

Углубленный Python

Лекция 5

Стандартная библиотека Python Строки, файлы, сокеты

Кандауров Геннадий



образование

Напоминание отметиться на портале

+ ОСТАВИТЬ ОТЗЫВ ПОСЛЕ ЛЕКЦИИ

mail

БлогиЛюдиПрограммаВакансииРасписание

python

| | | | | | |
|----------------|----------------|----------------------------------|----------------|----------------|----------------|
| сб, 16 октября | вс, 17 октября | пн, 18 октября | вт, 19 октября | ср, 20 октября | чт, 21 октября |
| Занятий нет | Занятий нет | 18:00 Back-end разработка ... | Занятий нет | Занятий нет | Занятий нет |

Backend разработка на Python

Привет!
Это блог курса Backend разработка на Python.
Все занятия проходят в зуме согласно расписанию, по ссылке:
<https://mailru.zoom.us/j/96845327537?pwd=SkFxQ0FmVXowQnR4dlh2eWM3ZmZmRdz09>

Записи:
0 Вебинар. Организационное собрание. - [ссылка](#) (нужно смотреть/скачать через облако mail)

82 читателя, 3 топика

ПодписатьсяСоздать топик

Поиск по авторам, заголовку и тексту топика...Найти

Материалы к первой лекции

Backend разработка на PythonСмешанное занятие 1

Прямой эфир

МоиВсе

Сергей Шаленко 2 дня назад
[Лекция 1. Знакомство. Введение в Linux. Работа с файлами. Просмотр ресурсов сервера.](#) 1

Сергей Шаленко 3 дня назад
[Linux + Лекция 1. Знакомство. Введение в Linux. Работа с файлами. Просмотр ресурсов сервера.](#) 1

Сергей Шаленко 3 дня назад
[Linux + Добро пожаловать на борту!](#) 0

Артур Сардарян 3 дня назад
[Разработка приложений на iOS | Осень 2021 + Рубежный контроль](#) 1 0

Константин Ермаков 3 дня назад
[Автоматизированное тестирование | Осень 2021 + Итоги 4 лекции \(семинар\)](#) 0

