

# Let's build something with MongoDB Recreating PyPI

Michael Kennedy

Host, Talk Python To Me podcast



# Wintellect Core Services



Custom software application development and architecture



Microsoft's #1 training vendor for over 14 years having trained more than 50,000 Microsoft developers



On-Demand Training  
World class, subscription based online training

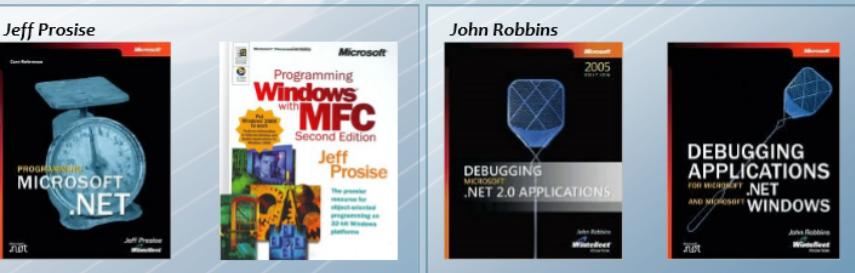


Gold Cloud Platform  
Silver DevOps  
Silver Application Development



# Industry Influencers

We wrote the book (over 30 of them)



We help  
companies  
build better  
software,  
faster.

# Some Microsoft Related Highlights

Gold Azure Partner

2016 IAMCP Gold Partner of the Year for the U.S. announced at WPC

CEO is the Microsoft Regional Director (RD) for Atlanta

DevOps competency partner

Multiple ALM Rangers

Software Development competency partner

Xamarin Premier Consulting Partner

Multiple Xamarin Certified Engineers

Chosen to teach the 2-day Xamarin University pre-con at Evolve 2016

Other: Visual Studio Integration Partner, Azure Circle Partner, ALM Inner Circle Partner, MVP of the Year, and more...

# FAQ

Can I get the code from this webcast?

Yes: <https://github.com/Wintellect/WintellectWebinars>

Can I get the slides?

Yes: <https://github.com/Wintellect/WintellectWebinars>

Will the webinar be recorded?

Yes

Where do I go to get more afterwards?

2 live, virtual workshops running soon

# Take the code with you

The screenshot shows a GitHub repository page for the user 'Wintellect' named 'WintellectWebinars'. The repository has 13 watchers, 24 stars, and 13 forks. The 'Code' tab is selected, showing 36 commits, 1 branch, 0 releases, 5 contributors, and Apache-2.0 license. The latest commit was on Jan 14. A list of commits is provided below:

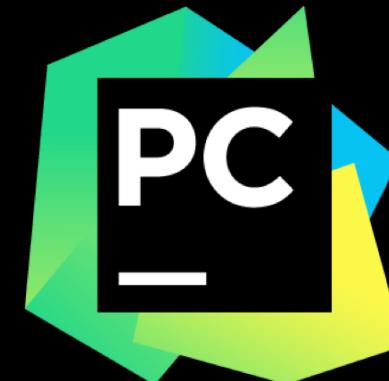
Commit	Description	Date
2017-04-06 - Pythonic Code Through 5 ...	Chained generators.	11 months ago
2017-04-20-python-for-dotnet-kennedy	Wintellect Pyramid web app	10 months ago
2017-05-11 Survey of the Azure Data Pla...	Ike Ellis 2017-05-11	9 months ago
2017-06-08 ASP.NET Core with Docker	Typos fixed	9 months ago
2017-07-06_Angular_Redux	added angular redux webinar files	8 months ago
2017-07-20_Intro_to_Angular_4	added intro to angular 4 webinar	7 months ago
2017-11-02_Concurrent_Programming_in...	Demos	4 months ago

