

Python

2.2

- `>>> a=1`
- `>>> b=2`
- `>>> a or b`
- `1`
- `>>> a and b`
- `2`
- `>>> a=True`
- `>>> not a`
- `False`
- `>>> a=1`
- `>>> not a`
- `False`
- `>>> not b`
- `False`

- `>>> a=u'привет'`
- `>>> a`
- `'привет'`
- `>>> a=r'm,cnsc \n wcd'`
- `>>> a`
- `'m,cnsc \\n wcd'`
- `>>> print (a)`
- `m,cnsc \n wcd`
- `>>>`

- `>>> tuplea=(1,)`
- `>>> tuplea`
- `(1,)`
- `>>> tuplea[0]`
- `1`

Числа

- `>>> 1.1+2.2`
- `3.300000000000000003`
- `>>> 0.1+0.1+0.1-0.3`
- `5.551115123125783e-17`
- `>>> >>> 0.1+0.1-0.2==0`
- `True`

decimal

- `>>> from decimal import *`
- `>>> Decimal(10)`
- `Decimal('10')`
- `>>> Decimal('3.14')`
- `Decimal('3.14')`
- `>>> Decimal(0.1)`
- `Decimal('0.10000000000000000000055511151231257827021181583404541015625')`
- `>>> Decimal((0,(3,1,4),-3))`
- `Decimal('0.314')`

- >>> Decimal((1,(3,1,4),-3))
- Decimal('-0.314')
- >>> Decimal((0,(3,1,4),-4))
- Decimal('0.0314')
- >>>

- `>>> a=Decimal('0.1')`
- `>>> a`
- `Decimal('0.1')`
- `>>> b=Decimal('0.3')`
- `>>> a+a+a-b`
- `Decimal('0.0')`
- `>>> a+a+a-b==0`
- `True`
- `>>>`

- `>>> TWOPLACES=Decimal(10)**-2`
- `>>> c.quantize(TWOPLACES)`
- `Decimal('0.37')`

Дроби Fraction

- `>>> from fractions import Fraction`
- `>>> a=Fraction(1,2)`
- `>>> a`
- `Fraction(1, 2)`
- `>>> b=Fraction(3,2)`
- `>>> b`
- `Fraction(3, 2)`
- `>>> a+b`
- `Fraction(2, 1)`
- `>>> int(a+b)`
- `2`
- `>>> float(a+b)`
- `2.0`
- `>>> c=Fraction(0.5)`
- `>>> c`
- `Fraction(1, 2)`
- `>>> d=Fraction('3/2')`
- `>>> d`
- `Fraction(3, 2)`
- `>>> c+d`
- `Fraction(2, 1)`
- `>>>`

- `>>> Fraction(1.1)`
- `Fraction(2476979795053773,
2251799813685248)`
- `>>> Fraction(1.1).limit_denominator()`
- `Fraction(11, 10)`
- `>>>`

tuple

- `>>> a=tuple()`
- `>>> a`
- `()`
- `>>> b=()`
- `>>> b`
- `()`
- `>>> c=('a',)`
- `>>> c`
- `('a',)`
- `>>> c=('a')`
- `>>> c`
- `'a'`
- `>>> d=tuple(['a','b','c'])`
- `>>> d`
- `('a', 'b', 'c')`

- `>>> a1=100`
- `>>> a2=200`
- `>>> a1,a2=a2,a1`
- `>>> a2`
- `100`
- `>>> a1`
- `200`
- `>>> a1,a2`
- `(200, 100)`
- `>>> def abc():`
- `return a,b,c,`

- `>>> x,y,z =abc()`
- `>>> x`
- `()`
- `>>> y`
- `()`
- `>>> z`
- `'a'`

- `>>> (1,2,3)+(4,5,6)`
- `(1, 2, 3, 4, 5, 6)`
- `>>> t=(1,2,3)+(4,5,6)`
- `>>> t[0]`
- `1`
- `>>> t[1:-1]`
- `(2, 3, 4, 5)`
- `>>> t[::-1]`
- `(6, 5, 4, 3, 2, 1)`
- `>>> len(t)`
- `6`
- `>>> t.index(2)`
- `1`
- `>>> t.count(1)`
- `1`

- `>>> a=[i for i in range(100)]`
- `>>> a`
- `[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99]`
- `>>> a.__sizeof__()`
- `444`
- `>>> b=(i for i in range(100))`
- `>>> b`
- `<generator object <genexpr> at 0x03B161E8>`
- `>>> b.__sizeof__()`
- `24`

- `>>> c=[1,2,3,4,5,6,7,8,9,0]`
- `>>> d=(1,2,3,4,5,6,7,8,9,0)`
- `>>> c.__sizeof__()`
- 60
- `>>> d.__sizeof__()`
- 52

- `>>> a={(1,2,3):'abc'}`
- `>>> a`
- `{(1, 2, 3): 'abc'}`

файлы

- `>>> f=open('text.txt')`
- `>>> f`
- `<_io.TextIOWrapper name='text.txt' mode='r' encoding='cp1251'>`

- `>>> f.read()`
- 'Реферат по математике и физике....'

