Author: Robert Cheung – [robertckcheung@gmail.com](mailto:robertckcheung@gmail.com)  
License: Creative Commons Attribution-ShareAlike (CC BY-SA)  
[http://i.creativecommons.org/l/by-sa/3.0/88x31.png](https://creativecommons.org/licenses/by-sa/4.0/legalcode)

## History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Change Notes | Author |
| 20140324 | 0.9 | Initial Version | Robert Cheung |
| 20140329 | 1.0 | Added Section “Programming Construct”  Added Lessions: 2-1, 2-2, 2-3 and 2-4 | Robert Cheung |
| 20150405 | 2.0 | Added Scratch Comparisons to “Programming Constructs” section. (scratch.mit.edu) | Robert Cheung |

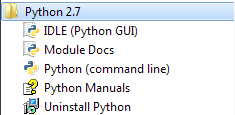
## Notes to mentors:

This tutorial has several stages:

1. How to get started in Python (on Windows and Macs)
2. Introduction to the IDLE Graphical Edit Environment
3. Introduction to Python – The Skeleton of a Program
4. Introduction to Turtle Graphics – Moving Pens Around
5. Programming Constructs
   1. Sequence – Lets draw something!
   2. Procedures – Make repeating myself easy
   3. Loops – Repeating
   4. Logic and Branching – Making Choices

# Getting Started on Python

## MS Windows

1. Download and install:   
   <https://www.python.org/downloads/>   
   This tutorial was developed in python 2.7 but any 2.x and 3.x versions should work.
2. Select IDLE (Python GUI) from Start > All Programs  
   

### Apple Mac OS X

On Macs Python is installed by default:

1. Go into your Applications folder
2. Find and double click on IDLE – the Python Graphical Editing Environment

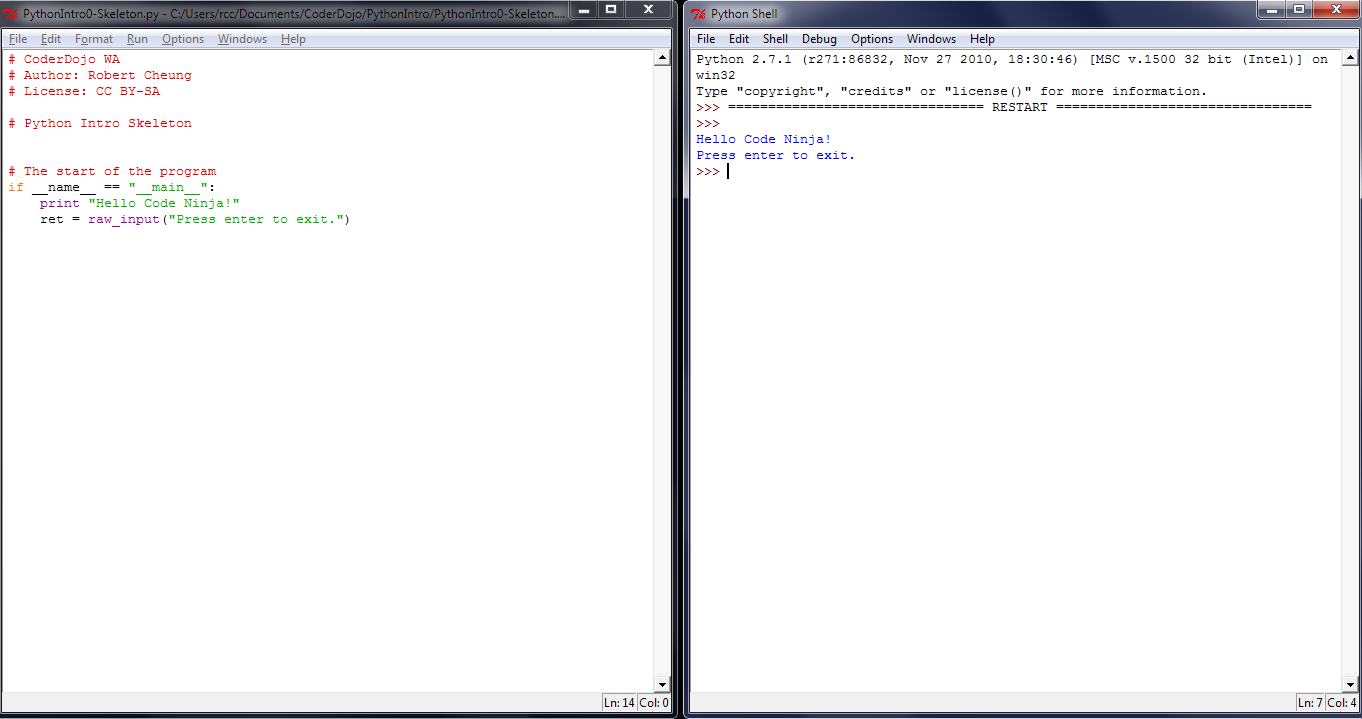
More information: http://docs.python.org/2/using/mac.html

