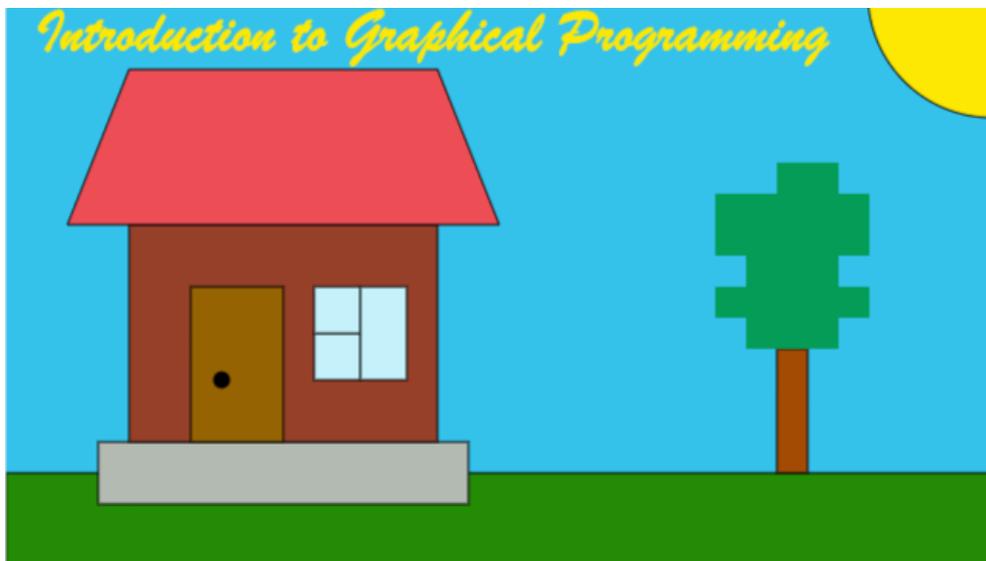


Introduction to Graphical Programming

Workbook

Programming Tutorials by PyAngelo

<https://pyangelo.com/tutorials/introduction-to-graphical-programming>



Name: _____

Task 1 - Coordinates

setCanvasSize(width, height)

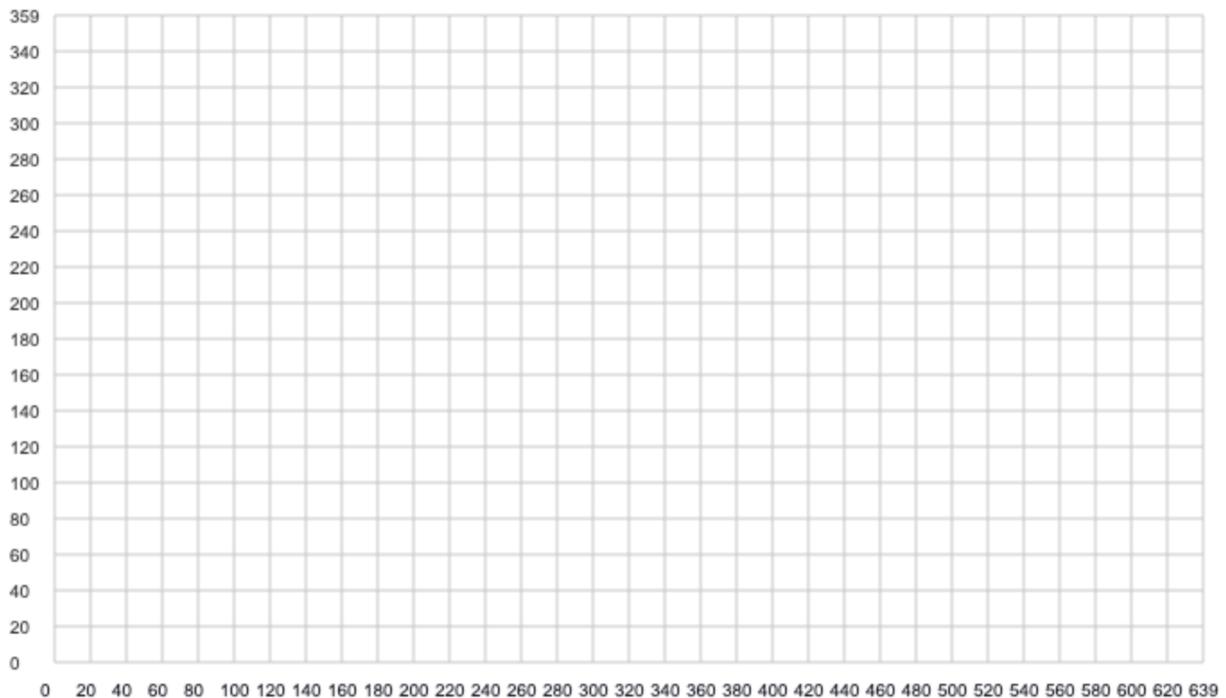
The setCanvasSize() function sets the width and height of the canvas that we will draw our graphics on. The first number we specify determines the width of the canvas, and the second determines the height. For example, if we used the following command:

```
setCanvasSize(640, 360)
```

