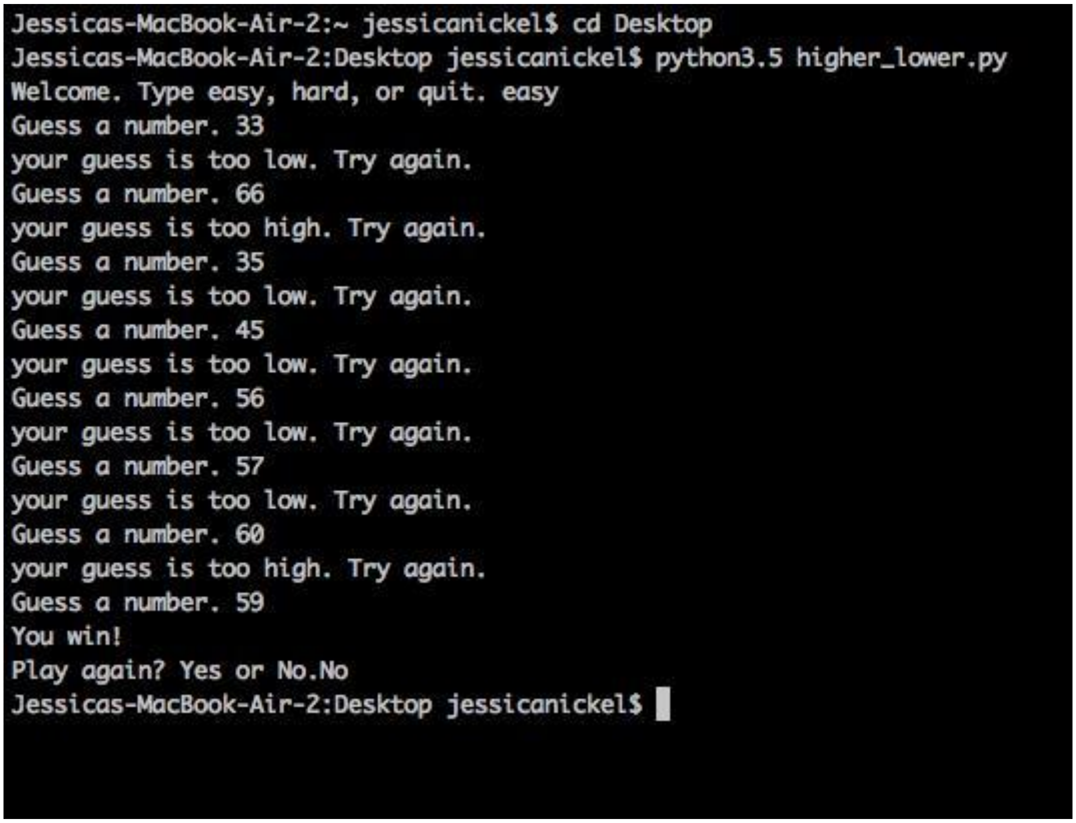
```
Jessicas-MacBook-Air-2:Chapter5 jessicanickel$ python higher_lower.py
Welcome. Type easy, hard, or quit. easy
Guess a number. 2
```



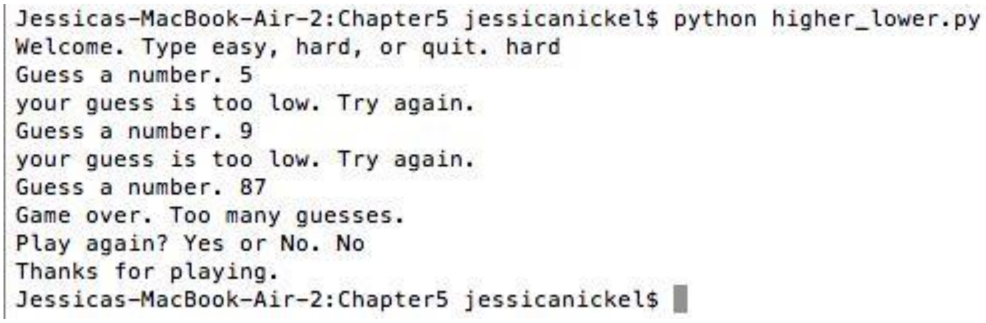
```
import random

game_on = None

guesses = None

secret = None


def difficulty_level_easy():
    global secret
    secret = int(random.randrange(0,100))
    while game_on:
        guess = int(raw_input('Guess a number .'))
        if guess > secret:
            print('your guess is too high. Try again.')
        elif guess < secret:
            print('your guess is too low. Try again.')
        elif guess == secret:
            print('You win!')
            play_again()
```



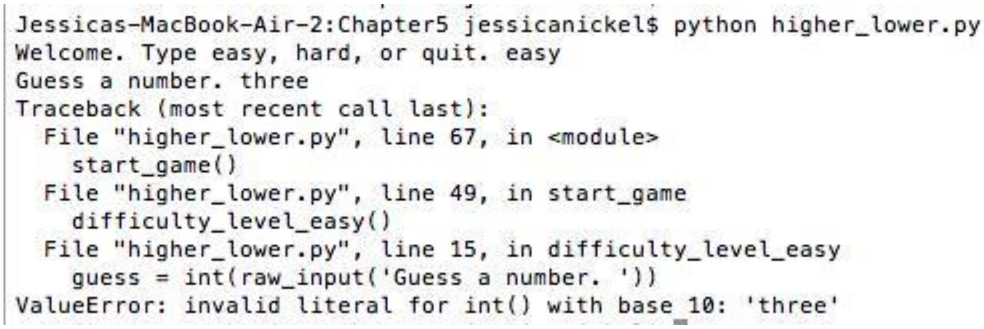
```python

def start_game():
    global game_on
    game_on = True
    level = raw_input('Welcome. Type easy, hard, or quit. ')
    if level == 'easy':
        difficulty_level_easy()
    elif level == 'hard':
        difficulty_level_hard()
    elif level == 'quit':
        game_on = False
        print('Thanks for playing')


def play_again():
    global game_on
    game_on = True
    play = raw_input('Play again? Yes or No. ')
    if play == 'Yes':
        start_game()
    else:
        game_on = False
        print ('Thanks for playing.')

start_game()
```



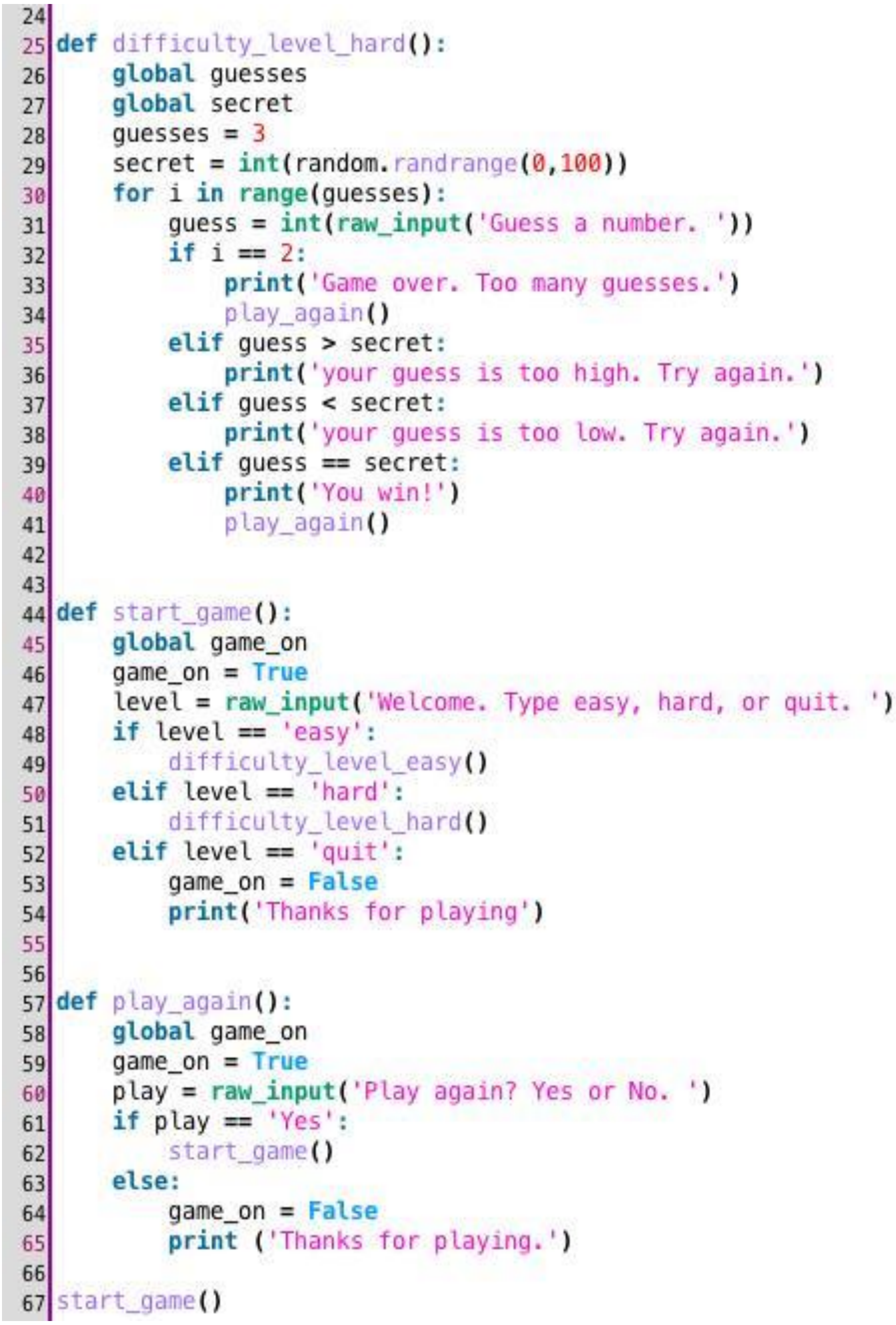
```
Jessicas-MacBook-Air-2:~ jessicanickel$ cd Desktop
Jessicas-MacBook-Air-2:Desktop jessicanickel$ python3.5 higher_lower.py
Welcome. Type easy, hard, or quit. easy
Guess a number. 33
your guess is too low. Try again.
Guess a number. 66
your guess is too high. Try again.
Guess a number. 35
your guess is too low. Try again.
Guess a number. 45
your guess is too low. Try again.
Guess a number. 56
your guess is too low. Try again.
Guess a number. 57
your guess is too low. Try again.
Guess a number. 60
your guess is too high. Try again.
Guess a number. 59
You win!
Play again? Yes or No.No
Jessicas-MacBook-Air-2:Desktop jessicanickel$ 
```



```
Jessicas-MacBook-Air-2:Chapter5 jessicanickel$ python higher_lower.py
Welcome. Type easy, hard, or quit. hard
Guess a number. 5
your guess is too low. Try again.
Guess a number. 9
your guess is too low. Try again.
Guess a number. 87
Game over. Too many guesses.
Play again? Yes or No. No
Thanks for playing.
Jessicas-MacBook-Air-2:Chapter5 jessicanickel$ 
```

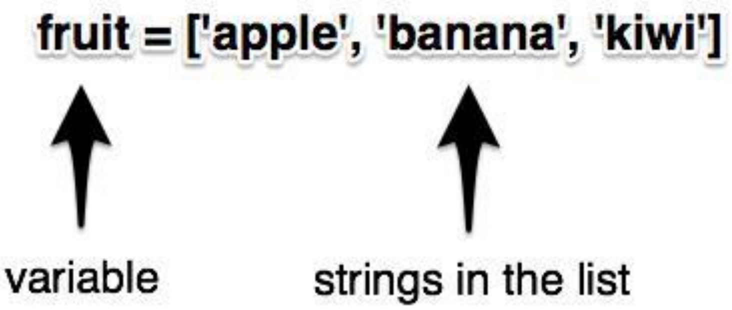


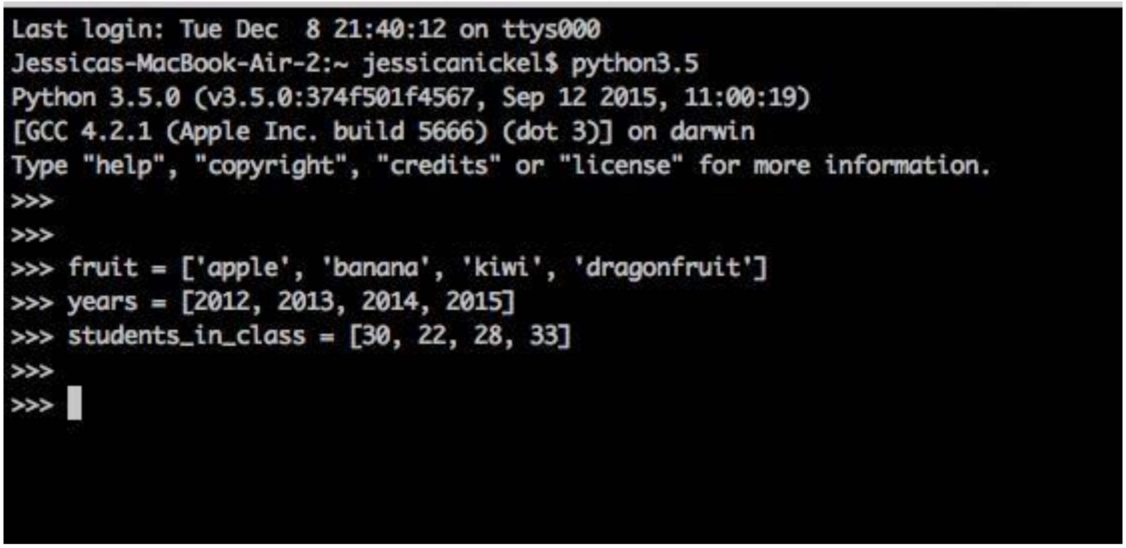
```
Jessicas-MacBook-Air-2:Chapter5 jessicanickel$ python higher_lower.py
Welcome. Type easy, hard, or quit. easy
Guess a number. three
Traceback (most recent call last):
  File "higher_lower.py", line 67, in <module>
    start_game()
  File "higher_lower.py", line 49, in start_game
    difficulty_level_easy()
  File "higher_lower.py", line 15, in difficulty_level_easy
    guess = int(raw_input('Guess a number. '))
ValueError: invalid literal for int() with base 10: 'three'
```



