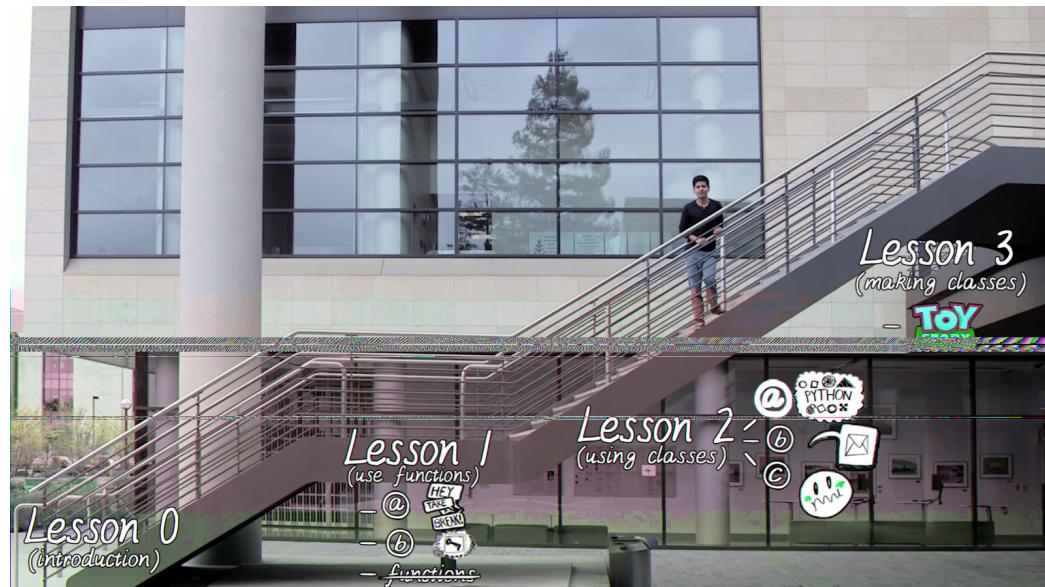


Lesson 02a Notes

Course Map

So, we started out lesson one using a lot of different functions, and we built two main projects. In the first one, we helped our friends take a break and in the second one we sent a secret message to our loved ones. Then we came across a scenario, while building a movie website, where using functions didn't quite present a very elegant solution. That solution requires something new. And that new idea, in programming, is called classes, and here in lesson two, we will spend a lot of time, and build a lot of different examples, showcasing the power of using classes.



Now, we will return to our movie database example in lesson three. But here, in lesson two, we will start out with ideas and examples that are the easiest for me to explain the power of classes to you. Let's begin.

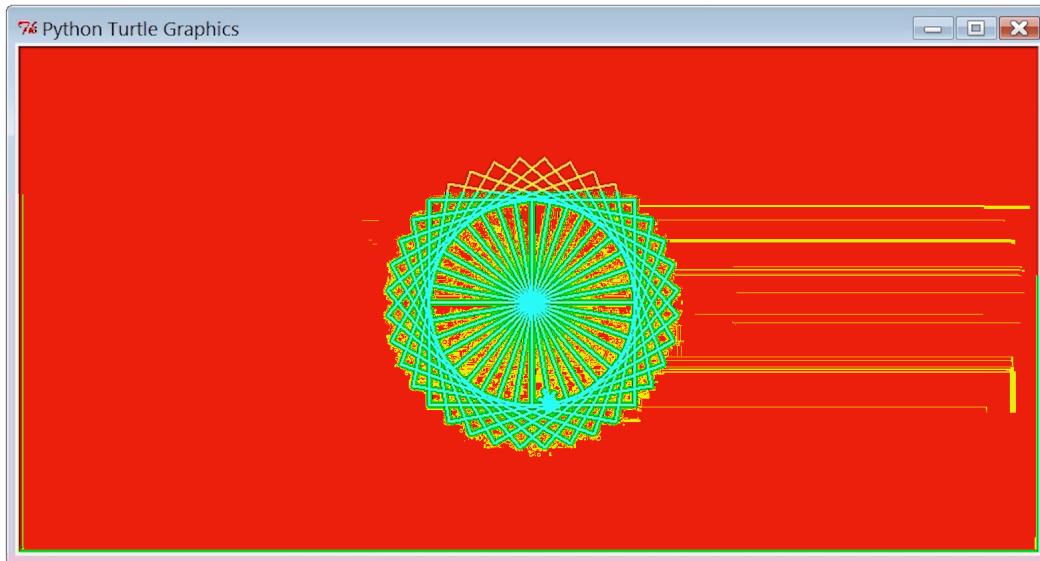
Drawing Turtles (Story)



