unit 3

Graphics and Animation:

Event-Driven Programming Part 1 - Responses and Handlers

Events are "listened" for by the Python turtle window

def: ***event-driven programming*** is writing code that produces an interactive experience for the user.

def: an ***event*** is something that a user of an app does that causes a reaction from the computer.

def: a ***response*** is the name given to the reaction that a computer provides when "handling" an event.

def: a ***handler*** is a custom Python verb that teaches the computer how to do a particular response.

Q:

How do I define a handler?

A:

Use the Python verb **def**

Means "define"

Example of handler definitions

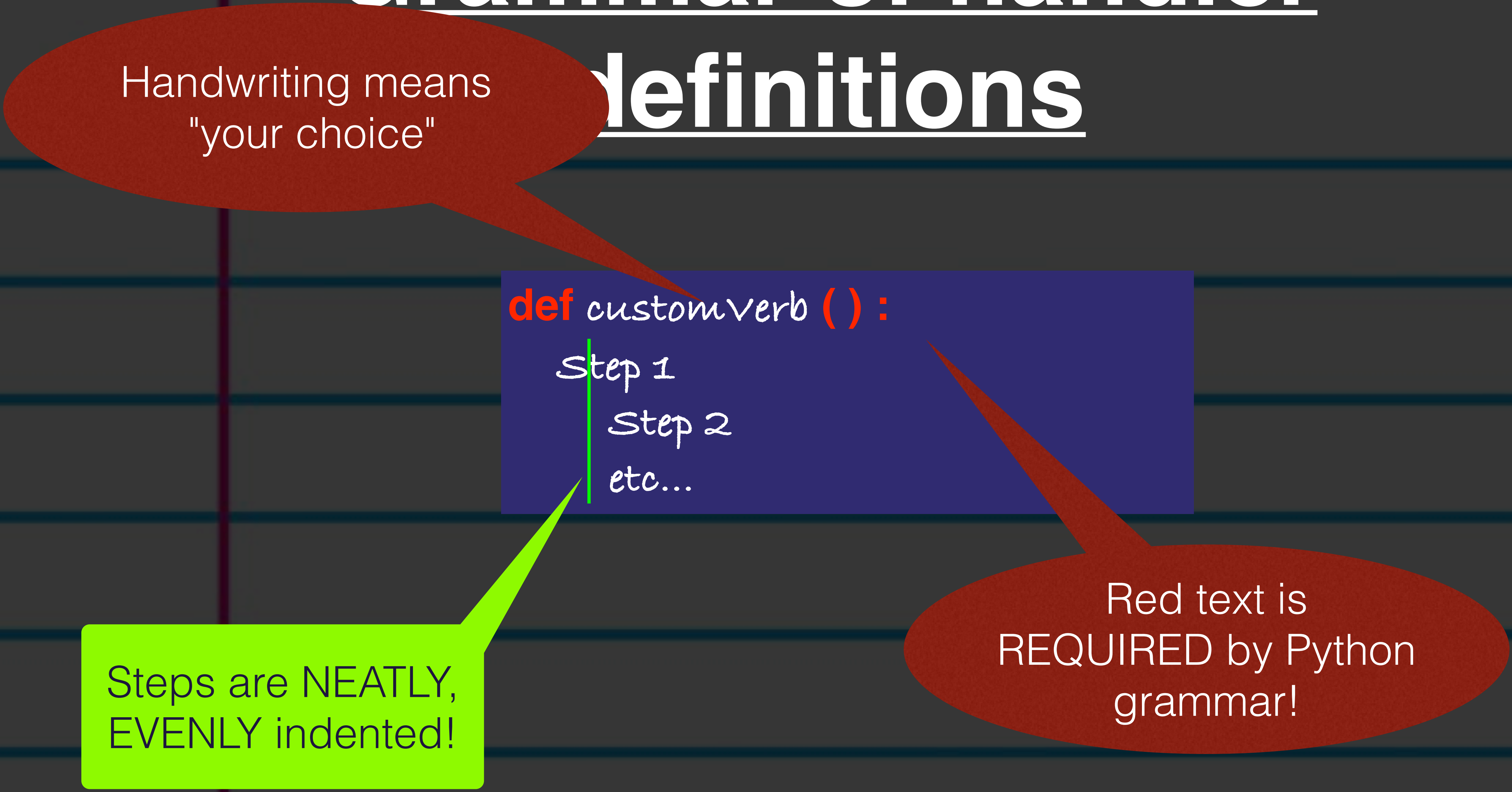
```
def drawDash ( ) :
    bob.forward (10)
    bob.penup ( )
    bob.forward (10)
    bob.pendown ( )
```

```
def pizza ( ) :
    bob.fillcolor ('red')
    bob.begin_fill ( )
    bob.circle (160, 90)
    bob.end_fill ( )
```

```
def drawDash ( ) :
    bob.forward (10)
    bob.penup ( )
    bob.forward (10)
    bob.pendown ( )
```

```
def happyFace ( ) :
    print ('c:')
```

Grammar of handler definitions



Handwriting means "your choice"

```
def customVerb ( ) :
    Step 1
      Step 2
      etc...
```

Steps are NEATLY, EVENLY indented!

Red text is REQUIRED by Python grammar!

Let's Write Handlers!

- Meet me at repl.it so that we can work together
- Create a new Python project called **handlerExamples**
- Wait for further instructions