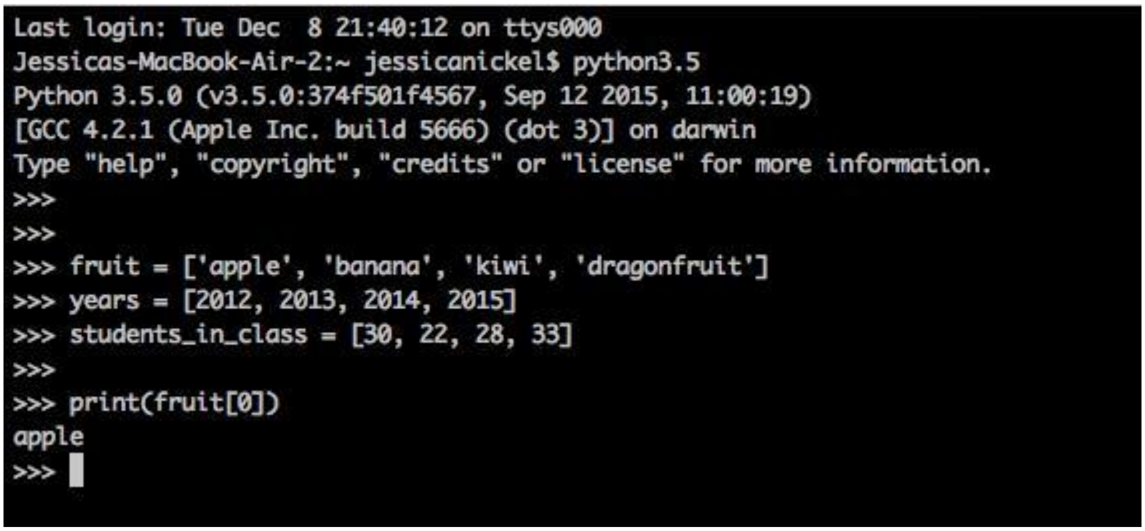
```python
def difficulty_level_hard():
    global guesses
    global secret
    guesses = 3
    secret = int(random.randrange(0,100))
    for i in range(guesses):
        guess = int(raw_input('Guess a number. '))
        if i == 2:
            print('Game over. Too many guesses.')
            play_again()
        elif guess > secret:
            print('your guess is too high. Try again.')
        elif guess < secret:
            print('your guess is too low. Try again.')
        elif guess == secret:
            print('You win!')
            play_again()


def start_game():
    global game_on
    game_on = True
    level = raw_input('Welcome. Type easy, hard, or quit. ')
    if level == 'easy':
        difficulty_level_easy()
    elif level == 'hard':
        difficulty_level_hard()
    elif level == 'quit':
        game_on = False
        print('Thanks for playing')


def play_again():
    global game_on
    game_on = True
    play = raw_input('Play again? Yes or No. ')
    if play == 'Yes':
        start_game()
    else:
        game_on = False
        print ('Thanks for playing.')

start_game()
```

Chapter 6: Working with Data – Lists and Dictionaries

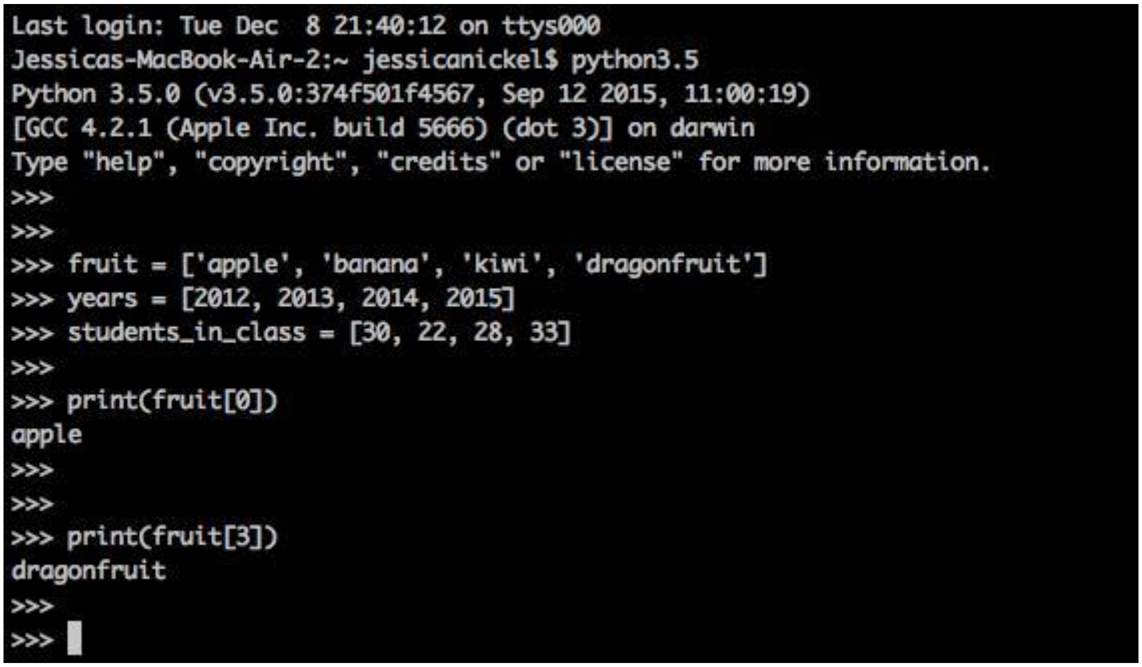




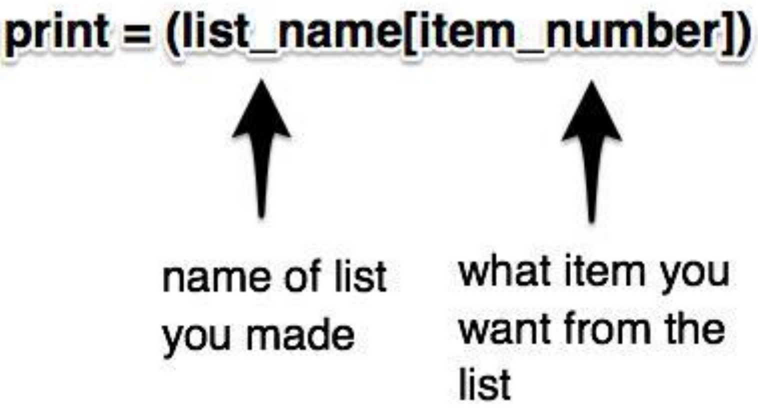
```
Last login: Tue Dec  8 21:40:12 on ttys000
Jessicas-MacBook-Air-2:~ jessicanickel$ python3.5
Python 3.5.0 (v3.5.0:374f501f4567, Sep 12 2015, 11:00:19)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>>
>>> fruit = ['apple', 'banana', 'kiwi', 'dragonfruit']
>>> years = [2012, 2013, 2014, 2015]
>>> students_in_class = [30, 22, 28, 33]
>>>
>>> ▌
```



```
Last login: Tue Dec  8 21:40:12 on ttys000
Jessicas-MacBook-Air-2:~ jessicanickel$ python3.5
Python 3.5.0 (v3.5.0:374f501f4567, Sep 12 2015, 11:00:19)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>>
>>> fruit = ['apple', 'banana', 'kiwi', 'dragonfruit']
>>> years = [2012, 2013, 2014, 2015]
>>> students_in_class = [30, 22, 28, 33]
>>>
>>> print(fruit[0])
apple
>>> ▌
```



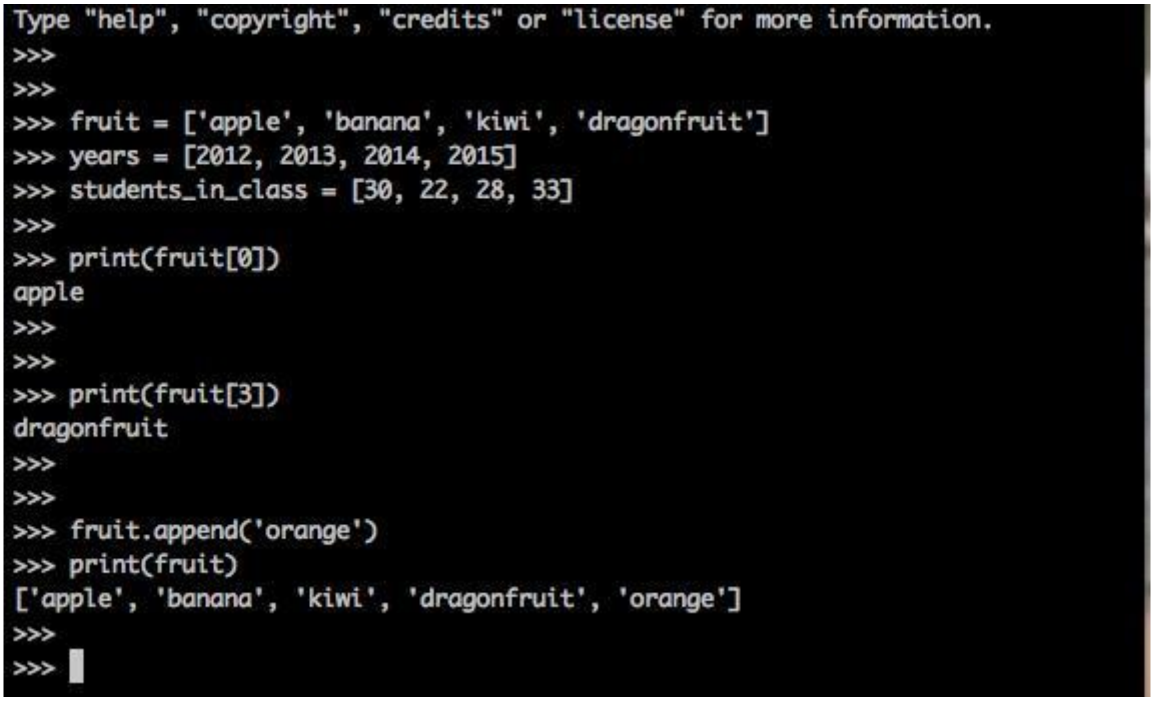
```
Last login: Tue Dec  8 21:40:12 on ttys000
Jessicas-MacBook-Air-2:~ jessicanickel$ python3.5
Python 3.5.0 (v3.5.0:374f501f4567, Sep 12 2015, 11:00:19)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>>
>>> fruit = ['apple', 'banana', 'kiwi', 'dragonfruit']
>>> years = [2012, 2013, 2014, 2015]
>>> students_in_class = [30, 22, 28, 33]
>>>
>>> print(fruit[0])
apple
>>>
>>>
>>> print(fruit[3])
dragonfruit
>>>
>>> 
```



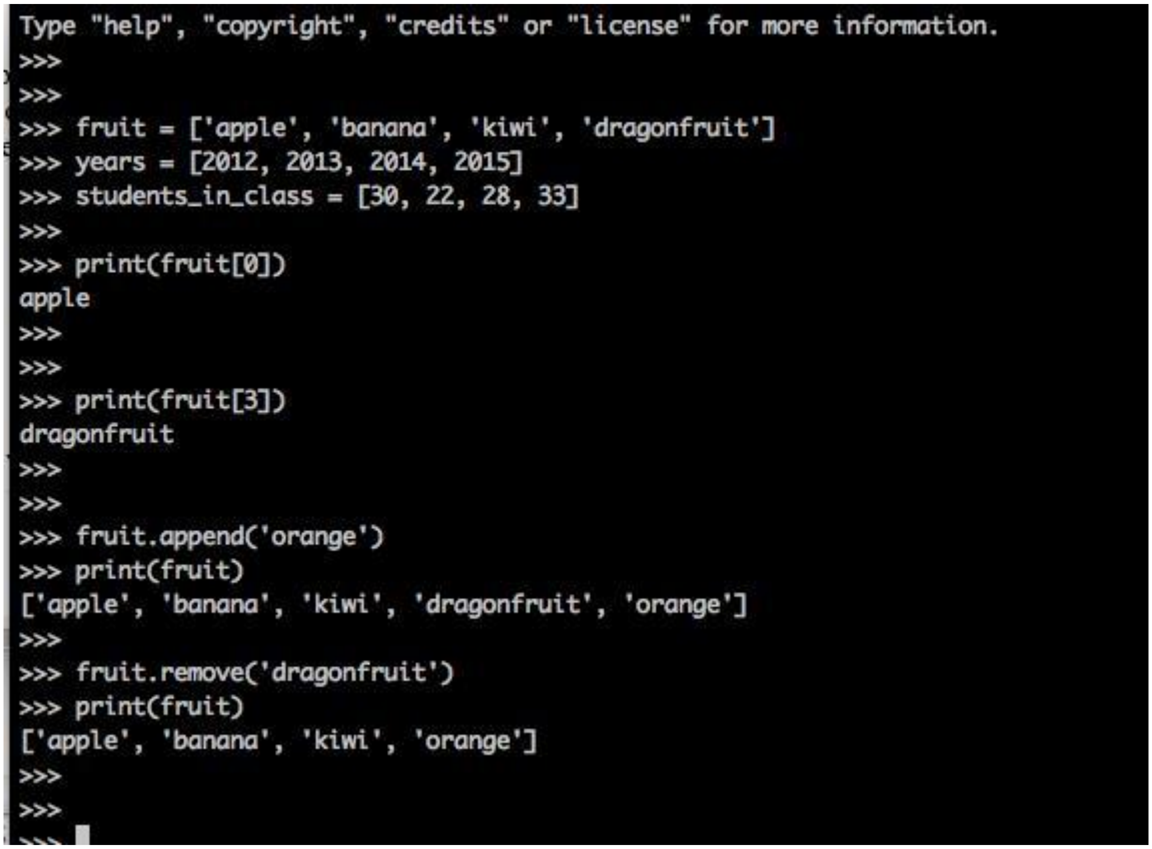
print = (list_name[item_number])

name of list you made

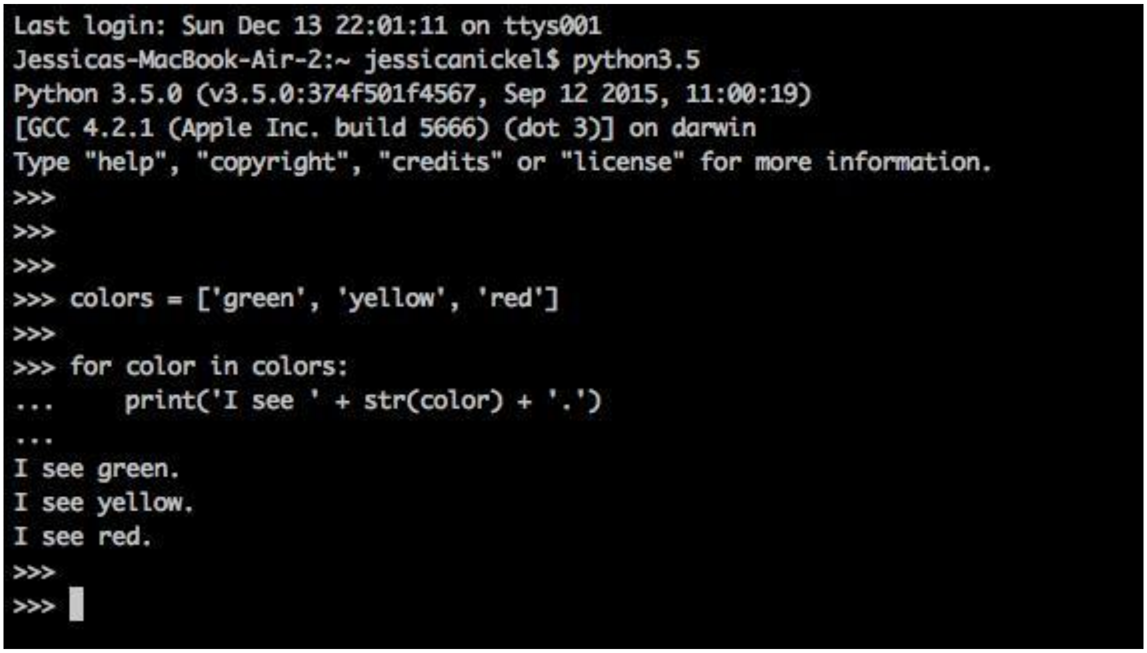
what item you want from the list



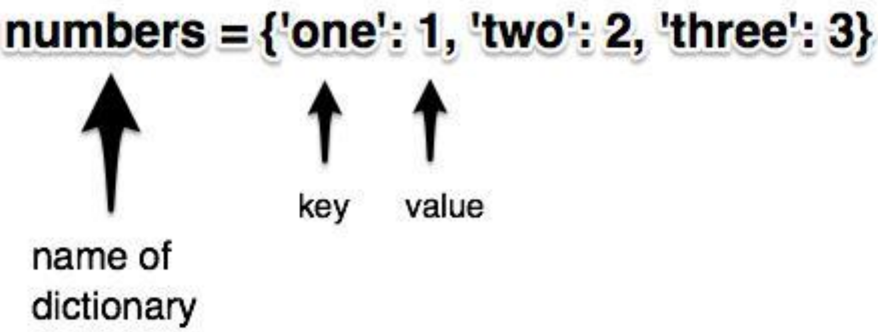
```
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>>
>>> fruit = ['apple', 'banana', 'kiwi', 'dragonfruit']
>>> years = [2012, 2013, 2014, 2015]
>>> students_in_class = [30, 22, 28, 33]
>>>
>>> print(fruit[0])
apple
>>>
>>>
>>> print(fruit[3])
dragonfruit
>>>
>>>
>>> fruit.append('orange')
>>> print(fruit)
['apple', 'banana', 'kiwi', 'dragonfruit', 'orange']
>>>
>>> 
```



