QA Light

Python

Михаил

Модули

Подключение модуля из стандартной библиотек

```
>>> import time, random

>>> time.time()
1376047104.056417

>>> random.random()
0.98745508333306869
```

Использование псевдонимов

```
>>> import math as m

>>> m.e
2.718281828459045
```

Инструкция from

```
from <Название модуля> import
    <Атрибут 1> [ as <Псевдоним 1> ],
    [<Атрибут 2> [ as <Псевдоним 2> ] ...]
from <Название модуля> import *
```

```
>>> from math import e, ceil as c
>>> e
2.718281828459045
>>> c(4.6)
5
```

+38 (063) 78-010-78
+38 (097) 78-010-78
+38 (099) 78-010-78
info@qalight.com.ua
qalight.com.ua

__main__

```
def hello():
    print('Hello, world!')

def fib(n):
    a = b = 1
    for i in range(n - 2):
        a, b = b, a + b
    return b

if __name__ == "__main__":
    hello()
    for i in range(10):
        print(fib(i))
```

__builtins__

```
>>> dir()
['__builtins__', '__doc__', '__name__', '__package__']
>>> dir(__builtins__)
['ArithmeticError', 'AssertionError', 'AttributeError', 'BaseException',
    'BufferError… '_', '__debug__', '__doc__', '__import__', '__name__',
    '__package__', 'abs', 'all', 'any', 'apply', 'basestring', 'bin', 'bool',
    'buffer', 'bytearray', 'bytes', 'callable', 'chr', 'classmethod', 'cmp',
    'coerce', 'compile', 'complex', 'copyright', 'credits', 'delattr', 'dict', 'dir',
    'divmod', 'enumerate', 'eval', 'execfile', 'exit', 'file', 'filter', 'float', 'format',
    'frozenset', 'getattr', 'globals', 'hasattr', 'hash', 'help', 'hex', 'id', 'input',
    'int', 'intern', 'isinstance', 'issubclass', 'iter', 'len', 'license', 'list', 'locals',
    'long', 'map', 'max', 'memoryview', 'min', 'next', 'object', 'oct', 'open',
    'ord', 'pow', 'print', 'property', 'quit', 'range', 'raw_input', 'reduce',
    'reload', 'repr', 'reversed', 'round', 'set', 'setattr', 'slice', 'sorted',
    'staticmethod', 'str', 'sum', 'super', 'tuple', 'type', 'unichr', 'unicode',
    'vars', 'xrange', 'zip']
```

collections.Counter

```
>>> import collections
>>> c = collections.Counter()
>>> for word in ['spam', 'egg', 'spam',
    'counter', 'counter', 'counter']:
...     c[word] += 1
...
>>> print(c)
Counter({'counter': 3, 'spam': 2, 'egg': 1})
>>> print(c['counter'])
3
>>> print(c['collections'])
0
```

+38 (063) 78-010-78
+38 (097) 78-010-78
+38 (099) 78-010-78
info@qalight.com.ua
qalight.com.ua

collections.Counter

```
>>> c = Counter(a=4, b=2, c=0, d=-2)
>>> list(c.elements())
['a', 'a', 'a', 'a', 'b', 'b']

>>>
    Counter('abracadabra').most_common(3
    )
[('a', 5), ('r', 2), ('b', 2)]

>>> c = Counter(a=4, b=2, c=0, d=-2)
>>> d = Counter(a=1, b=2, c=3, d=4)
>>> c.subtract(d)
Counter({'a': 3, 'b': 0, 'c': -3, 'd': -6})
```

+38 (063) 78-010-78
+38 (097) 78-010-78
+38 (099) 78-010-78
info@qalight.com.ua
qalight.com.ua

collections.Counter

sum(c.values()) - общее количество.

c.clear() - очистить счётчик.

list(c) - список уникальных элементов.

set(c) - преобразовать в множество.

dict(c) - преобразовать в словарь.

c.most_common()[:-n:-1] - n наименее часто встречающихся элементов.

c += Counter() - удалить элементы, встречающиеся менее одного раза.

