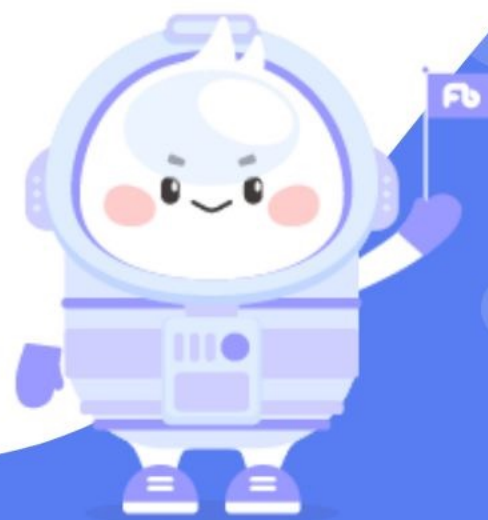


《信息技术》

Python程序设计 2/4

► 讲师：孙珍珍

更多干货关注  粉笔教师教育  粉笔教师



✿ 复习一下



◆ 指数值的补码进行运算，但输出的是原码

| 运算符 | 名称 | 规则 |
|-----|----|-----------|
| & | 与 | 全1为1，有0为0 |
| | 或 | 全0为0，有1为1 |
| ^ | 异或 | 不同为1 |
| ~ | 取反 | 取相反值 |
| << | 左移 | 右侧空余位补0 |
| >> | 右移 | 右侧移除位删掉 |

```
1 a = 10
2 b = 23
3 print("a&b=", a & b)
4 print("a|b=", a | b)
5 print("a^b=", a ^ b)
6 print("~a=", ~a)
7 print("a<<2=", a << 2)
8 print("a>>2=", a >> 2)
```

```
a&b= 2
a|b= 31
a^b= 29
~a= -11
a<<2= 40
a>>2= 2
```

0 0 0 0 1 0 1 0

0 0 0 1 0 1 1 1

0 0 0 0 1 0 1 0

0 0 0 1 0 1 1 1

0 0 0 0 1 0 1 0

0 0 0 1 0 1 1 1

◆ 指数值的补码进行运算，但输出的是原码

| 运算符 | 名称 | 规则 |
|-----|----|-----------|
| & | 与 | 全1为1，有0为0 |
| | 或 | 全0为0，有1为1 |
| ^ | 异或 | 不同为1 |
| ~ | 取反 | 取相反值 |
| << | 左移 | 右侧空余位补0 |
| >> | 右移 | 右侧移除位删掉 |

```
1 a = 10
2 b = 23
3 print("a&b=", a & b)
4 print("a|b=", a | b)
5 print("a^b=", a ^ b)
6 print("~a=", ~a)
7 print("a<<2=", a << 2)
8 print("a>>2=", a >> 2)
```

```
a&b= 2
a|b= 31
a^b= 29
~a= -11
a<<2= 40
a>>2= 2
```

00001010

◆ 指数值的补码进行运算，但输出的是原码

| 运算符 | 名称 | 规则 |
|-----|----|-----------|
| & | 与 | 全1为1，有0为0 |
| | 或 | 全0为0，有1为1 |
| ^ | 异或 | 不同为1 |
| ~ | 取反 | 取相反值 |
| << | 左移 | 右侧空余位补0 |
| >> | 右移 | 右侧移除位删掉 |

00001010

```
1 a = 10
2 b = 23
3 print("a&b=", a & b)
4 print("a|b=", a | b)
5 print("a^b=", a ^ b)
6 print("~a=", ~a)
7 print("a<<2=", a << 2)
8 print("a>>2=", a >> 2)
```

00001010

```
a&b= 2
a|b= 31
a^b= 29
~a= -11
a<<2= 40
a>>2= 2
```

(一) 输出-print

1. 基本要求：

- (1) 输出提示内容，放在引号里面；原样输出；
- (2) 输出变量时，不需要加引号；输出变量的值；
- (3) 各项信息之间，逗号隔开；
- (4) print默认换行，可使用end=""实现不换行

```
1 a = 3
2 print("输出a的值：")
3 print('a = ', a)
```

```
输出a的值：
a = 3
```

```
1 a = 3
2 print("输出a的值：", end=" ")
3 print('a = ', a)
```

```
输出a的值： a = 3
```

```
1 a = 3
2 b = 4.567
3 print("a = ", a, "\nb = ", b)
```

```
a = 3
b = 4.567
```