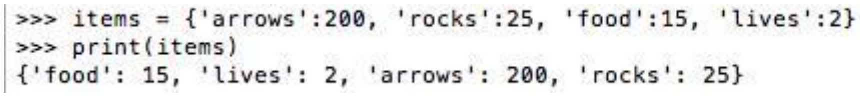
```
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>>
>>> fruit = ['apple', 'banana', 'kiwi', 'dragonfruit']
>>> years = [2012, 2013, 2014, 2015]
>>> students_in_class = [30, 22, 28, 33]
>>>
>>> print(fruit[0])
apple
>>>
>>>
>>> print(fruit[3])
dragonfruit
>>>
>>>
>>> fruit.append('orange')
>>> print(fruit)
['apple', 'banana', 'kiwi', 'dragonfruit', 'orange']
>>>
>>> fruit.remove('dragonfruit')
>>> print(fruit)
['apple', 'banana', 'kiwi', 'orange']
>>>
>>>
>>> 
```



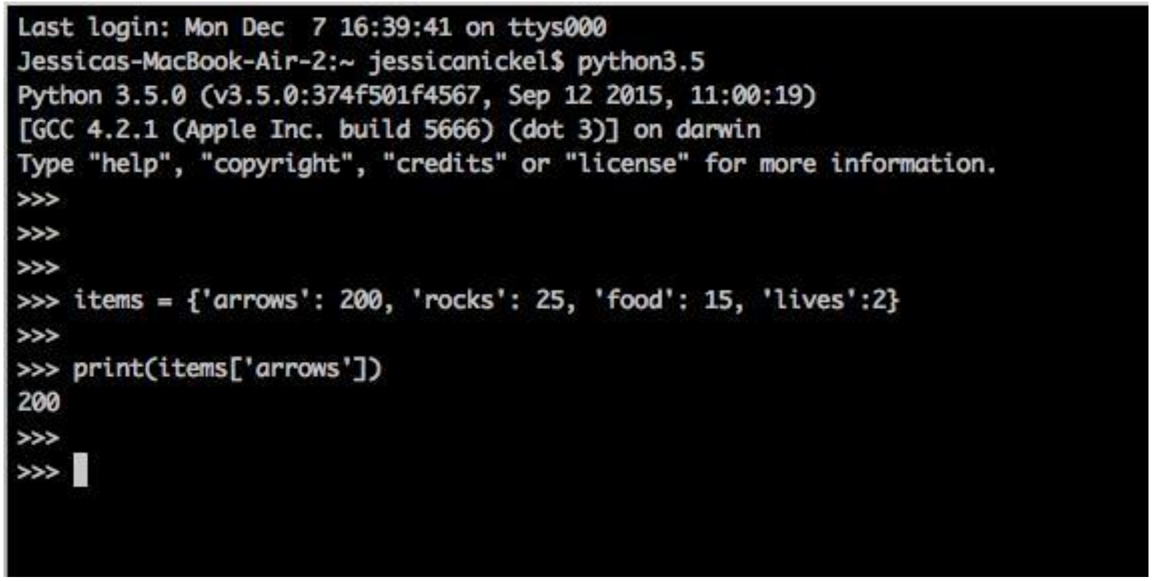
```
Last login: Sun Dec 13 22:01:11 on ttys001
Jessicas-MacBook-Air-2:~ jessicanickel$ python3.5
Python 3.5.0 (v3.5.0:374f501f4567, Sep 12 2015, 11:00:19)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>>
>>>
>>> colors = ['green', 'yellow', 'red']
>>>
>>> for color in colors:
...     print('I see ' + str(color) + '.')
...
I see green.
I see yellow.
I see red.
>>>
>>> 
```



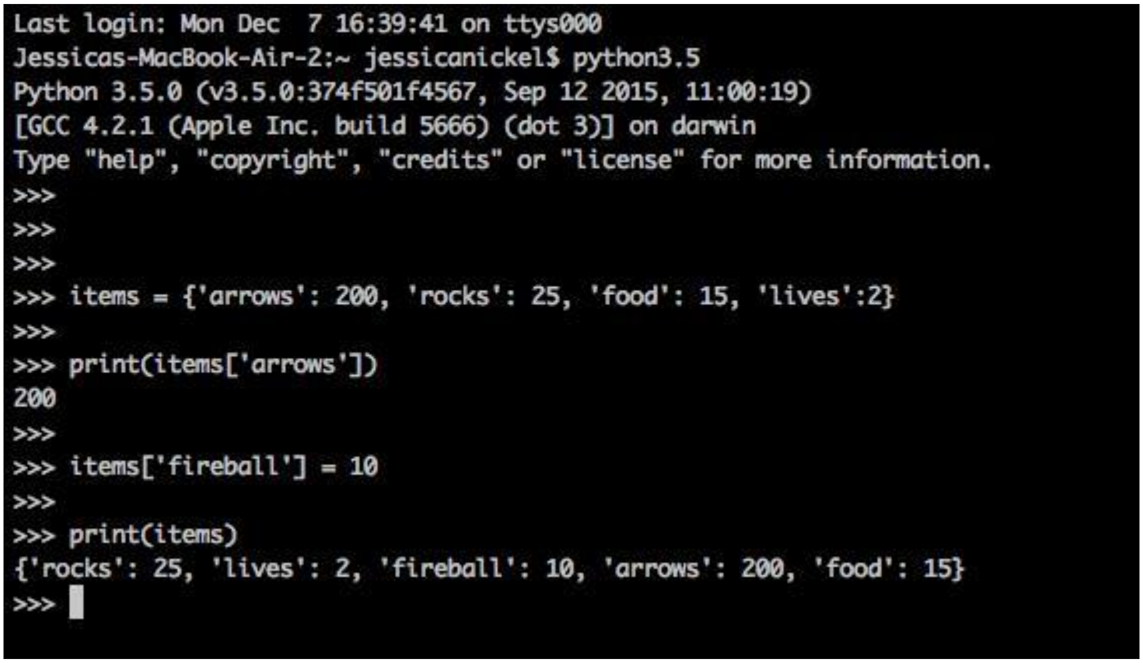
```
numbers = {'one': 1, 'two': 2, 'three': 3}
```



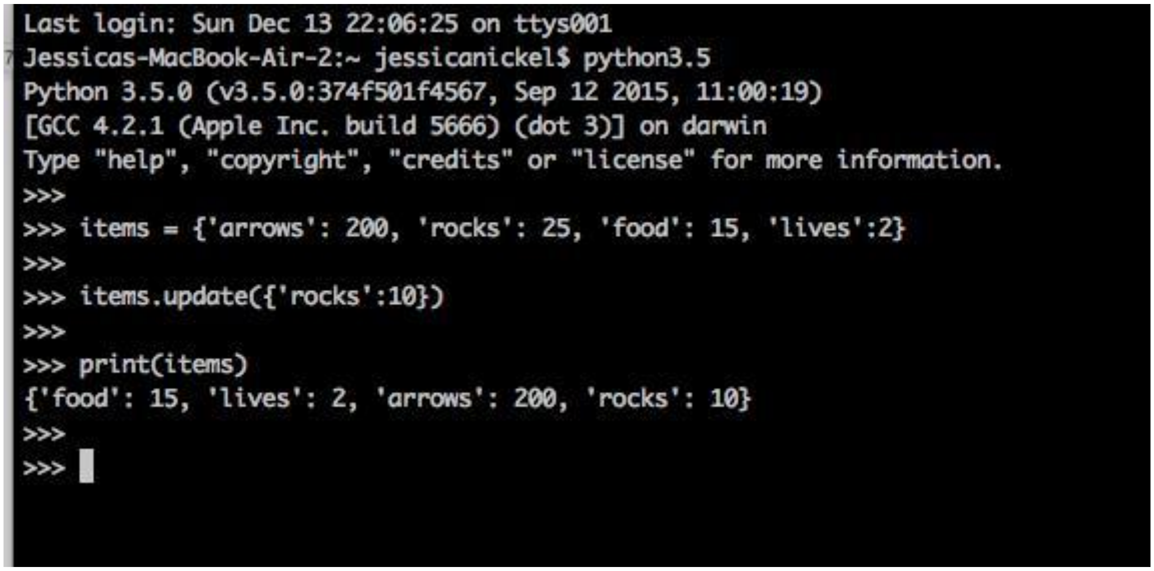
```
>>> items = {'arrows':200, 'rocks':25, 'food':15, 'lives':2}
>>> print(items)
{'food': 15, 'lives': 2, 'arrows': 200, 'rocks': 25}
```



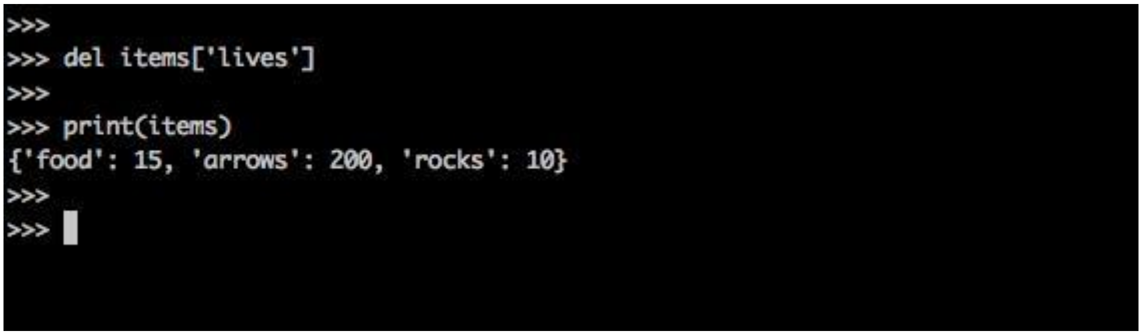
```
Last login: Mon Dec  7 16:39:41 on ttys000
Jessicas-MacBook-Air-2:~ jessicanickel$ python3.5
Python 3.5.0 (v3.5.0:374f501f4567, Sep 12 2015, 11:00:19)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>>
>>>
>>> items = {'arrows': 200, 'rocks': 25, 'food': 15, 'lives':2}
>>>
>>> print(items['arrows'])
200
>>>
>>> 
```



```
Last login: Mon Dec  7 16:39:41 on ttys000
Jessicas-MacBook-Air-2:~ jessicanickel$ python3.5
Python 3.5.0 (v3.5.0:374f501f4567, Sep 12 2015, 11:00:19)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>>
>>>
>>> items = {'arrows': 200, 'rocks': 25, 'food': 15, 'lives':2}
>>>
>>> print(items['arrows'])
200
>>>
>>> items['fireball'] = 10
>>>
>>> print(items)
{'rocks': 25, 'lives': 2, 'fireball': 10, 'arrows': 200, 'food': 15}
>>> 
```

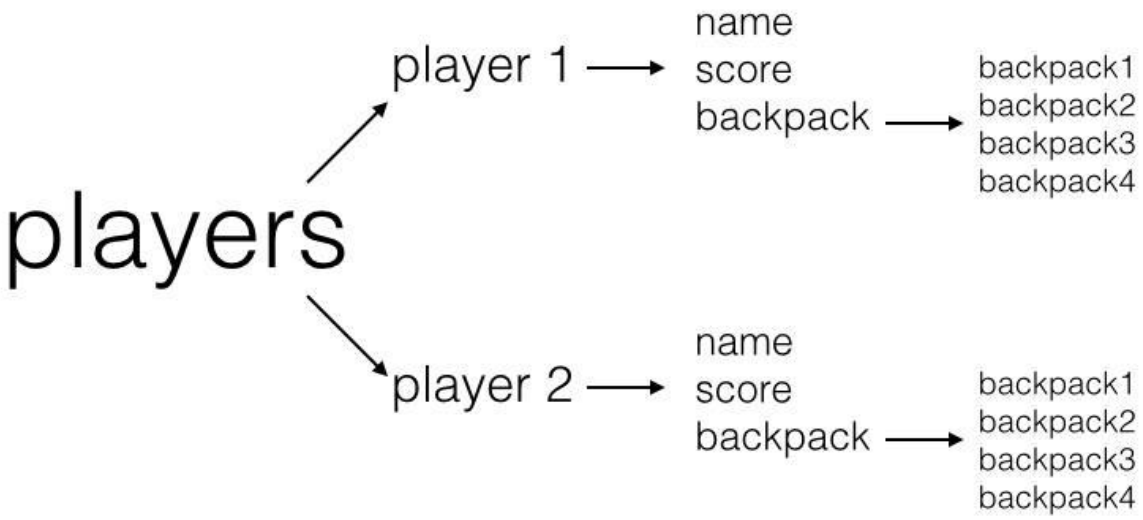


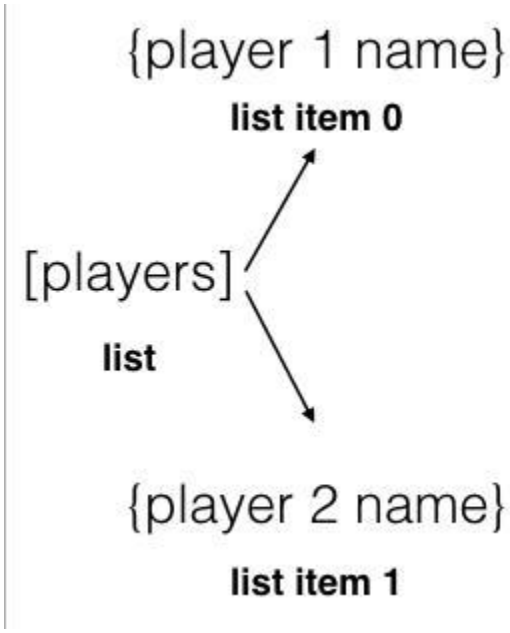
```
Last login: Sun Dec 13 22:06:25 on ttys001
Jessicas-MacBook-Air-2:~ jessicanickel$ python3.5
Python 3.5.0 (v3.5.0:374f501f4567, Sep 12 2015, 11:00:19)
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>> items = {'arrows': 200, 'rocks': 25, 'food': 15, 'lives':2}
>>>
>>> items.update({'rocks':10})
>>>
>>> print(items)
{'food': 15, 'lives': 2, 'arrows': 200, 'rocks': 10}
>>>
>>> 
```