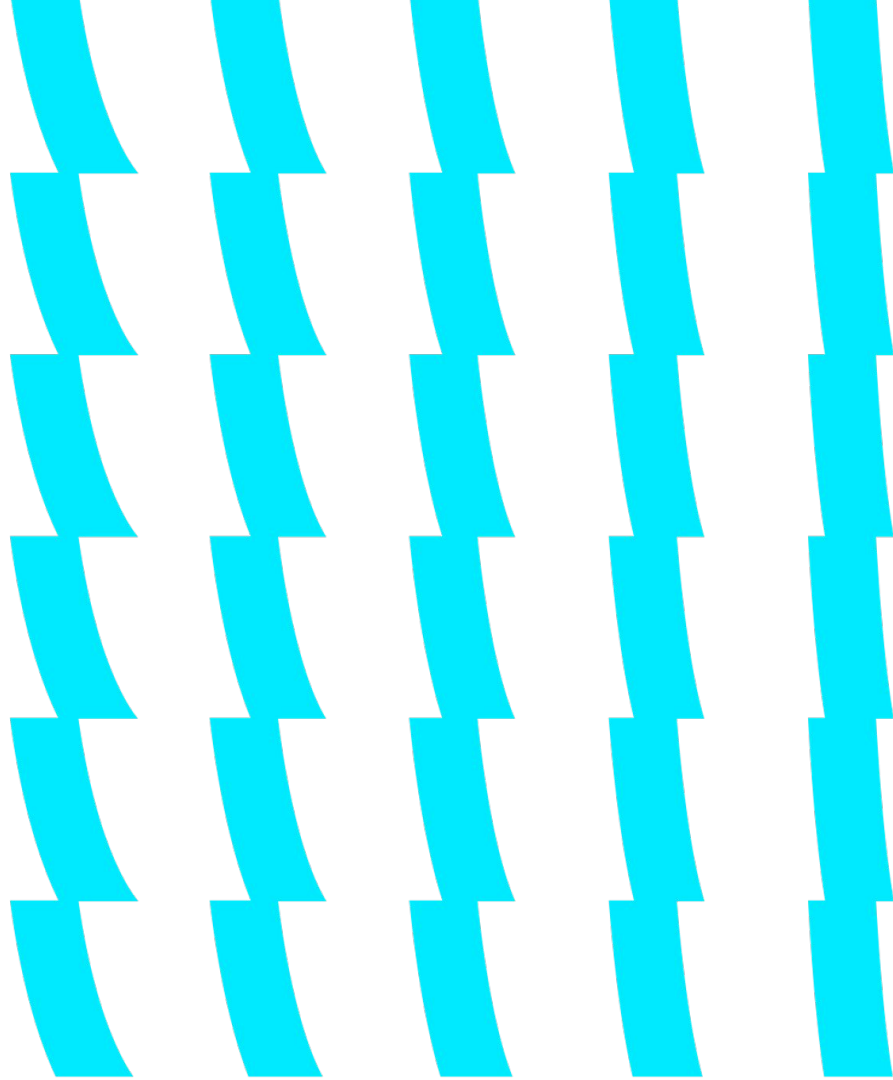
Квиз про прошлой лекции



Содержание занятия

1. Числа
2. Строки
3. collections
4. functools
5. itertools
6. heapq
7. socket
8. Файлы и каталоги

Числа



int, round

```
>>> round(6.5)  # ???
```

```
>>> round(6.6)  # ???
```

```
>>> round(-6.5)  # ???
```

```
>>> round(-6.6)  # ???
```

```
>>> int(6.5)  # ???
```

```
>>> int(6.6)  # ???
```

```
>>> int(-6.5)  # ???
```

```
>>> int(-6.6)  # ???
```

```
>>> int("ffff", 16), int("42", 7)
(65535, 30)
```

float

```
>>> float("-inf"), float("inf"), float("nan")
```

```
(-inf, inf, nan)
```

```
>>> 0.1 + 0.2 == 0.3
```

```
False
```

```
>>> 0.1 + 0.2 <= 0.3
```

```
False
```

```
>>> 0.1 + 0.2
```

```
0.30000000000000004
```

```
>>> (0.1).as_integer_ratio()
```

```
(3602879701896397, 36028797018963968)
```

```
>>> format(0.1, ".25f")
```

```
'0.1000000000000000055511151'
```

```
>>> math.isclose(0.1 + 0.2, 0.3)
```

```
True
```

Decimal, Fraction

```
>>> from decimal import Decimal
```

```
>>> Decimal("0.1") + Decimal("0.2") == Decimal("0.3")
```

```
True
```

```
>>> Decimal(1) / Decimal(7)
```

```
Decimal('0.1428571428571428571428571429')
```

```
>>> from fractions import Fraction
```

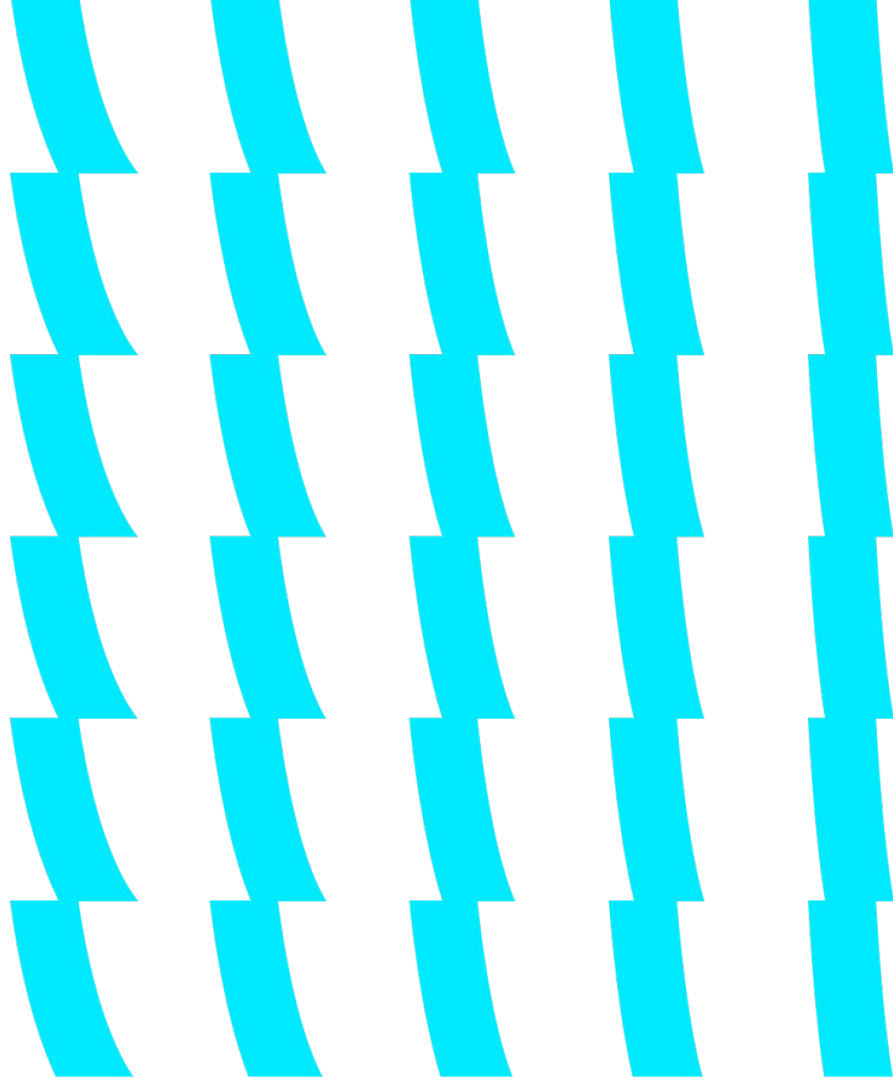
```
>>> Fraction(1, 10)
```

```
Fraction(1, 10)
```

```
>>> Fraction(1, 10) + Fraction(2, 10) == Fraction(3, 10)
```

```
True
```


