

Programming

MAGIC OF mentorSHIP!

- In collaboration with Athena SD, we've developed a forum for you to ask questions to professional women already paving the way in STEM!
- Go to the link below:
<https://www.allgirlsstemssociety.org/magic-of-mentorship.html> to post questions you have about being a woman in STEM, careers in STEM or really anything you'd like!
- Let us know if you have questions :)



Name THAT TURTLe!



Prizes



AGENDA

- Intro to Computer Science
- Mission #1: Hello World!
- Mission #2: Adventure Time
- Mission #3: BLAST OFFFFFFFFF
- Mission #4: We love turtles <3
- Snack break :)
- Mission #5: Turtles drawing shapes <3
- Mission #6: Turtles drawing houses <3
- Choose your own adventure!



DO YOU HAVE PYTHON?

Raise your hand if you are not sure and a volunteer will come help you out before we start coding.

Library Password: makestuff

Python 3.7 Idle



WHAT IS COMPUTER SCIENCE?



THE STUDY OF COMPUTERS... AND?

1. Study of problems, problem-solving, and the solutions that come out of the problem-solving process
1. Computers are just tools
2. Hardware vs. Software



WHAT IS THE GOAL OF COMPUTER SCIENTISTS?



WHAT IS THE GOAL OF COMPUTER SCIENTISTS?

1. To develop an **algorithm**!
 - a. What are elements of a good algorithm?
2. Algorithm: a step-by-step list of instructions for solving a problem→ the solutions
3. Need precision, creativity, and reasoning to solve problems



WHERE DO WE SEE COMPUTER SCIENCE IN REAL LIFE?



COMPUTER SCIENCE IN REAL LIFE

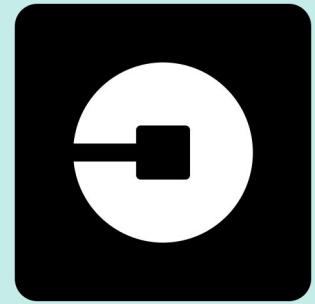
IT'S EVERYWHERE!



computer languages

1. Many types of different computer languages
2. Every language has a different syntax
 - a. Syntax - specific way of writing code
3. Programmers write code in this language, which is then translated for the computer to understand
 - a. We count in 10s, computers count in 2s!

PYTHON IN REAL LIFE



CRAZY RIGHT?

WHAT IS an ALGORITHM?

Step-by-step list of instructions for solving a
problem → a solution

WHAT IS SYNTAX?

Specific way of writing code: order, phrase, etc.

EXAMPLE OF APP made POSSIBLE BY PYTHON

Google, Instagram, Spotify, Netflix, Uber

ADA LOVELACE

1. 1st computer programmer!
2. Translated the “Analytical Engine” paper with extensive notes in 1843
 - a. Including description of 1st algorithm - calculating Bernoulli numbers!
3. If the engine could manipulate numbers, these numbers could represent other entities
 - a. First to express potential of computers outside of mathematics

