

# Python学习笔记

---

## 第一章 计算机基础

---

### 1.1 硬件

计算机基本的硬件由：CPU / 内存 / 主板 / 硬盘 / 网卡 / 显卡 等组成，只有硬件但硬件之间无法进行交流和通信。

### 1.2 操作系统

操作系统用于协同或控制硬件之间进行工作，常见的操作系统有那些：

- windows
- linux
  - centos 【公司线上一般用】
- mac

### 1.3 解释器或编译器

编程语言的开发者写的一个工具，将用户写的代码转换成010101交给操作系统去执行。

#### 1.3.1 解释和编译型语言

解释型语言就类似于：实时翻译，代表：Python / PHP / Ruby / Perl

编译型语言类似于：说完之后，整体再进行翻译，代表：C / C++ / Java / Go ...

### 1.4 软件（应用程序）

软件又称为应用程序，就是我们在电脑上使用的工具，类似于：记事本 / 图片查看 / 游戏

### 1.5 进制

对于计算机而言无论是文件存储 / 网络传输输入本质上都是：二进制（010101010101），如：电脑上存储视频/图片/文件都是二进制；QQ/微信聊天发送的表情/文字/语言/视频 也全部都是二进制。

进制：

- 2进制，计算机内部。
- 8进制
- 10进制，人来进行使用一般情况下计算机可以获取10进制，然后再内部会自动转换成二进制并操作。
- 16进制，一般用于表示二进制（用更短的内容表示更多的数据），一版是：\x 开头。

二进制	八进制	十进制	十六进制
0	0	0	0
1	1	1	1
10	2	2	2
11	3	3	3

## 第二章 Python入门

---

### 2.1 环境的安装

- 解释器：py2 / py3 （环境变量）
- 开发工具：pycharm

### 2.2 编码

#### 2.2.1 编码基础

- ascii
- unicode
- utf-8
- gbk
- gb2312

#### 2.2.2 python编码相关

对于Python默认解释器编码：

- py2: ascii
- py3: utf-8

如果想要修改默认编码，则可以使用：

```
# -*- coding:utf-8 -*-
```

注意：对于操作文件时，要按照：以什么编写写入，就要用什么编码去打开。

## 2.3 变量

问：为什么要有变量？

为某个值创建一个“外号”，以后在使用时候通过此外号就可以直接调用。

# 第三章 数据类型

## 3.1 整型 (int)

### 3.1.1 整型的长度

py2中有：int/long

py3中有：int (int/long)

### 3.1.2 整除

py2和py3中整除是不一样。

## 3.2 布尔 (bool)

布尔值就是用于表示真假。True和False。

其他类型转换成布尔值：

- str
- ...

对于：None / "" / 0 .... -> false

## 3.3 字符串 (str)

字符串是写代码中最常见的，python内存中的字符串是按照：unicode 编码存储。对于字符串是不可变。

字符串自己有很多方法，如：

1. 大写：upper

```

v = 'ALEX'
v1 = v.upper()
print(v1)
v2 = v.isupper() # 判断是否全部是大写
print(v2)

```

## 2. 小写: lower

```

v = 'alex'
v1 = v.lower()
print(v1)
v2 = v.islower() # 判断是否全部是小写
print(v2)

```

```

##### 了解即可
v = 'ß'
# 将字符串变小写 (更牛逼)
v1 = v.casefold()
print(v1) # ss
v2 = v.lower()
print(v2)

```

## 3. 判断是否是数字: isdecimal

```

v = '1'
# v = '二'
# v = '②'
v1 = v.isdigit() # '1' -> True; '二' -> False; '②' --> True
v2 = v.isdecimal() # '1' -> True; '二' -> False; '②' --> False
v3 = v.isnumeric() # '1' -> True; '二' -> True; '②' --> True
print(v1, v2, v3)
# 以后推荐用 isdecimal 判断是否是 10进制的数。

# ##### 应用 #####

v = ['alex', 'eric', 'tony']

for i in v:
    print(i)

num = input('请输入序号: ')
if num.isdecimal():
    num = int(num)
    print(v[num])
else:
    print('你输入的不是数字')