Строки



str

- `isalpha()`
- `isascii()`
- `isidentifier()`
- `isalnum()`
- `isdecimal()`
- `isdigit()`
- `isnumeric()`

```
>>> s = "1122344"
```

```
>>> s.isalnum() # ???
```

```
>>> s.isdigit() # ???
```

```
>>> s.isnumeric() # ???
```

```
>>> s.isdecimal() # ???
```

str

- `startswith(prefix[, start[, end]])`
- `endswith(suffix[, start[, end]])`
- `capitalize()`
- `upper()`
- `isupper()`
- `lower()`
- `islower()`
- `title()`
- `istitle()`
- `swapcase()`
- `isprintable()`
- `isspace()`

str

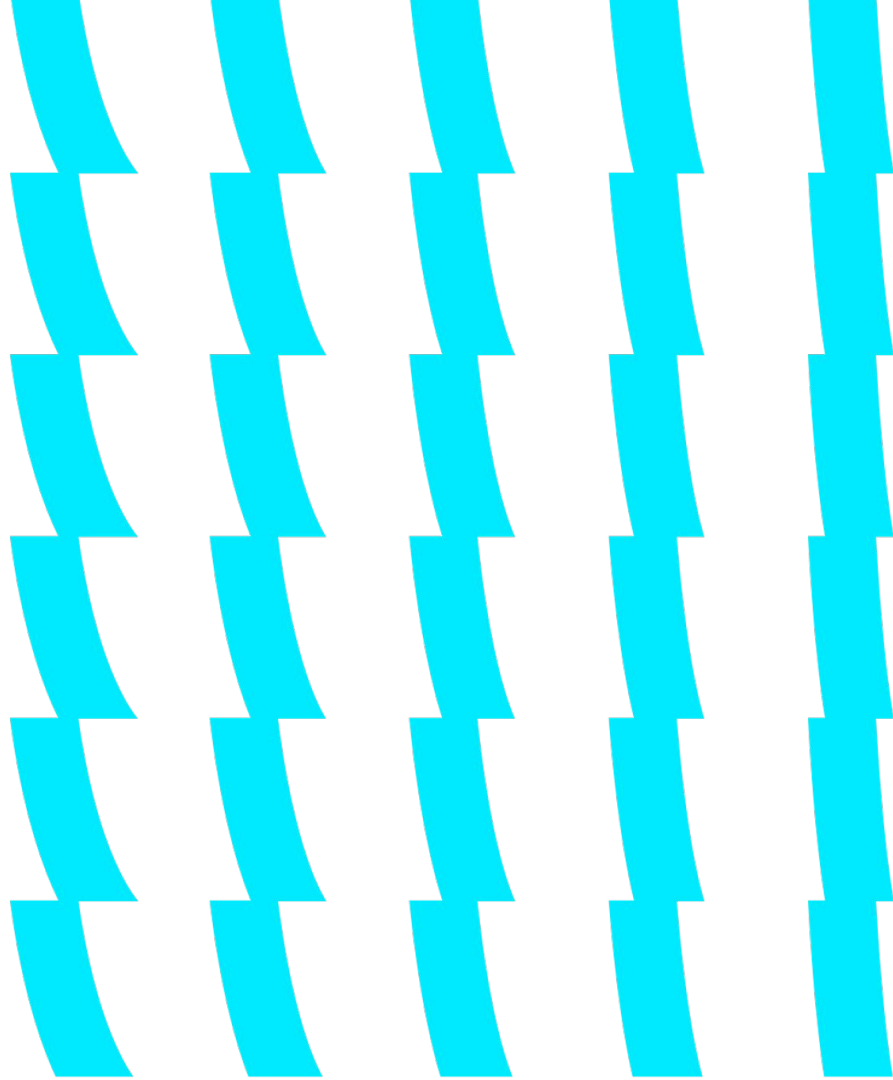
- `count(sub[, start[, end]])`
- `find(sub[, start[, end]]), rfind(sub[, start[, end]])`
- `index(sub[, start[, end]]), rindex(sub[, start[, end]])`
- `replace(old, new[, count])`
- `center(width[, fillchar])`
- `encode(encoding='utf-8', errors='strict')`
- `expandtabs(tabsize=8)`
- `format(*args, **kwargs)`
- `ljust(width[, fillchar]), rjust(width[, fillchar])`
- `lstrip([chars]), strip([chars]), rstrip([chars])`