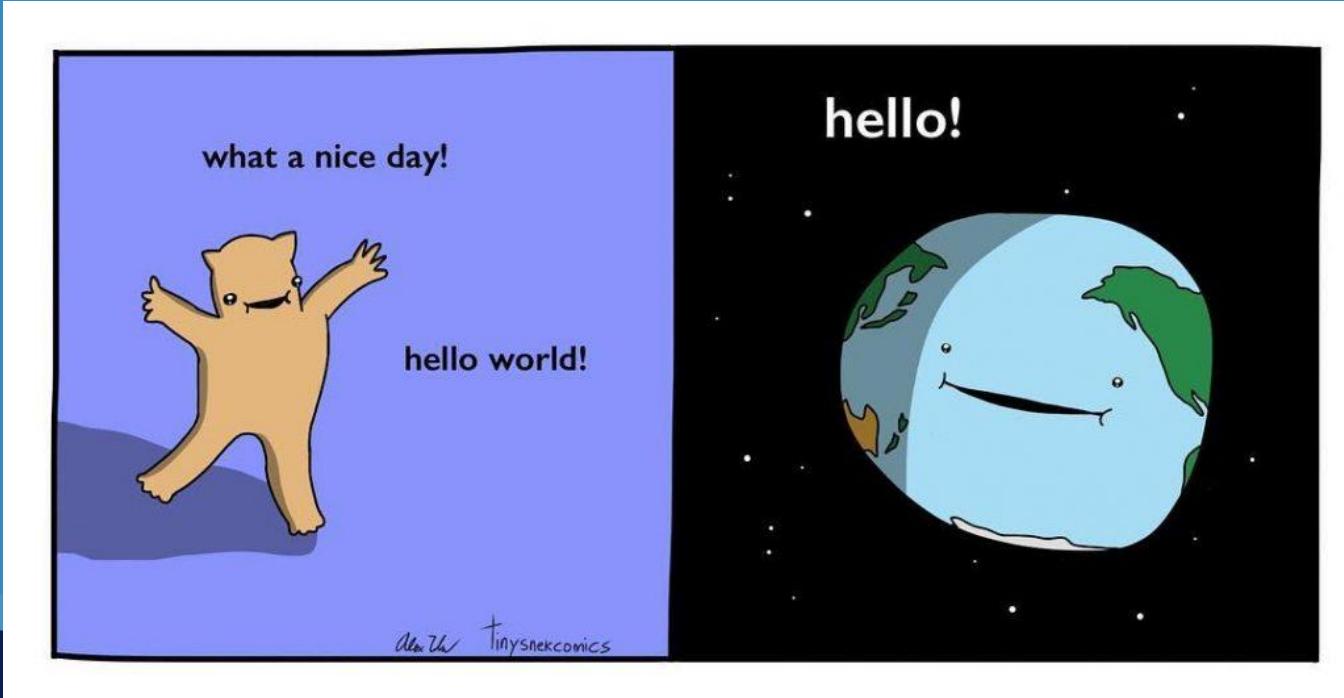


Name THAT TURTLE!



LET'S START CODING!

The Initiation Task all Programmers MUST complete: Hello World!



WHAT IS a STRING?

- a string of characters
- enclosed by either double or single quotation marks
 - “Kasie likes Drama Llamas.”
- Can be saved to variables to be used more than once later
 - `llamaLover = “Kasie likes Drama Llamas.”`



IS THIS a STRING: “ ” ?

Yes: Surprise! It's called an empty string

MISSION #1: HELLO WORLD

Example Code:

```
worldGreeting = "Hello World"
```

```
print(worldGreeting)
```

```
print("Hello World")
```

```
print("This is MAGICAL! I'm officially Hermione Granger")
```



REMEMBER: (parentheses) and “quotes” and camelCasing

HOW TO RUN a PROGRAM

1. Open up the python IDLE application
2. Open a file by clicking File, New in the top left corner
3. Type up your algorithm
4. Save the program and name it someName.py
 - a. Type command s or control s, or find it in top left corner
 - b. .py is like a label for the computer to know it's coded in python
5. Next, press “run module” in top left corner
 1. See if your program works!
 2. If not, edit your code, save, and try again!

DON'T BE AFRAID TO RAISE YOUR HAND FOR HELP

COMMENTING FOR LIFE

- A way to comment and make notes on your code
 - Helpful for organizing long programs and for collaborating with other programmers
- Use # so that the computer knows to ignore that line

#coding

#STEMisagirlthing

USER INPUT

- User input allows user to enter an answer (string or number) into the python shell
- Save it into a variable to be used later
 - `miraAnswer = input("Is Emma's favorite hero Groot?")`
 - `miraAnswer = input("What is Emma's mental age?")`



BOOLEANS

- A value of True or False
- ALWAYS capitalized
- Can save boolean to a variable
 - emmaIsCool = True
- You can use ==, <, >, <=, or >= in order to check the relationship between two things like numbers or strings
 - kasieIsADramaLlama = (2 >= 5)
 - emmaIsAMamaLlama = (1 == 1.01)
 - miraIsNoLlama = (1 < 7)



CONDITIONAL IF STATEMENTS

- If statements use booleans and work exactly how they sound!
- Basic format

```
if someThing:
```

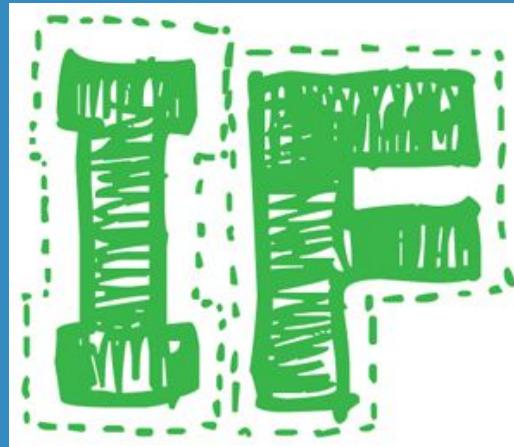
```
    Do this
```

```
elif somethingElse:
```

```
    Do this
```

```
else:
```

```
    Do this
```



- You must have the correct indents and colon after the if statement
- What happens when more than one are true?

