

Drawing with Turtles

ATLS 1300

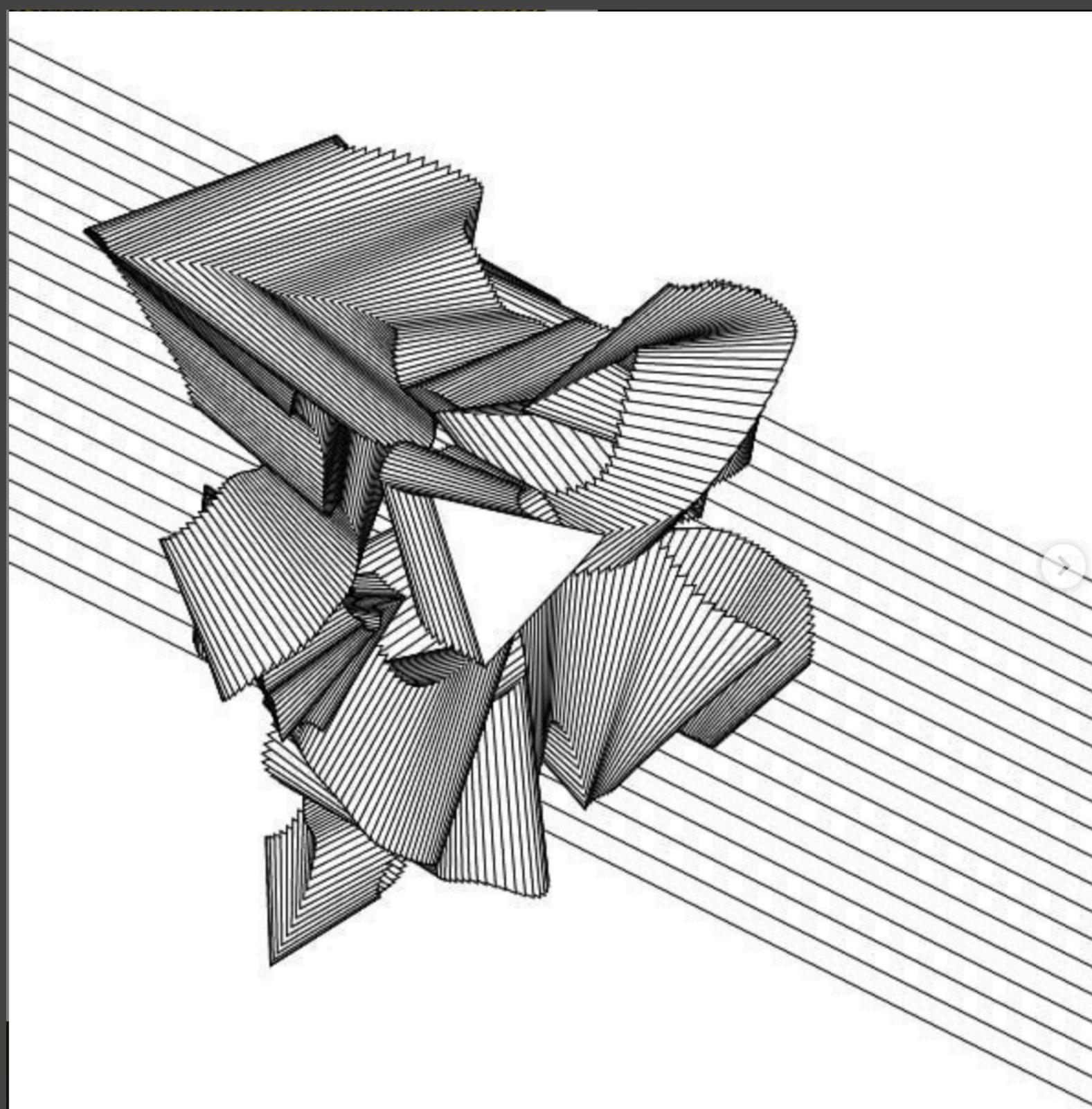
Thurs, Jan 23

Cool Stuff

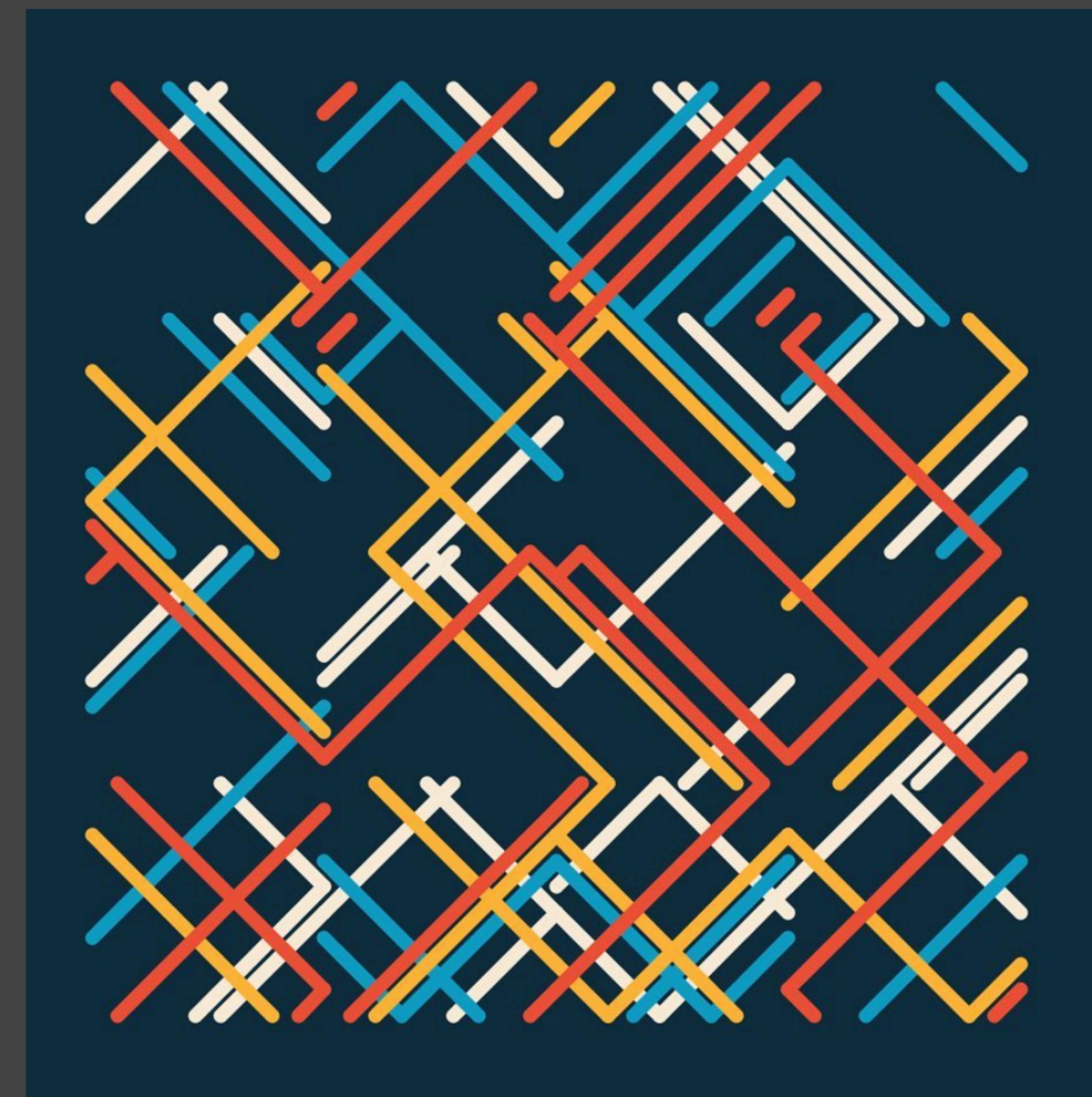
#creativecoding



@iso.hedron



@marcelo.o.r.prates



@iso.hedron

Class Objectives

- Color theory