2. 格式要求 -- (1) % 格式化

P351

FB 粉笔

"XXX%[标志][0][输出最小宽度][. 精度] 格式字符 " % 变量

| | % | 标志 [+/-] | 补位 [0] | 输出最小宽度 [m] | .精度 [.n] | 格式字符 |
|----|----|--------------------|-------------------|-------------------------------------|---|--|
| 说明 | 必有 | + / 省：右对齐 -：左对齐 | 省：空位补空格 0：空位补0 | 实际位数多，按实际输出； 否则用0/空格补齐 (谁大听谁) | 保留的小数位数 (四舍五入) 或 输出字符的位数 (直接截断) | d/o/x： 十/八/十六进制 f：浮点数 c：一个字符 s：字符串 |

格式字符

+
宽度
+
对齐
+
补位

【整型】

```
a = 15
print("0: a=%d" % a)
print("1: a=%5d" % a)
print("2: a=%05d" % a)
print("3: a=%-5d" % a)
print("4: a=%o" % a)
print("5: a=%X" % a)
```

```
0: a=15
1: a=   15
2: a=00015
3: a=15
4: a=17
5: a=F
```

2. 格式要求 -- (1) % 格式化

P351

粉笔

"XXX%[标志][0][输出最小宽度][.精度] 格式字符" % 变量

| | % | 标志 [+/-] | 补位 [0] | 输出最小宽度 [m] | .精度 [.n] | 格式字符 |
|----|----|--------------------|-------------------|-------------------------------------|---|--|
| 说明 | 必有 | + / 省：右对齐 -：左对齐 | 省：空位补空格 0：空位补0 | 实际位数多，按实际输出； 否则用0/空格补齐 (谁大听谁) | 保留的小数位数 (四舍五入) 或 输出字符的位数 (直接截断) | d/o/x： 十/八/十六进制 f：浮点数 c：一个字符 s：字符串 |

格式字符

+
宽度
+
精度

【浮点型】

```
b = 1234.5678
print("0: b=%f" % b)
print("1: b=%8.2f" % b)
print("2: b=%4.2f" % b)
print("3: b=%.2f" % b)
```

```
0: b=1234.567800
1: b= 1234.57
2: b=1234.57
3: b=1234.57
```


2. 格式要求 -- (1) % 格式化

P351

FB 粉笔

"XXX%[标志][0][输出最小宽度][. 精度] 格式字符 " % 变量

| | % | 标志 [+/-] | 补位 [0] | 输出最小宽度 [m] | .精度 [.n] | 格式字符 |
|----|----|--------------------|-------------------|-------------------------------------|---|--|
| 说明 | 必有 | + / 省：右对齐 -：左对齐 | 省：空位补空格 0：空位补0 | 实际位数多，按实际输出； 否则用0/空格补齐 (谁大听谁) | 保留的小数位数 (四舍五入) 或 输出字符的位数 (直接截断) | d/o/x： 十/八/十六进制 f：浮点数 c：一个字符 s：字符串 |

格式字符
+
宽度
+
精度

【字符串型】

```
s1 = "I"
s2 = "I love IT"
print("1:s = %c" % s1)
print("2:s = %s" % s2)
print("3:s2[3] = %c" % s2[3])
print("4:s = %10s" % s2)
print("5:s = %.5s" % s2)
```

```
1:s = I
2:s = I love IT
3:s2[3] = o
4:s = I love IT
5:s = I lov
```

输出语句

★ 基础

- 格式: ——— ("提示信息", 参数)
- 提示信息: ——— 用 ——— 括起来 ——— 原样 ———
- 参数: ——— 需要引号括起来 ——— 变量或表达式 ——— 输出其 ———
- 细节: ——— 各项信息用 ——— 隔开 ——— 默认换行否? ——— 不换行函数 ———

