

QA
Light

Python

Михаил

Базовые типы данных

Тип объекта	Пример литерала/создания
Числа	<code>1234, 3.1415, 3+4j, Decimal, Fraction</code>
Строки	<code>'spam', "guido's", b'a\x01c'</code>
Списки	<code>[1, [2, 'three'], 4]</code>
Словари	<code>{'food': 'spam', 'taste': 'yum'}</code>
Кортежи	<code>(1, 'spam', 4, 'U')</code>
Файлы	<code>myfile = open('eggs', 'r')</code>
Множества	<code>set('abc'), {'a', 'b', 'c'}</code>
Прочие базовые типы	Сами типы, None, логические значения

Числа

```
>>> 123 + 222 # Целочисленное сложение  
345
```

```
>>> 1.5 * 4 # Умножение вещественных чисел  
6.0
```

```
>>> 2 ** 100 # 2 в степени 100  
1267650600228229401496703205376
```

```
>>> len(str(2 ** 1000000)) # Сколько цифр в  
действительно БОЛЬШОМ числе?  
301030
```

Числа

```
>>> 3.1415 * 2 # repr: как программный код  
6.283000000000000004
```

```
>>> print(3.1415 * 2) # str: более  
дружественный формат  
6.283
```

```
>>> import math  
>>> math.pi  
3.1415926535897931  
>>> math.sqrt(85)  
9.2195444572928871
```

Числа

```
>>> import random  
>>> random.random()  
0.59268735266273953  
>>> random.choice([1, 2, 3, 4])  
1
```

```
>>> a = [1,2,3,4,5]  
>>> random.shuffle(a)  
>>> a  
[5, 1, 4, 3, 2]
```

Строки

```
>>> S = 'Spam'
>>> len(S)    # Длина
4
>>> S[0]      # Первый элемент в S, счет начинается с позиции 0
'S'
>>> S[1]      # Второй элемент слева
'p'
>>> S[-1]     # Последний элемент в конце S
'm'
>>> S[len(S)-1] # Отрицательная индексация, более сложный
                # способ
'm'
>>> S[-2]     # Второй элемент с конца
'a'
```

Строки (срезы)

```
>>> S          # Строка из 4 символов
'Spam'
>>> S[1:3]      # Срез строки S начиная со смещения 1 и до 2 (не 3)
'pa'
>>> S[1:]       # Все, кроме первого элемента (1:len(S))
'pam'
>>> S[0:3]      # Все, кроме последнего элемента
'Spa'
>>> S[:3]       # То же, что и S[0:3]
'Spa'
>>> S[:-1]      # Еще раз все, кроме последнего элемента, но проще (0:-1)
'Spa'
>>> S[:]        # Все содержимое S, как обычная копия (0:len(S))
'Spam'
```

Строки (неизменяемость)

```
>>> S + 'xyz'    # Конкатенация
```

```
'Spamxyz'
```

```
>>> S            # S остается без изменений
```

```
'Spam'
```

```
>>> S * 8        # Повторение
```

```
'SpamSpamSpamSpamSpamSpamSpamSpam'
```

```
>>> S[0] = 'z'    # Неизменяемые объекты нельзя изменить
```

```
TypeError: 'str' object does not support item assignment
```

```
>>> S = 'z' + S[1:] # Но с помощью выражений мы можем создавать  
                     # новые объекты
```

```
>>> S
```

```
'zspam'
```


Строки (методы)

```
>>> S.find('pa') # Поиск смещения подстроки
```

```
1
```

```
>>> S
```

```
'Spam'
```

```
>>> S.replace('pa', 'XYZ') # Замена одной  
подстроки другой
```

```
'SXYZm'
```

```
>>> S
```

```
'Spam'
```

Строки (методы)

```
>>> line= 'aaa,bbb,cccc,dd'
>>> line.split(',') # Разбивает строку по разделителю и создает список строк
['aaa', 'bbb', 'cccc', 'dd']
>>> S = 'spam'
>>> S.upper()      # Преобразование символов в верхний и в нижний
    регистр
'SPAM'
>>> S.isalpha()    # Проверка содержимого: isalpha, isdigit и так далее
True
>>> line = 'aaa,bbb,cccc,dd\n'
>>> line= line.rstrip() # Удаляет завершающие пробельные символы
>>> line
'aaa,bbb,cccc,dd'
```

Строки (форматирование)

