Laboratory 3

COMSC-122

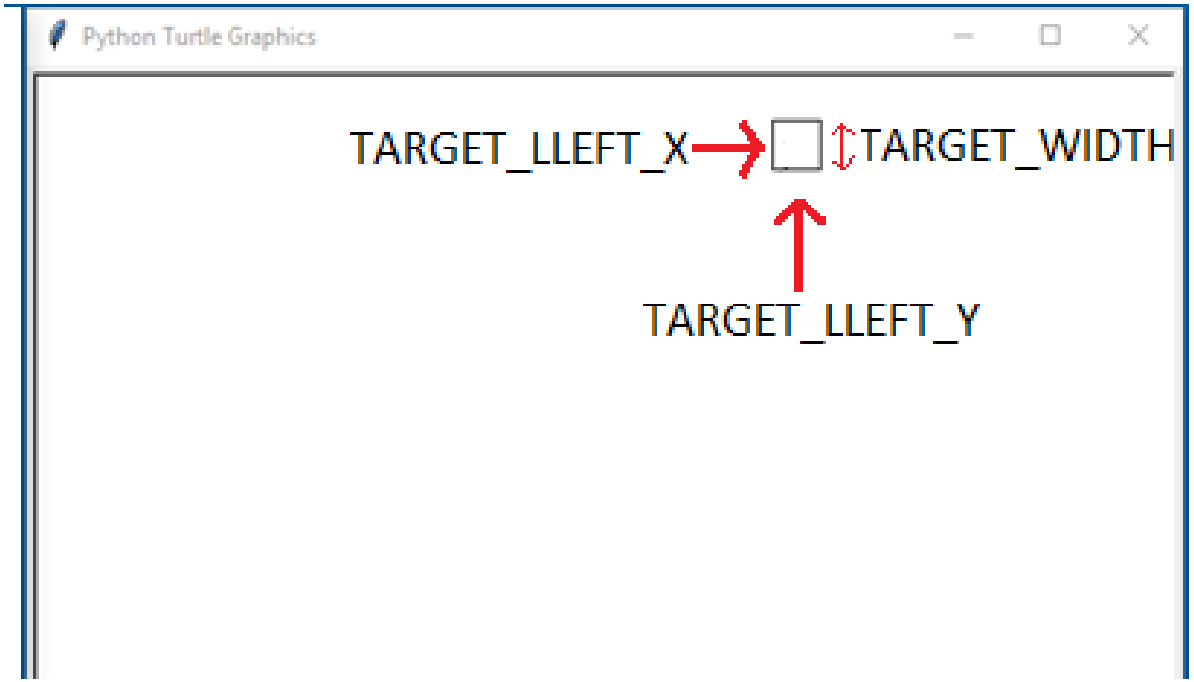
Fall 2017

Turtle Graphics: Determining the State of the Turtle

- In our laboratory, we are going to create a target and then see if we can hit it by firing a projectile at an **angle** we type in, with enough **force** to hit it, but not too much force so that we overshoot it.
- Below is shown a projectile fired at a particular **angle**, with just enough **force** to land on the target.