str

- `split(sep=None, maxsplit=-1)`, `rsplit(sep=None, maxsplit=-1)`
- `splitlines(keepends=False)`
- `partition(sep)`, `rpartition(sep)`
- `join(iterable)`
- `zfill(width)`
- `removeprefix(prefix, /)`
- `removesuffix(suffix, /)`

collections

Специализированные контейнерные
типы данных, предоставляющие
альтернативы для встроенных
`dict`, `list`, `set` и `tuple`



collections.namedtuple

`namedtuple(typename, field_names, *, rename=False, defaults=None, module=None)`

```
>>> Point = collections.namedtuple("Point", ["x", "y"])
```

```
>>> p = Point(11, y=22)  # p = (11, 22)
```

```
>>> p[0] + p[1]
```

```
33
```

```
>>> x, y = p
```

```
>>> x, y
```

```
(11, 22)
```

```
>>> p.x + p.y
```

```
33
```

```
>>> p._asdict()  # {'x': 1, 'y': 4}
```

collections.defaultdict

`collections.defaultdict([default_factory[, ...]])`

Словарь, у которого по умолчанию всегда вызывается функция **default_factory**.

```
>>> import collections
```

```
>>> defdict = collections.defaultdict(list)
```

```
>>> print(defdict)
```

```
defaultdict(<class 'list'>, {})
```

```
>>> for i in range(5):
```

```
...     defdict[i].append(i)
```

```
>>> print(defdict)
```

```
defaultdict(<class 'list'>, {0: [0], 1: [1], 2: [2], 3: [3], 4: [4]})
```


collections.OrderedDict

`collections.OrderedDict([items])`

Словарь, который помнит порядок, в котором ему были даны ключи.

```
>>> import collections
>>> d = collections.OrderedDict(
...     [("a", "A"), ("b", "B"), ("c", "C")]
... )
>>> for k, v in d.items():
...     print(k, v)
'a', 'A'
'b', 'B'
'c', 'C'
>>> d.move_to_end("b")
```

collections.Counter

`collections.Counter([iterable-or-mapping])`

Словарь для подсчёта хешируемых объектов.

- `elements()`
- `most_common([n])`
- `subtract([iterable-or-mapping])`
- `update([iterable-or-mapping])`

```
>>> words = re.findall(r"\w+", open("hamlet.txt").read().lower())
```

```
>>> Counter(words).most_common(5)
```

```
[('the', 1143), ('and', 966), ('to', 762), ('of', 669), ('i', 631)]
```