- Naming variables
- Composition
- Getting started programming
 - Planning with pseudocode

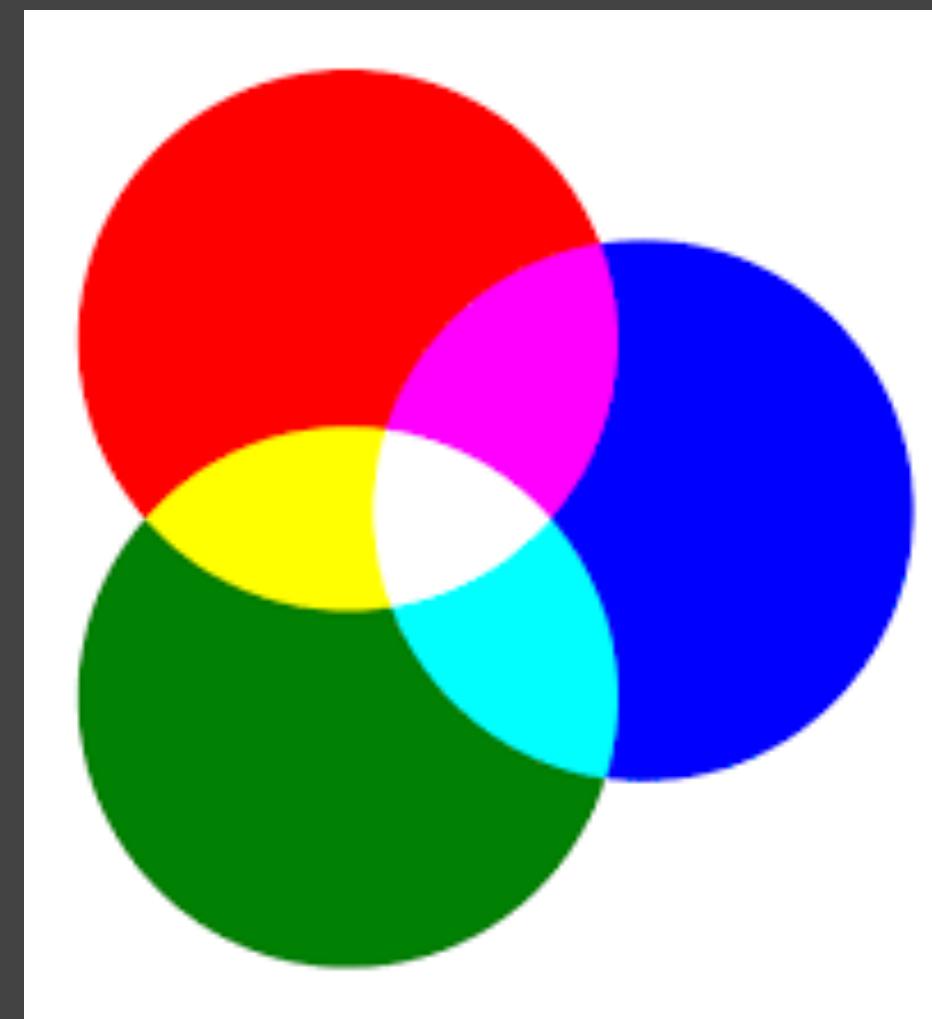
Digital Color theory

Primary digital colors

3 channels: Red, green, blue (RGB)

Values: 0 - 255

(255, 0, 0)



(0, 255, 0)