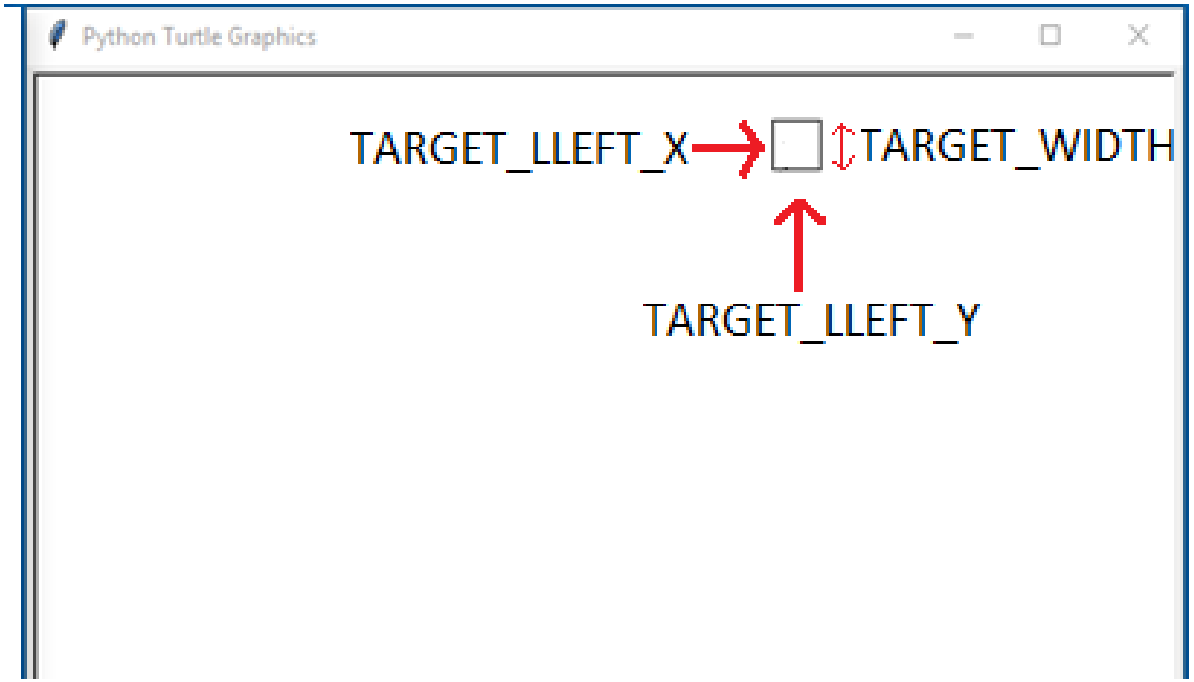
Turtle Graphics: Is Turtle Inside Target, or Outside?

- The X coordinate of the left hand side of the target square is: TARGET_LLEFT_X.
- The Y coordinate of the bottom side of the target square is: TARGET_LLEFT_Y.
- The X coord. of right hand side of target square is: TARGET_LLEFT_X + TARGET_WIDTH.
- The Y coord. of top side of target square is: TARGET_LLEFT_Y + TARGET_WIDTH.



The four tests that the Turtle must pass to be on Target are:

- turtle.xcor() >= TARGET_LLEFT_X
- turtle.xcor() <= TARGET_LLEFT_X + TARGET_WIDTH
- turtle.ycor() >= TARGET_LLEFT_Y
- turtle.ycor() <= TARGET_LLEFT_Y + TARGET_WIDTH



Lab03A

- Type in the program **hit_the_target.py**
- Once you have it typed in, determine the following two values that will cause you to hit the target successfully when you run it:
  - What is the projectile angle that will work?
  - What is the launch force that you need to use so that you will hit the target, but not overpass it?
- Determine two values that will cause the target to be hit for case A.

Program 3-9 (hit_the_target.py)

```
1  # Hit the Target Game
2  import turtle
3
4  # Named constants
5  SCREEN_WIDTH = 600      # Screen width
6  SCREEN_HEIGHT = 600     # Screen height
7  TARGET_LLEFT_X = 100    # Target's lower-left X
8  TARGET_LLEFT_Y = 250    # Target's lower-left Y
9  TARGET_WIDTH = 25       # Width of the target
10 FORCE_FACTOR = 30       # Arbitrary force factor
11 PROJECTILE_SPEED = 1    # Projectile's animation speed
12 NORTH = 90              # Angle of north direction
13 SOUTH = 270             # Angle of south direction
14 EAST = 0                # Angle of east direction
15 WEST = 180              # Angle of west direction
16
17 # Setup the window.
18 turtle.setup(SCREEN_WIDTH, SCREEN_HEIGHT)
19
```

Program 3-9
hit_the_target.py
page 1 of 3

```
20  # Draw the target.
21  turtle.hideturtle()
22  turtle.speed(0)
23  turtle.penup()
24  turtle.goto(TARGET_LLEFT_X, TARGET_LLEFT_Y)
25  turtle.pendown()
26  turtle.setheading(EAST)
27  turtle.forward(TARGET_WIDTH)
28  turtle.setheading(NORTH)
29  turtle.forward(TARGET_WIDTH)
30  turtle.setheading(WEST)
31  turtle.forward(TARGET_WIDTH)
32  turtle.setheading(SOUTH)
33  turtle.forward(TARGET_WIDTH)
34  turtle.penup()
35
36  # Center the turtle.
37  turtle.goto(0, 0)
38  turtle.setheading(EAST)
39  turtle.showturtle()
40  turtle.speed(PROJECTILE_SPEED)
41
```

