# README Template

## About the Project/Project Title

*The CRUD python file provides a way to implement python commands to create and read the data that is found under the AAC database and the animals collection dataset.*

## Motivation

*This project exists to demonstrate the ways of implementing the basic CRUD functionalities through the use of pymongo in MongoDB.*

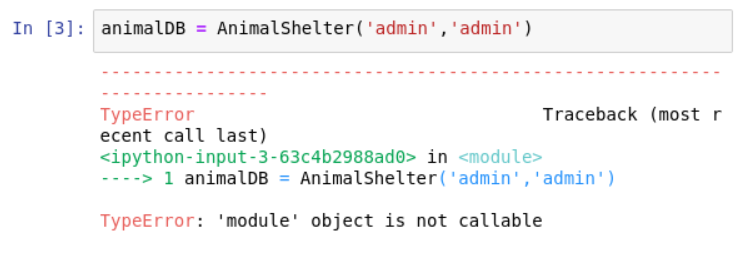
## Getting Started

*To*

*To get this project running first add the following imports (Second import is to view data pretty printed)*

**

Note: If only import AnimalShelter is run then creating a new AnimalShelter object will fail because of the TypeError ‘module’ object is not callable:

**

## Installation

If recreating the file make sure to include the import below to make use of pymongo.  
 ***from pymongo import MongoClient***

Simply download the file and put the file into the directory where your code is being ran from or follow instructions from the site below to create package into your site-packages directory and import from any directory

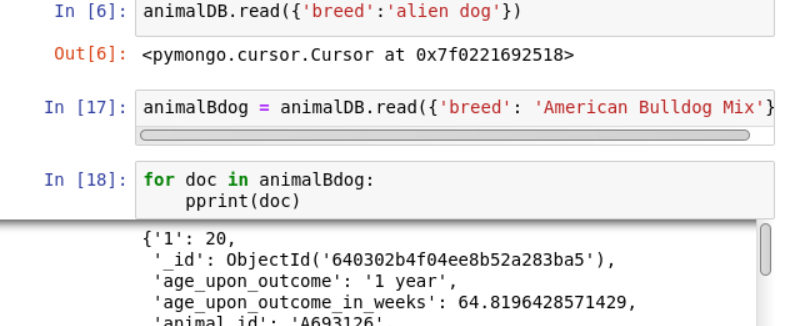
<https://packaging.python.org/en/latest/tutorials/installing-packages/>

## Usage

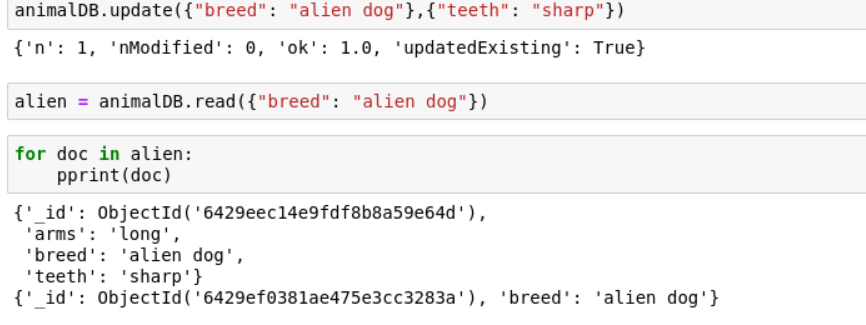
***This python module can be used to create and read data from a database.***

Example:





***Calling the update function is also available using .update(data\_to\_update, new\_data) function***

**

***Finally deletion of data can be simply processed through accessing the delete function call through the .delete(data\_to\_delete) function.***



### Code Example

*from pymongo import MongoClient*

*from bson.objectid import ObjectId*

*class AnimalShelter(object):*

*def \_\_init\_\_(self,username, password):*

*self.client = MongoClient('mongodb://%s:%s@localhost:48469' % (username,password))*

