

Python & Turtle Graphics Tutorial GGC S3



Programming Skills Objectives:

- Identify and program Python code in JES (Jython Environment for Students) and save into their file area with proper extension. (ProgramName.py)
- Use following commands to draw shapes and designs with Turtles.
- Use comments in program to describe what code will do.
- Troubleshoot and solve coding errors and problems.
- Run code from written programs and from program and command area.
- Define and use Python functions.
- Use the "for i in range(x)" commands to do repeats.

Opening JES Software

- Click JES icon on the desktop to open the JES software.



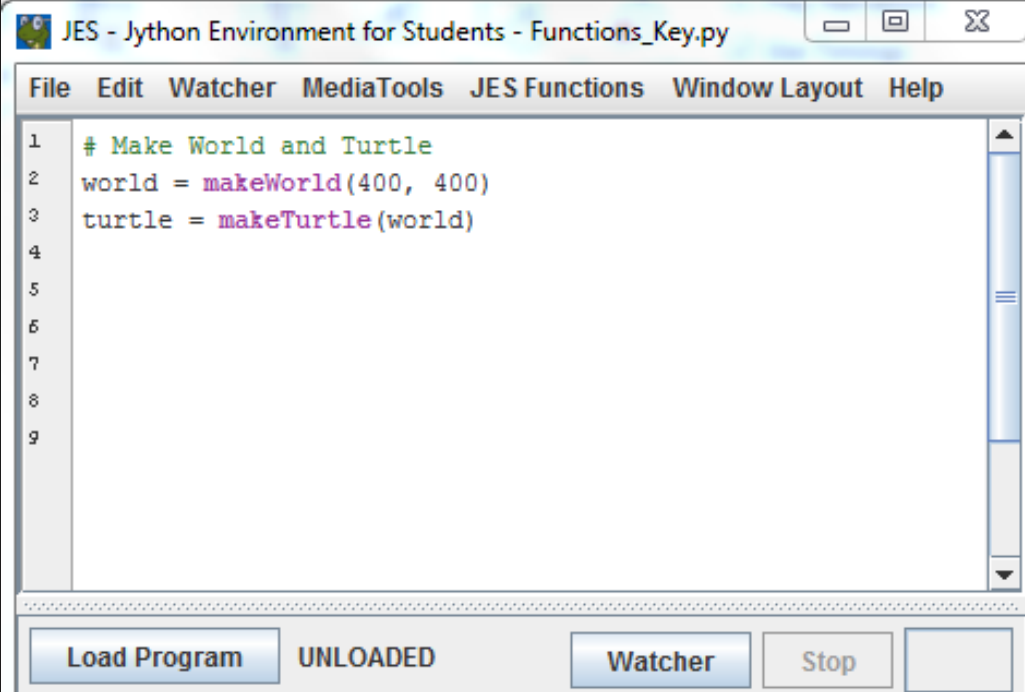
Tutorial 1 Overview:

Explore drawing a Square and a Rectangle with the Python Turtle Commands in JES

- In this tutorial, you will practice using the **Command Area** to test and draw a **Square** and a **Rectangle**.
- Use the up arrow key to show previous commands you made in the **Command Area** to make programming faster.
- Make sure you load the World and Turtle program you create **FIRST** before using the Command Area to draw your shapes.
- Modify any Turtle commands you typed in the **Program Area** in order to create the shape you want.
- Turtle Commands used in this Tutorial:
 - `turtle.forward()`
 - `turtle.turn()`
 - `turtle.penUp()`
 - `turtle.penDown()`
 - `turtle.moveTo(x, y)`
- Move on to the next slide to begin typing!

Create World and Turtle

- First step is to create a World and a Turtle.
- In the **Program Area** of the JES editor, type in the following code to create a world and a turtle:
- `world = makeWorld(400, 400)`
- `turtle = makeTurtle(world)`
- Double check your code with the picture to the right.



The screenshot shows the JES editor window titled "JES - Jython Environment for Students - Functions_Key.py". The menu bar includes "File", "Edit", "Watcher", "MediaTools", "JES Functions", "Window Layout", and "Help". The code editor contains the following Python code:

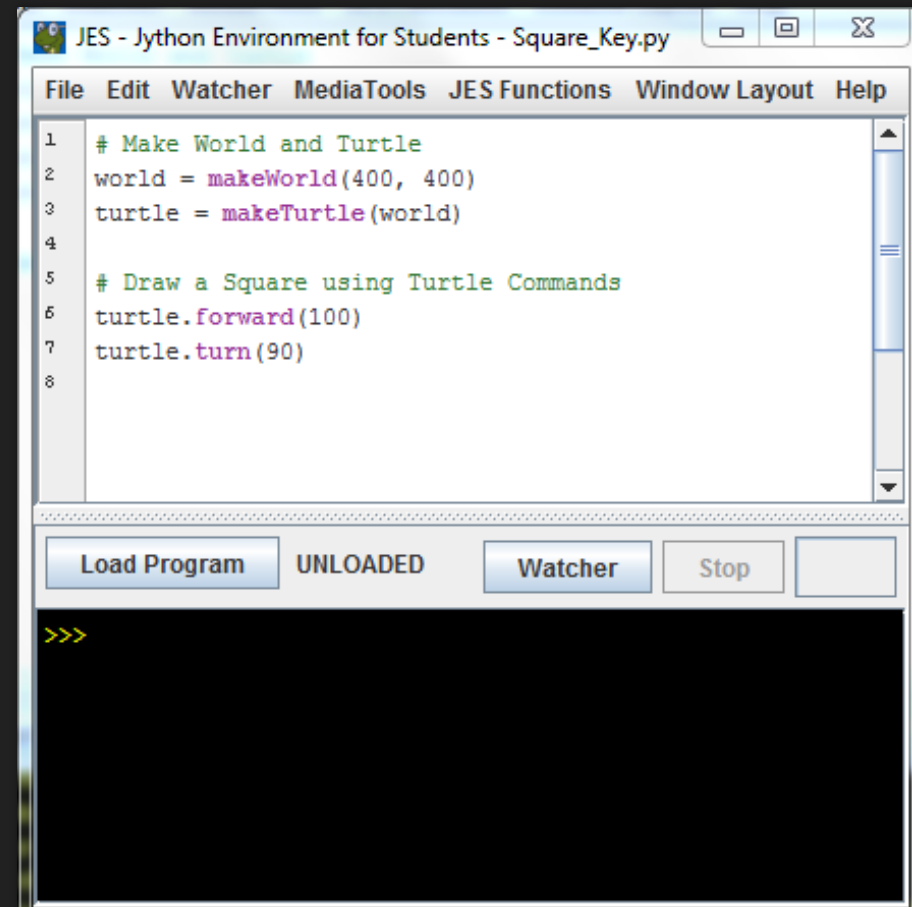
```
1 # Make World and Turtle
2 world = makeWorld(400, 400)
3 turtle = makeTurtle(world)
4
5
6
7
8
9
```

At the bottom of the window, there is a control bar with a "Load Program" button, a status indicator "UNLOADED", a "Watcher" button, and a "Stop" button.

Program commands to make the Turtle move

- Type the following commands below your World code to make your turtle move:
- `turtle.forward(100)`
- `turtle.turn(90)`

- Optional: Create a comment for your program.



