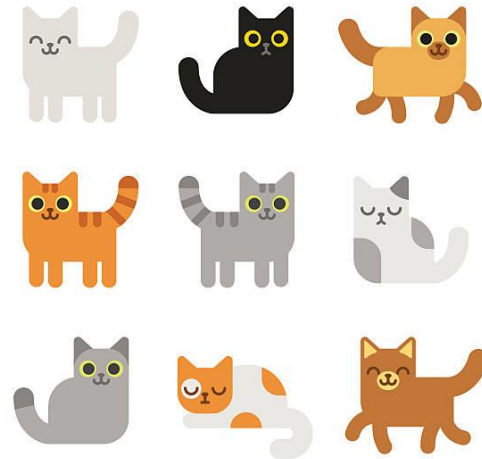


Projects for Level 1B

1. Screen Pet

Everyone loves animals, right? What about having your very own digital pet? Use Python Turtle to create an image/animation of any character/animal you wish. Give your pet **characteristics** and **behaviors** that can be: prompted by the user, affected by the environment, or that go in regular cycles. Use the **"Adding Images to Trinket"** link on the course page.



Example: Digital Kitten:

- Kitten could be idle, sleep, and walking around the screen
- Could be blinking or wagging tail while idle.
- If left idle for too long the Cat might stretch and move a bit.
- The user should be able to interact with the cat by typing in words or clicking on screen. For example: if you pet the cat it purrs, ignore the cat it meows, touch it's tail it hisses, feed the cat it eats.

Don't limit yourself to the example. Your pet doesn't have to even be an animal. Be creative in how you want the pet to behave, interact with the user, and interact with the environment

2. Rock Paper Scissors



Create a text-based version of the classic Rock, Paper, Scissors game in Python.

1. Have the user select one of the three items.
2. Have the computer randomly select one of the three.
3. Have the computer compare the selections and declare the winner!
4. Have the game repeat until the user chooses to stop.

Maybe even try adding Graphics...

Try using the **"Adding Images to Trinket"** link on the course page.

3. Text based Quiz or Adventure Game

Create a text based game or quiz that allows a user to interact with the computer in a way that is engaging as possible.

Quiz Examples/Criteria:

- Math quiz
- Python quiz
- Any quiz in any subject area even trivia about your favorite show/music/movie
- You should use the **random** function in your quiz.



Game Example/Criteria:

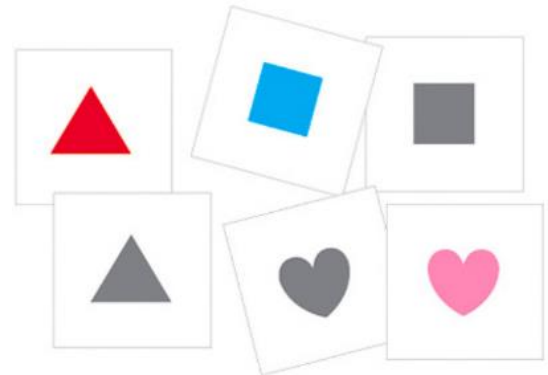
- Create a game based on a story about a wizard that has to rescue their friend.
- The game must involve: player choice (input), varied outcomes (including random outcomes, and use python's **random** function)
- Include ways for characters to **obtain/lose objects** or attributes: gold, energy, food, weapons.
- Include a points system and explicit way to "win".
- Game should have a **clear end goal** for the game.
- Include **elements of a story** that keep the player engaged.



4. Snap Game

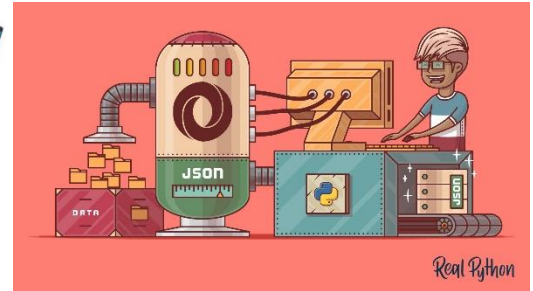
Criteria for the game:

- This a two player game.
- Randomly colored shapes appear on the screen and disappear. **When two identical shapes of identical color appear on the screen at the same time**, the players attempt to press their own designated key on the keyboard **before either shape disappears or the other person presses their key**.
- You get 1 point if you press your key correctly for identical shapes.
- If you click and you are mistaken - 2 point are taken off your score.
- The player with the most points at the **end of the allotted time** is the winner.
- Note: the shapes should: appear, disappear and then be replaced in a random fashion. Try starting with no more than 6 shapes appearing on the screen at one time and no less than 4.



5. Message Encoder/Decoder

You and your business partner the head of a technology company and you want create a way to send secure sensitive information to your managers. Create a program that will do the following:



- Allows you input a string (sentence)
- Convert that sentence into an encrypted string/list by:
 1. Swapping each character with the one next to it.
Then...
 2. Reversing the new string.