- `>>> f.read(10)`
- `"`
- `>>> f.seek(0)`
- `0`
- `>>> f.read(10)`
- `'Реферат по'`

- >>> for line in f:
- print(line)
-
- Реферат по математике и физике
- Тема: «Почему ненаблюдаемо умножение вектора на число?»...

- `>>> lines=f.readlines()`
- `>>> lines`
- `['Реферат по математике и физике\n', ...]`
- `>>> f.close()`

- `>>> f2=open('text2.txt','w')`
- `>>> f2`
- `<_io.TextIOWrapper name='text2.txt' mode='w' encoding='cp1251'>`
- `>>> l=[str(i) for i in range(10)]`
- `>>> l`
- `['0', '1', '2', '3', '4', '5', '6', '7', '8', '9']`
- `>>> for i in l:`
- `f2.write(i+'\n')`

- `>>> f2.close()`

- `>>> f2=open('text2.txt','r')`
- `>>> l=[line for line in f2]`
- `>>> l`
- `['0\n', '1\n', '2\n', '3\n', '4\n', '5\n', '6\n', '7\n', '8\n', '9\n']`
- `>>> f2.seek(0)`
- `0`
- `>>> l=[line.strip() for line in f2]`
- `>>> l`
- `['0', '1', '2', '3', '4', '5', '6', '7', '8', '9']`

- `>>> l=[line.strip() for line in open('text2.txt')]`
- `>>> l`
- `['0', '1', '2', '3', '4', '5', '6', '7', '8', '9']`
- `>>> max([line.strip() for line in open('text.txt')])`
- 'Функция выпуклая кверху облучает положительный резонатор. Кристалл синхронизирует бином Ньютона одинаково по всем направлениям. Линза охватывает лептон. Теорема Ферма определяет взрыв. Теорема представляет собой межядерный кварк. Учитывая, что $(\sin x)' = \cos x$, плазменное образование неупруго.'

Словари

- `>>> b={'key1':'value1','key2':2,'key3':[1,2,3]}`
- `>>> b`
- `{'key2': 2, 'key1': 'value1', 'key3': [1, 2, 3]}`
- `>>> a={}`
- `>>> a`
- `{}`

- `>>> c=dict(key='value', spam='eggs')`
- `>>> c`
- `{'spam': 'eggs', 'key': 'value'}`
- `>>> d=dict([(1,10),(2,20),(3,30)])`
- `>>> d`
- `{1: 10, 2: 20, 3: 30}`

- `>>> e=dict.fromkeys(['a','b','c'])`
- `>>> e`
- `{'a': None, 'b': None, 'c': None}`
- `>>> e=dict.fromkeys(['a','b','c'],'init')`
- `>>> e`
- `{'a': 'init', 'b': 'init', 'c': 'init'}`

- `>>> e['a']`
- `'init'`
- `>>> e['a']='spam'`
- `>>> e`
- `{'a': 'spam', 'b': 'init', 'c': 'init'}`
- `>>> del e['a']`
- `>>> e`
- `{'b': 'init', 'c': 'init'}`

- `>>> len(e)`
- `2`
- `>>> 'b' in e`
- `True`
- `>>> 'a' in e`
- `False`

- `>>> 'b' not in e`
- `False`
- `>>> not 'a' in e`
- `True`

- `>>> e.get('a')`
- `>>> e.get('b')`
- `'init'`
- `>>> e.get('d','initial')`
- `'initial'`
- `>>> e['d']=e.get('d',0)+1`
- `>>> e`
- `{'d': 1, 'b': 'init', 'c': 'init'}`
- `>>> e['d']=e.get('d',0)+1`
- `>>> e`
- `{'d': 2, 'b': 'init', 'c': 'init'}`

- `>>> e.items()`
- `dict_items([('d', 2), ('b', 'init'), ('c', 'init')])`
- `>>> e.keys()`
- `dict_keys(['d', 'b', 'c'])`
- `>>> e.values()`
- `dict_values([2, 'init', 'init'])`

- `>>> for key,value in e.items():`
- `print (key,value)`

-
- `d 2`
- `b init`
- `c init`
- `>>> for key in e.keys():`
- `print (key,e[key])`