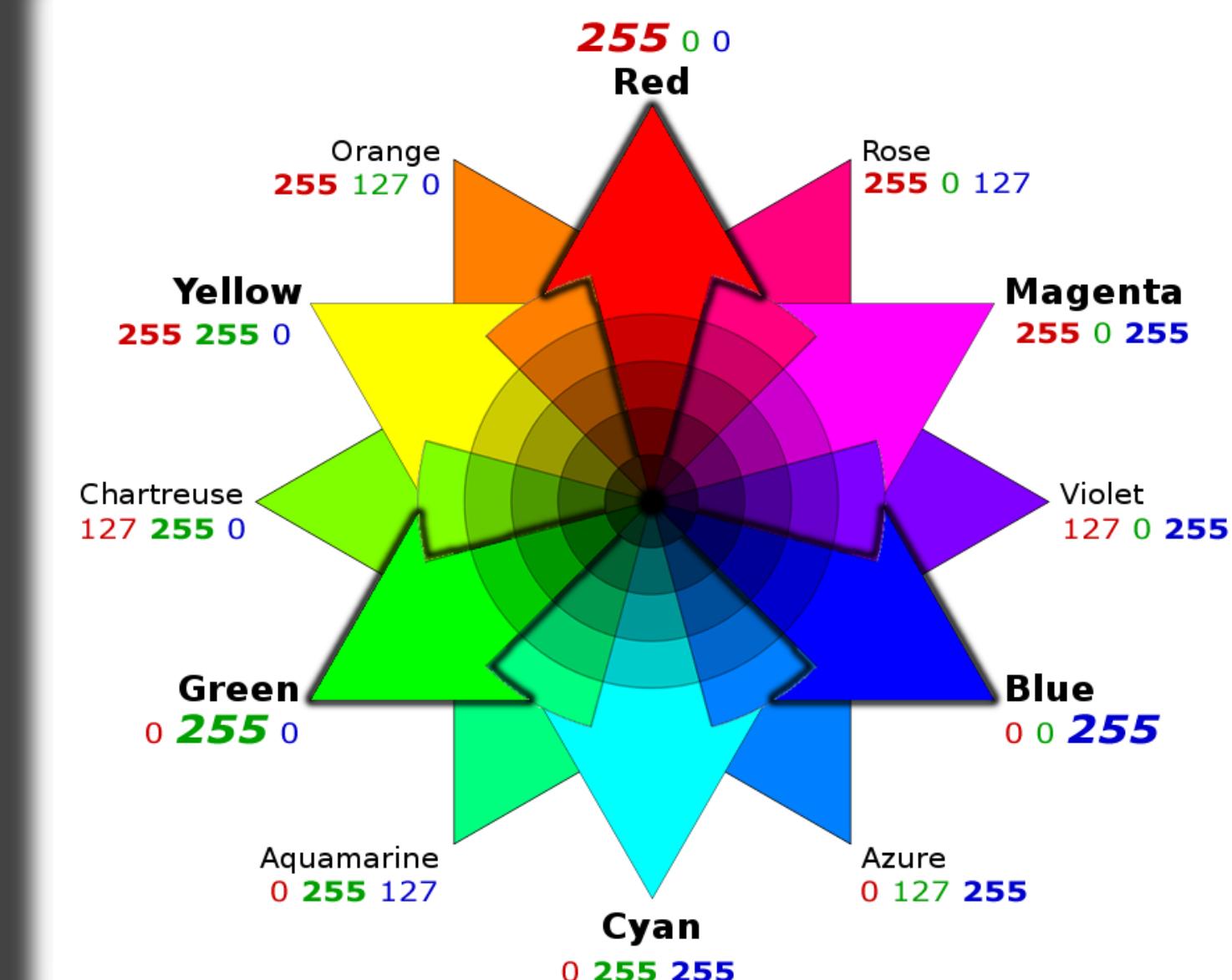
(0, 0, 255)

(0, 0, 0)

(255, 255, 255)