It would create a grid that is 640 pixels wide and 360 pixels high.

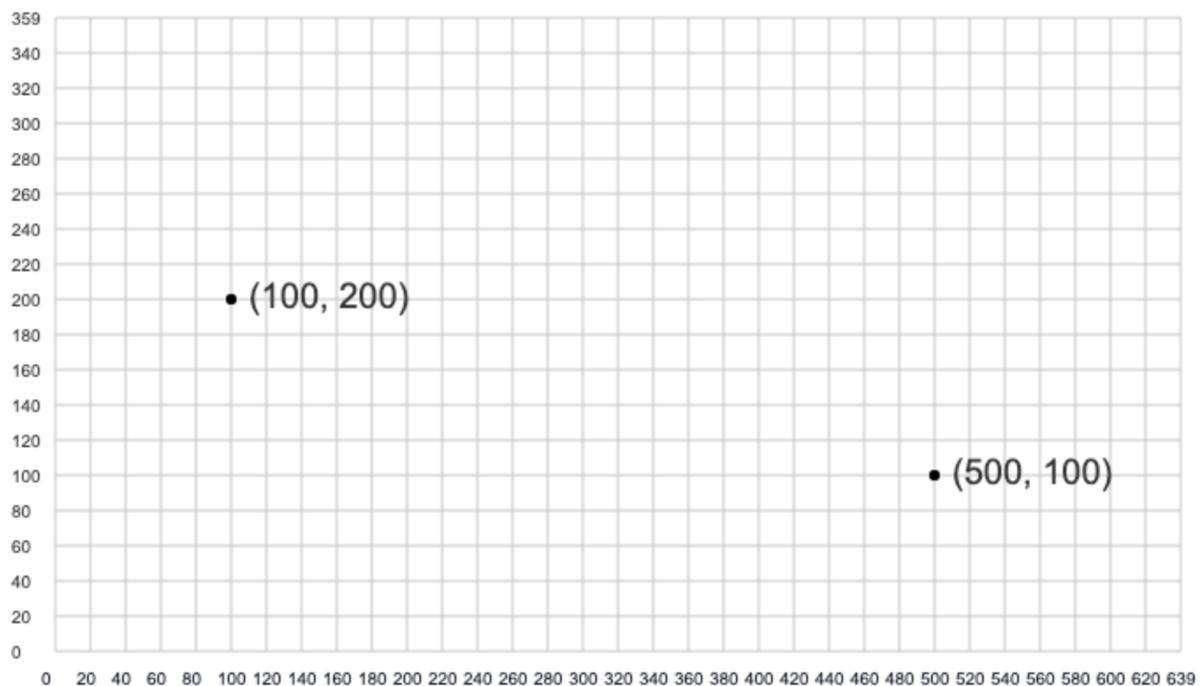
Coordinate System

Just like you learned in maths, the coordinate system has an x-axis which runs horizontally and a y-axis which runs vertically. If we specify the width and height to be 640 and 360 pixels respectively, the canvas would look like this:



The grid lines and numbers are not actually displayed when we run our program but are shown here to highlight the coordinate system. Note that the bottom left corner of the canvas is at the position (0, 0). Also note that along the x-axis we can specify a position from 0 to 639. Before you ask, yes that is 640 pixels wide as we can specify 0 as well as 1 through to 639. The same applies for the y-axis. In this example it is 360 pixels high with values ranging from 0 to 359.