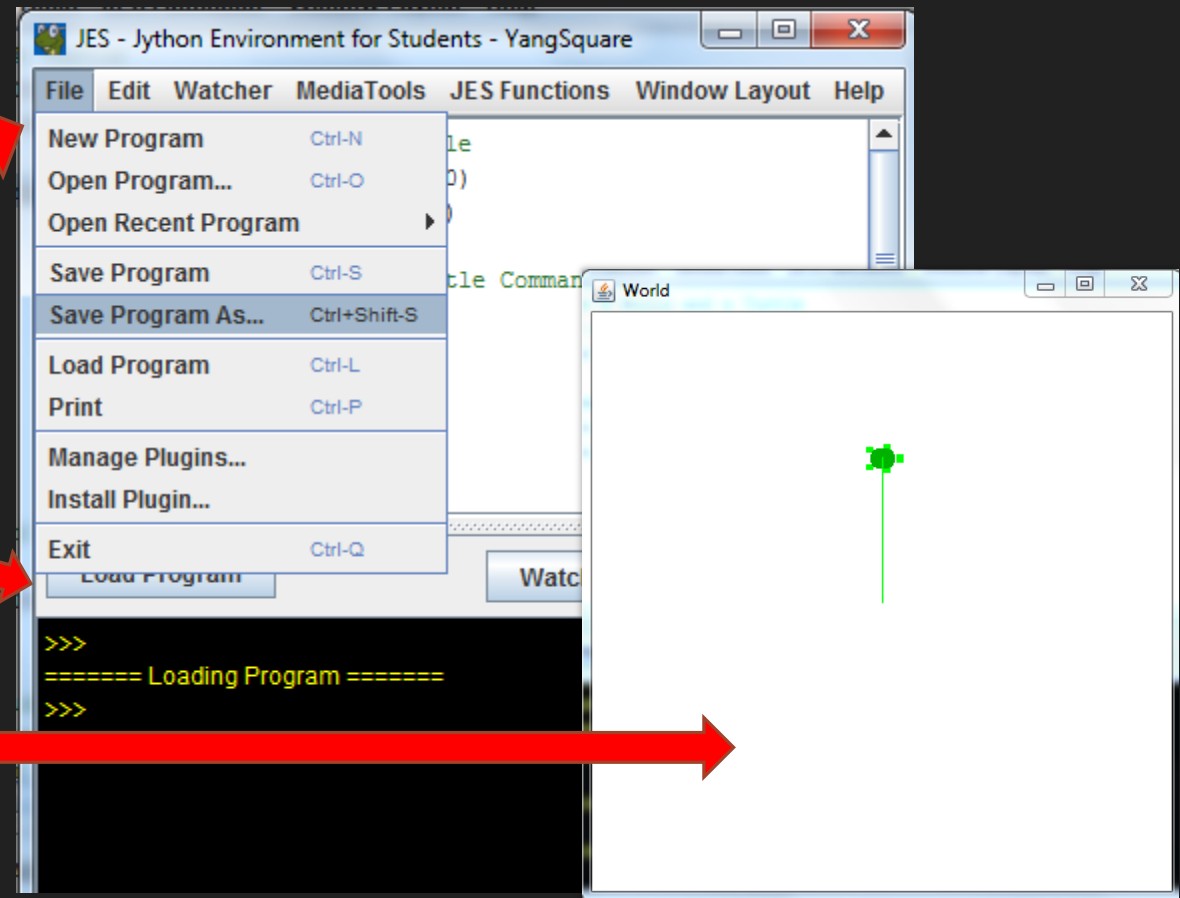
The screenshot shows the JES interface with a Python script in the editor. The script is as follows:

```
1 # Make World and Turtle
2 world = makeWorld(400, 400)
3 turtle = makeTurtle(world)
4
5 # Draw a Square using Turtle Commands
6 turtle.forward(100)
7 turtle.turn(90)
8
```

Below the editor, there are buttons for 'Load Program', 'UNLOADED', 'Watcher', and 'Stop'. At the bottom, there is a black console area with a yellow prompt '>>>>'.

Save and Load Program

- Save file as “Lastname_Square.py” inside the Python Turtles folder that is on the desktop.
- Click File → Save Program AS → Desktop → Python & Turtle Graphics → Save
- Click “Load Program” to create the World.
- A new window will open up and you should have a turtle in your window.



Using Command Area to test and draw your square

- The JES Command Area allows you to test your codes right away.
- Use the Command Area to test and find a correct path for your turtle to draw your square.
- Only use the following two turtle commands:
 - `turtle.forward(100)`
 - `turtle.turn(90)`
- Having Trouble?: Hit the space bar to reveal the remaining commands your turtle needs to draw a square.

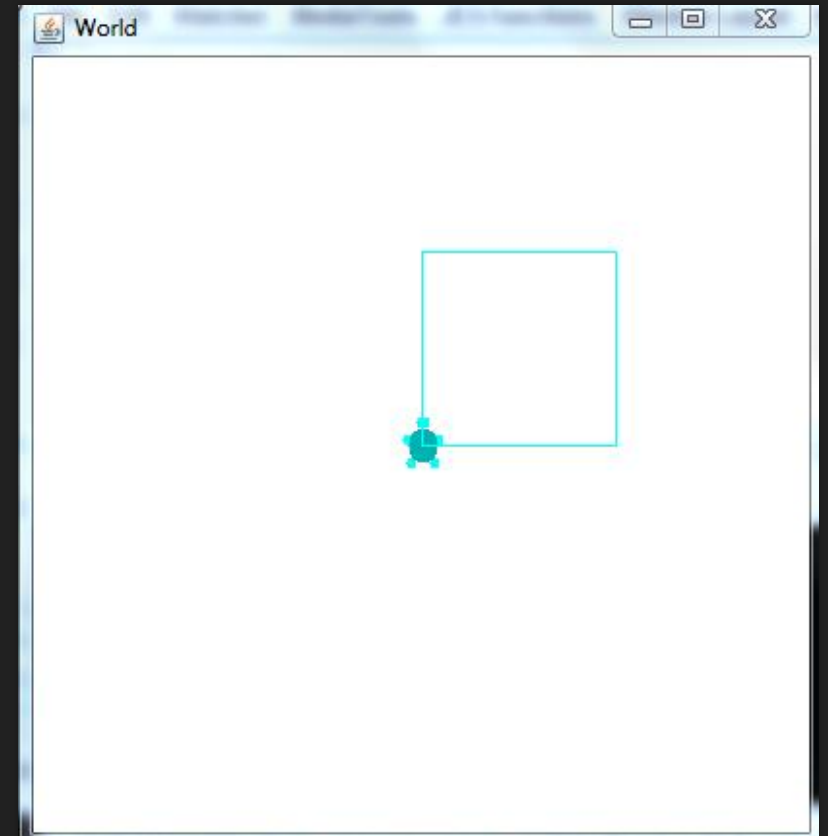


Turtle Square

- Now that you have figured out the path to create your square, type the correct code into the Program Area.
- Click “Load Program” and you should now have a Square!

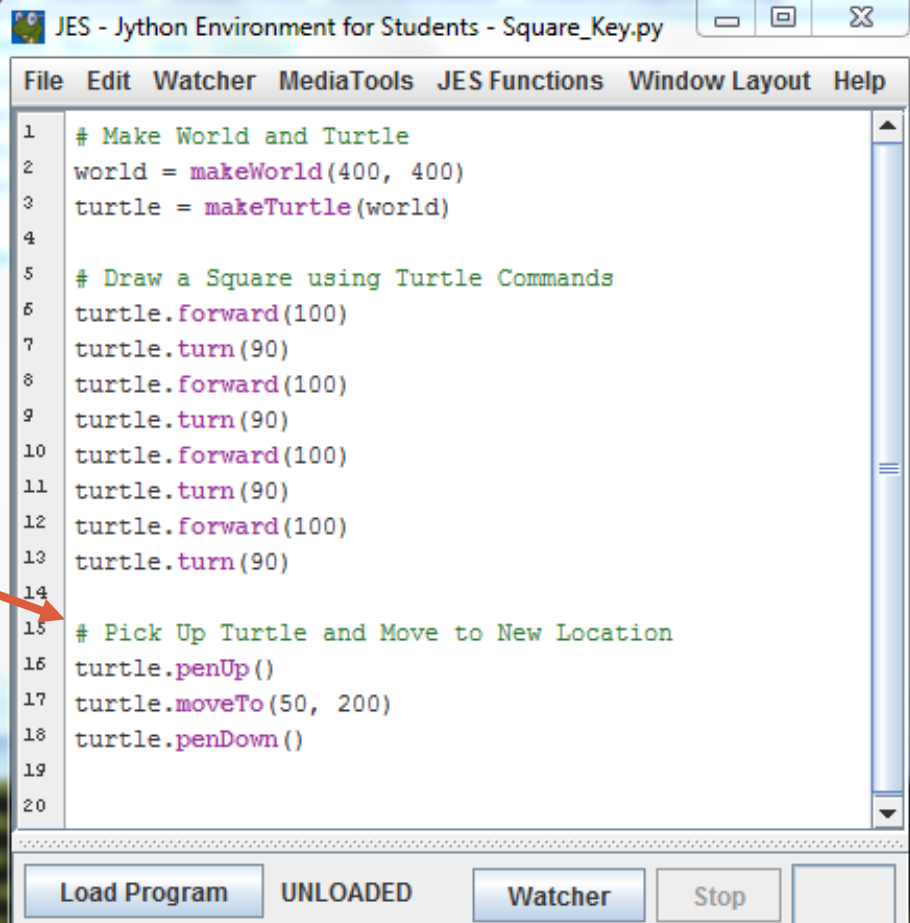
```
JES - Jython Environment for Students - Square_Key.py
File Edit Watcher MediaTools JES Functions Window Layout Help
1 # Make World and Turtle
2 world = makeWorld(400, 400)
3 turtle = makeTurtle(world)
4
5 # Draw a Square using Turtle Commands
6 turtle.forward(100)
7 turtle.turn(90)
8 turtle.forward(100)
9 turtle.turn(90)
10 turtle.forward(100)
11 turtle.turn(90)
12 turtle.forward(100)
13 turtle.turn(90)
14
15
```

Load Program UNLOADED Watcher Stop



Pick Up and Move Turtle

- Next step is to have the Turtle draw a Rectangle in the same World.
- But before you do that, the turtle needs to be moved to a new location.
- Use the `penUp()`, `moveTo()`, and `penDown()` to accomplish this task.
- Type in the code you see in the image to the right under your Square Turtle code.
- Option: Use Comments to make your program easy to read and for you to understand.

