

python基础（基本语法）

Lenovo

2021年9月1日





今日内容

- 第一个python程序
- 变量
- 数据类型-int/str/bool/list
- 用户指令-读取用户输入信息
- 运算符
- 格式化打印-输出好看的信息
- 逻辑判断

+ python环境安装

```
while.py x while.py x
# -*- coding:utf-8 -*-
# created by Alex Li - 路飞学城

age = 30
count = 0
while count < 3:
    n = int(input("你多大了: "))
    if n != age and count < 2:
        print("再来一次")
    else:
        print("666")
        exit()

    count += 1
```

your code

python 解释器



机器码

0101001001010010101001

01010011010101010.....

.....

.....

.....010101

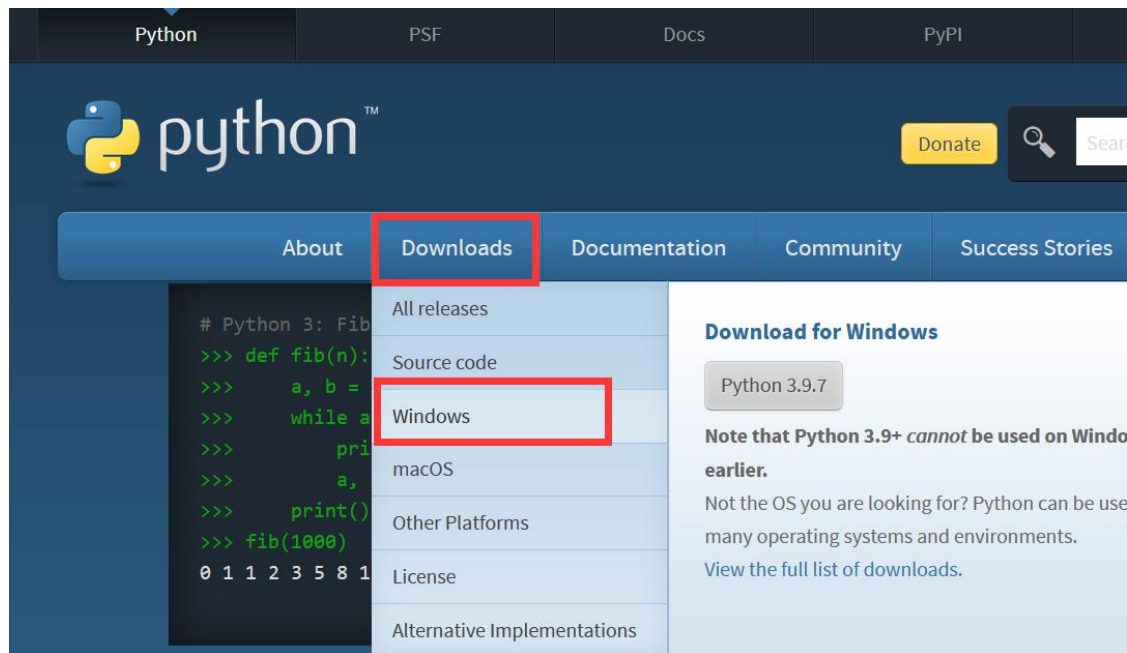
执行结果

```
Run: while x
/Users/alex/Documents/work/PyProjects
你多大了: 26
再来一次
你多大了: 50
再来一次
你多大了: 30
666
Process finished with exit code 0
```

+ python环境安装

官网下载<https://www.python.org/>

选择稳定版



Python Releases for Windows

- [Latest Python 3 Release - Python 3.9.7](#)
- [Latest Python 2 Release - Python 2.7.18](#)

Stable Releases

稳定版

- [Python 3.9.7 - Aug. 30, 2021](#)

Note that Python 3.9.7 cannot be used on Windows 7 or earlier.

- [Download Windows embeddable package \(32-bit\)](#)
- [Download Windows embeddable package \(64-bit\)](#)
- [Download Windows help file](#)
- [Download Windows installer \(32-bit\)](#)

- [Download Windows installer \(64-bit\)](#)

64位

- [Python 3.8.12 - Aug. 30, 2021](#)

Note that Python 3.8.12 cannot be used on Windows XP or earlier.

- No files for this release.