If you wish to specify a single pixel you always specify the x-axis first followed by the y-axis. For example the following diagram shows two points: (100, 200) and (500, 100)



background(r, g, b)

The background function draws a coloured rectangle over the entire canvas. It will draw this rectangle over anything that was previously drawn and hence is often used to clear the canvas. We will talk more about how to choose a colour by specifying the amount of red, green and blue later in this workbook. The following command paints the canvas yellow:

```
background(255, 255, 0)
```

point(x, y)

The point command draws a point at the position specified by the x and y parameters. The following two lines of code draw the two points shown in the diagram above:

```
point(100, 200)  
point(500, 100)
```

strokeWeight(weight)

The strokeWeight() function determines how many pixels will be drawn when we call the point() function we learned about above. In order to take effect this function must be called before we draw any points. The order of the code is very important. The strokeWeight() function also sets the width of any lines and the border of any shapes that we draw after it has been called.

```
strokeWeight(5)  
point(100, 200)  
point(500, 100)
```

The code above draws the same two points but this time they are bigger as we have set a weight of 5 pixels for each point.

Task 1 Challenge

Your task is to create a canvas that is 640 pixels wide and 360 pixels high. You are then to set the background colour of the canvas to yellow. Lastly, you are required to draw 5 points all with a weight of 5 pixels. You should draw one point in each corner of the canvas, on the 5th point in the middle of the screen. Good luck!

Check Your Knowledge - Task 1

Question 1a

The x-axis moves _____ the screen.

The y-axis moves _____ the screen.

Question 1b

What does the following command do?

```
setCanvasSize(1000, 300)
```

Question 1c

What does the following command do?

```
background(255, 255, 0)
```

Question 1d

What does the following command do?

```
strokeWeight(5)
```

Question 1e

What does the following command do?

```
point(25, 75)
```

Check Your Knowledge Answers - Task 1

Question 1a

The x-axis moves **across** the screen

The y-axis moves **up and down** the screen

Question 1b

It creates a canvas with a width of 1000 pixels and a height of 300 pixels.

Question 1c

It changes the colour of the canvas to yellow by drawing a rectangle over the entire canvas.

Question 1d

It sets the strokeWeight to be 5 which will affect any subsequent calls to the point() function.

Question 1e

It draws a point at the location (25, 75). That point is 25 pixels along the x-axis and 75 points up the y-axis.

