## CMD命令的基本控制

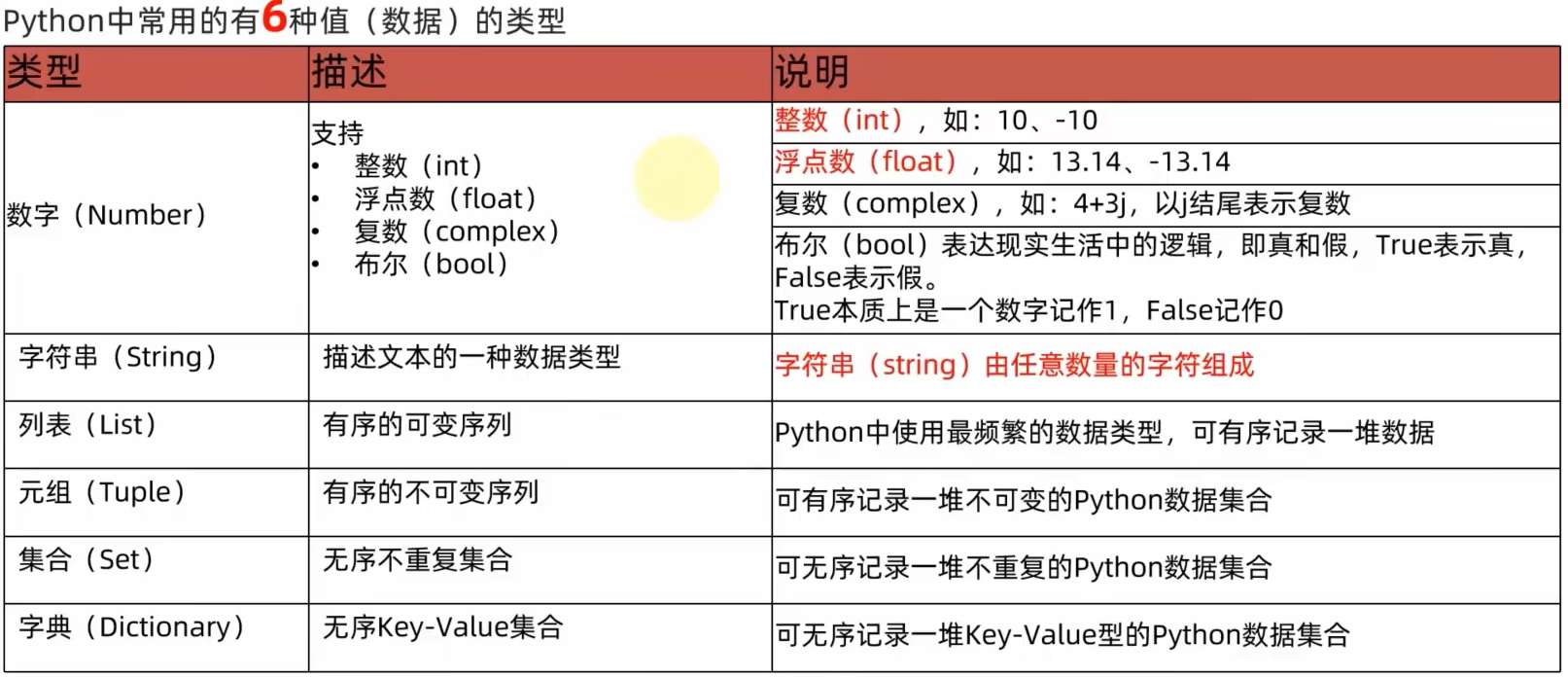
1、进入python解释器：python

2、退出python解释器：exit()

3、用python解释器运行 .py 文件：python 文件路径 //在控制台直接运行，不需要进入python解释器

## Python基本语法

### 字面量（字面量类型）



1、整数：666，-88

2、浮点数：3.14159，-5.21

3、字符串："我的世界" //必须用英文双引号""括起来

### 注释

1、单行注释：# 注释内容 //注释符#后面要加一个空格（标准形式）

Eg：# 我的世界

2、多行注释："""注释内容"""