Program 3-9
hit_the_target.py
page 2 of 3

```
42  # Get the angle and force from the user.
43  angle = float(input("Enter the projectile's angle: "))
44  force = float(input("Enter the launch force (1-10): "))
45
46  # Calculate the distance.
47  distance = force * FORCE_FACTOR
48
49  # Set the heading.
50  turtle.setheading(angle)
51
52  # Launch the projectile.
53  turtle.pendown()
54  turtle.forward(distance)
55
56  # Did it hit the target?
57  if (turtle.xcor() >= TARGET_LLEFT_X and
58      turtle.xcor() <= (TARGET_LLEFT_X + TARGET_WIDTH) and
59      turtle.ycor() >= TARGET_LLEFT_Y and
60      turtle.ycor() <= (TARGET_LLEFT_Y + TARGET_WIDTH)):
61          print('Target hit!')
62  else:
63          print('You missed the target.')
```

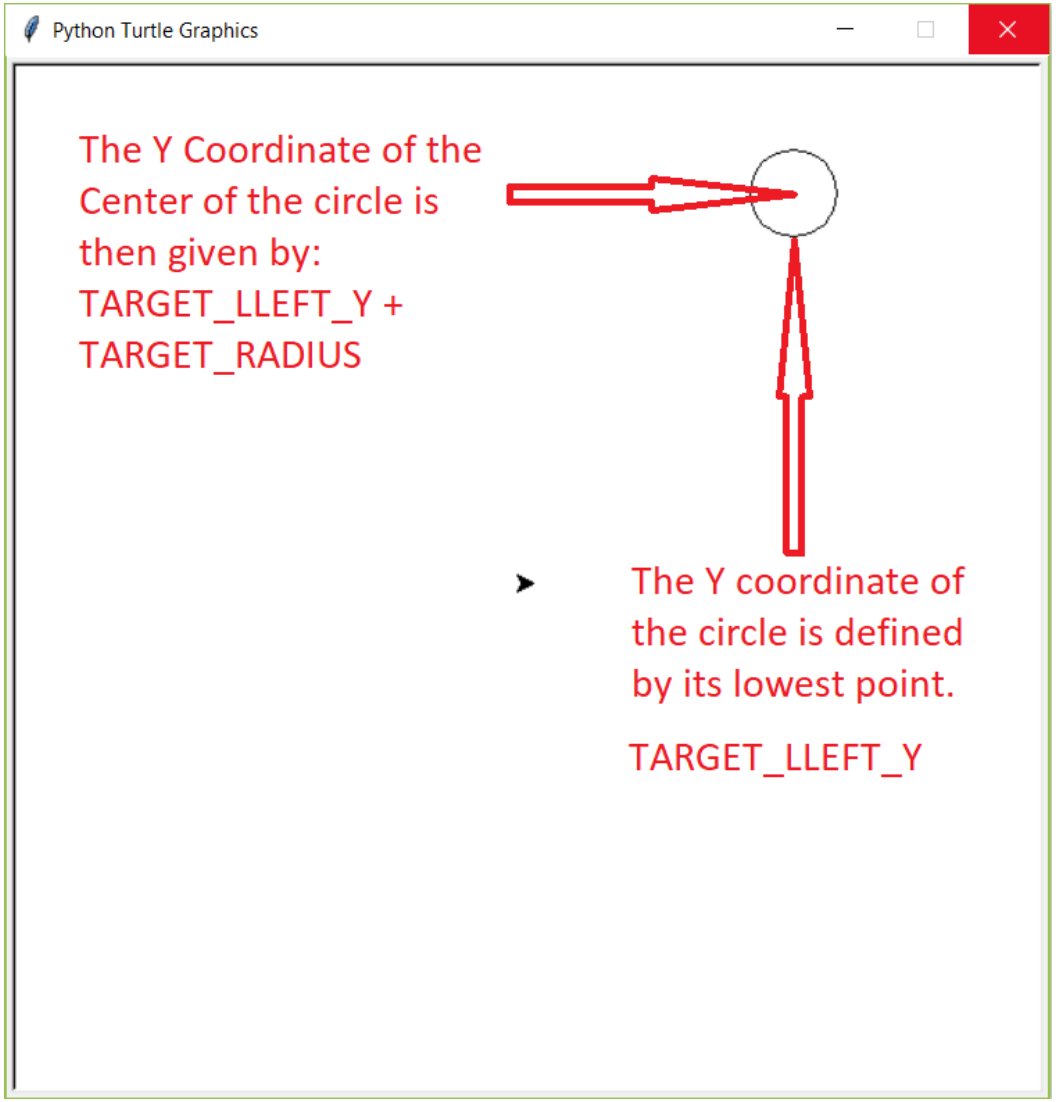
Program 3-9 hit_the_target.py page 3 of 3