So I'd been thinking about how to introduce this topic called Classes. In so doing, I went back to how it was explained to me in college. I realized, that those ideas, were filled with jargon, and were really confusing. Then, I landed on this thing in Python, that can help us draw shapes. Shapes like squares, circle, and fractals. And it turns, out that drawing these shapes, is one of the easiest ways to learn about classes. So lets begin, by drawing some shapes.

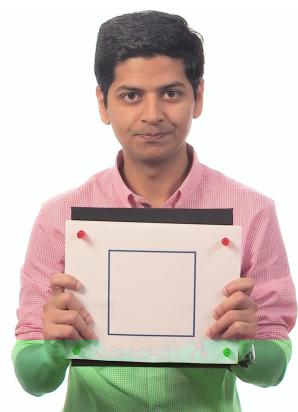
Drawing Turtles (Output)

So I wrote a program that is currently hidden behind this graphic. If I run it, it begins to draw squares. And out of those squares emerges a circle.

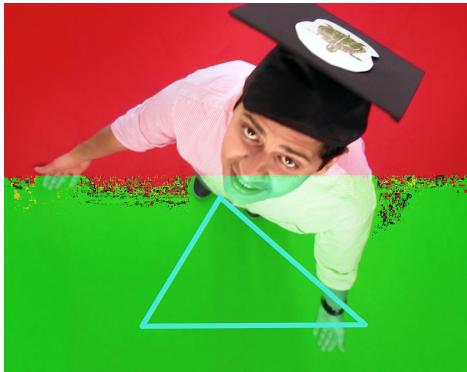


Lets build a shape and learn about classes at the same time.

How Would You Draw a Square



Now, one way to draw a whole bunch of squares, is to start by drawing just the one square. So, here's what I'm going to do. I'm going to step away from my computer for a little bit. Put this hat on. And I will draw on this red carpet, which will



be my canvas. So, when I move around on it, you will notice that I can draw. There it is. And in this case, I drew a triangle. So, here's a task for you. If you were ask me to draw a square on this red carpet, what step by step instructions would you give me? Write your answers in the box provided.

How Would You Draw a Square (Solution)

So, here is one way to draw a square. I can move forward, turn right, move forward, turn right, move forward one more time, turn right again, move forward and turn right. There it is.



Drawing a Square

Move Forward
Turn Right
Move Forward
Turn Right
Move Forward
Turn Right
Move Forward
Turn Right

Okay. Time to draw the square. But this time, using the computer. So, I created a new Python file and called it `mindstorms.py`. You could've called it something else. And then, inside it, I created a function called `draw_square`, which, for now, is empty.

