

一、课程安排

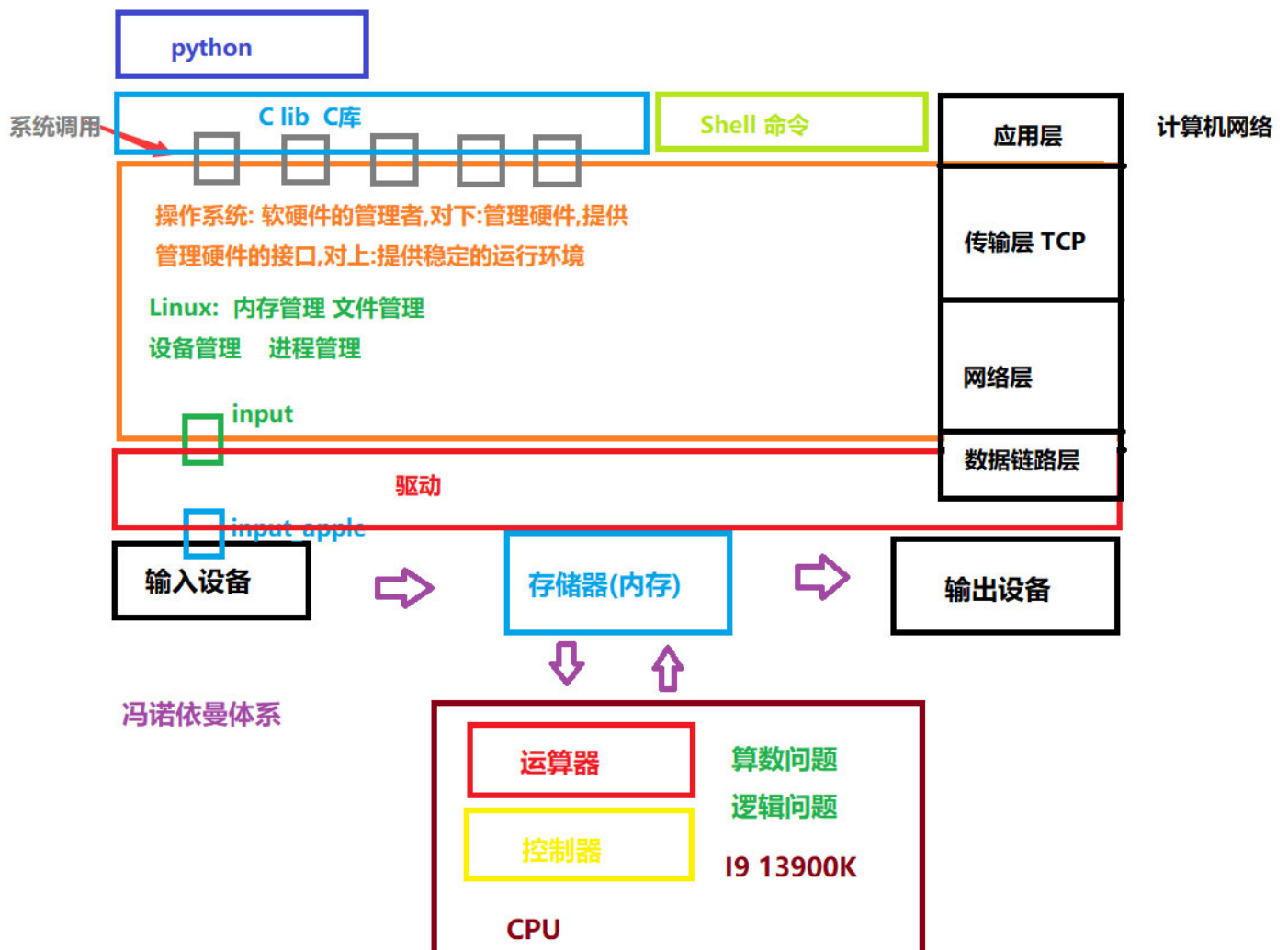
一、课程安排

- a. day1 ~ day4 python基础课程
- b. day5 ~ day7 人工智能

二、今日内容

- a. 计算机的组成
- b. python简介
- c. python环境安装
- d. python基础语法
- e. python逻辑语句

三、计算机体系



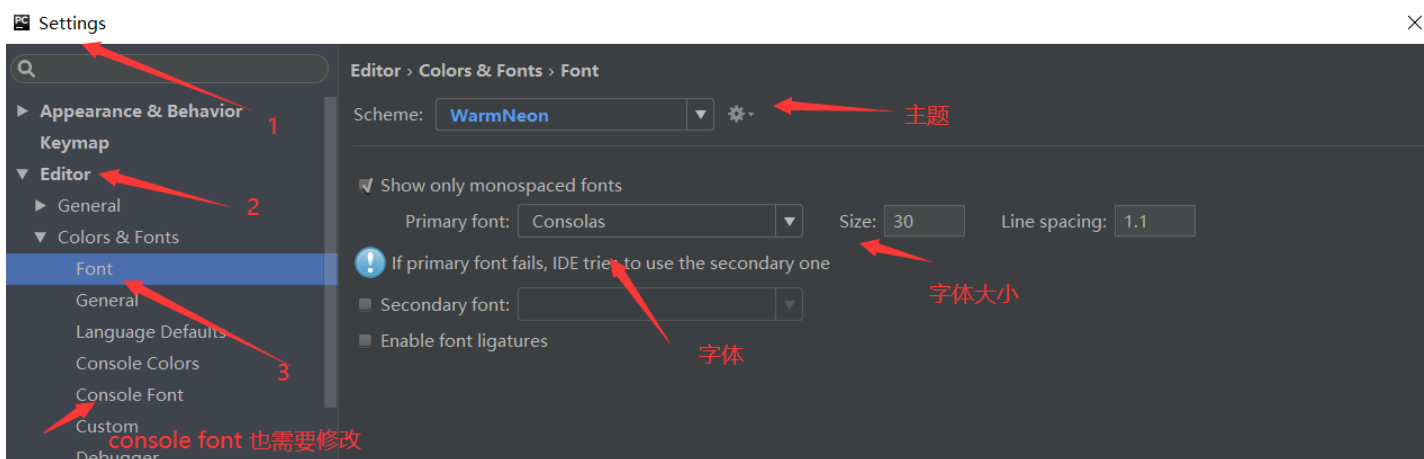
四、python简介

- a. python作者： 龟叔
- b. python发布年份： 1991年正式发布
- c. python版本： python2 python3
- d. python优点： 简单易学 python 高层语言 面向对象语言 开源
- e. python缺点： 相对于第一梯度的语言会慢很多
- f. python在嵌入式行业引用
 - i. python脚本文件配合工作
 - ii. python与C语言混编
 - iii. python在人工智能的应用

五、python环境安装

六、pycharm使用

- a. 主题设置： file ---> setting ---> editor



- b. 创建目录：项目目录名右击 ---> new ----> python package ---> 输入目录名称