Pre-releases

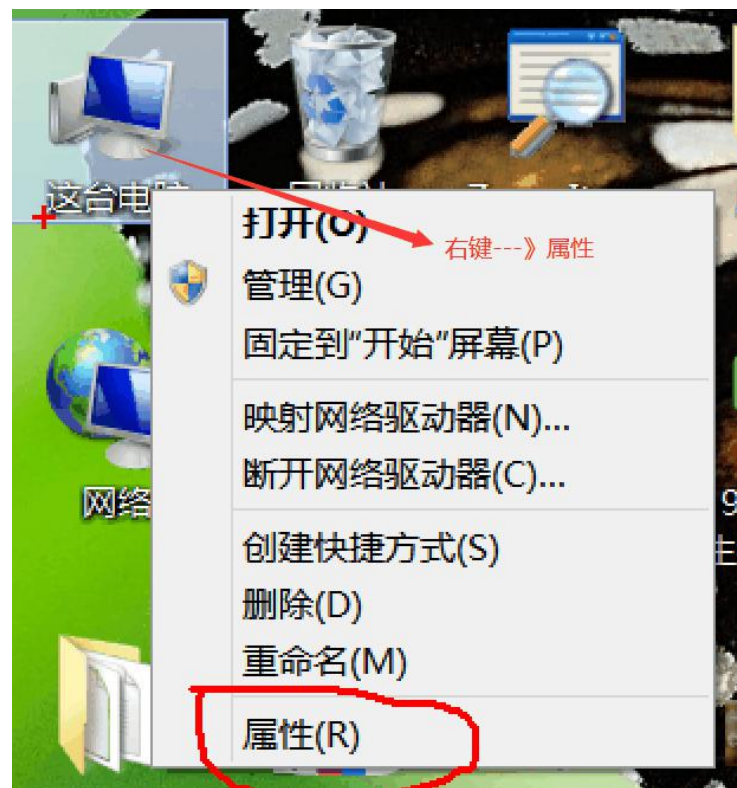
- [Python 3.10.0rc1 - Aug. 2, 2021](#)
 - [Download Windows embeddable package \(32-bit\)](#)
 - [Download Windows embeddable package \(64-bit\)](#)
 - [Download Windows help file](#)
 - [Download Windows installer \(32-bit\)](#)
 - [Download Windows installer \(64-bit\)](#)
- [Python 3.10.0b4 - July 10, 2021](#)
 - [Download Windows embeddable package \(32-bit\)](#)
 - [Download Windows embeddable package \(64-bit\)](#)
 - [Download Windows help file](#)
 - [Download Windows installer \(32-bit\)](#)
 - [Download Windows installer \(64-bit\)](#)

+ 二、python环境安装

加入环境变量



未加入安装后可修改环境变量



+ python环境安装

- windows —> 运行(win+r) —> 输入cmd , 然后回车, 弹出cmd程序, 输入python,如果能进入交互环境 , 代表环境变量成功。



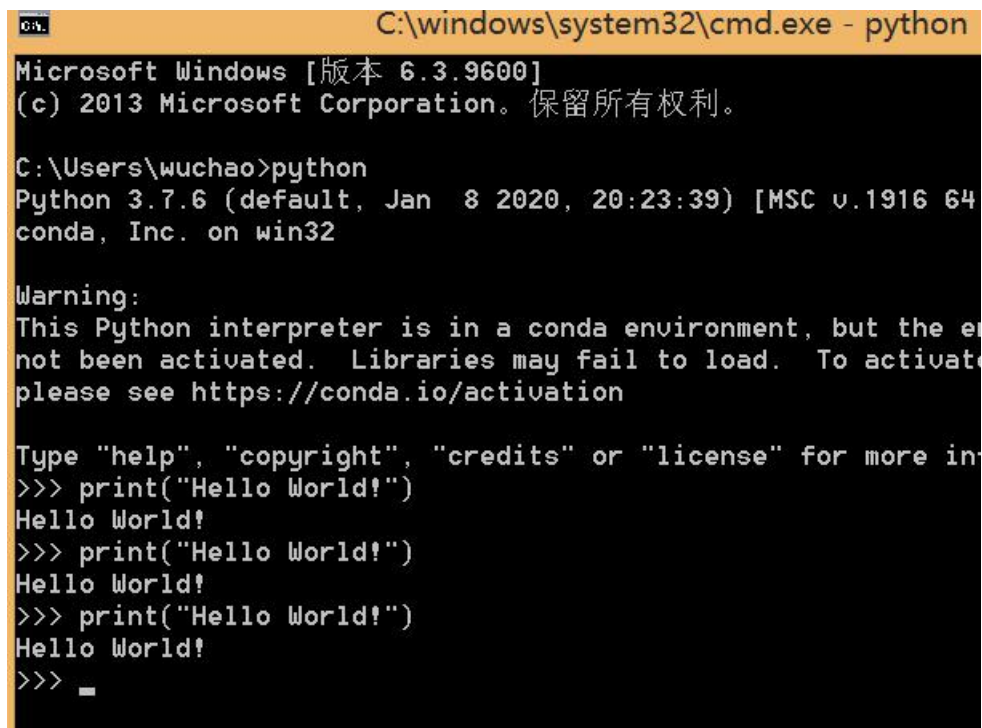
```
C:\Windows\system32\cmd.exe - python

Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

C:\Users\alex>python
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit
Type "help", "copyright", "credits" or "license" for more information.
>>> _
```