MISSION #2: ADVENTURE Game!

Example Code:

```
answer = input("Would you like to go to Hogsmeade, Forbidden Forest, or Diagon Alley?")
if answer == "Hogsmeade":
    print("Oh no, Voldemort appears. You must run away!")
elif answer == "Forbidden Forest":
    print("You meet a centaur. He's pretty chill, I guess.")
elif answer == "Diagon Alley":
    print("Ahhhh! A broom chase between Harry and Draco!")
else:
    print("Invalid Location, ya silly llama!")
```

Name THAT TURTLE!



© Disney/Pixar

FOR LOOPS

- Used to simplify the code when you repeat the exact same action many times
- Basic structure

```
for i in range(a, b, c):
```

Do this

- a = starting value, b = limit/where to stop, c = its step
- Must have colon after statement and parentheses for the range!
- What will this print?

```
for i in range(1, 5, 1):
```

```
    print(i)
```

MISSION #3: BLAST OFF!

- Example Code:

```
for i in range(10, 0, -1):
    print(i)
print("BLASTOFF KAPOW!")
```

- Try changing the values in the range and see what happens

- REMEMBER:** colon after for statement, parentheses for print statements, and quotes for strings!



CREATING A TURTLE

In order to create a turtle object and use its methods (actions), you **MUST** import the turtle class.

```
from turtle import *
```

To accomplish a method, type the turtle name, followed by a dot, and finally by the method with parentheses.

```
squirt = Turtle() # creates turtle  
squirt.shape("turtle")
```



MISSION #4: creating a TURTLE



```
from turtle import *
```

```
home = Screen()
```

```
home.bgcolor("light blue")
```

```
home.screensize(800, 800)
```

```
kasie = Turtle()
```

```
kasie.shape("turtle")
```

```
kasie.color("hot pink")
```

```
# creates a graphics window to draw in
```

```
# sets background of window to blue
```

```
# sets window size to 800 by 800 pixels
```

```
# creates a turtle object
```

```
# sets the turtle shape
```

```
# sets pen color to pink
```

Numbers in between parentheses are parameters. They tell the method necessary values to perform the action.



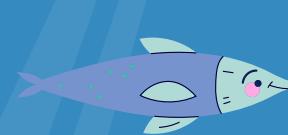
Name THAT TURTLe!



Break TIME!



Name THAT TURTLe!



BASIC TURTLE METHODS

- `turtleName.forward(integer distance for turtle to travel forwards)`

- `turtleName.backward(integer distance for turtle to travel backwards)`

- `turtleName.left(amount of degrees for turtle to turn left)`

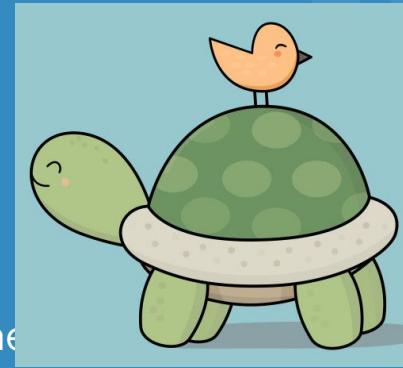
- `turtleName.right(amount of degrees for turtle to turn right)`

