



PRE UNIVERSITY OF  
**MIRROREE**  
Education

Program and Design

**Develop a mini game by programming language**

Author: Chuang Chih-Hsuan

Word count: 1673

# Introduction

This is a basic course of Learning basic knowledge of Python. To understand the features, advantages and applications of Python language. Learning Python syntax, and write a mini game by using Python language.

## ball collision Game Design and Programming

### Introduction of a Programming language---Python

Python is a very beginner-friendly programming language in general speaking. First of all, Python is an interpreted language, so written code can be run without being compiled. In addition, when the programmer wants to run the program, as long as this part of the code has no syntax errors, you can run this code, unlike other compiled language such as c + +, it needs to compile all codes, if found even one syntax error at compile time, all the code cannot run.(Chuang c. h. 2021)

Python, on the other hand, is an object-oriented programming(OOP), and the simple explanation for OOP is to use such a language to create different objects, give them instructions, and let them execute those instructions. Compare to using a process-oriented language, which has a disadvantage that you must simulate the process in your head, and it makes it harder for multiple people to co-write the program, using an object-oriented language, different programmer can write code for different objects, which can be much more productive.(Chuang c. h. 2021)

For Python applications, it can be used for web back-end development, crawlers (which automatically collect data and present it as user wishes), Artificial intelligence and data analysis (any language can do this, but Python's accessibility makes it easier for scholars in other fields, such as mathematicians, who have never learned a programming language. python help them to dig deeper into artificial intelligence)(Chuang c. h. 2021)

Of course, Python has its downsides, and one of its biggest downsides is that while Python is easy to write, it is very inefficient for running compared to other languages

In this course, we learned about Python's two compilers, Idle and VSCode.(Chuang c. h. 2021)

### Learning Process--Basic Syntax

#### Basic Variable

We use Integer to store a data type of integer number ; Float to store a data type of decimal number; String to store a data type of character(include numbers and symbols); And bool only have two type which are True and False.(Chuang c. h. 2021)

#### Basic Operational Rule

“+” represent addition; “-” represent subtraction; “\*” represents multiplication; “/” represents division; “//” represents exact division; “%” represents finding remainder; “\*\*” represents for calculating power.(Chuang c. h. 2021)

## List---features

As with Python ‘s variables, the types of data in a list can be changed flexibly, and different types of data can exist in the same list. And the list doesn't have to declare a length

Every elements in list has particular index which indicated its position in the list. And index starts from 0.(Chuang c. h. 2021)

E.g. `a = {1, "a", True}` // index for 1, "a", True are 0, 1, 2 respectively

## List---Common Methods

Table. 1 function of list method(Chuang.C.H, 2021)

Method	Function
Append()	add elements at the end of list
Insert()	insert elements in any specific position
Len()	obtain amount of elements in the list
Pop()	delete default last elements or specify which to delete by index
Remove()	Remove(): search and delete any element in the list

## Function

Functions are the internal encapsulation of Python. We can break a large piece of code into different functions. By calling different functions, we can break a complex task into simple tasks.(Chuang c. h. 2021)

E.g. `def addsum(number 1, number 2, number 3 ):`

`total = 0`

`For i in range(number+1):`

`total = total + i`

`return total`

// this is a function for calculating the sum of several numbers

## File

IO programming means interaction between user and program like input() and output(), or the interaction between software and hardware system for example, file system.

Reading and writing to files is the most common IO programming, and Python has built-in functions for reading and writing to files.

We must know that the function of read and write files on disk is provided by the operating system, and modern operating systems do not allow the ordinary procedure directly disk operation, so, speaking, reading and writing file requires to the operating system to open a file object, then through from this number to read from the file object or the data is written to the file object.

So, how to write a file?

Actually we have several modes for file writing: one is “w” mode, it clears all content in the file and write at beginning of the file. Another is “a” mode, it add content to end of the data in file. These two mode will also help you construct new file is you haven’t got one.

Format is: `f = file(“file name”, “mode”)`(Chuang c. h. 2021)

E.g. `f = file(“1.txt”, “w”)`

## Turtle

You can imagine turtle as a painter on the screen and programmer use code and functions to control it to move.

E.g.

```
import turtle
```

```
littleturtle.color("blue")
```

```
for i in range(4):
```

```
    littleturtle.forward(200)
```

```
    littleturtle.right(90)
```

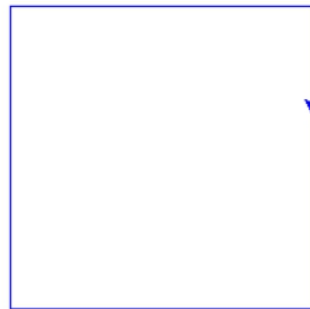


Fig.1 graph of left-hand side example (Chuang c.h., 2021)

## Final project

The game contain A and B two players, A use keys of “w” and “s” to shift paddle A upward and downwards respectively. b use keys of “up” and “down” to shift paddle B upward and downwards respectively. Two player use their paddle to bounce the ball to their opposite side, if one of the player doesn’t catch the ball, then the other player will acquire 1 point.(Chuang c.h. 2021)



