

Wikimedia Bots

Part 1: Intro

[\[\[User:Mike Peel\]\]](#)

27 May 2022
15.00-15.10



Created by
volunteers



Anyone can edit

Operated by a
charity (not a
tech company!)

Freely licensed
content (easy to
reuse/remix!)



Item Discussion

Read Labels Id View history More Search Wikidata

South Pole Telescope (Q151315)...

telescope at the South Pole	SPT The South Pole Telescope		
In more languages			
Language	Label	Description	Also known as
English	South Pole Telescope	telescope at the South Pole	SPT The South Pole Telescope
Spanish	Telescopio Polo Sur	No description defined	South Pole Telescope
French	South Pole Telescope	telescope situ au pôle Sud	
Portuguese	Telescopio do Polo Sul	No description defined	

All entered languages

Statements

instance of	Gregorian telescope ...	+ 1 reference
	radio telescope ...	+ 1 reference
	cosmic microwave background experiment ...	+ 0 references

part of	Amundsen–Scott South Pole Station ...	+ 0 references
	Event Horizon Telescope ...	+ 0 references

image	South pole telescope nov2009.jpg 2,112 x 2,416; 2.28 MB	+ 0 references
-------	--	----------------

media legend	The South Pole Telescope in November 2009 (English) Palyt aiglaus teleskopas 2009 m. lapkriči (Lithuanian) Јуриенковоја телескоп 2009 г. (Македонски) Telescopio Pequeño y Grande en la Antártida (Spanish) Il South Pole Telescope nel 2009 (Italian) Telescopio Polo Sur en noviembre de 2009 (Spanish) Telescopi Pol Sud, novembre de 2009 (Catalan) Телескоп Киверо Флореса, 2009 г. (Russian)	+ 0 references
--------------	---	----------------

short name	SPT (English)	+ 0 references
named after	South Pole ...	+ 0 references

century	Antarctica ...	+ 0 references
---------	--------------------------------	----------------

country	+ 0 value ...	+ 1 reference
located in the administrative territorial entity	Antarctic Treaty area ...	+ 0 references

location	South Pole ...	+ 1 reference
coordinate location	A map showing the location of the South Pole Telescope at 80°59'22"S, 45°07'W.	+ 1 reference

elevation above sea level	+ 2.8 kilometers ...	+ 1 reference
	80°59'22"S, 45°07'W	+ 1 reference

Sitelinks

Wikipedia (19 entries)	+ 1 reference
ar (١٩ مقالة)	+ 1 reference
de (19 Artikel)	+ 1 reference
en (South Pole Telescope)	+ 1 reference
es (Telescopio del Polo Sur)	+ 1 reference
fr (Télescope du pôle Sud)	+ 1 reference
it (Telescopio Polo Sud)	+ 1 reference
ja (南極点望遠鏡)	+ 1 reference
pt (Telescope do Polo Sul)	+ 1 reference
ru (Южный полюсный телескоп)	+ 1 reference
uk (Південний південний телескоп)	+ 1 reference
zh (南极望远镜)	+ 1 reference

Wikibooks (0 entries)	+ 1 reference
---------------------------------------	---------------

Wikinews (0 entries)	+ 1 reference
--------------------------------------	---------------

Wikisource (0 entries)	+ 1 reference
--	---------------

Wikiversity (0 entries)	+ 1 reference
---	---------------

Wikivoyage (0 entries)	+ 1 reference
--	---------------

Wiktionary (0 entries)	+ 1 reference
--	---------------

Multilingual sites (1 entry)	+ 1 reference
--	---------------

commons:Category:South Pole Telescope	+ 1 reference
---	---------------

Statements (property -> value)

