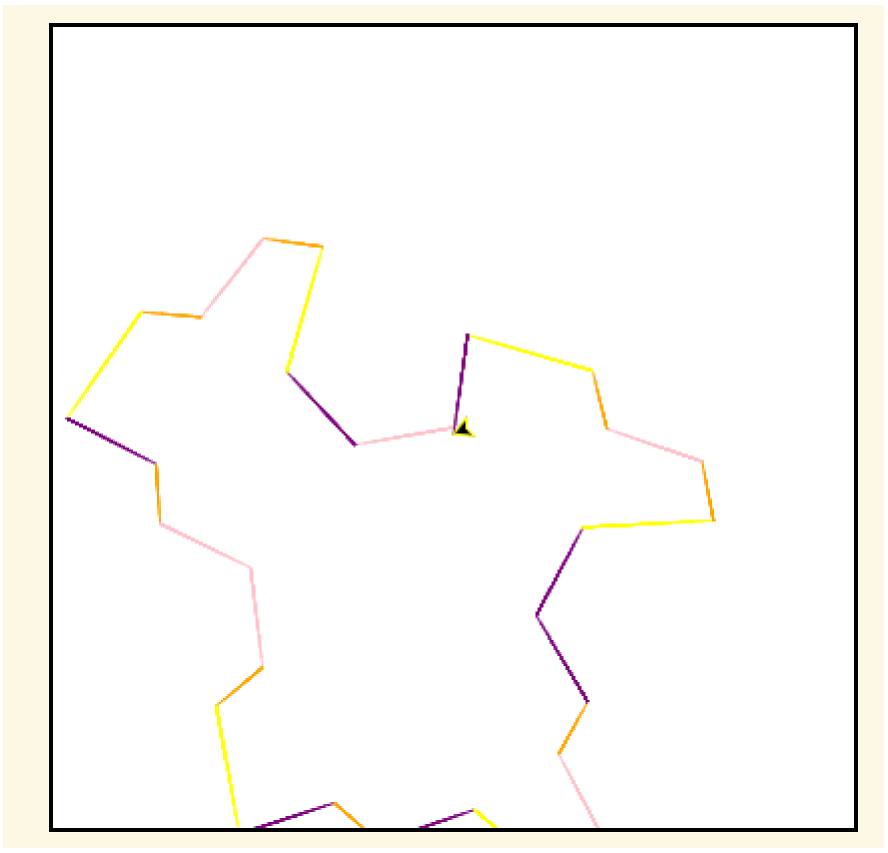
1.2 What we can do with Turtles and Conditionals

- conditional execution
    - if BOOLEAN EXPRESSION: STATEMENTS_1   # executed if condition evaluates to True else: STATEMENTS_2
    - if 和else语句后面都要跟冒号！！！ else里面要是还有if的话就再缩进在写if（nested conditionals）
        - if x < y:
        -     print("x is less than y")
        - else:
        -      if x > y:
        -             print("x is greater than y")
        -      else:
        -             print("x and y must be equal")
        - 这个可以改写一下变得更简洁
        - if x < y:
        -     print("x is less than y")
        - elif   x > y:
        -      print("x is greater than y")这之间可以加无数个elif 但是运行条件是前面的if和elif都不对的情况下
        - else:
        -       print("x and y must be equal")
    - 一个例子去count除了空格之外的字母
        - phrase = "What a wonderful day to program"
        - tot = 0
        - for char in phrase:
        -     if char != " ":
        -     tot = tot + 1
        - print(tot)
    - count元音的例子
        - s = "what if we went to the zoo"
        - x = 0
        - for i in s:
        -    if i in ['a', 'e', 'i', 'o', 'u']:
        -     x += 1