+ 写下你的第一个python程序

交互模式:退出交互模式, 代码丢失, 调试使用



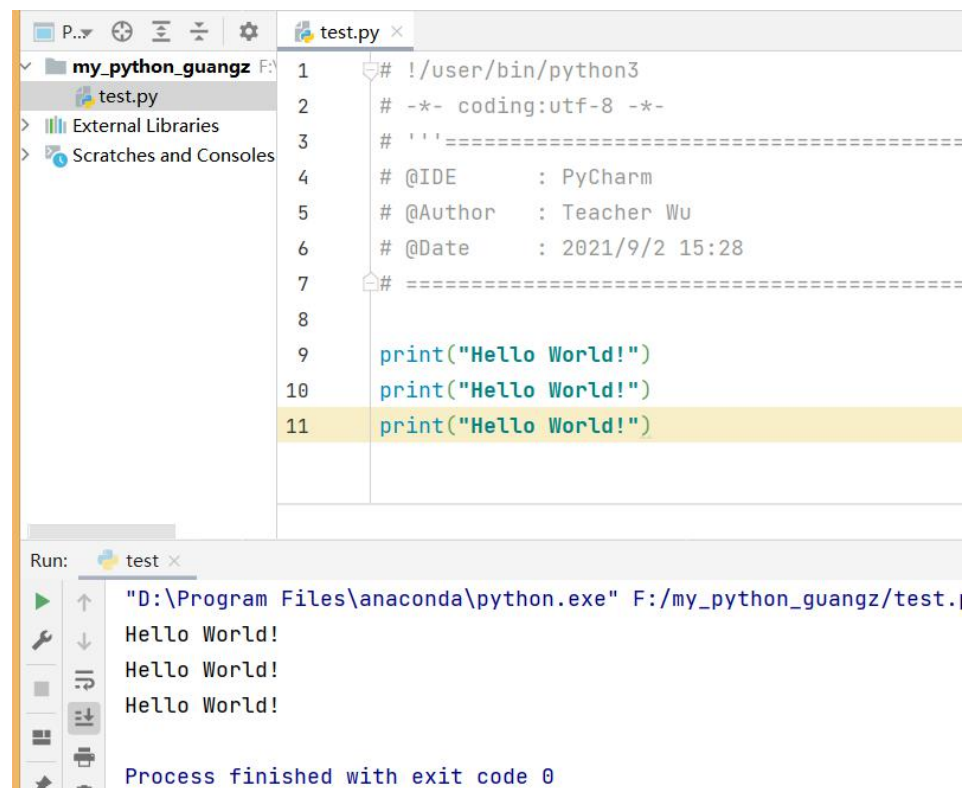
```
C:\windows\system32\cmd.exe - python
Microsoft Windows [版本 6.3.9600]
(c) 2013 Microsoft Corporation. 保留所有权利。

C:\Users\wuchao>python
Python 3.7.6 (default, Jan 8 2020, 20:23:39) [MSC v.1916 64
conda, Inc. on win32

Warning:
This Python interpreter is in a conda environment, but the en
not been activated. Libraries may fail to load. To activat
please see https://conda.io/activation

Type "help", "copyright", "credits" or "license" for more in
>>> print("Hello World!")
Hello World!
>>> print("Hello World!")
Hello World!
>>> print("Hello World!")
Hello World!
>>> _
```