collections.deque

`collections.deque([iterable[, maxlen]])`

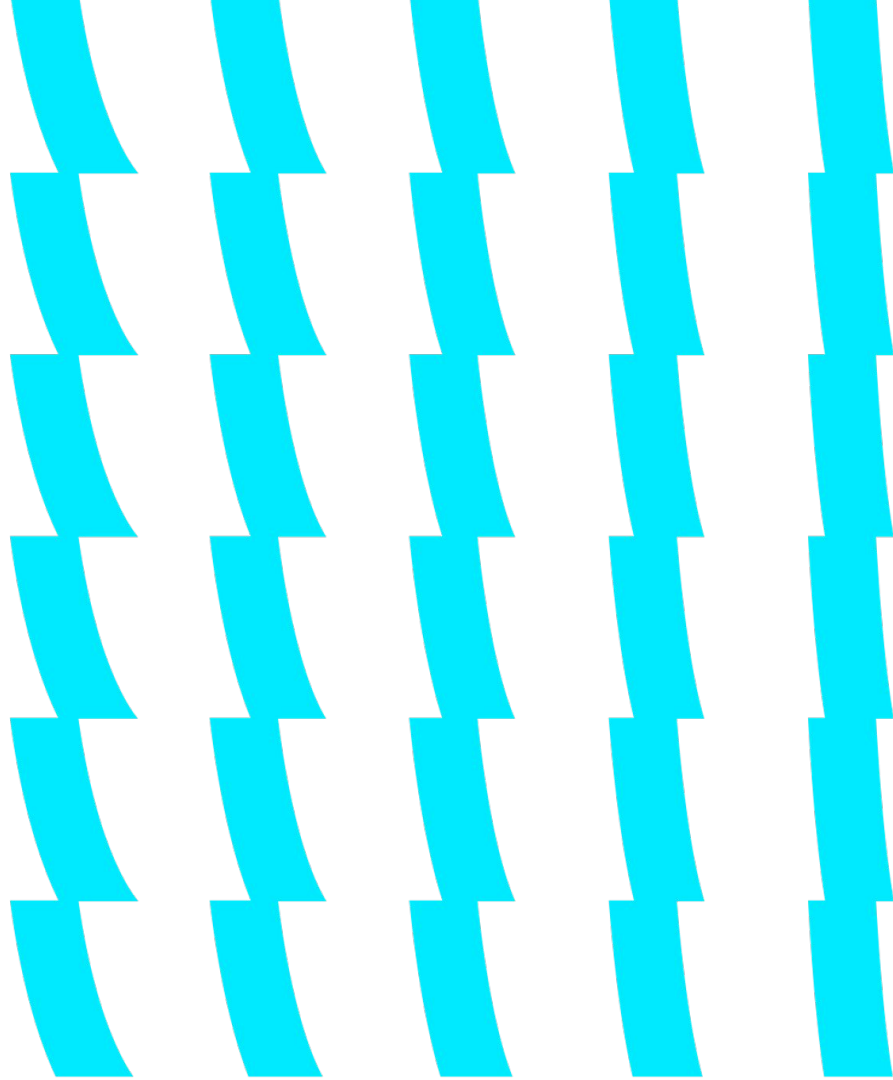
Двусторонняя очередь из итерируемого объекта с максимальной длиной `maxlen`.

Добавление и удаление элементов в начало или конец выполняется за константное время.

- `append(x)`
 - `appendleft(x)`
 - `extend(iterable)`
 - `extendleft(iterable)`
 - `insert(i, x)`
 - `pop()/popleft()`
 - `remove(value)`
- ```
>>> d = deque("ghi")
>>> d.append("j")
>>> d.appendleft("f")
>>> d
deque(['f', 'g', 'h', 'i', 'j'])
>>> d.pop()
'j'
>>> d.popleft()
'f'
>>> d
deque(['g', 'h', 'i'])
```

# functools

Для функций высшего порядка



# functools

- `cache(user_function)`
- `cached_property(func)`
- `lru_cache(user_function)`
- `lru_cache(maxsize=128, typed=False)`

```
@functools.cache
def factorial(n):
 if n <= 1:
 return 1
 return n * factorial(n - 1)
```

## singledispatch, singledispatchmethod(func)

```
@functools.singledispatch
def fun(arg, prefix="Hello"):
 print(f"{prefix} any type {arg}")
```

```
@fun.register(str)
def _(arg, prefix="Hi"):
 print(f"{prefix} str {arg}")
```

```
@fun.register
def _(arg: int, prefix="Hello"):
 print(f"{prefix} int {arg}")
```

```
>>> fun(123) # Hello int 123
```

```
>>> fun.registry.keys()
```

```
dict_keys([<class 'object'>, <class 'str'>, <class 'int'>])
```

# functools

- `partial(func, /, *args, **keywords)`
- `partialmethod(func, /, *args, **keywords)`
- `total_ordering`
- `reduce(function, iterable[, initializer])`
- `wraps(wrapped, assigned=WRAPPER_ASSIGNMENTS, updated=WRAPPER_UPDATES)`
- `update_wrapper(wrapper, wrapped, assigned=WA, updated=WU)`

