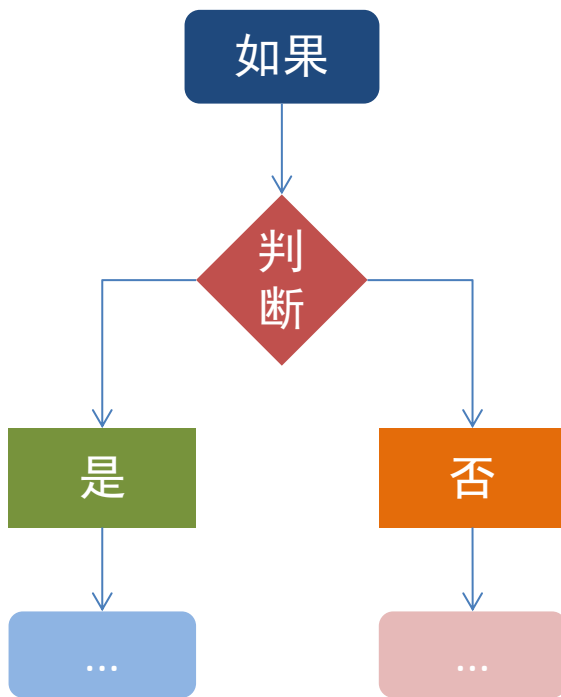


Python判断语句



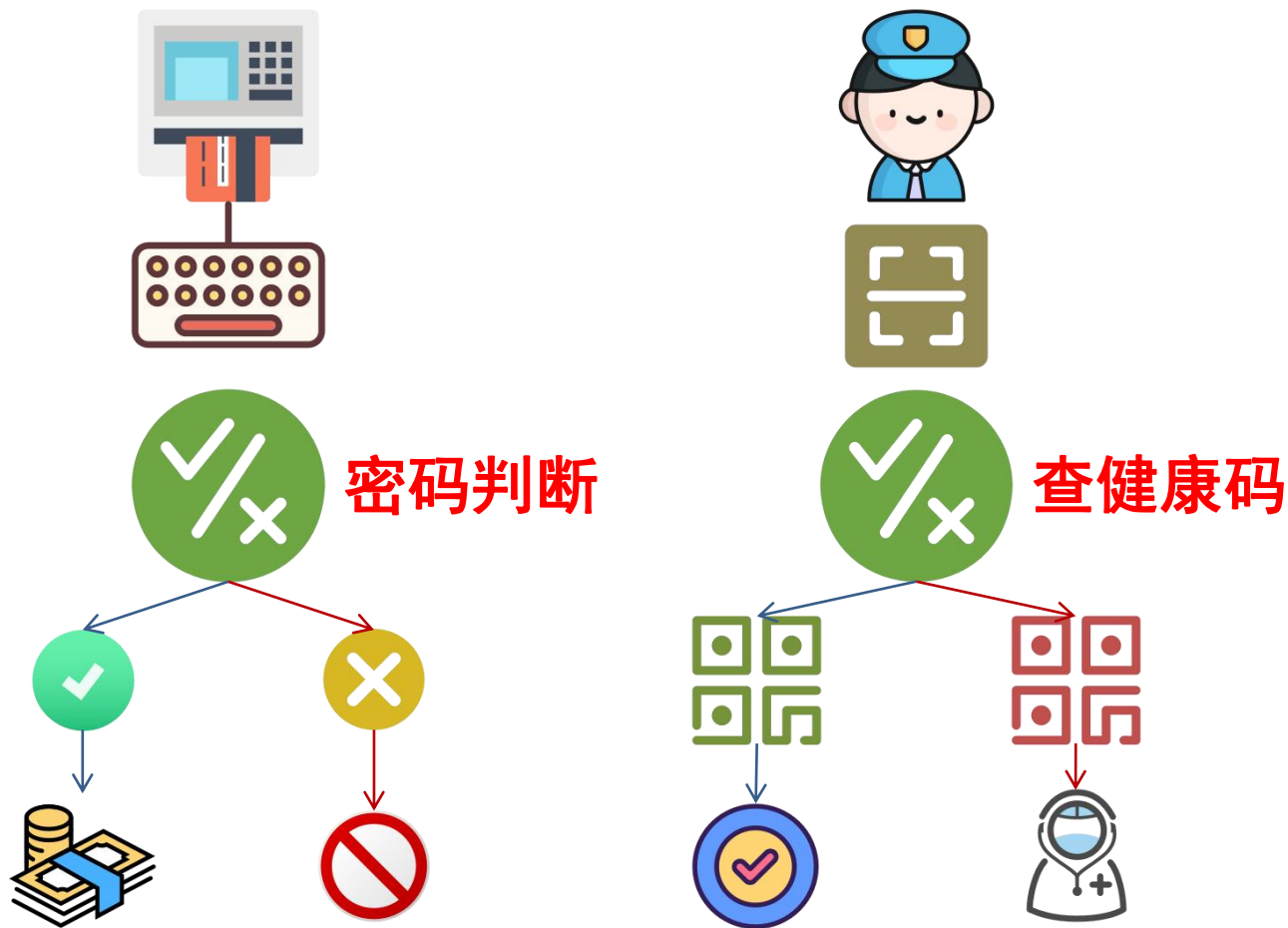
前言

进行逻辑判断，是生活中常见的行为。同样，在程序中，进行逻辑判断也是最为基础的功能。



为什么学习判断语句

判断在程序中广泛应用，如果没有它，这些功能都难以实现。



网站登陆

APP登陆

指纹识别

人脸解锁

景区验票

影院验票

门禁刷卡

车牌识别

.....

判断是程序最基础
最核心的逻辑功能



目录

Contents



◆ 布尔类型和比较运算符

- ◆ if语句的基本格式
- ◆ if else 语句
- ◆ if elif else语句
- ◆ 判断语句的嵌套
- ◆ 实战案例



学习目标

Learning Objectives

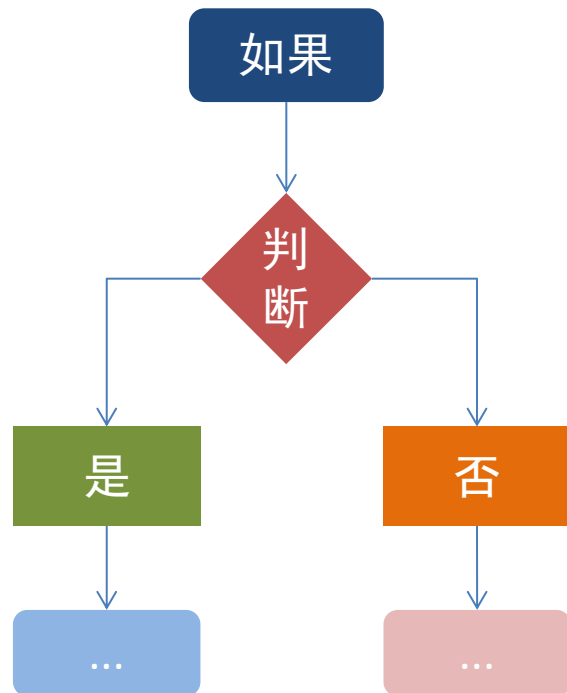
掌握布尔类型用于表示：真和假

掌握比较运算符用于计算：真和假

布尔类型

进行判断，只有2个结果：

- 是
- 否



程序中，如何描述：**是或否**呢？

使用：**布尔类型**

布尔类型

Python中常用的有**6**种值（数据）的类型

类型	描述	说明
数字 (Number)	<p>支持</p> <ul style="list-style-type: none">• 整数 (int)• 浮点数 (float)• 复数 (complex)• 布尔 (bool)	整数 (int)，如：10、-10
		浮点数 (float)，如：13.14、-13.14
		复数 (complex)，如：4+3j，以j结尾表示复数
		布尔 (bool) 表达现实生活中的逻辑，即真和假 <ul style="list-style-type: none">• True表示真• False表示假。 True本质上是一个数字记作1，False记作0
字符串 (String)	描述文本的一种数据类型	字符串 (string) 由任意数量的字符组成
列表 (List)	有序的可变序列	Python中使用最频繁的数据类型，可有序记录一堆数据
元组 (Tuple)	有序的不可变序列	可有序记录一堆不可变的Python数据集合
集合 (Set)	无序不重复集合	可无序记录一堆不重复的Python数据集合
字典 (Dictionary)	无序Key-Value集合	可无序记录一堆Key-Value型的Python数据集合