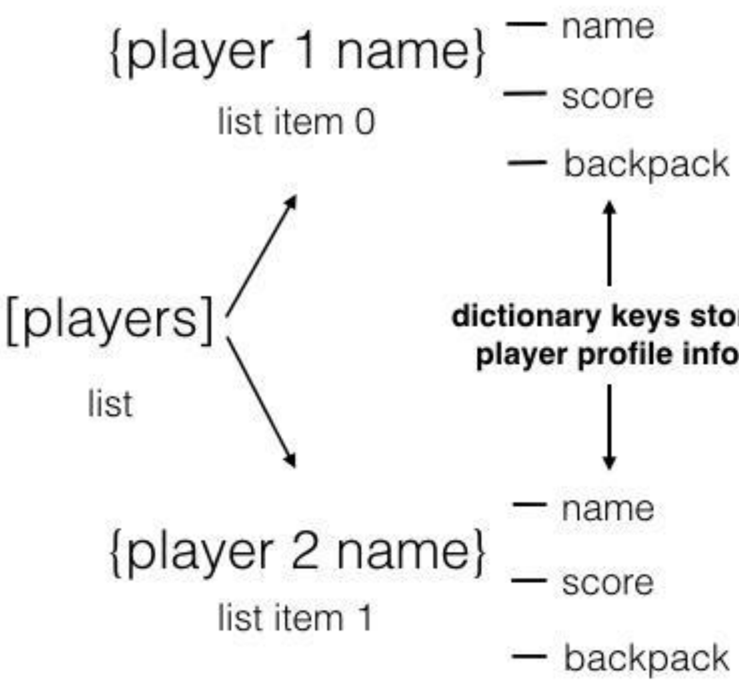


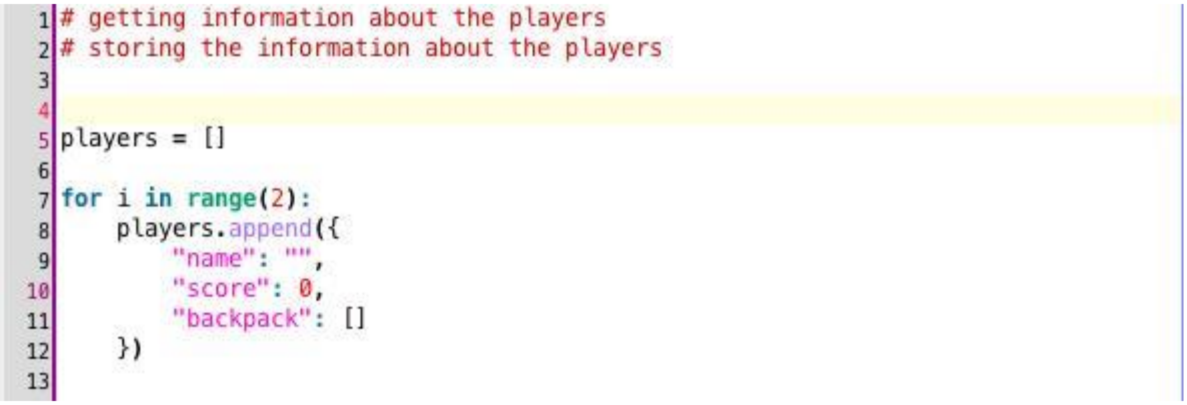
```
>>>
>>> del items['lives']
>>>
>>> print(items)
{'food': 15, 'arrows': 200, 'rocks': 10}
>>>
>>> 
```

Chapter 7: What's in Your Backpack?

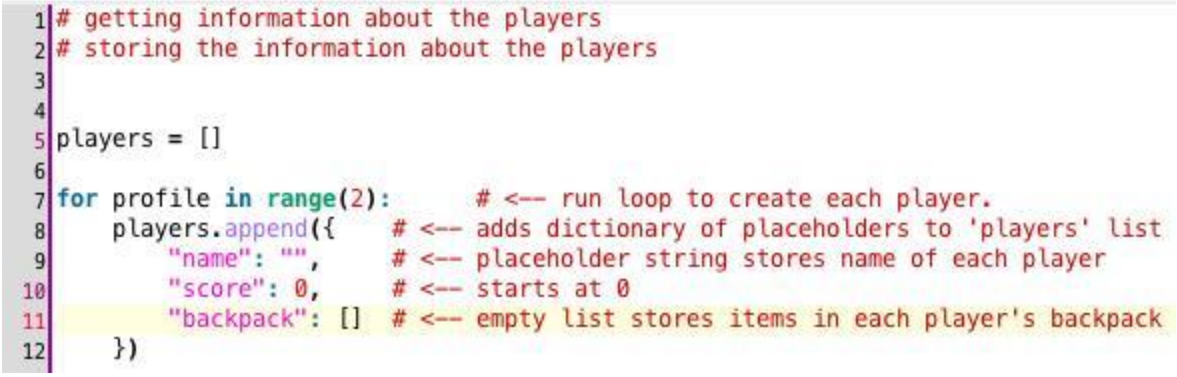




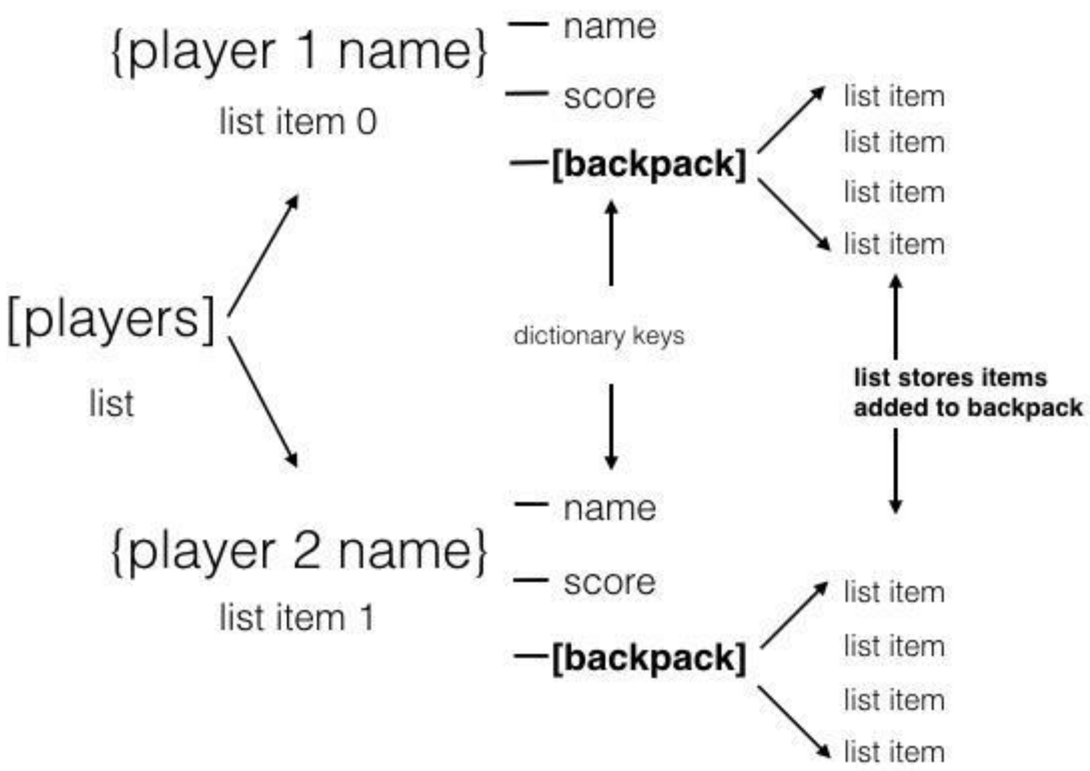


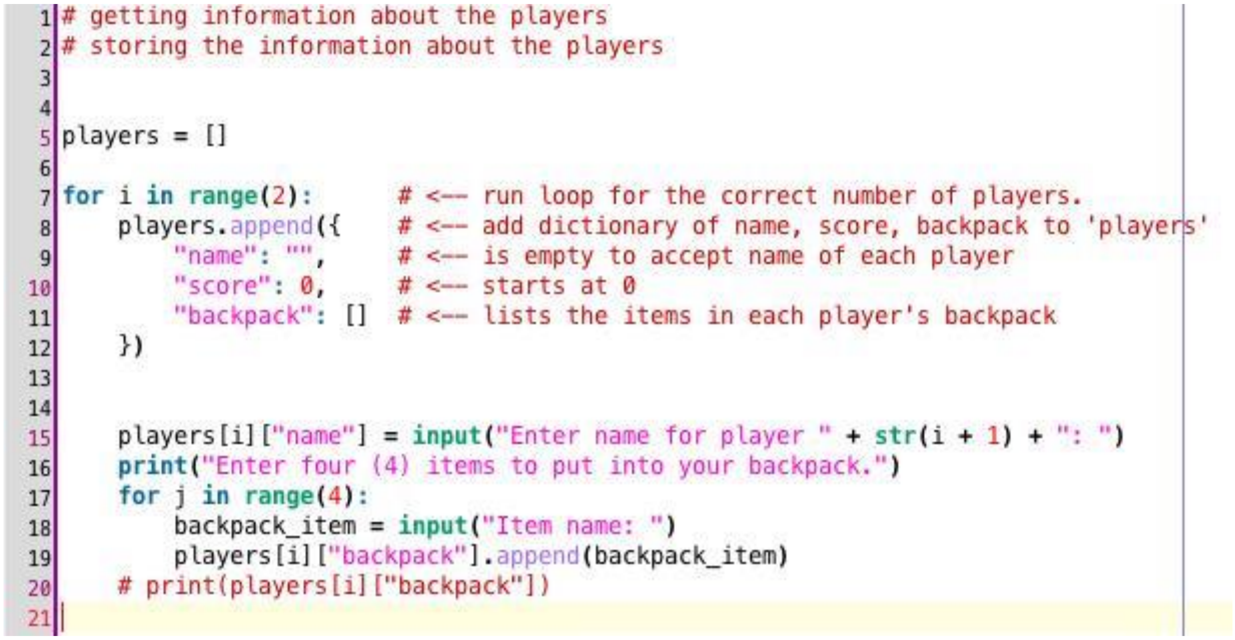


```python
# getting information about the players
# storing the information about the players


players = []

for i in range(2):
    players.append({
        "name": "",
        "score": 0,
        "backpack": []
    })

```

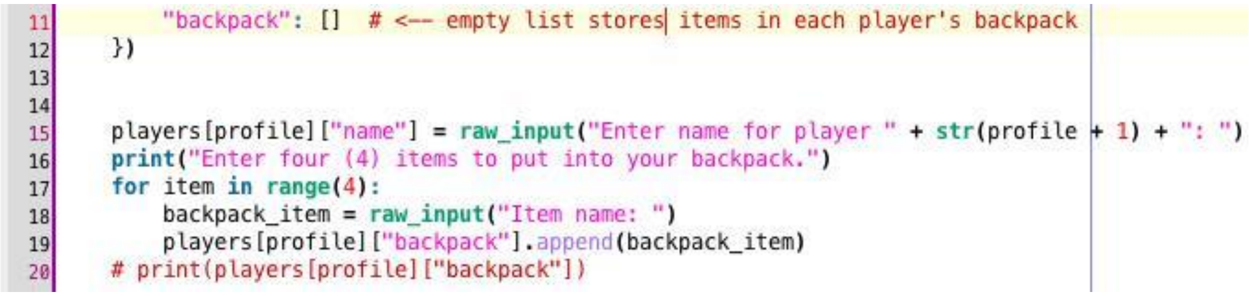


```python
# getting information about the players
# storing the information about the players


players = []

for profile in range(2):        # <-- run loop to create each player.
    players.append({    # <-- adds dictionary of placeholders to 'players' list
        "name": "",     # <-- placeholder string stores name of each player
        "score": 0,     # <-- starts at 0
        "backpack": []  # <-- empty list stores items in each player's backpack
    })
```

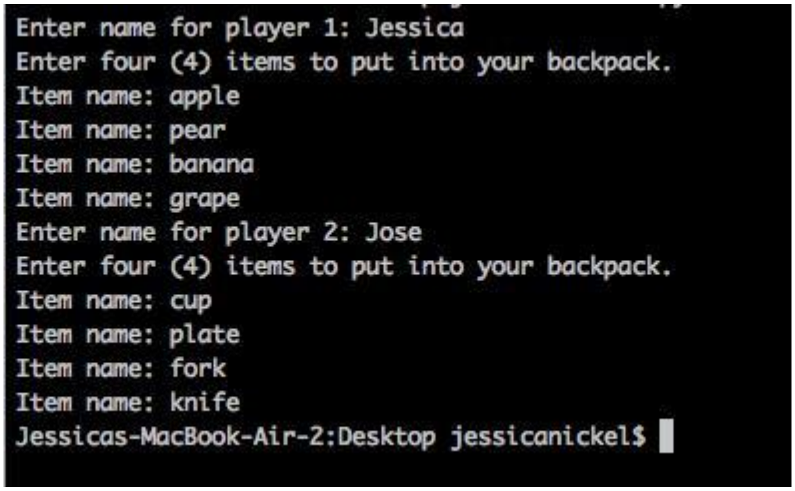




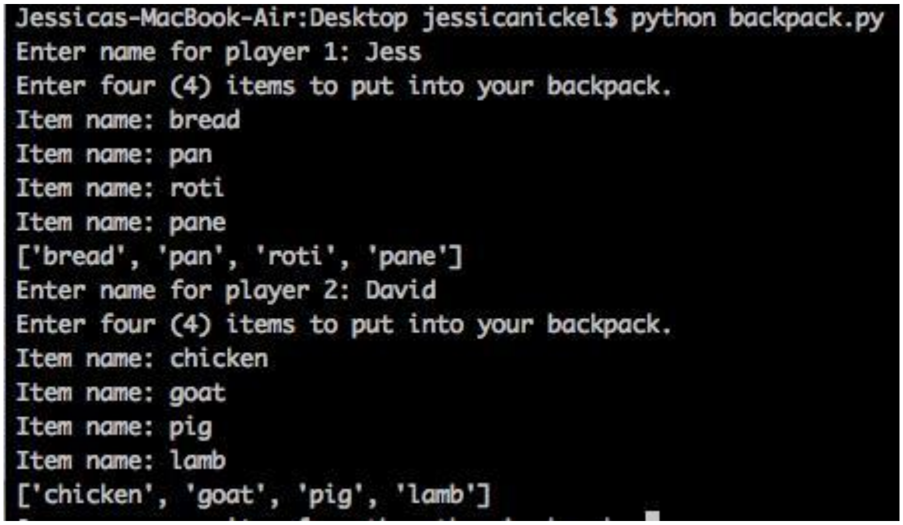
```python
# getting information about the players
# storing the information about the players


players = []

for i in range(2):      # <-- run loop for the correct number of players.
    players.append({    # <-- add dictionary of name, score, backpack to 'players'
        "name": "",     # <-- is empty to accept name of each player
        "score": 0,     # <-- starts at 0
        "backpack": []  # <-- lists the items in each player's backpack
    })


    players[i]["name"] = input("Enter name for player " + str(i + 1) + ": ")
    print("Enter four (4) items to put into your backpack.")
    for j in range(4):
        backpack_item = input("Item name: ")
        players[i]["backpack"].append(backpack_item)
    # print(players[i]["backpack"])