- c. 创建文件：项目目录名右击 ---> new ----> python file ---> 输入文件名

七、hello world程序

7.1 code

```
1 # python当中书写hello world程序
2 print("hello world")
```

7.2 代码运行

- i. 项目文件中 ---> 右击 ---> run object
- ii. python + pathname

八、python语法

8.1 注释

- i. 单行注释：以 # 开头的内容就是单行注释. 快捷方式： ctrl + /
- ii. 多行注释： 三引号之间的内容就是多行注释 快捷方式： 鼠标选中注释的行， ctrl + /
 - 1. python当中单引号与双引号的作用相同
 - 2. 单引号存在的意义： 方便套用

8.2 变量

- i. 概念：程序运行过程中，可以发送改变的量，称之为变量
- ii. 特点： 无需提前声明,即拿即用

8.3 标识符

- i. 定义：程序员自定义的符号，可以是函数名，变量名。
- ii. 规则：由数字,字母，下划线组成，数字不能开头。 不能与关键字重合
- iii. 规范：
 - 1. 见名知意
 - 2. 驼峰命名法
 - a. 大驼峰：每个单词的首字母大写。 类名
 - b. 小驼峰： 函数名，变量 windows环境下
 - 3. 通过下划线，将每一个单词连接起来， linux 环境下的编程。
- iv. 书籍推荐：高质量c/c++编程