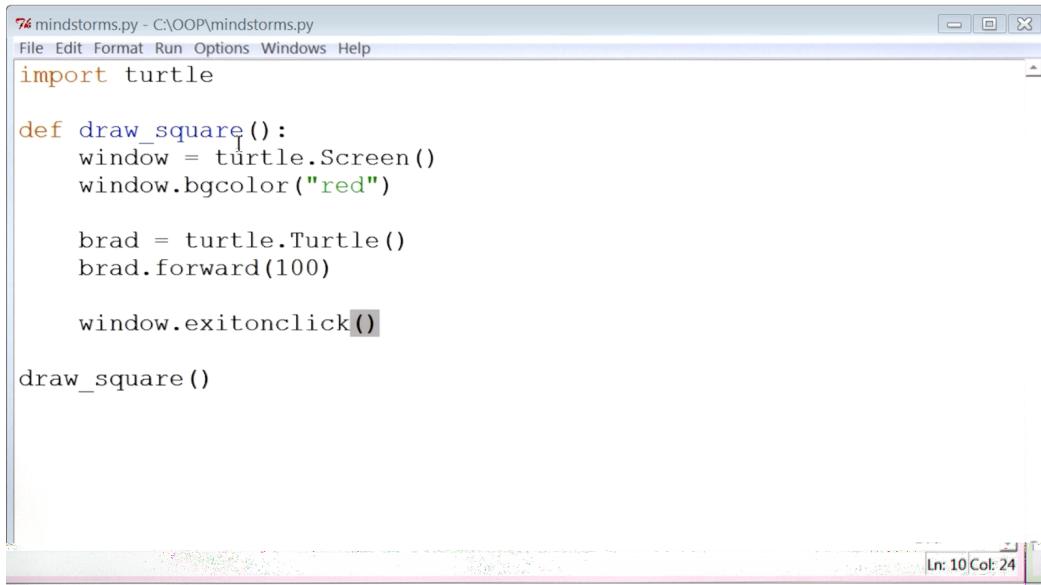
Now, I know there is this module in Python called `turtle`.

This is the actual thing that moves around and draws stuff on the computer. By the way, I encourage you to find out

why this thing is called `turtle` and not a dog, or a cat.

One way to start to draw is by saying, `turtle.Turtle`. Now, this syntax is a little weird and we will understand it by the end of this video, but it suffices to say that this is how we grab our turtle. I will call it `brad`. Now, the first thing we want to do is move `brad` forward. And we can use the `forward` function for that. This function takes in a number, which is the distance we want to move forward. Now, I just want to run

this code, and see if it's even working. But, you know what, the one thing we forgot, is that we still don't have a screen on which this turtle, aka brad, is going to move around. In other words, we need the red carpet equivalent in our program. I know that is done by adding a window screen, and this window is going to have a red background, and I want to be able to close this window any time I click on it. Now, let's save and run this program.



```
76 mindstorms.py - C:\OOP\mindstorms.py
File Edit Format Run Options Windows Help
import turtle

def draw_square():
    window = turtle.Screen()
    window.bgcolor("red")

    brad = turtle.Turtle()
    brad.forward(100)

    window.exitonclick()

draw_square()
```

The screenshot shows a Windows-style application window titled "76 mindstorms.py - C:\OOP\mindstorms.py". The menu bar includes File, Edit, Format, Run, Options, Windows, and Help. The main area contains Python code for drawing a square. The code imports the turtle module, defines a draw_square function, creates a red background window, initializes a turtle named brad, moves it forward 100 units, and then exits on click. The status bar at the bottom right indicates "Ln: 10 Col: 24".

All right. It seems that we have a window with a red background. And a turtle named brad, which has moved 100 steps forward. Okay, so I am back at the code, and now I'm going to finish making the square. So, after brad has moved 100 steps forward, I want to turn it right, and I want to do so by 90 degrees. So, I can say, brad.right(90), and because I want to do these two steps right here, four times to do a square, I'll just copy paste them a few times. So, I'm going to copy these two lines and I'm going to paste them. Second time, third time, fourth time Let me save and run, alright, it seems like we have our first square.

```
76 *mindstorms.py - C:\OOP\mindstorms.py*
File Edit Format Run Options Windows Help
import turtle

def draw_square():
    window = turtle.Screen()
    window.bgcolor("red")

    brad = turtle.Turtle()
    brad.forward(100)
    brad.right(90)
    brad.forward(100)
    brad.right(90)
    brad.forward(100)
    brad.right(90)

    window.exitonclick()
```

Change Turtle Shape Color and Speed

OK, so the next step is for us to understand how Python knows about this thing called the turtle. But before we do that, here's your task. I want you to reproduce this entire code on your computer. And oh, and by the way, you can customize this turtle. You can change its shape. You can also change its color, and you can change the speed at