格式化

- 格式: ——— "XXX % [对齐-] [补位0] [宽度m] [.精度n] 格式字符" ——— 参数
- 格式字符
 - 类型: ——— 十进制整数 ——— 浮点数 ——— 字符 ——— 字符串
 - 符号: ——— ——— ——— ——— ——— ———
- %
 - 位数: m.n
 - %f: 小数部分默认保留 ——— 位
 - n: 小数部分 ———; 字符串部分 ———
 - m: 与实际宽度相比 ——— (小数点占 ——— 位)
 - 补位方式: ——— 省略则补空格, 0则补0
 - 对齐方式: ——— 省略则右对齐, - 则左对齐
 - 其它说明: ——— 若参数个数 ≥ 2 个, 则需要用 ——— 括起来
- format ⑬



输出语句

★ 基础

- 格式: `print("提示信息", 参数)`
- 提示信息: 用【引号】括起来 — 原样输出
- 参数: 不需要引号括起来 — 变量或表达式 — 输出其最终的结果
- 细节: 各项信息用【逗号】隔开 — 默认【换行】 — 不换行函数 `end=''`

格式化

格式: `"XXX % [对齐-] [补位0] [宽度m] [.精度n] 格式字符" % 参数`

格式字符

- 类型: 十进制整数 — 浮点数 — 字符 — 字符串
- 符号: `d` — `f` — `c` — `s`

%

位数: m.n

- `%f`: 小数部分默认保留6位
- `n`: 小数部分【四舍五入】; 字符串部分 直接截断
- `m`: 与实际宽度相比【谁大听谁】 (小数点占【1】位)

补位方式:

- 省略则补空格, 0则补0

对齐方式:

- 省略则右对齐, - 则左对齐

其它说明:

- 若参数个数 ≥ 2 个, 则需要用圆括号括起来

format ⑬



(一) 输出

应用下

Fb 粉笔

◆ `a=23 , b=45.678 , c='hi'`

"XXX {参数序号:格式控制标记}XXX" .format(参数)

| | 参数序号 | : 格式控制标记 | | | |
|----|----------------|---------------------|---|--|---|
| | | 补位[0] | 宽度[m] | 精度[.n] | 格式字符 |
| 说明 | 指明值的 位置 | 省: 空位补空格 0: 空位补0 | 实际位数多, 按实际输出; 否则用0/空格补齐 (谁大听谁) | 保留的小数位数 (四舍五入) 或 输出字符的位数 (直接截断) | d: 十进制 (可省) o/x: 八/十六进制 f: 浮点数 c: 一个字符 s: 字符串 |

【参数序号】

```
>>> print("你好, {1}, 你的成绩为{0}~".format(98,"张三"))
你好, 张三, 你的成绩为 98~
>>> print("你好, {}, 你的成绩为{}~".format(98,"张三"))
你好, 98, 你的成绩为张三~
```

"XXX {参数序号:格式控制标记}XXX" .format(参数)

| | 参数序号 | : 格式控制标记 | | | |
|----|--------|---------------------|--------------------------------------|---|---|
| | | 补位[0] | 宽度[m] | 精度[.n] | 格式字符 |
| 说明 | 指明值的位置 | 省: 空位补空格 0: 空位补0 | 实际位数多, 按实际输出; 否则用0/空格补齐 (谁大听谁) | 保留的小数位数 (四舍五入) 或 输出字符的位数 (直接截断) | d: 十进制 (可省) o/x: 八/十六进制 f: 浮点数 s: 字符串 (可省) |

格式字符

+
宽度
+
补位

【整型】

```
a = 15
print("1:a={0}".format(a))
print("2:a={0:5}".format(a))
print("3:a={0:05}".format(a))
print("4:a={0:o}".format(a))
print("5:a={0:X}".format(a))
```

```
1:a=15
2:a=   15
3:a=00015
4:a=17
5:a=F
```

"XXX {参数序号:格式控制标记}XXX" .format(参数)

| | 参数序号 | : 格式控制标记 | | | |
|----|--------|---------------------|--------------------------------------|---|---|
| | | 补位[0] | 宽度[m] | 精度[.n] | 格式字符 |
| 说明 | 指明值的位置 | 省: 空位补空格 0: 空位补0 | 实际位数多, 按实际输出; 否则用0/空格补齐 (谁大听谁) | 保留的小数位数 (四舍五入) 或 输出字符的位数 (直接截断) | d: 十进制 (可省) o/x: 八/十六进制 f: 浮点数 s: 字符串 (可省) |