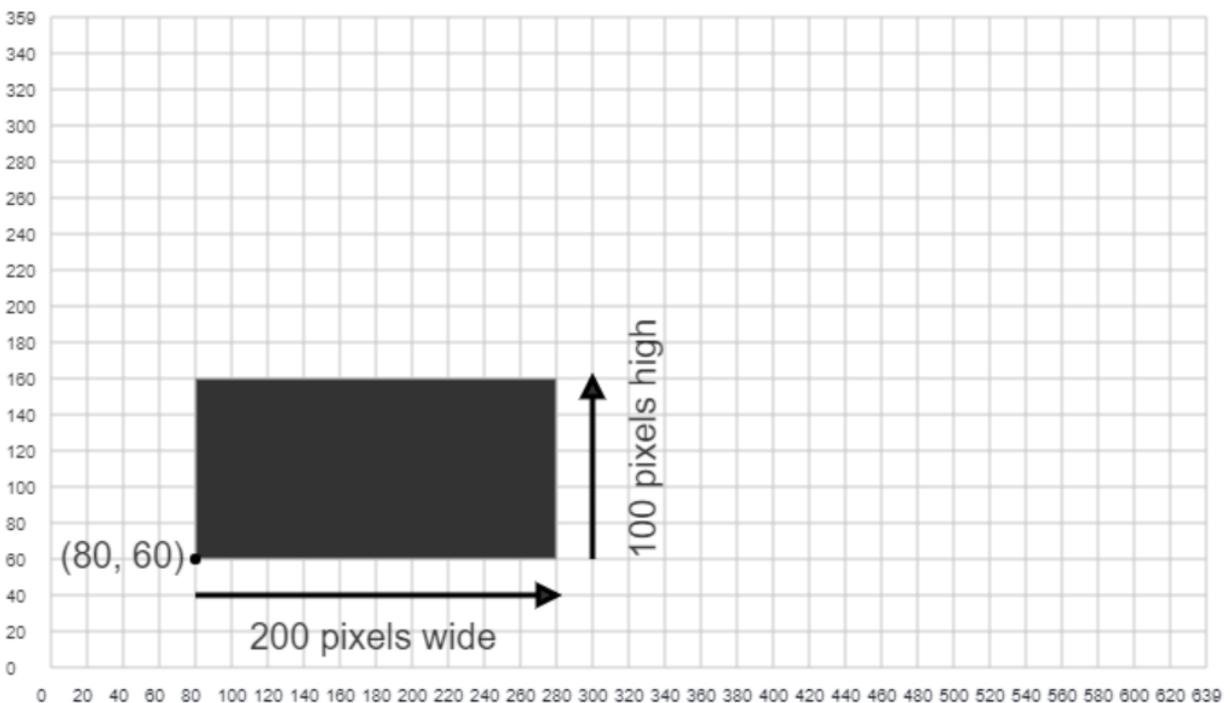
Task 2 - Drawing Shapes

`rect(x, y, width, height)`

The `rect()` function draws a rectangle on the screen. The numbers we pass to a function are called arguments. The first two arguments specify the bottom left corner of the rectangle. The first argument is the x location and the second argument is the y location. The third argument specifies the width of the rectangle and the fourth argument specifies the height of the rectangle. Here is an example:

```
setCanvasSize(640, 360)
background(255, 255, 255)
rect(80, 60, 200, 100)
```

And here is an annotated version of the result:

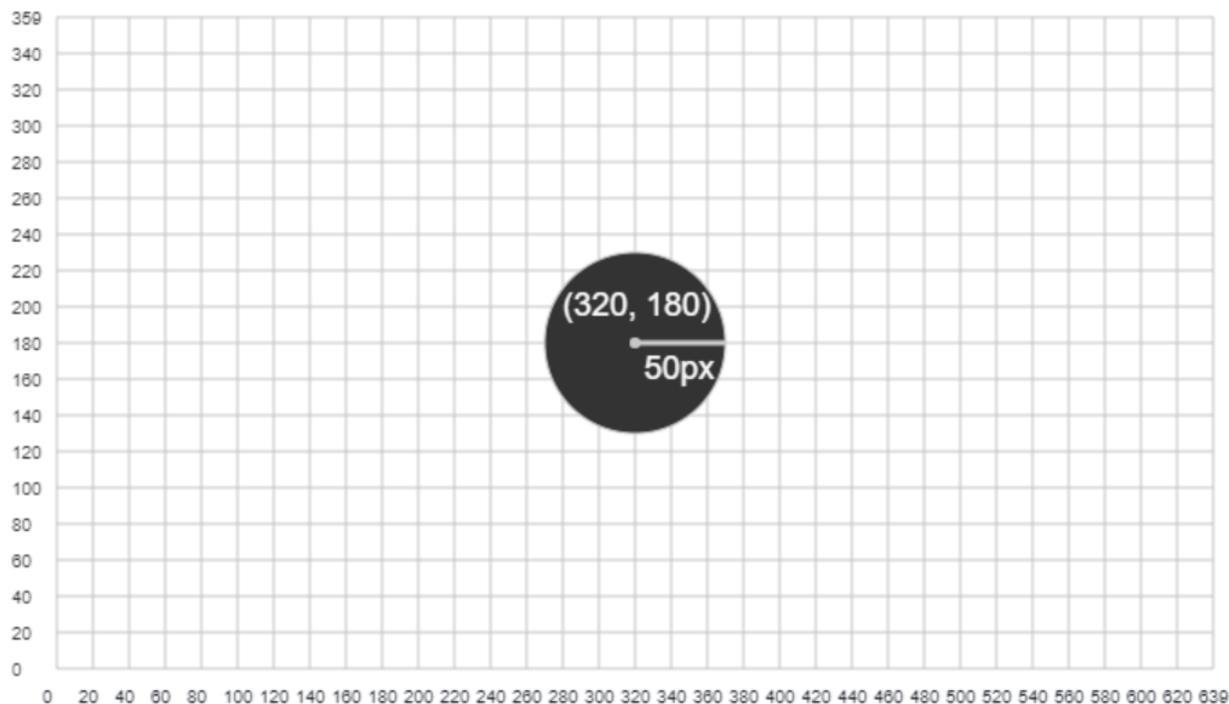


circle(x, y, radius)

The circle() function draws a circle on the screen. The first two arguments specify the centre of the circle. The first argument is the x location and the second argument is the y location. The third argument specifies the radius of the circle which is the distance from the centre to the edge of the circle. Here is an example:

```
setCanvasSize(640, 360)
background(255, 255, 255)
circle(320, 180, 50)
```

And here is an annotated version of the result:



Other Shapes

PyAngelo has the more functions you can use to draw other shapes including:

- ellipse(x, y, radiusX, radiusY)
- arc(x, y, radiusX, radiusY, startAngle, endAngle)

- `line(x1, y1, x2, y2)`
- `triangle(x1, y1, x2, y2, x3, y3)`
- `quad(x1, y1, x2, y2, x3, y3, x4, y4)`

I encourage you to experiment drawing these shapes. You can find all the information you need about these function on the PyAngelo reference page:

<https://pyangelo.com/reference#basic-shapes>

Task 2 Challenge

Your task is to draw a silhouette using the basic shapes you have learned. For example, you could attempt to draw Mickey Mouse, a tree, or a truck. Good luck!

Check Your Knowledge - Task 2

Question 2a

The first two arguments to the `rect()` function specify the _____ corner of the rectangle.

Question 2b

The third and fourth arguments to the `rect()` function specify the _____ and the _____ of the rectangle.

Question 2c

The first two arguments to the `circle()` function specify the _____ of the circle.

Question 2d

The third argument to the `circle()` function specifies the _____ of the circle.

Question 2e

What is the height of the rectangle drawn with the following code?

```
rect(10, 20, 50, 75)
```

Check Your Knowledge Answers - Task 2

Question 2a

The first two arguments to the `rect()` function specify the **bottom-left** corner of the rectangle.

Question 2b

The third and fourth arguments to the `rect()` function specify the **width** and the **height** of the rectangle.

Question 2c

The first two arguments to the `circle()` function specify the **centre** of the circle.

Question 2d

The third argument to the `circle()` function specifies the **radius** of the circle.

Question 2e

The height of the rectangle is **75 pixels**.

Task 3 - Colours

RGB - Red, Green, Blue

Colours in PyAngelo are specified using the RGB colour scheme. Using this colour scheme you specify the amount of red, the amount of green, and the amount of blue to be used in a colour. The values you can specify range from 0, which represents none of that colour, through to 255 which represents the full intensity of that colour.

Here is a table showing some common colours using the RGB colour scheme.