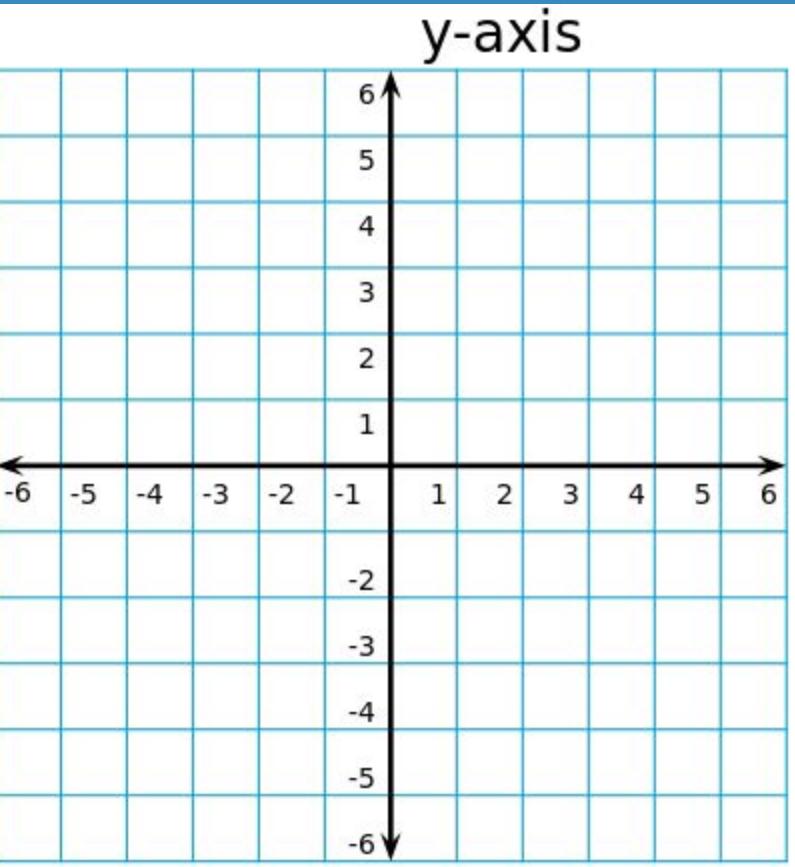
- `turtleName.setheading(angle for turtle's facing direction)`

- `turtleName.setpos(x position, y position) # window is coordinate plane`

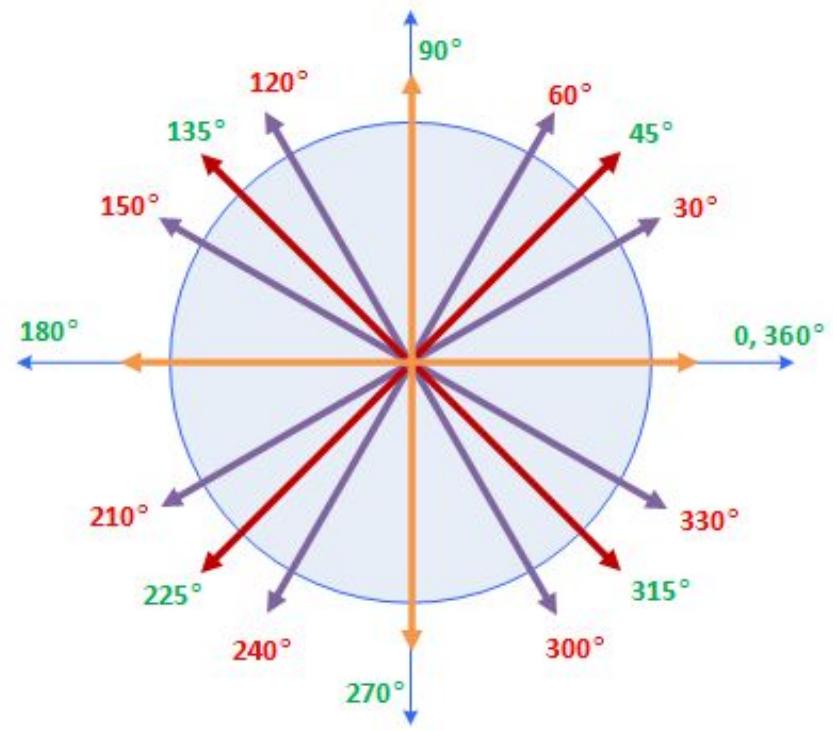
- `turtleName.home() # brings the turtle back to the origin`

- `turtleName.circle(radius of the circle the turtle draws)`





x-axis



MISSION #5: DRAWING A SHAPE

Now with your newfound knowledge on turtle methods, try drawing different shapes! You can use for loops to make repeated actions have less code.

Draw a circle, square, or a triangle.

