

Типы данных, преобразования и арифметические операции

Основные типы данных в языке Python

- `int` - целочисленный тип (без ограничений, например: 0 1 -104354 34395834985 584398548)
- `float` - тип с плавающей точкой (без ограничений, например: 0.1 -2.123 1231231.234234)
- `str` - строковый тип (например: 'A', 'Hello, world', '123', "321", "B")
- `bool` - логический тип (например: `True`, `False`)

Преобразование к нужному типу данных осуществляется с помощью одноимённой функции, например:

```
a = int('123')    # в переменную a будет записано число 123
b = str(321)      # в переменную b будет записана строка '321'
c = float(1)      # в переменную c будет записано число 1.0
d = bool(1)       # в переменную d будет записано логическое значение True
d = bool(0)       # в переменную d будет записано логическое значение False
```

Арифметические операторы

1. Оператор сложения (+)

```
a = 5 + 3    # в переменную a будет записано число 8
```

2. Оператор вычитания (-)

```
b = 5 - 3    # в переменную b будет записано число 2
```

3. Оператор умножения (*)

```
c = 5 * 3    # в переменную c будет записано число 15
```

4. Оператор деления (/)

```
d = 5 / 3    # в переменную d будет записано число 1.6666666666666667
```

5. Оператор целочисленного деления (//)

```
e = 5 // 3   # в переменную e будет записано число 1
```

6. Оператор взятия остатка от деления (%)

```
f = 5 % 3    # в переменную f будет записано число 2
```

7. Оператор возведения в степень (**)

```
g = 5 ** 3   # в переменную g будет записано число 125
```

Все операторы работают корректно и не вызывают ошибок для типов `int` и `float` в любых комбинациях. Исключением является лишь деление на ноль. Например:

```
a = 0.1 + 2    # в переменную a будет записано число 2.1
b = 4**0.5      # в переменную b будет записано число 2.0
c = 2.1 % 2     # в переменную c будет записано число 0.10000000000000009
```

Однако для строк справедливы лишь следующие две операции:

1. Сложение двух строк, например:

```
a = '123' + '321' # в переменную a будет записана строка '123321'
```

2. Умножение строки на целое число, например:

```
a = 'abc'*3       # в переменную a будет записана строка 'abcabcabc'
```

Операторы сравнения

1. Оператор больше (>). Возвращает `True`, если значение слева больше значения справа, и `False` в противном случае

```
f = 5 > 3      # в переменную f будет записано значение True
f = 5 > 5      # в переменную f будет записано значение False
f = 3 > 5      # в переменную f будет записано значение False
```

2. Оператор меньше (<). Возвращает `True`, если значение слева меньше значения справа, и `False` в противном случае

```
f = 5 < 3      # в переменную f будет записано значение False
f = 5 < 5      # в переменную f будет записано значение False
f = 3 < 5      # в переменную f будет записано значение True
```

3. Оператор равенства (==). Возвращает `True`, если значение слева равно значению справа, и `False` в противном случае

```
f = 5 == 3     # в переменную f будет записано значение False
```

```
f = 5 == 5      # в переменную f будет записано значение True
f = 3 == 5      # в переменную f будет записано значение False
```

4. Оператор равенства (`>=`). Возвращает `True`, если значение слева больше либо равно значению справа, и `False` в противном случае

```
f = 5 >= 3      # в переменную f будет записано значение True
f = 5 >= 5      # в переменную f будет записано значение True
f = 3 >= 5      # в переменную f будет записано значение False
```

5. Оператор равенства (`<=`). Возвращает `True`, если значение слева меньше либо равно значению справа, и `False` в противном случае

```
f = 5 <= 3      # в переменную f будет записано значение False
f = 5 <= 5      # в переменную f будет записано значение True
f = 3 <= 5      # в переменную f будет записано значение True
```

6. Оператор равенства (`!=`). Возвращает `True`, если значение слева не равно значению справа, и `False` в противном случае

```
f = 5 != 3      # в переменную f будет записано значение True
f = 5 != 5      # в переменную f будет записано значение False
f = 3 != 5      # в переменную f будет записано значение True
```

Логические операторы

1. Оператор И (`and`). Возвращает `True`, если условные выражения по обе стороны оператора истинны, и `False` в противном случае

```
f = True and True      # в переменную f будет записано значение True
f = True and False     # в переменную f будет записано значение False
f = False and True     # в переменную f будет записано значение False
f = False and False    # в переменную f будет записано значение False
```

2. Оператор ИЛИ (`or`). Возвращает `True`, если хотя бы одно условное выражения (слева или справа от оператора) истинно, и `False` в противном случае

```
f = True or True       # в переменную f будет записано значение True
f = True or False      # в переменную f будет записано значение True
f = False or True      # в переменную f будет записано значение True
f = False or False     # в переменную f будет записано значение False
```

3. Оператор НЕ (`not`). Возвращает `True`, если выражение, стоящее справа от оператора, ложно, и `False` в противном случае

```
f = not True          # в переменную f будет записано значение False
f = not False         # в переменную f будет записано значение True
```

Приоритет операций

1. Сначала выполняются операции возведения в степень (**) справа налево
2. Затем операции умножения (*), деления (/), целочисленного деления (//) и деления с остатком (%)
3. Следом выполняются операции сложения (+) и вычитания (-)
4. Затем выполняются операторы сравнения
5. И, наконец, логические операторы