

1. 今日内容大纲

- pycharm的安装以及简单使用
 - 辅助开发软件，代码逐行调试，设置高端，不会提示，你在书写代码时，他不提示，debug的模式，最好用的还是pycharm。
- **格式化输出**
- **while循环**
- 运算符 and or not
- 编码的初识

2. 昨日内容回顾

- 编译型与解释型
 - 编译型：一次性编译成2进制，在执行
 - 优点：执行效率高
 - 缺点：不能跨平台，开发效率低
 - 代表语言：C
 - 解释型：逐行解释成二进制，在执行
 - 优点：可以跨平台，开发效率高
 - 缺点：执行效率低。
 - 代表语言：python
- 变量：
 - 数字，字母，下划线任意组合。
 - 不能以数字开头。
 - 不能用Python的关键字：print if...
 - 不能使用中文。
 - 描述性。
 - 区分变量与数据类型的区别。

```
name = 'Alex'  
name = '太白'  
print(name)  
name = 'wusir'  
print(name)
```

- 常量
 - 一直不变的量，与变量几乎一样。
- 注释：解释说明
- 基础数据类型：

- 1, 2, 3, 4000, int 数字, +-* / % **
- 'fdsalk中国' str 字符串 + *int
- True False bool 布尔值
- 用户输入input

```
name = input('>>>')
print(type(name))
```

- if
 - if 条件:
 - if else:
 - if elif elif
 - if elif elif else
 - if 嵌套

3. 今日内容

1. while 循环

- why: 大气循环, 吃饭, 上课, 睡觉, 日复一日, 歌曲列表循序环, 程序中: 输入用户名密码,
- what: while 无限循环。
- how:

1. 基本结构:

```
while 条件:
    循环体
```

2. 初识循环

```
while True:
    print('狼的诱惑')
    print('我们不一样')
    print('月亮之上')
    print('庐州月')
    print('人间')
```

3. 基本原理:



4. 循环如何终止?

1. 改变条件。

```
flag = True
while flag:
    print('狼的诱惑')
    print('我们不一样')
    print('月亮之上')
    flag = False
    print('庐州月')
    print('人间')
```

```
# 练习题: 1~ 100 所有的数字
count = 1
flag = True
while flag:
    print(count)
    count = count + 1
    if count == 101:
        flag = False

count = 1
while count < 101:
    print(count)
    count = count + 1
```

1 + 2 + 3 + 100 的最终结果:

```
s = 0
count = 1
while count < 101:
    s = s + count
    count = count + 1
print(s)
```

2. break

```
# while True:
#     print('狼的诱惑')
#     print('我们不一样')
#     print('月亮之上')
#     break
#     print('庐州月')
#     print('人间')
# print(111)
```

3. 系统命令（今天不讲）

4. continue

```
# continue : 退出本次循环，继续下一次循环
flag = True
while flag:
    print(111)
    print(222)
    flag = False
    continue
    print(333)
```

```
# while else: while 循环如果被break打断，则不执行else语句。
count = 1
while count < 5:
    print(count)
    if count == 2:
        break
    count = count + 1
else:
    print(666)
```

- where: 你需要重复之前的动作，输入用户名密码，考虑到while循环。

2. 格式化输出

- 当你遇到这样的需求：字符串中想让某些位置变成动态可传入的，首先要考虑到格式化输出。

3. 运算符：算数运算符 + -，比较运算符 > ==，赋值运算符=,+=,逻辑运算符，and or，成员运算符。

```

i1 = 2
i2 = 3
print(2 ** 3)
print(10 // 3)
print(10 % 3)

print(3 != 4)

count = 1
count = count + 1
count += 1
print(count)

```

```

# and or not

# 1 在没有()的情况下, 优先级: not > and > or, 同一优先级从左至右依次计算
# 情况1: 两边都是比较运算
# print(2 > 1 and 3 < 4 or 4 > 5 and 2 < 1)
# print(True or False)

# 情况2: 两边都是整数
'''
x or y , x为真, 值就是x, x为假, 值是y
'''

# print(1 or 2)
# print(3 or 2)
# print(4 or 2)
# print(-1 or 2)
# print(0 or 2)

# print(1 and 2)

```

数据类型之间的转换

```

# str ----> int : 只能是纯数字组成的字符串
s1 = '00100'
print(int(s1))
# int ----> str
i1 = 100
print(str(i1), type(str(i1)))

# int ----> bool : 非零即True , 0为False.
i = 0
print(bool(i))
# bool ----> int
print(int(True)) # 1
print(int(False)) # 0

```

4. 编码的初识重点

计算机存储文件，存储数据，以及将一些数据信息通过网络发送出去，存储发送数据什么内容？底层都是01010101.

我带这张理穿越，1937，我俩研究电报：

真正密码本：

滴滴 走

滴滴滴 跑

第一版：没有段位，

101 今

1101 晚

1 吃

11 鸡

1011101111

第二版：

0000101 今

0001101 晚

0000001 吃

0000011 鸡

0000101 0001101 0000001 0000011

密码本：01010110 二进制与 文字之间的对应关系。

最早起的密码本：

ASCII码：只包含：英文字母，数字，特殊字符。

0000 0001 : a

0000 0101 : ;

8bit == 1byte

'hello123': 8byte

gbk: 英文字母，数字，特殊字符和中文。国标

一个英文字母：0000 0001 : a

一个中文 中：0000 0001 0100 0001 : 中

Unicode: 万国码：把世界上所有的文字都记录到这个密码本。

起初一个字符用2个字节表示：

0000 0001 0000 0011： a

0000 0001 0100 0001： 中

后来为了涵盖全部文字：

0000 0001 0000 0011 0000 0001 0000 0011： a

0000 0001 0100 0001 0000 0001 0000 0011： 中

浪费空间，浪费资源。

Utf-8:升级：最少用8bit1个字节表示一个字符。

0000 0011： a 1字节

0000 0011 0000 0011 欧洲 2个字节

0000 0011 0000 0011 0000 0011 中： 3个字节。

重点：

'中国12he'： GBK: 8个字节

'中国12he'： UTF-8: 10个字节

```
8bit = 1byte
1024byte = 1KB
1024KB = 1MB
1024MB = 1GB
1024GB = 1TB
1024TB = 1PB
1024PB = 1EB
1024EB = 1ZB
1024ZB = 1YB
1024YB = 1NB
1024NB = 1DB
```

7.6MB ----> $7.6 * 1024 * 1024 * 8$

明日内容：

1. 二进制与十进制之间的转换
2. str bool int 转换
3. str具体操作方法：索引切片步长，常用操作方法，
4. for 循环