

Операции с числами

8 класс. Программирование на языке **Python**

Арифметические операции с числами

* - умножение

** - возведение в степень

- - вычитание

+ - сложение

/ - деление

// - деление нацело

% - остаток от деления



Примеры арифметических операций

- `print(5 * 2)` #10 - умножение
- `print(5 ** 2)` #25 – возведение в степень
- `print(5 + 2)` #7 - сложение
- `print(5 - 2)` #3 - вычитание
- `print(5 / 2)` #2.5 - деление
- `print(5 // 2)` #2 – целочисленное деление
- `print(5 % 2)` #1 – остаток от деления



Примеры на вычисление

□ `print(3 * 4)` □ 12

□ `print(4 ** 3)` □ 64

□ `print(6 + 3)` □ 9

□ `print(3 - 9)` □ -6

□ `print(9 / 4)` □ 2.25

□ `print(11 // 3)` □ 3

□ `print(17 % 3)` □ 2



Приоритеты выполнения операций

- 1) Возведение в степень
- 2) Умножение
- 3) Деление
- 4) Целочисленное деление
- 5) Остаток от деления
- 6) Сложение
- 7) Вычитание

Если в примере только операции + или – то они выполняются по порядку

** * / // % + -



Примеры вычисления выражений

```
number = 5 * 3 ** 2 / 2 + 10
```

```
print(number)
```

□ Порядок:

□ $3 ** 2 = 9$

`type(number) = int`

□ $5 * 9 = 45$

`type(number) = int`

□ $45 / 2 = 22.5$

`type(number) = float`

□ $22.5 + 10 = 32.5$

`type(number) = float`

```
number = 5 * 3 ** 2 / 2 - 10 + 15
```

```
print(number)
```

□ Порядок:

□ $3 ** 2 = 9$

□ $5 * 9 = 45$

□ $45 / 2 = 22.5$

□ $22.5 - 10 = 12.5$

□ $12.5 + 15 = 27.5$



Приоритеты выполнения операций

Если выражение содержит скобки, то сначала выполняются действия в скобках

```
number = (5 * 3) ** (2 / 2) + 10
```

```
print(number)
```

□ Порядок:

□ $5 * 3 = 15$

□ $2 / 2 = 1$

□ $15 ** 1 = 15$

□ $15 + 10 = 25$



Примеры вычисления выражений

```
number = (3 + 4) * (5 ** 2 + 7)
```

```
print(number)
```

□ Порядок:

□ $3 + 4 = 7$

□ $5 ** 2 = 25$

□ $25 + 7 = 32$

□ $7 * 32 = 224$



Арифметические операции с присвоением

Ряд специальных операций позволяют присвоить результат операции первому операнду:

+= #Присвоение результата сложения

-= #Присвоение результата вычитания

***=** #Присвоение результата умножения

/= #Присвоение результата от деления

//= #Присвоение результата целочисленного деления

****=** #Присвоение степени числа

%= #Присвоение остатка от деления



Арифметические операции с присвоением

```
number = 10
```

```
number += 5
```

```
print(number) # 15
```

```
number -= 3
```

```
print(number) # 12
```

```
number *= 4
```

```
print(number) # 48
```



Функции преобразования

```
J: > Py > intro.py > ...
```

```
1 ch = input()
```

```
2 ch += 4
```

Возникло исключение: **TypeError** ×

can only concatenate str (not "int") to str

File "J:\Py\intro.py", line 2, in <module>

```
ch += 4
```

```
3 print(ch)
```

```
4
```

ПРОБЛЕМЫ

ВЫХОДНЫЕ ДАННЫЕ

ТЕРМИНАЛ

КОНСОЛЬ ОТЛАДКИ

```
PS J:\Py> j:; cd 'j:\Py'; & 'C:\Users\UMka\AppData\Local\Programs\Python\Python39\p  
.1159798656\pythonFiles\lib\python\debugpy\launcher' '57478' '--' 'j:\Py\intro.py'
```

```
5
```



Функции преобразования

- Для преобразования переменной типа `str` в `int` используется функция `int()`

- Код с ошибкой:

```
first_number = '3' #тип str
second_number = 5 #тип int
third_number = first_number + second_number #ошибка
print(third_number)
```

- Код с преобразованием `str` в `int`:

```
first_number = '3' #тип str
second_number = 5 #тип int
third_number = int(first_number) + second_number
print(third_number) #8
```



Функции преобразования

□ Для преобразования переменной типа `str` в `float` используется функция **`float()`**

□ Код с преобразованием `str` в `float`:

```
first_number = '2.001' #вещественные числа записываются через «.», а не через «,»
```

```
second_number = 5
```

```
third_number = float(first_number) + second_number
```

```
print(third_number) #7.001
```



```
7.0009999999999994
```

Происходит округление вещественного числа до ближайшего, хранимого в памяти



Округление вещественных чисел

- Для округления вещественных чисел можно использовать функцию **round()**
- **Формат:** `round(<число>, <количество разрядов после запятой>)`
- Пример: `round(number, 3)`
- Код:

```
first_number = '2.001'
```

```
second_number = 5
```

```
third_number = float(first_number) + second_number
```

```
print(round(third_number,4))
```

```
PS J:\Py> j:; cd 'j:\Py'; & 'C:\Users\  
.1159798656\pythonFiles\lib\python\debu  
7.001
```



Источники

- <https://metanit.com/>
- «Изучаем Python» Марк Лутц. Том I, 5-е издание. 2019
- Авторский материал



Автор

- Учитель математики и информатики МБОУ гимназия №9 г. Воронежа Уразов М.Ю.