写在文件里:可永久保存, 正式开发都写在文件里



```
test.py
1 #!/user/bin/python3
2 # -*- coding:utf-8 -*-
3 # '''=====
4 # @IDE      : PyCharm
5 # @Author   : Teacher Wu
6 # @Date     : 2021/9/2 15:28
7 # =====
8
9 print("Hello World!")
10 print("Hello World!")
11 print("Hello World!")

Run: test
"D:\Program Files\anaconda\python.exe" F:/my_python_guangz/test.py
Hello World!
Hello World!
Hello World!

Process finished with exit code 0
```


+ cpu 内存 硬盘的关系

- 电脑之父——冯·诺伊曼提出了计算机的五大部件：输入设备、输出设备、存储器、运算器和控制器。



cpu(负责计算)

20000MB/s



内存(负责数据临时存储)

10000~15000MB/s



硬盘(负责数据存储)

200~600MB/s)



+ 变量

- 变量是什么
- 变量的定义和调用
- 变量名、变量值、内存地址
- 变量的定义规则
- 变量的命名规范
- 变量的定义方式
- 变量的修改和删除



+ 变量

- 变量是什么

- 是用于在内存中存放程序数据的容器

- 变量的定义和调用

- 定义：变量名 = 值
 - 调用：变量名

- 变量的使用规则

- 程序是从上到下执行的，所以变量必须先定义，后调用，否则会报错

```
name = "小明"  
age = 22  
height = 160
```



```
name = "小明"  
age = 22  
height = 160  
print(name)  
print(age)
```

+ 变量

- 变量名定义规则

- 1. 变量名只能是 字母、数字或下划线的任意组合
- 2. 变量名的第一个字符不能是数字
- 3. 以下关键字不能声明为变量名['and' , 'as' , 'assert' , 'break' , 'class' , 'continue' , 'def' , 'del' , 'elif' , 'else' , 'except' , 'exec' , 'finally' , 'for' , 'from' , 'global' , 'if' , 'import' , 'in' , 'is' , 'lambda' , 'not' , 'or' , 'pass' , 'print' , 'raise' , 'return' , 'try' , 'while' , 'with' , 'yield']

⊕ 常用定义方式

- 驼峰体

- AgeOfOldboy = 56
- NumberOfStudents = 80

- 下划线

- age_of_oldboy = 56
- number_of_students = 80

- 不好的方式举例

- 1.- 变量名为中文、拼音 2.- 变量名过长 3.- 变量名词不达意

变量

- 变量的修改
 - 相当于重新赋值
- 变量的删除
 - `del` 变量名

注释 python编辑器会忽略的文本

- # 单行注释
- `""" """` 多行注释
- `''' '''` 多行注释

➤ 基本数据类型

● 数据类型是什么

- **数据类型**是编程语言中为了对数据进行描述的定义,因为对于机器不能识别数据,而不同数据间的相互运算,在机器内部的执行方式是不一样的.这就要用户先定义数据的特性再进行其它操作.这里的特性也就是数据类型

➕ 基本的数据类型

- 数字类型:(int \ long \ float)
- 字符串str:存储文字
- 列表list: 购物列表、学员列表、客户列表等。。
- 布尔bool: 处理逻辑运算
- 集合set: 处理2组数据间的关系
- 字典dict: 存储更多信息,
- 字节bytes: 二进制数据、处理图片、视频、数据流等。。

+ 数字类型

- int (整型)
 - 在64位系统上，整数的位数为64位，取值范围为 $-2^{63} \sim 2^{63}-1$ ，即-9223372036854775808 ~ 9223372036854775807
- long (长整型)
 - 跟C语言不同，Python的长整数没有指定位宽，即：Python没有限制长整数数值的大小，但实际上由于机器内存有限，我们使用的长整数数值不可能无限大。
 - **注意**，自从Python2.2起，如果整数发生溢出，Python会自动将整数数据转换为长整数，所以如今在长整数数据后面不加字母L也不会导致严重后果了。
 - **注意**：在Python3里不再有long类型了，全都是int
- float (浮点型) 即小数