Example: The cat in the hat:

1. hT eac tnit ehH ta
2. at hhe tint cae Th



Encode-Decode Data with Python

- The program should also be able to **decode** the encoded messages by reversing the process.

You might have to look up/use the following:

1. Swapping elements of a list/string
2. The **reverse()** method in python
3. For loops

6. Analog Clock animation in Python Turtle



Simple explanation here:

- Create an analog clock animation similar to the one you see the image above
- The clock doesn't have to be very detailed, **but it should have a second hand that moves around the face once every minute.**
- The hour hand and minute hand **don't** have to move (bonus if you wish).
- There should be 12 identifiable positions (preferably with numbers) to indicate the hours on the clock.

7. Reading a CSV file

- One common way for programs to received data is through a **CSV** file
- CSV stand **for Comma Separated Values** and comes in a form similar to this:

```
Steve, 4D, 50  
Paula, 4A, 44  
Pingu, 4B, 90  
Angela, 5A, 20  
Heather, 3D, 99
```



- Create a program that can read a and organize a CSV file that you have created. Your CSV file should include: music artists, what country they are from, a current song they have on the charts, and what position their song is ranked on the charts:

Example: Drake, Canada, Rappy Rap Song, 4,
Metallica, United State, Unforgiven, 102,

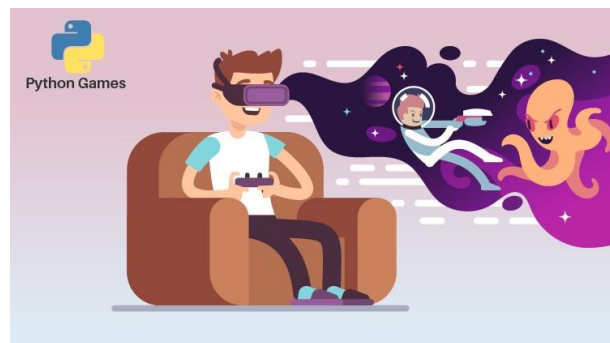
- Once the file has been read:
 1. Display the artists by alphabetical order.
 2. Display their songs rank by their position on the charts (show the song, name, position, and artist on your chart list.

See the link for CSV files and Trinket on the course page. <https://trinket.io/python/984156ebbd>

8. Falling Turtle Catcher Game

Expand on the turtle bird game presented in the 3rd assignment on the course page. In this new version:

- Have **multiple** turtles (or objects falling down from the top of the screen)
- The user/player will control a separate turtle/object at the bottom of the screen and "catch" as many turtles they can (for points).



You may think about adding the following features (or any others you wish):

- Turtles falling increase speed as time goes on.
- Turtles accelerate as they fall.
- Add your own images. Add your own background.
- Add a title screen where players can choose avatar.

9. Bouncing Ball



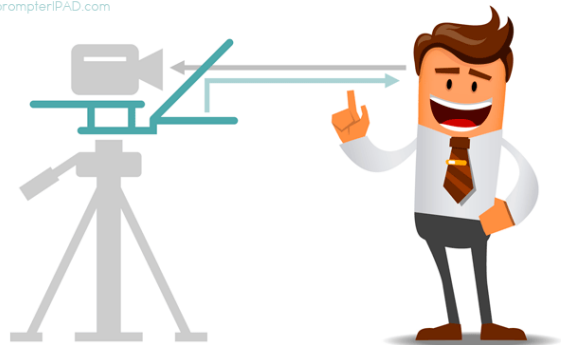
Use Turtle to create an animation of a bouncing ball. Your animation should have the following characteristics:

- Be as realistic as possible
- Ball should be released from a height Slow to a stop (as ball loses energy)
- Have both x and y components to its motion (moving forward as well as up and down)
- Should not be cut and paste from the internet.

10. Teleprompter

People who give speeches or work in live television sometimes have to use a teleprompter. This device electronically scrolls text up screen *in a slow and uniform way* so a person can read the text without having to look down at notes.

www.teleprompterIPAD.com



Example: <https://www.youtube.com/watch?v=8XJqUcOfsAw>

Create a teleprompter in trinket. You can use trinket turtle or any other method you wish.

11. Master Mind GAME (Correct and Almost)

Randomly generate a 4-digit number. Ask the user to guess a 4-digit number. For every digit that the user guessed *correctly in the correct place*, they have a "**Correct**". For every digit the user guessed correctly *in the wrong place* is a "**Almost**". Every time the user makes a guess, tell them how many "cows" and "bulls" they have. Once the user guesses the correct number, the game is over. Keep track of the number of guesses the user makes throughout the game and tell the user at the end.

