

Python Key Points

基本语法

Python中严格区分大小写
Python中的每一行是一条语句，每条语句以换行结束
Python中每一行语句不要过长（规范中建议每行不要超过80个字符）："rulers":[80],
一条语句可以分多行编写，多行编写时语句后面以\结尾
Python是缩进严格的语言，所以在Python中不要随便写缩进
Python中使用#来表示注释，一般习惯上#后会跟着一个空格

数据类型

数值	整数	分成整数、浮点数（小数）、复数
		所有的整数都是int类型
		整数的大小没有限制，可以是一个无限大的整数
	浮点数	如果数字的长度过大，可以使用下划线作为分隔符： c=123_456_789
		所有小数都是float类型
	字符串	单引号和双引号都可以，但不要混用
		三重引号来表示一个长字符串
		转义字符\，使用\uxxxx打印特殊unicode字符
		字符串之间可以进行加法运算
		占位符%s：在字符串中表示任意字符
		占位符%f：表示浮点数，使用%.2f表示小数点后面保留两位并四舍五入
布尔值和空值	格式化字符串	占位符%d：表示整数
		在字符串前添加一个f来创建一个格式化字符串，可以直接嵌入变量，如c = f'hello {a} {b}'
	字符串复制	
	类型检查	
	对象	对象就是内存中专门用来存储指定数据的一块区域，实际上就是一个容器
		id用来标识对象的唯一性，可以通过id()函数查看对象的ID，是由解析器生成的，就是对象的内存地址，对象一旦创建，id永远不会改变
	对象的结构	每个对象都要保存三种数据
		type用来标识当前对象所属的类型，类型决定了对象有哪些功能，通过type()函数查看对象的类型，Python是一门强类型语言，也不会改变
变量和对象	value（值）即对象中存储的具体的数据，对于有些对象，值是可以改变的。	可变对象
		不可变对象，目前所学的对象都是不可变对象
	对象并没有直接存储到变量中，在Python中变量更像是给对象起了一个别名	
	变量中存储的不是对象的值，而是对象的id（内存地址）	
	变量中保存的对象，只有在为变量重新赋值时才会改变	
	变量和变量之间是相互独立的，修改一个变量不会影响另一个变量	
类型转换	将一个类型的对象转换为其他对象	
	类型转换不是改变对象本身的类型，而是根据当前对象的值创建一个新对象	
	四个函数	布尔值：True -> 1, False -> 0
		浮点数：直接取整，省略小数点后的内容，注意不四舍五人
	int()：可以用来将其他的对象转换为整型，注意转换时不会对原来的变量产生影响，它是对象转换为指定的类型并将其作为返回值返回，如果希望修改原来的变量，则需要对变量进行重新赋值	字符串：合法的整数字符串，直接转换为对应的整数数字；如果不是合法的整数字符串，则报错 ValueError: invalid literal for int() with base xx: 'xy.z'
		对于其他的不可转换为整型的对象，如None，直接异常 ValueError
	float(), 和int()基本一致，不同的是它会将对象转换为浮点数	布尔值：True -> 1.0, False -> 0.0
		整数：变成x.0
运算符（操作符）	算术运算符：+ - * / // ** %	字符串：合法的小数字符串，直接转换为对应的小数数字；如果不是合法的小数字符串，则报错 ValueError: invalid literal for int() with base xx: 'xy.'
		对于其他的不可转换为浮点型的对象，如None，直接异常 ValueError
	赋值运算符：= += -= *= /= //= **= %=	
	关系运算符（比较运算符）	
	逻辑运算符	
	条件运算符（三元运算符）	
字符串	str()	True -> 'True', False -> 'False'
	bool()	所有表示空性的会转换为False，如0, 0.0, "", None，其他都是True
	赋值运算符：= += -= *= /= //= **= %=	
	关系运算符（比较运算符）	
	逻辑运算符	
条件运算符（三元运算符）	算术运算符：+ - * / // ** %	
	赋值运算符：= += -= *= /= //= **= %=	
	关系运算符（比较运算符）	
	逻辑运算符	
	条件运算符（三元运算符）	