mass	+ 260 tonnes ...	+ 1 reference
participant in	Event Horizon Telescope ...	+ 0 references
replaces	Antarctic Submillimeter Telescope and Remote Observatory ...	+ 0 references
significant event	construction ...	+ 1 reference
	start time	November 2008 ...
	end time	February 2007 ...
diameter	+ 10.0 metre ...	+ 1 reference
	aperture to part	primary mirror ...
	+ 1 metre ...	+ 1 reference
	aperture to part	secondary mirror ...
reference URL	https://ipole.uchicago.edu/public/telescope_optics.html	
title	Telescope Optics (English)	
publisher	South Pole Telescope	
released	5 April 2017	
angular resolution	+ 1 arcminute ...	+ 0 references
service entry	+ 16 February 2007 ...	+ 0 references
area	+ 78.5 square metre ...	+ 0 references
official website	https://ipole.uchicago.edu	
	language of work or name	English ...
webcam page URL	https://www.usgs.gov/video/polarcams/webcams.cfm	
	released	17 May 2021 ...
statement is subject of	New South Pole Telescope to study dark energy ...	+ 0 references
	Detection of B-Mode Polarization in the Cosmic Microwave Background with Data from the South Pole Telescope ...	+ 0 references
funding schema	United States Department of Energy ...	+ 0 references
has part(s) of the class	Telescope mount ...	+ 1 reference
	instance of	altazimuth mount ...
Commons category	South Pole Telescope ...	+ 0 references

show derived statements

External identifiers

Freebase ID	+ Jm06 2Bb	+ 1 reference
Microsoft Academic ID	+ 2779770393 ...	+ 0 references
OpenAlex ID	+ C2779770393	+ 1 reference
Twitter usernames	+ SPT telescope	+ 0 references

Twitter user name ID	1956077080	+ 0 references
start time	11 October 2013 ...	
end time	14 March 2022 ...	

This page was last edited on 20 March 2022, at 09:27.
All structured data from the main, Property, Lemma, and EntitySchema namespaces is available under the Creative Commons CC0 License; text in the other namespaces is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy.

Privacy policy About Wikidata Contributors Developers Statistics Cookie statement





Media page
Admin board / backlog
Community portal
Village pump
Help center

Participate

Upload file
Recent changes
Latest files
Random file
Contact us

Tools

What links here
Related changes
Special pages
Printable link
Page information
Wikidata item
GLAMorous
Perform batch task
RSS feed
Nominate category for discussion
AC/DC

Print/export
Create a book
Download as PDF
Printable version

In other projects

In Wikipedia
Cyrillic
Deutsch
English
Español
Français
日本語
Polski
Русский
Українська

View 10 more

Edit links

Category Discussion

English Mike Peel Talk Preferences Beta Watchlist Uploads Contributions Log out

View Edit History More Search Wikimedia Commons

Help

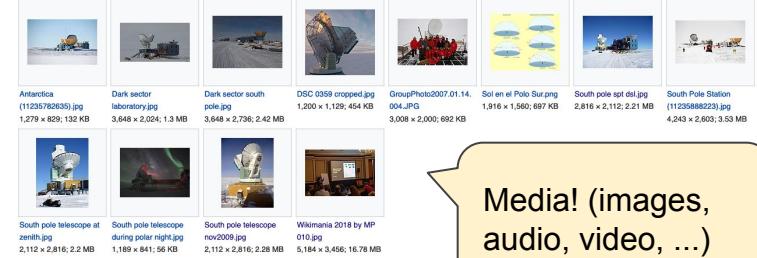
Category:South Pole Telescope

Categories (+7): Radio telescopes (-) (1) (t) Science and technology in Antarctica (-) (z) (1) (t) Amundsen–Scott South Pole Station (-) (z) (1) (t) (+)
Non-lexical index: Uses of Wikidata Infobox · Uses of Wikidata Infobox with maps · Pages with maps

From Wikimedia Commons, the free media repository

Media in category "South Pole Telescope"

The following 12 files are in this category, out of 12 total.



Media! (images, audio, video, ...)