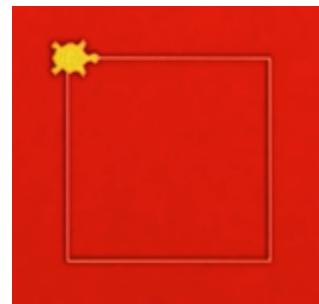
```
76 *mindstorms.py - C:\OOP\mindstorms.py*
File Edit Format Run Options Windows Help
import turtle

def draw_square():
    window = turtle.Screen()
    window.bgcolor("red")

    brad = turtle.Turtle()
    brad.shape("turtle")
    brad.color("yellow")
    brad.speed(2)

    brad.forward(100)
    brad.right(90)
    brad.forward(100)
    brad.right(90)
    brad.forward(100)
    brad.right(90)
    brad.forward(100)
```

which it moves.



Let me quickly save and show you the output. Alright. So it seems like we've changed a few things about the turtle. Its color, which is yellow now, and shape and also its speed. Oh, by the way, if you need more information about how to customize your turtle. There are some helpful links in the instructor notes.

Links

- Change Turtle Shape - <http://docs.python.org/2/library/turtle.html#turtle.shape>
- Change Turtle Speed - <http://docs.python.org/2/library/turtle.html#turtle.speed>
- Change Turtle Color - <http://docs.python.org/2/library/turtle.html#turtle.color>

After you have created your own custom square I want you to take a screenshot of your output and post that on the forum. Make sure to check this box before you continue.

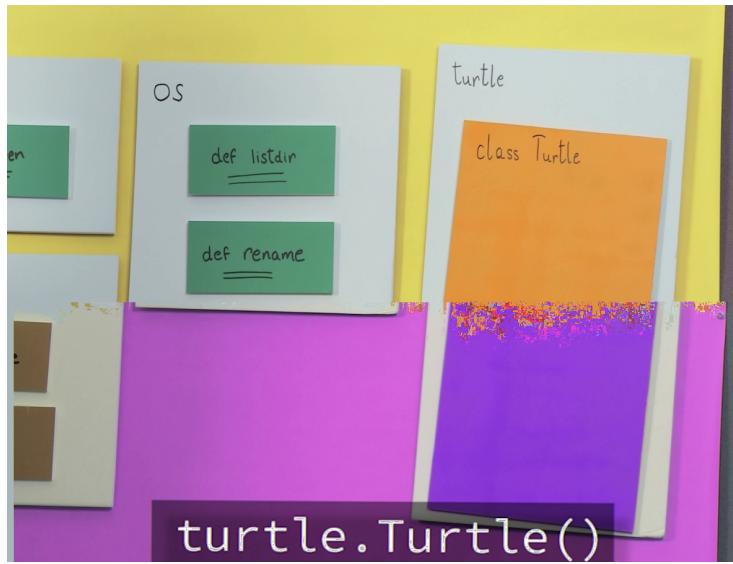
1. Draw a square using code
2. Customize it by changing the turtle's shape, color and speed
(more information on how to do this in the instructor notes)
3. Take a screenshot of your output and post it on the forum

Check this box after submitting your response on the forum

(To access the forum click the "Drawing Square Turtles" discussion thread)