Eg："""

网易

我的世界

"""

### 变量（创建变量）

创建变量：变量名称 = 变量值

### 数据类型 / 类型转换

查看类型：type(字面量/变量) //返回值：<class ‘数据类型'>

Eg：print(type("123")) //<class ‘str'>

转为字符串：str(x) //将x转换为字符串

转为整数：int(x) //将x转换为整数

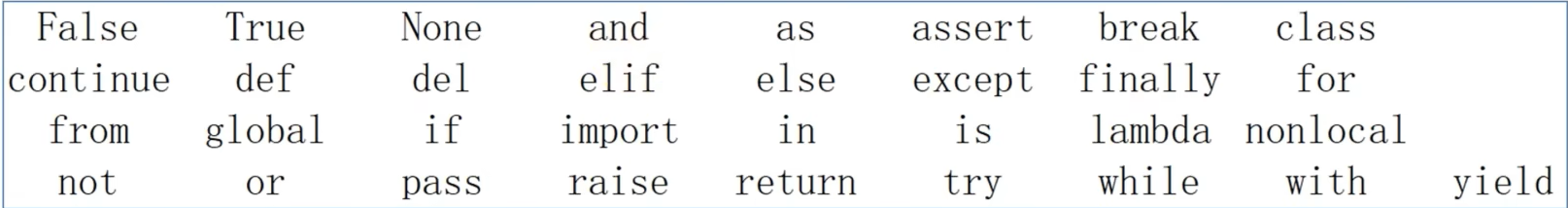
转为浮点数：float(x) //将x转换为浮点数

### 标识符（命名规则）

标识符：用户编写代码时，对变量、类、方法等编写的名字，叫做标识符

标识符命名：英文、中文、数字、\_ //大小写敏感

不可用关键字：



### 运算符







### 字符串的扩展（字符串格式化）

#### 字符串 的三种定义方式

单引号定义字符串：name = '字符串'

双引号定义字符串：name = "字符串"

三引号定义字符串：name = '''字符串''' //和多行注释一样，当没有变量赋值时就是多行注释

注意：想要定义的字符串本身，是包含:单引号、双引号时

1、单引号定义法，可以内含双引号

2、双引号定义法，可以内含单引号

3、可以使用转义字符(\)来将引号解除效用，变成普通字符串

#### 字符串的拼接

字符串拼接："字符串" + "字符串"

Eg："我的" + "世界" //"我的世界"

#### 字符串格式化

格式符号：%s（转换为字符串占位）、%d（转换为整数占位）、%f（转换为浮点数占位）

字符串占位："%s" % "字符串" //将%s占的位置替换为字符串

Eg：print("我的%s" % "世界") //输出：我的世界

Eg：print("%s%s" % ("我的","世界")) //输出：我的世界

整数占位："%d" % 数字 //将%d占的位置替换为数字

Eg：print("我的%d" % 12) //输出：我的12

Eg：print("%d%d" % ("我的",12)) //输出：我的12

#### 数字精度控制

控制数字宽度：m //整数和小数位数和，小于m时前面用空格补充，大于m时无效

控制数字精度：n //小数点后面保留n位小数，进行四舍五入（常用）

精度控制：%m.nf //替换占位时进行控制

Eg：print("%5d" % 12) //输出 [空格][空格][空格]12

Eg：print("%.2f" % 3.145) //输出 3.15

#### 快速字符串格式化

快速格式化：f"{数据}{数据}{数据}"

Eg：print(f"我的{"世界"}") //我的世界

Eg：print(f"我的{123}") //我的123

#### 表达式格式化

表达式格式化："%d" % (表达式)

Eg：print("%d" % (2 \* 5)) //10

表达式快速格式化：f"{表达式}"

Eg：print(f"{2 \* 6}") //12

### 输入与输出语句

输出：print(输出内容) //输出多个内容时print(内容1,内容2,内容3……)

输入：input(提示内容) //当代码运行到input时暂停运行，等待用户自行输入值，输入值都是字符串

## Python判断语句

#### 