South Pole Telescope [Hide]



The South Pole Telescope in November 2009

Upload media ↗
Wikipedia ↗

Instance of Gregorian telescope, radio telescope, cosmic microwave background experiment

Part of Amundsen–Scott South Pole Station, Event Horizon Telescope

Named after South Pole

Location South Pole, Antarctic Treaty area

Service entry 16 February 2007

Significant construction event (2006–2007)

Mass 260 t

Diameter 10 m (primary mirror)
1 m (secondary mirror)

Angular resolution 1'

Area 78.5 m²

Elevation above sea level 2.8 km

Replaces Antarctic Submillimeter Telescope and Remote Observatory
official website ↗



2 km | Wikimeda map ↗ | Map data ...
89° 59' 23" S, 45° 00' 00" W ↗

Authority control [hide]

O151315

Wikidata ↗ · PetScan ↗ · Scholia ↗ ·

Statistics ↗ · OpenStreetMap ↗ · Locator tool ↗ ·

WikiSchooMe ↗ · Search deprecate

Categories (normally at bottom)

Infobox from Wikidata (multilingual!)

What can be done?

Create/Edit/Delete/Undelete/
protect/unprotect/etc... pages

Search / Query

Download content

Scrape websites

Compare with other databases

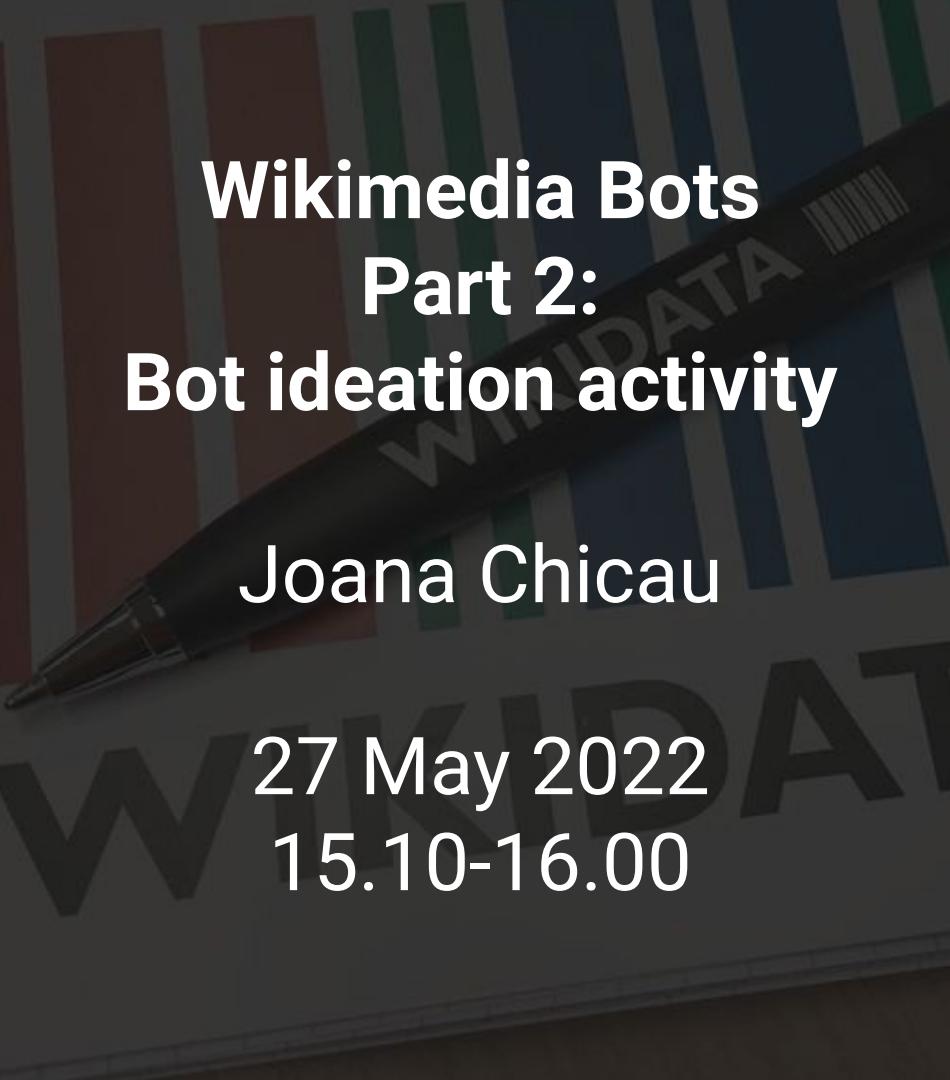
... and lots more!