格式字符
+
宽度
+
精度

【浮点型】

```
b = 1234.5678
print("1:b={0}".format(b))
print("2:b={0:f}".format(b))
print("3:b={0:8.2f}".format(b))
print("4:b={0:4.2f}".format(b))
print("5:b={0:.2f}".format(b))
```

```
1:b=1234.5678
2:b=1234.567800
3:b= 1234.57
4:b=1234.57
5:b=1234.57
```

"XXX {参数序号:格式控制标记}XXX" .format(参数)

| | 参数序号 | : 格式控制标记 | | | |
|----|--------|-------------------|---------------------------------------|---|---|
| | | 补位[0] | 宽度[m] | 精度[n] | 格式字符 |
| 说明 | 指明值的位置 | 省：空位补空格 0：空位补0 | 实际位数多，按实际输出； 否则用0/空格补齐 (谁大听谁) | 保留的小数位数 (四舍五入) 或 输出字符的位数 (直接截断) | d：十进制 (可省) o/x：八/十六进制 f：浮点数 s：字符串 (可省) |

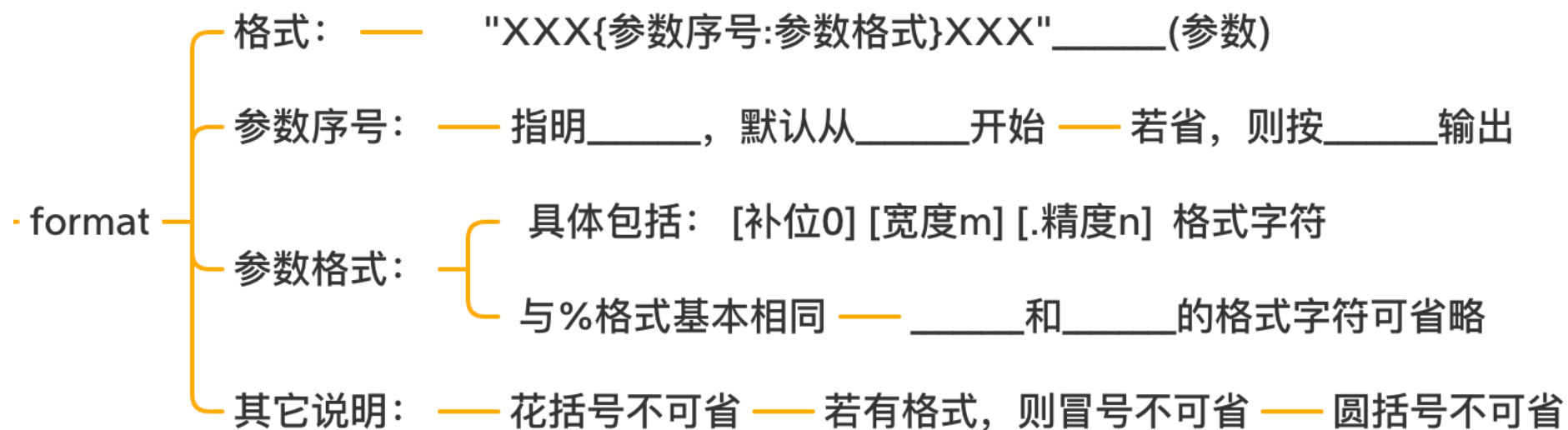
格式字符
+
精度

【字符串型】

```
s = "I love IT"
print("1:s={0}".format(s))
print("2:s={0:.5}".format(s))
```

```
1:s=I love IT
2:s=I lov
```


◆ a=23 , b=45.678 , c='hi'



◆input() 获取的是 “字符型”

```
1 name = input("请输入你的姓名:")
2 print('您刚刚输入的姓名是:', name)
```

```
请输入你的姓名:szz
您刚刚输入的姓名是: szz
```