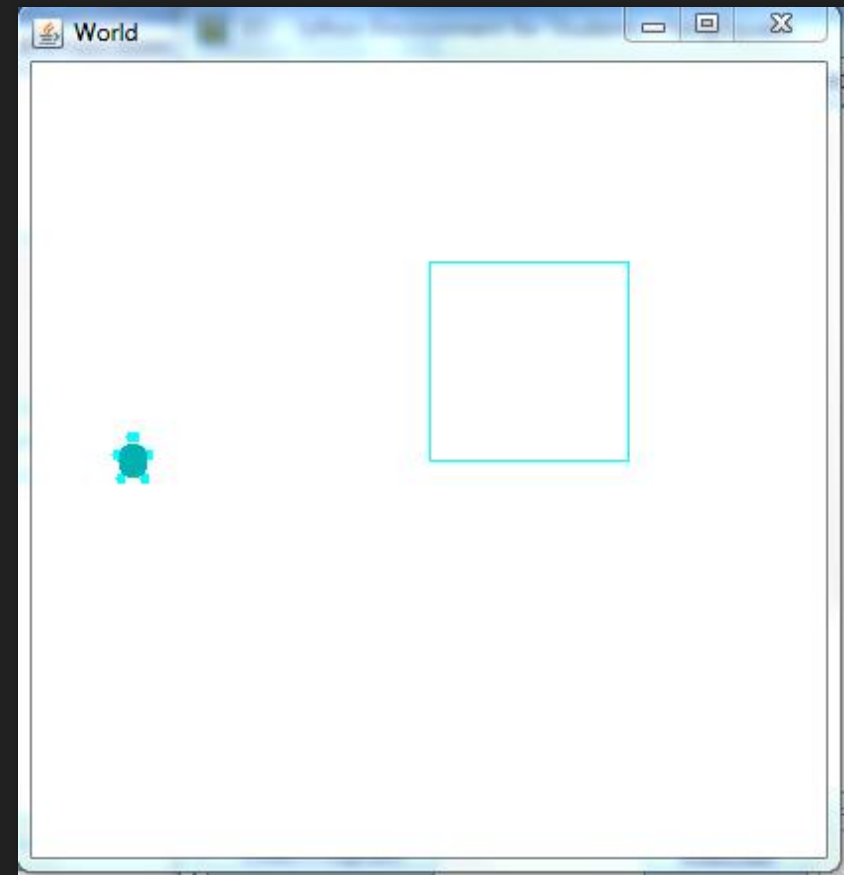


```
JES - Jython Environment for Students - Square_Key.py
File Edit Watcher MediaTools JES Functions Window Layout Help
1 # Make World and Turtle
2 world = makeWorld(400, 400)
3 turtle = makeTurtle(world)
4
5 # Draw a Square using Turtle Commands
6 turtle.forward(100)
7 turtle.turn(90)
8 turtle.forward(100)
9 turtle.turn(90)
10 turtle.forward(100)
11 turtle.turn(90)
12 turtle.forward(100)
13 turtle.turn(90)
14
15 # Pick Up Turtle and Move to New Location
16 turtle.penUp()
17 turtle.moveTo(50, 200)
18 turtle.penDown()
19
20
```

Buttons: Load Program UNLOADED Watcher Stop

Pick Up and Move Turtle

- Save and then click “Load Program”.
- Your Turtle should now be in a new location.
- **Note: Multiple JES windows will open when you click “Load Program”. Be sure to close the windows when you are finish with each tutorial. This avoids slowing down the computer.**



STOP! Time to Switch Roles

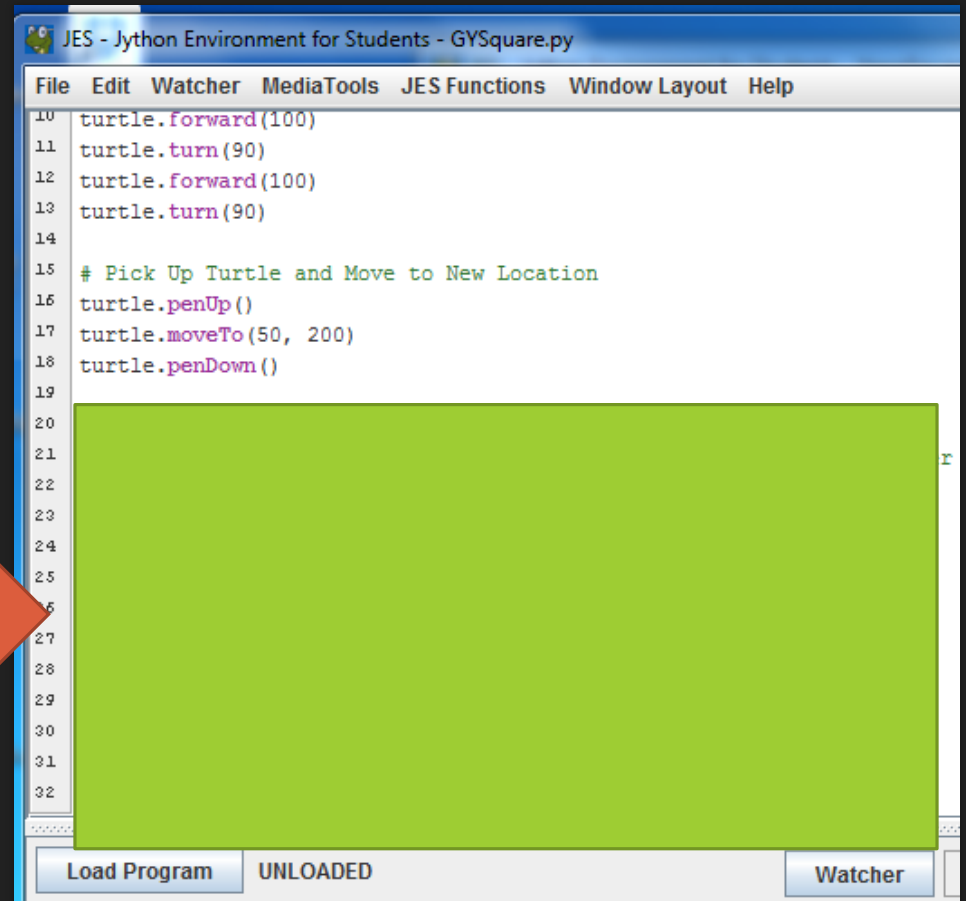
- If you need help, ask for assistance.
- Be sure SWITCH ROLES before moving on to the next tutorial!



Draw a Rectangle

- Squares and Rectangles are very similar.
- What can we modify or use in the Square Turtle commands to draw a Rectangle?
- Be sure to switch positions and have the other partner draw the Rectangle.
- Use the Command Area to test your code out, then type in the correct code into the Program Area.

○

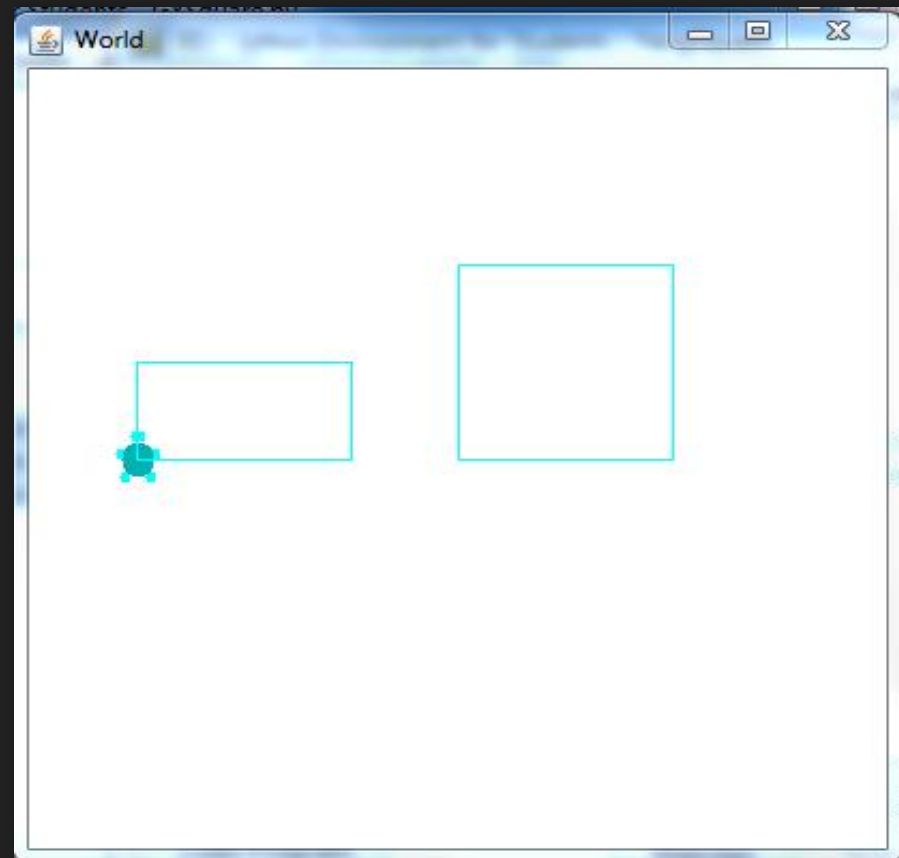


```
JES - Jython Environment for Students - GYSquare.py
File Edit Watcher MediaTools JES Functions Window Layout Help
10 turtle.forward(100)
11 turtle.turn(90)
12 turtle.forward(100)
13 turtle.turn(90)
14
15 # Pick Up Turtle and Move to New Location
16 turtle.penUp()
17 turtle.moveTo(50, 200)
18 turtle.penDown()
19
20
21
22
23
24
25
26
27
28
29
30
31
32
```

Load Program UNLOADED Watcher

Draw a Rectangle

- Save and then click “Load Program”.
- You should now have a Rectangle and a Square in your World.

