第二讲-数据类型转换&运算符

目标

- 输入功能的语法
- 输入input的特点
- 数据类型转换的必要性
- 数据类型转换常用方法
- 掌握常用运算符的作用
- 总结

1.输入

学习目标:

- 输入功能的语法
- 输入input的特点

1.1输入

人类有两个基本的行为用于和他人进行沟通交流：1.说话(输出) 2.写字(输入)

在Python中，程序接收用户输入的数据的功能即是输入。

1.2输入的语法

```
input("提示信息")
```

1.3输入的特点

- 当程序执行到`input`，等待用户输入，输入完成之后才继续向下执行。

- 在Python中，`input`接收用户输入后，一般存储到变量，方便使用。
- 在Python中，`input`会把接收到的任意用户输入的数据都当做字符串处理。

```
password = input('请输入您的密码：')

print(f'您输入的密码是{password}')
# <class 'str'>
print(type(password))
```

控制台输出的结果：

1.4输入总结

- 输入功能
    - input('提示文字')
- 输入的特点
    - 一般将input接收的数据存储到变量
    - input接收的任何数据默认都是字符串数据类型

2.转换数据类型学习目标

- 数据类型转换的必要性
- 数据类型转换常用方法

2.1.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

2.2.转换数据类型的函数

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

2.3 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. 接收用户输入
num = input('请输入您的幸运数字：')

# 2. 打印结果
print(f"您的幸运数字是{num}")


# 3. 检测接收到的用户输入的数据类型 -- str类型
print(type(num))

# 4. 转换数据类型为整型 -- int类型
print(type(int(num)))
```

2.4实验

```
# 1. float() -- 转换成浮点型
num1 = 1
print(float(num1))
print(type(float(num1)))

# 2. str() -- 转换成字符串类型
num2 = 10
print(type(str(num2)))

# 3. tuple() -- 将一个序列转换成元组
list1 = [10, 20, 30]
```

```
print(tuple(list1))
print(type(tuple(list1)))


# 4. list() -- 将一个序列转换成列表
t1 = (100, 200, 300)
print(list(t1))
print(type(list(t1)))

# 5. eval() -- 将字符串中的数据转换成Python表达式原本类型
str1 = '10'
str2 = '[1, 2, 3]'
str3 = '(1000, 2000, 3000)'
print(type(eval(str1)))
print(type(eval(str2)))
print(type(eval(str3)))
```

2.5转换数据类型总结

转换数据类型常用的函数

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

3.运算符学习目标

掌握常用运算符的作用

3.1运算符的分类

- 算数运算符
- 赋值运算符
- 复合赋值运算符
- 比较运算符
- 逻辑运算符

1.算数运算符

运算符	描述	实例
+	加	1+1输出结果为2
-	减	1-1 输出结果为 0
*	乘	2 * 2 输出结果为 4
/	除	10 / 2 输出结果为 5
//	整除	9 // 4 输出结果为2
%	取余	9 % 4 输出结果为 1
**	指数	2 ** 4 输出结果为 16，即 2 * 2 * 2 * 2
（）	小括号	小括号用来提高运算优先级，即 (1 + 2) * 3 输出结果为 9

> 注意：

- 混合运算优先级顺序：`()` 高于 `**` 高于 `*` `/` `//` `%` 高于 `+` `-`

2.赋值运算符

运算符	描述	实例
=	赋值	将=右侧的结果赋值给等号左侧的变量

- 单个变量赋值

```
num=1
print(num)
```

- 多个变量赋值

```
num,float1,strl =10,0.5,"hello word"
print(num)
print(float1)
print(str1)
```

结果：

多个变量赋相同的值