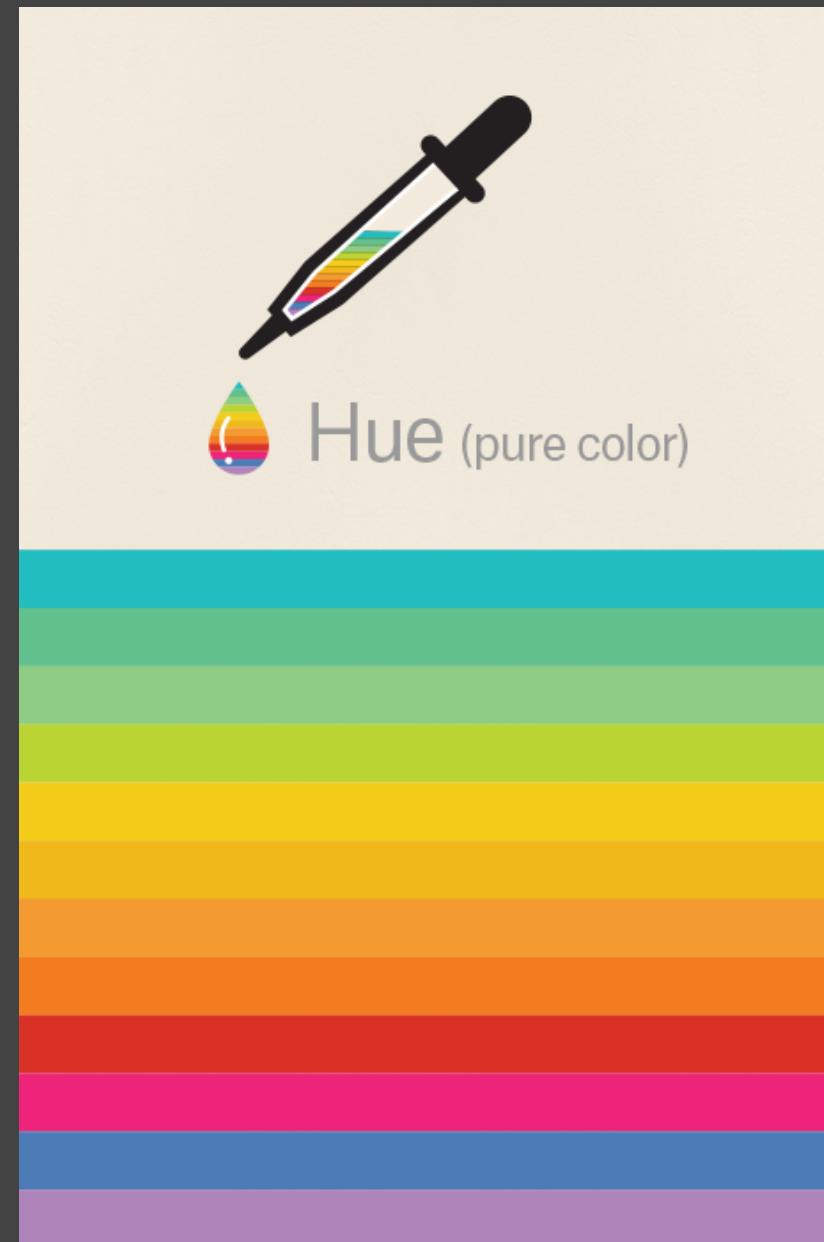
This means the pixel is off

This means the pixel is on



Digital Color Theory

RGB value of color



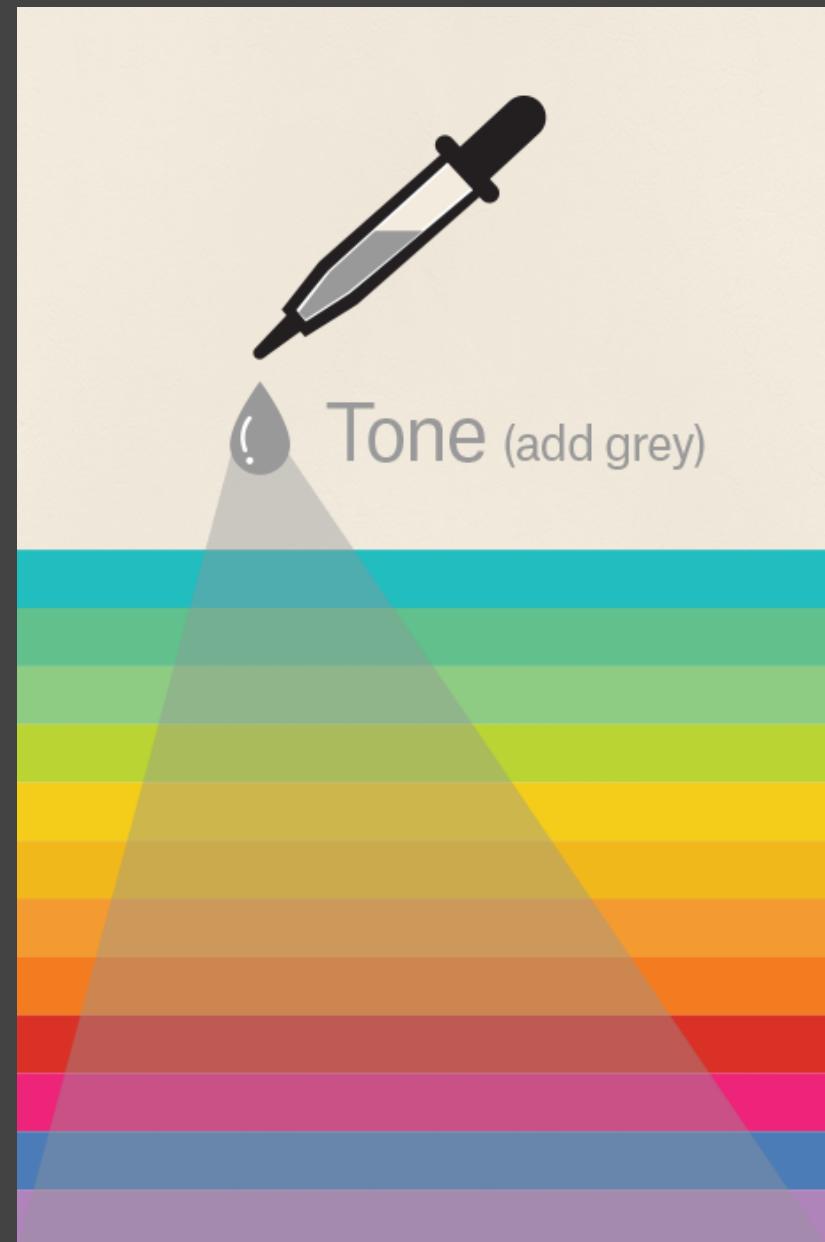
Darkness of color



Lightness of color



Grayness of color



Value across channels

(200, 100, 100)

Channel values decrease

(70, 0, 0)