数字类型

运算

a=10, b=20

运算符	描述	实例
+	加 - 两个对象相加	a + b 输出结果 30
-	减 - 得到负数或是一个数减去另一个数	a - b 输出结果 -10
*	乘 - 两个数相乘或是返回一个被重复若干次的字符串	a * b 输出结果 200
/	除 - x除以y	b / a 输出结果 2
%	取模 - 返回除法的余数	b % a 输出结果 0
**	幂 - 返回x的y次幂	a**b 为10的20次方， 输出结果 100000000000000000000
//	取整除 - 返回商的整数部分	9//2 输出结果 4 , 9.0//2.0 输出结果 4.0

+ 数字类型

● 比较运算

假设变量：a=10, b=20

运算符	描述	实例
==	等于 - 比较对象是否相等	(a == b) 返回 False。
!=	不等于 - 比较两个对象是否不相等	(a != b) 返回 true。
<>	不等于 - 比较两个对象是否不相等。python3 已废弃。	(a <> b) 返回 true。这个运算符类似 !=。
>	大于 - 返回x是否大于y	(a > b) 返回 False。
<	小于 - 返回x是否小于y。所有比较运算符返回1表示真，返回0表示假。这分别与特殊的变量True和False等价。	(a < b) 返回 true。
>=	大于等于 - 返回x是否大于等于y。	(a >= b) 返回 False。
<=	小于等于 - 返回x是否小于等于y。	(a <= b) 返回 true。

+ 字符串str类型

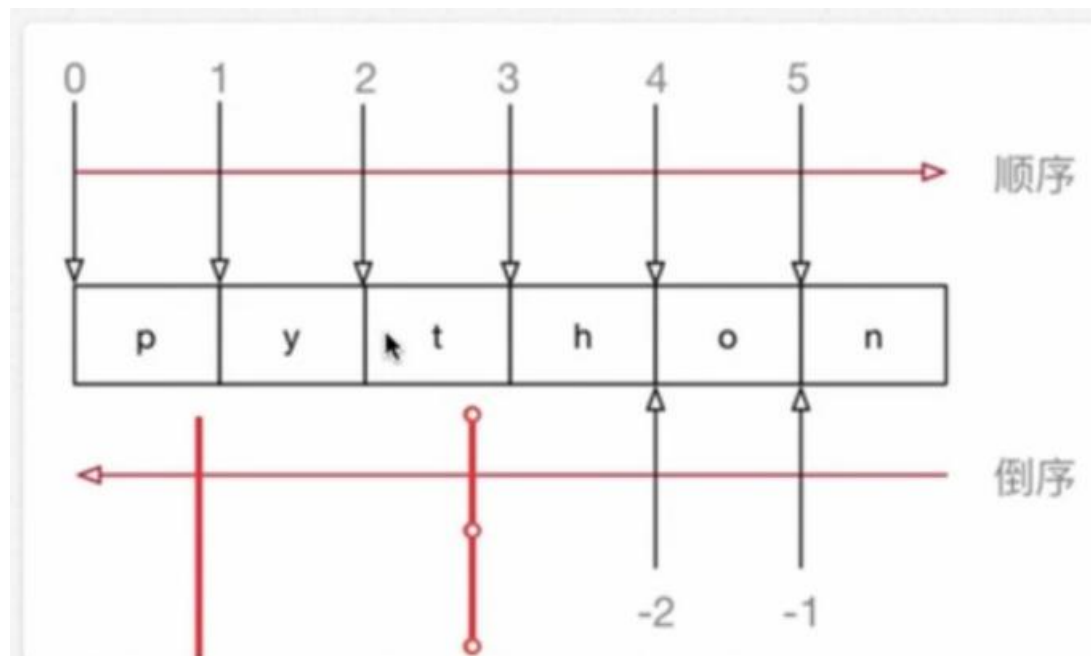
- 在Python中,加了成对引号的字符都被认为是字符串!