【例】输入两个数并计算和

```
1 a = input("请输入第一个数:")
2 b = input("请输入第二个数:")
3 c = a + b
4 print("两数之和为:", c)
```

```
请输入第一个数:2
请输入第二个数:3
两数之和为: 23
```

```
1 a = int(input("请输入第一个数:"))
2 b = int(input("请输入第二个数:"))
3 c = a + b
4 print("两数之和为:", c)
```

```
请输入第一个数:2
请输入第二个数:3
两数之和为: 5
```

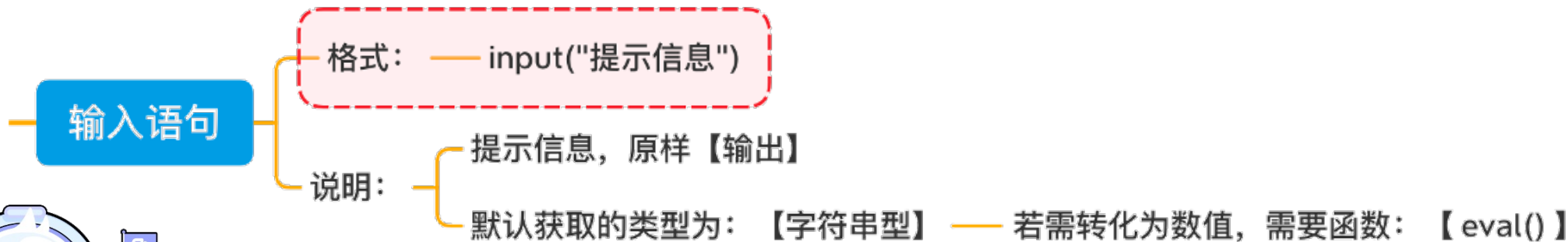
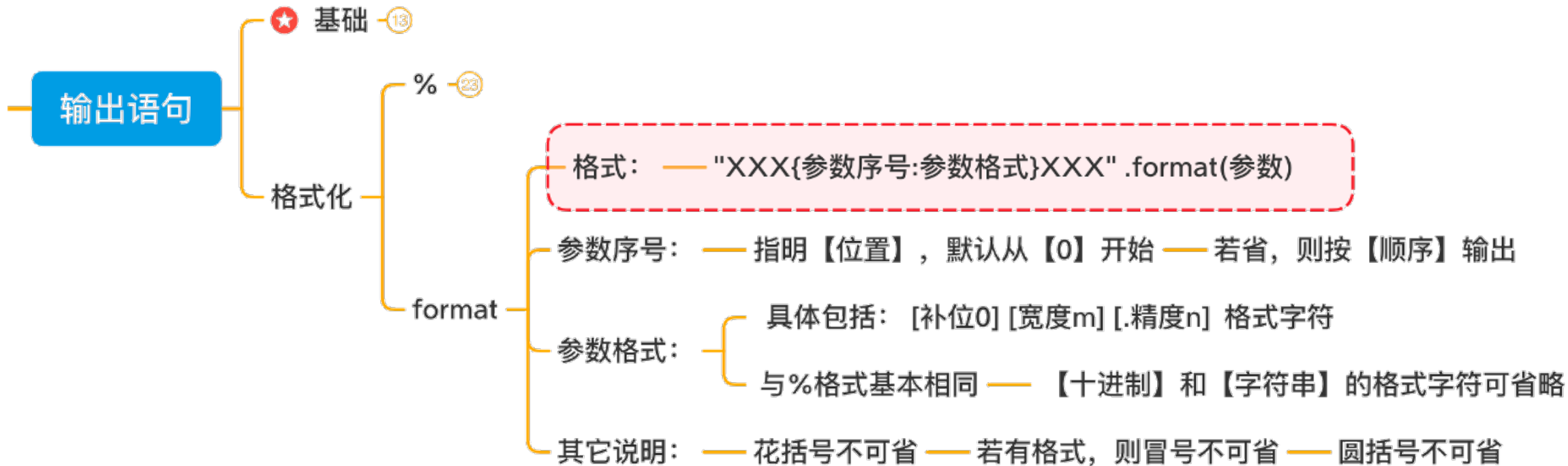
◆input() 获取的是 “字符型”

```
1 a = float(input("请输入第一个数:"))
2 b = float(input("请输入第二个数:"))
3 c = a + b
4 print("两数之和为:", c)
```

```
5
请输入第一个数:2.2
请输入第二个数:3.3
两数之和为: 5.5
```

```
1 a = eval(input("请输入第一个数:"))
2 b = eval(input("请输入第二个数:"))
3 c = a + b
4 print("两数之和为:", c)
```

```
5
请输入第一个数:2
请输入第二个数:3.3
两数之和为: 5.3
```