<https://github.com/Wintellect/WintellectWebinars>

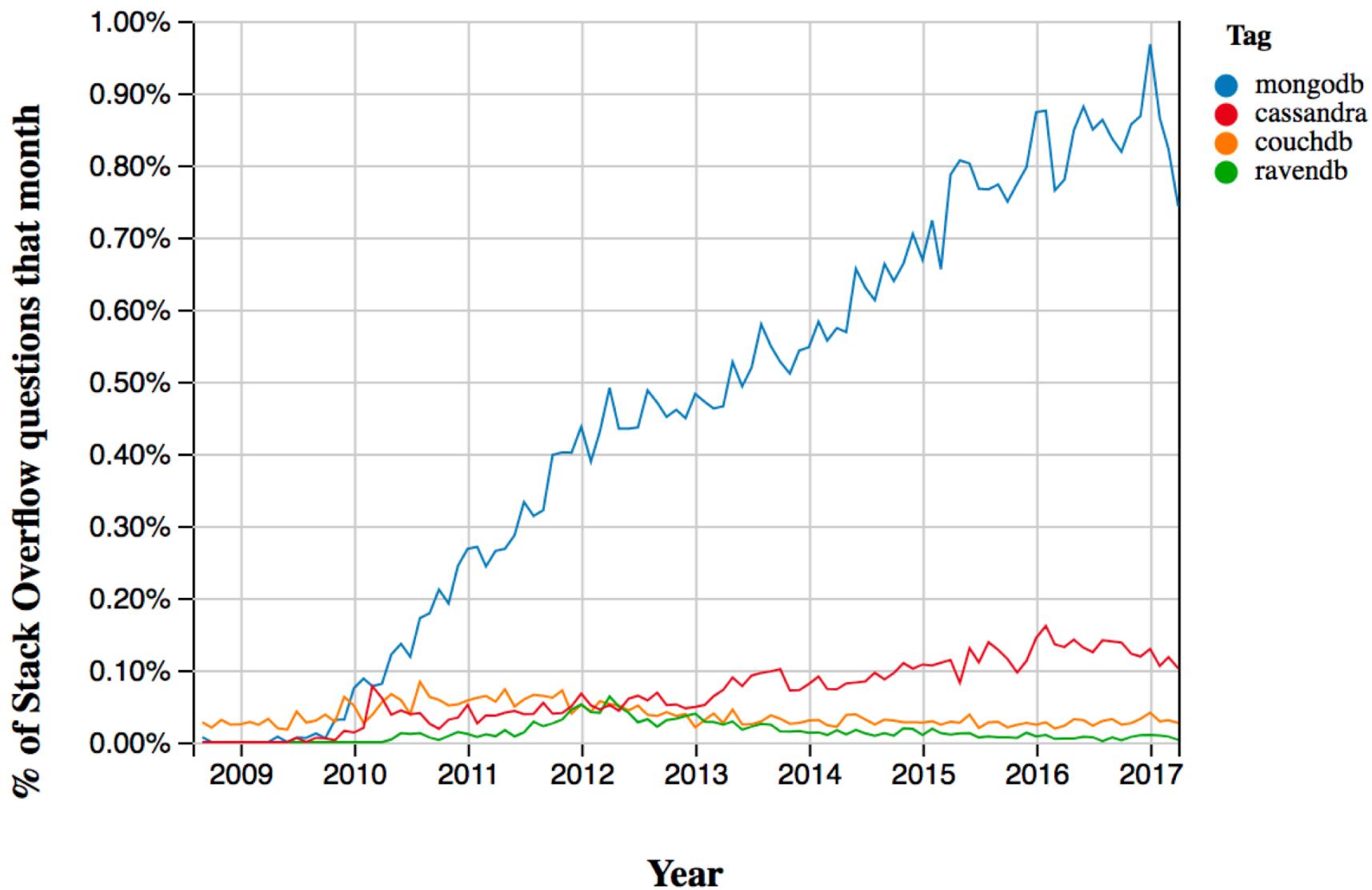
# Topics

- Tools
- Why MongoDB?
- Building something fun!
- Concepts
- Questions

# Tools



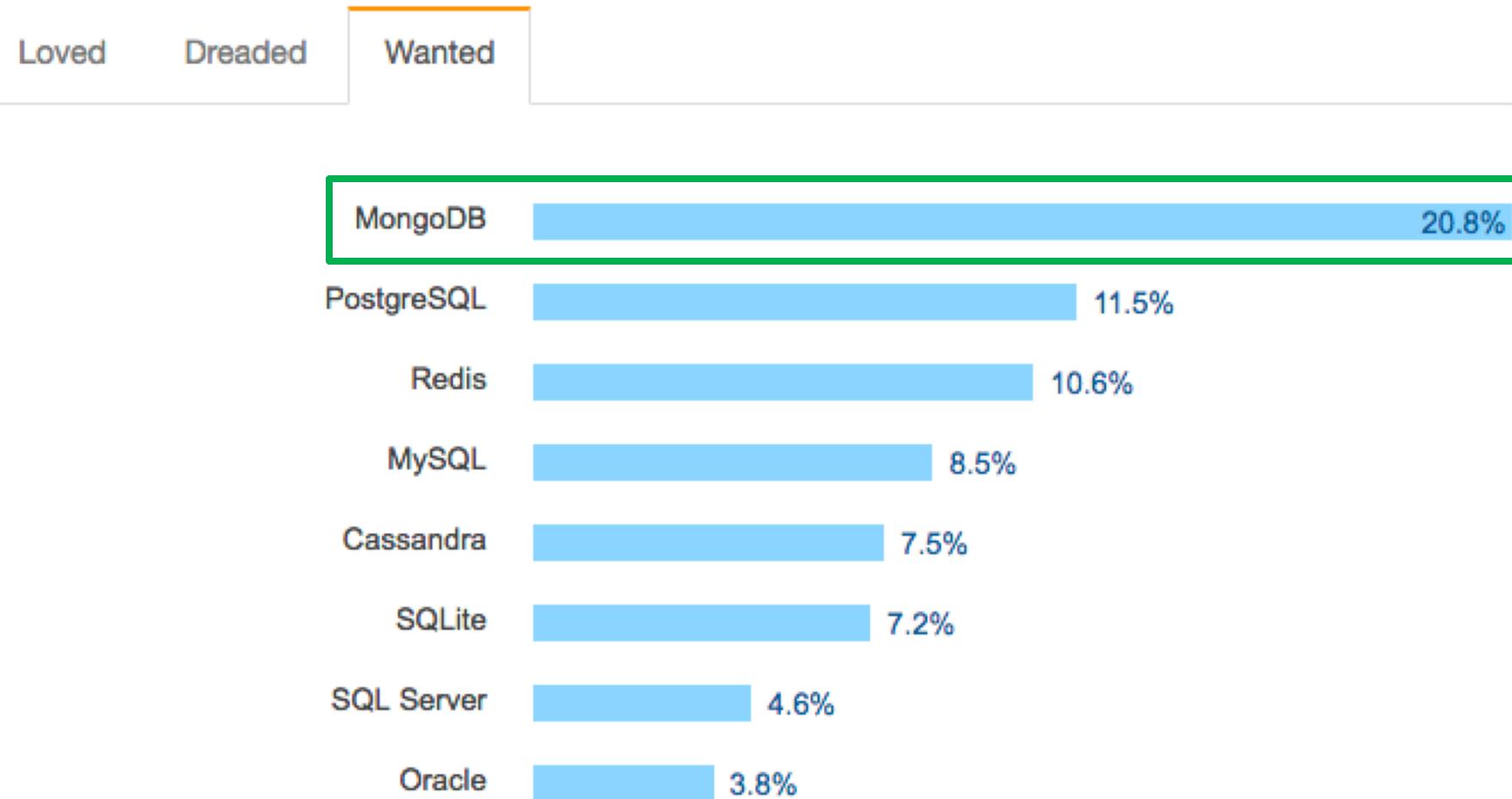
# MongoDB is popular



<https://insights.stackoverflow.com/trends?tags=mongodb%2Ccouchdb%2Cravendb%2Cassandra>

# MongoDB is wanted

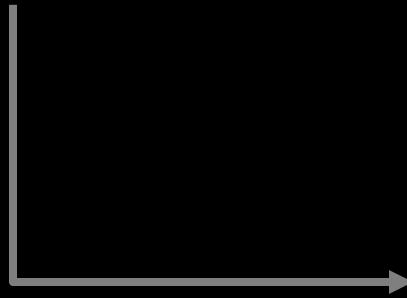
## Most Loved, Dreaded, and Wanted Databases



*% of developers who are not developing with the language or technology but have expressed interest in developing with it*

# How do document databases work?

Consider lectures to be a pre-computed **join**.