***Challenge: What about a star?
(6-spiked and 8-spiked are easier)

Raise your hand if you need help :)



More TURTLE METHODS

- `turtleName.speed(integer 0-10)` # sets the speed of the turtle
- `turtleName.dot()` # draws a dot at turtle's location with the turtle's color
- `turtleName.stamp()` # "stamps" turtle's shape at its location
- `turtleName.penup()` # brings "pen" up→ its path doesn't draw anything
- `turtleName.pendown()` # brings "pen" down→ turtle's path draws
- `turtleName.begin_fill()` # called before the shape to be filled is drawn
- `turtleName.end_fill()` # called after the shape to be filled is drawn
- `t2 = turtleName.clone()` # clones the turtle to create another turtle

Name THAT TURTLe!

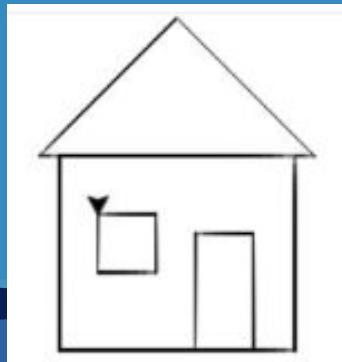


MISSION #6: DRAWING A HOUSE

Look at the Turtle Graphics Guide to experiment with different turtle methods in order to draw a house!

Try to draw a square base, a triangle roof, one door, and one window. Graphics are frustrating, so just do your best!

Raise your hand if you need help. I'll show you a solution in 10 minutes.



- WHAT PHRASE MUST YOU INCLUDE AT THE TOP OF THE PROGRAM?

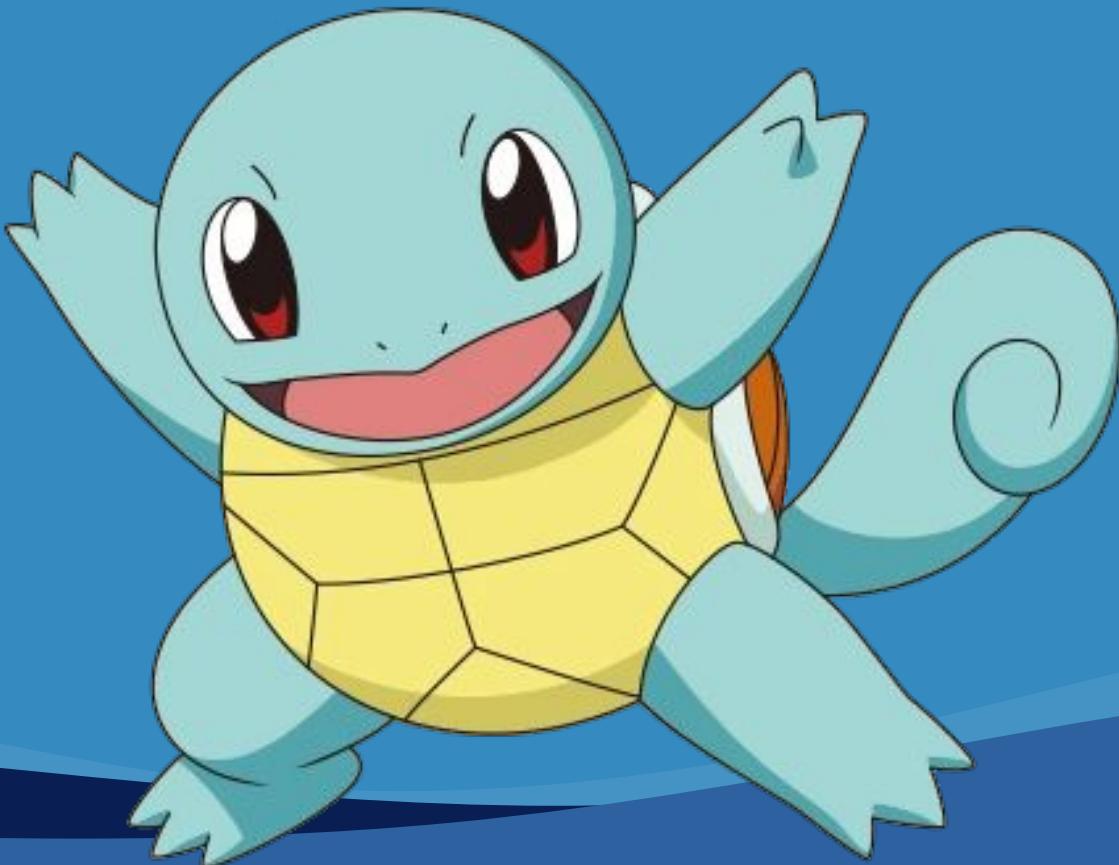
- from turtle import *

WINDOW METHODS

windowName.title(string title)	# gives the window a title at the top
windowName.clear()	# clears entire window including background
windowName.reset()	# resets drawings in window
windowName.bye()	# closes the window
windowName.exitonclick()	# closes window when user clicks on it



Name THAT TURTLE!