第二节 选择结构程序设计

一、单分支if 语句

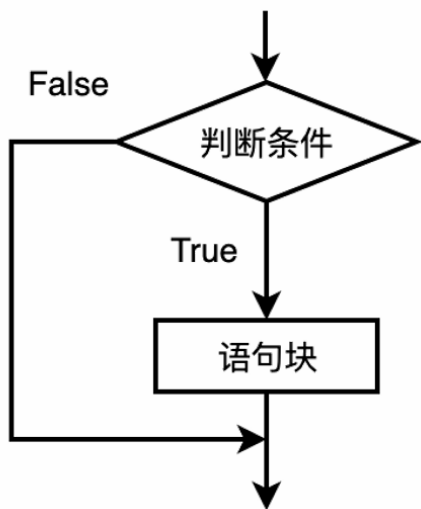
P356

Fb 粉笔

基本格式

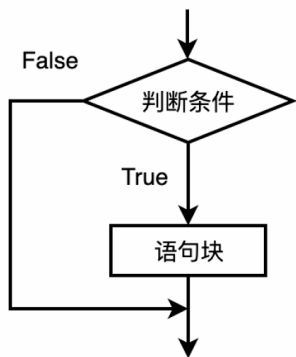
if 判断条件:
 语句块
后续语句

说明



基本格式

if 判断条件:
 语句块
后续语句



说明：

- ①使用场景：只有判断成功后有相关语句时
- ②格式要求：
 - 表达式：无括号，有冒号
 - 多条语句：同等缩进
- ③作用范围：冒号后所有缩进
 - 真，则执行所有缩进的语句块
 - 假，则执行后续语句

实例

【例1】编程实现，输入成绩，若 ≥ 60 分则输出“噢耶”和“及格啦”两行语句；无论成绩如何均会输出“考试辛苦了”语句。

说明：

实例

【例1】编程实现，输入成绩，若 ≥ 60 分则输出“噢耶”和“及格啦”两行语句；无论成绩如何均会输出“考试辛苦了”语句。

```
1 grade = eval(input("输入成绩: "))
2 if grade >= 60:
3     print("噢耶~~")
4     print("及格啦~~")
5     print("考试辛苦啦!")
```

输入成绩: 78
噢耶~~
及格啦~~
考试辛苦啦!

输入成绩: 58
及格啦~~
考试辛苦啦!

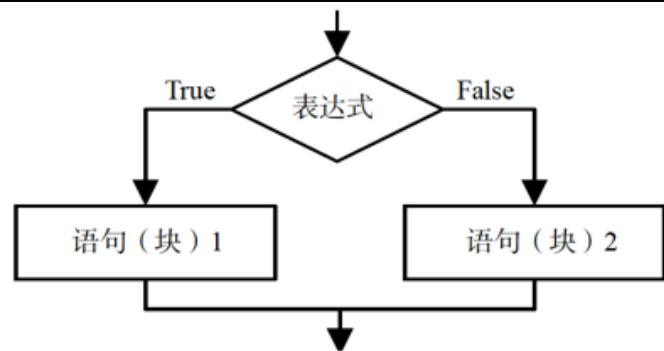
```
1 grade = eval(input("输入成绩: "))
2 if grade >= 60:
3     print("噢耶~~")
4     print("及格啦~~")
5     print("考试辛苦啦!")
```

输入成绩: 78
噢耶~~
及格啦~~
考试辛苦啦!

输入成绩: 58
考试辛苦啦!