Fig.2 graph of game windows (Chuang c.h., 2021)

```
import turtle
```

### **#window**

```
wn = turtle.Screen()
```

```
//define the author information, size,  
background format
```

```
wn.title("Pong by @庄芷亘")
```

```
wn.bgcolor("black")
```

```
wn.setup(width=800,height=600)
```

```
wn.tracer(0)
```

### **#ball**

```
ball = turtle.Turtle()
```

```
//define the speed, shape, color,initial  
position of the ball for collision
```

```
ball.speed(0)
```

```
ball.penup()
```

```
ball.shape("square")
```

```
ball.color("white")
```

```
ball.goto(0,0)
```

```
ball.dy = 0.1
```

```
ball.dx = 0.1
```

### **#paddle A**

```
paddle_a = turtle.Turtle()
```

```
//define shape, size speed, color of  
paddle A
```

```
paddle_a.speed(0)
```

```
paddle_a.shape("square")
```

```
paddle_a.color("white")
```

```
paddle_a.penup()
```

```
paddle_a.goto(-350,0)
```

```
paddle_a.shapesize(stretch_wid=5,stretch_len=1)
```

### **#paddle B**

```
paddle_b = turtle.Turtle()
```

```
//define shape, size speed, color of  
paddle B
```

```
paddle_b.speed(0)
```

```
paddle_b.shape("square")
```

```
paddle_b.color("white")
```

```
paddle_b.penup()
```

```
paddle_b.goto(350,0)
```

```
paddle_b.shapesize(stretch_wid=5,stretch_len=1)
```

### **#score board**

```
pen = turtle.Turtle()
```

```
//define shape, size speed, color of score  
board. Define the content, position and  
font of text on the board
```

```
pen.speed(0)
pen.shape("square")
pen.color("white")
pen.penup()
pen.hideturtle()
pen.goto(0,260)
pen.write("Player A : 0   Player B : 0",align = "center",font=("Courier",24,"normal"))
```

```
score_a =0
score_b =0
```

### #function control board to move

```
def paddle_a_up():
    y = paddle_a.ycor()
    y += 20
    paddle_a.sety(y)
```

//define the function for moving paddle .  
move paddle up and down by adding or  
subtracting 20 unit of distance at y  
coordinate of paddle for every time click on  
the control key

```
def paddle_a_down():
    y = paddle_a.ycor()
    y -= 20
    paddle_a.sety(y)
```

```
def paddle_b_up():
    y = paddle_b.ycor()
    y += 20
    paddle_b.sety(y)
```

```
def paddle_b_down():
    y = paddle_b.ycor()
    y -= 20
    paddle_b.sety(y)
```

### #keyboard bindings

```
wn.listen()
wn.onkeypress(paddle_a_up,"w")
wn.onkeypress(paddle_a_down,"s")
```

//define which control key correspond to  
which locomotion

//A use keys of "w"and "s" to shift paddle A  
upward and downwards respectively. b use  
keys of "up"and "down" to shift paddle B  
upward and downwards respectively.

```
wn.onkeypress(paddle_b_up,"Up")
wn.onkeypress(paddle_b_down,"Down")
```

### #main loop

while True:

```
    ball.setx(ball.xcor()+ ball.dx)
    ball.sety(ball.ycor()+ ball.dy)
```

//The ball move by repetition of adding same difference to its x and y coordinates at same time

```
    wn.update()
```

### # Top and bottom

```
    if ball.ycor()>290:
        ball.sety(290)
        ball.dy = ball.dy* -1
```

// if the ball reach the top or bottom side of the windows, turn the difference negatively at direction on y axis , then the ball will bounce at the edge of window.

```
    if ball.ycor()<-290:
        ball.sety(-290)
        ball.dy = ball.dy* -1
```

### #left and right

// if the ball reach the paddle, turn the difference negatively at direction on x axis , then the ball will bounce at the paddle.

### #catch by paddles-> bounce

```
    if ball.xcor()<-340 and ball.ycor()<paddle_a.ycor()+50 and ball.ycor()>paddle_a.ycor()-50:
        ball.dx*=-1
```

```
    elif ball.xcor()>340 and ball.ycor()<paddle_b.ycor()+50 and ball.ycor()>paddle_b.ycor()-50:
        ball.dx*=-1
```

// if the ball doesn't reach the paddle, place the ball to its initial position at beginning of the game

### #Not catch by paddles-> move the ball to origin and modify score

```
    if ball.xcor()>390:
        score_a = score_a + 1
        ball.goto(0,0)
        ball.dx = ball.dx* -1
        pen.clear()
        pen.write("Player A : {} Player B : {}".format(score_a,score_b),align =
"center",font=("Courier",24,"normal"))
```

//then if A doesn't catch the ball, score of B add 1 point, else if B doesn't catch the board, score of A at 1 point

```
    if ball.xcor() < -390:
        score_b = score_b + 1
```

---

```
ball.goto(0,0)
ball.dx = ball.dx* -1
pen.clear()
pen.write("Player A : {}  Player B : {}".format(score_a,score_b),align =
"center",font=("Courier",24,"normal"))
```

## Conclusion

After the course, I gained a lot. I had a basic understanding of Python, and step in learning of Python officially. I learned a lot of functions of Python and build a strong foundation Both for in-depth learning of Python and learning other programming languages in the future. In addition, this course also makes me know the direction of further learning in Python by my own,(Chuang c.h. 2021)