Red Value	Green Value	Blue Value	Colour
0	0	0	Black
255	255	255	White
20	20	20	Dark Gray
230	230	230	Light Gray
255	0	0	Red
0	255	0	Green
0	0	255	Blue
255	255	0	Yellow
255	0	255	Pink
0	255	255	Aqua

background(r, g, b)

We have talked about the background() function before. It draws a rectangle over the entire canvas of the colour specified. Now we know more about colours, we can set the

background to a colour of our choice. For example the command below will set the background to aqua:

```
background(0, 255, 255)
```

fill(r, g, b)

The fill() command is similar to the background() command but it tells PyAngelo what colour to fill in any subsequent shapes you draw rather than the background colour. The following code will draw two rectangles both of which will be pink:

```
fill(255, 0, 255)  
rect(20, 30, 100, 200)  
rect(320, 330, 100, 200)
```

noFill()

If you do not wish the shape to have any colour you can call the noFill() function. This function does not take any arguments. You must remember to include the brackets when calling this function or it will not execute.

stroke(r, g, b)

The stroke() command is passed the same information as the fill command. That is the amount of red, green, and blue to display. However, with the stroke command, this sets the colour of the border of any subsequent shapes we draw. The following command will cause any subsequent shapes to have a black border:

```
stroke(0, 0, 0)
```

noStroke()

If you do not wish the shape to have any borders you can use the `noStroke()` function. Just like the `noFill()` function, the `noStroke()` function does not take any arguments. You must remember to include the brackets when calling this function or it will not execute.

Task 3 Challenge

Your task is to create a colourful image such as a tree or a cloud.

Check Your Knowledge - Task 3

Question 3a

What does the following command do?

```
background(255, 0, 0)
```

Question 3b

What does the following command do?

```
fill(255, 255, 255)
```

Question 3c

What does the following command do?

```
stroke(0, 255, 0)
```

Question 3d

What do the following commands do?

```
noStroke()
```

```
noFill()
```

Check Your Knowledge Answers - Task 3

Question 3a

`background(255, 0, 0)` draws a red rectangle over the entire canvas, effectively clearing the screen with a red colour.

Question 3b

`fill(255, 255, 255)` sets the fill colour of any shapes subsequently drawn to white.

Question 3c

`stroke(0, 255, 0)` sets the border colour of any shapes subsequently drawn to green. It also sets the colour to green for any lines or points that are drawn after this function is called.

Question 3d

`noStroke()` causes any shapes that are drawn to have no border. It also means that any calls to the `line()` function or `point()` function will not be displayed.

Task 4 - Errors

Error Messages

One of the scariest things when learning to program is dealing with errors. If there is an error in your program an error message will show up in the right hand pane of the PyAngelo editor which we will call the output window. The following program has a number of errors:

```
setCanvasSize(400, 400)
back(0, 0, 0)
fill(255, 255, 0)
rectangle(100, 100)
```

If you copy this code exactly as written and then click the “Start” button and you should see the first error appear in the output window:

```
NameError: name 'back' is not defined on line 2
```

```
Error found in file main.py
```

The first thing to do when you see an error is not panic! Take a deep breath and then simply read the error message. Here it tells us there is a NameError: name ‘back’ is not defined on line 2. This gives us some important information. Firstly the error is on line 2. It also tells us there is no such command as ‘back’. My approach would be to ask what I was trying to achieve with this line of code. In this case I wanted to change the background colour and this helps me figure out that I need to change the line of code from back(0, 0, 0) to background(0, 0, 0). Go ahead and update the code with this change now and then run the code again. Now we get a new error message:

```
NameError: name 'rectangle' is not defined on line 2
```

```
Error found in file main.py
```

Now we can see that the next error is on line 4 and that there is no such command as rectangle. If you remember from the section on drawing basic shapes, to draw a

rectangle we use the `rect()` command. Go ahead and update line 4 to `rect(100, 100)` and click the start button. Now we get a new error:

```
TypeError: rect() takes exactly 4 arguments (2 given) on line 4
```

```
Error found in file main.py
```

The error is once again on line 4. However, this time it informs us that the `rect()` function takes four arguments and we only provided two. If you recall, the `rect()` function requires the `x`, `y`, `width` and `height` arguments. So go ahead and update line 4 to provide a `width` and `height` for the rectangle. I'm going to use a `width` of 200 and `height` of 150.

