Занятие 9

Приемы работы с функциями.

```
def safe_div(x, y):
    """Do a safe division :-)
for fun and profit"""
    if y != 0:
        z = x / y
        print z
        return z
    else:
        print "Motherf___er!"
print safe_div.__doc__
```

```
>>> from ftplib import FTP
>>> print FTP.__doc__
```

Приемы работы с функциями.

```
x = 50

def func():
    global x
    x = 2


def say(message, times = 1):
    print(message * times)

say(‘Hello’)
say('world', 5)


def func(a, b=5, c=10):
    print('a равно', a, ', b равно', b, ', а с равно', c)
```

Приемы работы с функциями.

```
def total(initial=5, *numbers, **keywords):
    count = initial
    for number in numbers:
        count += number
    for key in keywords:
        count += keywords[key]
    return count

print(total(10, 1, 2, 3, vegetables=50, fruits=100))
```

Django

```
def dispatch(self, *args, **kwargs):

    return super(LoginView, self).dispatch(*args, **kwargs)
```

Лямбда - функции. map, filter, reduce.

```
f = lambda x: x + 1
f = function(x) { x + 1 }
mlist = filter(lambda x: x % 3 == 0, [1, 2, 3, 4, 5, 6, 7, 8, 9])
>> [3, 6, 9]
mlist = map(lambda x: x * 3, [1, 2, 3, 4, 5, 6, 7, 8, 9])
>>[3, 6, 9, 12, 15, 18, 21, 24, 27]
result = reduce(lambda x, y: x+y, [1, 2, 3, 4, 5])
>> 15
```

Установка и мониторинг cerery, redis.

Redis is an open source, BSD licensed, advanced **key-value cache** and **store**. It is often referred to as a **data structure server** since keys can contain strings, hashes, lists, sets, sorted sets, bitmaps and hyperloglogs.

Celery is an asynchronous task queue/job queue based on distributed message passing.

It is focused on real-time operation, but supports scheduling as well.

Celery is used in production systems to process millions of tasks a day.

```
sudo apt-get install redis

pip install celery==3.1.11

pip install django-celery==3.1.10

INSTALLED_APPS = ( .. 'djcelery' .. )


import djcelery

djcelery.setup_loader()

BROKER_URL = 'redis://localhost:6379/0'

./manage.py celeryd
```

Отложенные задачи

tasks.py

```
from celery import task
from celery.utils.log import get_task_logger
logger = get_task_logger(__name__)
logger.setLevel('DEBUG')

@task(name='save_message')
def save_message(message):
        logger.info('Save message...')
        message.save()
        logger.info('Success!')

from main.tasks import save_chat_message
save_chat_message.delay(message)
```