Where Does Turtle Come From

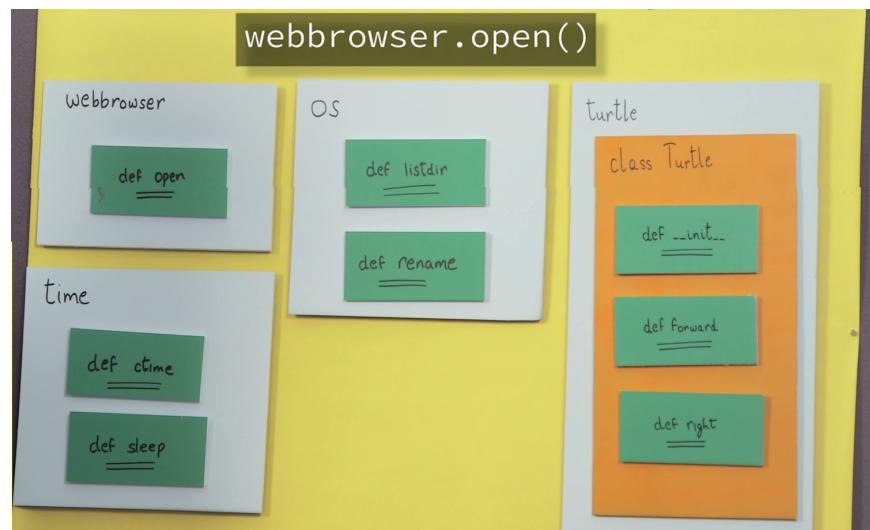
Thank you for finishing your assignments. I will be providing feedback on the discussion forums periodically. Okay, so I now want to talk about how Python knows about this thing called the turtle. Well, inside the Python Standard Library, there is this file called turtle. turtle with a lowercase t, and inside that file is something new.



And that something new is called a class. A class Turtle with an uppercase T. Now, this is a little different than what we have seen thus far.

What we have done thus far is some sort of file, like web browser, dot some sort of function, like open. But we haven't seen this thing called a class. Which begs the

question, what is a class? For now you can think of a class as a neatly packaged box, that puts things together really well. And it allows us to write code like `turtle.Turtle()`, with these strange looking brackets. As if we were calling a function.

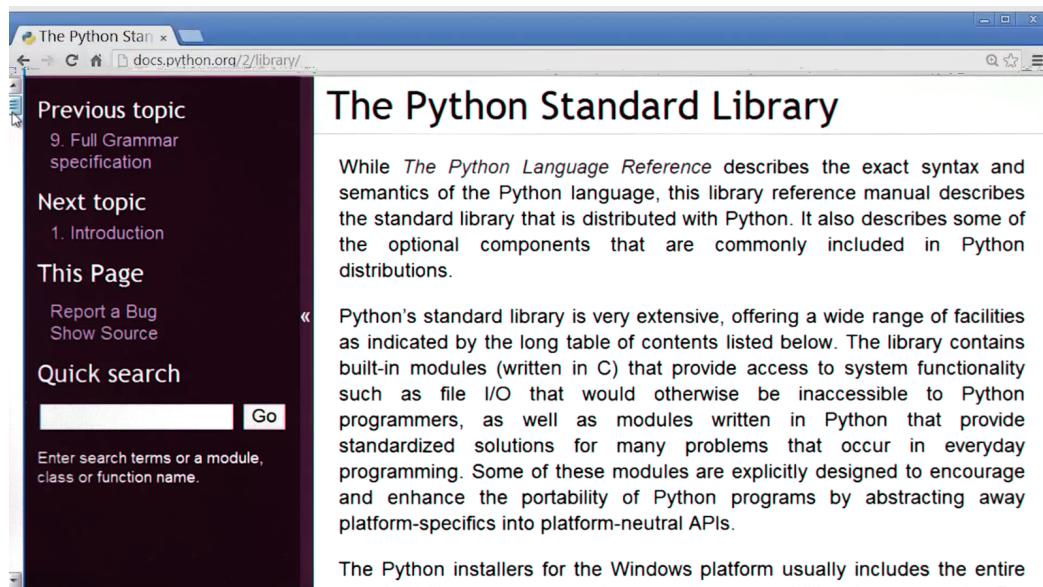


In fact, when we do it on this piece of code, we are in fact calling a function, a function called init

which is defined inside the class Turtle. Init stands for initialize, and what it does is it creates space in memory for a new instance or a new object of the class Turtle. This instance we called brad. Brad can now access all of the rest of the methods that are inside class Turtle. Like forward, and right. Forward makes brad turn forward, and right makes brad turn to the right. Don't take my word for it, lets look for them in the documentation.

