

Основы программирования в python

4 октября 2024 г.

1 Базовый синтаксис языка программирования Python

1 Базовый синтаксис языка программирования Python

- Типы данных, операторы сравнения, арифметические и логические операторы, приведение типов

Типы данных

- Числовые типы:
 - `int` — целые числа
 - `float` — числа с плавающей точкой
 - `complex` — комплексные числа
- Логический тип:
 - `bool` — булевы значения (`True`, `False`)
- Последовательности:
 - `str` — строки
 - `list` — списки
 - `tuple` — кортежи
- Множества и словари:
 - `set` — множества
 - `dict` — словари

Для определения типа данных используется команда - `Type()`

Арифметические операторы

- Сложение (+): складывает два числа.
- Вычитание (-): вычитает одно число из другого.
- Умножение (*): умножает два числа.
- Деление (/): делит одно число на другое.
- Целочисленное деление (//): возвращает целую часть от деления.
- Остаток от деления (%): возвращает остаток от деления.
- Возведение в степень (**): возводит число в степень.

Операторы сравнения

- Равенство: `==`
- Неравенство: `!=`
- Меньше: `<`
- Меньше или равно: `<=`
- Больше: `>`
- Больше или равно: `>=`

Операторы сравнения

```
a = 10  
b = 5  
print(a > b)  # True  
print(a == b) # False
```

Логические операторы

- И: and
- Или: or
- Не: not

```
x = True
y = False
print(x and y)  # False
print(x or y)   # True
print(not x)    # False
```

Приведение типов

```
a = 2
b = 4.5
c = '3'
#БЕРНО
x = a + b
#НЕ БЕРНО
x = a + c
```

Приведение типов

Преобразование в строку

```
num = 123
```

```
str_num = str(num)
```

Преобразование в целое число

```
float_num = 12.34
```

```
int_num = int(float_num)
```

Преобразование в вещественное число

```
str_float = "56.78"
```

```
float_from_str = float(str_float)
```

Ввод данных

```
user_input = input("Введите что-нибудь: ")
age = int(input("Введите ваш возраст: "))
temperature = float(input("Введите температуру: "))
```

Форматированный вывод

Базовый синтаксис:

```
"Hello, {}!".format("world")
```

```
"Hello, world!"
```

Множественные заполнители:

```
"{} is {} years old.".format("Alice", 30)
```

Именованные заполнители:

```
"{name} is {age} years old.".format(name="Alice", age=30)
```

Позиционные и именованные аргументы:

```
"{0} is from {country}".format("Alice", country="USA")
```

Указание индексов:

```
"{0} {1} {0}".format("one", "by")
```

Форматированный вывод

```
num = 56.67893456
print("{:.2f}".format(num))
56.68
print("{:,.2f}".format(10001.23554))
10,001.24
print(f"{num:10.2f}")
    56.68
# Выравнивание вправо
print("{:>10}".format("текст"))
# Выравнивание влево
print("{:<10}".format("текст"))
# Выравнивание по центру
print("{:^10}".format("текст"))
```