Question: Will I be able to find lecture 10106 directly?

```
// Chapter 1001 from Python Jumpstart by Building 10 Apps
{
    "_id" : 1001,
    "title" : "Welcome to the course",
    "course_id" : 1,
    "duration_in_sec" : 1707,
    "lectures" : [
        {
            "id" : 10101,
            "title" : "Welcome and thanks for coming",
            "video_url" : "https://s3.amazonaws.com/.../00-00-welcome.mp4",
            "duration_in_sec" : 380
        },
        // ...
        {
            "id" : 10106,
            "title" : "Linux: Installing Python and PyCharm",
            "video_url" : "https://s3.amazonaws.com/.../00-05-linux-setup.mp4",
            "duration_in_sec" : 307
        }
    ]
}
```

# Our demo application

A screenshot of a web browser showing the PyPI project page for mongoengine 0.15.0. The page has a blue header with the PyPI logo, a search bar, and navigation links for Help, Donate, Log In, and Register. The main title is "mongoengine 0.15.0". Below it is a button with the command "pip install mongoengine" and a QR code. To the right is a green button labeled "Latest Version" with a checkmark icon. The date "Nov 6, 2017" is also visible. The page content includes a brief description: "MongoEngine is a Python Object-Document Mapper for working with MongoDB." A sidebar on the left contains sections for Navigation (Project Description, Release History, Download Files), Project Links (Homepage, Download), and Meta. The "Project Description" section is currently selected. The "About" section provides documentation links and supported MongoDB versions information. The "Installation" section recommends using virtualenv and pip.

mongoengine - Warehouse (Py)

Secure | <https://pypi.org/project/mongoengine/#description>

Michael

Search Projects

Help    Donate    Log In    Register

**mongoengine 0.15.0**

`pip install mongoengine`

Latest Version

Nov 6, 2017

MongoEngine is a Python Object-Document Mapper for working with MongoDB.

**Navigation**

- [Project Description](#)
- [Release History](#)
- [Download Files](#)

---

**Project Links**

- [Homepage](#)
- [Download](#)

---

**Meta**

**About**

MongoEngine is a Python Object-Document Mapper for working with MongoDB. Documentation is available at <https://mongoengine-odm.readthedocs.io> - there is currently a [tutorial](#), a [user guide](#), and an [API reference](#).