# Python IDLE

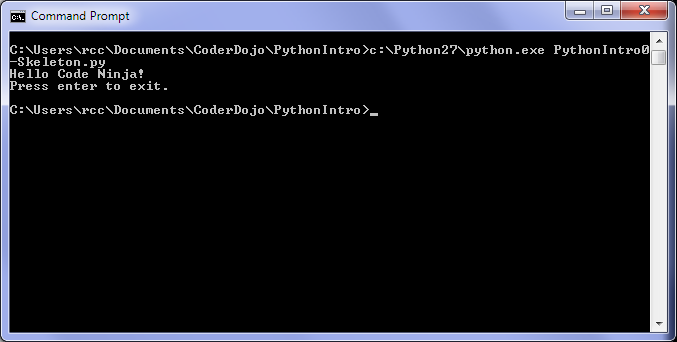
The Python programming language comes with an editing and execution environment. It allows you edit your code and run it in the same environment.

After starting IDLE, load up an example program and run it:

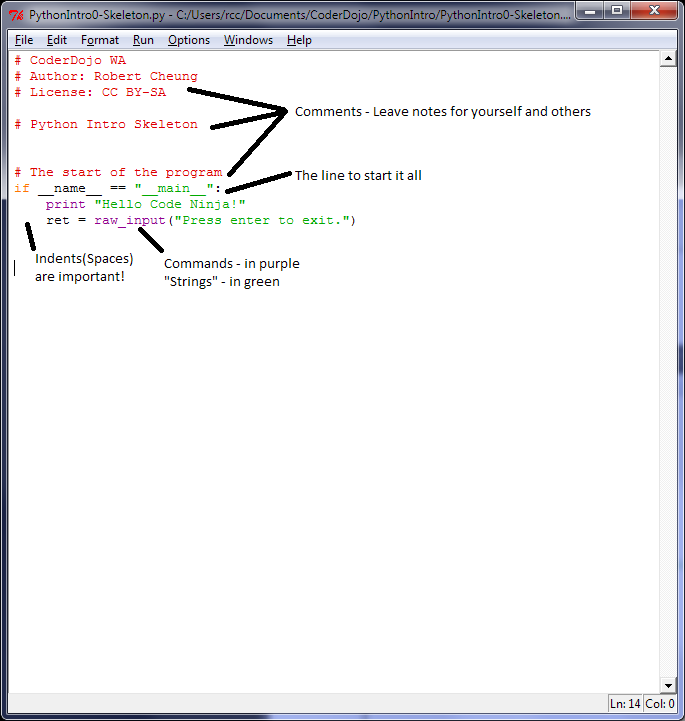
1. Select (click) on the IDLE Window
2. File > Open
3. Select the file “PythonIntro0-Skeleton.py” – this will open a new editor window with your program in it.
4. Select (click) on the code editor window
5. Run > Run Module (F5)
6. Your program will be running in the “Python Shell” window.