Lab03B

- Let's change the position and the size of the target for case B:
  - TARGET_LLEFT_X = 150
  - TARGET_LLEFT_Y = 200
  - TARGET_WIDTH = 5
- What values would you have to give for **angle**, and **force** in order to hit the target now?
- Using a whole number for the **angle**:
  - What is the smallest **force** that will work?
  - What is the largest **force** that will work?
- Now call the instructor over to review your work.
- Remember, only work demonstrated in the Lab, to the instructor, will be accepted.

Lab03C

- As it turns out, if we change the target from a rectangle to a circle, the solution turns out to be easier.
  - In Lab03A we are testing to see if the X-coordinate of the Turtle is greater than the near side of the rectangle but less than the far side of the rectangle, and testing to see if the Y-coordinate of the Turtle is greater than the lower side of the rectangle but less than the top of the rectangle.
  - This amounts to four tests.
- If we use a circle, then all we have to do is to test to see if the Turtle position is less than the distance of the radius of the circle from the center of the circle.
  - This is one test.
  - Note that we have changed the name of TARGET_WIDTH to TARGET_RADIUS.
- The only trick here is to note that the center of the circle is located at:
  - X-Coord_Center_Of_Circle = TARGET_LLEFT_X
  - Y-Coord_Center_Of_Circle = TARGET_LLEFT_Y + TARGET_RADIUS



The reason for the Y coordinate of the center of the circle being: TARGET_LLEFT_Y + TARGET_RADIUS is shown at left.

- The X coordinate remains the same: TARGET_LLEFT_X

Introducing the **turtle_distance()** Method

- **turtle_distance(x, y)** will tell you the distance between where the turtle currently is, and where the coordinates x and y are.
- This can be used to tell you how far the turtle is from the center of the Target.
- If the turtle position is less than the **TARGET_RADIUS** from the center of the target, then you hit the target!

Lab03C

1. Given this change, the if statement in lines 57 – 63 now reduces down to:

```
if(turtle.distance(TARGET_LLEFT_X, TARGET_LLEFT_Y + TARGET_RADIUS) <= TARGET_RADIUS):
        print('Target hit!')
else:
        print('You missed the target.')
```

2. In addition, delete those statements, (lines 26 – 33), which draw the rectangle, and replace them with the single statement:

```
turtle.circle(TARGET_RADIUS)
```

3. You will also need to change line 9 so that it no longer says TARGET_WIDTH, but now says: TARGET_RADIUS = 25

Lab03C

- Make the changes indicated.
- Then find the angle and the force necessary to hit the target.
- Call the instructor over to give you proper credit for your work.

Appendix:
Tip For Changing Python Console Width

- You may want to change the Python Console Width so that it doesn't end up covering the graphics window every time you run it.