```

## 4. 去空白+\\t+\\n + 指定字符串

```
v1 = "alex "  
print(v1.strip())  
  
v2 = "alex\t"  
print(v2.strip())  
  
v3 = "alex\n"  
print(v3.strip())  
  
v1 = "alexa"  
print(v1.strip('a'))
```

5. 替换 replace
6. 开头 / 结尾
7. 编码，把字符串转换成二进制
8. format
9. join
10. split
11. 其他【可选】

## 3.4 列表

## 3.8 公共功能

- len
- 索引
- 切片
- 步长
- for循环

## 3.9 嵌套

# 第四章 文件操作

---

## 4.1 文件基本操作

```
obj = open('路径', mode='模式', encoding='编码')  
obj.write()  
obj.read()  
obj.close()
```

## 4.2 打开模式

- r / w / a
- r+ / w+ / a+
- rb / wb / ab
- r+b / w+b / a+b

## 4.3 操作

- read(), 全部读到内存
- read(1)
  - 1表示一个字符

```
obj = open('a.txt',mode='r',encoding='utf-8')
data = obj.read(1) # 1个字符
obj.close()
print(data)
```

- 1表示一个字节

```
obj = open('a.txt',mode='rb')
data = obj.read(3) # 1个字节
obj.close()
```

- write(字符串)

```
obj = open('a.txt',mode='w',encoding='utf-8')
obj.write('中午你')
obj.close()
```

- write(二进制)

```
obj = open('a.txt',mode='wb')

# obj.write('中午你'.encode('utf-8'))
v = '中午你'.encode('utf-8')
obj.write(v)

obj.close()
```

- seek(光标字节位置), 无论模式是否带b, 都是按照字节进行处理。

```
obj = open('a.txt',mode='r',encoding='utf-8')
obj.seek(3) # 跳转到指定字节位置
data = obj.read()
obj.close()

print(data)
```

```
obj = open('a.txt',mode='rb')
obj.seek(3) # 跳转到指定字节位置
data = obj.read()
obj.close()

print(data)
```

- tell(), 获取光标当前所在的字节位置

```
obj = open('a.txt',mode='rb')
# obj.seek(3) # 跳转到指定字节位置
obj.read()
data = obj.tell()
print(data)
obj.close()
```

- flush, 强制将内存中的数据写入到硬盘

```
v = open('a.txt',mode='a',encoding='utf-8')
while True:
    val = input('请输入: ')
    v.write(val)
    v.flush()

v.close()
```

## 4.4 关闭文件

文艺青年

```
v = open('a.txt',mode='a',encoding='utf-8')

v.close()
```

二逼

```
with open('a.txt',mode='a',encoding='utf-8') as v:
    data = v.read()
    # 缩进中的代码执行完毕后, 自动关闭文件
```

## 4.5 文件内容的修改

```
with open('a.txt',mode='r',encoding='utf-8') as f1:
    data = f1.read()
    new_data = data.replace('飞洒','666')

with open('a.txt',mode='w',encoding='utf-8') as f1:
    data = f1.write(new_data)
```

大文件修改

```
f1 = open('a.txt',mode='r',encoding='utf-8')
f2 = open('b.txt',mode='w',encoding='utf-8')

for line in f1:
    new_line = line.replace('阿斯','死啊')
    f2.write(new_line)
f1.close()
f2.close()
```

```
with open('a.txt',mode='r',encoding='utf-8') as f1, open('c.txt',mode='w',encoding='utf-8')
as f2:
    for line in f1:
        new_line = line.replace('阿斯','死啊')
        f2.write(new_line)
```

## 第五章 函数

---

## 第六章 模块

---

## 第七章 面向对象

---

## 第八章 网络编程

---

## 第九章 并发编程

---

## 第十章 数据库

---

## 第十一章 前端开发

---

## 第十二章 Django框架

---

## 附录 常见错误和单词

---



## 单词











3.