

Питон, начало

Хашин С.И.

<http://math.ivanovo.ac.ru/dalgebra/Khashin/index.html>

Ивановский университет

Питон, начало

Иваново-2023

План

IDE

Комментарии

int

float

Строки

boolean

time

Исключения

IDLE, PyCharm

Основные оболочки: IDLE, PyCharm.

Где искать IDLE, если нет на рабочем столе и в меню:

C:/Program Files/Python/Lib/idlelib/idle.bat

Настройка интерпретатора Питон в PyCharm:

Меню -> File -> Setting -> Project... -> Python Interpreter

Add Interpreter -> Add local interpreter

Здесь самое главное: выбрать «Existing», а не «New»!

Затем список скриптов для запуска (вверху, слева от зелёного треугольника) -> Edit Configuration -> Python Interpreter

Выбрать основной, из Program Files.

Комментарии

```
x = 7 # однострочный комментарий
y = 2.1 ''' многострочный
        комментарий
        '''
z = 'ee' """ можно в двойных
        кавычках
        """
u = "ee" # или даже так
a = 3; b = 2 # так можно, но не рекомендуется
c, d = 4, -1 # лучше так
```

Целые числа

```
a = 123456789
```

```
b = 111_222_333
```

```
print(a,b)
```

```
> 123456789 111222333
```

```
print(f'a={a},b={b:10d}', end=' ')
```

```
print(f"a+b={a+b}, a*b={a*b}")
```

```
> a=123456789,b= 111222333 a+b=234679122, a*b=1373115209720
```

```
print('a**2=', a**2)
```

```
a**2= 15241578750190521
```

```
print('a**4=', a**4)
```

```
a**4= 232305722798259244150093798251441
```

```
print('a**6=', a**6)
```

```
a**6= 3540705968149597751242378595390670323015412790761
```

Целые операции

```
print(10//3, 10%3, 10/3)
> 3 1 3.3333333333333335
```

```
print(13%10, 3%10, -3%10)
> 3 3 7
```

Частное и остаток:

```
q, r = divmod(-137, 10) # q=-14, r=3
```

Битовые операции

```
print( 6&3, 6|3, 6^3)    # 2 7 5
print( 5<<1, 5<<2)      # 10 20
print( 5>>1, 5>>2)      # 2 1
```

Операции с присваиванием

```
x = 11
```

```
x += 3; print(x) # 14
```

```
x *= 3; print(x) # 42
```

```
x /=10; print(x) # 4
```

```
x = 65
```

```
x %= 10; print(x) # 5
```

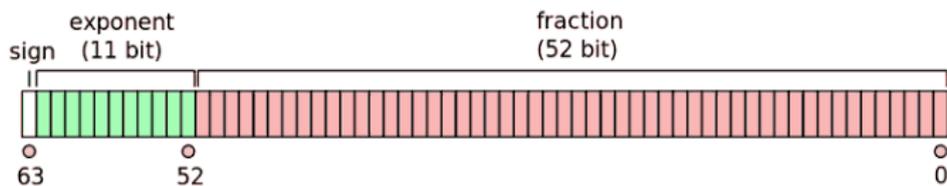
```
x ++; print(x) # ОШИБКА! нет ++, --
```

Факториал

```
import math
print(math.factorial(5))
> 120
print(math.factorial(155))
> 4789142901463393876335775239063022722176295591337767\
  1740700963399291533816224332641465693292743476559561\
  1048437231158693602074917542907666100321627438247547\
  7806479918110524333880196139452687559896255940215628\
  508414806740389616633144934400000000000000000000000\
  000000000000000
```

float, структура

Всего 8 байт, 64 бита.



Значение:

$$value = (-1)^{sign} (1.b_{51}b_{50} \dots b_0)_2 \times 2^{e-1023}$$

float, границы

Целые числа имеют неограниченный размер, а плавающие — фиксированный (8 байт)

```
k = 10**3; x = k*1.23; print('x=',x)
```

```
> x= 1230.0
```

```
k = 10**30; x = k*1.23; print('x=',x)
```

```
> x= 1.23e+30
```

```
k = 10**300; x = k*1.23; print('x=',x) print(f'x={x:10.3e}')
```

```
> x= 1.23000000000000001e+300
```

```
k = 10**310; x = k*1.23; print('x=',x)
```

```
> x=1.230e+300
```

```
k = 10**310; x = k*1.23; print('x=',x)
```

```
> OverflowError: int too large to convert to float
```

float, границы

```
x = 1.0; y = x + 0.0001
print(y-x)
> 9.999999999998899e-05
print(f'y-x={y-x:10.5e}')
> y-x=1.00000e-04
```

```
y = x + 1e-15
print(f'y-x={y-x:10.5e}')
> y-x=1.11022e-15
```

```
y = x + 1e-16
print(y-x)
> 0.0
```

float, границы

```
x=3.14; print(x.is_integer())
```

```
> False
```

```
x=314.0; print(x.is_integer())
```

```
> True
```

```
x=10.01; print(1, x.is_integer())
```

```
> False
```

Задание. При каком наименьшем k число $10.01 * *k$ окажется целым?