*self.database = self.client['AAC']*

*def create(self, data):*

*if data is not None and isinstance(data,dict):*

*self.database.animals.insert(data)*

*return True*

*else:*

*raise Exception("Nothing to save, because data is empty or not dict type")*

*return False*

*def read(self, data):*

*if data is not None and isinstance(data,dict):*

*if(self.database.animals.count(data) != 0):*

*return self.database.animals.find(data)*

*else:*

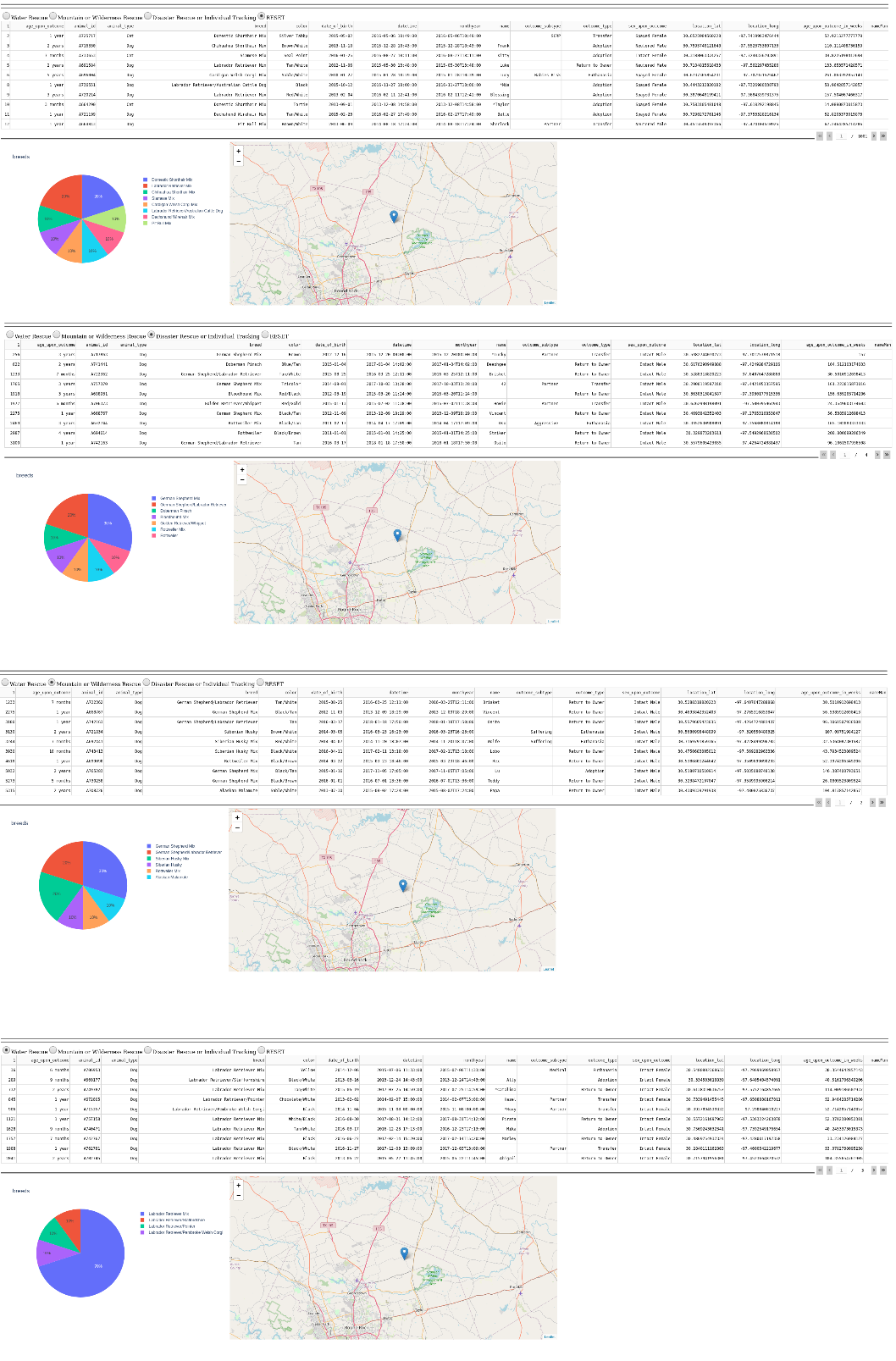
*raise Exception("document not existent in database")*

*else:*

*raise Exception("data is not dictionary data type or is empty")*

**User Interface**

For the user interface you are able to see the breeds at the bottom and the radio button items allow for easy selection of different data displaying the breeds as a pie chart per new dataframe.



### Tests

*As shown above the code for creation and reading is as simple as using the functions Cursor.read(), Cursor.create() and including the data you are searching/creating for in the parameters. Tests can also be run with the using of the Cursor.delete()and Cursor.update() function. Simple calls to these functions can allow for you to make use of CRUD operations to ensure correct data processes are being performed.*

## Roadmap/Features (Optional)

## Contact

Your name: Joshua Flores