- 字符串特性

- 不可修改
- 有索引, 可切片

- 多行字符串

- 字符串拼接



+ 字符串str类型

- 字符串是不可变的
 - python 中的每个对象都存储在内存中的某个位置。 我们可以使用 id() 获得该内存地址

```
>>> str1 = "welcome"  
>>> str2 = "welcome"
```

```
>>> id(str1)  
78965411  
>>> id(str2)  
78965411
```

- str1 和 str2 指的是存储在内存中 某个位置
的相同字符串对象

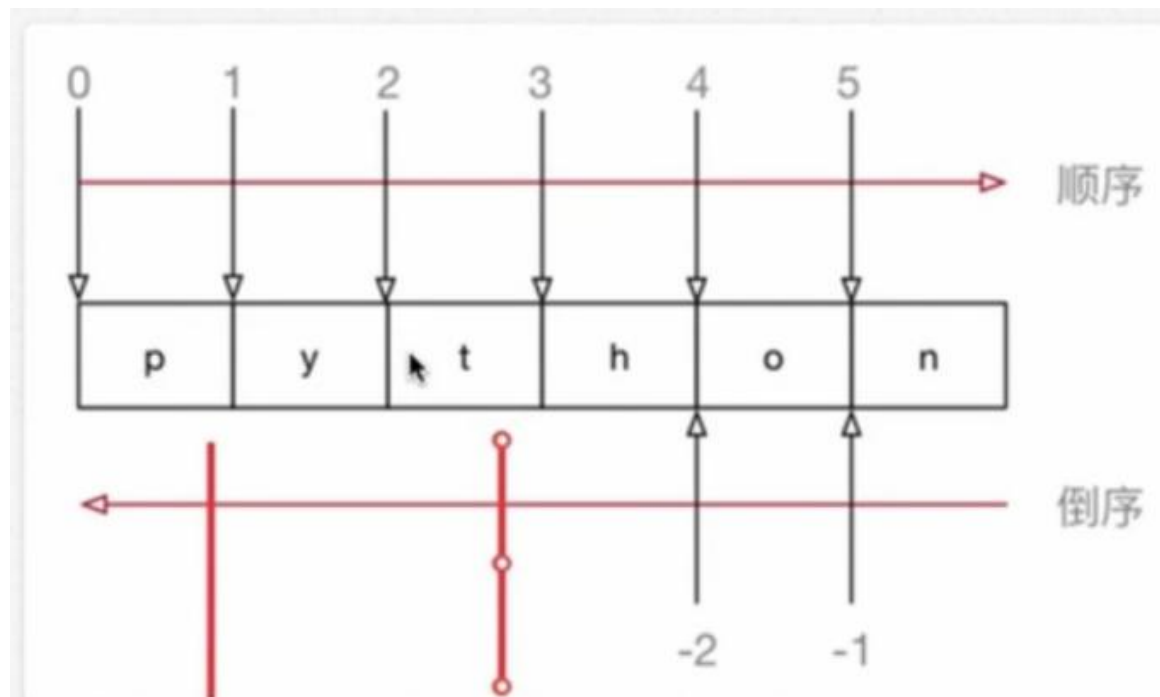


+ 字符串str类型

- 有索引，可切片

- 语法： `s[start:end]`
- 这将从索引 `start` 到索引 `end - 1` 返回字符串的一部分
- `start` 索引和 `end` 索引是可选的
- 如： `s[1:4]` `s[1:]` `s[:4]`

`s[-4:-1]` `s[:-2]`



+ 字符串str类型

- 多行字符串

- 三个引号 `"""` `"""` `'''` `'''`

- 字符串拼接

- `+` 号拼接
 - `*` 数字

➤ 布尔型(bool)

布尔类型很简单，就两个值，主要进行逻辑运算

- 一个True(真)
- 一个False(假)