```



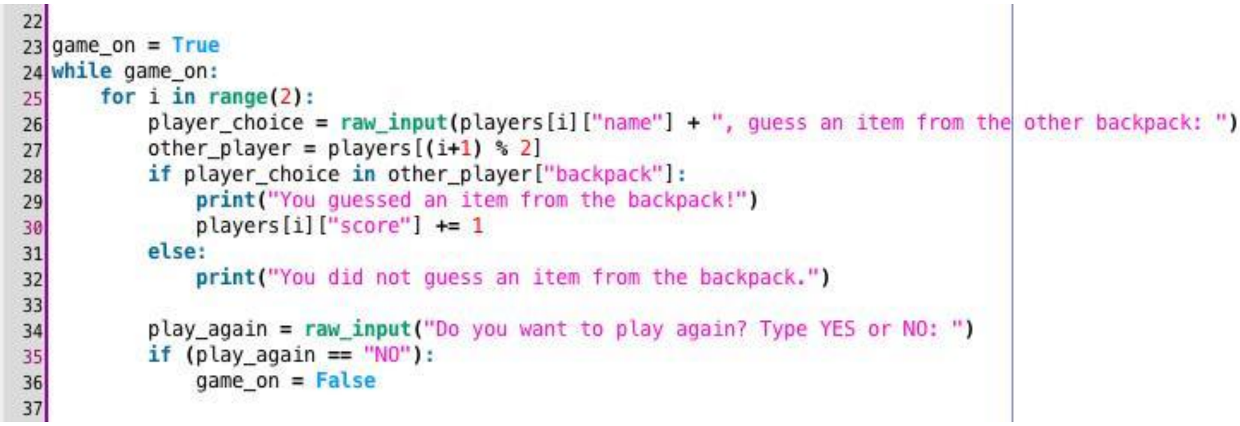
```
        "backpack": []  # <-- empty list stores items in each player's backpack
    })


    players[profile]["name"] = raw_input("Enter name for player " + str(profile + 1) + ": ")
    print("Enter four (4) items to put into your backpack.")
    for item in range(4):
        backpack_item = raw_input("Item name: ")
        players[profile]["backpack"].append(backpack_item)
    # print(players[profile]["backpack"])
```



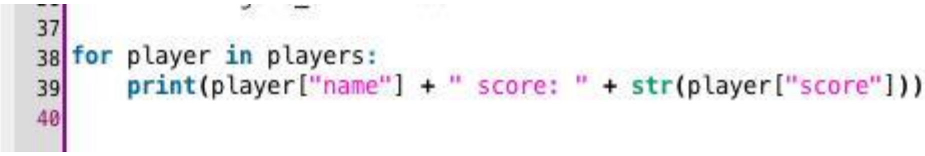
```
Enter name for player 1: Jessica
Enter four (4) items to put into your backpack.
Item name: apple
Item name: pear
Item name: banana
Item name: grape
Enter name for player 2: Jose
Enter four (4) items to put into your backpack.
Item name: cup
Item name: plate
Item name: fork
Item name: knife
Jessicas-MacBook-Air-2:Desktop jessicanickel$
```



```
Jessicas-MacBook-Air:Desktop jessicanickel$ python backpack.py
Enter name for player 1: Jess
Enter four (4) items to put into your backpack.
Item name: bread
Item name: pan
Item name: roti
Item name: pane
['bread', 'pan', 'roti', 'pane']
Enter name for player 2: David
Enter four (4) items to put into your backpack.
Item name: chicken
Item name: goat
Item name: pig
Item name: lamb
['chicken', 'goat', 'pig', 'lamb']
```

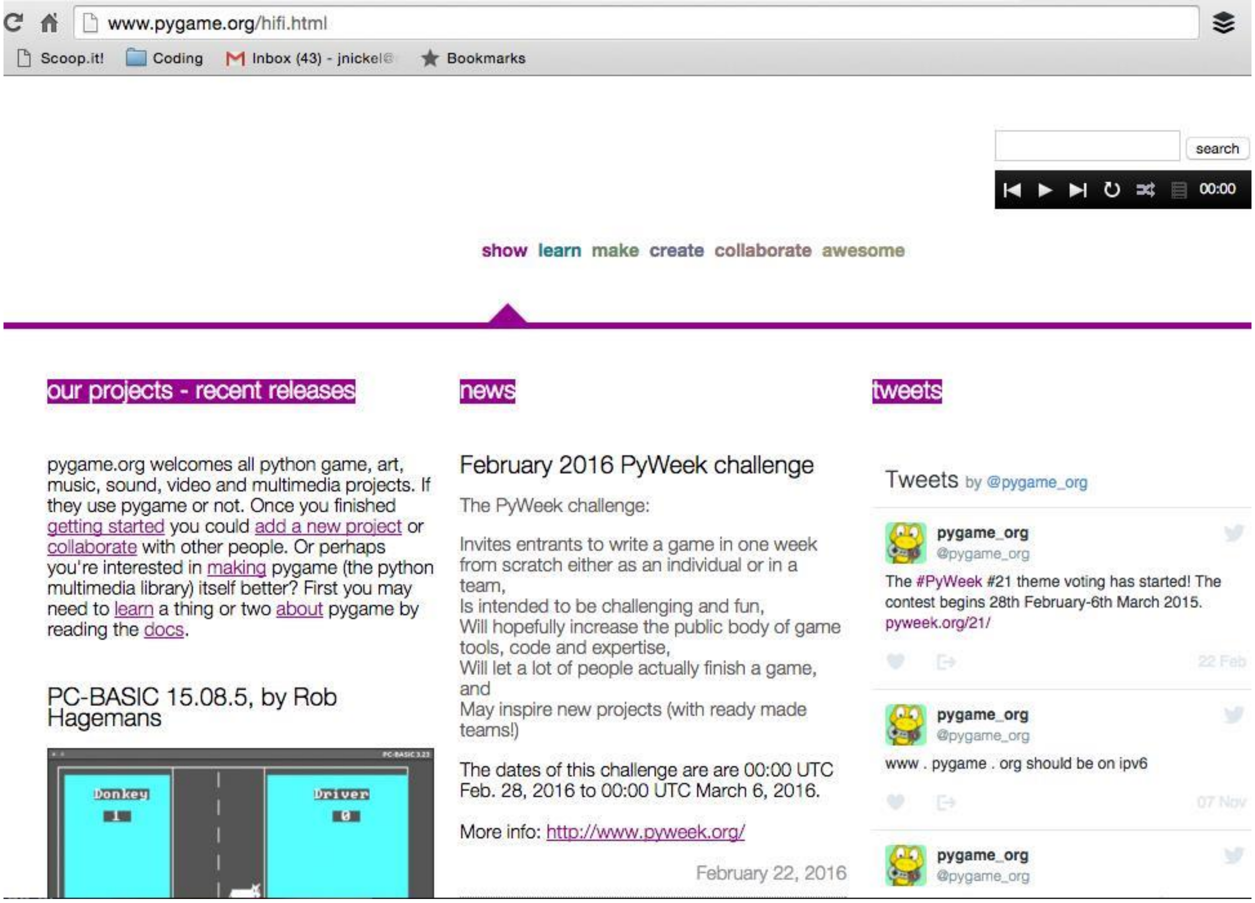


```
game_on = True
while game_on:
    for i in range(2):
        player_choice = raw_input(players[i]["name"] + ", guess an item from the other backpack: ")
        other_player = players[(i+1) % 2]
        if player_choice in other_player["backpack"]:
            print("You guessed an item from the backpack!")
            players[i]["score"] += 1
        else:
            print("You did not guess an item from the backpack.")

        play_again = raw_input("Do you want to play again? Type YES or NO: ")
        if (play_again == "NO"):
            game_on = False

```



```

for player in players:
    print(player["name"] + " score: " + str(player["score"]))

```

Chapter 8: pygame



our projects - recent releases

pygame.org welcomes all python game, art, music, sound, video and multimedia projects. If they use pygame or not. Once you finished getting started you could add a new project or collaborate with other people. Or perhaps you're interested in making pygame (the python multimedia library) itself better? First you may need to learn a thing or two about pygame by reading the docs.

PC-BASIC 15.08.5, by Rob Hagemans

news

February 2016 PyWeek challenge

The PyWeek challenge:

Invites entrants to write a game in one week from scratch either as an individual or in a team,
Is intended to be challenging and fun,
Will hopefully increase the public body of game tools, code and expertise,
Will let a lot of people actually finish a game, and
May inspire new projects (with ready made teams!)

The dates of this challenge are are 00:00 UTC Feb. 28, 2016 to 00:00 UTC March 6, 2016.

More info: http://www.pyweek.org/

February 22, 2016

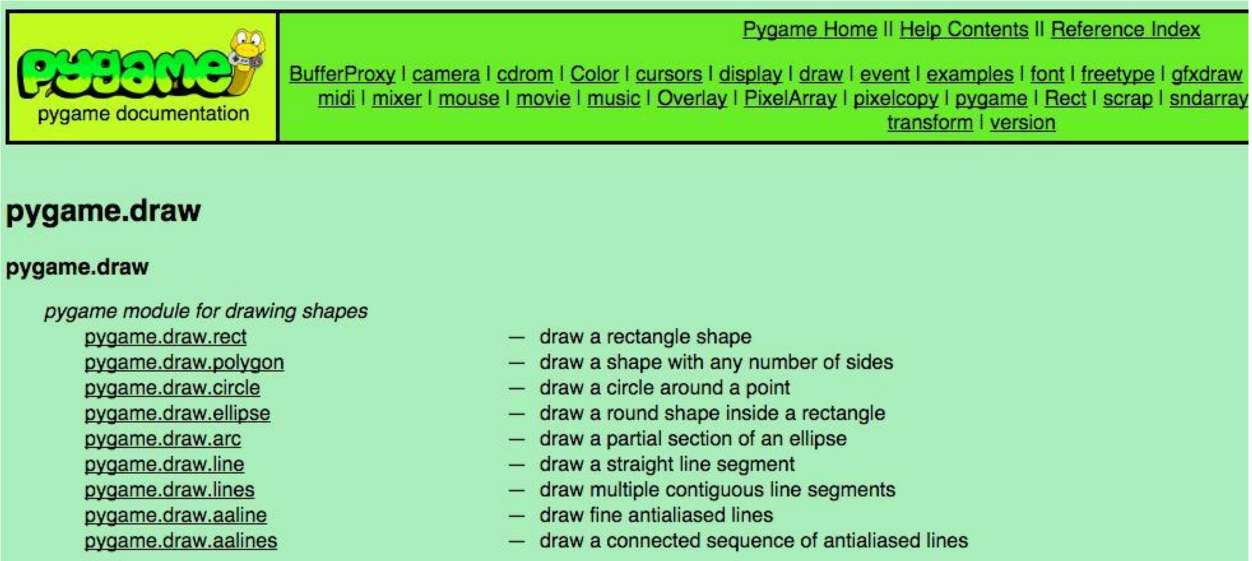
tweets

Tweets by @pygame_org

pygame_org
@pygame_org
The #PyWeek #21 theme voting has started! The contest begins 28th February-6th March 2015. pyweek.org/21/
22 Feb

pygame_org
@pygame_org
www . pygame . org should be on ipv6
07 Nov

pygame_org
@pygame_org

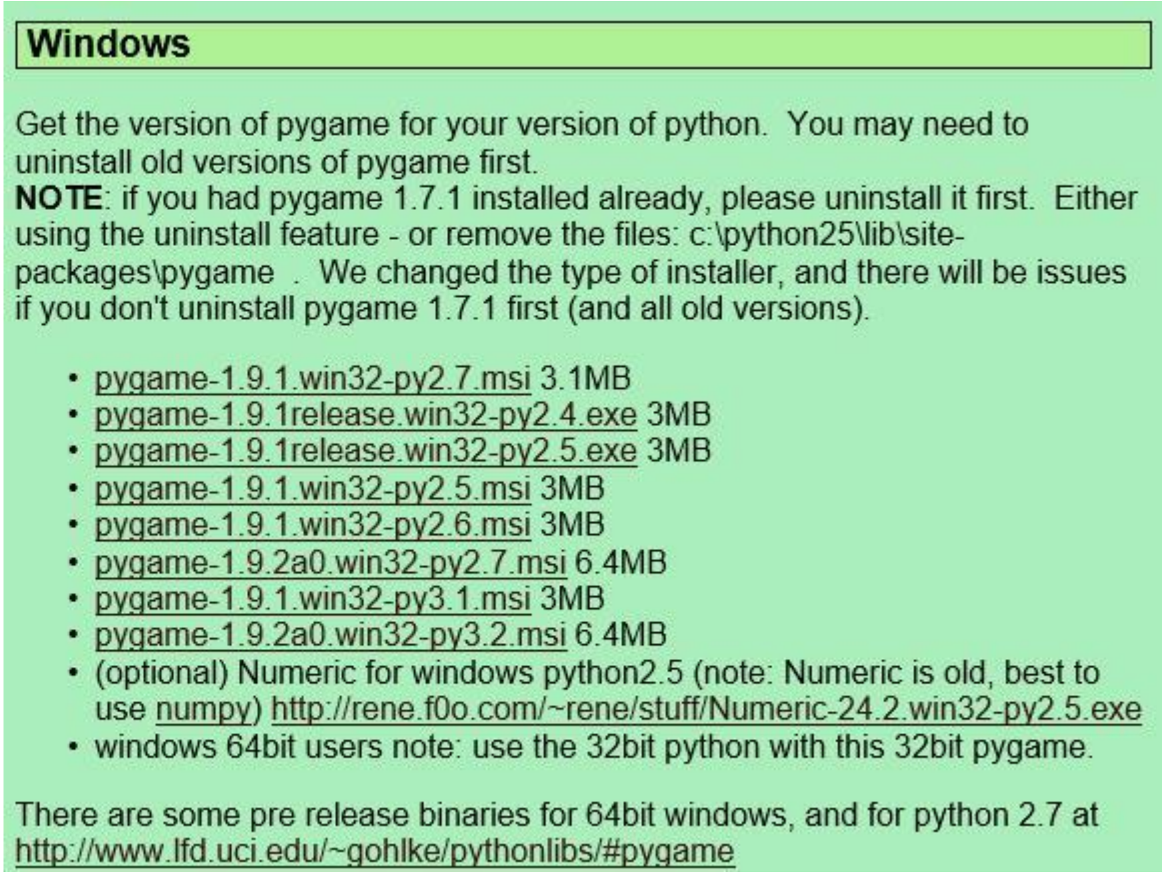

pygame documentation
Pygame Home || Help Contents || Reference Index
BufferProxy | camera | cdrom | Color | cursors | display | draw | event | examples | font | freetype | gfxdraw
midi | mixer | mouse | movie | music | Overlay | PixelArray | pixelcopy | pygame | Rect | scrap | sndarray
transform | version

pygame.draw

pygame.draw

pygame module for drawing shapes

pygame.draw.rect	— draw a rectangle shape
pygame.draw.polygon	— draw a shape with any number of sides
pygame.draw.circle	— draw a circle around a point
pygame.draw.ellipse	— draw a round shape inside a rectangle
pygame.draw.arc	— draw a partial section of an ellipse
pygame.draw.line	— draw a straight line segment
pygame.draw.lines	— draw multiple contiguous line segments
pygame.draw.aaline	— draw fine antialiased lines
pygame.draw.aalines	— draw a connected sequence of antialiased lines