Changing the Console Width to 40

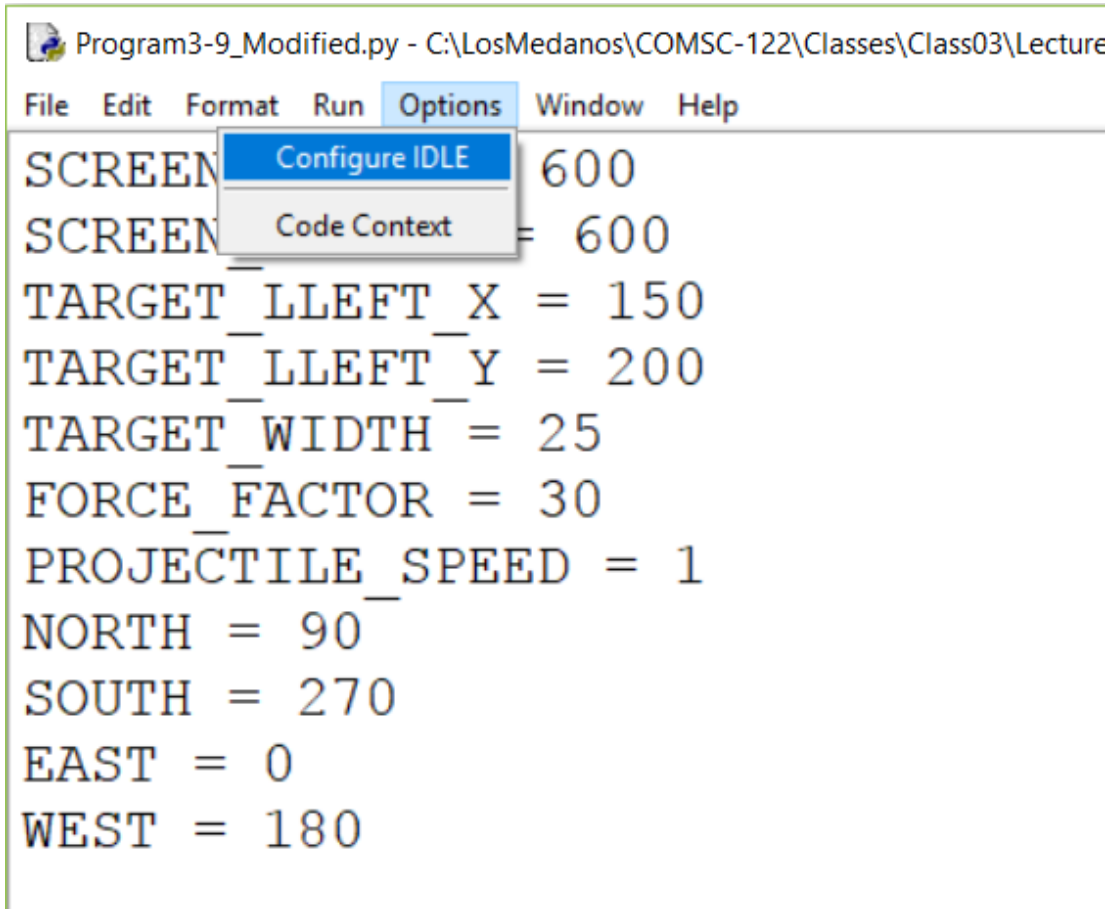
Program3-9_Modified.py - C:\LosMedanos\COMSC-122\Classes\Class03\Lecture2\