Channel values increase

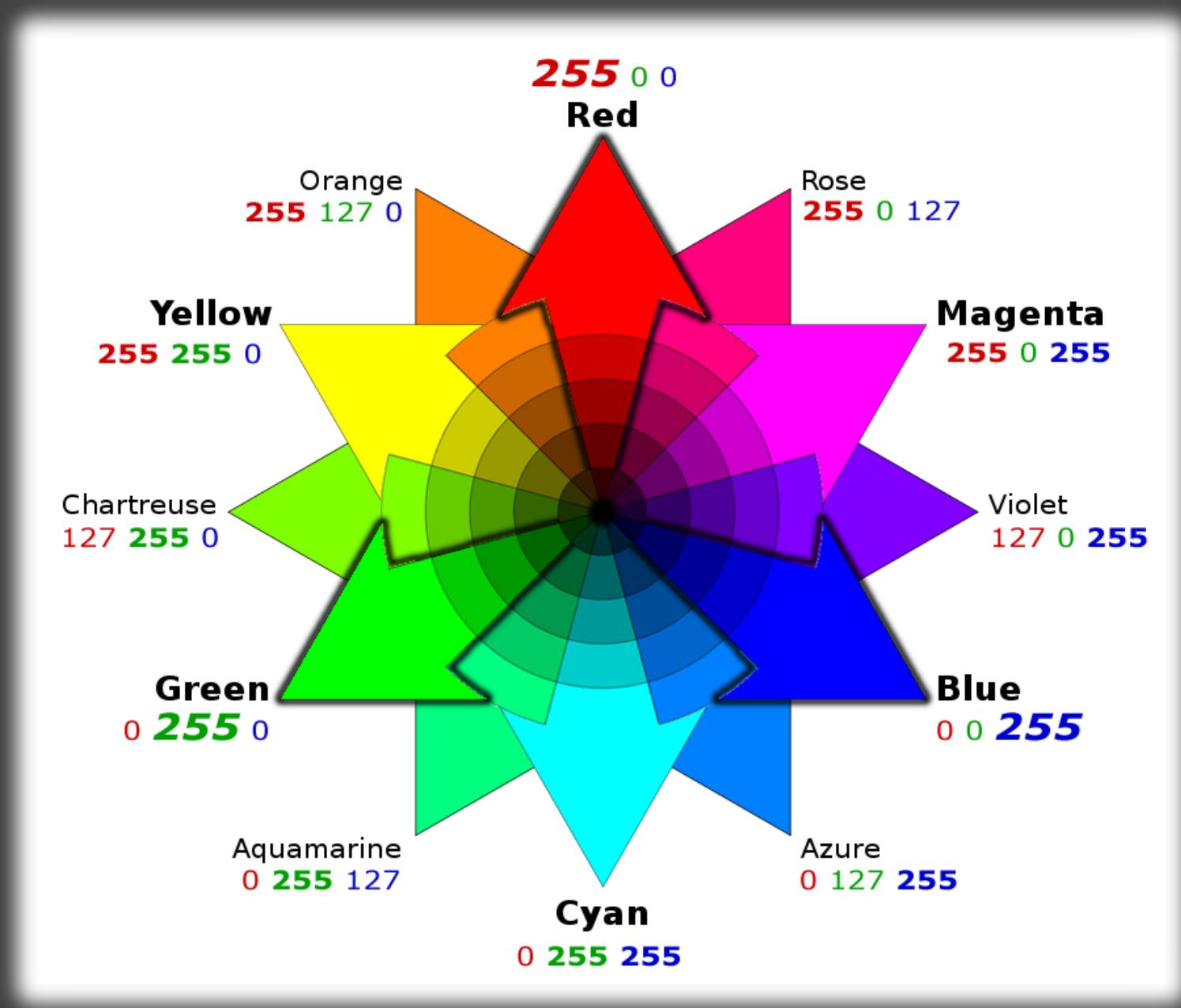
(255, 0, 0)

Channel values even out

(200, 180, 180)

Digital Color theory

Color themes



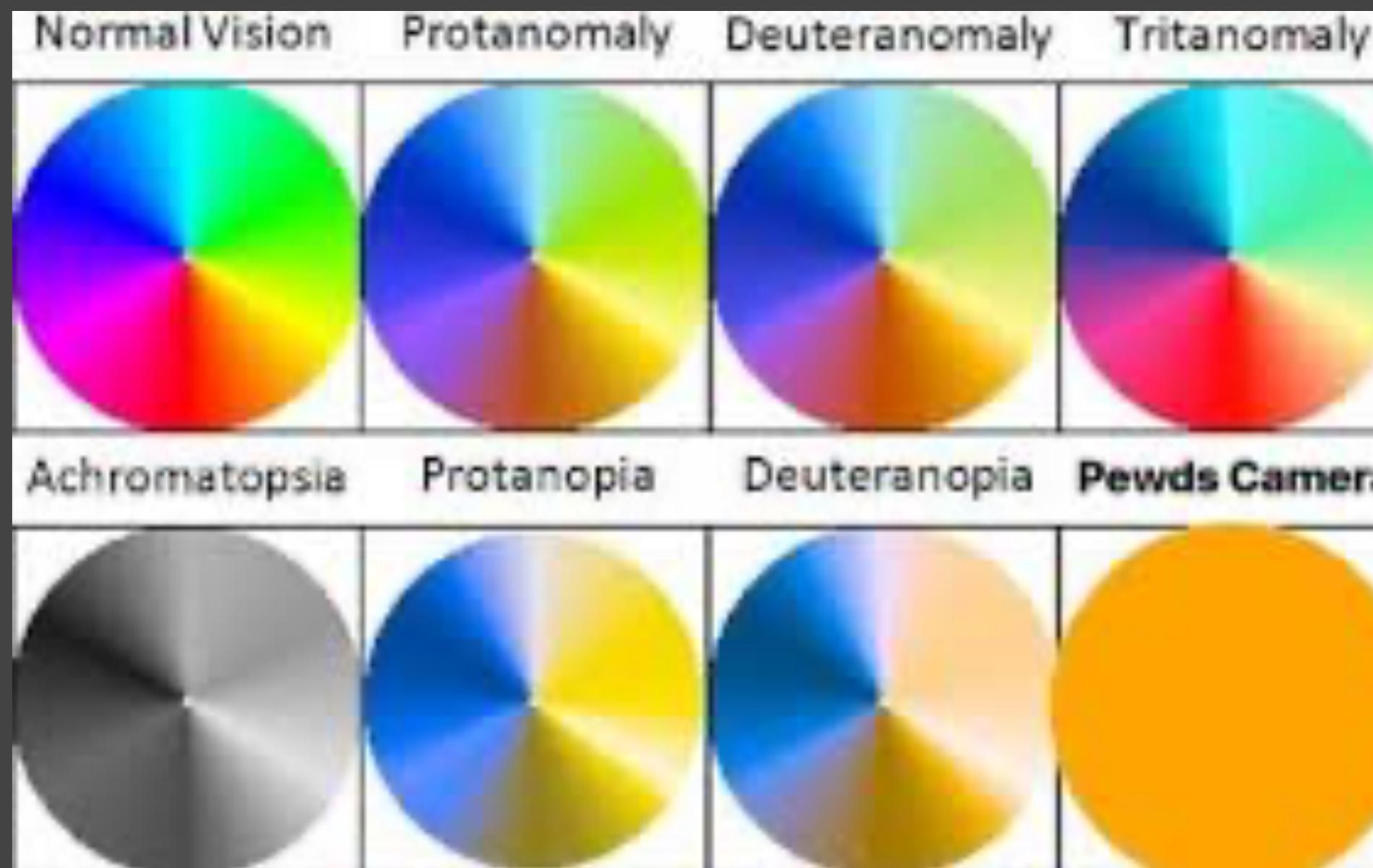
- **Monochrome**
 - Range values along one color channel
- **Analogous**
 - Range values in two color channels
- **Complementary**
 - One channel moves in opposition of other two (R increases, G and B decrease)