Reading Turtle Documentation

So here we are in the Python documentation, and if I scroll down, I can find a module named turtle. (<http://docs.python.org/2/library/turtle.html>)



The screenshot shows a web browser window titled "The Python Standard Library". The URL in the address bar is "docs.python.org/2/library/". The left sidebar contains navigation links: "Previous topic" (link to "9. Full Grammar specification"), "Next topic" (link to "1. Introduction"), and "This Page". Below these are links for reporting a bug and viewing the source code. A "Quick search" section features a text input field with placeholder text "Enter search terms or a module, class or function name." and a "Go" button. The main content area is titled "The Python Standard Library" and contains two paragraphs of text. The first paragraph describes the library reference manual, stating it covers the standard library distributed with Python and some optional components. The second paragraph discusses Python's standard library being very extensive, providing built-in modules for system functionality like file I/O and standardized solutions for common programming problems. It also mentions the portability of Python programs through abstracted APIs. At the bottom of the page, there is a note: "The Python installers for the Windows platform usually includes the entire".

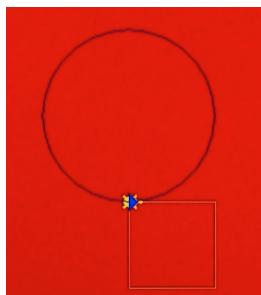
There is a lot of helpful information about the module here, including information on why turtle is actually called turtle.

And if I scroll down on this page, I will find a class Turtle. Turtle with a capital T, which is the class we used. There it is. I'm going to go back, and if I scroll down again, I can see all of the methods we use. Like forward, right, and a lot of others.

Two Turtles

So, now that we know where turtle exists within Python, I want to return back to our code. By the way, I had to reduce the font size of the code a little bit, so we could see all of it in the same frame.

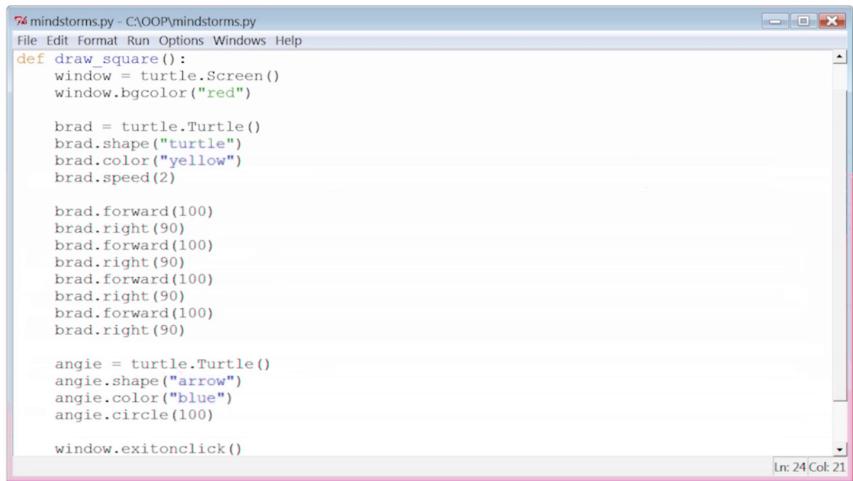
Now, what we have done thus far is, created an instance of a class called turtle, we called it brad and brad draws a square. If we can do that, we can make another example or instance of the class turtle. And we can call it, let's say, angie. There it is.



Now, I want angie to draw a circle. A circle of radius, let's say, 100. I also want angie's shape to be in the form of an arrow and I want angie's color to be blue. There it is. So, let me save and run this thing and see how it looks. Aha, it seems we have brad making a yellow square and angie making a blue circle. It seems like we are getting somewhere.

Whats Wrong with This Code

Now honestly, if I showed this code to my computer science professor from college, she would be really upset. Can you guess why she would be upset with the code



The screenshot shows a Windows-style code editor window titled "76 mindstorms.py - C:\OOP\mindstorms.py". The menu bar includes File, Edit, Format, Run, Options, Windows, and Help. The code itself is as follows:

```
76 mindstorms.py - C:\OOP\mindstorms.py
File Edit Format Run Options Windows Help
def draw_square():
    window = turtle.Screen()
    window.bgcolor("red")

    brad = turtle.Turtle()
    brad.shape("turtle")
    brad.color("yellow")
    brad.speed(2)

    brad.forward(100)
    brad.right(90)
    brad.forward(100)
    brad.right(90)
    brad.forward(100)
    brad.right(90)
    brad.forward(100)
    brad.right(90)

    angie = turtle.Turtle()
    angie.shape("arrow")
    angie.color("blue")
    angie.circle(100)

    window.exitonclick()
```

In the bottom right corner of the editor, there is a status bar with "In: 24 Col: 21".

