



ICSSC Workshop: Build a Reddit Bot in Python

Workshop by Will Castelli



What is Reddit?

- Community based website comprised of “subreddits” dedicated to a particular topic.
- Submissions voted to the “frontpage” of the site.
- Millions of submissions, millions of comments (good for us!).
- Check out [/r/learnpython](https://www.reddit.com/r/learnpython/) for some more information about Python.



What is a bot?

- A bot is an application that runs a script, autonomously.
- “Bots perform tasks that are both simple and structurally repetitive, at a much higher rate than would be possible for a human alone”.
- Can you imagine looking through every comment posted on Reddit for a specific keyword?



Why build bots?

- Can be useful for other Redditors.
- Building a bot is an interesting way to show off your Python skills.
- Companies use bots to get information about users.
- Can you think of any Reddit bots you've seen?

[-] [craftsmany](#) [S] 1 point 7 hours ago

!rolladice

permalink source embed save save-RES report give gold reply hide child comments

[-] [Y3JhZnRzbWFueQ](#) 1 point 7 hours ago

rolling the dice... A "3" was rolled - Dice rolled for [/u/craftsmany](#) .

Beep Boop I'm a Robot. My Owner is [/u/craftsmany](#) please contact him if you have sc

permalink source embed save save-RES parent report give gold reply

[-] [Seventh_Planet](#) 10 points 4 hours ago

Is there an even longer version of this video? What is the context to this?

permalink parent

[-] [TheAdBlockMoose](#) 4 points 3 hours ago

<https://www.youtube.com/watch?v=kujo7V9m0gk>

permalink parent

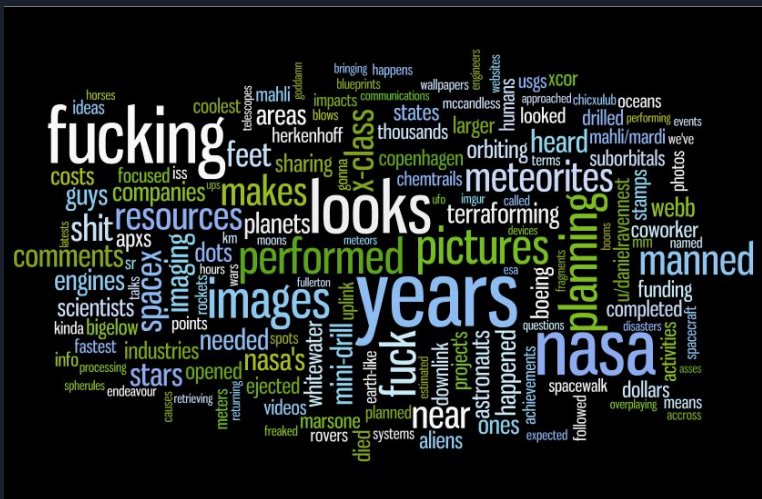
[-] [youtubefactsbot](#) 5 points 3 hours ago

EVEN MY IMMEDIATE FAMILY BOUGHT IT [1:38]

Vince McMahon revealing himself to be the Greater Power (Raw: June 7, 1999).
216,212 views since Mar 2011

[youtubefactsbot](#)

permalink parent



[-] [TheSweetestKill](#) 12 points 6 days ago

During part of the tour at KSC, one of the things they tell you is that the sound alone would kill you within 800 feet. Because it's a behind closed doors tour there isn't any video or anything, but I found an article in an Orlando news paper from when the particular exhibit opened stating the exact same thing.

http://articles.orlandosentinel.com/2007-05-25/news/A2STORY25_1_shuttle-launch-experience-space-launch-launch-speed

Short of you going on the KSC tours and learning it for yourself, that's about all the supporting evidence I can provide right now.

permalink parent report give gold reply buffer

[-] [MetricConversionBot](#) 15 points 6 days ago

800 feet \approx 243.84 meters

*In Development | [FAQ](#) | [WHY](#) *

permalink parent report give gold reply buffer



Rules for bots (“bottiquette”)

- Use common sense (don't use a bot on /r/suicidewatch).
- Don't respond to common words or phrases.
- Don't harass people/groups of people...
- Don't vote with your bot.
- You will get banned if you don't follow the rules.
- Read full rules: <https://www.reddit.com/wiki/bottiquette>



Our goals...