Choosing Palettes

Many of these color schemes will be difficult for color blind people (~8%)

Be mindful of how to keep your art inclusive!

coolors.co - automatically chooses colorblind-friendly palettes



Linked in this week's toolkit

The screenshot shows the coolors.co website interface. At the top, there are tabs for 'Generate', 'Explore', 'iOS App', 'Add-on', 'More', 'Login', and 'Sign Up'. Below the tabs, it says 'Press the spacebar to generate color schemes!'. The main area displays a color palette with five squares: dark teal (#OB3C49), light blue (#C9DAEA), reddish-pink (#C89F9C), orange-brown (#C97C5D), and dark red (#AB2346). Each square has a small lock icon and a hex code below it. To the right of the palette, there is an advertisement for Adobe Stock.

Naming Variables

- Single words (no spaces)
- Descriptive (what is the data for, what will the variable do)
- **Case types:**
 - camelCase - don't capitalize first word, capitalize the rest
 - snake_case - no caps, use underscores

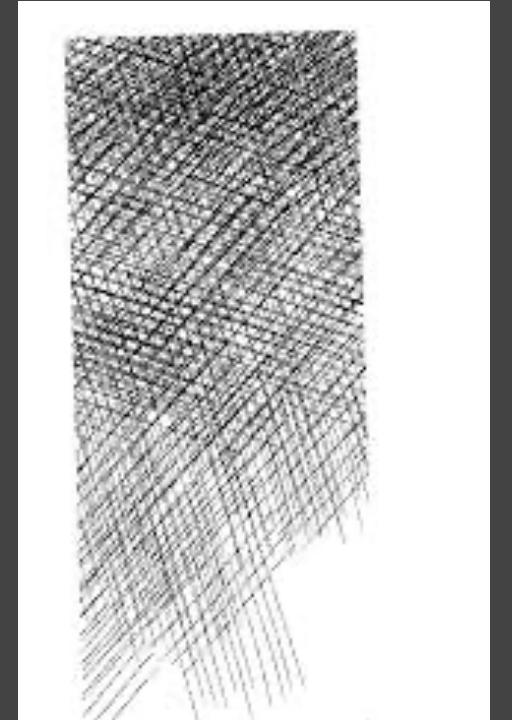
Name your turtles

You have to create two turtles do draw two different components of your piece

- The first turtle is red and will draw spirals
- The second turtle is black and will draw hatches

What are two names that *best* describe the turtles?

- A. red, black
- B. turtle1, turtle2
- C. redSpiral, blackHatch
- D. Barb, Hector



Practice: Let's move your turtles

- `forward(steps)`
 - `backward(steps)`
 - `left(degrees)`
 - `right(degrees)`
 - `up()` – picks up pen (no drawing)
 - `down()` – drops pen (drawing)
 - `circle(degrees)` – draw the outline of a circle (360 = full circle)
- Remember to use dot notation!**
- TURTLENAME.FUNCTION()**

Stopping drawing

- `done()` – stops Python from expecting more draw commands
- `bye()` – closes window when called
- `exitonclick()` – closes window with a mouse click

Variable Types

- Simple variable types
 - Stores single values
 - **Integer, boolean, string, float**
- Complex variable types
 - Stores multiple values

Complex Variable Types

- **List**
 - [value, value, value]
 - Indicate with *brackets*
 - Accepts any simple data type
 - “List of bools”, “list of integers”, “list of strings”
 - Separate values with comma

Complex Variable Types

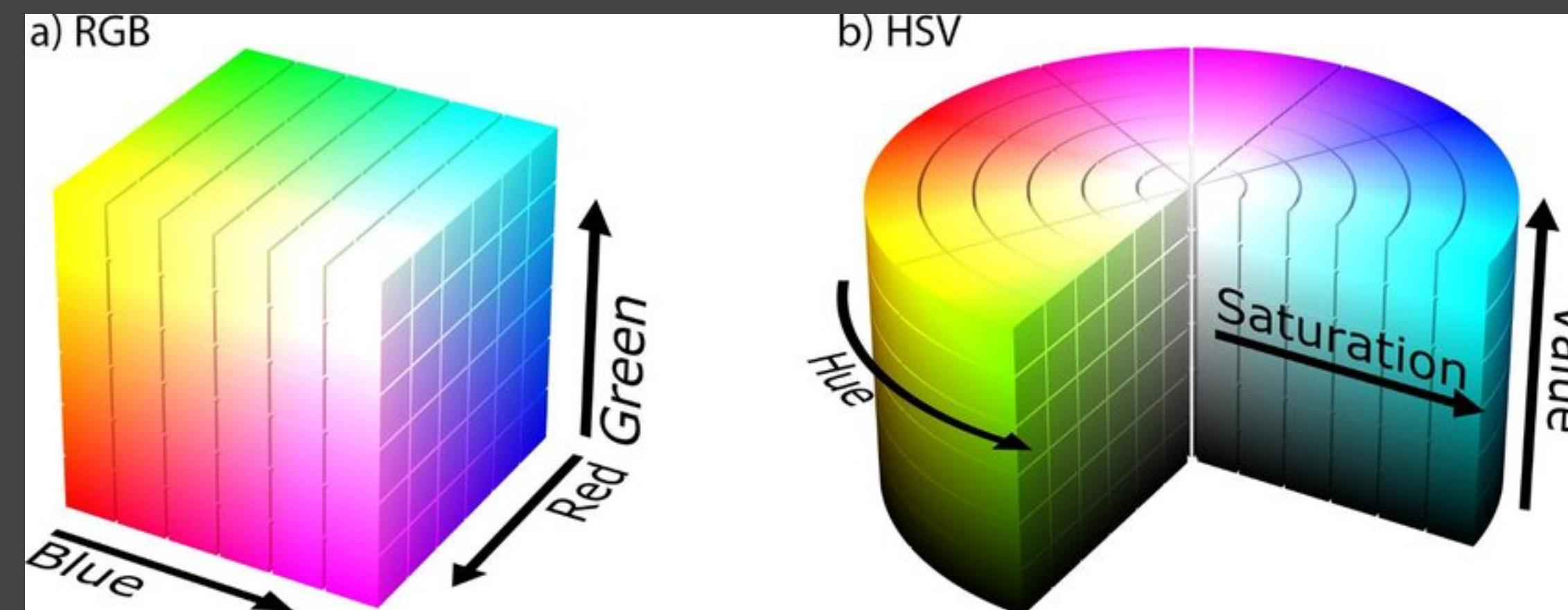
- **Tuple**
 - (value, value, value)
 - Indicate with ***parentheses***
 - Accepts any simple data type
 - “Tuple of bools”, “tuple of integers”, “tuple of strings”
 - Separate values with comma