File Edit Format Run Options Window Help

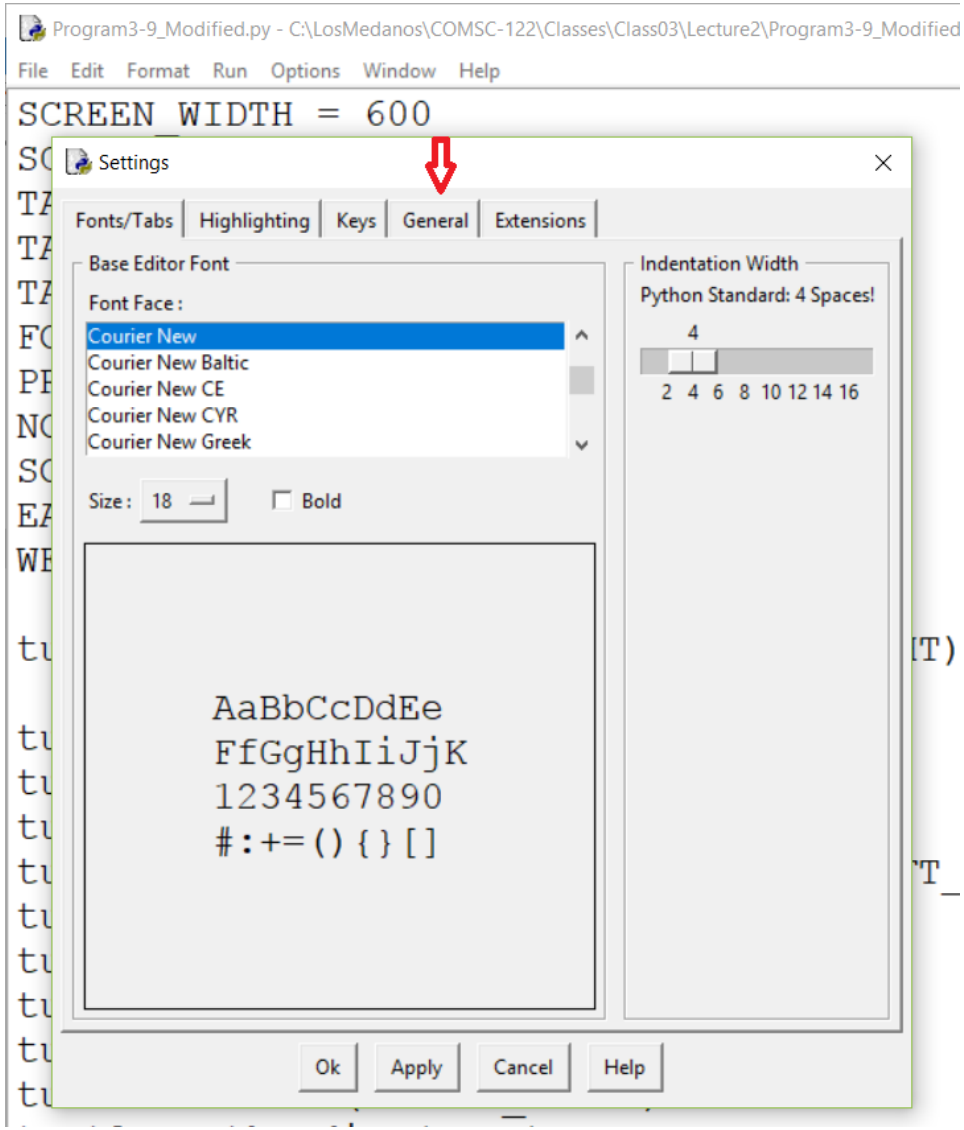
```
SCREEN_WIDTH = 600
SCREEN_HEIGHT = 600
TARGET_LLEFT_X = 150
TARGET_LLEFT_Y = 200
TARGET_WIDTH = 25
FORCE_FACTOR = 30
PROJECTILE_SPEED = 1
NORTH = 90
SOUTH = 270
EAST = 0
WEST = 180
```

- You will probably find it most helpful if you change the width of the console to 40.
- Otherwise the graphics window will be pretty much covered up by the console window.
- You do this by clicking on "**Options**" shown above.

