

List

Python 基础入门

回顾

Turtle 补充

控制 位置

控制 方向

添加 文字

for 乌龟赛跑



乌龟赛跑

到达终点后 停止 并添加 文字

```
import turtle
import random
import time

t = turtle.Turtle()
t.speed(10)
t.penup()
t.goto(-200, 150)
for i in range(21):
    t.write(i)
    t.right(90)
    t.pendown()
    t.forward(300)
    t.penup()
    t.backward(300)
    t.left(90)
    t.forward(20)
```

```
alice = turtle.Turtle()
alice.shape('turtle')
alice.color('red')
alice.penup()
alice.goto(-220, 100)
ben = turtle.Turtle()
ben.shape('turtle')
ben.color('blue')
ben.penup()
ben.goto(-220, 0)
claire = turtle.Turtle()
claire.shape('turtle')
claire.color('green')
claire.penup()
claire.goto(-220, -100)

max_speed_a = 10
max_speed_b = 10
max_speed_c = 10
terminal = 200
order = 0
```

```
while True:
    alice.forward(random.randrange(max_speed_a))
    ben.forward(random.randrange(max_speed_b))
    claire.forward(random.randrange(max_speed_c))
    time.sleep(0.1)
    if alice.xcor() >= terminal and max_speed_a > 1:
        max_speed_a = 1
        t.goto(240, 90)
        if order == 0:
            t.write('冠军', font=('Arial', 20, 'normal'))
        elif order == 1:
            t.write('亚军', font=('Arial', 20, 'normal'))
        elif order == 2:
            t.write('季军', font=('Arial', 20, 'normal'))
        order = order + 1
    if ben.xcor() >= terminal and max_speed_b > 1:
        max_speed_b = 1
        t.goto(240, -10)
        if order == 0:
            t.write('冠军', font=('Arial', 20, 'normal'))
        elif order == 1:
            t.write('亚军', font=('Arial', 20, 'normal'))
        elif order == 2:
            t.write('季军', font=('Arial', 20, 'normal'))
        order = order + 1
    if claire.xcor() >= terminal and max_speed_c > 1:
        max_speed_c = 1
        t.goto(240, -110)
        if order == 0:
            t.write('冠军', font=('Arial', 20, 'normal'))
        elif order == 1:
            t.write('亚军', font=('Arial', 20, 'normal'))
        elif order == 2:
            t.write('季军', font=('Arial', 20, 'normal'))
        order = order + 1
    if order == 3:
        break
```

列表

列表就像一条 队伍

- ✓ 一个简单的例子: `[5, 4, 3, 2, 1]`
- ✓ 种类 多样:
 - ✓ 队伍中可以有男生、女生、老人、小孩
 - ✓ 列表中可以有整数、浮点数、字符串
- ✓ 数量 可变:
 - ✓ 队伍中的人数不断变化
 - ✓ 列表中的元素数量可变



■ 列表

我们已经 使用过 列表

列表中有 多少个 元素

这些元素分别是什么 类型

```
for i in [5, 4, 3, 2, 1]:  
    print(i)  
print("Blastoff!")
```

```
5  
4  
3  
2  
1  
Blastoff!
```

列表

列表的好处

- ✓ 用 一个 变量就可以表示 很多个 元素
 - ✓ 列表属于 集合
 - ✓ 整数、浮点数 不属于 集合
- ✓ 列表中的元素可以 动态调整：增加、删除

```
>>> x = 2
>>> print(x)
2
>>> x = 4
>>> print(x)
4
>>>
```

```
friends = [ 'Joseph', 'Glenn', 'Sally' ]
carryon = [ 'socks', 'shirt', 'perfume' ]
```

列表

如何 定义 列表

- ✓ 用 中括号 包裹一些元素
- ✓ 元素之间用 逗号 分隔
- ✓ 列表中的元素可以是 任意类型
- ✓ 甚至可以是 另一个列表
- ✓ 列表可以是 空的

```
>>> odd = [1, 3, 5]
>>> print(odd)
[1, 3, 5]
>>> color = ['red', 'green', 'blue']
>>> print(color)
['red', 'green', 'blue']
>>> mix = ['red', 3, 15.6]
>>> print(mix)
['red', 3, 15.6]
>>> nested = [1, mix, 'hello']
>>> print(nested)
[1, ['red', 3, 15.6], 'hello']
>>> empty = []
>>> print(empty)
[]
```

■ 列表

列表 和 for 是一对好朋友

使用 for 可以非常方便地 遍历 列表中的 每一个元素

```
friends = ['Joseph', 'Glenn', 'Sally']  
for friend in friends :  
    print ("Happy New Year:", friend)  
print ("Done!")
```

```
Happy New Year: Joseph  
Happy New Year: Glenn  
Happy New Year: Sally  
Done!
```


索引

使用 索引 操作列表中的元素

- ✓ 好比队伍中每个人的 编号
- ✓ 索引也称作 下标, 从 0 开始
- ✓ 通过索引可以 获取 列表中的元素
- ✓ 通过索引可以 修改 列表中的元素

Joseph	Glenn	Sally
0	1	2

```
>>> friends = [ 'Joseph', 'Glenn', 'Sally' ]  
>>> print(friends[1])  
Glenn
```

```
>>> friends[1] = 'Ben'  
>>> print(friends)  
['Joseph', 'Ben', 'Sally']
```