Automated edits need to be 100%
accurate (or >99.9%)

Semi-automated can be less accurate -
but check changes before saving!

Focus on routine tasks that are easily
described

Keep things modular/simple as much
as possible



Wikimedia Bots

Part 2:

Bot ideation activity

Joana Chicau

27 May 2022
15.10-16.00

Your ideas here!

Wikimedia Bots

Part 3: Implementation

[[User:Mike Peel]]

27 May 2022
16.00-17.00

Existing bots
Bot approval processes
Writing bot code



Examples of existing bots - <https://en.wikipedia.org/wiki/Wikipedia:Bots>

Examples [edit source]

Some examples of bots are:

- [User:AAlertBot](#) – delivering [article alerts](#) to WikiProjects about ongoing discussions.
- [User:AnomieBOT](#) – large variety of tasks, most well known for adding dates to amboxes.
- [User:ClueBot NG](#) – reverts possible [vandalism](#).
- [User:Citation bot](#) – Adds PMID, ISBNs, etc to pages.
- [User:CommonsDelinker](#) – removes links to files deleted at [Wikimedia Commons](#) due to policy infractions.
- [User:DatBot](#) – patrols the edit filters and resizes non-free images.
- [User:DumbBOT](#) – often removes protection templates from recently unprotected pages.
- [User:InternetArchiveBot](#) – retrieves archived copies of dead links.
- [User:ListeriaBot](#) – Experimental bot by [Magnus Manske](#). It generates and updates lists on Wikipedia.
- [User:Lowercase sigmabot III](#) – archives talk pages.
- [User:ST47ProxyBot](#) – automatically blocks proxies due to the local policy against [open proxies](#).
- [User:SineBot](#) – signs comments left on talk pages.
- [User:WP 1.0 bot](#) – works with the [Version 1.0 Editorial Team](#).

Examples of existing bots - https://meta.wikimedia.org/wiki/User:Pi_bot

Code	Description		
wir_newpages.py	Add sitelinks, or create new items, for humans with new Wikipedia entries	commonscat_p910_tidy.py	Items with P910 and a sitelink in both the topic and category items, where one redirects to the other
wir_newpages_duplicity.py wir_newpages_category.py wir_newpages_special.py	Add sitelinks, or create new items, for humans with new Wikipedia entries (backlog)	commonscat_move_from_P1754.py commonscat_move_from_P910.py commonscat_copy_from_P373.py commonscat_check.py	Move commons sitelinks from list to category items Move commons sitelinks from topic to category items Copy commons sitelinks from P373 Checking for sitelinks to Commons category redirects. Mismatch between P373 and sitelink.
wikidata_p301_inverse.py wikidata_p910_inverse.py wikidata_p1753_inverse.py wikidata_p1754_inverse.py	Add reciprocal values for items that have inverse statements	Commons , enwp commons_wikidata_search.py commons_wikidata_infobox.py commons_qi_by_user.py commons_qi_by_user.py commons_qi_by_user.py	Import coordinates from Commons and Wikipedias Search for new Commons sitelinks to add to Wikidata Wikidata Infobox deployment List of Commons QI, FI and VI by user
wikidata_newshipname.py wikidata_viewof.py	Create new category items to link via category items	commons_migrate_ids_to_wikidata.py commons_import_ids_to_wikidata.py commons_defaultsort_conflicts.py commons_date_find.py commons_check_id.py cochrane.py cochrane_fr.py check_tgwiki.py Bitbucket	Trim duplicate external IDs that are provided by Wikidata Import IDs from Commons to Wikidata Maintain commons infoboxes by checking for default sort conflicts Regularly add Commons category links for date categories/items Commons sitelinks from ID matching Cochrane updates Cochrane updates Remove broken sitelinks to deleted pages Copy human names between languages and dialects
wikidata_newitem.py	Create new Wikidata entries for people with Commons categories	commons_wikidata_infobox_tidy.py	
wikidata_new_from_wikipedia_query_article.py wikidata_new_from_wikipedia_query_category.py	Create new items from Wikipedias		
wikidata_import_labels_from_commons.py	Add English labels from commons category names		
wikidata_import_infobox_qid.py commons_wikidata_infobox_tidy.py	Wikidata Infobox tidying		
wikidata_en_biosgraphy_names.py	Copy labels for humans from other languages to English		
wikidata_bot_requests.py	Archive Wikidata bot requests		
wikidata_bad_p373.py	Remove bad P373 values		
whs_infobox.py	Migration of WHS Wikidata infoboxes		
simplewp_commonscat_fix.py	Commons link maintenance		

