1. 1 Python Basis

- 入门
  - statement
    - 赋值语句 x=1+2　1+2是expression 整体语句是statement
  - Expression
    - literal expression： expression和value一模一样
  - value
  - Type
    - integer 100
    - float 3.14
    - string: a sequence of characteristics “hello, Coursera!”这样的
  - print statement: print()才会打印出来 如果只是一个数字就打不出来
    - print(30+42)
    - print(10%3) modulo operator----1 保留余数 模运算 可以检查一个数字是否能被另一个数字整除
    - // 剪掉余数 print(1.0//3.0)会输出2.0 输出的同样是float但是已经剪掉余数 print(18.0 // 4)答案是4.0
    - print(4**2) 4的2次幂
    - / 除出来的是float 就算是2输出也会是2.0
    - 优先等级是 括号幂乘除加减
    - print(2 ** 3 ** 2)　# the right-most ** operator gets done first! print((2 ** 3) ** 2) # use parentheses to force the order you want!
    - print((square(3)+2)　print(sub(square(3),square(1+1)))
    - print(type(3)) 输出 ‘int'
    - 单引号双引号三引号都是string 运用规则跟中文里 最外面到最里面不一样是相同的
    - print("""This message will span several lines of the text.""") 三引号可以跨越多行显示
  - int() float() str()可以转换字符的type
    - int(3.999)---3 不是四舍五入而是直接取整数
    - int(-3.999)----变成-3
    - print(int("23bottles"))会错误 因为 里面含有string 识别不出来
    - print("2345", int("2345")) # parse a string to produce an int 输出是2345 2345

        - 有时候想要把一些string和数字结合起来但是python识别不到两种type e.g. val=50+5 print("the value is" +str(50+5))
    - variable
        - 只能字母开始 只能包含数字和letter和下划线 可大写小写 python区分大小写 有些像lambda and 这种是本身带有含义的不能拿来作为定义variable
        - python中可以定义 myname=5 myname="Cynthia" 这是两个不同的变量虽然有相同的名字 所以定义variable的时候没有固定的形式 不是说myname只能是str 这个跟C java中不一样
        - a=15 print(a+4) 会输出19
        - a=5 b=a print(a,b) 会出来 5 5 但是如果多写一句a=3 print出来的结果其实是3 5 这个数学上不一样 因为我只改了a的赋值 b默认还是保持5的原样而不是跟着a变了
        - python中用=赋值 用==相等 跟R中不一样 R中用<-赋值
        - x=6 print(x)现在输出x是3 继续x += 3意思是 # increment x by 3; same as x = x + 3 把x加上3个单位 print(x) 就会得到9 再来 x -= 1 看作x=x-1 # decrement x by 1 输出就是8 把x减去1个单位
    - input Function
        - input永远是把输入的东西当作string 不管输入的是数字还是啥
        - n = input("Please enter your name: ")
        - 一个算时间的program
            - str_seconds = input("Please enter the number of seconds you wish to convert") 给一个input语句赋值成为str_seconds
            - total_secs = int(str_seconds) 把这个语句转化成整数type
            - hours = total_secs // 3600 整除输入的秒数取整数记作小时
            - secs_still_remaining = total_secs % 3600 取输入秒数除以3600的模算出小时整除时舍弃了多少秒记作...
            - minutes = secs_still_remaining // 60 整除这些秒数得到分钟
            - secs_finally_remaining = secs_still_remaining % 60 再取模把弄成分钟还剩下的秒数算出来
            - print("Hrs=", hours, "mins=", minutes, "secs=", secs_finally_remaining) 输出小时分钟秒
- Turtle 画画
    - define class
    - method
        - 特殊种类的方程 比如说alex.forward()
    - instance
        - 这里就是指 alex Turtle in a class rather than a function, 有点像是 因为大Turtle是defined in 小turtle的module里的 所以要把他们两个放在一起写成 turtle.Turtle()然后再

命名 大Turtle是小turtle module里的乌龟

- attribute / instance variables
  - 属性 可以为属性赋值 有时候又叫properties 就是可以做的事情alex.price = 500 tess.price = 600 print(alex.price + tess.price)
  -

turtle-11-6: What is the name of jane's attribute (not method) that is referred to in the following code?

```
import turtle

jane = turtle.Turtle()
jane.forward(20)
print(jane.x)
```

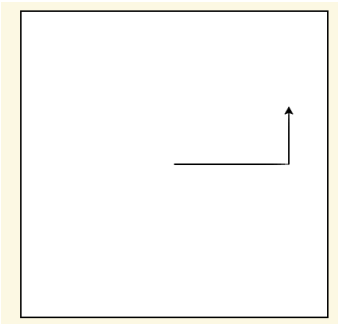
The attribute is

x

Check me | Compare me

Good work!