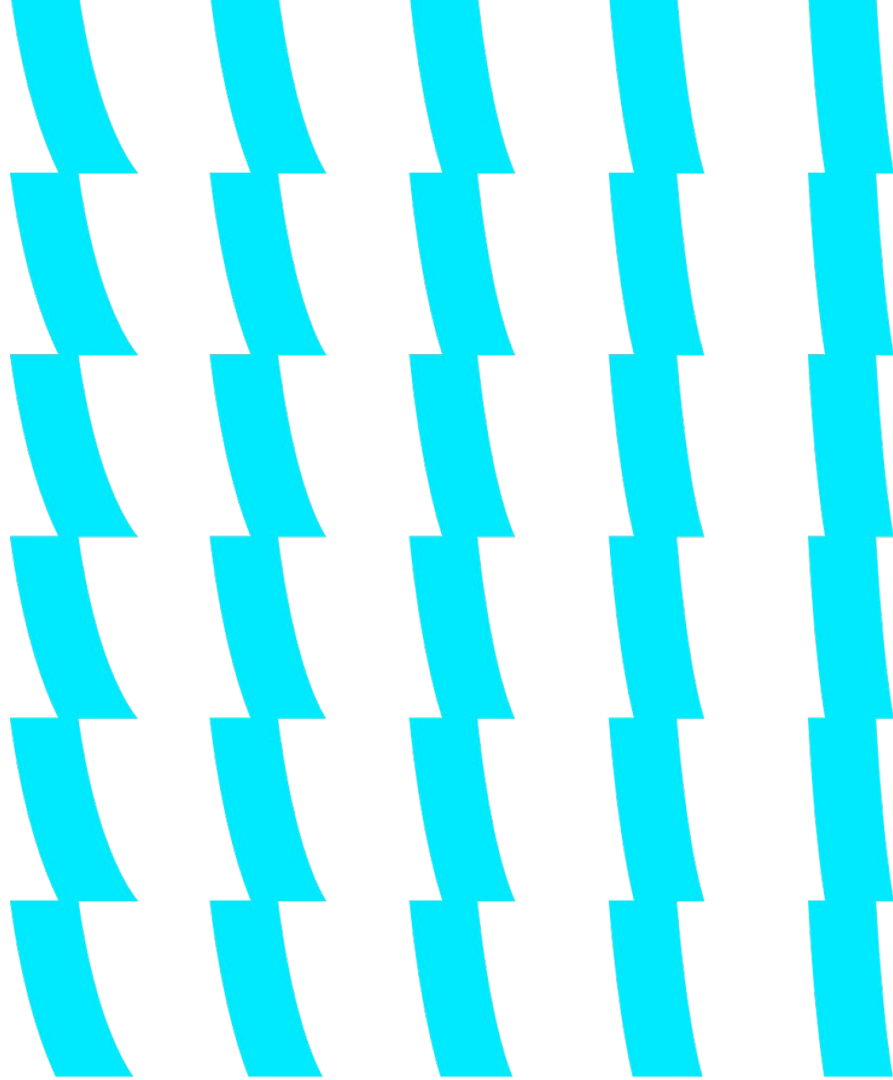
```
>>> basetwo = partial(int, base=2)
```

```
>>> basetwo("10010")
```

```
18
```