注意： 第一个字母为大写

+ 列表

- 列表的定义

- 支持字符，数字，字符串甚至可以包含列表（即嵌套）

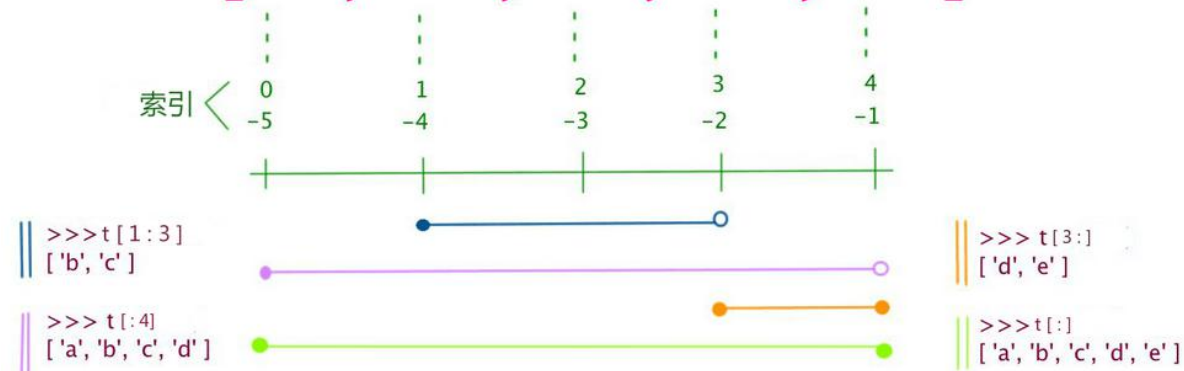
- 列表的索引

- 通过下标来标记元素位置的。下标从0开始，每添加一个元素，就自动+1

- 列表的切割

- 变量 [头下标:尾下标]

t = ['a', 'b', 'c', 'd', 'e']



➕ 列表

- 元素添加-- 通过索引
- 元素追加--append
- 修改 --- 通过索引
- 删除元素 --- del
- 判断元素是否在列表里 in
- 嵌套操作

➕ 读取用户指令和输出


- input() -- 输入
 - 程序会等待输入内容
 - input()方法接收的只是字符串，即使你输入的是数字，它也会按字符串处理
- print() -- 输出
 - 内容直接输入
 - 多个内容用逗号分隔
 - 可进行格式化输出(%字符：标记转换说明符的开始)
 - 默认换行，不换行你应该写成 `print(i, end = " ")`

+ print格式化

引号后的%字符：标记转换说明符的开始

占位符 号	描述
%s	格式化字符串
%d	格式化整数
%f	格式化浮点数字，可指定小数点后的精度

```
>>>str = "the length of (%s) is %d" %('tom',len('tom'))
>>> print(str)
the length of (tom) is 3
```



+ 格式化打印练习

- 现有一练习需求，问用户的姓名、年龄、工作、爱好，然后打印成以下格式

```
----- info of tom -----  
Name : tom  
Age  : 22  
job   : Teacher  
Hobbie: girl  
----- end -----
```

Python 运算符

- 运算按种类可分为算数运算、比较运算、逻辑运算、赋值运算、成员运算、身份运算、位运算，今天我们暂只学习算数运算、比较运算、逻辑运算、赋值运算、成员运算

● 算数运算

a=10, b=20

运算符	描述	实例
+	加 - 两个对象相加	a + b 输出结果 30
-	减 - 得到负数或是一个数减去另一个数	a - b 输出结果 -10
*	乘 - 两个数相乘或是返回一个被重复若干次的字符串	a * b 输出结果 200
/	除 - x除以y	b / a 输出结果 2
%	取模 - 返回除法的余数	b % a 输出结果 0
**	幂 - 返回x的y次幂	a**b 为10的20次方， 输出结果 100000000000000000000
//	取整除 - 返回商的整数部分	9//2 输出结果 4，9.0//2.0 输出结果 4.0

+ Python 运算符

● 比较运算

假设变量：a=10, b=20

运算符	描述	实例
==	等于 - 比较对象是否相等	(a == b) 返回 False。
!=	不等于 - 比较两个对象是否不相等	(a != b) 返回 true。
<>	不等于 - 比较两个对象是否不相等。python3 已废弃。	(a <> b) 返回 true。这个运算符类似 !=。
>	大于 - 返回x是否大于y	(a > b) 返回 False。
<	小于 - 返回x是否小于y。所有比较运算符返回1表示真，返回0表示假。这分别与特殊的变量True和False等价。	(a < b) 返回 true。
>=	大于等于 - 返回x是否大于等于y。	(a >= b) 返回 False。
<=	小于等于 - 返回x是否小于等于y。	(a <= b) 返回 true。

+ Python 运算符

● 赋值运算

以下假设变量：a=10, b=20

运算符	描述	实例
=	简单的赋值运算符	<code>c = a + b</code> 将 <code>a + b</code> 的运算结果赋值为 <code>c</code>
+=	加法赋值运算符	<code>c += a</code> 等效于 <code>c = c + a</code>
-=	减法赋值运算符	<code>c -= a</code> 等效于 <code>c = c - a</code>
*=	乘法赋值运算符	<code>c *= a</code> 等效于 <code>c = c * a</code>
/=	除法赋值运算符	<code>c /= a</code> 等效于 <code>c = c / a</code>
%=	取模赋值运算符	<code>c %= a</code> 等效于 <code>c = c % a</code>
**=	幂赋值运算符	<code>c **= a</code> 等效于 <code>c = c ** a</code>
//=	取整除赋值运算符	<code>c //= a</code> 等效于 <code>c = c // a</code>