-
- `d 2`
- `b init`
- `c init`
- `>>> for val in e.values():`
- `print(val)`

-
- `2`
- `init`
- `init`

- `>>> d={'a':'alpha','b':'beta','g':'gamma'}`
- `>>> d`
- `{'a': 'alpha', 'b': 'beta', 'g': 'gamma'}`
- `>>> for key in sorted(d.keys()):`
- `print(key,d[key])`
-
- a alpha
- b beta
- g gamma

- >>> d={'one':10,'more':44,'thing':9}
- >>> d=sorted(d.items(),key=lambda x:x[-1])
- >>> d
- [('thing', 9), ('one', 10), ('more', 44)]
- >>> for key,value in d:
- print ('{1}{0}'.format(key,value))
-
- 9thing
- 10one
- 44more
- >>> for key,value in d:
- print ('{1:3} {0}'.format(key,value))
-
- 9 thing
- 10 one
- 44 more
- >>> d=dict(d)
- >>> d
- {'thing': 9, 'more': 44, 'one': 10}

- `>>> d.pop('one')`
- `10`
- `>>> d`
- `{'thing': 9, 'more': 44}`
- `>>> d.popitem()`
- `('thing', 9)`
- `>>> d`
- `{'more': 44}`

- `>>> d.setdefault('x','spam')`
- `'spam'`
- `>>> d`
- `{'more': 44, 'x': 'spam'}`
- `>>> d.setdefault('more','spam')`
- `44`
- `>>> d`
- `{'more': 44, 'x': 'spam'}`

- `>>> dct={i:i**2 for i in range(10)}`
- `>>> dct`
- `{0: 0, 1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81}`

Множества set

- `>>> a=set()`
- `>>> a`
- `set()`
- `>>> b=set(['a','b','c','a'])`
- `>>> b`
- `{'a', 'b', 'c'}`

- `>>> c=set('hello word')`
- `>>> c`
- `{'w', 'd', 'l', ' ', 'r', 'e', 'h', 'o'}`
- `>>> d={'a','b','c','a'}`
- `>>> d`
- `{'a', 'b', 'c'}`
- `>>> e={i**2 for i in range (10)}`
- `>>> e`
- `{0, 1, 64, 4, 36, 9, 16, 49, 81, 25}`

- `>>> len(e)`
- `10`
- `>>> 81 in e`
- `True`
- `>>> 81 not in e`
- `False`
- `>>> not 81 in e`
- `False`
- `>>> c1=set('hello')`
- `>>> c2={'h','l'}`
- `>>> c1==c2`
- `False`
- `>>>`

- `>>> s1={'a','b','c'}`
- `>>> s2={'b','z','x'}`
- `>>> s1<=s2`
- `False`
- `>>> s2<=s1`
- `False`
- `>>> s3={'a','b'}`
- `>>> s3<s1`
- `True`

- `>>> s3.issubset(s1)`
- `True`
- `>>> s4=s1 | s2`
- `>>> s4`
- `{'z', 'b', 'x', 'a', 'c'}`
- `>>> s5=s1.union(s2)`
- `>>> s5`
- `{'z', 'b', 'x', 'a', 'c'}`

- `>>> s5=s1&s2`
- `>>> s5`
- `{'b'}`
- `>>> s6=s1-s2`
- `>>> s6`
- `{'a', 'c'}`
- `>>> s7=s1^s2`
- `>>> s7`
- `{'a', 'z', 'x', 'c'}`

- `>>> s1.update(s2)`
- `>>> s1`
- `{'z', 'b', 'x', 'a', 'c'}`
- `S1=s1+s2`
- `>>> s1.add('d')`
- `>>> s1`
- `{'z', 'b', 'x', 'd', 'a', 'c'}`

- `>>> s1.remove('x')`
- `>>> s1`
- `{'z', 'b', 'd', 'a', 'c'}`
- `>>> s1.discard('d')`
- `>>> s1`
- `{'z', 'b', 'a', 'c'}`
- `>>> s1.pop()`
- `'z'`
- `>>> s1`
- `{'b', 'a', 'c'}`

- `>>> s1.clear()`
- `>>> s1`
- `set()`

frozenset

- `>>> a=frozenset()`
- `>>> a`
- `frozenset()`
- `>>> a=frozenset('hello word')`
- `>>> a`
- `frozenset({'w', 'd', 'l', ' ', 'r', 'e', 'h', 'o'})`
- `>>> a.add('x')`
- Traceback (most recent call last):
- File "`<pyshell#252>`", line 1, in `<module>`
- `a.add('x')`
- `AttributeError: 'frozenset' object has no attribute 'add'`