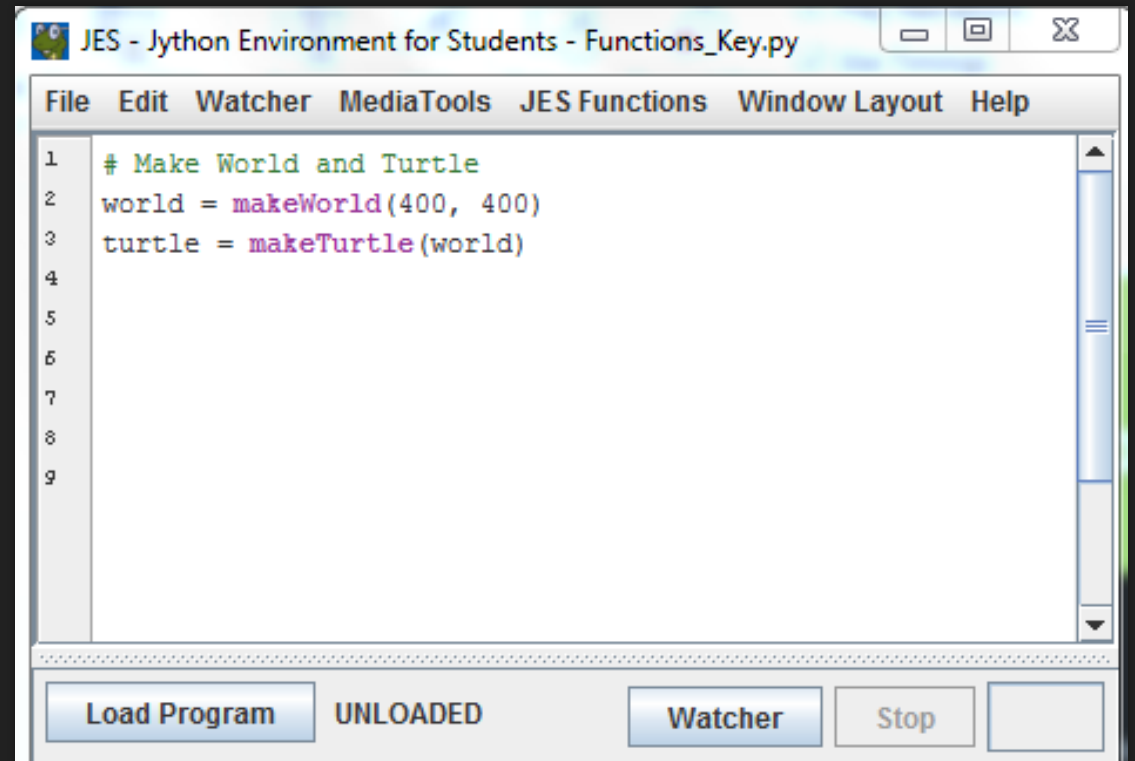


Tutorial 2: Creating Python Functions for Shapes and Designs in JES

- Remember using the Command Area is good for short experiments.
- Longer programs need to be coded and then saved in the Program Area.
- You can achieve this by placing your Turtle commands in Functions.
- Save your “LastName_Square.py” program. We will create a new JES program file to use for Tutorial 2 through 4.
- Go to File → New Program → Save Program As → Desktop → Python & Turtle Graphics → Save File Name as “LastName_Functions.py”.
- **Reminder: Multiple JES windows will open when you click “Load Program”. Be sure to close the windows when you are finish with each tutorial. This avoids slowing down the computer.**

Create a World and a Turtle

- Begin by creating your World and Turtle In the **Program Area** of the JES editor, type in the following code below:
- `world = makeWorld(400, 400)`
- `turtle = makeTurtle(world)`
- Double check your code with the picture to the right.



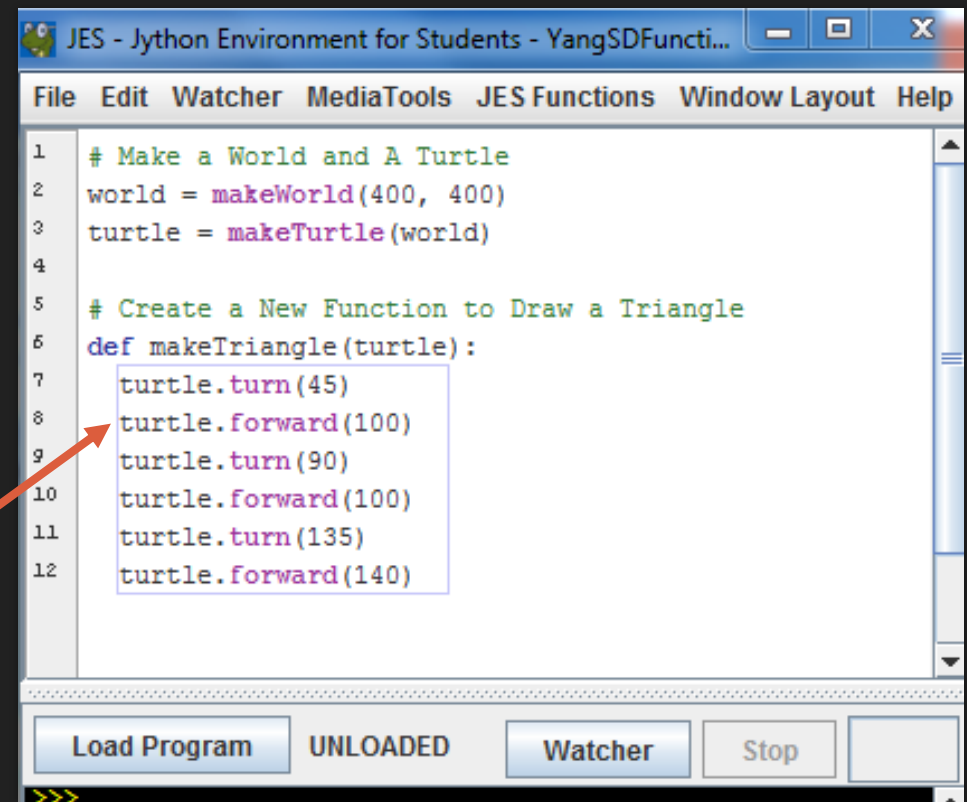
The screenshot shows the JES editor window titled "JES - Jython Environment for Students - Functions_Key.py". The menu bar includes "File", "Edit", "Watcher", "MediaTools", "JES Functions", "Window Layout", and "Help". The code editor contains the following Python code:

```
1 # Make World and Turtle
2 world = makeWorld(400, 400)
3 turtle = makeTurtle(world)
4
5
6
7
8
9
```

At the bottom of the window, there are several buttons: "Load Program", "UNLOADED", "Watcher", "Stop", and an empty button.

Create a Function to Draw a Triangle

- Begin the function with **def**
 - **def** keyword means you are creating a new command for your turtle.
- Then, type the name of the function, and the input “turtle” between parentheses:
makeTriangle(turtle)
- End the line with a colon (“ : ”)
- Next, type in the following turtle commands you see in the image to right to create a Triangle. These commands makes up the *body* of the function.
- The *body* of the function needs to be indented (Hint: Use two spaces).