+ Python 运算符

● 逻辑运算

以下假设变量：a=10, b=20

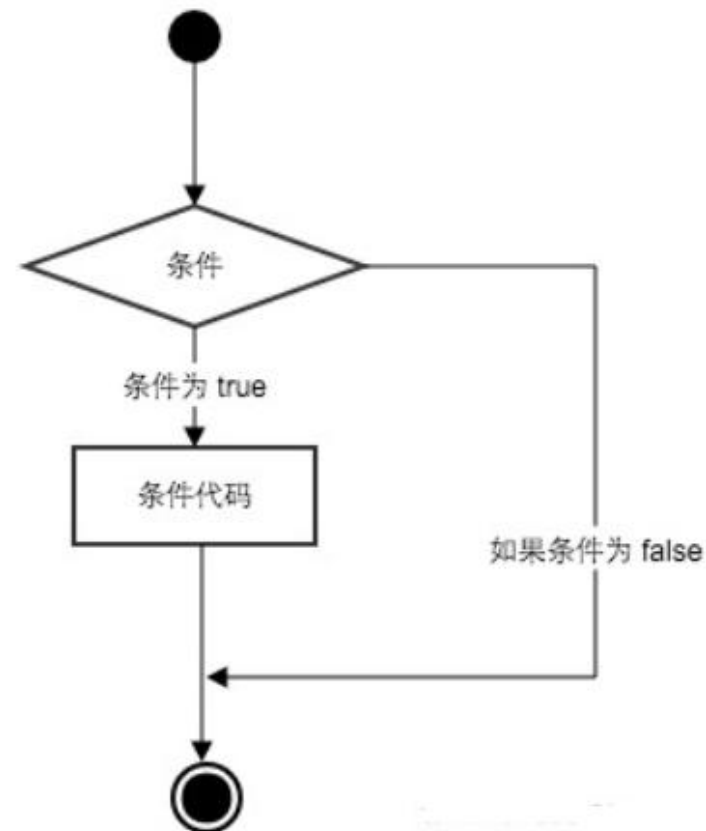
运算符	描述	实例
and	判断多个条件均为真时，返回真	a>10 and b > 10 结果为False
or	判断多个条件任意条件为真时，返回真	a> 10 or b > 10 结果为True
not	取反	not a > b 结果为True

● 成员运算

运算符	描述	实例
in	如果在指定的序列中找到值返回 True，否则返回 False。	x 在 y 序列中，如果 x 在 y 序列中返回 True。
not in	如果在指定的序列中没有找到值返回 True，否则返回 False。	x 不在 y 序列中，如果 x 不在 y 序列中返回 True。

+ 流程控制

- 单分支 if
- 双分支 if else
- 缩进
- 多分支 if elif else



+ 练习

- 1.用print打印出下面内容:

- 文能提笔安天下,
- 武能上马定乾坤.
- 心存谋略何人胜,
- 古今英雄唯是君

- 2. 变量名的命名规范?

- 1.如下那个变量名是正确的? :

- `name = '明明'`
- `_ = tom`
- `_9 = "老司机"`
- `9name = "宝宝"`
- `oldboy(edu = 666`

- 2. 设定一个理想数字比如: 66, 让用户输入数字, 如果比66大, 则显示猜测的结果大了; 如果比66小, 则显示猜测的结果小了;只有等于66, 显示猜测结果正确

+ 练习

- 5.写程序，成绩有ABCDE5个等级，与分数的对应关系如下：
 - A 90-100
 - B 80-89
 - C 60-79
 - D 40-59
 - E 0-39
- 6.查看2、2.22、“小猿圈”分别是什么数据类型的语法是什么？
- 7.实现用户输入用户名和密码,当用户名为 seven 且 密码为 123 时,显示登陆成功,否则登陆失败!

要求用户输入0-100的数字后,
你能正确打印他的对应成绩等级



谢谢大家!