Can you guess why Kunal's computer science professor from college may be upset with this code?



we've written this far? Submit your answer in this box.

Improving Code Quality

Well, I can think of at two reasons why my computer science professor from college may be upset with the quality of this code.

The first is that I'm doing something repetitive inside my code. Things like move forward, turn right, move forward, turn right. I do that four times, but I don't use any sort of a loop.

Another reason is that the name of my function is draw_square. Which I do. But then I go ahead and draw a circle inside that function also. That doesn't seem quite logical to me.

So, based on those two ideas, what changes would you make to the code that we've written thus far to make it more usable? I want you to make those changes and then upload a screenshot of the code to the discussion forum. Here's an extra credit assignment, which is to create a third turtle, and this one will draw a triangle. Once you are done with that, I want you to upload a screenshot of the new output to the discussion forum. Make sure you check this box after you've submitted your responses to the forum.

Improving Code Quality

So I made a few changes to the code, see if you agree with them. In the

```
import turtle

def draw_square(some_turtle):
    for i in range(1,5):
        some_turtle.forward(100)
        some_turtle.right(90)
```

draw_square()
function, I now
use a loop, and
this loop does
my repetitive
tasks, tasks
like move the
turtle forward
and turn it

right, and this loop runs 4 times.

Then I created a new function called draw_art(), in this function I create the window screen. This is the screen that has the background color as red, and then I create my first turtle, which I named brad. I change brad's shape, color and speed, and then I call the draw square function, so that I can actually draw my first square.

After that, I create my second turtle, angie. And here are angie's shape and color, and then angie draws a circle. And then towards the end, I call the draw_art() function which is right here. So let me go ahead and save, and run this program to make sure everything is running fine. And there's our two turtles. One drawing a square, the other one drawing a circle.

What Is a Class

So, before we continue with this project, I want to highlight a few new words that we have used in the code thus far. We've identified turtle as a class. And brad and angie as instances of that class. Now, I want to take a moment and talk about these new words. You can think of a class as a blueprint of a building.

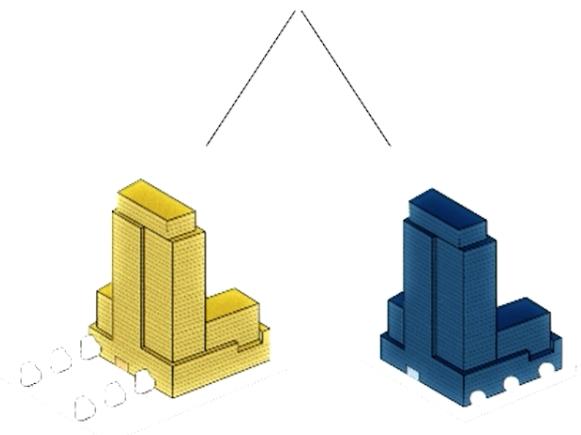
Now, a blueprint as we know, contains certain pieces of information about the building - the height of the building, the number or rooms in the building. Now, the same blueprint can be used to build multiple different buildings. For instance, an office with yellow walls, and a residential complex with blue walls.

A class is like a blueprint. So let's take the example of class Turtle that we've used thus far. A class Turtle, much like a blueprint, contains certain pieces of information about the turtle. The turtle's size, its color, the fact that the turtle can move forward and turn right. And from this class, or blueprint we can create multiple different instances of that class. For example, Brad is an instance of class Turtle that draws a yellow square, and Angie is also an instance of class Turtle that draws a blue circle. So, in summation, you can think of a class as a blueprint, and you can think of its objects as example or instances of that blueprint.