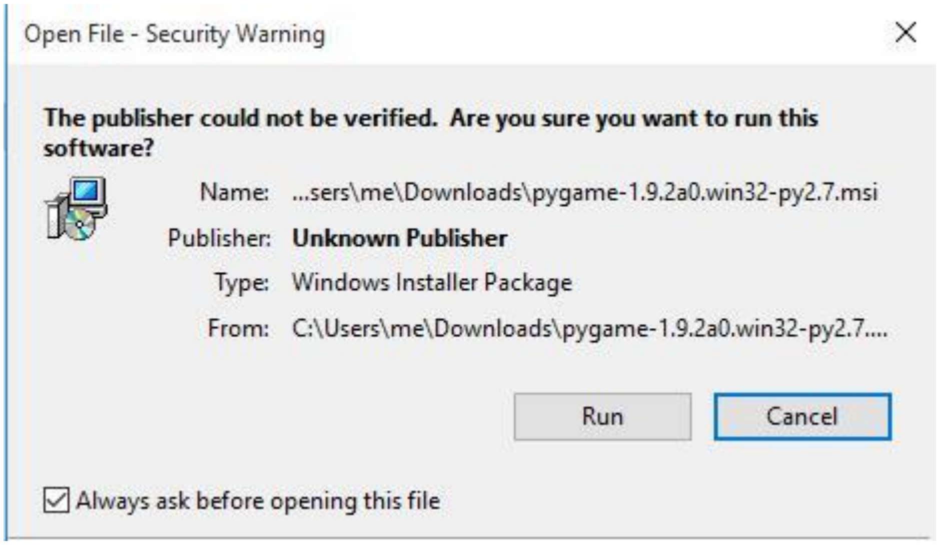


Windows

Get the version of pygame for your version of python. You may need to uninstall old versions of pygame first.
NOTE: if you had pygame 1.7.1 installed already, please uninstall it first. Either using the uninstall feature - or remove the files: c:\python25\lib\site-packages\pygame . We changed the type of installer, and there will be issues if you don't uninstall pygame 1.7.1 first (and all old versions).

- pygame-1.9.1.win32-py2.7.msi 3.1MB
- pygame-1.9.1release.win32-py2.4.exe 3MB
- pygame-1.9.1release.win32-py2.5.exe 3MB
- pygame-1.9.1.win32-py2.5.msi 3MB
- pygame-1.9.1.win32-py2.6.msi 3MB
- pygame-1.9.2a0.win32-py2.7.msi 6.4MB
- pygame-1.9.1.win32-py3.1.msi 3MB
- pygame-1.9.2a0.win32-py3.2.msi 6.4MB
- (optional) Numeric for windows python2.5 (note: Numeric is old, best to use numpy) http://rene.f0o.com/~rene/stuff/Numeric-24.2.win32-py2.5.exe
- windows 64bit users note: use the 32bit python with this 32bit pygame.

There are some pre release binaries for 64bit windows, and for python 2.7 at http://www.lfd.uci.edu/~gohlke/pythonlibs/#pygame



Open File - Security Warning

The publisher could not be verified. Are you sure you want to run this software?

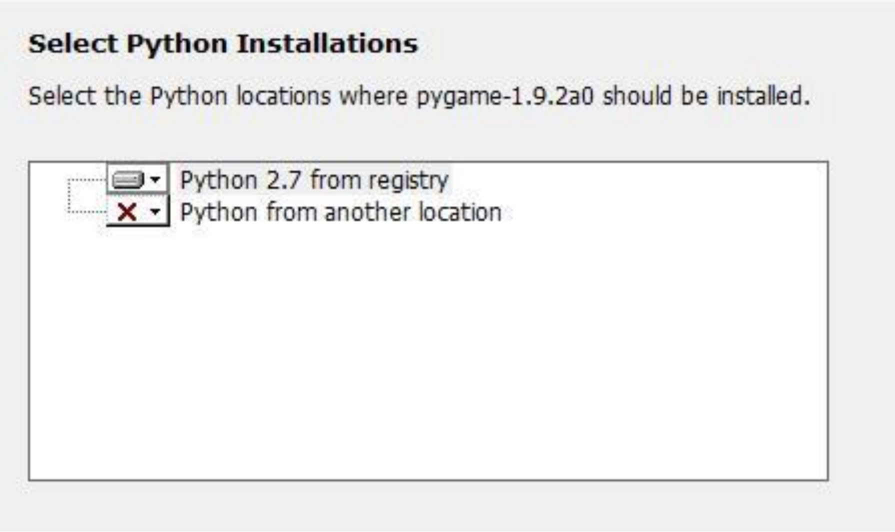
Name: ...sers\me\Downloads\pygame-1.9.2a0.win32-py2.7.msi

Publisher: **Unknown Publisher**

Type: Windows Installer Package

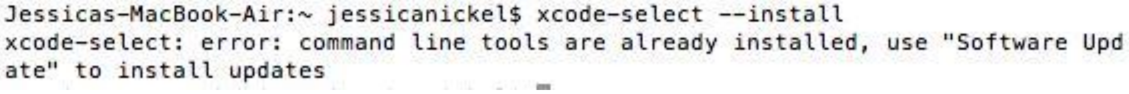
From: C:\Users\me\Downloads\pygame-1.9.2a0.win32-py2.7....

☑ Always ask before opening this file

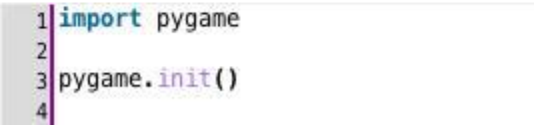


Select Python Installations

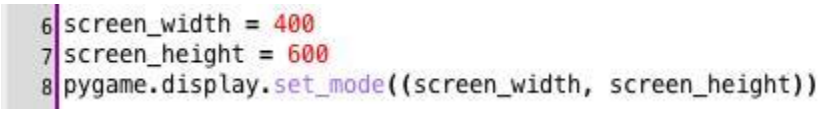
Select the Python locations where pygame-1.9.2a0 should be installed.



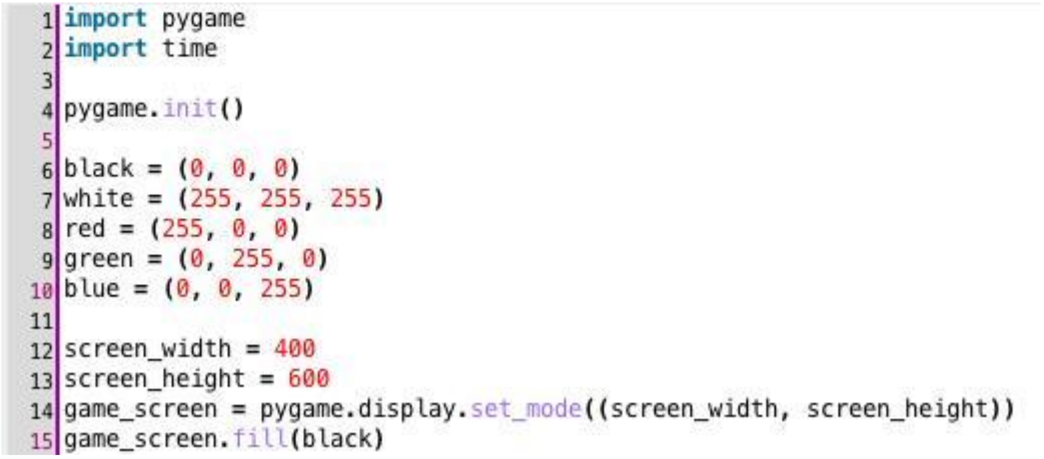
```
Jessicas-MacBook-Air:~ jessicanickel$ xcode-select --install
xcode-select: error: command line tools are already installed, use "Software Upd
ate" to install updates
```



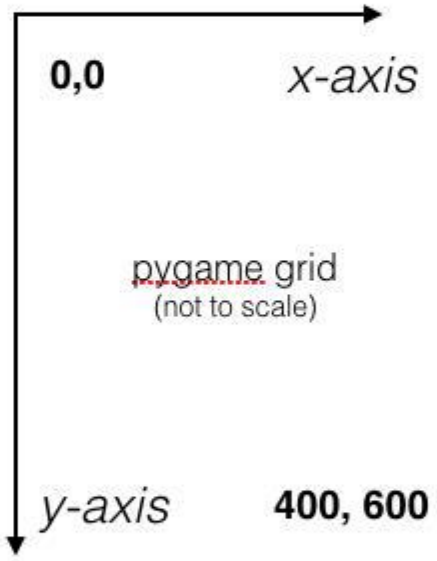
```python
import pygame

pygame.init()

```

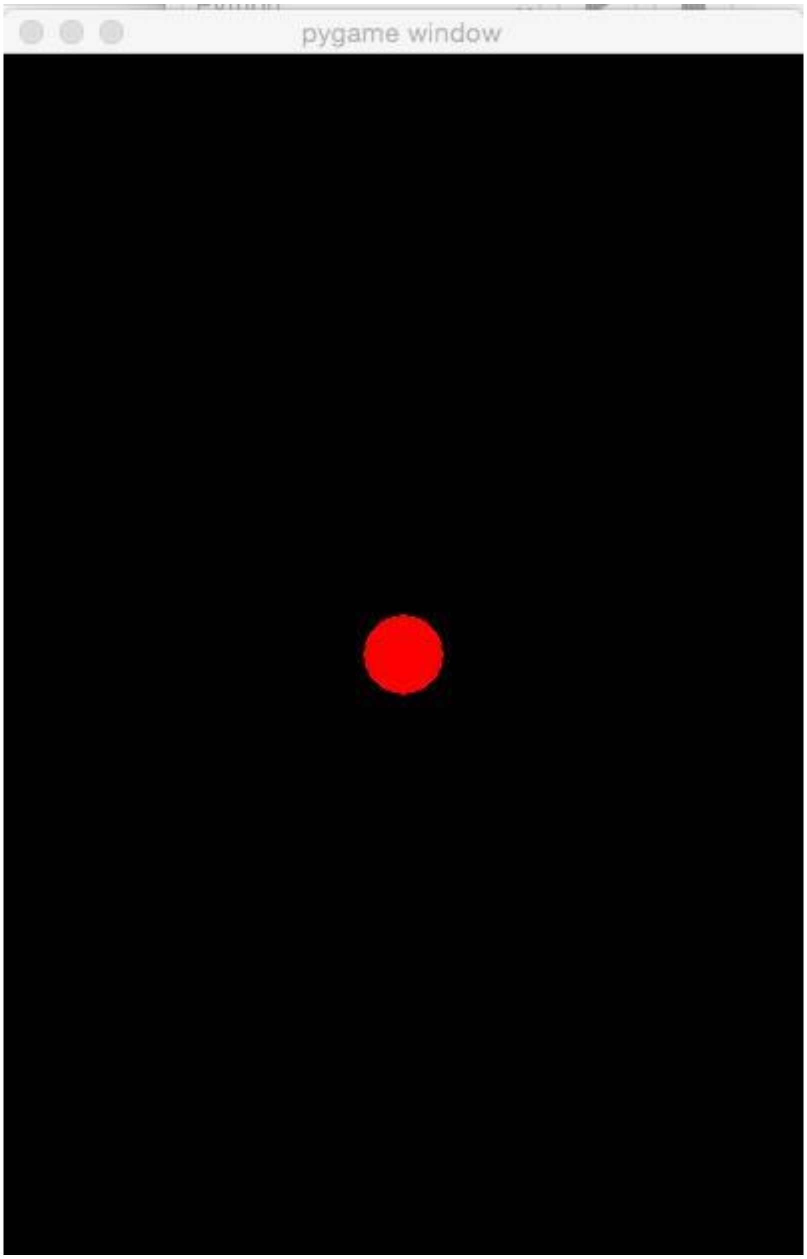


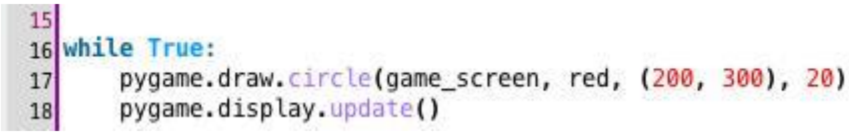
```python
screen_width = 400
screen_height = 600
pygame.display.set_mode((screen_width, screen_height))
```



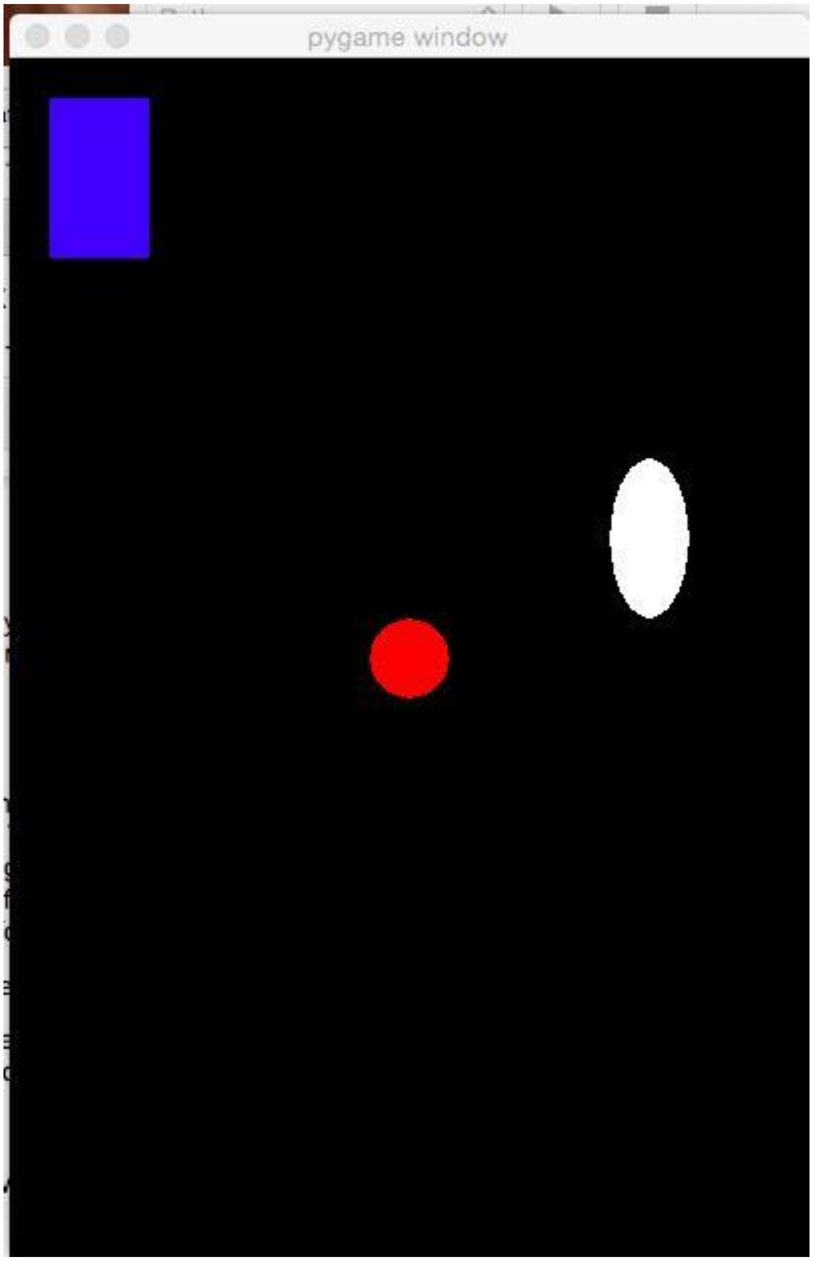
```python
import pygame
import time

pygame.init()

black = (0, 0, 0)
white = (255, 255, 255)
red = (255, 0, 0)
green = (0, 255, 0)
blue = (0, 0, 255)

screen_width = 400
screen_height = 600
game_screen = pygame.display.set_mode((screen_width, screen_height))
game_screen.fill(black)
```