MISSION #7: CHOOSE your own adventure

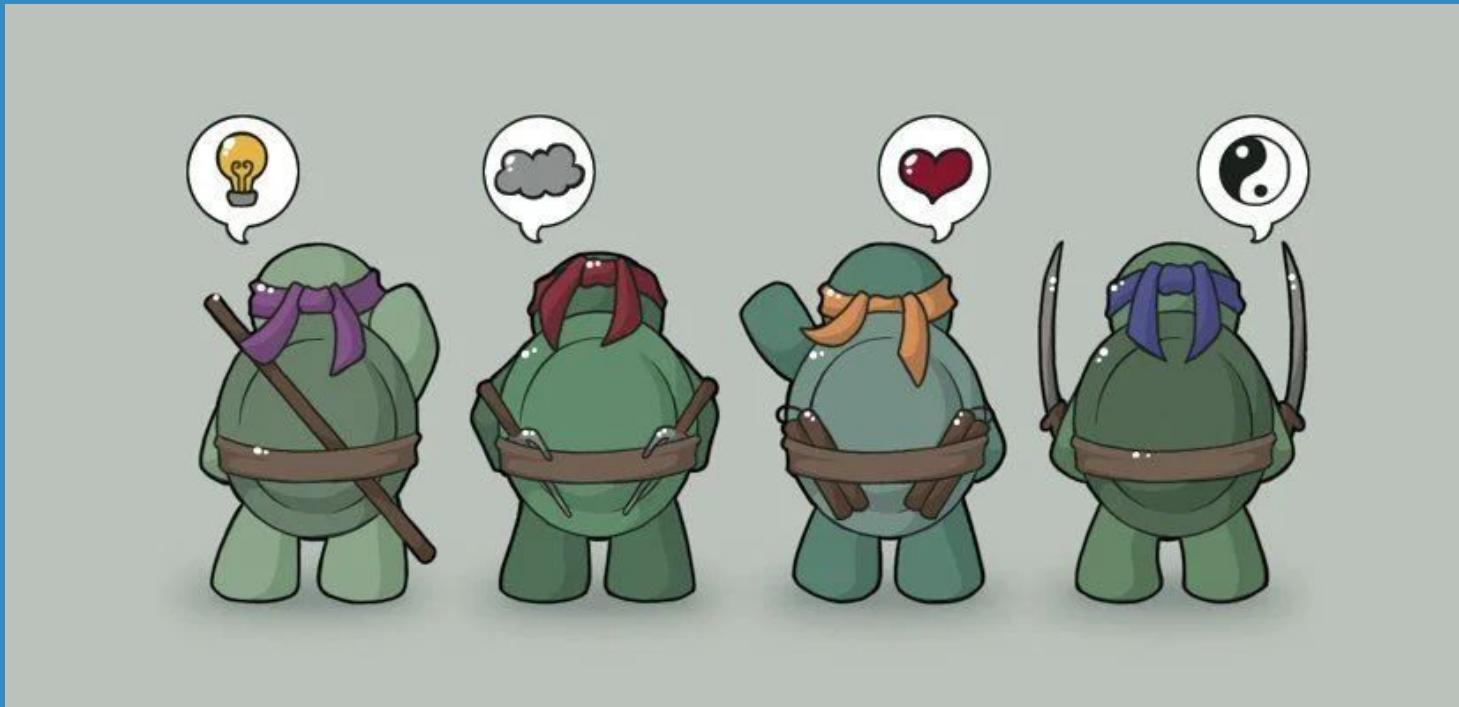
Use the Turtle Graphics Guide to experiment with different turtle methods to draw anything you wish! Try drawing with different shapes, colors, pen sizes. Draw a volunteer, a cool scene, or an animal... anything!

You can also tackle the Extra Challenges worksheet if you dare ;)

Raise your hand if you need any help!



SHARE YOUR FINISHED DESIGN WITH NEIGHBORS!



Recursion

function: a mini program that you can call inside a program to perform an action

- Basically, recursion is when a function calls itself

- In order for this to occur, you need an if statement to test if the base case

- is satisfied. If it is, the recursion should stop calling itself.