Turtle color uses complex variables

- Setting turtle colors:
 - `Ari.color()`
 - What type of data gets returned?
 - Returns two values: (**line, fill**)
 - `Ari.color('new color', 'new color')`
 - Python has an **extensive** list of colors, try to make your turtle different colors and move it `forward()` and `backward()`

What about custom colors?

- Colors can be defined in many ways:
 - **RGB** - (Red, Green, Blue)
 - **HSV** - (Hue, Saturation, Value)
 - **Hexadecimal** - “hex” - Base 16 (16 unique characters), 0-9 + A-F



What about custom colors?

- **Hexadecimal**
 - **#16E7CF**
 - 16 characters means easy translation into bytes (2 bytes to describe color)
 - Programmers love that stuff
 - Really easy to grab colors from apps/websites

...but a hash makes text into a comment. How do we pass hex values?

Giving turtles custom colors

- The `color()` function takes **RGB** or **HEX** values
- Takes RGB values range from 0 to 1
- To translate, divide each RGB value by 255

$(255, 0, 0)$ becomes $(1, 0, 0)$

$(127, 127, 127)$ becomes $(0.5, 0.5, 0.5)$

Giving turtles custom colors

Turn Ari blue:

Line

Fill

```
Ari.color((0,0,255/255), (0,0,255/255))
```

```
Ari.color("#0000FF", "0000FF")
```

Your turn!

<https://coolors.co>

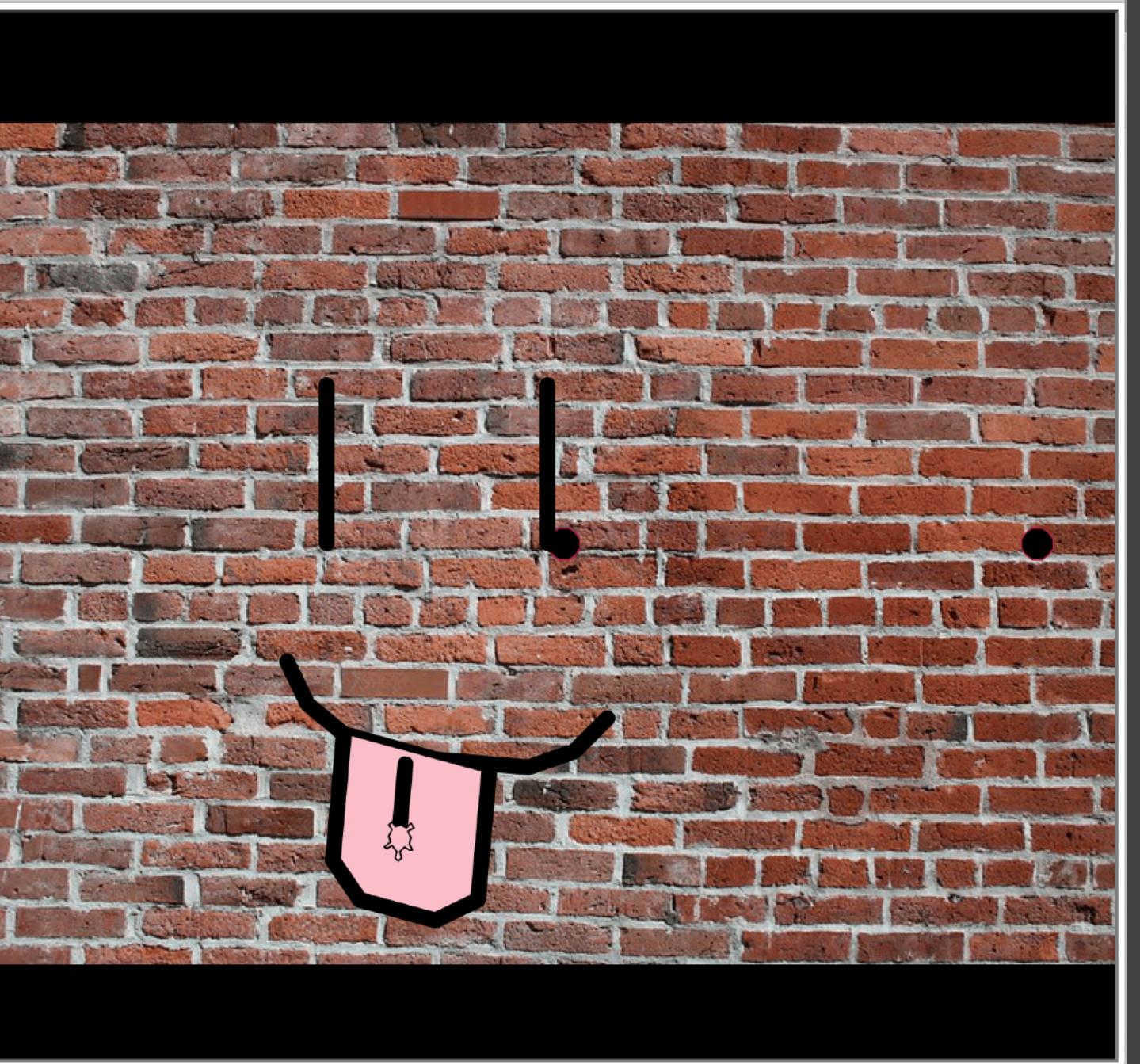
- Use the palette picker to select a color (or two)
- Create a turtle and change its colors!