0,0
x-axis
pygame grid
(not to scale)
y-axis
400, 600

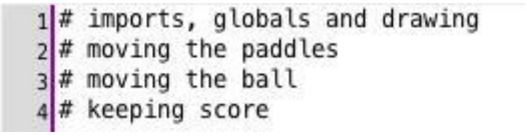




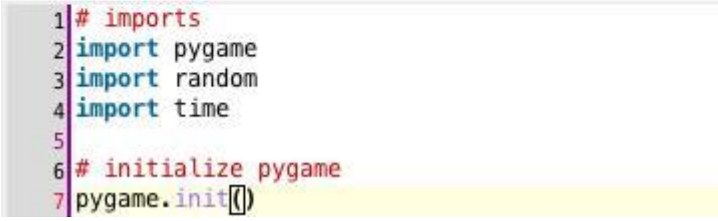
```

while True:
    pygame.draw.circle(game_screen, red, (200, 300), 20)
    pygame.display.update()
```


pygame window

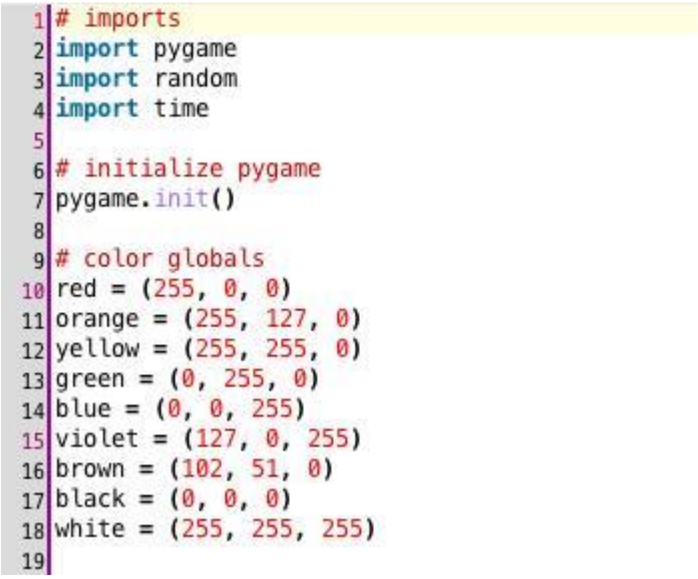
Chapter 9: Tiny Tennis



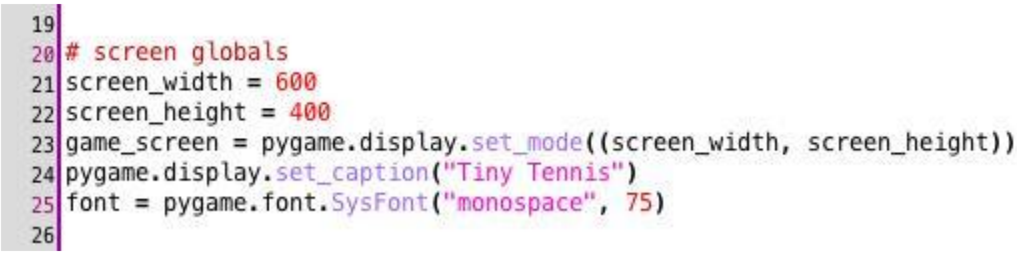
```
# imports, globals and drawing
# moving the paddles
# moving the ball
# keeping score
```



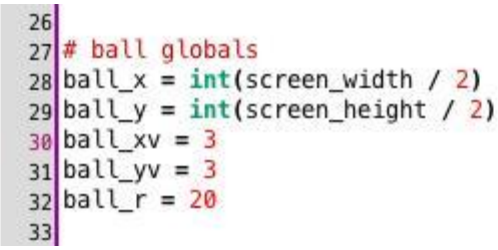
```
# imports
import pygame
import random
import time

# initialize pygame
pygame.init()
```



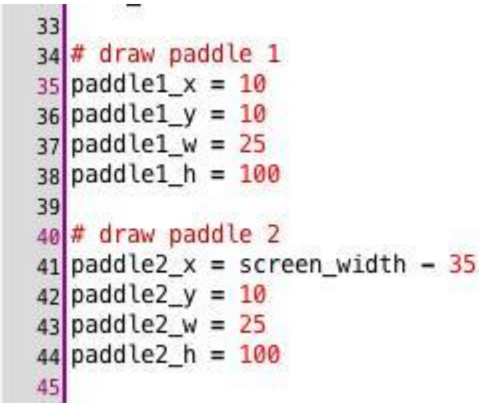
```
# imports
import pygame
import random
import time

# initialize pygame
pygame.init()

# color globals
red = (255, 0, 0)
orange = (255, 127, 0)
yellow = (255, 255, 0)
green = (0, 255, 0)
blue = (0, 0, 255)
violet = (127, 0, 255)
brown = (102, 51, 0)
black = (0, 0, 0)
white = (255, 255, 255)

```



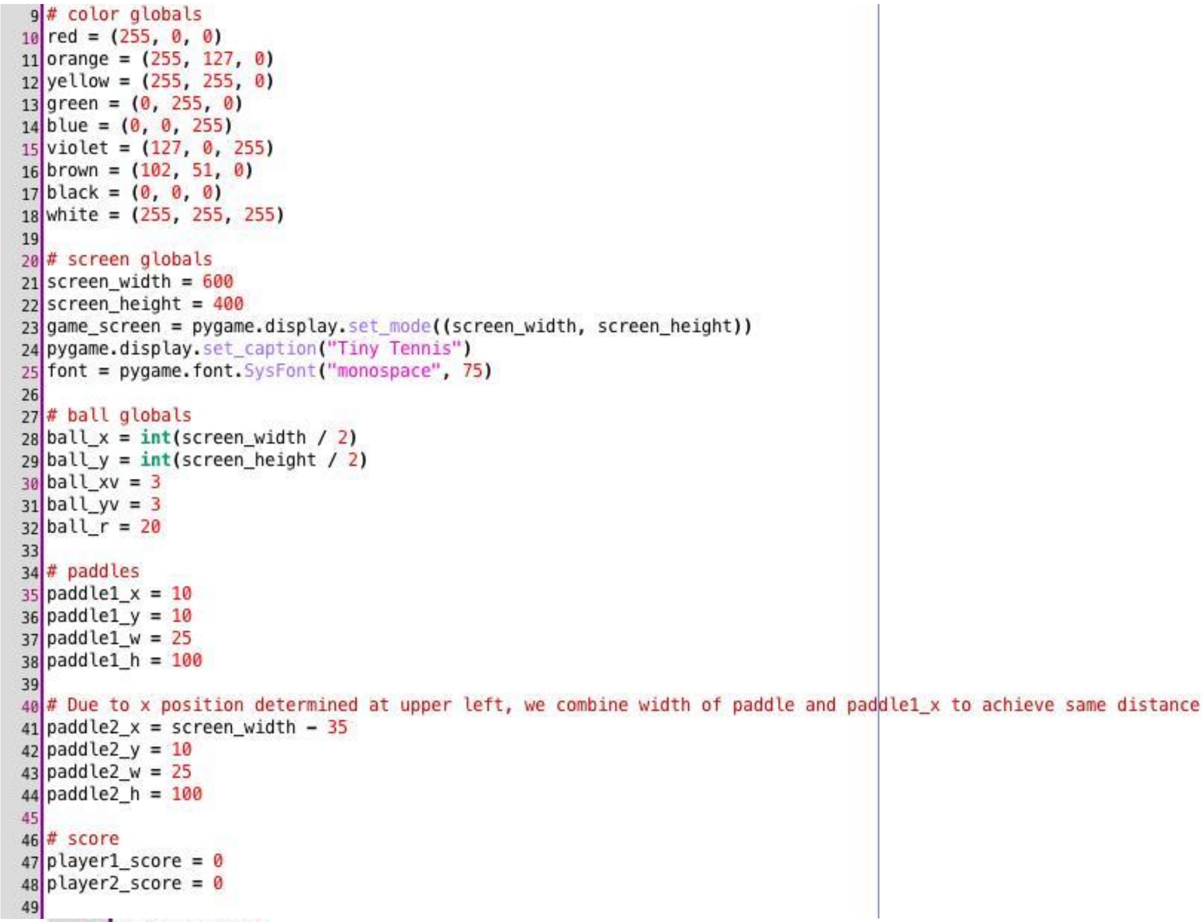
```

# screen globals
screen_width = 600
screen_height = 400
game_screen = pygame.display.set_mode((screen_width, screen_height))
pygame.display.set_caption("Tiny Tennis")
font = pygame.font.SysFont("monospace", 75)

```



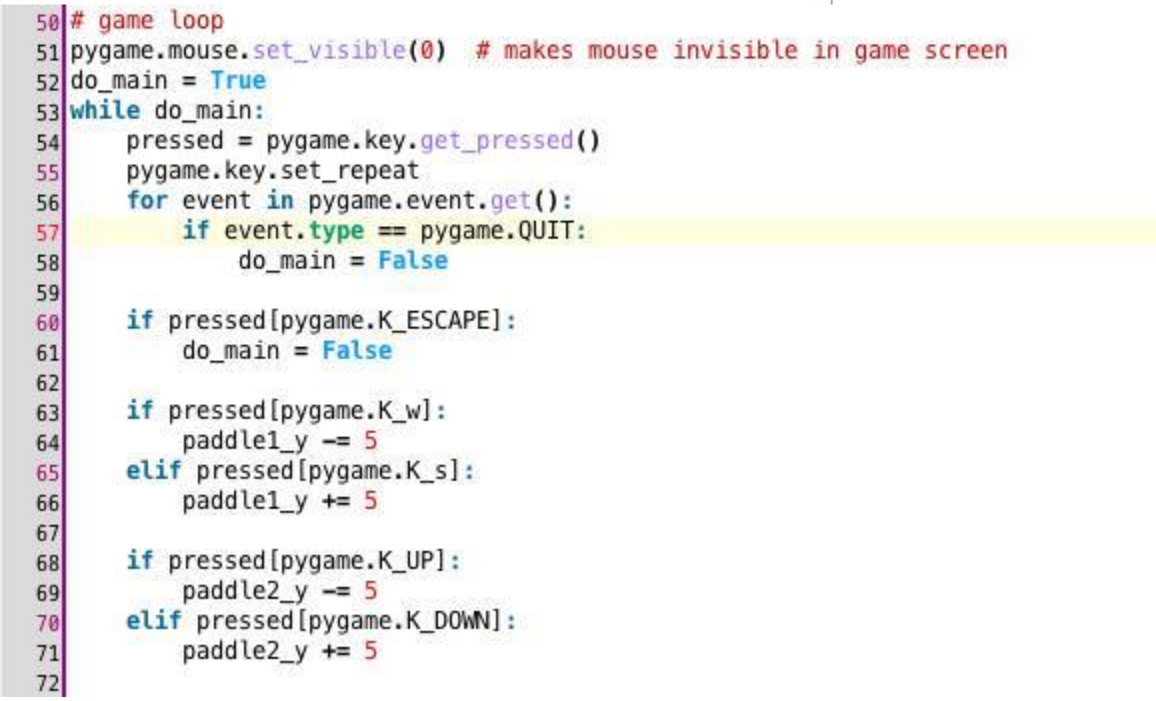
```

# ball globals
ball_x = int(screen_width / 2)
ball_y = int(screen_height / 2)
ball_xv = 3
ball_yv = 3
ball_r = 20

```



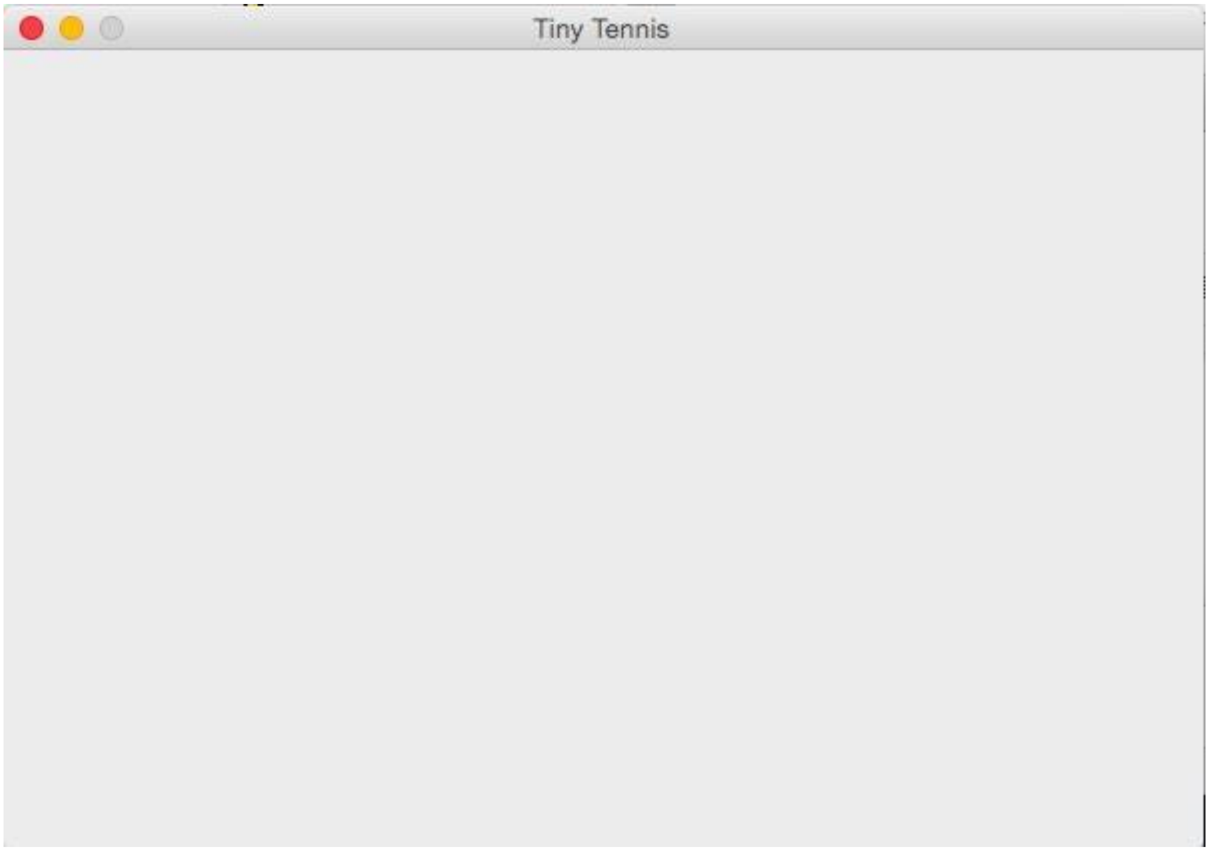
```
# draw paddle 1
paddle1_x = 10
paddle1_y = 10
paddle1_w = 25
paddle1_h = 100

# draw paddle 2
paddle2_x = screen_width - 35
paddle2_y = 10
paddle2_w = 25
paddle2_h = 100
```