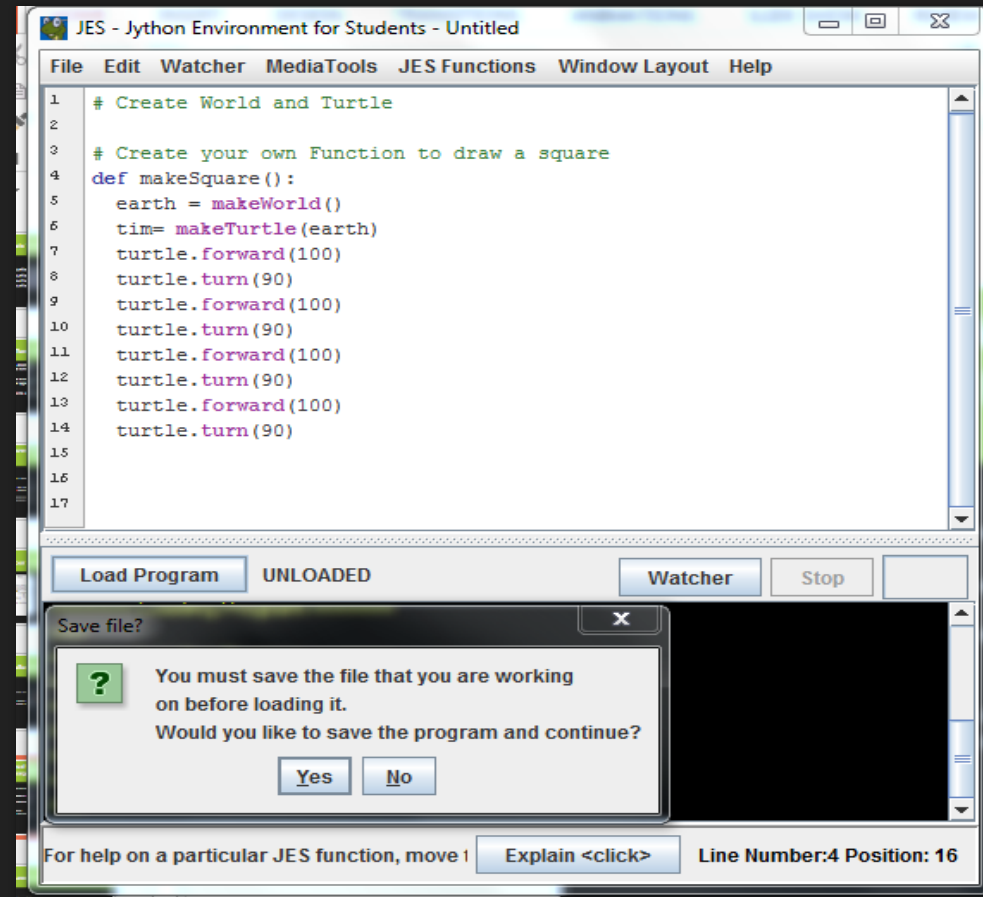


```
JES - Jython Environment for Students - YangSDFuncti...
File Edit Watcher MediaTools JES Functions Window Layout Help
1 # Make a World and A Turtle
2 world = makeWorld(400, 400)
3 turtle = makeTurtle(world)
4
5 # Create a New Function to Draw a Triangle
6 def makeTriangle(turtle):
7     turtle.turn(45)
8     turtle.forward(100)
9     turtle.turn(90)
10    turtle.forward(100)
11    turtle.turn(135)
12    turtle.forward(140)

Load Program UNLOADED Watcher Stop
```

REMEMBER: The Most Common JES Issue is Forgetting to Load

- Your function does **NOT** exist for JES until you *load* it
 - Before you load it, the program is just a bunch of characters.
 - Loading *encodes* it as an executable function
- Save and Save As
 - You must Save before Loading
 - You must Load before you can use your function

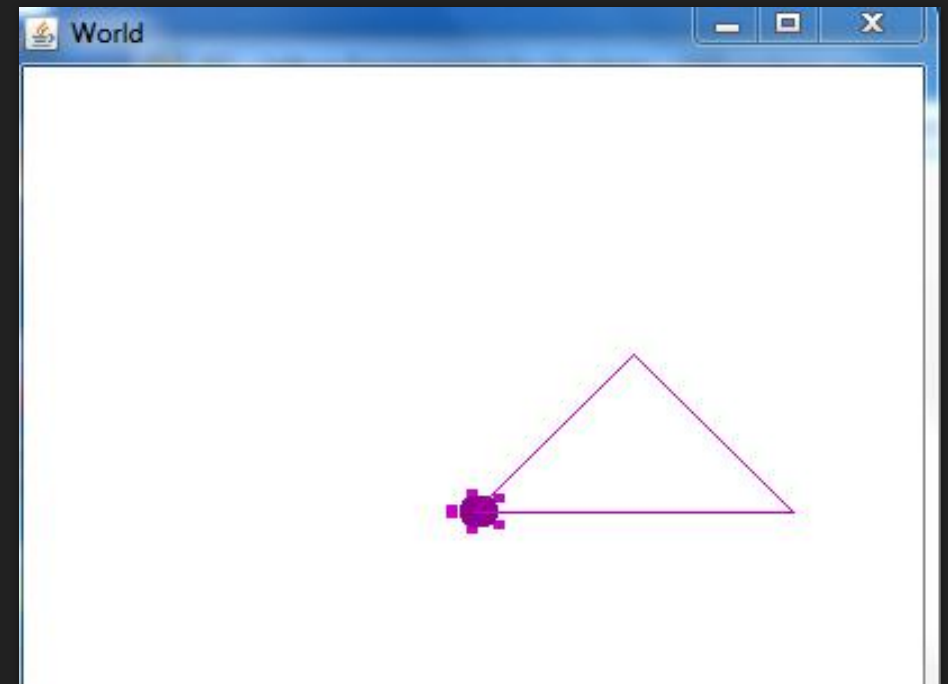


An “Unloaded” function doesn’t exist yet.

Call Triangle Function

- Save and then click “Load Program”.
- A World window with a Turtle will open.
- Now, call your **makeTriangle(turtle)** function by typing the function name in the Command Area.
- Then hit the enter key.
- This calling will allow the Turtle to draw a Triangle in the World.

```
>>>  
===== Loading Program =====  
>>> makeTriangle(turtle)  
>>>
```



Congrats! You have successfully created a Python Function!

STOP! Time to Switch Roles

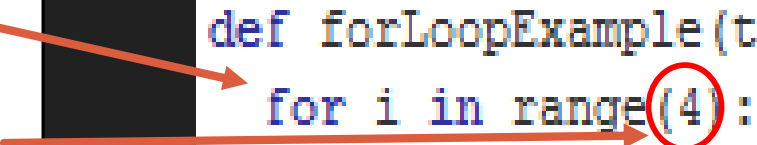
- If you need help, ask for assistance.
- Be sure SWITCH ROLES before moving on to the next tutorial!
- **Reminder: Multiple JES windows will open when you click “Load Program”. Be sure to close the windows when you are finish with each tutorial. This avoids slowing down the computer.**



Tutorial 3: Introducing the For Loop- Using a For Loop in a Function

- **Loops** are simply parts of a program that run over and over again to reduce the amount of times you type the same command.
- The phrase “**for i in range**” sets the number of times you want to run a command or a series of commands.
- You specify how many times you want to run the turtle commands to repeat inside the parenthesis of the range function.
- **TIP: In JES, a For Loop must end with a colon (:)** and be indented by two spaces in a Function.

```
def forLoopExample(turtle):  
    for i in range(4):  
        turtle.forward(100)
```



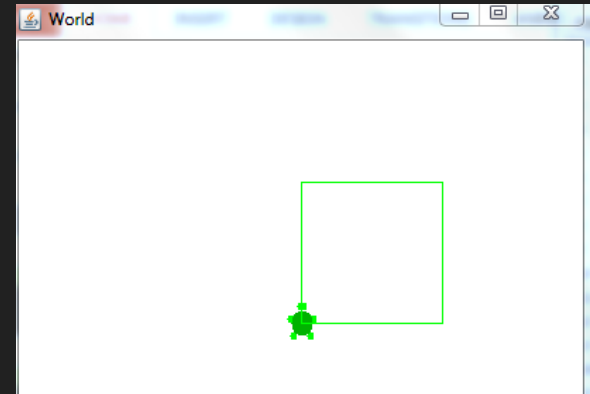
betterSquare(turtle) Function

- Begin by defining the **betterSquare(turtle)** function under your **makeTriangle(turtle)** function.
- Then type in the for loop and turtle commands you see in the image.
- Be sure to indent the for loop and your turtle commands.
- Next, save and click “**Load Program**”
- Now call your function by typing **betterSquare(turtle)** into the Command Area.
- Be sure to hit the enter key to complete the call of your function.

```
6 def makeTriangle(turtle):
7     turtle.turn(45)
8     turtle.forward(100)
9     turtle.turn(90)
10    turtle.forward(100)
11    turtle.turn(135)
12    turtle.forward(140)
13
14    #Introducing the For Loop:
15    # Better Square Function-
16    def betterSquare(turtle):
17        for i in range(4):
18            turtle.forward(100)
19            turtle.turn(90)
20
```

Load Program

```
===== Loading Program =====
>>>  betterSquare(turtle) 
>>>
```



betterSquare(turtle) Function Review

- Remember the two commands to draw a square
 - `turtle.forward(100)`
 - `Turtle.turn(90)`
- See how you can use a for loop to minimize the lines of code needed to draw a square.
- Notice how the for loop tells the turtle to make 4 rounds of the `forward(100)` and `turn(90)` command to draw the square.