基本格式

if 判断条件:
 语句块1
else:
 语句块2



说明：

- ①使用场景：判断成功与否均有对应的语句时
- ②格式要求：
 - 与单分支相同，else后有冒号
- ③作用范围：冒号和同等缩进
 - 真，则执行条件后的所有缩进的语句块
 - 假，则执行else后的所有缩进语句

实例

【例2】编程实现，输入成绩，若 ≥ 60 分则输出“噢耶”和“及格啦”两行语句；若 < 60 分则输出“哎”和“不及格”两行语句；无论成绩如何均会输出“考试辛苦了”语句。

分析：

基本格式

if 判断条件:

语句块1

else:

语句块2

说明：

使用场景：判断成功与否均有对应的语句时

②格式要求：

- 与单分支相同，else后有冒号

③作用范围：冒号和同等缩进

- 真，则执行条件后的所有缩进的语句块
- 假，则执行else后的所有缩进语句

实例

【例2】编程实现，输入成绩，若 ≥ 60 分则输出“噢耶”和“及格啦”两行语句；若 < 60 分则输出“哎”和“不及格”两行语句；无论成绩如何均会输出“考试辛苦了”语句。

```
1 grade = eval(input("输入成绩: "))
2 if grade >= 60:
3     print("噢耶~~")
4     print("及格啦~~")
5 else:
6     print("哎--")
7     print("不及格--")
8 print("考试辛苦啦!")
```

输入成绩: 90

噢耶~~

及格啦~~

考试辛苦啦!

输入成绩: 45

哎--

不及格--

考试辛苦啦!

| 基本格式 | 流程图 | 说明 |
|--|--|--|
| <pre>if 判断条件1: 语句块1 elif 判断条件2: 语句块2 elif 判断条件3: 语句块3 ... elif 判断条件n: 语句块n else : 语句块n+1</pre> | <pre>graph TD Start(()) --> E1{表达式1} E1 -- 真 --> S1[语句1] E1 -- 假 --> E2{表达式2} E2 -- 真 --> S2[语句2] E2 -- 假 --> E3{表达式3} E3 -- 真 --> S3[语句3] E3 -- 假 --> Dots1[...] Dots1 --> En{表达式n} En -- 真 --> Sn[语句n] En -- 假 --> Sn1[语句n+1] S1 --> Merge(()) S2 --> Merge S3 --> Merge Sn --> Merge Sn1 --> Merge Merge --> End(())</pre> | <p>①使用场景：多条件进行判断时</p> <p>②格式要求</p> <ul style="list-style-type: none">• elif：必须简写• 最后else及对应缩进可以省略 <p>③作用范围：冒号和缩进</p> <ul style="list-style-type: none">• 真，则执行缩进，然后退出• 假，则判断下一个条件• 若全假，则执行else的缩进 |

| 基本格式 | 实例 |
|--|--|
| <pre>if 判断条件1: 语句块1 elif 判断条件2: 语句块2 elif 判断条件3: 语句块3 ... elif 判断条件n: 语句块n else : 语句块n+1</pre> | <p>【例3】已知某课程的百分制成绩grade与五级制成绩的对应关系如下所示，编写程序要求输入某学生的百分制成绩，输出对应的等级。</p> <p>$90 \leq \text{grade} \leq 100$（优）；$80 \leq \text{grade} < 90$（良）；$70 \leq \text{grade} < 80$（中）； $60 \leq \text{grade} < 70$（及格）；$0 \leq \text{grade} < 60$（不及格）</p> <p>分析：</p> |