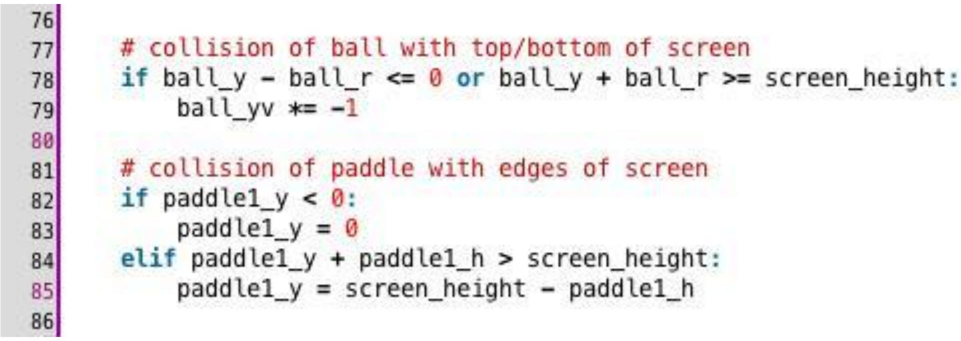


```
# color globals
red = (255, 0, 0)
orange = (255, 127, 0)
yellow = (255, 255, 0)
green = (0, 255, 0)
blue = (0, 0, 255)
violet = (127, 0, 255)
brown = (102, 51, 0)
black = (0, 0, 0)
white = (255, 255, 255)

# screen globals
screen_width = 600
screen_height = 400
game_screen = pygame.display.set_mode((screen_width, screen_height))
pygame.display.set_caption("Tiny Tennis")
font = pygame.font.SysFont("monospace", 75)

# ball globals
ball_x = int(screen_width / 2)
ball_y = int(screen_height / 2)
ball_xv = 3
ball_yv = 3
ball_r = 20

# paddles
paddle1_x = 10
paddle1_y = 10
paddle1_w = 25
paddle1_h = 100

# Due to x position determined at upper left, we combine width of paddle and paddle1_x to achieve same distance
paddle2_x = screen_width - 35
paddle2_y = 10
paddle2_w = 25
paddle2_h = 100

# score
player1_score = 0
player2_score = 0

```

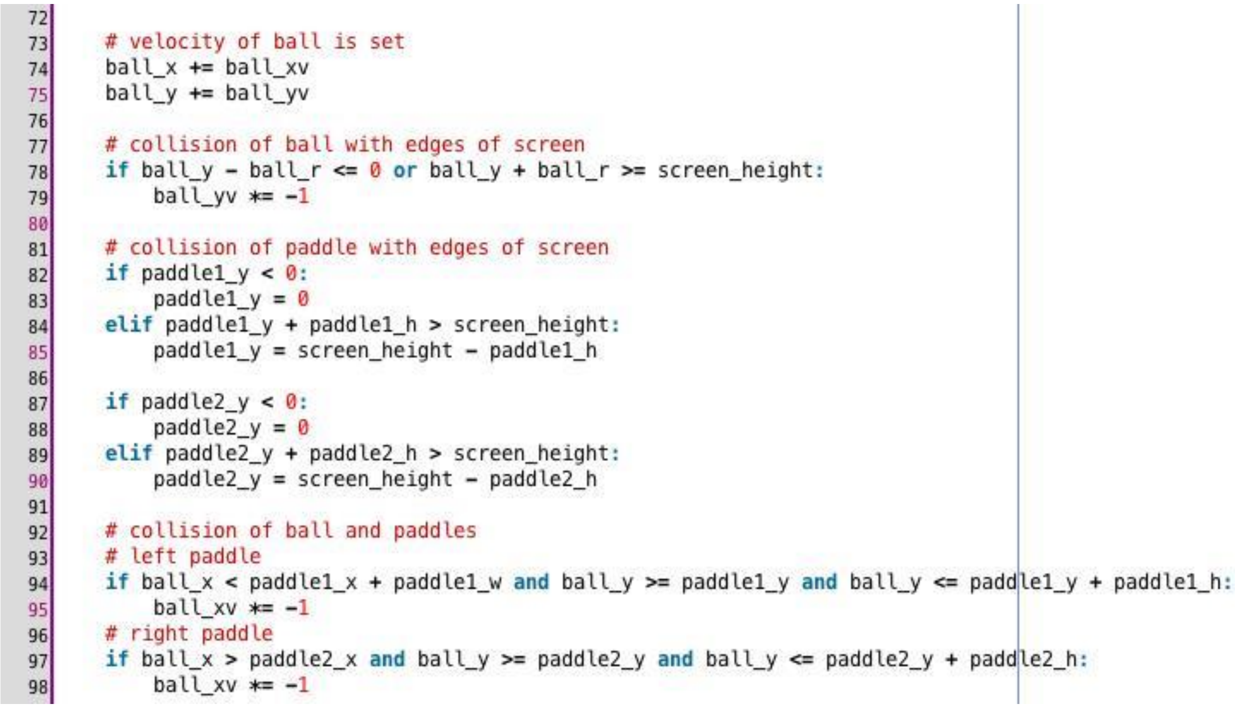


```
# game loop
pygame.mouse.set_visible(0)  # makes mouse invisible in game screen
do_main = True
while do_main:
    pressed = pygame.key.get_pressed()
    pygame.key.set_repeat
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            do_main = False

    if pressed[pygame.K_ESCAPE]:
        do_main = False

    if pressed[pygame.K_w]:
        paddle1_y -= 5
    elif pressed[pygame.K_s]:
        paddle1_y += 5

    if pressed[pygame.K_UP]:
        paddle2_y -= 5
    elif pressed[pygame.K_DOWN]:
        paddle2_y += 5

```

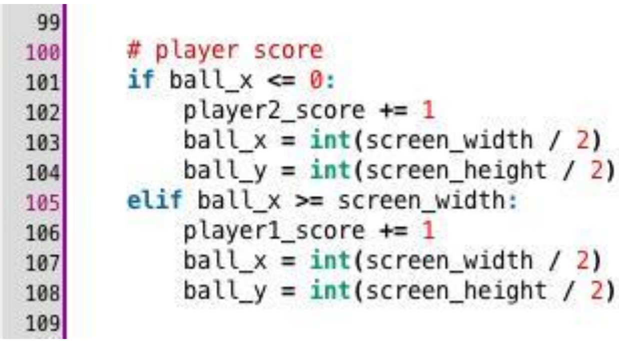




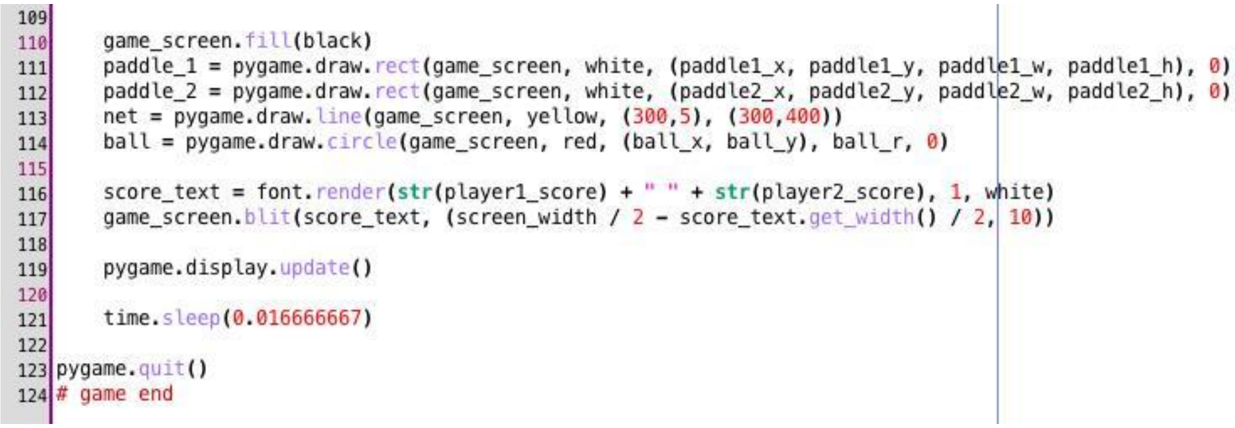
```python

    # collision of ball with top/bottom of screen
    if ball_y - ball_r <= 0 or ball_y + ball_r >= screen_height:
        ball_yv *= -1

    # collision of paddle with edges of screen
    if paddle1_y < 0:
        paddle1_y = 0
    elif paddle1_y + paddle1_h > screen_height:
        paddle1_y = screen_height - paddle1_h

```



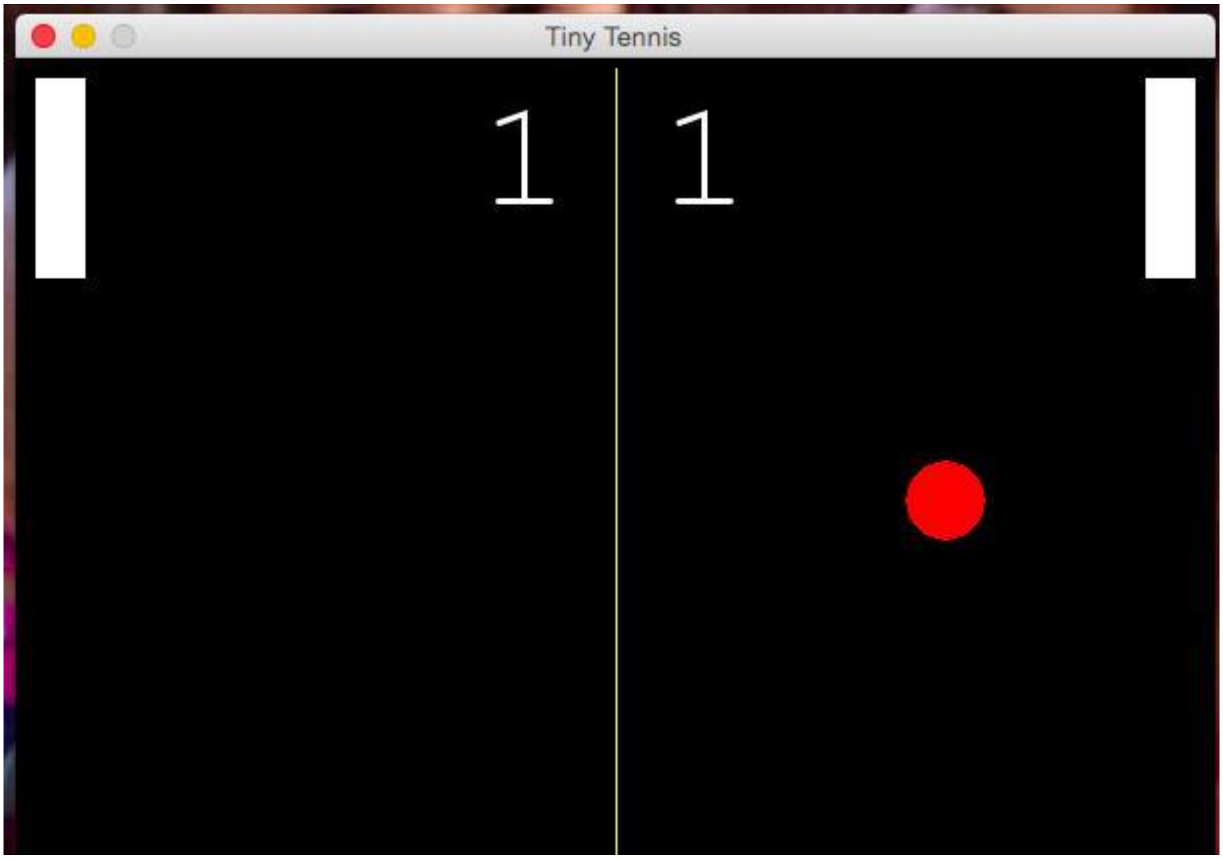
```

    # velocity of ball is set
    ball_x += ball_xv
    ball_y += ball_yv

    # collision of ball with edges of screen
    if ball_y - ball_r <= 0 or ball_y + ball_r >= screen_height:
        ball_yv *= -1

    # collision of paddle with edges of screen
    if paddle1_y < 0:
        paddle1_y = 0
    elif paddle1_y + paddle1_h > screen_height:
        paddle1_y = screen_height - paddle1_h

    if paddle2_y < 0:
        paddle2_y = 0
    elif paddle2_y + paddle2_h > screen_height:
        paddle2_y = screen_height - paddle2_h

    # collision of ball and paddles
    # left paddle
    if ball_x < paddle1_x + paddle1_w and ball_y >= paddle1_y and ball_y <= paddle1_y + paddle1_h:
        ball_xv *= -1
    # right paddle
    if ball_x > paddle2_x and ball_y >= paddle2_y and ball_y <= paddle2_y + paddle2_h:
        ball_xv *= -1
```



```

    # player score
    if ball_x <= 0:
        player2_score += 1
        ball_x = int(screen_width / 2)
        ball_y = int(screen_height / 2)
    elif ball_x >= screen_width:
        player1_score += 1
        ball_x = int(screen_width / 2)
        ball_y = int(screen_height / 2)

```



```

    game_screen.fill(black)
    paddle_1 = pygame.draw.rect(game_screen, white, (paddle1_x, paddle1_y, paddle1_w, paddle1_h), 0)
    paddle_2 = pygame.draw.rect(game_screen, white, (paddle2_x, paddle2_y, paddle2_w, paddle2_h), 0)
    net = pygame.draw.line(game_screen, yellow, (300,5), (300,400))
    ball = pygame.draw.circle(game_screen, red, (ball_x, ball_y), ball_r, 0)

    score_text = font.render(str(player1_score) + " " + str(player2_score), 1, white)
    game_screen.blit(score_text, (screen_width / 2 - score_text.get_width() / 2, 10))

    pygame.display.update()

    time.sleep(0.016666667)

pygame.quit()
# game end
```


Tiny Tennis
1
1

Chapter 10: Keep Coding!