Mentors Note: It is not required to create or run your python programs within IDLE. There are many choices, but IDLE is a standard part of python and hence used by this tutorial. You can create a .py file with any text editor and run python program is straight from the command line.



# Introduction to Python – The Skeleton of a Program

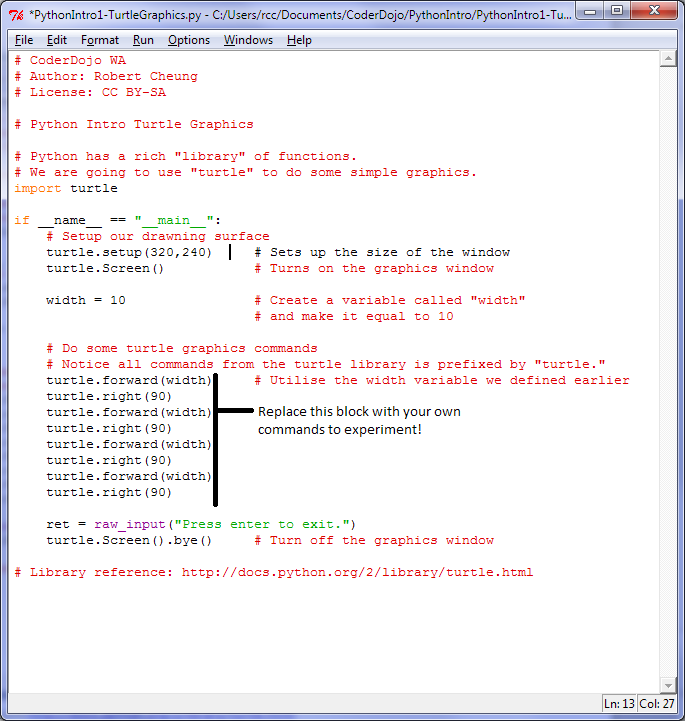


At the end of the last section you would have opened the first python, PythonIntro0-Skeleton.py, program included with this lesson. Let’s examine this closer:

1. The lines starting with a hash, ‘#’, are called comments. They have no effects to the program. Their purpose is to allow the programmers to leave notes to what the program does.
2. Technical Python mumbo jumbo. We will ignore for now, apart from knowing that this line is where the program will start.
3. Commands are automatically coloured purple by IDLE. Strings are in green.
4. Generates some output
5. Accept input after displaying a message

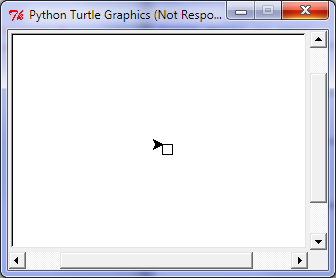
Remember: You can hit F5 (or Run > Run Module) to see how the program works. Do it now.

# Introduction to Turtle Graphics – Moving Pens Around



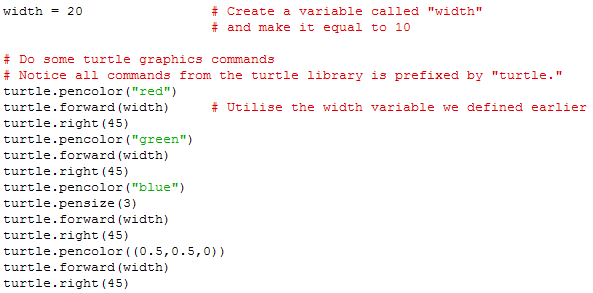
More information: <http://docs.python.org/2/library/turtle.html>

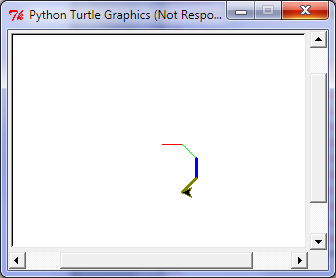
Imagine you have a canvas to draw on and you have turtles with pens that could follow your instructions. That is turtle graphics! To see what these turtles can do for you, load up “PythonIntro1-TurtleGraphics.py” and run it.



