

人工智能课程

Python



目录 |

CONTENTS

01. 强制类型转换

02. 条件语句

03. 循环语句



数据类型及类型转换

type()函数：用来查看存储数据的变量的变量类型，换句话说就是查看数据类型的。

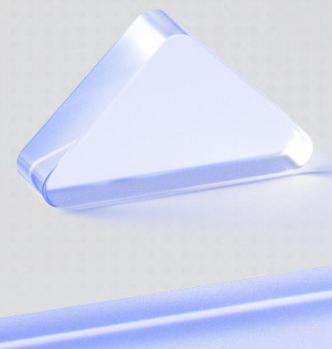
```
<class 'int'>
<class 'float'>
<class 'bool'>
<class 'str'>
<class 'list'>
<class 'tuple'>
<class 'set'>
<class 'dict'>
```



数据类型转换

数据类型转换指一个数据从当前的数据类型转换为另一种数据类型，分为显式类型转换和隐式类型转换两种。

华清远见开源实验室
yyzlab.com.cn





数据类型转换

隐式类型转换是指解释器自动将一种数据类型的值转换为另一种数据类型的过程，程序员无需操作。

隐式类型转换的常用场景：

- 算数运算中的类型转换

需要注意的是，虽然Python允许一些隐式类型转换，但并不意味着所有类型都可以随意转换。因此为了确保程序的稳定性，应尽量避免不必要的隐式类型转换，或者使用显式类型转换。



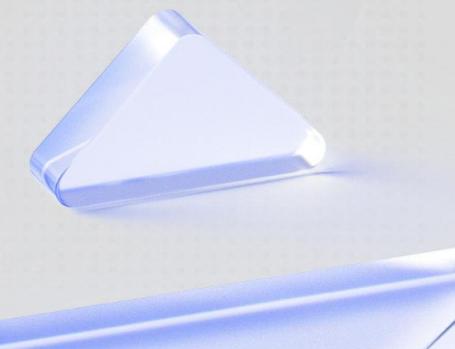
数据类型转换

显式类型转换：也叫做强制类型转换，由程序员明确地将某种数据类型的数据转化为其他数据类型。

显示类型转换的使用场景：

- 代码逻辑中所有需要数据类型转换的地方都可以使用显式类型转换**

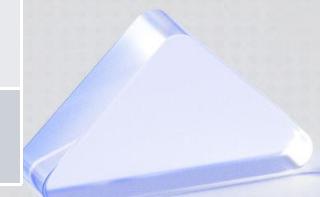
需要注意的是，即使是强制类型转换，也不是所有对象都可以安全的转换为任意其他类型，转换的过程中可能会报错。





比较常用的强制类型转换函数

函数	描述
int(x)	将x转化为整型
float(x)	将x转化为浮点型
complex(x,y)	将x、y转化为一个复数，x为实部，y为虚部
str(x)	将x转化为字符串
tuple(s)	将序列s转化为一个元组
list(s)	将序列s转化为一个列表
hex(x)	将整数x转化为一个十六进制字符串



目录 |

CONTENTS

01. 强制类型转换

02. 条件语句

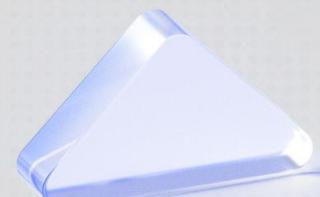
03. 循环语句



条件语句

条件语句，也叫做判断语句，是程序控制流程的基础结构之一，用于根据特定条件决定执行不同的代码块。简单来说就是如果满足某些条件，就去做某种特定的事情，不满足就不做。

条件语句使用if关键字作判断，根据不同的情况搭配不同的关键字来进行判断，共有三种判断格式。





条件语句

if格式

if 表达式:

代码块

执行逻辑:

先执行表达式，如果表达式为真，则执行代码块，否则不执行代码块



条件语句

if-else格式

if 表达式:

代码块1

else:

代码块2

执行逻辑:

先执行表达式，如果表达式为真，则执行代码块1，否则执行代码块2。



条件语句

if-elif-else格式

if 表达式1:

代码块1

elif 表达式2:

代码块2

elif 表达式n:

代码块n

else:

其它分支。

执行逻辑:

先执行表达式1，如果表达式1为真则执行代码块1，否则继续执行表达式2；
如果表达式2为真则执行代码块2，否则继续向下执行；
如果所有的表达式都为假，则执行else分支。



条件语句

嵌套的条件语句：

if 表达式1:

如果满足表达式1，执行此代码块

elif 表达式2:

if 表达式2-1:

如果满足表达式2，且满足表达式2-1，执行此代码块

else:

如果满足表达式2，但不满足表达式2-1，执行此代码块

else:

如果上面的表达式都不满足，执行此代码块



条件语句

注意事项：

- **else关键字不能独立出现，前面必须要有if与之对应**
- **关于if...else语句的表达式，一般情况下是由比较运算符和逻辑运算符组成的**
- **当有多个if语句时，每个if语句之间没有任何关系，是独立的**

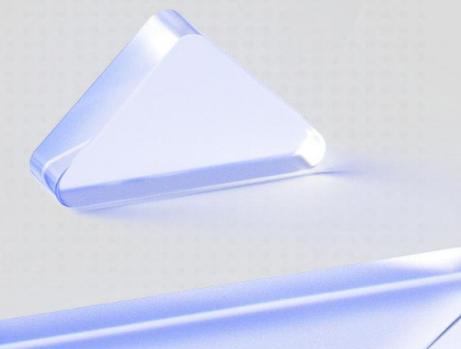


条件语句

案例：

华清远见
HQYJ.COM

华清远见元宇宙实验中心
yyzlab.com.cn





条件语句

在Python3.10版本以后，Python解释器推出了match...case语句，其作用也是用来做条件判断，可以用于更清晰的处理复杂的条件分支，取代了较为繁琐的多个if-elif-else语句，其语法如下：

match subject:

case condition1:

代码块1

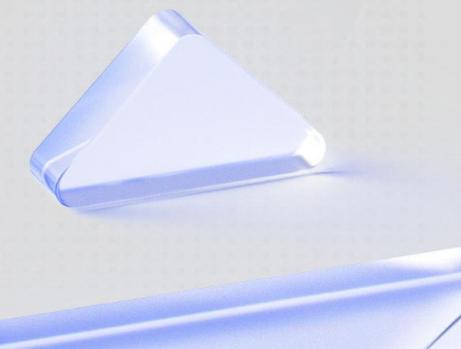
case condition2:

代码块2

case _:

代码块3

华清远见元宇宙实验室
yyzlab.com.cn



目录 |

CONTENTS

01. 强制类型转换

02. 条件语句

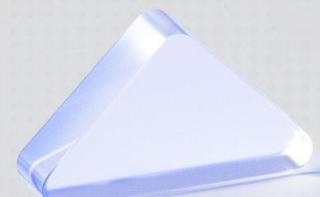
03. 循环语句



循环语句

在Python中，循环语句是程序重复执行一段代码直到满足特定条件为止的关键结构，共有2种格式，分别使用for和while两个关键字来完成。

华清远见实验室
yyzlab.com.cn





循环语句

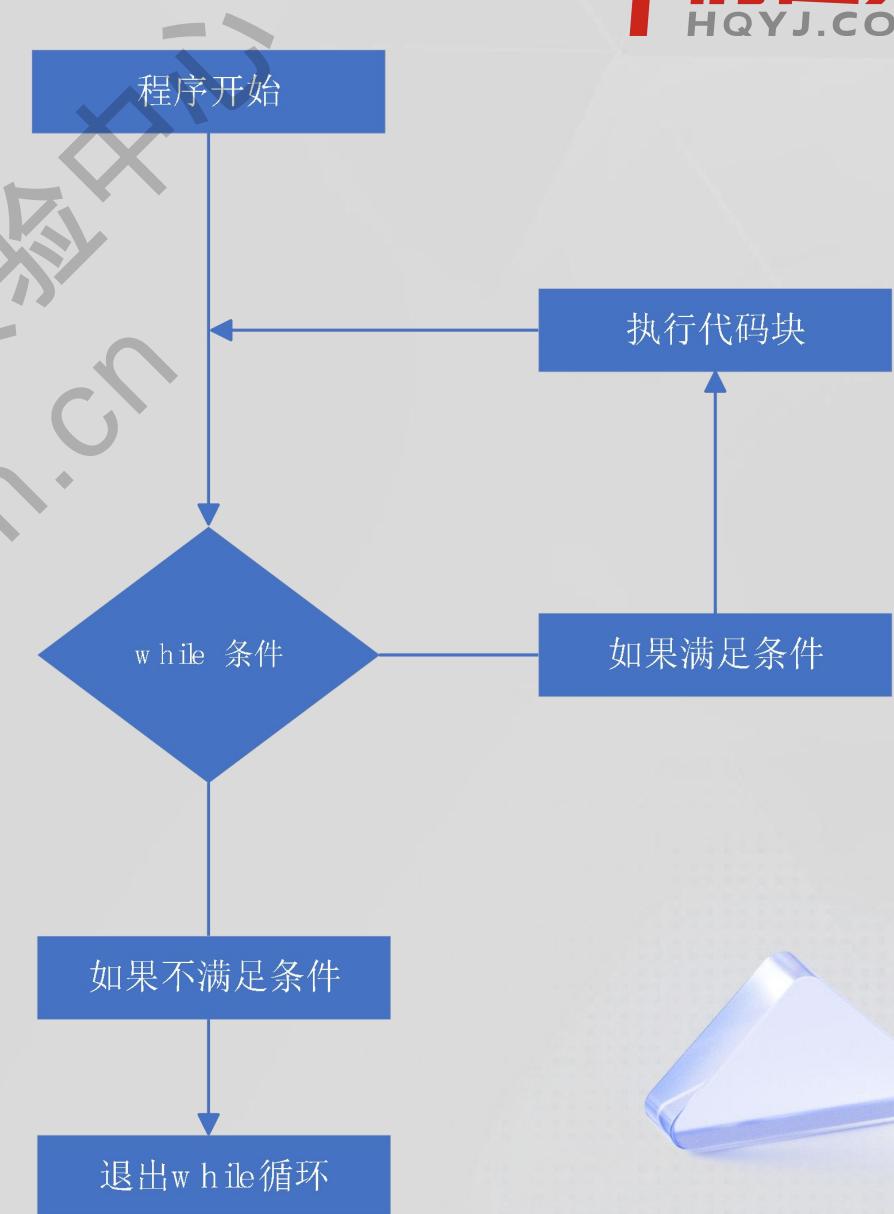
while循环语法

while 条件:

代码块

如果条件的返回结果为True，则进入while循环执行代码块，执行完毕后返回条件处再次判断条件的返回结果，直到条件的返回结果为False，退出循环。其过程如右图所示。

注意：要在while循环里设置好循环终止条件，否则就成了死循环。





循环语句

while 嵌套

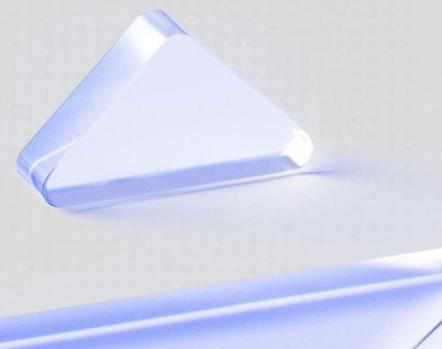
while 条件1:

while 条件2:

代码块

嵌套的while循环通常用于处理多个条件或执行多个操作，直到满足某个退出条件。

华清远见元宇宙实验中心
hqyjlab.com.cn





循环语句

for循环

for 临时变量 in 可遍历对象:

代码块

for循环可以循环遍历任何序列，比如字符串、列表等可遍历对象，只要可遍历对象没有遍历完，那么代码块就会一直执行，直到可遍历对象遍历完毕。



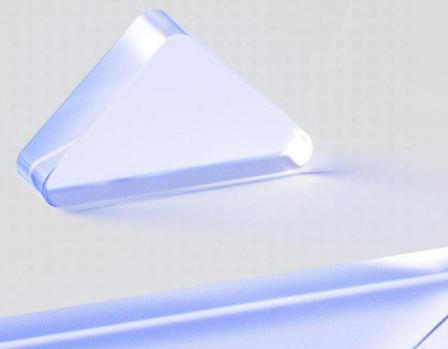
循环语句

临时变量的作用域：

从语法的角度来说，在循环结束后，是可以访问得到临时变量的值的。

但是编程规范不推荐在循环结束后直接访问临时变量，直接访问程序也可以运行，但是不建议。

如果需要访问临时变量，可以在循环之前先定义一下，再访问。



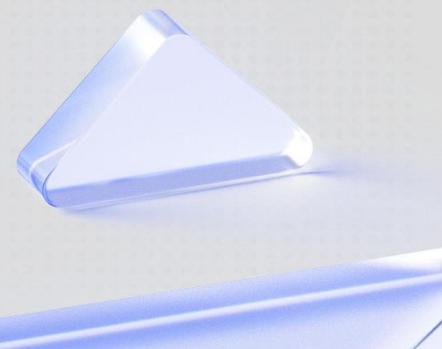


循环语句

range()函数：返回一个整数序列的对象，其语法如下：

range(start, stop, step)

计数从start开始，默认为0且为0时省略不写；计数到end结束，但是不包括end；步长默认为1且为1时省略不写。

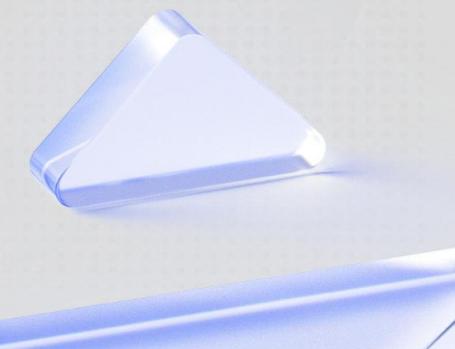




循环语句

len()函数：用于返回一个对象的长度或元素个数。这个对象可以是字符串、列表、元组、字典等数据类型。

可搭配range()函数配合for循环完成对可遍历对象中的每个元素的运算。





for 嵌套

使用for嵌套 完成九九乘法表

```
1 * 1 = 1
1 * 2 = 2 2 * 2 = 4
1 * 3 = 3 2 * 3 = 6 3 * 3 = 9
1 * 4 = 4 2 * 4 = 8 3 * 4 = 12 4 * 4 = 16
1 * 5 = 5 2 * 5 = 10 3 * 5 = 15 4 * 5 = 20 5 * 5 = 25
1 * 6 = 6 2 * 6 = 12 3 * 6 = 18 4 * 6 = 24 5 * 6 = 30 6 * 6 = 36
1 * 7 = 7 2 * 7 = 14 3 * 7 = 21 4 * 7 = 28 5 * 7 = 35 6 * 7 = 42 7 * 7 = 49
1 * 8 = 8 2 * 8 = 16 3 * 8 = 24 4 * 8 = 32 5 * 8 = 40 6 * 8 = 48 7 * 8 = 56 8 * 8 = 64
1 * 9 = 9 2 * 9 = 18 3 * 9 = 27 4 * 9 = 36 5 * 9 = 45 6 * 9 = 54 7 * 9 = 63 8 * 9 = 72 9 * 9 = 81
```



循环控制关键字：

- **break**: 在代码块执行过程中终止循环，并且跳出本层循环。
- **continue**: 在代码块执行过程中终止本次循环，执行下一次循环。
- **pass**: 空语句，相当于一个占位符，它的作用是在语法上需要一个语句，但程序不需要任何操作时使用。例如，在定义一个函数或循环时，你可能需要一些默认的参数或条件，但还没有决定具体要做什么。在这些情况下，可以使用**pass**语句来满足语法要求。



循环语句

while循环的使用场景：

while循环适用于需要根据条件来判断是否继续循环的情况。

例如，当需要不断接收用户输入直到用户输入特定的值时，可以使用while循环，或者程序的运行结果满足某条件时需要结束循环，也可以使用while循环。



循环语句

for循环的使用场景：

for循环适用于遍历序列（如列表、元组、字符串等）中的元素。当需要对一个序列中的每个元素执行相同的操作时，可以通过**range()**函数与**len()**函数来配合**for**循环来完成。

科技赋能·智引未来



技术领先
品质保障



超多干货
实时更新



海量视频
贴身学习