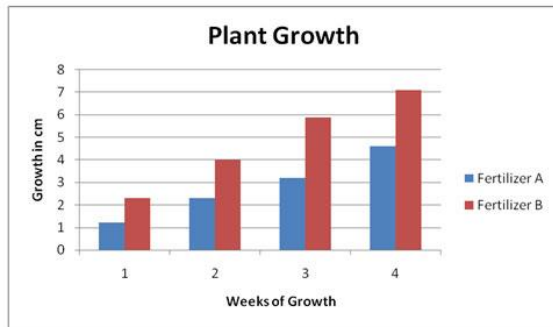
Say the number generated by the computer is 1038. An example interaction could look like this:

```
Welcome to the MasterMind Game!  
Enter a number: 1234  
2 Correct, 0 Almost  
Enter a number: 1256  
1 Correct, 1 Almost
```

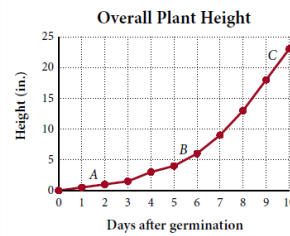
...Until the user guesses the number.



12. Graph of Plant Growth



Day	Plant Height
0	0
1	0.5
2	1
3	1.5
4	3
5	4
6	6
7	9
8	13
9	18
10	23



Design a program that can take input from a user and display it in a graph format.

- Your graph can be a Bar graph or a Line graph.
- Your input data can be anything you wish (plant growth is just one example).
- Your graph should have a title and both axis properly labeled with correct units.
- Your program should welcome the user and have an easy input interface that explains to the user exactly what to do and what should happen as a result.

13. Race Track Design

In this challenge, we will use Python Turtle to create a program that allow the user to draw/design a racing track.



Your program should welcome the user and then ask them how to draw each section in order.

Example: For section 1, do you want the track to go straight(s), turn left(l), or turn right(r)? s
 For how long? 40
 Great! For section 2, do you want the track to go straight(s), turn left(l), or turn right(r)? s

Your program should at least the following three **functions**:

straightLine(distance): To draw a straight line. The distance is given as a number of pixels (e.g. 100).

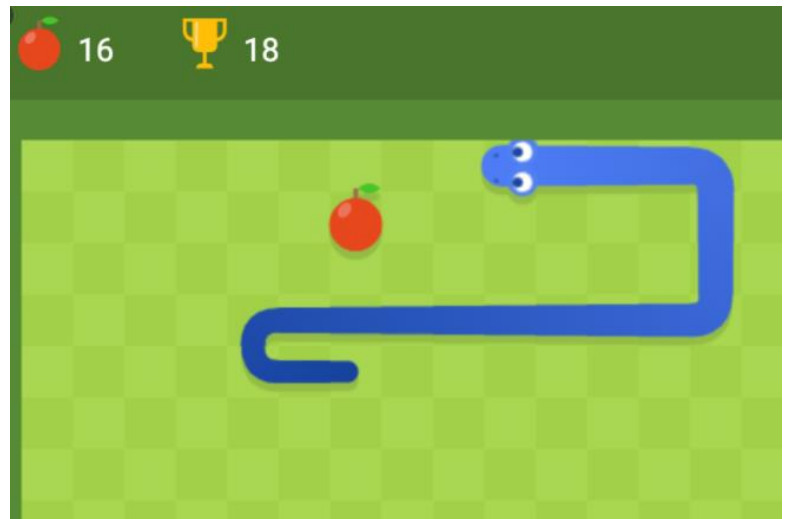
turnLeft(angle, radius): To draw a turn/arc of an angle in degrees (e.g. 180 for a u-turn), and a radius to represent how sharp the turn is. (Sharp turn = low radius e.g. 10).

turnRight(angle, radius): Same as above but to turn to the right.

Your program should make sure the track is one continuous loop (even if the user doesn't do this). Make it look as cool as possible.

14. Snake

Create the classic game of SNAKE using Turtle graphics. There are several tutorials online for how to do this. If you decide to use one. You MUST reference the tutorial in your submission, or you will lose marks. Also, don't just cut and paste from tutorials. You can get an idea how to make things work using sample code and then try on your own. Make sure you are learning something.



Connect 4!

Create the classic super fun game Connect 4 using Python's Turtle graphics.

Not sure how it works? Check out the following video:

<https://www.youtube.com/watch?v=utXzIFEVPjA>

Your game must be a two player game.



Two Player Tank Battle

Create a 2D, two player tank game where two players are allowed to control a tank on a field. The game should have at least the following features:

- Each tank should be able to shoot shells at opposing players. If the other player gets hit 3 times they lose.
- Both tanks should be able to move forward, rotate 360 in either direction, and shoot in any direction.
- There should be some obstacles that the tanks cannot go through or shoot through.
- The goal of the game should be to destroy the other player or capture the other players territory by rolling into a specific zone on the other players side of the screen.
- You may add, subtract, or adjust any rules you wish to improve the games enjoyment.