Now replace the block of turtle commands with some of your own. <http://docs.python.org/2/library/turtle.html#pen-control>

Try changing pen colors and angles. If you are stuck, here is a suggestion:

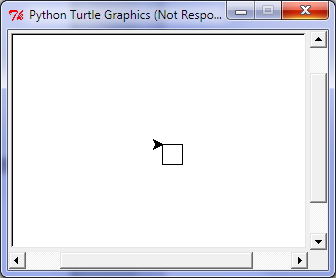


You can see “PythonIntro1-TurtleGraphics2.py”

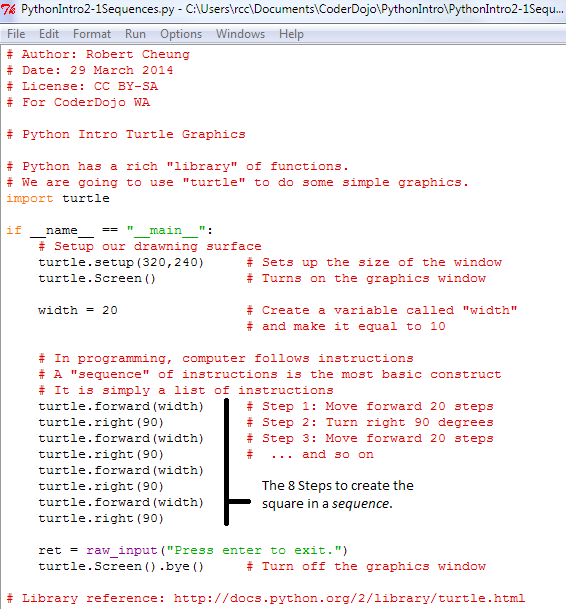
# Programming Constructs

## Sequence – Lets draw something!

Load and run “PythonIntro2-1Sequences.py”.



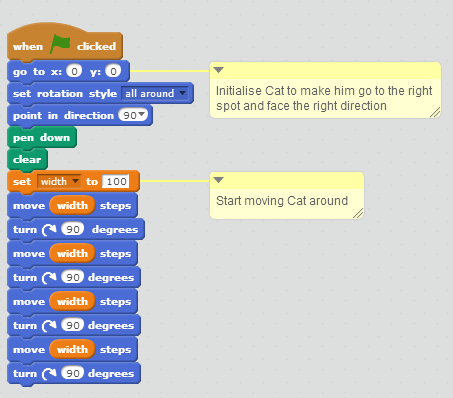
You can see that a square has been drawn. The instruction to create the square took 8 lines of code:

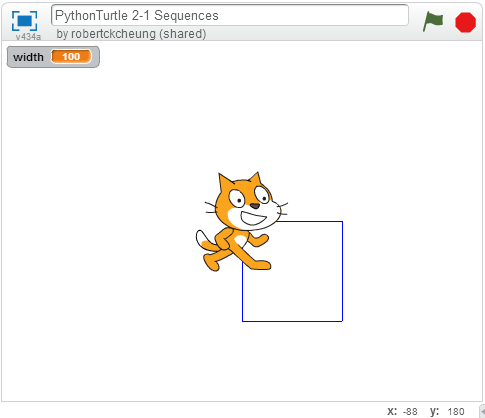


### Attention CoderDojo Ninjas

For people who are familiar with Scratch (http://scratch.mit.edu), the following Scratch snippet performs a very similar role. You can find all my projects by one of the following methods:

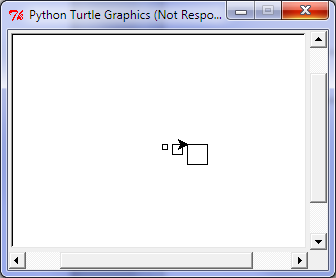
* My Scratch login of robertckcheung.
* Studio: <https://scratch.mit.edu/studios/1103980/>
* Project: https://scratch.mit.edu/projects/55770118/

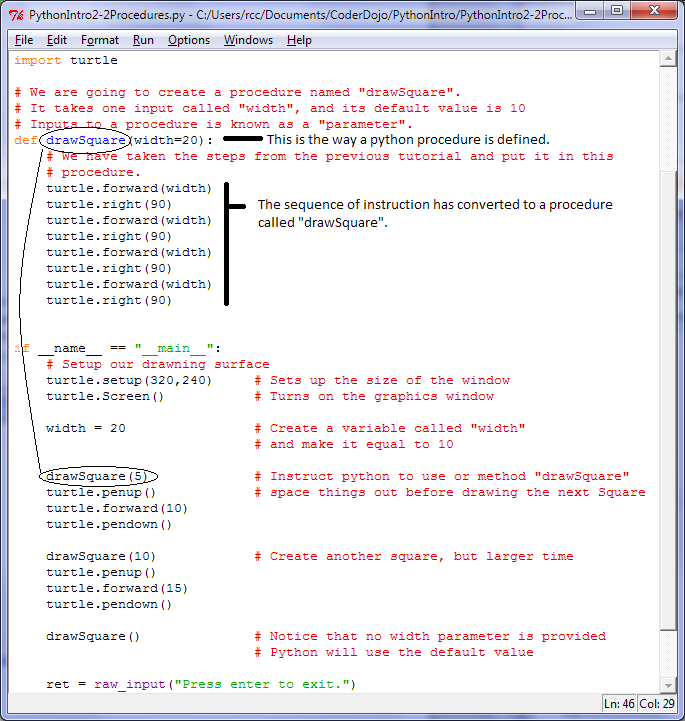




## Procedures – Make repeating myself easy

*Procedure* is the term given to a snippet of code that is named. By giving it a name, it allows you to tell the computer to repeat the code easily. Think of it as a recipe that you might want to repeat. Depending on the language and environment, they are also known as *methods* and *functions.*

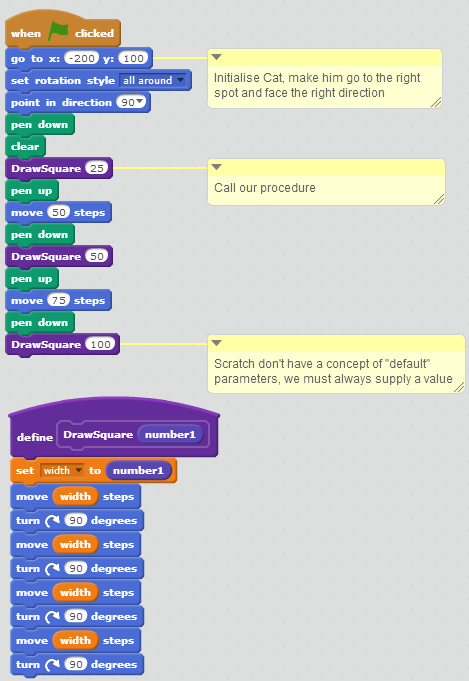
Load and run “PythonIntro2-2Procedures.py”.  
 

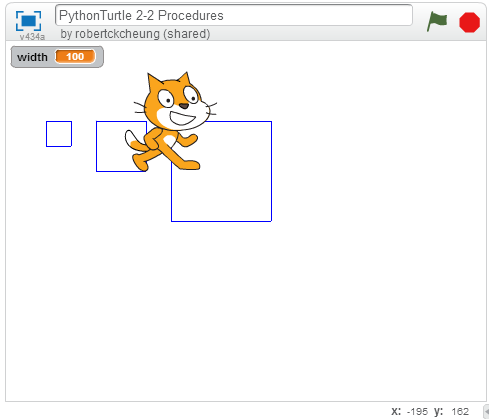


### Attention CoderDojo Ninjas

Scratch Project “PythonTurtle 2-2 Procedures” has the Scratch equivalent to this tutorial. Note that the concept of “Default parameter” does not exist in scratch.

