

Compute with numbers

- Variables: name that holds a value that has the ability to change
 - ex. Score of a game, amount of money for something, a username or password
- Multiplication = *
- Division = /

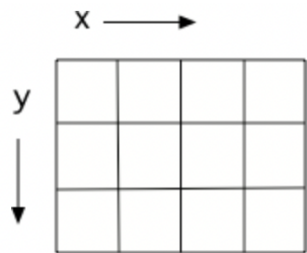
Compute with words

- String; sequences of characters that MUST be in between parentheses
 - ex. basic = "Ma"
- + puts the two words together without a space
 - ex. basic3 = basic + basic + basic
 - print(basic3)
- * repeats strings
 - ex. next = "Mow"
 - next3 = next * 3
 - What was printed: "MowMowMow"
- Dot notation cheat sheet
 - example variable: sentence = "THIS IS A TEST"
 - sentence.lower()
 - This makes the variable (sentence) completely lowercase
 - sentence.capitalize()
 - This makes the first letter of the string uppercase

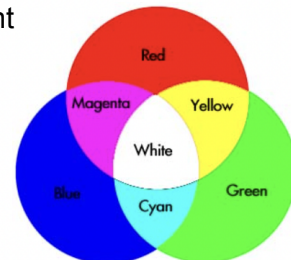
Compute with turtles

- Turtles: basically a pen that the coder can control using certain commands
- Steps to setting up turtle
 1. `from turtle import *` # use the turtle library
 2. `space = Screen()` # create a turtle space
 3. `alex = Turtle()` # create a turtle named alex (or whatever you want)
 4. Move alex by using commands like `alex.forward(enter however far you want your turtle to go)`. You can use other words like backward, left, and right as well.

Compute with images



- Pictures on computers are made of pixels
- They are laid out on a grid
 - from left to right is the x dimension
 - from top to bottom is the y dimension
- All pixels have a certain amount of red, blue, and green to create their exact color.
 - can be in range between 0 to 255
 - pixel colors are made of light not paint so creating colors is a little different



Copy and Pasted code for removing the red with my edits in the comments

from PIL import Image #allows you to work with images from library

Create an image from a file

```
img = Image.open('arch.jpg')
```

Get the photo dimensions

```
width = img.size[0]
```

```
height = img.size[1]
```

this creates a loop that selects the entire photo

pixels = img.load() #a variable you can use to read and write the color of an individual pixel.

```
for col in range(width):
```

```
    for row in range(height):
```

```
        r, g, b = pixels[col, row] # Read pixel color
```

```
        pixels[col, row] = (0, g, b) # the 0 is the red saturation
```

Show the changed image

```
img.show()
```