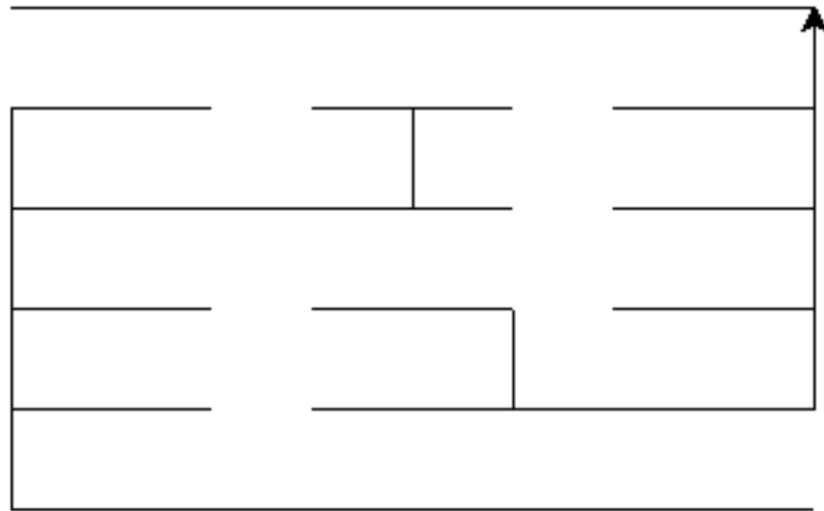


迷宫大作战

从下往上，从左往右，先画横再画竖



turtle模块---会画画的海龟

创建海龟对象

```
import turtle
t1 = turtle.Turtle() # 第一个海龟对象
t2 = turtle.Turtle() # 第二个海龟对象
```

一个海龟对象可看作一个画笔，

运行程序会出现画布

一闪而过怎么办

类，属性，方法，eg前进，后退，设置画笔颜色，粗细

成员访问运算符“.”以对象成员的方式调用里面的方法， eg .upper()

画图解释

```
import turtle

s = 'hello'
print(s.upper())

t1 = turtle.Turtle() # 第一个海龟对象
n = 5
t1.forward(100)

input('回车结束程序')
```

设置海龟造型

```
t1.shape("turtle")
circle
square
triangle
```

隐藏画笔

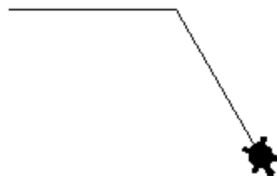
```
t1.hideturtle()
```

移动&转向

```
t1.forward(100)
t1.right(90)
t1.forward(100)
t1.left(90)
t1.backward(100)
```

练习

绘制一个120°的角



本节的第二次课

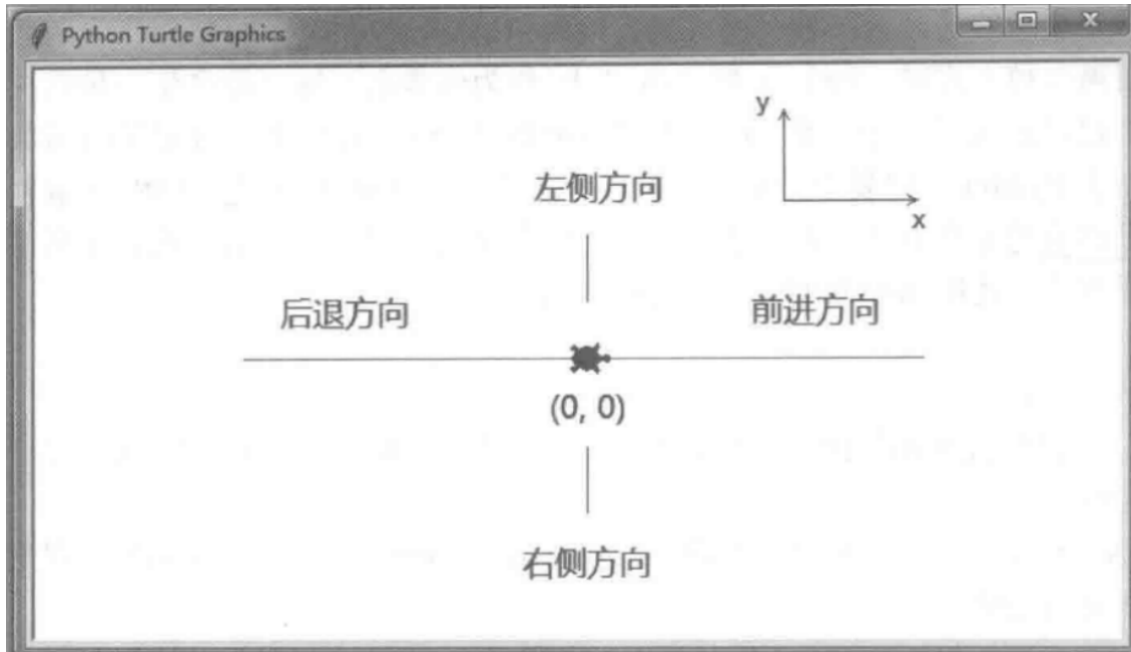
画一个圆（尽可能的圆）

想一想，我们可以画出开头的那个迷宫了嘛？如果不能，请提出疑问

疑问1：迷宫各边是多长，迷宫在画布的哪个位置，都不清楚

疑问2：迷宫不能一笔画出

给图做一个详细的标注，把他放在一个坐标系下面，这样我们就知道每个直线是多长



当海龟被创建时，其初始位置在画布中央 (0, 0) 初始方向水平向右，即x轴正方向。

```
t1.forward(100) # 100是像素长
```

设置画笔的位置和方向

```
t1.goto(100,100)
t1.setheading(30) # 设置画笔方向与x轴正方向的夹角
```

抬笔落笔

```
t1.penup()
t1.pendown()
```

练习

画两条长度为100像素相距50像素的平行线

```
t1 = turtle.Turtle()
t2 = turtle.Turtle()
t1.forward(100)
t2.penup()
t2.goto(0, 50)
t2.pendown()
t2.forward(100)
```