An error in programming is also referred to as a bug. This originates from the first ever "bug" where an actual moth got stuck in the computer and caused the program to fail. They removed the moth and ever since the process of fixing errors in your code has been known as debugging. Here's our final debugged version of the code:

```
setCanvasSize(400, 400)
background(0, 0, 0)
fill(255, 255, 0)
rect(100, 100, 200, 150)
```

Check Your Knowledge - Task 4

Question 4a

On what line is the following error?

```
NameError: name 'background' is not defined on line 2
```

```
Error found in file main.py
```

Question 4b

How would you fix the error reported in question 4a?

Question 4c

How would you fix the following error?

```
setCanvasSize(400)
```

```
TypeError: setCanvasSize() missing 1 required positional arguments: on line 1
```

```
Error found in file main.py
```

Check Your Knowledge Answers - Task 4

Question 4a

The error is on line 2 as stated in the error message.

Question 4b

The error message informs us there is no such function as **background**. The coder has misspelt the function name. To fix the error we need to change **background** to **background**.

Question 4c

The error message informs us that we are missing 1 positional argument to the `setCanvasSize` function on line 1. Looking at the documentation for the `setCanvasSize()` function tells us we need to provide a width and a height. Hence we can fix the problem by ensuring we pass both the width and the height to the function. Here's an example of how we could fix the bug:

```
setCanvasSize(600, 400)
```

Task 5 - Comments

What are Comments?

Comments in a computer program are lines of text in the code that are ignored by the computer. They are added to provide information to us humans as to what the code is doing, or why it was written a certain way.

To add a comment in a Python program we use the # (hash) symbol. The following program has a comment on line 1:

```
# Set the height to 600 and width to 400  
setCanvasSize(600, 400)
```

When should you use comments?

The comment above is not very helpful because the `setCanvasSize()` function is fairly self-explanatory as to what it does. However, when it is not clear what a program is doing then comments are extremely helpful. This is especially true when you revisit a program after a week or month. Here's an example of a program that should have comments. Can you tell what it does by reading the code?

```
setCanvasSize(400, 400)  
background(0, 0, 0)  
fill(164, 116, 73)  
noStroke()  
rect(100, 20, 40, 120)  
fill(0, 255, 40)  
circle(120, 200, 80)
```

Now, here is the same program but with 2 comments added. Can you tell what the program is doing now?

```
setCanvasSize(400, 400)
background(0, 0, 0)

# Draw the tree trunk a brown colour
fill(164, 116, 73)
noStroke()
rect(100, 20, 40, 120)

# Draw the leaves a green colour
fill(0, 255, 40)
circle(120, 200, 80)
```

This program is drawing a tree. We also know what the `rect()` function is doing (drawing the tree trunk) and what colour is represented by 164, 116, 73 (brown). In addition, we know what the `circle` function is doing (drawing the leaves) and that the colour 0, 255, 40 is a kind of green.

Hopefully this example shows you how useful comments can be. Whenever you think it would help someone understand what your code is doing you should leave a comment.

Task 5 Challenge

Your task is to create a drawing of a face including at least the eyes, pupils, and a mouth. Ensure your code is commented so that another person could easily identify which section of code is drawing which part of the face.

Check Your Knowledge - Task 5

Question 5a

How do you create a comment in Python?

Question 5b

When should you use a comment in your code?

Question 5c

Do comments affect how the computer runs your program?

Check Your Knowledge Answers - Task 5

Question 5a

To create a comment in Python you use the # (hash) symbol. Here is an example of a comment:

```
# Draw the eyes of our character
```

Question 5b

You should use a comment in your code when it helps explain to another coder (or yourself) what the code is doing. This makes the code easier to maintain or modify in the future.

Question 5c

No. Comments do not affect how the computer runs your program. Comments are only for humans to help them identify what the code is doing.

Task 6 - Draw Your Own Picture

Task 6 Challenge

Your task is to create a drawing using the basic shapes available in PyAngelo (rectangles, circles, ellipses, triangles, arcs, lines, and quads) and bring it to life using code! But before you start coding, draw your picture on the grid below. This will help you identify the correct coordinates for your shapes. Good luck!