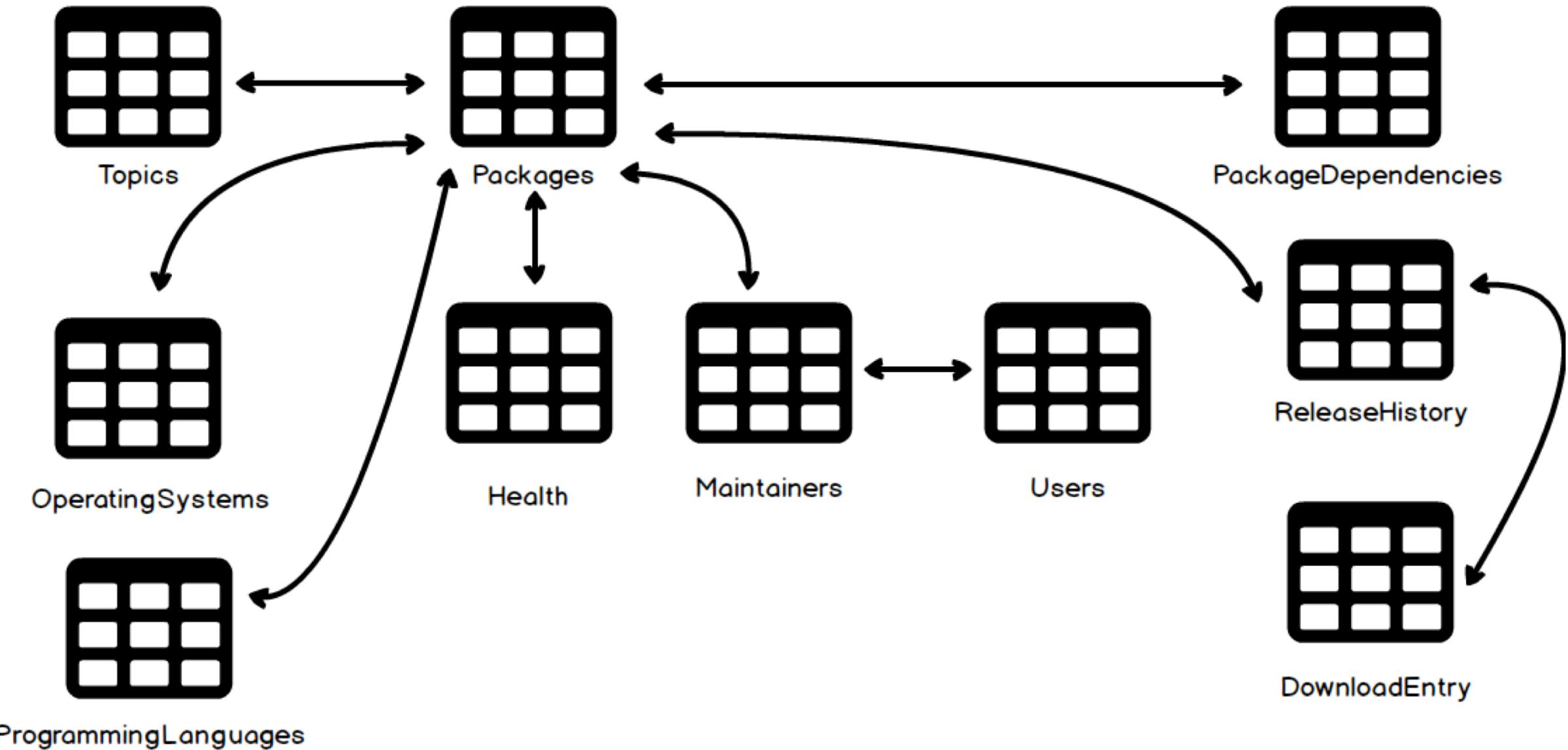
**Supported MongoDB Versions**

MongoEngine is currently tested against MongoDB v2.4, v2.6, and v3.0. Future versions should be supported as well, but aren't actively tested at the moment. Make sure to open an issue or submit a pull request if you experience any problems with MongoDB v3.2+.