<https://scratch.mit.edu/projects/55770374/>





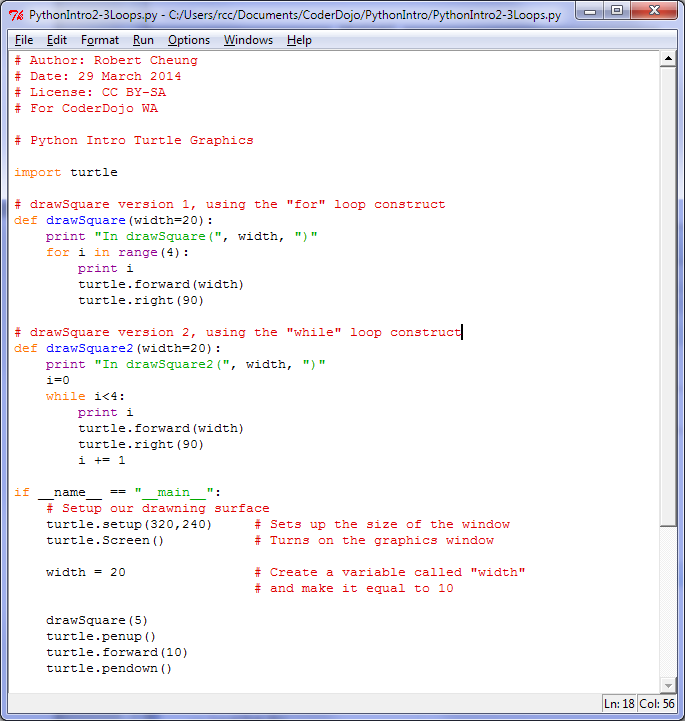
## Loops – Repeating

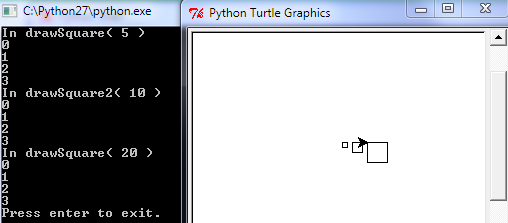
There are several types of loops in Python.

* Fixed loops use the “for *variable* in *list*” syntax.
* Variable loops use the “while *test*:” syntax.

Load and run “PythonIntro2-3Loops.py”. Pay particular attention to:

1. The two versions of “drawSquare()”. Both of them use a looping command.
2. The output in the main screen, it prints some interesting information as the program proceeds.





