МФТИ

Числа и функции

Докладчик: Евграфов Михаил

Числа

Числовые типы данных

Числовые типы в Python:

- **целые числа**
- **числа с плавающей точкой**
- **~~комплексные числа~~**

Часто используемые значения

Вычисления с часто используемыми значениями

```
>>> num1, num2 = 5, 10
>>> num3 = num2 - num1
>>> num1 is num3
True
```

Вычисления с нечасто используемыми значениями

```
>>> num1, num2 = 500, 1000
>>> num3 = num2 - num1
>>> num1 is num3
False
```

int: основные сведения

- **целочисленный тип - неизменяемый тип данных**
- **допустимые значения: целые числа от -inf до +inf**
- **функция для конвертации:** `int()`

Конвертация объектов в int

```
>>> int(True)
1

>>> int(3.14)
3

>>> int("123")
123

>>> int(b"123")
123
```

Выбор системы счисления

```
>>> int("101", base=2)
5

>>> int("123", base=8)
83

>>> int("ABC", base=16)
2748
```

Целочисленные литералы

```
0b1000101          # двоичная форма записи
0o105              # восьмеричная форма записи
69                 # десятичная форма записи
0xa9               # шестнадцатеричная форма записи

# разделители
1_000_000_000
0xF1c5_910D_FF0A
```

float: основные сведения

- **float - неизменяемый тип данных**
- **допустимые значения: числа с плавающей точкой**
- **максимальное значение: зависит от платформы**
- **минимальное значение: зависит от платформы**
- **функция для конвертации:** `float()`

Конвертация объектов во float

```
>>> float(True)
1.0

>>> float(123)
123.0

>>> float("56.7")
56.7
```

Специальные значения float

```
>>> float("+inf")
inf

>>> float("-inf")
-inf

>>> float("nan")
nan
```

Литералы float

```
3.14                # 3.14
-3.14               # -3.14
00003.14            # 3.14
.314                # 0.314
3.14e0              # 3.14
314e-2              # 3.14
```

Арифметические операции

```
-5              # унарный минус
+3.14           # унарный плюс
3.14 + 5        # сложение
5 - 3.14        # вычитание
3.14 * 5        # умножение
5 / 3.14        # честное деление (truediv)
5 // 3.14       # целая часть от деления
5 % 3.14        # остаток от деления
3.14 ** 5       # возведение в степень
```

Арифметические функции

```
# обычное возведение в степень
pow(2, 4)        # == 16

# возведение в степень по модулю 10
pow(2, 4, 10)    # == 6

# вычисления целой части и остатка
divmod(5, 3)     # == (1, 2)

# вычисление модуля
abs(-3.14)       # == 3.14
```

Логические операции

```
5 == 3.14          # равенство
5 != 3.14          # неравенство
5 > 3.14           # больше
5 >= 3.14          # больше или равно
5 < 3.14           # меньше
5 <= 3.14          # меньше или равно
```

Сравнение float

```
eps = 1e-6
num1 = 3.14
num2 = 2.72

if abs(num1 - num2) < eps:
    print("equals")

else:
    print("not equals")
```

Битовые операции

```
# изменение бита 1 на бит 0 и наоборот
~5                  # битовое НЕ
# операция И между соответствующими битами операндов
5 & 4               # битовое И
# операция ИЛИ между соответствующими битами операндов
5 | 4               # битовое ИЛИ
# операция исключающего ИЛИ между соответствующими битами операндов
5 ^ 4               # ислючающее ИЛИ
# смещение битов 5 на 2 позиции влево
5 << 2              # смещение битов влево
# смещение битов 5 на 2 позиции вправо
5 >> 2              # смещение битов вправо
```

Функции

Пользовательские функции

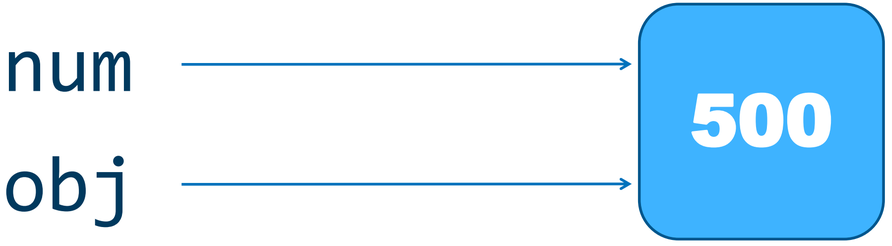
вызов функции

```
function_identifier(arg1,
arg2)
```

Параметры функции

```
def print_object_id(obj: object) -> None:
    obj_id = id(obj)
    print(f"{obj_id = }")

num = 500
print_object_id(num)
```



Параметры по умолчанию

```
def function_identifier(
    param1: int , param2: int = 5
) -> int:
    ...

# param=1 = 1, param2 == 2
function_identifier(1, 2)

# param1 == 3, param2 == 5
function_identifier(3)
```

Аннотации типов

пример игнорирования аннотаций типов

```
def print_number(num: int) -> None:
    print(f"{num = }")

print_number(num="123")
```

Возвращаемое значение

```
def print_object_id(obj: object) -> None:
    obj_id = id(obj)
    print(f"{obj_id = }")

print(print_object_id(5))
# obj_id = 140706553521064
# None
```

return

```
def print_number(num: int) -> int:
    print(f"{num = }")
    return num
    print("This message will never be printed")

num = print_number(5)
# num = 5
```

min и max

```
min(1, 2)                    # == 1
max(1, 2)                    # == 2
min(-1, 5, 8, -18)           # == -18
max(-1, 5, 8, -18)           # == 8
```

Практическая часть