

## 例 1.1 语法示意代码

```
"""
author: ZHANG
data:2022 年 1 月
"""

#这一行是注释，程序功能：若 a 是最大的则输出 a 的值
a=\
10 #a=10 写在两行上，使用续行符斜杠“\”，不建议使用
b=50;c=30 #在一行上写多条语句，使用分号“;”分开，不建议使用
if a<b: #表示层次结构的顶层，用冒号表示开始
    pass#使用 pass 语句表示什么也不做，是空语句。不写会出错
else:
    if a>=c:
        print(a) #不同的缩进代表不同的层次
```

## 例 1.2 输入函数的使用

a=input('请输入第一个加数：')#输入当成字符串处理	运行结果：
print(type(a))	请输入第一个加数： 200
print(a)	<class 'str'>
b=input('请输入第二个加数：')	200
print(a+b)#字符串的加，是连接作用	请输入第二个加数： 300
a=int(a)#类型转换为浮点型好进行加法运算	200300
b=int(b)	500
print(a+b)	
x=100	
print(type(x))	

## 例 1.3 格式化输出

1. a=30
2. b=20
3. print('hello world:',a,'+',b,'=',a+b)
4. # 连续输出多个不同类型的值，非格式化输出的时候会用。像这种格式化输出一般不用
5. print('hello world:%d+%d=%d'%(a,b,a+b))
6. # 较早的输出格式，不建议用
7. print('hello world:{0}+{1}={2}'.format(a,b,a+b ))
8. # 连续输出多个不同类型的值，使用字符串的格式化 format 操作
9. print(f'hello world:{a}+{b}={a+b}')
10. # 连续输出多个不同类型的值。新格式规范，使用 f 引导字符串，{ } 引用变量

输出结果如下：

```
hello world: 30 + 20 = 50
hello world:30+20=50
```

hello world:30+20=50

hello world:30+20=50

## 例 1.4 更复杂一点的格式控制输出示例

---

```
1 a=3.1415926
2 b='重庆理工大学'
3 print("@%f# @%.2f#, @%.2f#, @%.2f#, @%.10s#"%(a,a,a,b))
4 #仔细观察上面输出语句中格式控制的作用。
5 #为了看清每个输出的开始和结束，特意使用@和#，表示每个输出的起止。
6 print("%.2f"%(a),end=",")
7 print("{0:9.2f}".format(a),end=",")
8 print(f"{a:9.2f}")
9 #仔细观察以上三条语句，体会哪种更容易理解和使用。
10 print(f"假设π的值为{a:.2f},那么 2π的值为{2*a:.2f}。")
```

---

输出结果为：

```
@3.141593# @      3.14#, @3.14#, @      3.14#, @      重庆理工大学#
      3.14,      3.14,      3.14
假设π的值为 3.14,那么 2π的值为 6.28。
```