collections.Counter

```
>>> c = Counter(a=3, b=1)
>>> d = Counter(a=1, b=2)
>>> c + d
Counter({'a': 4, 'b': 3})
>>> c - d
Counter({'a': 2})
>>> c & d
Counter({'a': 1, 'b': 1})
>>> c | d
Counter({'a': 3, 'b': 2})
```

collections.OrderedDict

QA Light

словарь помнит порядок, в котором ему были даны ключи

```
>>> d = {'banana': 3, 'apple':4, 'pear': 1, 'orange': 2}

>>> OrderedDict(sorted(d.items(), key=lambda t: t[0]))
OrderedDict([('apple', 4), ('banana', 3), ('orange', 2), ('pear', 1)])
>>> OrderedDict(sorted(d.items(), key=lambda t: t[1]))
OrderedDict([('pear', 1), ('orange', 2), ('banana', 3), ('apple', 4)])
>>> OrderedDict(sorted(d.items(), key=lambda t: len(t[0])))
OrderedDict([('pear', 1), ('apple', 4), ('orange', 2), ('banana', 3)])
```

collections.namedtuple()

Ведущий себя как кортеж, с тем дополнением, что каждому элементу присваивается имя, по которому можно в дальнейшем получать доступ:

```
>>> Point = namedtuple('Point', ['x', 'y'])
>>> p = Point(x=1, y=2)
>>> p
Point(x=1, y=2)
>>> p.x
1
>>> p[0]
1
```

datetime

Класс datetime.date(year, month, day) - стандартная дата. Атрибуты: year, month, day. Неизменяемый объект.

Класс datetime.time(hour=0, minute=0, second=0, microsecond=0, tzinfo=None) - стандартное время, не зависит от даты. Атрибуты: hour, minute, second, microsecond, tzinfo.

Класс datetime.timedelta - разница между двумя моментами времени, с точностью до микросекунд.

datetime

```
>>> from datetime import datetime, date,
    time
>>> # Using datetime.combine()
>>> d = date(2005, 7, 14)
>>> t = time(12, 30)
>>> datetime.combine(d, t)
datetime.datetime(2005, 7, 14, 12, 30)
>>> # Using datetime.now() or
    datetime.utcnow()
```

math

math.log(X, [base]) - логарифм X по основанию base. Если base не указан, вычисляется натуральный логарифм.

math.log1p(X) - натуральный логарифм (1 + X). При X → 0 точнее, чем math.log(1+X).

math.log10(X) - логарифм X по основанию 10.

math.log2(X) - логарифм X по основанию 2. Новое в Python 3.3.

math.pow(X, Y) - XY.

math.sqrt(X) - квадратный корень из X.

math.acos(X) - арккосинус X. В радианах.

math.asin(X) - арксинус X. В радианах.

math.atan(X) - арктангенс X. В радианах.

round

```
>>> print round(1123.456789, -1)
1120.0

>>> import decimal
>>> a = decimal.Decimal("8.83333333339")
>>> print(round(a,2))
8.83

>>> decimal.Decimal('8.333333').quantize(decimal.Decimal('.01'),
    rounding=decimal.ROUND_UP)
Decimal('8.34')
>>> decimal.Decimal('8.333333').quantize(decimal.Decimal('.01'),
    rounding=decimal.ROUND_DOWN)
Decimal('8.33')
```

random

```
>>> random.random()       # Random float x, 0.0 <= x < 1.0
0.37444887175646646
>>> random.uniform(1, 10)  # Random float x, 1.0 <= x < 10.0
1.1800146073117523
>>> random.randint(1, 10)  # Integer from 1 to 10, endpoints included
7
>>> random.randrange(0, 101, 2)  # Even integer from 0 to 100
26
>>> random.choice('abcdefghij')  # Choose a random element
'c'
>>> items = [1, 2, 3, 4, 5, 6, 7]
>>> random.shuffle(items)
>>> items
[7, 3, 2, 5, 6, 4, 1]
>>> random.sample([1, 2, 3, 4, 5],  3)  # Choose 3 elements
[4, 1, 5]
```

statistics.mean

```
>>> mean([1, 2, 3, 4, 4])
2.8
>>> mean([-1.0, 2.5, 3.25, 5.75])
2.625

>>> from fractions import Fraction as F
>>> mean([F(3, 7), F(1, 21), F(5, 3), F(1, 3)])
Fraction(13, 21)

>>> from decimal import Decimal as D
>>> mean([D("0.5"), D("0.75"), D("0.625"), D("0.375")])
Decimal('0.5625')
```