长度

使用 `len()` 获取列表的长度

- ✓ 得到一个 整数，即列表中元素的个数
- ✓ `len()`也可以用来获取 字典 和 字符串 的长度

```
>>> ls = [1, 2, 'joe', 2.5]
>>> print(len(ls))
4
```

■ 列表

搭配使用 `len()` 和 `range()`

- ✓ 遍历列表的 两种 常用方法
- ✓ 第一种 只能读不能改

```
Happy New Year: Joseph  
Happy New Year: Glenn  
Happy New Year: Sally
```

```
friends = ['Joseph', 'Glenn', 'Sally']  
  
for friend in friends:  
    print('Happy New Year:', friend)
```

```
for i in range(len(friends)):  
    friend = friends[i]  
    print('Happy New Year:', friend)
```

练习

如何 修改 列表

- ✓ 定义一个列表: [5, 4, 3, 2, 1]
- ✓ 使用 `for` 和 `range()`, 将每一个数字变成原来的 两倍

```
numbers = [5, 4, 3, 2, 1]

for i in numbers:
    i = i * 2

print(i)
print(numbers)
```



列表

一些 有用 的方法

- ✓ 全部为 数字：整数、浮点数
- ✓ 全部为 字符串 时，哪些方法可用

```
>>> nums = [3, 41, 12, 9, 74, 15]
>>> len(nums)
6
>>> max(nums)
74
>>> min(nums)
3
>>> sum(nums)
154
>>> sum(nums) / len(nums)
25.666666666666668
```

■ 列表

列表的 加法 和 乘法

- ✓ 列表 + 列表: 连接
- ✓ 列表 * 整数: 重复

```
numbers = [1, 1.5, 2]
words = ['You', 'He', 'She']
print(numbers + words)

print(numbers * 3)
print(3 * words)
```

列表

列表的 切片 (slice)

- ✓ 使用 冒号 进行分片
- ✓ 使用 两个 索引
 - ✓ 包括前面的索引
 - ✓ 但不包括后面的索引
- ✓ 只使用 一个 索引
- ✓ 使用 负索引

```
>>> t = [9, 41, 12, 3, 74, 15]  
>>> t[1:3]
```

```
[41, 12]
```

```
>>> t[:4]
```

```
[9, 41, 12, 3]
```

```
>>> t[3:]
```

```
[3, 74, 15]
```

```
>>> t[:]
```

```
[9, 41, 12, 3, 74, 15]
```

```
>>> t1 = t[1:-1]
```

```
>>> print(t1)
```

```
[41, 12, 3, 74]
```

```
>>> print(t)
```

```
[9, 41, 12, 3, 74, 15]
```

列表

从零开始添加元素

- ✓ 定义一个 空列表
- ✓ 使用 `append()` 添加元素
- ✓ 使用 `remove()` 删除元素
- ✓ 使用 `sort()` 进行排序

```
>>> numbers = [6, 12, 8, 53, 21, 2]
>>> numbers.sort()
>>> print(numbers)
[2, 6, 8, 12, 21, 53]
```

```
>>> mylist = []
>>> print(mylist)
[]
>>> mylist.append('Hello')
>>> mylist.append(99)
>>> print(mylist)
['Hello', 99]
>>> mylist.append('Nice')
>>> print(mylist)
['Hello', 99, 'Nice']
>>> mylist.remove(99)
>>> print(mylist)
['Hello', 'Nice']
```

```
>>> friends = ['Joseph', 'Glenn', 'Sally']
>>> friends.sort()
>>> print(friends)
['Glenn', 'Joseph', 'Sally']
```


列表

在列表中 查找 元素

- ✓ 使用 `in` 判断元素是否 在 列表中
- ✓ 使用 `not in` 判断元素是否 不在 列表中
- ✓ 以上判断 不会改变 列表中的元素

```
>>> some = [1, 9, 21, 10, 16]
>>> 9 in some
True
>>> 15 in some
False
>>> 15 not in some
True
```

列表

通过 `input()` 向列表中输入元素

不用列表 可以实现同样的功能吗

```
Enter a number: 10
Enter a number: 7
Enter a number: 98
Enter a number: 103
Enter a number: done
Average: 54.5
```

```
numlist = list()
while True :
    inp = input('Enter a number: ')
    if inp == 'done' :
        break
    value = float(inp)
    numlist.append(value)

average = sum(numlist) / len(numlist)
print ('Average:', average)
```

■ 作业

1 不用列表

- ✓ 改写 上一页 的代码
- ✓ 不用 列表, 实现 同样的 功能

■ 作业

2 评分

- ✓ 不断输入 一些分数，以 done 结束
- ✓ 分别去掉一个 最高分 和 最低分
- ✓ 计算剩下分数的 平均值
- ✓ 至少使用 两种 方法实现



Python

去实现你想做的任何事情