8.4 运算符

8.4.1 算数运算符

- 1. /：除法： $9 / 2 = 4.5$
- 2. //： 整除： $9 // 2 = 4$
- 3. **: 幂运算： $2 ** 10$
- 4. =： 赋值运算符
- 5. python中没有 ++ -- 这种带有赋作用的运算符

```
1 # python支持同时赋值
2 a, b = 10, 20 # 将 10赋值给a, 20赋值给b
3 """
4 int a = 10;
5 int b = 5;
6
7 方案1:
```

```

8  int c = a;
9  a = b;
10 b = a;
11
12 方案2:
13 a = a + b
14 b = a - b
15 a = a - b
16
17 方法3:   ^ 性质: 0 异或任何数都能以任何数, 两个相同的数字异或 为 0
18           异或与顺序无关.
19
20 a = a ^ b
21 b = a ^ b    a ^ b ^ b
22 a = a ^ b    a ^ b ^ a
23
24 # 有一个数组,里面的值都是成对儿出现的,只有一个值是单独出现的, 找出这个数字,
25 [1, 2, 3, 4, 4,3,2, 1, 5,7]
26
27 思路:
28 <1>. 将数据进行分组 [5, 7] [1,1,2,2,3,3,4,4]
29 [5, 1,1, 3,3]
30 [7, 2,2, 4,4]
31
32 5 : 0000 0101
33 7 : 0000 0111
34 -----
35      0000 0010
36
37 ""
38
39 l1 = [1, 2, 3, 4, 4, 3, 2, 1, 5789789789, 78767687]
40
41 ret = 0
42
43 # 整体异或 ,找到不同的位,组成新的数字
44 for i in l1:
45
46     ret ^= i
47
48 print(ret)
49
50 i = 0
51 # int 32位,找打不相同的哪一位数字

```

```
52 while i < 32: #int 32位
53
54     if (ret >> i) & 1 == 1:
55         break
56
57     i += 1
58
59 print(i)
60
61 dog1 = 0
62 dog2 = 0
63
64 for x in l1:
65
66     # 将位中为1的放到1组, 为0 的放到另一组, 然后异或
67     if (x >> i) & 1 == 1:
68         dog1 ^= x
69     else:
70         dog2 ^=x
71
72 print(dog2)
73 print(dog1)
74
75
76 #include <stdio.h>
77
78 int main()
79 {
80
81     // 定义数组
82     int array[10] = {1,1,2,2,3,3,4,4,5,6};
83
84     // 全部异或
85     int i = 0;
86     int ret = 0;
87     for(i = 0; i < 10; i++)
88     {
89         ret ^= array[i];
90     }
91
92     // 找到ret当中哪一位为1
93     int pos = 0;
94
95     while (pos < 32)
```

```

96     {
97         if ((ret >> pos) & 1 == 1)
98         {
99             break;
100        }
101        pos++;
102    }
103    int dog1 = 0;
104    int dog2 = 0;
105    // 分组
106    for(i = 0; i < 10; i++)
107    {
108        if ((array[i] >> pos & 1) == 0)
109        {
110            dog1^= array[i];
111        }
112        else
113        {
114            dog2^=array[i];
115        }
116    }
117
118    printf("dog1 = %d, dog2 = %d\n",dog1, dog2);
119
120
121
122
123    return 0;
124 }
125

```

8.4.2 逻辑运算符

1. and : 逻辑与,两真且为真
2. or : 有真就为真
3. not: 真假互换

8.5 输入输出

8.5.1 print 输出

1. python支持直接输出变量
2. python支持格式化输出

3. format函数,填充输出
4. print会自动换行,因为print拥有参数 end 来表示是否换行
5. print支持一个函数输出多个内容

```
1 apple = 200
2
3 # 直接输出变量 apple
4 print(apple)
5
6 name = "小白"
7 age = 18
8 # 需要严格按照格式控制输出
9 print("my name is %s, age is %d"%(name, age))
10
11 # format格式输出: format是一个函数, 用于替换的函数
12 print("my name is {}, age is {}".format(name, name))
13
14 # print换行输出
15 print("hello world", end=" ")
16 print("88888888888888888888")
17
18 # print支持在同一行输出多个内容,多个内容中使用, 隔开
19 print("你好","I love study, study make my mother happy","my mother happy, my famliy happ")
20
21 print("name : ",name)
```

8.5.2 输入

```
1 # python中输入使用 input函数进行输出
2 """
3 input函数使用
4
5 功能: 从终端中输入数据, 到程序中.
6
7 使用方式:
8     result = input(提示语句)
9
10 返回值:
11     result 输入的结果.
12     注意: 无论从终端输入何种类型的数据, 最终都是以字符串的形式展示的
```

```
13
14 MD5 --> 单向不可逆加密
15
16 """
17
18 # 情景模拟：去银行取款
19
20 password = input("请输入您的取款密码:>>>")
21
22 print("您输入的密码: >> ", password)
```