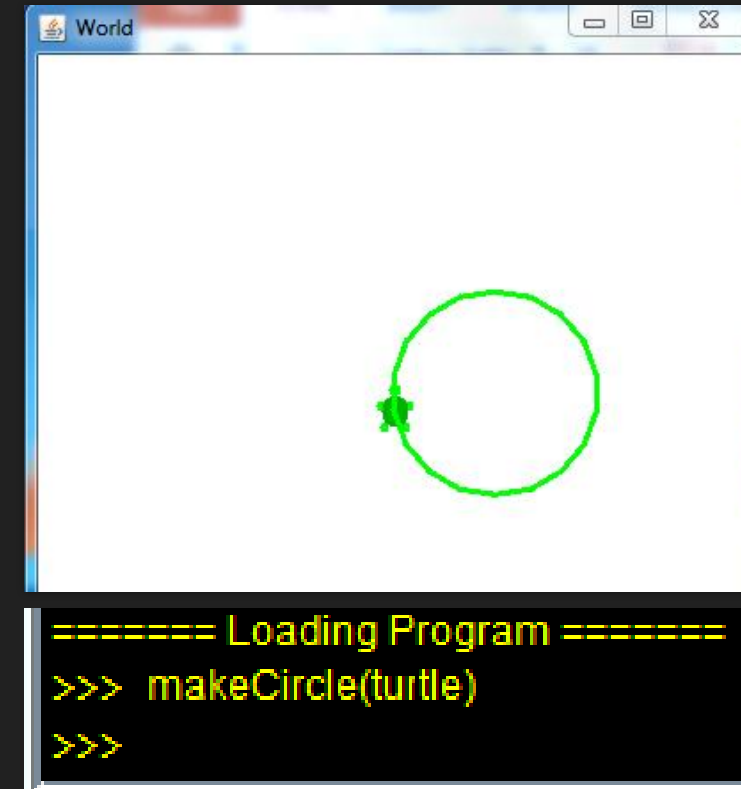
```
JES - Jython Environment for Students - YangSquare
File Edit Watcher MediaTools JES Functions Window Layout Help
1 # Make a World and A Turtle
2 world = makeWorld(400, 400)
3 turtle = makeTurtle(world)
4
5 # Draw a Square using Turtle Commands
6 turtle.forward(100)
7 turtle.turn(90)
8 turtle.forward(100)
9 turtle.turn(90)
10 turtle.forward(100)
11 turtle.turn(90)
12 turtle.forward(100)
13 turtle.turn(90)
14
```

```
13
14 #Introducing the For Loop:
15 # Better Square Function-
16 def betterSquare(turtle):
17     for i in range(4):
18         turtle.forward(100)
19         turtle.turn(90)
20
Load Program
```


makeCircle(turtle) Function

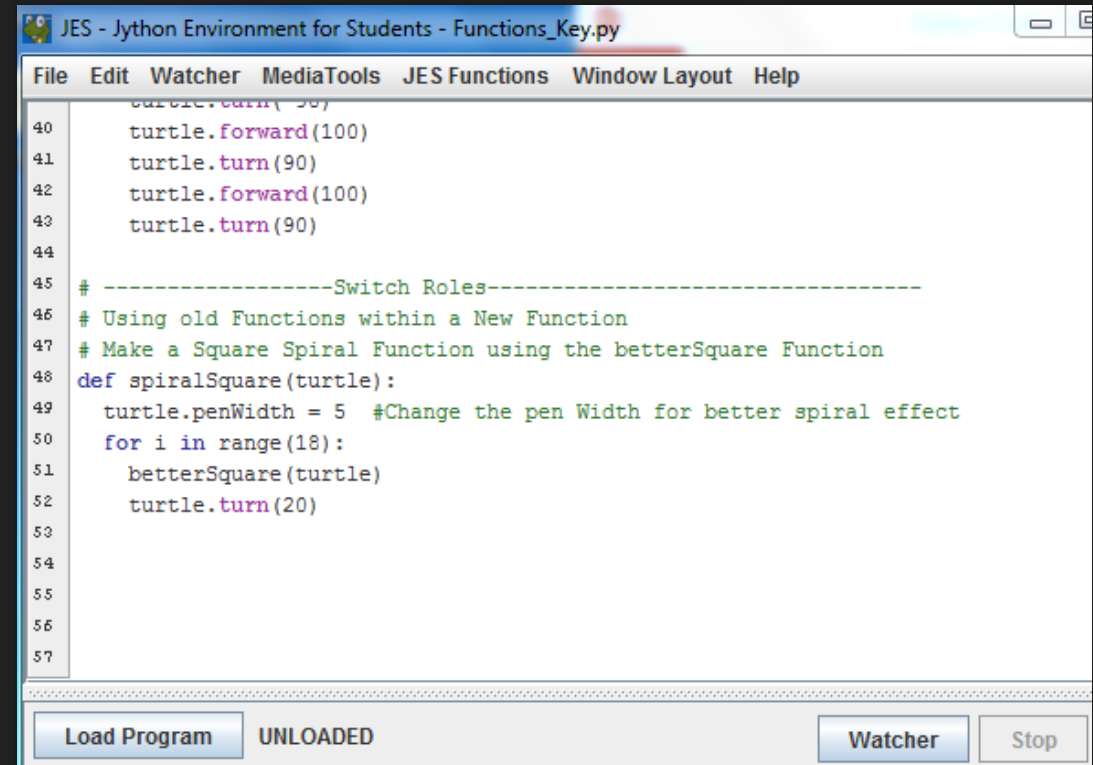
- Begin by defining the **makeCircle(turtle)** function.
- Then type in the for loop and turtle commands you see in the image.
- Be sure to indent the for loop and your turtle commands.
- Next, save and click “**Load Program**”.
- Now call your function by typing **makeCircle(turtle)** into the **Command Area**.
- This will allow the Turtle to draw a Circle.
- Notice how the for loop tells the Turtle to repeat the commands forward(20) and turn(20) 18 times to complete the Circle.

```
JES - Jython Environment for Students - Functions_Key.py
File Edit Watcher MediaTools JES Functions Win
8 turtle.forward(100)
9 turtle.turn(90)
10 turtle.forward(100)
11 turtle.turn(135)
12 turtle.forward(140)
13
14 #Introducing the For Loop:
15 # Better Square Function-
16 def betterSquare(turtle):
17     for i in range(4):
18         turtle.forward(100)
19         turtle.turn(90)
20
21 # Make a Circle using For Loop
22 def makeCircle(turtle):
23     turtle.penWidth = 3
24     for i in range(18):
25         turtle.forward(20)
26         turtle.turn(20)
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```



Tutorial 4: Using an Old Function within a New Function (Square)

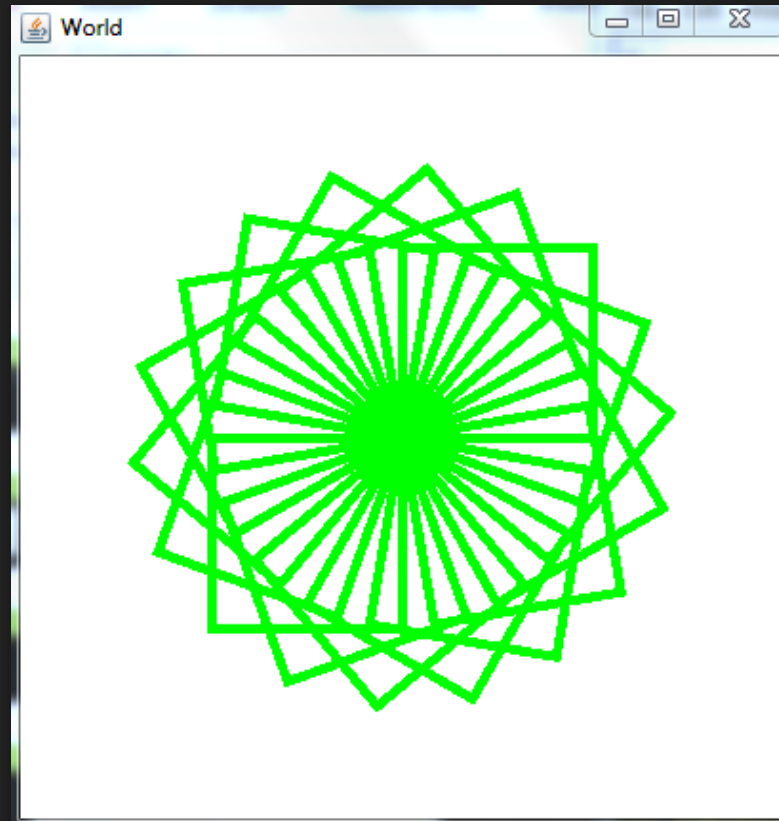
- Create a `spiralSquare(turtle)` Function by using the `betterSquare(turtle)` function.
- Re-using an old shape function within a new function allows your Python program to create cool designs!
- Type in the code you see in the image.
- Save and Load program!
- Be sure to call you **`spiralSquare(turtle)`** function in Command Area!