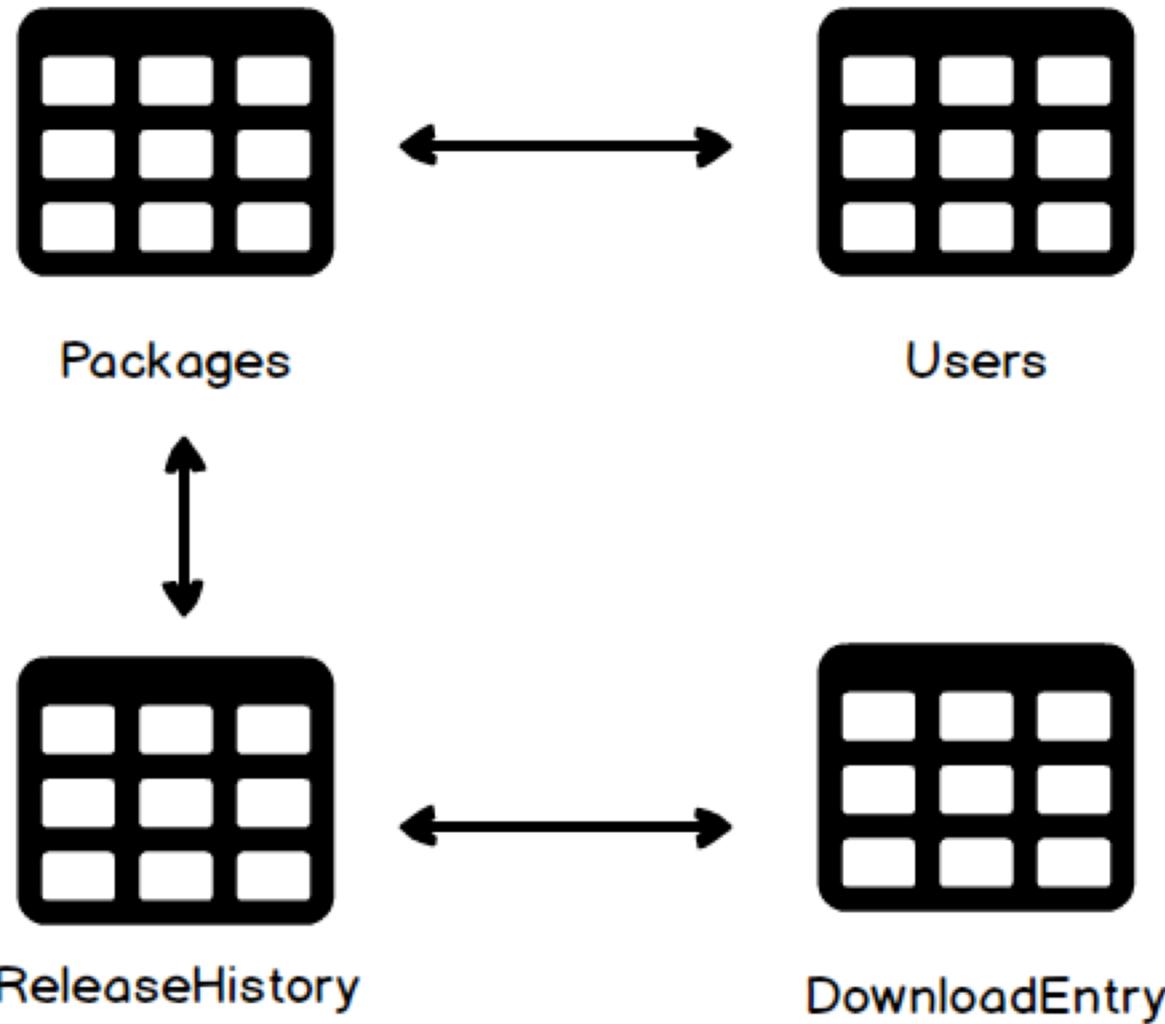
**Installation**

We recommend the use of [virtualenv](#) and of [pip](#). You can then use `pip install -U mongoengine`. You may