Class Objectives

- Color theory
 - Naming variables
- Composition
- Getting started programming
 - Planning with pseudocode

PC01 - Graffiti

- You'll be making art on a brick wall background
- Use turtles of different shapes and colors (pick a palette!)
- Use documentations to get an idea of what tools you have



Composition

- The arrangement of visual elements
- Main components:
 - **Color** - hues (RGB, baby!)
 - **Line** - visual path that eyes take when looking at a piece
 - **Shape** - areas defined by edges (or in our case, turtle pen lines)
 - **Space** - where a shape is (positive) and is not (negative)
 - **Texture** - surface qualities that make us think of how things feel
 - **Tone** - how bright or dark is the overall piece?
 - **Depth** - perceived distance from the observer

Composition Hack (know your viewer)

Human eyes are evolved to do certain things. Your art can take advantage of that and then people will like to look at it.

1. **Contrast** - The human eye is drawn to high contrast. Make the “center” of your piece have the highest contrast
 - Human contrast: Black/white, orange/blue, red/green, yellow/purple
2. **Colors** - We loooooooove red (when we can see it)
 - Many things crucial to human survival are red, like fruit and...body parts
 - Red will stick out to us even when it's the same tone as other colors around it



Composition Hack

1. Predictable patterns - We are automatic predictors

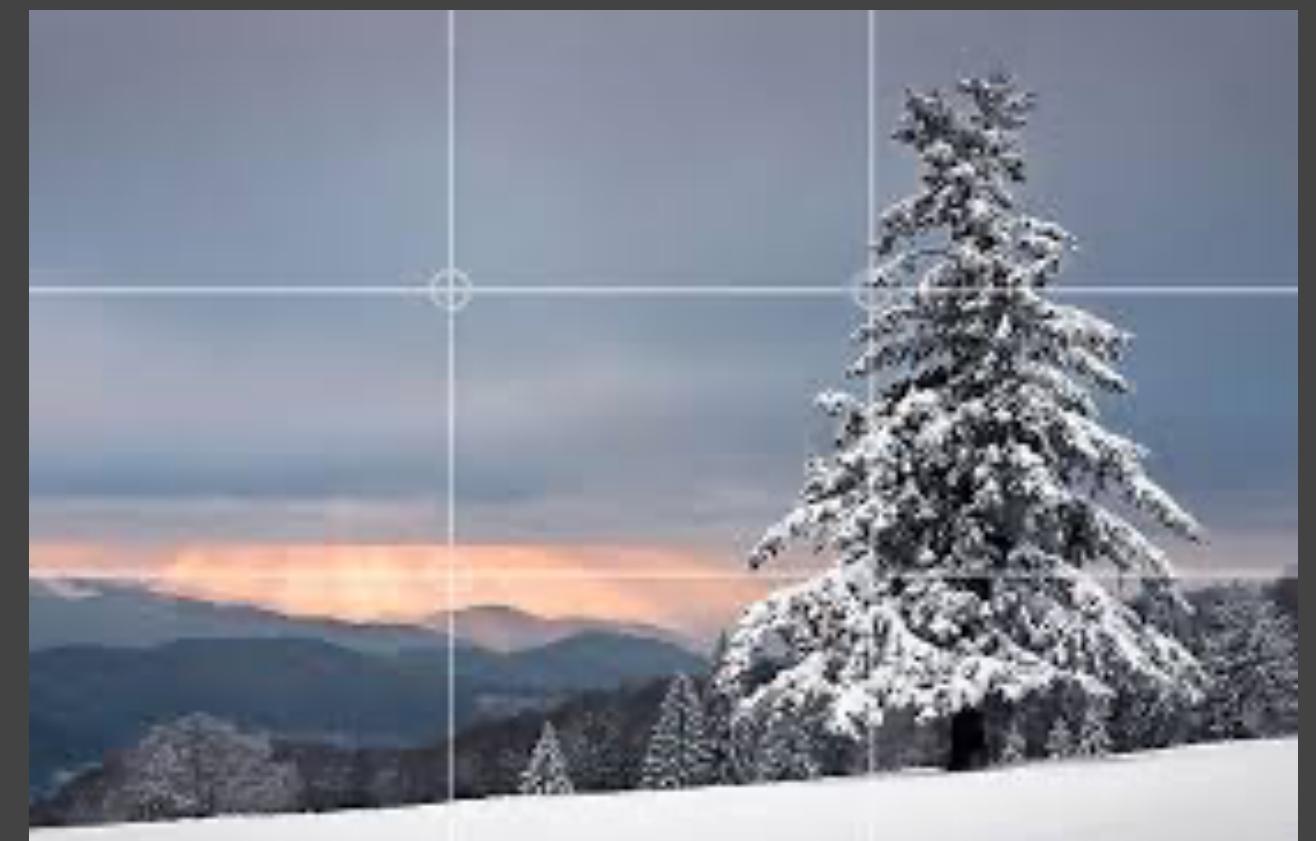
- Regular, smoothly changing patterns are pleasing



@iso.hedron

2. Spatial imbalance drives us wild

- How we perceive depth/distance/volume
- Ties together patterns and contrast
- **Rule of thirds** - place items of interest off-center



What to do with all this theory?

Play!

These aren't rules, just canon. Do what feels right.

What happens if you do the opposite of things humans like?

Can you evoke certain feelings?

Getting started programming

1. Sketch!
 - A. What do you want your image to look like?
 - B. What colors do you want? (Pick your palette)
 - C. Draw it on paper!
2. Write out the order you're going to make stuff (**pseudocode**)
3. Start coding the steps you just figured out

For tomorrow

- Sketch some (SIMPLE) ideas of what to draw as graffiti
- Try using a color picking app to find some neat colors
- Finish reading Ch 2!