Ответ:

```
x=10.01; print(1, x.is_integer())  
> False  
x=10.01**15; print(10, x.is_integer())  
> False  
x=10.01**16; print(10, x.is_integer())  
> False
```

float, особые значения

sign	exponenta	mantissa	value
0	00000000000	0000...0000	+0
1	00000000000	0000...0000	-0
0	11111111111	0000...0000	+infy
1	11111111111	0000...0000	-infy
0	11111111111	0000...0001	NaN (sNaN)
0	11111111111	1000...0001	NaN (qNaN)
0	11111111111	1111...1111	NaN (an alternative)

Задачи

Задания. 1. Что больше: e^π или π^e ?

2. Известно, что $\lim_{n \rightarrow \infty} \left(1 + \frac{1}{n}\right)^n = e \approx 2.718281828$. При каком

n , $\left(1 + \frac{1}{n}\right)^n > 2.7$? > 2.71 ? > 2.71828 ?

3. Для какого наибольшего целого n можно вычислить e^n ? А для какого наименьшего?

4а. Известно, что, что при больших n :

$$n! \approx \sqrt{2\pi n} \left(\frac{n}{e}\right)^n.$$

Найдите погрешность этой формулы (в процентах) при $n = 10, 30, 100, 300, 1000, 3000$.

4б. Уточнение:

$$n! \approx \sqrt{2\pi n} \left(\frac{n}{e}\right)^n \left(1 + \frac{1}{12n} + \frac{1}{288n^2}\right).$$

Строки

```
s, r, t = 'abc', 'Русские буквы', "Двойные кавычки"
print(s+t)      #abcДвойные кавычки
print(s*4)      # abcabcabcabc !Повтор 4 раза
Многострочная константа:
    s = '''Тройные кавычки!
        Продолжение
        кавычки " внутри и одинарная кавычка' конец'''
print(s)
```

Срезы:

```
print(r[:7])      # Русские
print(r[2:7])     # сские
print(r[-5:])     # буквы          Начиная с конца
print(r[-11:-2]) # сские бук
print(r[-11:-2:2]) # скебк          Через одну букву
```

Удаление пробелов

strip:

```
line = '  один два три  '
print('<', line.lstrip(), '>') # < один два три >
print('<', line.rstrip(), '>') # <  один два три >
print('<', line.strip(), '>')  # < один два три >
```

Поиск и замена

strip:

```
line = 'мама мыла раму'  
print(line.find('a'))  
print(line.rfind('a'))  
print(line.count('a'))  
print(line.replace('a', '*'))  
print(line.replace('a', '*', 2))  
print(line.replace('ма', '*'))
```

Задачи

- Задание. 1. а) Сколько нулей в числе 300! ?
б) На сколько нулей оканчивается число 300! ?
2. Пусть `s='Сколько нулей в числе'`; `ch='щ'`. Входит ли символ `ch` в строку `s`?
3. Найдите индексы первого и последнего вхождения символа `ch='о'` в строку `s`.
4. Дополните строку `s` подчеркиваниями справа до длины 40.
5. В строке `s` заменить знаки препинания `'.,;:&'` на пробелы.

boolean

```
t, f = True, False
print(type(t)) # Prints <class 'bool'>
print(t and f) # Logical AND; prints "False"
print(t or f)  # Logical OR;  prints "True"
print(not t)   # Logical NOT; prints "False"
print(t != f) # Logical XOR; prints "True"
```

Дата и время

```
import time
print(time.time())      # текущее время в секундах
> 1553929407.9740036
```

Дата и время:

```
import datetime
now = datetime.datetime.now()
print(now)              # 2018-09-20 17:46:38.502740
print(now.date())       # 2018-09-20
print(now.time())       # 17:46:38.502740
dt = datetime.date(1924, 6, 24)
print(dt)               # 1924-06-24
print(now.date()-dt)    # 34421 days, 0:00:00
```

Задания

Задание 1. Найти количество дней, прошедших с 1 января 1900 г.

Задание 2. Найти количество дней с вашего рождения.

Задание 3. Найти количество секунд с начала года.

Задание 4. 1 млрд. секунд — это сколько времени?

Исключения

```
def one_x(x):  
    if x==0:  
        raise Exception(f'one_x({x})')  
    return 1/x  
  
print(one_x(2))  
print(one_x(0))
```