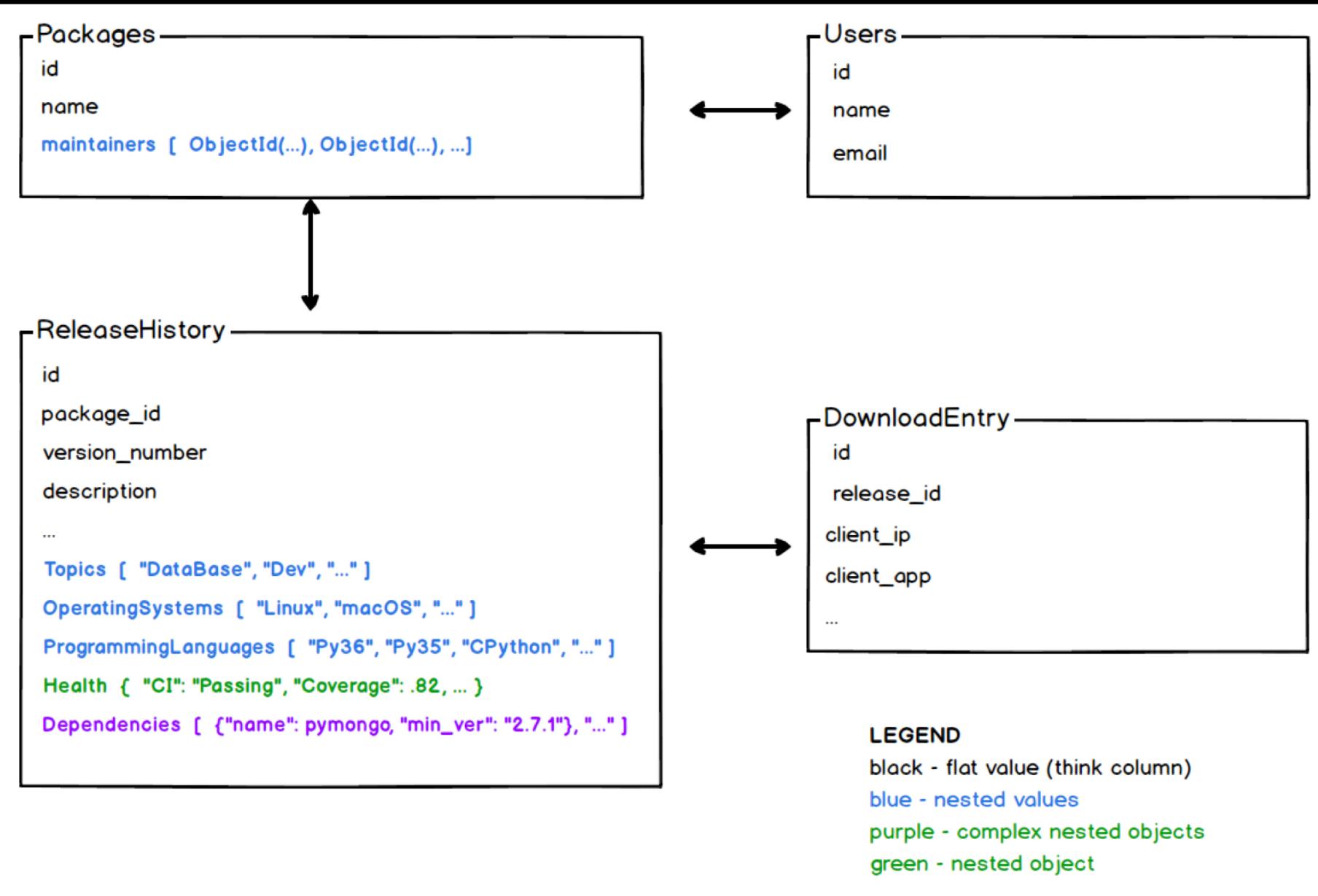
# Relational model



# Document DB model



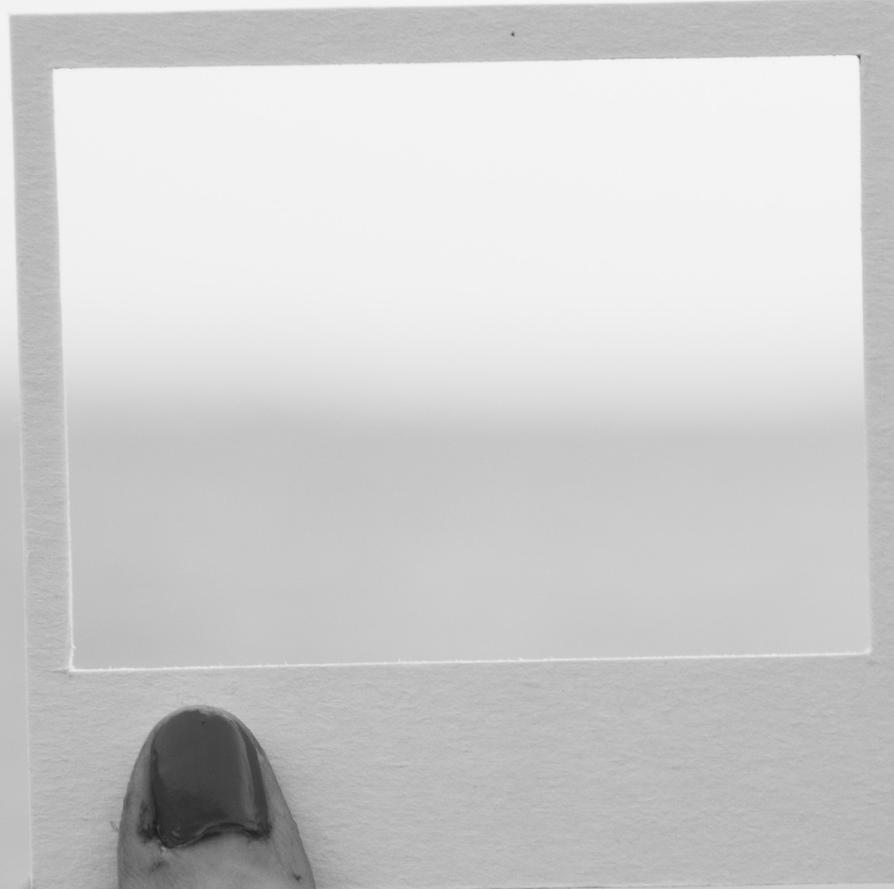
# Document DB model (detailed)



