

目标

- 数据类型转换的必要性
- 数据类型转换常用方法

一. 转换数据类型的作用

问：input()接收用户输入的数据都是字符串类型，如果用户输入1，想得到整型该如何操作？

答：转换数据类型即可，即将字符串类型转换成整型。

二. 转换数据类型的函数

函数	说明
<code>int(x [,base])</code>	将x转换为一个整数
<code>float(x)</code>	将x转换为一个浮点数
<code>complex(real [,imag])</code>	创建一个复数，real为实部，imag为虚部
<code>str(x)</code>	将对象 x 转换为字符串
<code>repr(x)</code>	将对象 x 转换为表达式字符串
<code>eval(str)</code>	用来计算在字符串中的有效Python表达式,并返回一个对象
<code>tuple(s)</code>	将序列 s 转换为一个元组
<code>list(s)</code>	将序列 s 转换为一个列表
<code>chr(x)</code>	将一个整数转换为一个Unicode字符
<code>ord(x)</code>	将一个字符转换为它的ASCII整数值
<code>hex(x)</code>	将一个整数转换为一个十六进制字符串
<code>oct(x)</code>	将一个整数转换为一个八进制字符串
<code>bin(x)</code>	将一个整数转换为一个二进制字符串

三. 快速体验

需求：input接收用户输入，用户输入“1”，将这个数据1转换成整型。

```
1 # 1. 接收用户输入
2 num = input('请输入您的幸运数字: ')
3
4 # 2. 打印结果
5 print(f"您的幸运数字是{num}")
6
7
8 # 3. 检测接收到的用户输入的数据类型 -- str类型
9 print(type(num))
10
11 # 4. 转换数据类型为整型 -- int类型
12 print(type(int(num)))
```

四. 实验

```
1 # 1. float() -- 转换成浮点型
2 num1 = 1
3 print(float(num1))
4 print(type(float(num1)))
5
6 # 2. str() -- 转换成字符串类型
7 num2 = 10
8 print(type(str(num2)))
9
10 # 3. tuple() -- 将一个序列转换成元组
11 list1 = [10, 20, 30]
12 print(tuple(list1))
13 print(type(tuple(list1)))
14
15
16 # 4. list() -- 将一个序列转换成列表
17 t1 = (100, 200, 300)
18 print(list(t1))
19 print(type(list(t1)))
20
21 # 5. eval() -- 将字符串中的数据转换成Python表达式原本类型
22 str1 = '10'
23 str2 = '[1, 2, 3]'
24 str3 = '(1000, 2000, 3000)'
25 print(type(eval(str1)))
26 print(type(eval(str2)))
27 print(type(eval(str3)))
```

总结

- 转换数据类型常用的函数

- int()
- float()
- str()
- list()
- tuple()
- eval()