- Create a Reddit bot that replies to comments with a Wikipedia summary.
- Our bot will...
 1. Respond to comments with a keyword.
 2. Respond to comments with a [wikipedia.org](https://www.wikipedia.org) link.


For starters....

- Lets create new Reddit accounts! (You can use your main account, but don't blame me if you get banned for doing something wrong...)
- Remember your username and password.



Registering our bot with Reddit

- Go to <https://reddit.com/prefs/apps/>
- Create a SCRIPT.
- Input a name, description, and an about url.
- Scripts don't use the redirect uri, so just put <http://localhost:8080> and forget about it.



WikiBot
personal use script
MyLR0f0xNdHiqQ

change icon

secret 0HiOXaQG-F0IUIRo8HWMtGECB2E

name

description

about url

redirect uri

Wikipedia Reddit Bot

developers TheZotBot (that's you!) [remove](#)

add developer:

**KEEP THIS
PAGE OPEN!
We need it
soon...**



About Praw



- PRAW (Python Reddit API Wrapper) gives us an easy way to use the API for our bots.
- Abides by all the Reddit API rules so we don't have to worry.
- Read the docs here: <https://praw.readthedocs.io/en/latest/>



About Wikipedia

- Wikipedia is a Python package that gives us easy access to Wikipedia information.
- Much easier than parsing the HTML ourselves (I don't think we have time for that)
- Read the docs:
<https://wikipedia.readthedocs.io/en/latest/code.html#api>



Let's begin...

- Open a Terminal.
- `pip3 install praw`
- `pip3 install wikipedia`

Open your favorite text editor

- Let's start with the basics.
- First, we need connect to Reddit.
- We need to obtain a "Reddit Instance"
- Remember the app registration page? Copy and paste your client_id and client_secret id.
- User agent should be a unique string in the form of
<platform>:<app ID>:<version string> (by /u/<Reddit username>)

```
import praw
import wikipedia

reddit = praw.Reddit(client_id = 'MyLR0f0xNdHiqQ',
                     client_secret = '0HiOXaQG-F0IUIRo8HWMtGECB2E',
                     username = 'TheZotBot',
                     password = 'password',
                     user_agent = 'web:WikiBot:1.0 (by /u/TheZotBot)')
```

Run the
program...make sure it
doesn't crash



Still setting up...

- Create a subreddit instance (you can enter any subreddit...)

```
subreddit = reddit.subreddit("test")
```

- And to get an iterator with posts (specify new, hot, or top, and how many posts)

```
posts = subreddit.new(limit = 5)
```



Starting to get information

- Lets see if it works. Lets print out some information about these posts.

```
for p in posts:  
    print(p.author)  
    print(p.title + '\n')
```

- Hopefully you see the information for your posts. You can go to the subreddit in a browser to verify this information is correct if you can't believe your eyes.



FYI....

- We can do ALOT with posts. We can interact by replying, and get more information.
- If you want to see everything, you can

```
print(dir(post))
```

to see all attributes and methods.

- You can only read the docs as well.



Comments

- Comments are arguably the most interesting piece of data we can get from Reddit.
- We can view the comments in the posts we just viewed (limit = 1 to test)

```
for p in posts:  
    print(p.title)  
    for comment in p.comments:  
        print(comment.body)
```

Notice anything strange?

Getting (pretty much) all comments

- We want to get all the comments, not just parent comments.
- This is a recursive problem....
- Luckily, praw provides a method that will give us all the comments without having to write our own recursive method.

```
for p in posts:  
    print("TITLE: ", p.title)  
    p.comments.replace_more(limit=None)  
    for comment in p.comments.list():  
        print("COMMENT: ", comment.body)
```

Possible errors...





What can we do with this information?

- Now, we have enough information to create a basic bot.
- Remember our goal? Creating a wikipedia bot.
- Test out how Wikipedia works in the shell...

```
>>> import wikipedia
>>> wikipedia.summary("python", sentences = 3)
```

- You should get a string with the first 3 sentences of the wikipedia summary.
- This Wikipedia python package can do a lot more (we are just using it for summary).



Calling the bot ourselves

- Let's start by calling our bot with a custom keyword.
- Go on /r/test and make a comment....



