6/1/2020-6/26/2020 homework and class notes

by Austin

> Teacher emails: wangqianjiang@gmail.com

> send hw here: wangqianjiang@live.com

> github link: https://github.com/jwang1122/python1-1130

setting up python and github

install python:

> Google Search: python download Use this result: Download Python | Python.org

> Website

we used command prompt to set up python

cd = change directory

```
cd workspace
cd python
```

To check for version of aplication type the application and --version

EX:

```
git --version
```

with this the computer will tell you the current version of github, or whatever you type, on your computer.

Install gihub:

> Google Search: github download Use this result: Git - Downloads - Git SCM

> Website

code things

basics of printing

```
print("hello, world!")
print("2","3","4")
```

more complex:

```
print("4","3","2",end=' ', sep="; ")
```

here the it will print 4; 3; 2; because of sep=";". It will also put a space after the line it printed.

you can also set the things you print to letters:

```
a = "joe"
b = "ham"
print(a,b)
```

when you run this script it will print joe ham. The above end=" " and sep="; " will also work with this.

we opened .env folder with python -m -venv env

in python, ' and " are the same thing

Tuple: a type of variable where there are three pieces of information.

help() will give you help

type() will tell you the class of whatever is in the ()

if all your code is right but it says there are problems, then the editor has a setup issue.

In python numbers are automaticly set to integers.

In python you can out set mutiple pieces of information on one line seperated by commas:

```
a,b, = 1,5
a,b,c,d, = 1,3,5,2
```

You can do math with python:

multiplication: * | division: / | addition: + | subtraction: - |

```
12*4, 92*3, a/b, c+d,
```

you can also print the answers python caculates:

drawing

basics of drawing

ball1

```
from turtle import *

t1 = Turtle()
t1.dot(20)
mainloop()
```

pythonhas a thing called turtle built in to draw stuff.

Here we import turtle o python can draw stuff. t1 = Turtle() set the thing python is going to draw to t1. It can be set to anything except turtle or python will be confused. t1.dot(20) makes the dot's radius 20. mainloop() keeps a window open to show you the drawing. you will see a black triangle. That is the thing that draws whatever you input into python.

add stuff

ball2

```
from turtle import *

t1 = Turtle()
t1.dot(20,'red')
t1.hideturtle()
mainloop()
```

here we have made the dot red. t1.dot(20,'red') Also we hid the black triangle so u cant see it anymore. 😃 t1.hideturtle()

now we add more

ball3

```
from turtle import *

t1 = Turtle(visible=False)

def draw():
    t1.pu()
    tracer(False)
    t1.setpos(0, -180)
    t1.dot(20)

draw()
mainloop()
```

here we used another way to hide the turtle: t1 = Turtle(visible=False) draw() will tell python to run everything in def draw() As you can see there are lines of code that are indented by pressing the tab button. The indention is how python knows that is what to do when def draw(): is run. This code will tell python to draw the circle at (0,-180) instead of at (0,0) t1.setpos(0, -180)

now this one is much more complex:

ball4

```
from turtle import *
from freegames import vector

t1 = Turtle(visible=False)
t1.pu()

def draw():
    t1.clear()
    t1.goto(pos.x, pos.y)
    t1.dot(20)

    ontimer(draw, 20)

def west():
    pos.x -= 30


def east():
    pos.x += 30

pos = vector(0,-200)
tracer(False)
onkey(west, 'Left')
onkey(east, 'Right')
listen()

draw()

exitonclick()
```

Basically we just put pen up so it doesnt draw we we move the dot with left and right arrow keys with this:

```
def west():
    pos.x -= 30


def east():
    pos.x += 30
```

these lines of code will tell python to execute the code above then right and left arrow key is pressed:

```
onkey(west, 'Left')
onkey(east, 'Right')
```

we also made the ball move when fired

now lets Move the ball gradually on fire and addd some customization

ball5

```
from turtle import *
from freegames import vector

t1 = Turtle(visible=False)
t1.penup()


def draw():
    t1.pu()
    t1.clear()
    t1.goto(pos.x, pos.y)
    t1.dot(20)

    if speed.y != 0:
        speed.y += 0.35
        pos.move(speed)

    ontimer(draw, 20)

def west():
    pos.x -= 30


def east():
    pos.x += 30

def fire():
    speed.y = 25

pos = vector(0,-200)
speed = vector(0, 0)
tracer(False)
onkey(west, 'Left')
onkey(east, 'Right')
onkey(fire, "space")
listen()

draw()

exitonclick()
```

here is what we did to move the ball gradually on fire

```
    if speed.y != 0:
        speed.y += 0.35
        pos.move(speed)
```

and we have a timer

```
ontimer(draw, 20)
```

now lets make it so when the ball is out of the screen, python will put the ball back.

ball6

```
from turtle import *
from freegames import vector

cannon = Turtle(visible=False)
cannon.pu()
cannon.screen.bgcolor("aqua")

def draw():
    cannon.pu()
    cannon.clear()
    cannon.goto(pos.x, pos.y)
    cannon.dot(20,'red')

    if speed.y != 0:
        speed.y += 0.35
        pos.move(speed)
        if pos.y > 300:
            speed.y = 0
            pos.y = -200


    ontimer(draw, 20)

def west():
    pos.x -= 30


def east():
    pos.x += 30

def fire():
    speed.y = 10

pos = vector(0,-200)
```

```
speed = vector(0, 0)
tracer(False)
onkey(west, 'Left')
onkey(east, 'Right')
onkey(fire, "space")
listen()

draw()

exitonclick()
```

now after firing python will exevute this script:

```
pos = vector(0,-200)
```

now draw() has been changed:

```
def draw():
    cannon.pu()
    cannon.clear()
    cannon.goto(pos.x, pos.y)
    cannon.dot(20,'red')
```

as you can see, t1 is now named cannon and it's color has been change to red.

the background color has changed too:

```
cannon = Turtle(visible=False)
cannon.pu()
cannon.screen.bgcolor("aqua")
```

we used cannon.screen.bgcolor("aqua") to change the bg color to aqua.