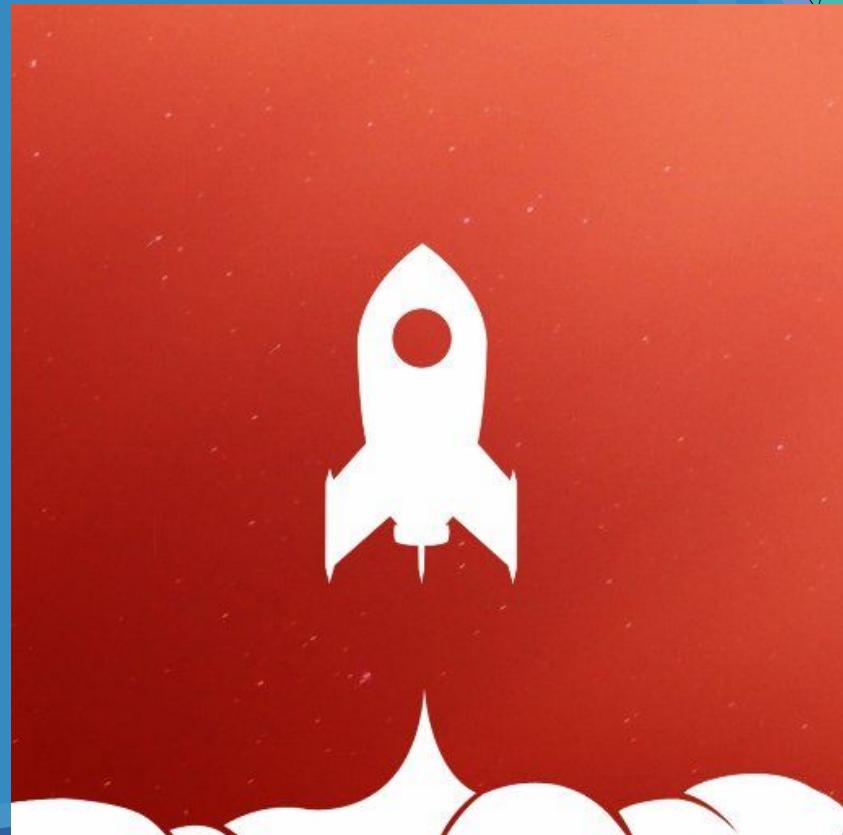
- If not, the function will call itself infinitely until the computer runs out of memory or blows up

Recursion EXAMPLE: BLASTOFF 2.0

```
def recursion(n):  
    if n == 0:  
        print("BLASTOFF!")  
    else:  
        print(n)  
        recursion(n-1)
```

```
def main():  
    recursion(10)
```

```
main()
```



MISSION #8: RECURSIVE FRACTALS CHALLENGE

Super challenging!

Try your own recursion for printing a pyramid of *.

I will show some examples to inspire you.

Super super challenge:

Use recursion and turtle graphics to draw a recursive fractal

ASK FOR HELP IF YOU NEED

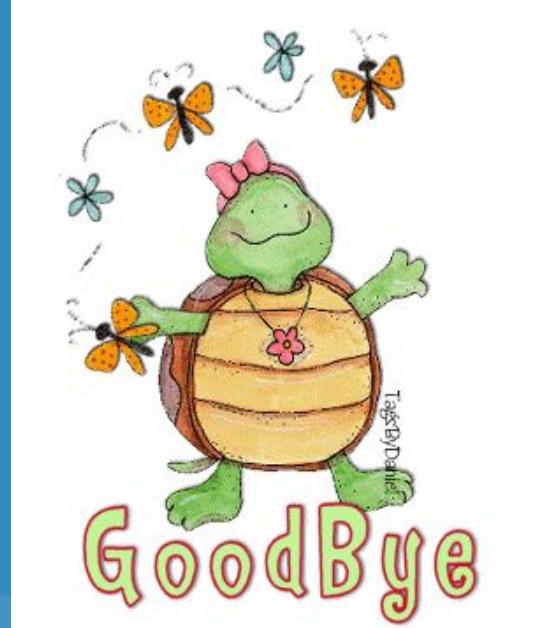
Don't forget the base case!



Twitter: @allgirlsstem
Instagram: @allgirlsstem
Facebook: @allgirlsstemsociety



Please FOLLOW US on SOCIAL media!



THANKS FOR COMING!