

Приведение типов в Python

Приведение типов — это процесс преобразования значения из одного типа данных в другой. В Python часто нужно переводить числа в строки, строки в числа и т.д.

Основные функции приведения типов

1. `int()` — преобразует значение в целое число (integer).
 2. `float()` — преобразует значение в число с плавающей точкой (вещественное).
 3. `str()` — преобразует значение в строку.
 4. `bool()` — преобразует значение в логический тип (`True` или `False`).
 5. `list()`, `tuple()`, `set()` — преобразуют коллекции в список, кортеж или множество.
-

Примеры кода

1. Преобразование строки в число

Если строка содержит число, можно преобразовать её в `int` или `float`:

```
num_str = "42"
num = int(num_str) # Преобразуем строку "42" в число 42
print(num + 8) # 50

decimal_str = "3.14"
decimal = float(decimal_str) # Преобразуем строку "3.14" в вещественное
число 3.14
print(decimal + 1.0) # 4.14
```

2. Преобразование числа в строку

Если нужно объединить число с текстом, его надо превратить в строку:

```
age = 25
print("Мне " + str(age) + " лет.") # Преобразуем 25 в строку и объединяем с
```

3. Преобразование в логический тип

- Значения `0`, `0.0`, `None`, `""` (пустая строка) преобразуются в `False`.
- Все остальные значения — в `True`.

```
print(bool(0)) # False
print(bool(42)) # True
print(bool("")) # False
print(bool("Python")) # True
```

4. Преобразование между коллекциями

Можно преобразовать список в кортеж, множество или наоборот:

```
my_list = [1, 2, 3]
my_tuple = tuple(my_list) # Преобразуем список в кортеж
print(my_tuple) # (1, 2, 3)

my_set = set(my_list) # Преобразуем список в множество
print(my_set) # {1, 2, 3}

new_list = list(my_tuple) # Преобразуем кортеж обратно в список
print(new_list) # [1, 2, 3]
```

5. Преобразование строки в список или множество

Строку можно разбить на отдельные символы и преобразовать в список или множество:

```
word = "hello"
letters_list = list(word) # ['h', 'e', 'l', 'l', 'o']
letters_set = set(word) # {'h', 'e', 'l', 'o'}
print(letters_list)
print(letters_set)
```

Обработка ошибок при приведении типов

Если преобразование невозможно, Python выбросит ошибку:

```
num_str = "abc"
try:
    num = int(num_str) # Попытка преобразовать "abc" в число
except ValueError:
    print("Невозможно преобразовать строку в число!")
```

Пример из жизни

Сложение чисел, введённых пользователем

Ввод через `input()` всегда возвращает строку. Чтобы сложить два числа, нужно преобразовать их в `int` или `float`:

```
num1 = input("Введите первое число: ")
num2 = input("Введите второе число: ")

# Преобразуем строки в числа
sum_result = int(num1) + int(num2)

print("Сумма чисел:", sum_result)
```

Если пользователь введёт "5" и "7", программа выведет:

```
Сумма чисел: 12
```