

Контекстные менеджеры

```
In [1]: with open('access_log.log', 'a') as f:
        f.write('New Access')
```

```
In [2]: class open_file:
        def __init__(self, filename, mode):
            self.f = open(filename, mode)

        def __enter__(self):
            return self.f

        def __exit__(self, *args):
            self.f.close()
```

```
In [3]: with open_file('test.log', 'w') as f:
        f.write('Inside `open_file` context manager')
```

```
In [4]: with open_file('test.log', 'r') as f:
        print(f.readlines())
```

```
['Inside `open_file` context manager']
```

```
In [5]: class suppress_exception:
        def __init__(self, exc_type):
            self.exc_type = exc_type

        def __enter__(self):
            return

        def __exit__(self, exc_type, exc_value, traceback):
            if exc_type == self.exc_type:
                print('Nothing happend.')
                return True
```

```
In [6]: with suppress_exception(ZeroDivisionError):
        really_big_number = 1 / 0
```

```
Nothing happend.
```

```
In [7]: import contextlib

        with contextlib.suppress(ValueError):
            raise ValueError
```