# itertools

Можно бесконечно смотреть на  
бесконечный цикл





# Itertools: бесконечные итераторы

- `count(start=0, step=1)`
- `cycle(iterable)`
- `repeat(object[, times])`

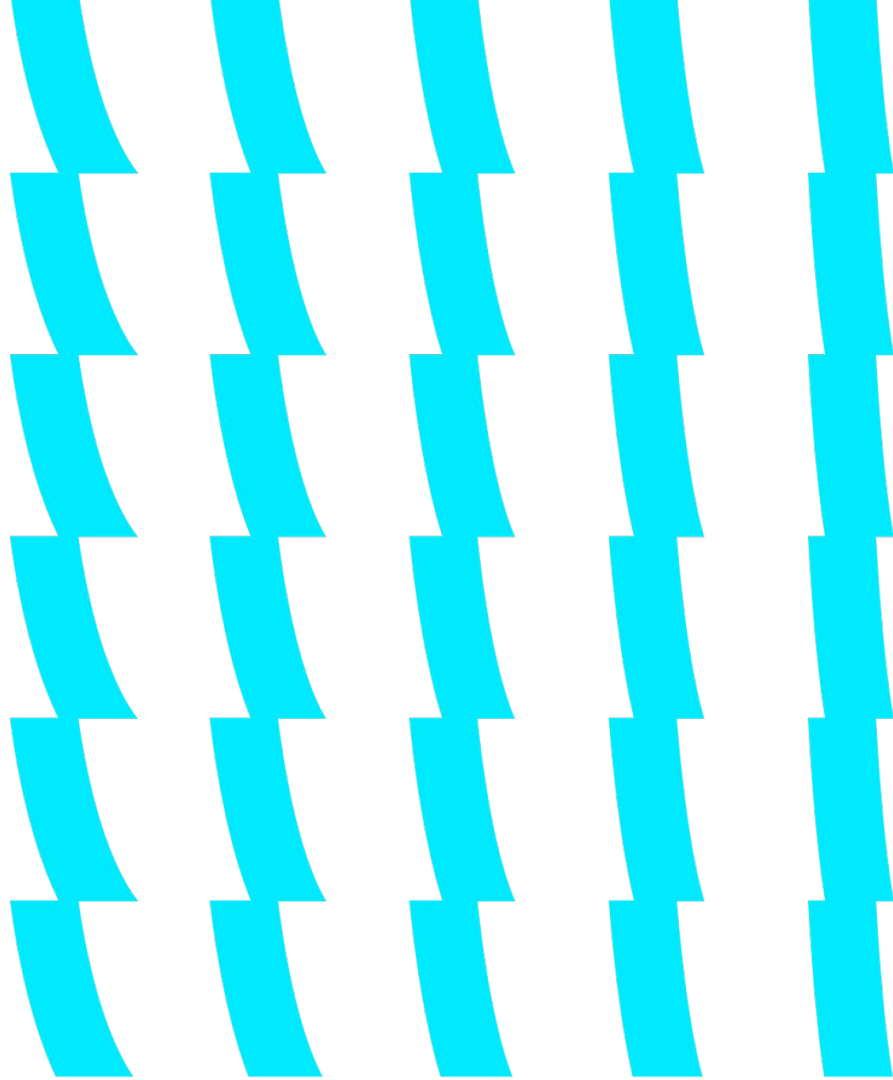
# Itertools

- `accumulate(iterable[, func, *, initial=None])`
- `chain(*iterables)`
- `compress(data, selectors)`
- `dropwhile(predicate, iterable)`
- `takewhile(predicate, iterable)`
- `filterfalse(predicate, iterable)`
- `groupby(iterable, key=None)`
- `islice(iterable, start, stop[, step])`
- `pairwise(iterable)`
- `starmap(function, iterable)`
- `tee(iterable, n=2)`
- `zip_longest(*iterables, fillvalue=None)`

# Itertools: комбинаторика

- `product(*iterables, repeat=1)`
- `permutations(iterable, r=None)`
- `combinations(iterable, r)`
- `combinations_with_replacement(iterable, r)`

**Разное**



# heapq

`heappush(heap, item)`

`heappop(heap)`

`heappushpop(heap, item)`

`heapify(x)`

`heapreplace(heap, item)`

`merge(*iterables, key=None, reverse=False)`

`nlargest(n, iterable, key=None)`

`nsmallest(n, iterable, key=None)`

# Path

```
from pathlib import Path
```

```
p = Path(".")
```

```
pdf_path = p / "storage" / "slides.pdf"
```

```
abs_path = "/usr" / p / "storage" / "slides.pdf"
```

```
p.is_dir(), p.is_file()
```

```
p.is_absolute()
```

```
p.exists()
```

```
p.glob(pat)
```

```
p.open(), p.read_text(), p.write_text()
```

```
p.unlink()
```

## socket: сервер

```
sock = socket.socket()
sock.bind(("", 10001))
sock.listen()
conn, addr = sock.accept()

print("connected:", addr)

while True:
 data = conn.recv(1024)
 if not data:
 break

 conn.send(data.upper())

conn.close()
```