Class



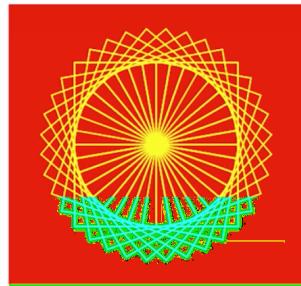
Instance
or
Object



Making a Circle Out of Squares

Okay, so here we are, back at the code. For now, what I will do is, continue to draw the square, but I will comment the rest of the code out. So, let me go ahead and do that. Let me quickly save, and run to make sure everything's running well. All right, the program still seems to be drawing a square, which is good.

Now, if you remember the original challenge, It was to draw a circle, out of squares. And thus far, all you've done is just draw, just the one square. So, given that, here's a question for you. How would you've build on the code that we've written thus far, to draw this image (circle out of squares)? Now, don't worry about writing code right now. All I want you to do, is identify in plain simple English, the steps you would take to solve this problem. You can submit your answers on the discussion forum. Once



*How would you build on the code we have written
thus far to draw this image?*



Check this box after submitting your response on the forum

(To access the forum click the “Planning To Draw” discussion thread)

you're done, make sure you check this box before you continue.

Making a Circle Out of Squares

So, one way of doing this is to draw a square and then turn a little bit to the right and then draw another square. And then turn a little bit to the right again and then draw another square. And keep doing that. Let's build that.

So, here's the piece of code that draws a square. And after I've done that, what I want the turtle to do is turn a little bit to the right. So, let me add a piece of code that says brad.right, let's say by ten degrees. Now, because I want these two statements to

happen over and over again, I will put them inside a loop. So let me do that. So I'm going to add a for loop by saying for i in range, let's say this loop runs four times. I'll add a colon and then put these two things inside the loop. Let me save, and run, and see what happens. Ah-ha, so here is the output. We've drawn two squares, three squares and it should stop, there it is four squares.

So all I have to do now, is let the turtle keep going around the circle, which has 360 degrees. So, this loop should run not four times, but 36 times instead. Let me save and run that. All right. There it is, folks. A circle made out of squares. Now I'm going to put some background music behind this video and let you enjoy our final creation.

Turtle Mini Project

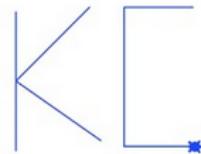
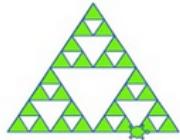
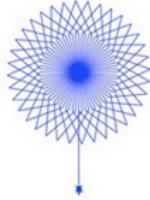
Congratulations on getting this far. Now that you know how to draw things using code, here's your mini project. We want you to write code to draw a flower, or some fractals, or write your initials. Once you're done, post a video or a screenshot of the output of your code on the discussion forum. Check this box after you've submitted your responses on the forum.



Turtle - Mini Project

Draw any of the following shapes using code

(once you are done, post a video or photos of the output on the forum)

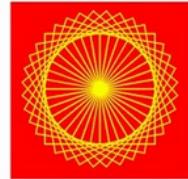


Check this box after submitting your response on the forum

(To access the forum click the "Turtle - Mini Project" discussion thread)

Comfort Level

Thank you for submitting your responses on the discussion forum. Now, we've written three different programs in the course thus far. And at this point I want to ask you again, how confident do you feel in your ability to write computer programs? Here are some answer choices.



How confident do you feel in your ability to write computer programs?



They Look So Similar

Thank you for sharing your answer. Now that we've used classes for the very first time, I want to highlight something that took me a really long time to understand when I was in college.

When we wanted to use functions previously, we wrote code like `webbrowser.open()`. But when we wanted to create instances of this class, `class turtle`, we wrote code that said, `turtle.Turtle()`. Now, look at both of these statements. They look rather similar. It seems like we're calling a function in both of these cases. But something different is happening behind the scene. When we call `webbrowser.open`, all we are doing is calling a function. But when we call `turtle.Turtle()`, we are calling a function called `init` defined inside the class `Turtle` which constructs space in memory for a new instance or object of the class `Turtle`. This instance we called Brad.