Выражение (во всех версиях)

```
>>> '%s, eggs, and %s' % ('spam', 'SPAM!')  
'spam, eggs, and SPAM!'
```

Метод (2.6, 3.0)

```
>>> '{0}, eggs, and {1}'.format('spam', 'SPAM!')  
'spam, eggs, and SPAM!'
```

Строки (получение помощи)

```
>>> dir(S)
['__add__', '__class__', '__contains__', '__delattr__',
 '__doc__', '__eq__',
 '__format__', '__ge__', '__getattribute__', '__getitem__',
 '__getnewargs__', ...]
```

```
>>> help(S.replace)
```

Help on built-in function replace:

(Справка о встроенной функции replace:)

```
replace(...)
```

S.replace (old, new[, count]) -> str...

```
'\naaaaaaaaaaaa\nbbb\'\'\'bbbbbbbbbb\'\'\'bbbbbb\'\'bbb\nccccccccc\ncccccccc'
```

Списки

```
>>> L = [123, 'spam', 1.23] # Список из трех объектов разных типов
```

```
>>> len(L) # Число элементов в списке
```

```
3
```

```
>>> L[0] # Доступ к элементу списка по его индексу
```

```
123
```

```
>>> L[: -1] # Операция получения среза возвращает новый список
```

```
[123, 'spam']
```

```
>>> L + [4, 5, 6] # Операция конкатенации также возвращает новый список
```

```
[123, 'spam', 1.23, 4, 5, 6]
```

```
>>> L # Наши действия не привели к изменению оригинального списка
```

```
[123, 'spam', 1.23]
```

Вложенные списки

```
>>> M = [[1, 2, 3], # Матрица 3 x 3 в виде вложенных списков  
         [4, 5, 6], # Выражение в квадратных скобках может  
         [7, 8, 9]] # занимать несколько строк
```

```
>>> M  
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
```

```
>>> M[1] # Получить строку 2  
[4, 5, 6]
```

```
>>> M[1][2] # Получить строку 2, а затем элемент 3 в этой  
            строке  
6
```

Приведения типов

Выражение	Результат выполнения
-----------	----------------------

int ("56")	56
int (4.03)	4
int ("comp 486")	О ш и б к а
str (56)	'56'
str (4.03)	'4.03'
float (56)	56.0
float ("56")	56.0

Во сколько раз площадь Луны меньше площади Земли?



$$S = \pi R^2$$

Радиус Земли = 6371,0 км

Радиус Луны = 1737,1 км

теорема Пифагора

Задача 1. Найдите гипотенузу прямоугольного треугольника, если катеты равны 6 см и 8 см.

Дано:

$\triangle ABC$

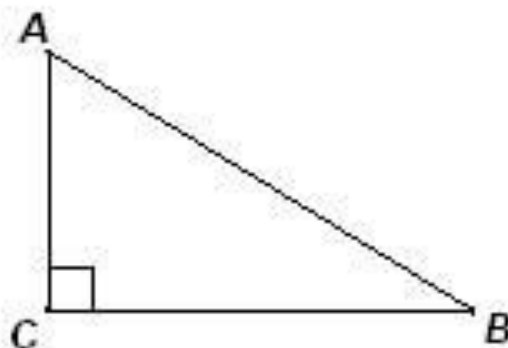
$AC = 6 \text{ см}$

$BC = 8 \text{ см}$

$\angle C = 90^\circ$

Найдите:

$AB - ?$



Решение: По теореме Пифагора:

$$AB^2 = AC^2 + BC^2,$$

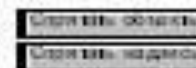
$$AB^2 = 6^2 + 8^2,$$

$$AB^2 = 36 + 64,$$

$$AB^2 = 100,$$

$$AB = 10.$$

Ответ: $AB = 10 \text{ см.}$



Вычислите, во сколько раз увеличится скорость реакции при повышении температуры от 30 до 70 °C, если температурный коэффициент скорости равен 2.

Решение:

По правилу Вант-Гоффа

$$v = v_0 \cdot \gamma^{(t_2 - t_1)/10}$$

По условию задачи требуется определить v/v_0 :

$$v/v_0 = 2^{(70 - 30)/10} = 2^4 = 16$$

Python

Киев
ул. Космонавта Комарова 1
НАУ, корп.11

+38 (097) 78 - 010 - 78

+38 (099) 78 - 010 - 78

+38 (063) 78 - 010 - 78

info@qalight.com.ua