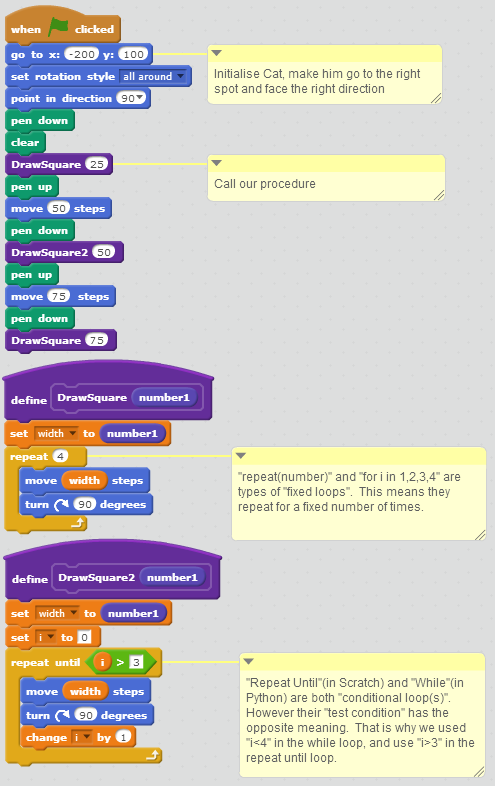
### Attention CoderDojo Ninjas

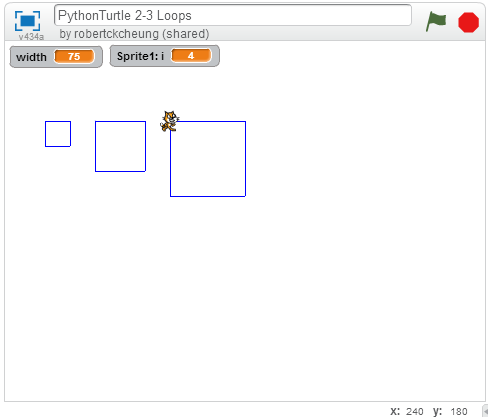
Again the “PythonTurtle 2-3 Loops” Scratch project has mirrored this python/turtle graphics tutorial.

<https://scratch.mit.edu/projects/55770750/>

In Scratch, the “Repeat(number)” Control block is the same as python’s “For <variable> in <range>” construct, and they are both examples of a fixed loop.

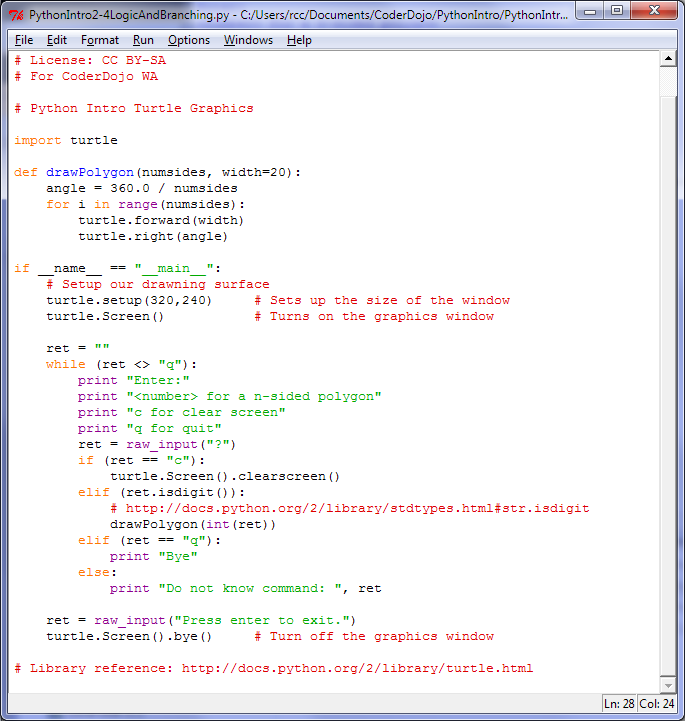
The “Repeat Until <condition>” in Scratch is similar to the “While <condition>” construct in Python. They are examples of “conditional loops”, that is the loop will continue until the condition becomes false (repeat until) or true (while).