布尔类型的定义

布尔类型的字面量:

- True 表示真（是、肯定）
- False 表示假（否、否定）

定义变量存储布尔类型数据:

```
变量名称 = 布尔类型字面量
```

**布尔类型不仅可以自行定义
同时也可以通过计算的来。**

**也就是使用比较运算符进行比较运算得到布
尔类型的结果。**



比较运算符

布尔类型的数据，不仅可以通过定义得到，也可以通过比较运算符进行内容比较得到。

如下代码：

```
result = 10 > 5
print(f"10 > 5 的结果是: {result}, 类型是: {type(result)}")
```

test x

```
D:\dev\python\python3.10.4\python.exe D:/python-learn/03_10 > 5 的结果是: True, 类型是: <class 'bool'>
```

```
result = "itcast" == "itheima"
print(f"字符串itcast是否和itheima相等, 结果是: {result}, 类型是: {type(result)}")
```

test x

```
D:\dev\python\python3.10.4\python.exe D:/python-learn/03_Python判断语句/test.py
字符串itcast是否和itheima相等, 结果是: False, 类型是: <class 'bool'>
```



比较运算符

运算符	描述	示例
==	判断内容 是否相等 ，满足为True，不满足为False	如a=3,b=3，则(a == b) 为 True
!=	判断内容 是否不相等 ，满足为True，不满足为False	如a=1,b=3，则(a != b) 为 True
>	判断运算符左侧内容 是否大于 右侧 满足为True，不满足为False	如a=7,b=3，则(a > b) 为 True
<	判断运算符左侧内容 是否小于 右侧 满足为True，不满足为False	如a=3,b=7，则(a < b) 为 True
>=	判断运算符左侧内容 是否大于等于 右侧 满足为True，不满足为False	如a=3,b=3，则(a >= b) 为 True
<=	判断运算符左侧内容 是否小于等于 右侧 满足为True，不满足为False	如a=3,b=3，则(a <= b) 为 True



总结

1. 在Python中，可以表示真假的数据类型是：

布尔类型，字面量True表示真，字面量False表示假

2. 除了可以定义布尔类型外，还可以通过____计算得到布尔类型？

通过<比较运算符>计算得到布尔类型的结果

- == 判断是否相等，!= 判断是否不相等
- >判断是否大于，<判断是否小于
- >=判断是否大于等于，<=判断是否小于等于



目录

Contents



- ◆ 布尔类型和比较运算符
- ◆ if语句的基本格式
- ◆ if else 语句
- ◆ if elif else语句
- ◆ 判断语句的嵌套
- ◆ 实战案例

学习目标

Learning Objectives

掌握逻辑判断语句（if）的基本语法格式

掌握布尔类型数据，在判断语句中的作用

if判断语句

生活中的判断



如果小美喜欢我
明天我就去表白

程序中的判断

`if` 要判断的条件：
条件成立时，要做的事情

```
# 定义变量  
age = 30  
  
# 进行判断  
if age >= 18:  
    print("我已经成年了")
```



if判断语句

```
age = 18

print(f"今年我已经{age}岁了")
if age >= 18:
    print("我已经成年了")
    print("即将步入大学生活")

print("时间过的真快")
```

当if判断条件结果为True时，代码会被执行
为False时，代码不会执行

当age为18时，执行结果：

```
D:\dev\python\python
今年我已经30岁了
我已经成年了
即将步入大学生活
时间过的真快
```

当age为10时，执行结果：

```
D:\dev\python\python
今年我已经10岁了
时间过的真快
```



if语句的注意点

```
age = 18

print(f"今年我已经{age}岁了")
if age >= 18:
    print("我已经成年了")
    print("即将步入大学生活")

print("时间过的真快")
```

判断语句的结果，必须是布尔类型True或False

True会执行if内的代码语句

False则不会执行



if语句的注意点

```
print(f"今年我已经{age}岁了")  
if age >= 18: → 这个引号不要忘记  
    print("我已经成年了")  
    print("即将步入大学生活")  
  
print("时间过的真快")
```