Changing the Console Width to 40



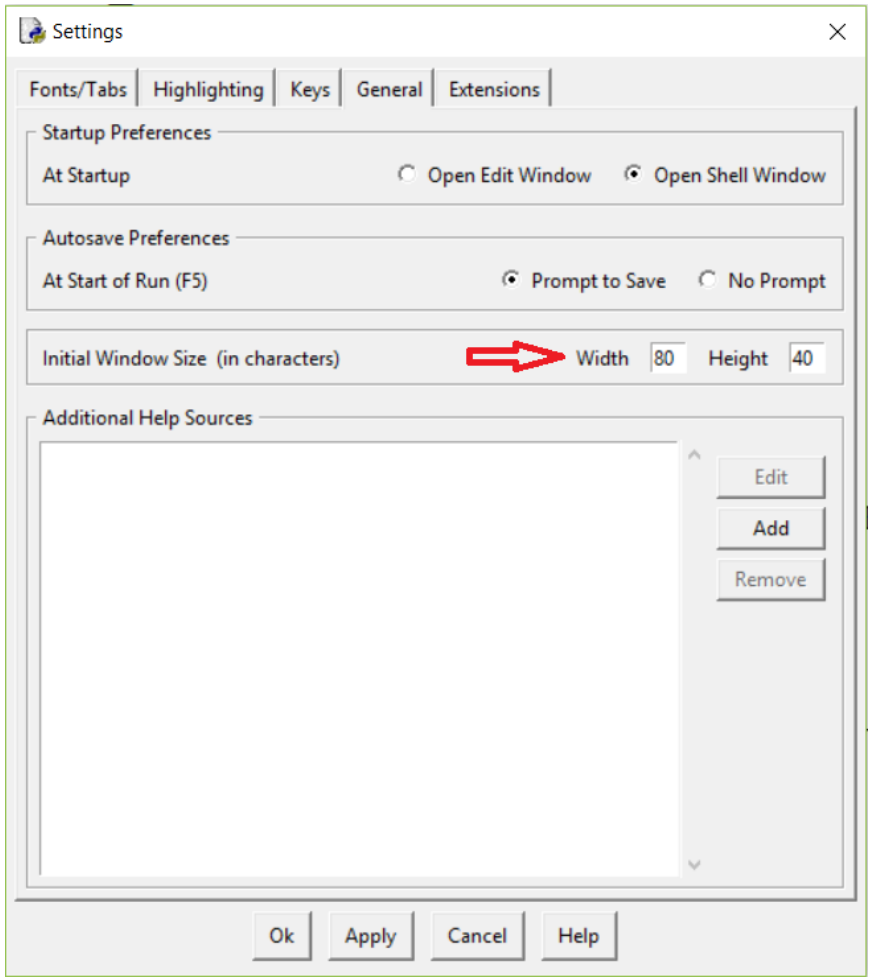
- Next click on **"Configure IDLE"**



Changing the Console Width to 40

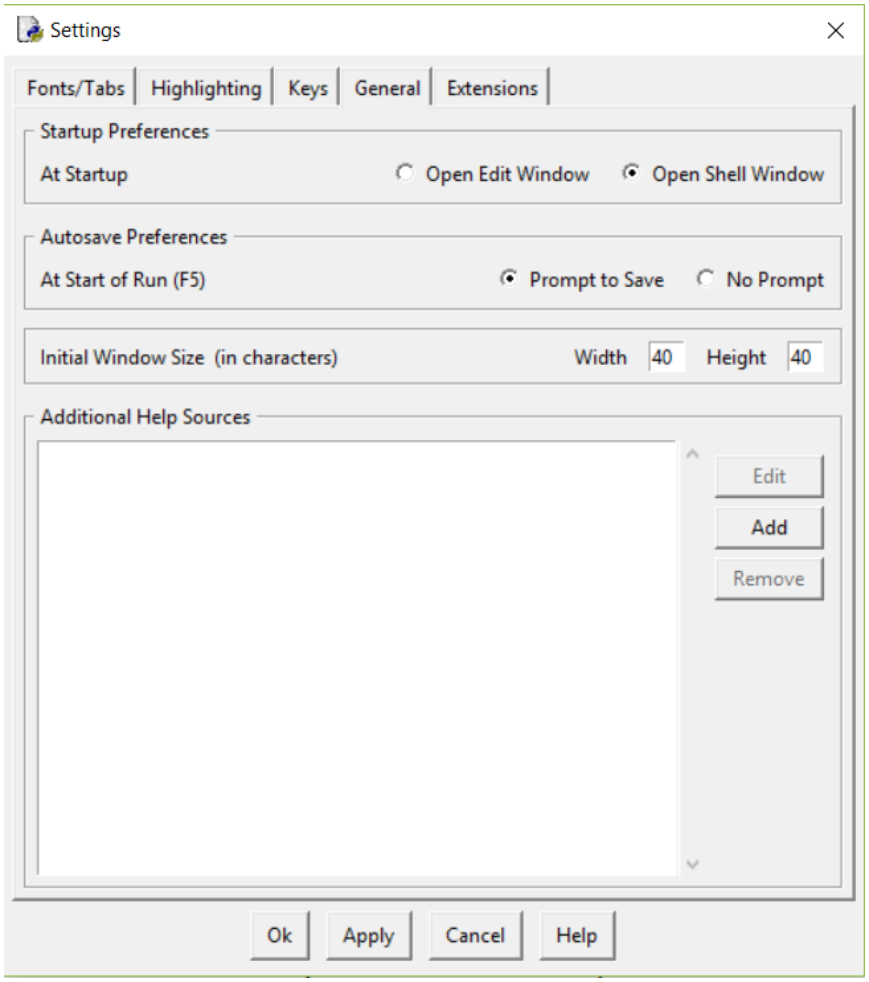
- Now click on: **“General”**

Changing the Console Width to 40



- Notice that the default Width is **“80”**.
- We will change that to “**40**”

Changing the Console Width to 40



- Once you have typed in "**40**", now click on "**Ok**", and your Python console will no longer cover the right hand side of the graphics Window when you run the program.