```
a=b=6
print(a)
print（b）
```

结果：

3.复合赋值运算符

运算符	描述	实例
+=	加法赋值运算符	c += a 等价于 c = c + a
-=	减法赋值运算符	c -= a 等价于 c = c- a
*=	乘法赋值运算符	c *= a 等价于 c = c * a
/=	除法赋值运算符	c /= a 等价于 c = c / a
//=	整除赋值运算符	c //= a 等价于 c = c // a
%=	取余赋值运算符	c %= a 等价于 c = c % a
**=	幂赋值运算符	c ** = a 等价于 c = c ** a

```
a = 100
a += 1
# 输出101  a = a + 1,最终a = 100 + 1
print(a)

b = 2
b *= 3
# 输出6  b = b * 3,最终b = 2 * 3
print(b)

c = 10
c += 1 + 2
# 输出13, 先算运算符右侧1 + 2 = 3,  c += 3 , 推导出c = 10 + 3
print(c)
```

4. 比较运算符

比较运算符也叫关系运算符， 通常用来判断。

运算符	描述	实例
==	判断相等。如果两个操作数的结果相等，则条件结果为真(True)，否则条件结果为假(False)	如a=3,b=3，则（a == b) 为 True
!=	不等于 。如果两个操作数的结果不相等，则条件为真(True)，否则条件结果为假(False)	如a=3,b=3，则（a == b) 为 True如a=1,b=3，则(a != b) 为 True
>	运算符左侧操作数结果是否大于右侧操作数结果，如果大于，则条件为真，否则为假	如a=7,b=3，则(a > b) 为 True
<	运算符左侧操作数结果是否小于右侧操作数结果，如果小于，则条件为真，否则为假	如a=7,b=3，则(a < b) 为 False
>=	运算符左侧操作数结果是否大于等于右侧操作数结果，如果大于，则条件为真，否则为假	如a=7,b=3，则(a < b) 为 False如a=3,b=3，则(a >= b) 为 True
<=	运算符左侧操作数结果是否小于等于右侧操作数结果，如果小于，则条件为真，否则为假	如a=3,b=3，则(a <= b) 为 True

```
a = 7
b = 5
print(a == b)  # False
print(a != b)  # True
print(a < b)   # False
print(a > b)   # True
print(a <= b)  # False
print(a >= b)  # True
```

5. 逻辑运算符

运算符	逻辑表达式	描述	实例
and	x and y	布尔"与"：如果 x 为 False，x and y 返回 False，否则它返回 y 的值。	True and False， 返回 False。
or	x or y	布尔"或"：如果 x 是 True，它返回 True，否则它返回 y 的值。	False or True， 返回 True。
not	not x	布尔"非"：如果 x 为 True，返回 False 。如果 x 为 False，它返回 True。	not True 返回 False, not False 返回 True

```
a = 1
b = 2
c = 3
print((a < b) and (b < c))  # True
print((a > b) and (b < c))  # False
print((a > b) or (b < c))   # True
print(not (a > b))          # True
```

5.1拓展

数字之间的逻辑运算

```
a = 0
b = 1
c = 2

# and运算符，只要有一个值为0，则结果为0，否则结果为最后一个非0数字
print(a and b)  # 0
print(b and a)  # 0
print(a and c)  # 0
print(c and a)  # 0
print(b and c)  # 2
print(c and b)  # 1

# or运算符，只有所有值为0结果才为0，否则结果为第一个非0数字
print(a or b)  # 1
print(a or c)  # 2
print(b or c)  # 1
```

6.运算符总结

- 算数运算的优先级
  - 混合运算优先级顺序：`()` 高于 `**` 高于 `*` `/` `//` `%` 高于 `+` `-`
- 赋值运算符
  - =
- 复合赋值运算符
  - +=
  - -=
  - 优先级
    1. 先算复合赋值运算符右侧的表达式
    2. 再算复合赋值运算的算数运算
    3. 最后算赋值运算
- 比较运算符
  - 判断相等： ==
  - 大于等于： >=
  - 小于等于：<=
  - 不等于： !=
- 逻辑运算符
  - 与： and
  - 或：or
  - 非：not