  - 比如说 each turtle has a color attribute 就可以写成alex.color("red") 让他画出来的曲线变成红色
  - The color of the turtle, the width of its pen(tail), the position of the turtle within the window, which way it is facing, and so on are all part of its current state. Similarly, the window object has a background color which is part of its state.状态
- 一个走了的程序
  - 乌龟 the turtle starts out facing east.
  - 加载一个名叫turtle的总模块 总模块底下有其中两个type 一个是Turtle 一个是Screen
  - import turtle # allows us to use the turtles library 导入了一个turtle module而不是创造了一只龟的意思
  - wn = turtle.Screen() # creates a graphics window turtle和Screen中间的点是有意义的 Screen也可以叫window或者canvas 就是加一块屏幕窗口画布的意思 方便之后作画 取名叫wn
  - 给乌龟爬的地方做了个画布 画布就是Screen type
  - 大小写是区分开来的 module模块是小写 type Turtle是大写 这里就是创造了一只乌龟是Turtle type的 然后alex是Turtle type的一只龟 叫alex（variable）
  - alex = turtle.Turtle() # create a turtle named alex turtle.Turtle意思就是 the Turtle type that is defined within the turtle module
  - alex.forward(150) # tell alex to move forward by 150 units 向前走150单位
  - alex.left(90) # turn by 90 degrees向左转90度
  - alex.forward(75) # complete the second side of a rectangle 向前走90单位

- wn.bgcolor("lightgreen") # set the window background color 画布的颜色变成了亮绿色bg是background的缩写
- 
- alex.color("blue") 变成了蓝乌龟
- alex.pensize(3) # set the width of her pen尾巴画出的宽度是3
- wn.exitonclick() # wait for a user click on the canvas 如果在画布上点一下就会发现 画布自动消失了 关闭turtle窗口并退出(停止执行)Python程序。

- 可以有多只乌龟 a herd of Turtles
  - 有一个注意的点是 比如说让乌龟画了一个三角形 到最后一笔的时候乌龟头的方向不是向起始一样往东走了 但是我们需要把他归位 哪怕接下来乌龟不爬了也是这样 这样就不用换算单位 方便写码

- For loop
  - 有个小程序
    - for _ in range(3):
    - print("This line will execute three times")
    - print("This line will also execute three times")
    - 结果是会2条语句121212重复3次而不是111222
    - for底下的缩进很重要！没有缩进就在for之外了
    - for句子最后需要有冒号不要忘了！
  - 继续画高等乌龟
    - 乌龟转的角度可以是正的也可以是负数，但是方向会不一样 比如向左-30就是直接向右了 但是向左330要转一大圈再变到那个位置 同样前进和后退也可以负数
    - alex.backward(-100)就是往前走100
    - 乌龟的尾巴也可以不拖着 用alex.up() 就行
    - Every turtle can have its own shape. The ones available “out of the box” are arrow, blank, circle, classic, square, triangle, turtle.
    - 速度也可以设置alex.speed(10)
    - tess.stamp() 印下自己的脚印
    -

Method	Parameters	Description
Turtle	None	Creates and returns a new turtle object
forward	distance	Moves the turtle forward
backward	distance	Moves the turle backward
right	angle	Turns the turtle clockwise
left	angle	Turns the turtle counter clockwise
up	None	Picks up the turtles tail
down	None	Puts down the turtles tail
color	color name	Changes the color of the turtle's tail
fillcolor	color name	Changes the color of the turtle will use to fill a polygon
heading	None	Returns the current heading
position	None	Returns the current position
goto	x,y	Move the turtle to position x,y
begin_fill	None	Remember the starting point for a filled polygon
end_fill	None	Close the polygon and fill with the current fill color
dot	None	Leave a dot at the current position
stamp	None	Leaves an impression of a turtle shape at the current location
shape	shapename	Should be 'arrow', 'triangle', 'classic', 'turtle', 'circle', or 'square'
speed	integer	0 = no animation, fastest; 1 = slowest; 10 = very fast

- module：含有很多python的class function...的集合
  - import module就是把这些集合都导入进你的python程序中 所以之后在用module里面的东西的时候就要写 e.g. alex = turtle.Turtle() 来自turtle module的Turtle Instance 然后命名为alex
  - 或者写from A import B 下面就可以直接用B而不是A.B了
  - random module
    - 小项目
      - import random
      - prob = random.random()
      - print(prob)
      - diceThrow = random.randrange(1,7) # return an int, one of 1,2,3,4,5,6 这里注意了是有1无7 如果只有(7)那假定第一个数字是0 从0-6
      - print(diceThrow)
- Degugging
  - Syntax errors 语法错误

    - statement not well formed
    - print("Hello World!"会说SyntaxError: EOF in multi-line statement on line 3 因为并没有把括号扩上 所以程序一直不停运行
    - 有时候告诉你line 4 错误可能是因为前面的错误导致的
    - 有个tips就是你可以#掉显示错误的那一行 如果继续往下运行还是错的 那可能就要往上看看了
- Runtime errors
    - print(3/0) 这种 illegal process
    - 在运行程序之前错误不会发现
- Semantic errors 语义错误
    - can't produce 比如忘记了/100变成百分数这种
    - 你想写的程序和做出来的东西有差别
-