    - print(x)
- 算最大值的例子
    - nums = [9, 3, 8, 11, 5, 29, 2]
    - best_num = 0 从我设定的0开始看
    - for n in nums:
    -     if n > best_num:
    -         best_num = n
    - print(best_num)
    - 如果上面一条改成best_num = nums[0] 那么就是 从num的第一个元素开始看
- 想了很久加过去式的例子
    - words = ["adopt", "bake", "beam", "confide", "grill", "plant", "time", "wave", "wish"]
    - past_tense=[]
    - for word in words:
    -     if word[-1]=="e":
    -         past_tense +=[word+"d"]
    -     else:
    -         past_tense +=[word+"ed"]
- string中有数字需要先分开再算的例子 字符串都是不能直接做事情的 得先弄成list
    - rainfall_mi = "1.65, 1.46, 2.05, 3.03, 3.35, 3.46, 2.83, 3.23, 3.5, 2.52, 2.8, 1.85"
    - num=rainfall_mi.split(",")
    - num_rainy_months=0
    - for n in num:
    -     if float(n) > 3.0: 这里还有个注意的点就是 就算split之后也是各个string而不是float 所以要转换过才能做
    -     num_rainy_months +=1
- 有点集合的例子
    - sentence = "python is a high level general purpose programming language that can be applied to many different classes of problems."需要算有a或者e的单词个数
    - word=sentence.split(" ")分开单词
    - num_a_or_e=0 定义variable
    - for n in word:
    -     if "e" in n:
    -         num_a_or_e+=1 如果没有e的话在其他单词里看是否有a
    -     elif "a" in n:
    -         num_a_or_e +=1

    - 这样就不用去想集合先加再减了 因为会有单词又有e又有a
- 一个很炫酷的例子
    - import turtle
    - wn = turtle.Screen()
    - amy = turtle.Turtle()
    - amy.pencolor("Pink")
    - amy.right(170)
    - colors = ["Purple", "Yellow", "Orange", "Pink", "Orange", "Yellow", "Purple", "Orange", "Pink", "Pink", "Orange", "Yellow", "Purple", "Orange", "Purple", "Yellow", "Orange", "Pink", "Orange", "Purple", "Purple", "Yellow", "Orange", "Pink", "Orange", "Yellow", "Purple", "Yellow"]
    - for color in colors:
    -   if amy.pencolor() == "Purple":
    -       amy.forward(50)
    -       amy.right(59)
    -   elif amy.pencolor() == "Yellow":
    -       amy.forward(65)
    -       amy.left(98)
    -   elif amy.pencolor() == "Orange":
    -       amy.forward(30)
    -       amy.left(60)
    -   elif amy.pencolor() == "Pink":
    -       amy.forward(50)
    -       amy.right(57)
    -       amy.pencolor(color)
    -



-
- Boolean Expressions 布尔数学逻辑体系的 以英国数学家Boole命名
  - literal： store for truth value----“True" "False" 其他任何形式的这个words都不是boolean
  - comparasion operator
    - 比较左右两边的东西
    - ==是相等 !=是不相等 <=小于等于
    - ==和=要区别 =是给...赋值 ==是compare 相等与否
    - 不能写1==5 or 6 or 7这样去比较 需要每个分开来写 但是可以写1==[5,6,7]
  - Logical operator
    - and or notB（取反）
  - in 和 not in
    - 用来检查某个...里是否有...
    - print('p' in 'apple') 输出True
    - print('' in 'apple') True
    - print('apple' in 'apple') True
    - print('x' not in 'apple') True
    - print("a" in ["apple", "absolutely", "application", "nope"]) False 因为在找一个”a“的string 而不是找字母里是否有a
  - precedence of operators
    - notB and or 在所有运算之后 括号幂乘除加减

Level	Category	Operators
7(high)	exponent	**
6	multiplication	*,/,//,%
5	addition	+,-
4	relational	==,!=,<=,>=,>,<
3	logical	not
2	logical	and
1(low)	logical	or

- 一个步骤
    - 5 * 3 > 10 and 4 + 6 == 11
    - 5 * 3 > 10 and 4 + 6 == 11
    - 15 > 10 and 4 + 6 == 11
    - True and 4 + 6 == 11
    - True and 10 == 11
    - True and False
    - False