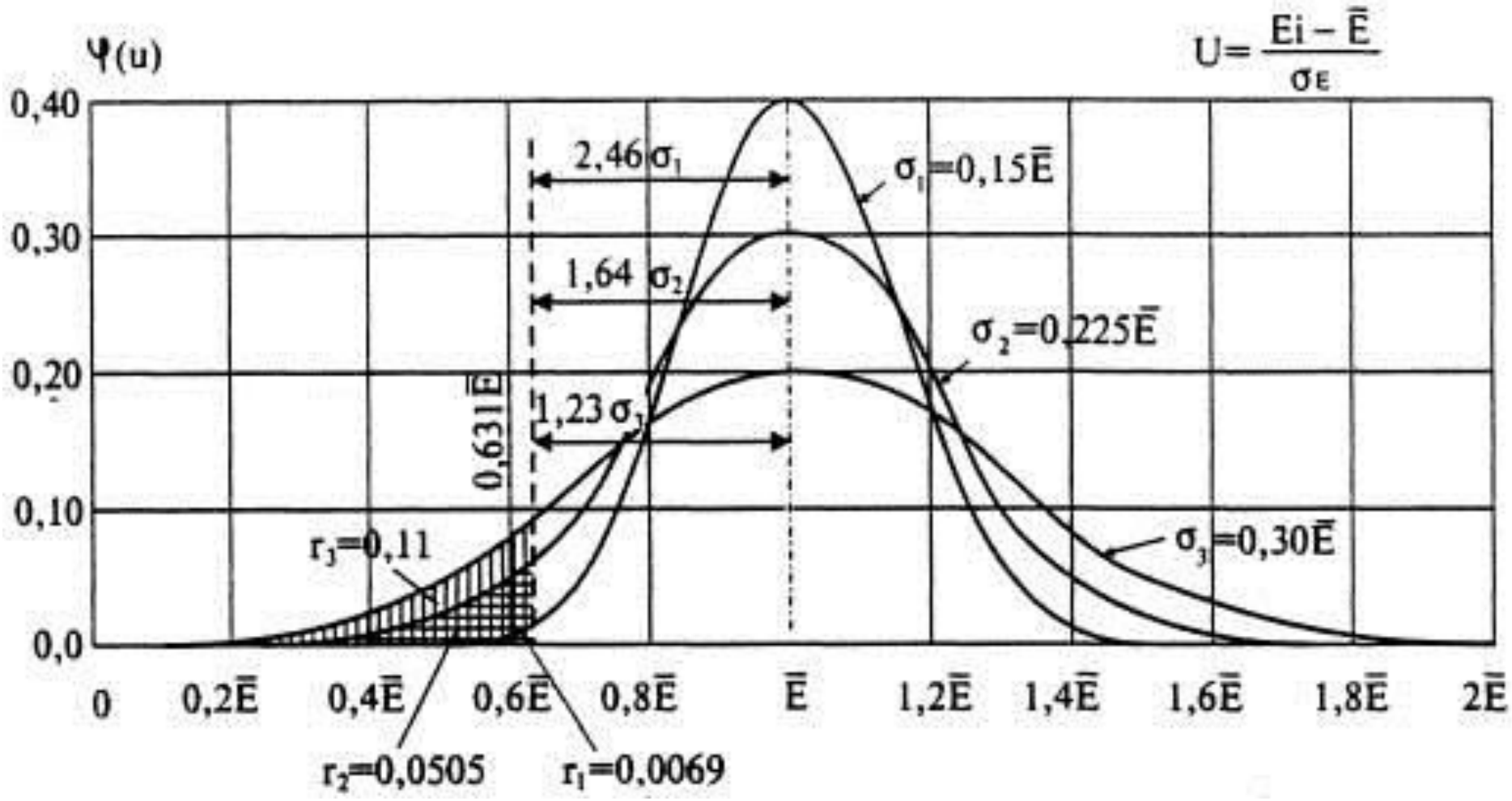
statistics.median

```
>>> median([1, 3, 5])
3
>>> median([1, 3, 5, 7])
4.0

>>> median_low([1, 3, 5])
3
>>> median_low([1, 3, 5, 7])
3

>>> median_high([1, 3, 5])
3
>>> median_high([1, 3, 5, 7])
5
```