绘制迷宫

如何保证迷宫中的线断不重不漏

从下往上，从左往右，先画横再画竖

函数，让程序变得精简

我们以前接触过函数吗？

```
print()
int()
len()
.upper()
```

函数的特征： 函数名+ ()

我们之前就是调用别人写好的函数，这叫**函数调用**

将完成某个功能的函数写在一起，然后封装成函数，用的时候直接调用。简化我们的程序。

今天我们要学习**函数定义**

```
def 函数名():
    语句1
    语句2
    ....
```

def是函数定义的关键字，

: 表示即将开始函数内部的语句

封装在函数内部的代码有相同的缩进，表示它们都隶属于这个函数

来个实操!!

```
def noodles():
    print('开始制作牛肉面', end=',')
```

```

print('烧水', end=',')
print('煮面', end=',')
print('配味', end=',')
print('加汤', end=',')
print('加辣椒', end=',')
print('加牛肉', end=',')
print('牛肉面做好了')

n = int(input("客官要几碗牛肉面"))
while n >= 1:
    noodles()
    print('第{}碗'.format(n))
    n = n - 1

```

函数定义要在函数调用的前面，要在调用之前定义！就像让厨师做面之前，要教会厨师怎么做

函数如何运行，跑一遍。

在函数定义时设置参数-----增加菜品的规格选项

在函数调用中传递参数-----提出菜品的具体要求

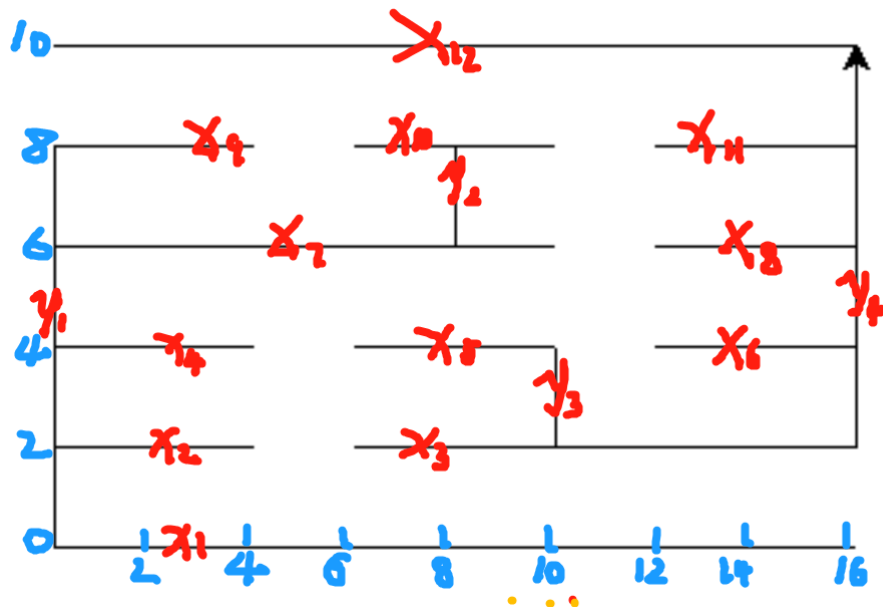
```

def noodles(lajiao,xiangcai):
    print('开始制作牛肉面', end=',')
    print('烧水', end=',')
    print('煮面', end=',')
    print('配味', end=',')
    print('加汤', end=',')
    if lajiao:
        print('加辣椒', end=',')
    print('加牛肉', end=',')
    if xiangcai:
        print('加香菜', end=',')
    print('牛肉面做好了')

n = int(input("客官要几碗牛肉面"))
while n >= 1:
    noodles(lajiao=True, xiangcai=False)
    print('第{}碗'.format(n))
    n = n - 1

```

利用函数画迷宫



```
import turtle
```

```
def drawLine(t,x,y,l): #传参: 画笔, 起点, 走多远
    size = 20
    t.penup()
    t.goto(x*size,y*size)
    t.pendown()
    t.forward(l*size)
```

```
t = turtle.Turtle()
```

```
#画横线
```

```
drawLine(t,0,0,16)#x1
drawLine(t,0,2,4)#x2
drawLine(t,6,2,10)#x3
drawLine(t,0,4,4)#x4
drawLine(t,6,4,4)#x5
drawLine(t,12,4,4)#x6
drawLine(t,0,6,10)#x7
drawLine(t,12,6,4)#x8
drawLine(t,0,8,4)#x9
drawLine(t,6,8,4)#x10
drawLine(t,12,8,4)#x11
drawLine(t,0,10,16)#x12
```

```
#画笔左转90°为垂直向上方向
```

```
t.left(90)
```

```
#画纵线
```

```
drawLine(t,0,0,8)#y1
drawLine(t,8,6,2)#y2
drawLine(t,10,2,2)#y3
```

```
drawLine(t,16,2,8)#y4
```

```
input('回车结束程序')
```