归属于if判断的代码语句块，需在前方填充4个空格缩进

Python通过缩进判断代码块的归属关系。





总结

1. if语句的基本格式

`if` 要判断的条件:

条件成立时，要做的事情

2. if语句的注意事项:

- 判断条件的结果一定要是布尔类型
- 不要忘记判断条件后的： 引号
- 归属于if语句的代码块，需在前方填充4个空格缩进

练习**练习案例：成年人判断**

结合前面学习的input输入语句，完成如下案例：

1. 通过input语句，获取键盘输入，为变量age赋值。（注意转换成数字类型）
2. 通过if判断是否是成年人，满足条件则输出提示信息，如下：

```
欢迎来到黑马儿童游乐场，儿童免费，成人收费。  
请输入你的年龄：30  
您已成年，游玩需要补票10元。  
祝您游玩愉快。
```

提示：您已成年，需要补票的信息输出，来自if判断



目录

Contents



- ◆ 布尔类型和比较运算符
- ◆ if语句的基本格式
- ◆ **if else 语句**
- ◆ if elif else语句
- ◆ 判断语句的嵌套
- ◆ 实战案例



学习目标

Learning Objectives

掌握if else 语句的组合用法



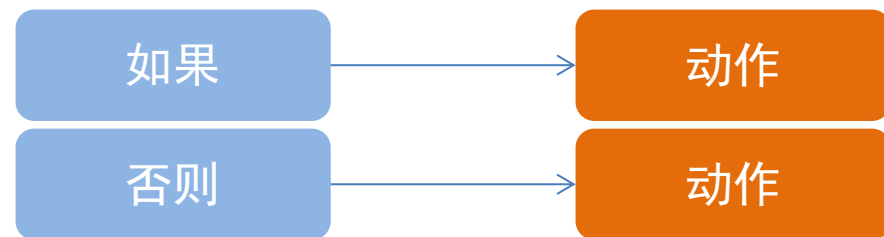
if满足条件会执行相应的代码语句，如果不满足呢？

有没有不满足的情况下，可供执行的代码呢？

if else 语句可以实现

if else语句

生活中的判断



如果小美喜欢我
明天我就去表白

否则我就去
追求小新，哼

程序中的判断

`if` 条件:

满足条件时要做的事情1

满足条件时要做的事情2

满足条件时要做的事情3

... (省略) ...

`else`:

不满足条件时要做的事情1

不满足条件时要做的事情2

不满足条件时要做的事情3

... (省略) ...



if else语句

```
print("欢迎来到黑马儿童游乐场，儿童免费，成人收费。")
age = int(input("请输入你的年龄："))
if age >= 18:
    print("您已成年，游玩需要补票10元。")
else:
    print("您未成年，可以免费游玩。")

print("祝您游玩愉快。")
```

条件成立时执行

条件不成立时执行



if else语句注意点

```
print("欢迎来到黑马儿童游乐场，儿童免费，成人收费。")
age = int(input("请输入你的年龄："))
if age >= 18:
    print("您已成年，游玩需要补票10元。")
else:
    print("您未成年，可以免费游玩。")
print("祝您游玩愉快。")
```

else, 不需要判断条件

同样需要4个空格，作为代码的缩进哦

1. else后，不需要判断条件
2. 和if的代码块一样，else的代码块同样需要4个空格作为缩进





总结

1. if else 语句，其中

- if和其代码块，条件满足时执行
- else搭配if的判断条件，当不满足的时候执行

2. if else语句的注意事项：

- else不需要判断条件，当if的条件不满足时，else执行
- else的代码块，同样要4个空格作为缩进

练习**练习案例：我要买票吗**

通过input语句获取键盘输入的身高

判断身高是否超过120cm，并通过print给出提示信息。

欢迎来到黑马动物园。
请输入你的身高（cm）：**130**
您的身高超出**120cm**，游玩需要够票**10元**。
祝您游玩愉快。

欢迎来到黑马动物园。
请输入你的身高（cm）：**111**
您的身高未超出**120cm**，可以免费游玩。
祝您游玩愉快。





目录

Contents

- ◆ 布尔类型和比较运算符
- ◆ if语句的基本格式
- ◆ if else 语句
-  ◆ if elif else语句
- ◆ 判断语句的嵌套
- ◆ 实战案例