## socket: клиент

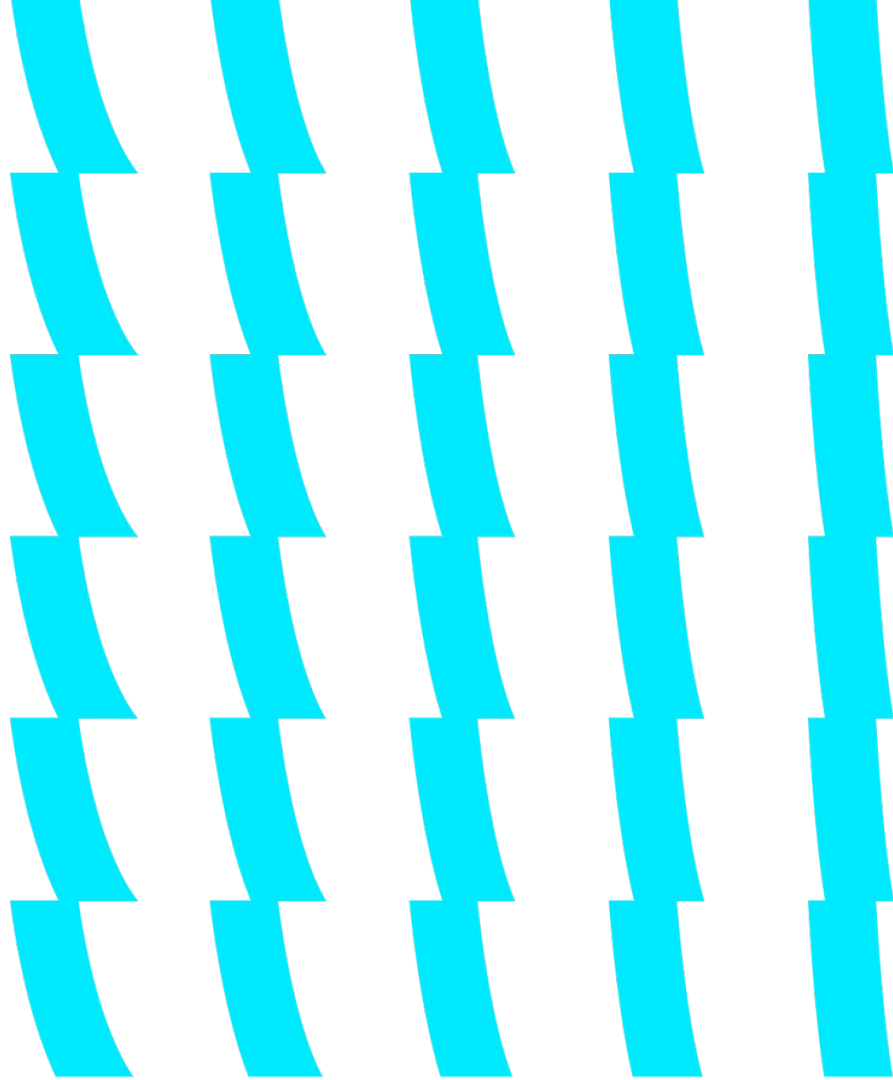
```
sock = socket.socket()
sock.connect(("127.0.0.1", 10001))
sock.sendall("hello, world!".encode("utf-8"))

data = sock.recv(1024)
sock.close()
print(data.decode("utf-8"))
```



## Домашнее задание #5

- LRU cache без OrderedDict
- +тесты
- flake8 + pylint перед сдачей



# Напоминание отметиться на портале Vol 2

+ ОСТАВИТЬ ОТЗЫВ

mail

БлогиЛюдиПрограммаВакансииРасписание

+

Q

🔔

python

|                |                |                                  |                |                |                |
|----------------|----------------|----------------------------------|----------------|----------------|----------------|
| сб, 16 октября | вс, 17 октября | пн, 18 октября                   | вт, 19 октября | ср, 20 октября | чт, 21 октября |
| Занятий нет    | Занятий нет    | 18:00<br>Back-end разработка ... | Занятий нет    | Занятий нет    | Занятий нет    |

## Backend разработка на Python

↓ 0 ↑

**Привет!**  
Это блог курса Backend разработка на Python.  
**Все занятия проходят в зуме согласно расписанию, по ссылке:**  
<https://mailru.zoom.us/j/96845327537?pwd=SkFxQ0FmVXowQnR4dlh2eWM3ZmZ Rdz09>

**Записи:**  
**0 Вебинар.** Организационное собрание. - [ссылка](#) (нужно смотреть/скачать через облако mail)

82 читателя, 3 топика

ПодписатьсяСоздать топик

Поиск по авторам, заголовку и тексту топика...

Найти

### Материалы к первой лекции

Backend разработка на PythonСмешанное занятие 1

### Прямой эфир

МоиВсе

Сергей Шаленко 2 дня назад  
[Лекция 1. Знакомство. Введение в Linux. Работа с файлами. Просмотр ресурсов сервера.](#) 💬 1

Сергей Шаленко 3 дня назад  
[Linux + Лекция 1. Знакомство. Введение в Linux. Работа с файлами. Просмотр ресурсов сервера.](#) 💬 1

Сергей Шаленко 3 дня назад  
[Linux + Добро пожаловать на борт!](#) 💬 0

Артур Сардарян 3 дня назад  
[Разработка приложений на iOS | Осень 2021 + Рубежный контроль](#) 1 💬 0

Константин Ермаков 3 дня назад  
[Автоматизированное тестирование | Осень 2021 + Итоги 4 лекции \(семинар\)](#) 💬 0

Спасибо за  
внимание



образование