```
JES - Jython Environment for Students - Functions_Key.py
File Edit Watcher MediaTools JES Functions Window Layout Help
40 turtle.turn(90)
41 turtle.forward(100)
42 turtle.turn(90)
43 turtle.forward(100)
44 turtle.turn(90)
45 # -----Switch Roles-----
46 # Using old Functions within a New Function
47 # Make a Square Spiral Function using the betterSquare Function
48 def spiralSquare(turtle):
49     turtle.penWidth = 5 #Change the pen Width for better spiral effect
50     for i in range(18):
51         betterSquare(turtle)
52         turtle.turn(20)
53
54
55
56
57
Load Program UNLOADED Watcher Stop
```

YOUR RESULTS SHOULD LOOK LIKE THIS!

`spiralSquare(turtle)`



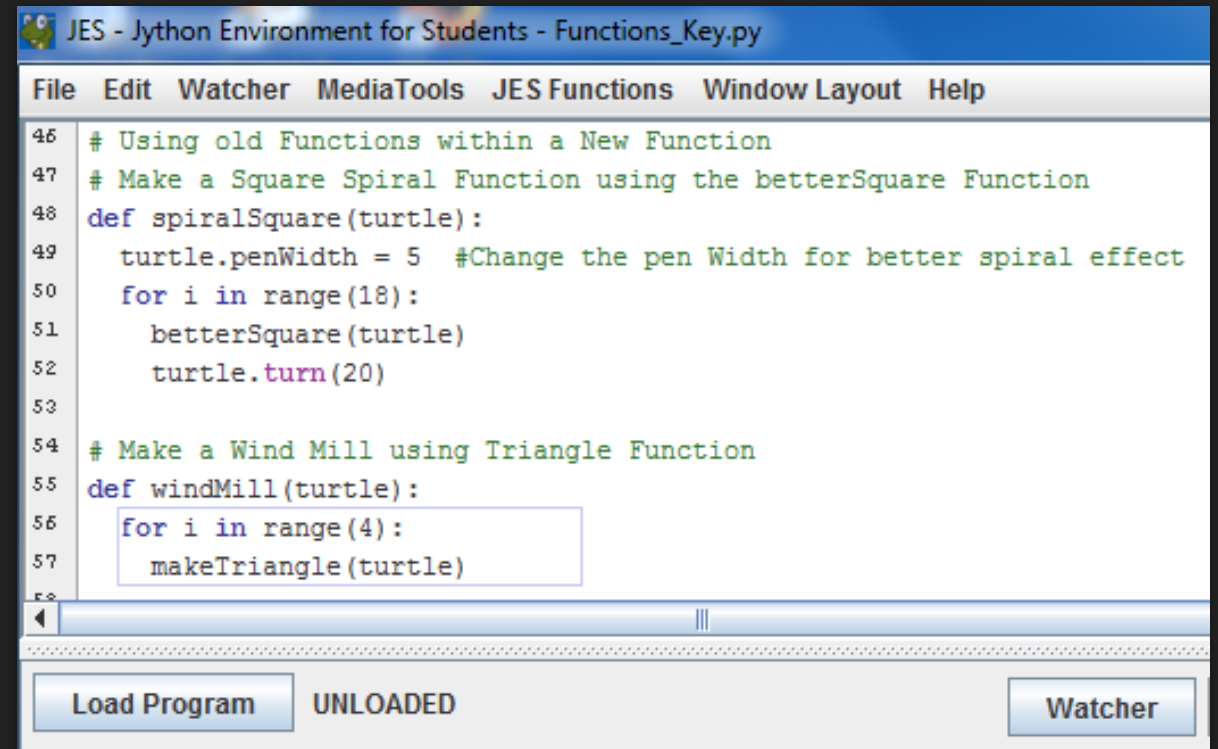
STOP! Time to Switch Roles

- If you need help, ask for assistance.
- Be sure SWITCH ROLES before moving on to the next tutorial!
- **Reminder: Multiple JES windows will open when you click “Load Program”. Be sure to close the windows when you are finish with each tutorial. This avoids slowing down the computer.**



Tutorial 4: Using an Old Function within a New Function (Triangle)

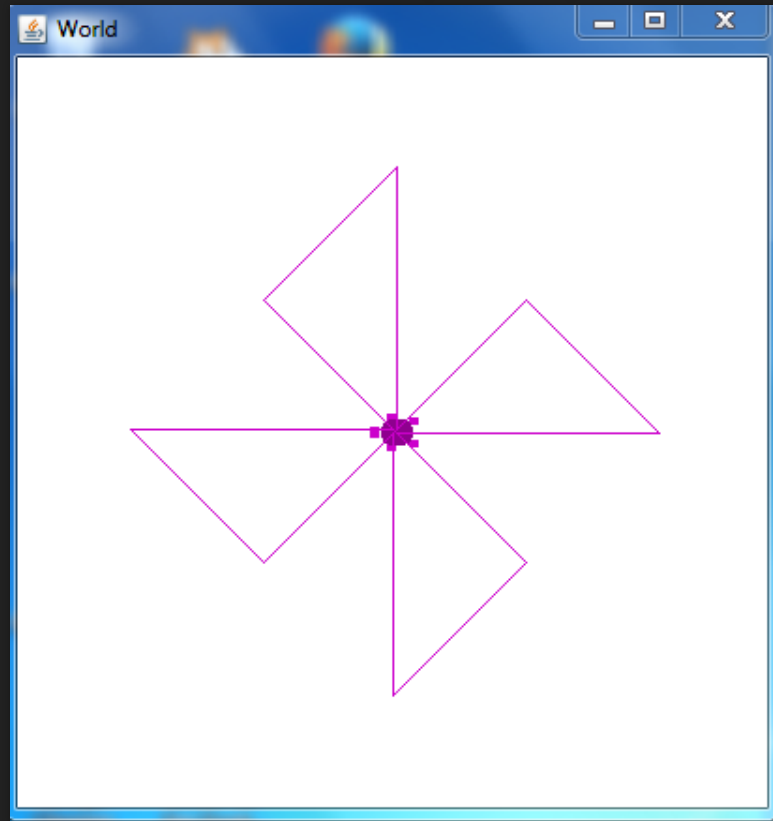
- Create a windMill(turtle) Function by using the makeTriangle(turtle) function.
- Re-using an old shape function within a new function allows your Python program to create cool designs!
- Type in the code you see in the image.
- Save and Load program!
- Be sure to call your **windMill(turtle)** function in Command Area!