mode, variance



+38 (063) 78-010-78
+38 (097) 78-010-78
+38 (099) 78-010-78

info@qalight.com.ua
qalight.com.ua

mode, variance

```
>>> mode([1, 1, 2, 3, 3, 3, 3, 4])
3
>>> mode(["red", "blue", "blue", "red", "green", "red", "red"])
'red'

>>> data = [2.75, 1.75, 1.25, 0.25, 0.5, 1.25, 3.5]
>>> variance(data)
1.3720238095238095
>>> from decimal import Decimal as D
>>> variance([D("27.5"), D("30.25"), D("30.25"), D("34.5"), D("41.75")])
Decimal('31.01875')
>>> from fractions import Fraction as F
>>> variance([F(1, 6), F(1, 2), F(5, 3)])
Fraction(67, 108)
```

itertools

product('ABCD', repeat=2)
AA AB AC AD BA BB BC BD CA CB CC CD DA DB DC
DD

permutations('ABCD', 2)
AB AC AD BA BC BD CA CB CD DA DB DC

combinations('ABCD', 2)
AB AC AD BC BD CD

combinations_with_replacement('ABCD', 2)
AA AB AC AD BB BC BD CC CD DD

operator

```
operator.lt(a, b) # a < b
operator.le(a, b) # a <= b
operator.eq(a, b) # a == b
operator.ne(a, b) # a != b
operator.ge(a, b) # a > b
operator.gt(a, b) # a >= b
```

os

os.name - имя операционной системы. Доступные варианты: 'posix', 'nt', 'mac', 'os2', 'ce', 'java'.

os.environ - словарь переменных окружения.

os.chdir(path) - смена текущей директории.

os.listdir(path=".") - список файлов и директорий в папке.

os.mkdir(path, mode=0o777, *, dir_fd=None) - создаёт директорию. OSError, если директория существует.

os.makedirs(path, mode=0o777, exist_ok=False) - создаёт директорию, создавая при этом промежуточные директории.

os.remove(path, *, dir_fd=None) - удаляет путь к файлу.

os.path

os.path.dirname(path) - возвращает имя директории пути path.

os.path.exists(path) - возвращает True, если path указывает на существующий путь или дескриптор открытого файла.

os.path.isfile(path) - является ли путь файлом.

os.path.isdir(path) - является ли путь директорией.

os.path.join(path1[, path2[, ...]]) - соединяет пути с учётом особенностей операционной системы.

sys

```
>>> import sys
>>> sys.version_info
(3, 0, 0, 'beta', 2)
>>> sys.version_info[0] >= 3
True

for i in sys.argv:
    print(i)

$ python3 using_sys.py we are arguments
Аргументы командной строки:
using_sys.py
we
are
arguments
```

virtualenv

```
$ sudo pip install virtualenv

$ mkdir venv && cd venv
$ virtualenv --no-site-packages <venv_name>

$ source <venv_name>/bin/activate

$ deactivate
```

pip

pip freeze > requirements.txt
pip install -r requirements.txt

```
BeautifulSoup==3.2.0
Django==1.3
Fabric==1.2.0
Jinja2==2.5.5
PyYAML==3.09
Pygments==1.4
SQLAlchemy==0.7.1
South==0.7.3
amqplib==0.6.1
anyjson==0.3
...
```

QA Light

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