# Demo time



# Concepts



The primary question becomes

## To embed or not to embed?

1. Is the embedded data wanted **80% of the time?**
2. How often do you want the embedded data **without the containing document?**
3. Is the embedded data **a bounded set?**
4. Is that bound **small?**
5. **How varied** are your queries?
6. Is this an **integration DB** or an **application DB?**

# Registering connections

```
import mongoengine

alias_core = 'core'
db = 'snakebnb'

mongoengine.register_connection(alias=alias_core, name=db)
```

# Creating basic classes

```
import mongoengine

class Snake(mongoengine.Document):
    registered_date = mongoengine.DateTimeField()
    length = mongoengine.FloatField()
    name = mongoengine.StringField()
    species = mongoengine.StringField()
    is_venomous = mongoengine.BooleanField()
```

# Nested data (fields and lists)

```
import mongoengine

class Cage(mongoengine.Document):
    name = mongoengine.StringField(required=True)
    price = mongoengine.FloatField(required=True)
    square_meters = mongoengine.FloatField(required=True)

    bookings = mongoengine.EmbeddedDocumentListField()

    def __str__(self):
        return self.name

    def __repr__(self):
        return "<{}: {}>".format(self.__class__.__name__, self.name)

    def to_json(self):
        return {
            "_id" : ObjectId("59d3951c3ae740cfffe16427d"),
            "registered_date" : ISODate("2017-10-03"),
            "name" : "Slither's first cage",
            "price" : 29.95,
            "square_meters" : 1.5,
            "is_carpeted" : true,
            "has_toys" : false,
            "allow_dangerous_snakes" : false,
            "bookings" : [
                {
                    "check_in_date" : ISODate("2018-02-19"),
                    "check_out_date" : ISODate("2018-02-28"),
                    "rating" : 0
                },
                {
                    "check_in_date" : ISODate("2018-03-01"),
                    "check_out_date" : ISODate("2018-03-30"),
                    "rating" : 0
                }
            ]
        }
```

# Querying (direct match)

```
def find_owner_by_email(email: str) -> Owner:  
  
    owner = Owner.objects().filter(email=email).first()  
    return owner
```

Filter on 1 or more fields. **email** must match passed **email** value.



Call **first** to execute the query and return 1 owner or None

# Querying (subdocuments)

```
def get_bookings_for_user(user_id: ObjectId) -> List[Booking]:  
    owner = Owner.objects(id=user_id).first()  
    booked_cages = Cage \  
        .objects(bookings_guest_snake_id_in=owner.snake_ids) \  
        .all()  
  
    return list(booked_cages)
```

Use `_` to separate / navigate levels in subdocuments



Best practice: Execute the query before returning results.

# Want more?



MongoDB Quickstart with Python, 2.5 hrs

[freemongodbcourse.com](http://freemongodbcourse.com)