```
JES - Jython Environment for Students - Functions_Key.py
File Edit Watcher MediaTools JES Functions Window Layout Help
46 # Using old Functions within a New Function
47 # Make a Square Spiral Function using the betterSquare Function
48 def spiralSquare(turtle):
49     turtle.penWidth = 5 #Change the pen Width for better spiral effect
50     for i in range(18):
51         betterSquare(turtle)
52         turtle.turn(20)
53
54 # Make a Wind Mill using Triangle Function
55 def windMill(turtle):
56     for i in range(4):
57         makeTriangle(turtle)
58
Load Program UNLOADED Watcher
```

YOUR RESULTS SHOULD LOOK LIKE THIS!

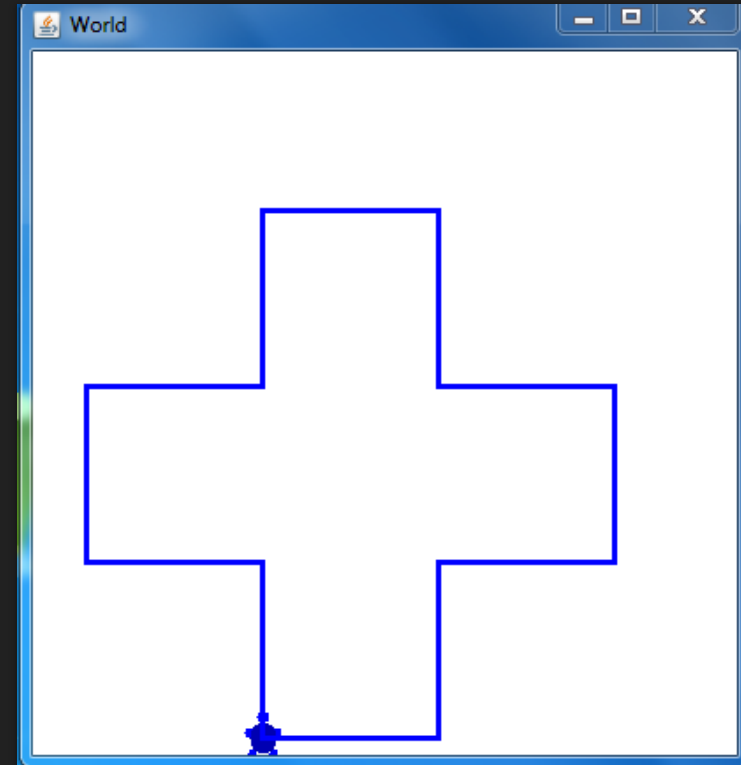
windmill(turtle)



Tutorial 5: Putting it All Together

Create a Plus Sign

- Do It Yourself!
- Combine all the Turtle Commands you've learned and create a Plus sign!
- Use Paper and Pencil if needed.
- Raise your hand if you need assistance!
- Make sure you save and "Load Program"
- Then call your function in Command Area.
- Try completing the task, before looking at the Hint.



Tutorial 5: Putting it All Together

Create a Plus Sign

Use any of the following turtle commands

- **turtle.forward(distance)**- command to move the turtle in the directions it's facing. Default distance is 100 pixels.
- **turtle.turn(degrees)**- turns the turtle in 90 degrees depending on the direction its facing.
- **turtle.moveTo(x, y)**- command to place the turtle in a new location in the world.
- **turtle.penUp()**- command to pick the pen up.
- **turtle.penDown()**- command to put the pen down again to draw.
- **turtle.penWidth = (1-10)** - allows you to change the size of the pen
- **turtle.color = red (blue, green . . .) -or- (255, 255, 255) RGB notation**- allows you to change the color of the pen and turtle.

makePlus(turtle) Function Hint Example

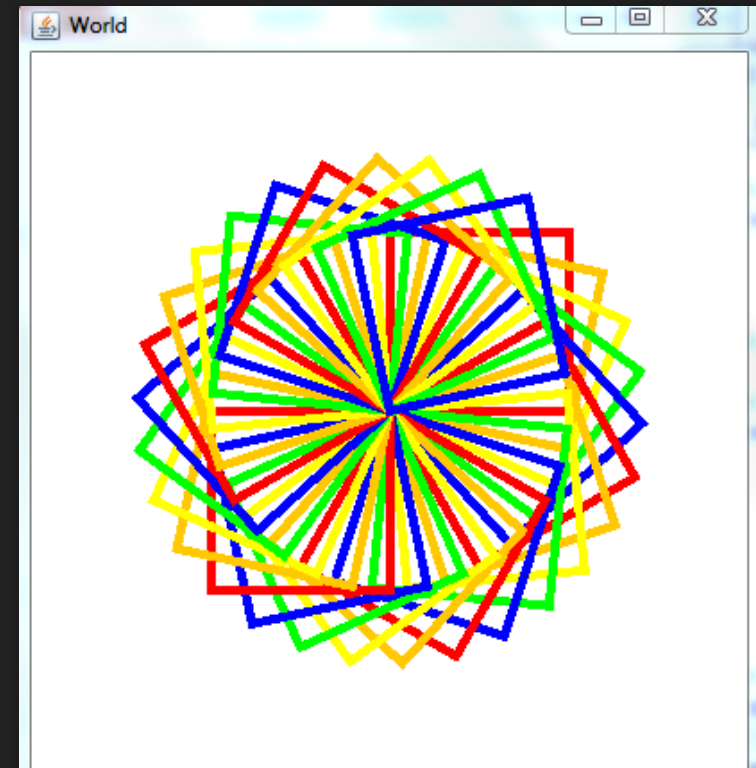


CONGRATULATIONS!

- You have successfully completed the Python Turtles Tutorial.
- BONUS: Do more with your spiral and windmill designs by adding colors!
- CHALLENGE: Create your initials using the JES and Turtle commands!

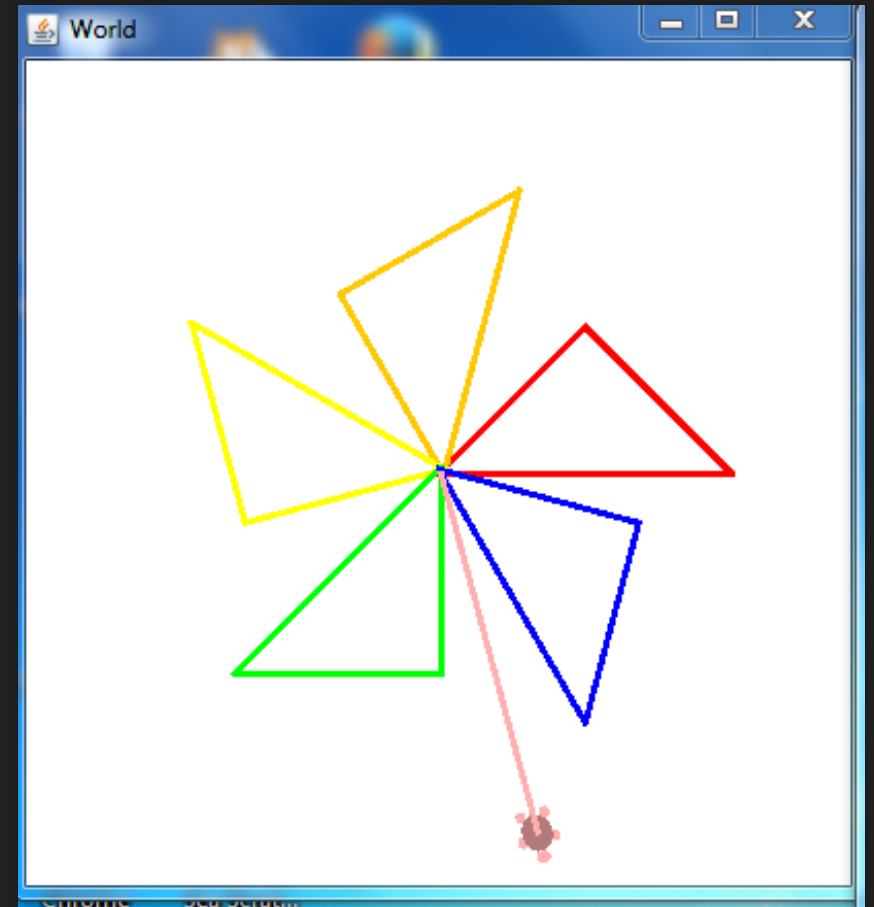
BONUS: Doing More with Turtle Colors!

```
JES - Jython Environment for Students - Functions_Key.py
File Edit Watcher MediaTools JES Functions Window Layout Help
59 # Combining it All together! Doing More with Turtle Colors!
60
61 # RainbowSpiral
62 def rainbowSpiral(turtle):
63     colors = [red, orange, yellow, green, blue]
64     turtle.penWidth = 5
65     for i in range(30):
66         turtle.color = colors[i % 5]
67         betterSquare(turtle)
68         turtle.turn(12)
69
```



BONUS: Doing More with Turtle Colors!

```
JES - Jython Environment for Students - Functions_Key.py
File Edit Watcher MediaTools JES Functions Window Layout Help
63 colors = [red, orange, yellow, green, blue]
64 turtle.penWidth = 5
65 for i in range(30):
66     turtle.color = colors[i % 5]
67     betterSquare(turtle)
68     turtle.turn(12)
69
70 # Make Rainbow Wind Mill with a Pink Stick
71 def rainbowWindMill(turtle):
72     colors = [red, orange, yellow, green, blue]
73     turtle.penWidth = 3
74     for i in range(5):
75         turtle.color = colors[i % 5]
76         makeTriangle(turtle)
77         turtle.turn(15)
78     turtle.color = pink
79     turtle.turn(180)
80     turtle.forward(180)
```



Challenge: Use all the turtle commands you have learned and create your initials!

- **turtle.forward(distance)**- command to move the turtle in the directions it's facing. Default distance is 100 pixels.
- **turtle.turn(degrees)**- turns the turtle in 90 degrees depending on the direction its facing.
- **turtle.moveTo(x, y)**- command to place the turtle in a new location in the world.
- **turtle.penUp()**- command to pick the pen up.
- **turtle.penDown()**- command to put the pen down again to draw.
- **turtle.penWidth = (1-10)** - allows you to change the size of the pen
- **turtle.color = red (blue, green . . .) -or- (255, 255, 255) RGB notation**- allows you to change the color of the pen and turtle.

Survey

- Copy and paste the URL below into your browser to take the survey!
- Thanks, we appreciate your feedback!
- <http://tinyurl.com/ggctech>