学习目标

Learning Objectives

掌握if elif else语句进行多条件判断的语法



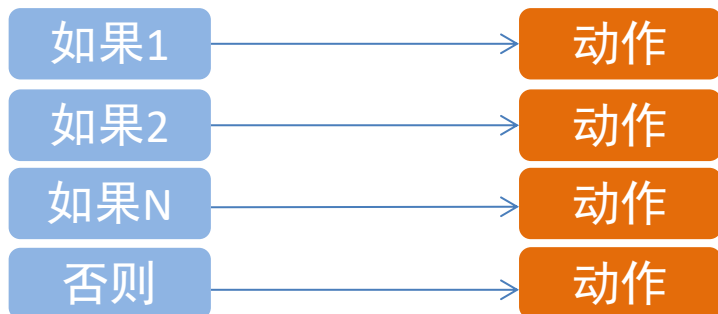
某些场景下，判断条件不止一个，可能有多个。

这种需求能用Python实现吗？

if elif else 语句可以实现

if else语句

生活中的判断



如果小美喜欢我
明天我就去表白

或者小甜喜欢我
明天我就去表白

否则我就去
追求小新，哼

程序中的判断

```
if 条件1:  
    条件1满足应做的事情  
    条件1满足应做的事情  
    .....  
elif 条件2:  
    条件2满足应做的事情  
    条件2满足应做的事情  
    .....  
elif 条件N:  
    条件N满足应做的事情  
    条件N满足应做的事情  
    .....  
else:  
    所有条件都不满足应做的事情  
    所有条件都不满足应做的事情  
    .....
```

if elif else语句

```
print("欢迎来到黑马动物园。")
height = int(input("请输入你的身高 (cm): "))
vip_level = int(input("请输入你的vip级别(1~5): "))
if height < 120:
    print("您的身高小于120CM, 可以免费游玩。")
elif vip_level > 3:
    print("您的vip级别大于3, 可以免费游玩。")
else:
    print("不好意思, 所有条件都不满足, 需要购票10元。")

print("祝您游玩愉快。")
```



if elif else语句 注意点

```
print("欢迎来到黑马动物园。")
height = int(input("请输入你的身高 (cm): "))
vip_level = int(input("请输入你的vip级别(1~5): "))
day = int(input("请输入今天的日期(1~30): "))
if height < 120:
    print("您的身高小于120CM, 可以免费游玩。")
elif vip_level > 3:
    print("您的vip级别大于3, 可以免费游玩。")
elif day == 1:
    print("今天是1号免费日, 可以免费游玩。")
else:
    print("不好意思, 所有条件都不满足, 需要购票10元。")

print("祝您游玩愉快。")
```

elif 语句可以写多个。



if elif else语句 注意点

```
print("欢迎来到黑马动物园。")
height = int(input("请输入你的身高(cm): "))
vip_level = int(input("请输入你的vip级别(1~5): "))
day = int(input("请输入今天的日期(1~30): "))
if height < 120:
    print("您的身高小于120CM, 可以免费游玩。")
elif vip_level > 3:
    print("您的vip级别大于3, 可以免费游玩。")
elif day == 1:
    print("今天是1号免费日, 可以免费游玩。")
else:
    print("不好意思, 所有条件都不满足, 需要购票10元。")
print("祝您游玩愉快。")
```

判断是互斥且有顺序的。

- **满足1（如图编号）将不会理会2和3**
- **满足2，将不会理会3**
- **1、2、3均不满足，进入else**
- **else也可以省略不写，效果等同3个独立的if判断**



if elif else语句 注意点

```
if height < 120:  
    print("您的身高小于120CM, 可以免费游玩。")  
elif vip_level > 3:  
    print("您的vip级别大于3, 可以免费游玩。")  
elif day == 1:  
    print("今天是1号免费日, 可以免费游玩。")  
else:  
    print("不好意思, 所有条件都不满足, 需要购票10元。")
```

空格缩进同样不可省略



if elif else语句 注意点

```
print("欢迎来到黑马动物园。")
if int(input("请输入你的vip级别(1~5): ")) < 120:
    print("您的身高小于120CM, 可以免费游玩。")
elif int(input("请输入今天的日期(1~30): ")) > 3:
    print("您的vip级别大于3, 可以免费游玩。")
elif int(input("请输入你的身高(cm): ")) == 1:
    print("今天是1号免费日, 可以免费游玩。")
else:
    print("不好意思, 所有条件都不满足, 需要购票10元。")

print("祝您游玩愉快。")
```

可以如上图，将input输入语句直接写入判断条件中

。

节省代码量





总结

1. if elif else语句的作用是?