| 基本格式 | 实例 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|--|-----------------------------------|--------------------|---|-----------------------------------|---------------------|---|------------------------------|---------------------|---|------------------|----------------------|---|-------------------|----------------------|---|------------|--|---|-------------------|--|---|------------|--|---|-------------------|--|---|------------|--|----|-------------------|--|----|-------------|--|----|-------|--|----|--------------|
| <div>if 判断条件1: 语句块1</div> <div>elif 判断条件2: 语句块2</div> <div>elif 判断条件3: 语句块3</div> <div>...</div> <div>elif 判断条件n: 语句块n</div> <div>else : 语句块n+1</div> | <p>【例3】已知某课程的百分制成绩grade与五级制成绩的对应关系如下所示，编写程序要求输入某学生的百分制成绩，输出对应的等级。</p> <table><tr><td>90≤grade≤100 (优)</td><td>1</td><td>grade = eval(input("请输入百分制成绩: "))</td></tr><tr><td>80≤grade < 90 (良)</td><td>2</td><td>if grade > 100 or grade < 0:</td></tr><tr><td>70≤grade < 80 (中)</td><td>3</td><td> print("输入数据无意义")</td></tr><tr><td>60≤grade < 70 (及格)</td><td>4</td><td>elif grade >= 90:</td></tr><tr><td>0≤grade < 60 (不及格)</td><td>5</td><td> print("优")</td></tr><tr><td></td><td>6</td><td>elif grade >= 80:</td></tr><tr><td></td><td>7</td><td> print("良")</td></tr><tr><td></td><td>8</td><td>elif grade >= 70:</td></tr><tr><td></td><td>9</td><td> print("中")</td></tr><tr><td></td><td>10</td><td>elif grade >= 60:</td></tr><tr><td></td><td>11</td><td> print("及格")</td></tr><tr><td></td><td>12</td><td>else:</td></tr><tr><td></td><td>13</td><td> print("不及格")</td></tr></table> <div>请输入百分制成绩: 88.8 良</div> <div>请输入百分制成绩: 56 不及格</div> <div>请输入百分制成绩: 120 输入数据无意义</div> | | 90≤grade≤100 (优) | 1 | grade = eval(input("请输入百分制成绩: ")) | 80≤grade < 90 (良) | 2 | if grade > 100 or grade < 0: | 70≤grade < 80 (中) | 3 | print("输入数据无意义") | 60≤grade < 70 (及格) | 4 | elif grade >= 90: | 0≤grade < 60 (不及格) | 5 | print("优") | | 6 | elif grade >= 80: | | 7 | print("良") | | 8 | elif grade >= 70: | | 9 | print("中") | | 10 | elif grade >= 60: | | 11 | print("及格") | | 12 | else: | | 13 | print("不及格") |
| 90≤grade≤100 (优) | 1 | grade = eval(input("请输入百分制成绩: ")) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 80≤grade < 90 (良) | 2 | if grade > 100 or grade < 0: | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 70≤grade < 80 (中) | 3 | print("输入数据无意义") | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 60≤grade < 70 (及格) | 4 | elif grade >= 90: | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0≤grade < 60 (不及格) | 5 | print("优") | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 6 | elif grade >= 80: | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 7 | print("良") | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 8 | elif grade >= 70: | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 9 | print("中") | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 10 | elif grade >= 60: | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 11 | print("及格") | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 12 | else: | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 13 | print("不及格") | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

(2021下·初中) 某手机计步程序为了鼓励用户持续使用，会根据一段时间的统计步数给用户兑换奖品。奖励规则见下表。请使用 C 语言或者 Python 语言编写程序，根据输入步数输出相应奖品等级。

| 统计步数 | 奖品等级 |
|-------------------|------|
| 450000以上（含） | 一等奖 |
| 300000（含）— 450000 | 二等奖 |
| 150000（含）— 300000 | 三等奖 |
| 100000（含）— 150000 | 优秀奖 |
| 100000以下 | 无奖励 |

【参考答案】

```
1 bs = eval(input("请输入步数: "))
2 if bs < 0:
3     print("输入的数值是负数, 请重新输入")
4 elif 0 <= bs < 100000:
5     print("无奖励")
6 elif 100000 <= bs < 150000:
7     print("优秀奖")
8 elif 150000 <= bs < 300000:
9     print("三等奖")
10 elif 300000 <= bs < 450000:
11     print("二等奖")
12 else:
13     print("一等奖")
```

请输入步数: 5000000
一等奖

请输入步数: 150000
三等奖

请输入步数: -100000
输入的数值是负数, 请重新输入



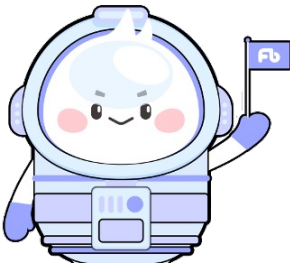


有疑问没？等你吖

下节内容

| | |
|--------------------|-----|
| 第三节 循环结构程序设计 | 361 |
| 第四节 序 列 | 365 |

P360 ~ P370



岸上等你

THE TEST

光芒万丈
不负理想

粉笔
教师



机读卡

姓名:

考号