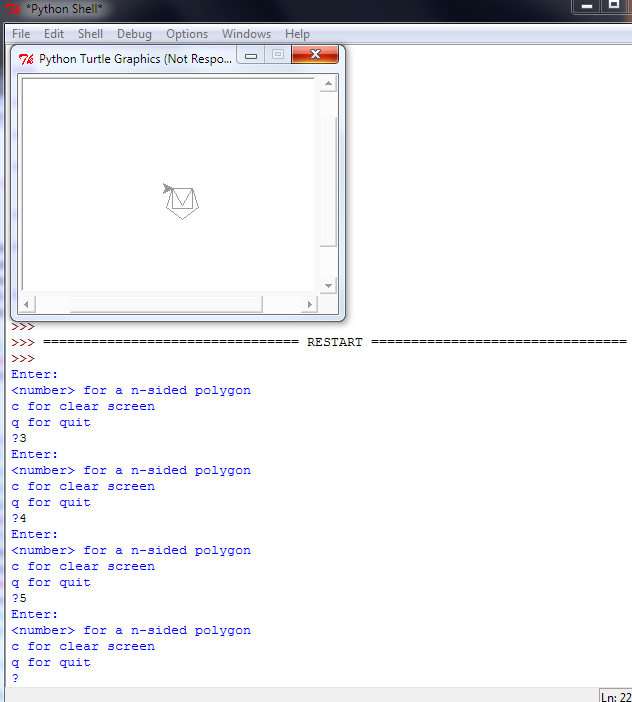




## Logic and Branching – Making Choices

Load and run “PythonIntro2-4LogicAndBranching.py”





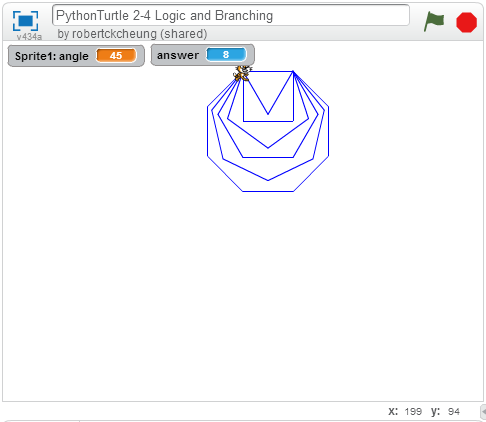
Pay particular attention to the “if/elif/else” statement.

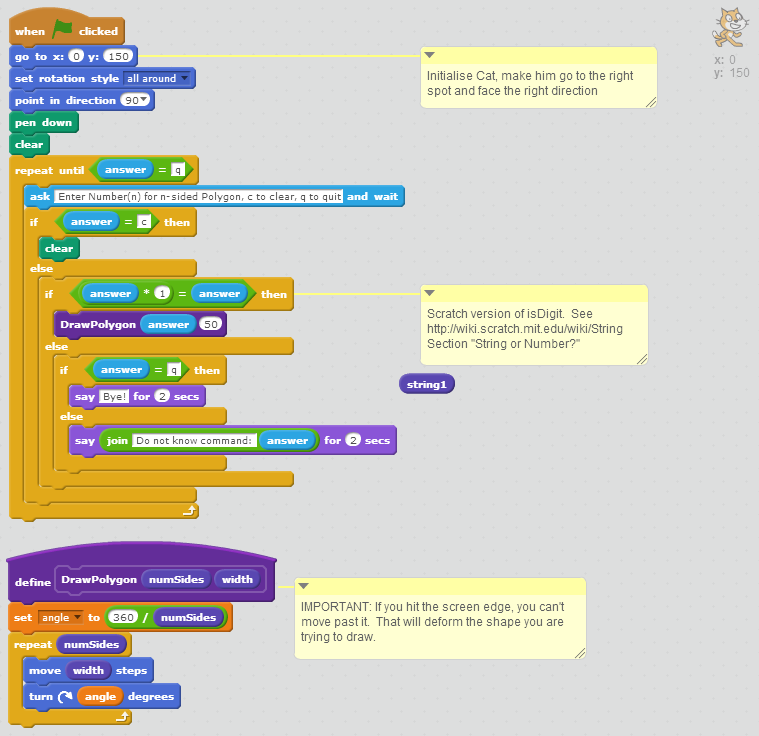
Also observe the new “drawPolygon” procedure.

Insert some print statements into the program to learn what it is doing.

### Attention CoderDojo Ninjas

https://scratch.mit.edu/projects/55771410/





# Whats Next?

Create a series of procedures that will draw letters of the alphabet. Then call on the procedures one at a time to write your name!