可以完成多个条件的判断

2. 使用if elif else的注意点有:

- elif可以写多个
- 判断是互斥且有序的，上一个满足后面的就不会判断了
- 可以在条件判断中，直接写input语句，节省代码量

练习

练习案例：猜猜心里数字

1. 定义一个变量，数字类型，内容随意。
2. 基于input语句输入猜想的数字，通过if和多次elif的组合，判断猜想数字是否和心里数字一致。

```
D:\dev\python\python3.10.4\python
请输入第一次猜想的数字： 1
不对，再猜一次： 2
不对，再猜最后一次： 3
Sorry，全部猜错啦，我想的是： 10
```





目录

Contents

- ◆ 布尔类型和比较运算符
- ◆ if语句的基本格式
- ◆ if else 语句
- ◆ if elif else语句
- ◆ 判断语句的嵌套
- ◆ 实战案例



学习目标

Learning Objectives

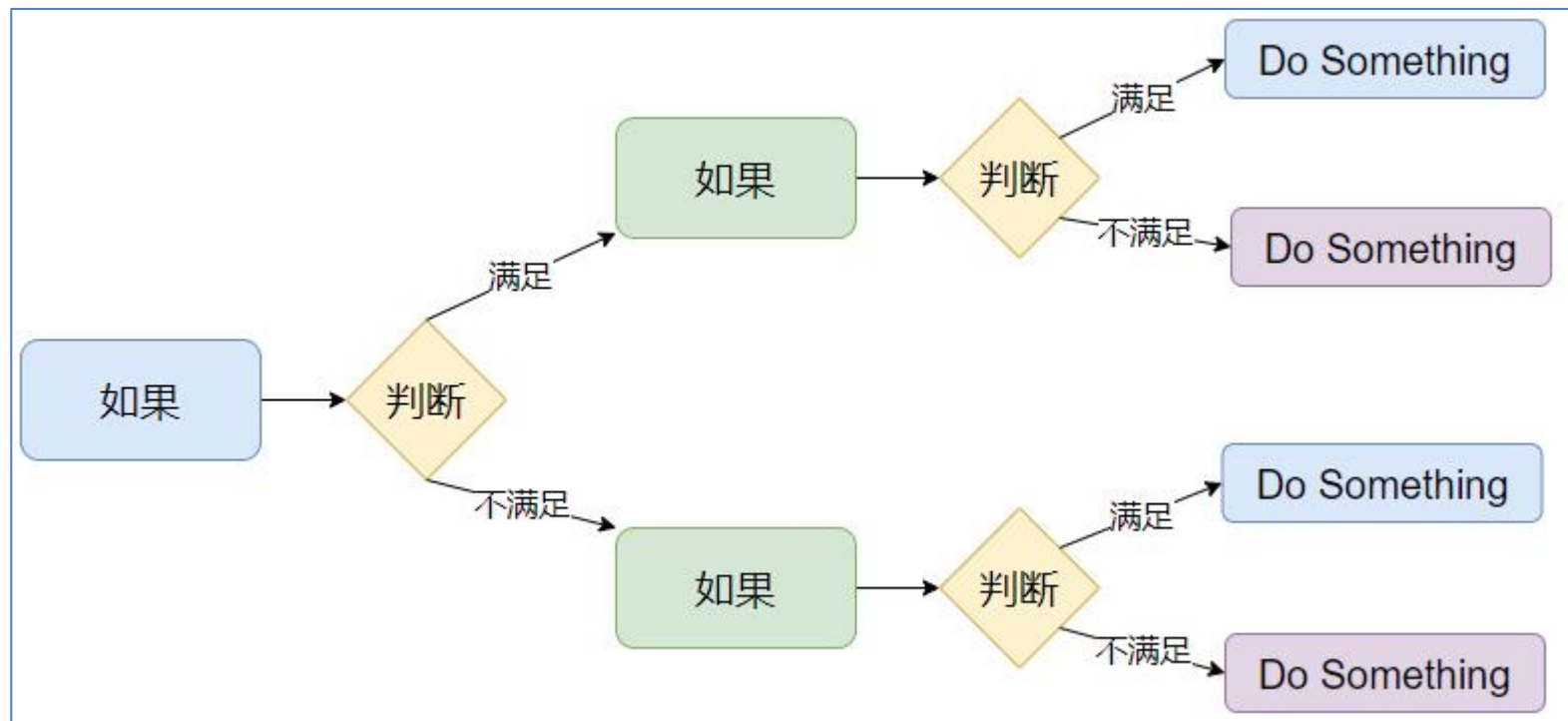
掌握Python判断语句的终极用法：嵌套使用



有很多场景，不仅仅是多个并列条件，还会有**满足前置条件才会二次判断**的多层判断需求。

对这种需求，嵌套判断语句可以实现

判断语句的嵌套



许多逻辑的判断，是嵌套的，多层次的。

对于这种需求，我们可以：自由组合 `if elif else`，完成特定需求的要求



判断语句的嵌套

基础语法格式如下：

```
if 条件1:
```

```
    满足条件1 做的事情1
```

```
    满足条件1 做的事情2
```

```
    if 条件2:
```

```
        满足条件2 做的事情1
```

```
        满足条件2 做的事情2
```

如上图，第二个if，属于第一个if内，只有第一个if满足条件，才会执行第二个if

嵌套的关键点，在于：**空格缩进**

通过空格缩进，来决定语句之间的：**层次关系**



判断语句的嵌套

简单嵌套:

```
print("欢迎来到黑马动物园。")
if int(input("输入你的身高: "))1 > 120:
    print("你的身高大于120cm, 不可以免费")
    print("不过如果你的vip等级高于3, 可以免费游玩")

    if int(input("请告诉我你的vip级别: "))2 > 3:
        print("恭喜你, 你的vip级别大于3, 可以免费游玩。")
    else:
        print("Sorry, 你需要补票, 10元。")
else:
    print("欢迎你小朋友, 可以免费游玩。")
```

如图:

- 判断有2层
- 当外层if满足条件（图中编号1）时，才会执行内层if判断（图中编号2）
- 当外层if（编号1）不满足，直接执行外层else



判断语句的嵌套

自由组合嵌套，需求如下：

公司要发礼物，条件是：

1. 必须是大于等于18岁小于30岁的成年人
2. 同时入职时间需满足大于两年，或者级别大于3才可领取

```
if age >= 18:
    print("成年人符合，继续判断")
    if age < 30:
        print("年龄达标继续判断")
        if year > 2:
            print("小于30岁的成年人且入职超过2年，满足条件，可以领取")
        else:
            print("Sorry, 年龄符合，但入职时间不足")
    elif level > 3:
        print("级别大于3的成年人可直接领取礼物")
    else:
        print("您的年龄过大或级别小于等于3，不可领取")
else:
    print("Sorry, 未成年不可领取礼物")
```

如图：

- **if elif else 可以自由组合**
- **满足缩进的要求即可**





总结

1. 嵌套判断语句可以用于多条件、多层次的逻辑判断
2. 嵌套判断语句可以根据需求，自由组合if elif else来构建多层次判断
3. 嵌套判断语句，一定要注意空格缩进，Python通过空格缩进来决定层次关系



目录

Contents

- ◆ 布尔类型和比较运算符
- ◆ if语句的基本格式
- ◆ if else 语句
- ◆ if elif else语句
- ◆ 判断语句的嵌套
- ◆ 实战案例



学习目标

Learning Objectives

通过逻辑判断语句，完成猜数字的案例代码编写

案例需求:

定义一个数字（1~10，随机产生），通过3次判断来猜出来数字

案例要求:

1. 数字随机产生，范围1-10
2. 有3次机会猜测数字，通过3层嵌套判断实现
3. 每次猜不中，会提示大了或小了

提示，通过如下代码，可以定义一个变量num，变量内存储随机数字。

```
import random  
num = random.randint(1, 10)
```





传智教育旗下高端IT教育品牌