[-] TheZotBot 1 point just now

uciWikiBot: reddit

[permalink](#) [save](#) [edit](#) [disable inbox replies](#) [delete](#) [reply](#)



Replying to comments

- Get the keyword from the body (string manipulation...)
- `comment.reply` with the summary and a message
- Did you see anything? Make sure the subreddit and posts you are looking at are within range...

```
for p in posts:
    p.comments.replace_more(limit=None)
    for comment in p.comments.list():
        if "uciWikiBot:" in comment.body:
            keyword = comment.body.split()[-1]
            comment.reply("**Here is a summary of this article: " +
                           w.summary(keyword, sentences=3))
            print("Comment posted")
```



What about live comments?

- So far, we are looking at past comments.
- We can look at live comments with streams.
- Let's look at live comments on a subreddit (maybe switch to an active subreddit, like /r/politics)
- A lot of comments...possibility for characters that crash.

```
for comment in subreddit.stream.comments():  
    try:  
        print(comment.body)  
    except:  
        pass
```

|



We can respond to our live comments too

- Make a comment on whatever subreddit you are on, and see if you get a reply.

```
for comment in subreddit.stream.comments():
    if "uciWikiBot:" in comment.body:
        keyword = comment.body.split()[-1]
        comment.reply("**Here is a summary of this article:** " +
                       w.summary(keyword, sentences=3))
    print("Comment posted")
```



A more interesting challenge

- Now that we know how to make a bot that replies to a comment, lets reply to wikipedia.org links.
- Find comments with wikipedia in it, and get the link from that comment.
- There is probably a better way to accomplish this.

```
for comment in subreddit.stream.comments():
    if "wikipedia.org" in comment.body:
        split_comment = comment.body.split()
        for word in split_comment:
            if "wikipedia.org" in word:
                link = word
```




The format of a wikipedia link

`https://en.m.wikipedia.org/wiki/Pyramid_scheme`

`https://en.wikipedia.org/wiki/Cringe_comedy`

`https://en.wikipedia.org/wiki/Palestinian_tunnel_warfare_in_the_Gaza_Strip`

`https://en.wikipedia.org/wiki/Aeroflot_Flight_6502`

`https://en.wikipedia.org/wiki/Aeroflot_Flight_3352`

- Think about how we can get the keywords of of these links



Formatting wikipedia link

- Generally, a wikipedia link has the format...
<https://en.wikipedia.org/wiki/KEYWORD>
- Splitting by the “/” will (usually) give the keyword as the last item in the list.

```
for comment in subreddit.stream.comments():
    if "wikipedia.org" in comment.body:
        split_comment = comment.body.split()
        for word in split_comment:
            if "wikipedia.org" in word:
                link = word
                keyword = link.split('/')[ -1]
                print(keyword)
                print(w.summary(keyword, 3))
```

Possible errors

- You may have encountered an error while testing Wikipedia.
- For example, some keywords will have many pages associated with it.

```
for comment in subreddit.stream.comments():
    if "wikipedia.org" in comment.body:
        split_comment = comment.body.split()
        for word in split_comment:
            if "wikipedia.org" in word:
                link = word
                keyword = link.split('/')[1]
                try:
                    print(w.summary(keyword, 3))
                except w.exceptions.DisambiguationError as e:
                    keyword = e.options[0]
                    print(w.summary(keyword, 3))
            except:
                pass
```

If a DisambiguationError, we pick the first article that it could refer to.

If some other error, we can ignore it.

Add this to our "uciWikiBot" code too.

Replying to links

- If you were able to print out Wikipedia summaries, lets try replying to them.

```
for comment in subreddit.stream.comments():
    if "wikipedia.org" in comment.body:
        split_comment = comment.body.split()
        for word in split_comment:
            if "wikipedia.org" in word:
                link = word
                keyword = link.split('/')[-1]
                try:
                    print(w.summary(keyword, 3))
                except w.exceptions.DisambiguationError as e:
                    keyword = e.options[0]
                    print(w.summary(keyword, 3))
                except:
                    pass
```

We did it? What's next?

- That is the extent of our bot's features.
- Feel free to improve upon your bot, or add any other features you like.
- Keeping your bot open in the background isn't ideal (consider putting it on a server to run indefinitely...)
- Thanks for coming