Bot approvals process

Different on each wiki

For enwiki, see

https://en.wikipedia.org/wiki/Wikipedia:Bot_policy

"Automated editing processes, known as "bots", must be harmless and useful, have approval, use separate user accounts, and be operated responsibly."

Changes have to be significant (not cosmetic), but restricted from creating complete new articles

Wikidata:

<https://www.wikidata.org/wiki/Wikidata:Bots>

Commons:

<https://commons.wikimedia.org/wiki/Category:Commons:Bots>

Wikidata and Commons a lot more welcoming than enwiki! Inc. creating new Wikidata items, bulk uploads to Commons. But more technical.

Submitting a bot request

Wikidata:Requests for permissions/Bot/Pi bot

< Wikidata:Requests for permissions | Bot

The following discussion is closed. **Please do not modify it.** Subsequent comments should be made in a new section. A summary of the conclusions reached follows.

Approved--Ymblanter (talk) 18:53, 1 March 2018 (UTC)

Pi bot [edit]

Pi bot (talk • contribs • new items • SUL • Block log • User rights log • User rights)

Operator: Mike Peel (talk • contribs • logs)

Task/s: Fix mismatches between Commons category (P373) and the commons sitelink that are due to one linking to a category redirect.

Code: https://bitbucket.org/mikepeel/wikicode/src/master/commonscat_check.py

Function details: Query for cases where Commons category (P373) is not the same as the commons sitelink (but both are present). If one of them points to a commons category that has commons:Template:Category redirect in it, and that redirect points to a category with the same name as the other value on Wikidata, then update the value to avoid the redirect. Run daily to catch new cases caused by category moves on Commons. Example edits: [1] [2] (for the two different cases). --Mike Peel (talk) 14:36, 22 February 2018 (UTC)

Looks good, pls run 50 test edits.--Ymblanter (talk) 08:37, 25 February 2018 (UTC)

@Ymblanter: Done, see [3]. The test run showed up a few issues, so I've modified the code to cope with them. If there are multiple P373 values present, then the code skips that entry. If one of the two commons links leads to a missing page, then it counts them and doesn't make an edit at the moment (although I've drafted some code that will also handle this case, which I'll enable once I find some test cases). And if Wikidata reports an interwiki conflict, then the code logs that and moves on (I don't know if there's a better way to check for this in pywikibot?). Thanks. Mike Peel (talk) 13:36, 25 February 2018 (UTC)

Thanks. I will approve the bot in a couple of days provided there have been no objections raised.--Ymblanter (talk) 14:26, 25 February 2018 (UTC)

The above discussion is preserved as an archive. **Please do not modify it.** Subsequent comments should be made in a new section.

Separate
bot account

Code online
somewhere

Do a test
run

(Normally lots
more discussion
here!)

Short task
description

Full details

Approval

Category: Archived requests for permissions

Semi-automated methods

It is possible to 'pre-fill' articles using Wikidata (infobox, introduction section, references, ...), for example see

<https://en.wikipedia.org/wiki/Wikipedia:Mobile>

AutoWikiBrowser (AWB) can be used on Windows, you have to request access, can semi-automate many repetitive tasks:

<https://en.wikipedia.org/wiki/Wikipedia:AutoWikiBrowser>

Anti-vandalism tools like
<https://en.wikipedia.org/wiki/Wikipedia:Huggle>

... and you can run Python scripts in semi-automated modes, e.g., by manually having to enter 'y' to save an edit:

```
# Print things here
text = input("Save this edit? ")
if text == 'y':
    # Save edit
```

Even when running semi-automated, should still check with community first - otherwise likely to be blocked!

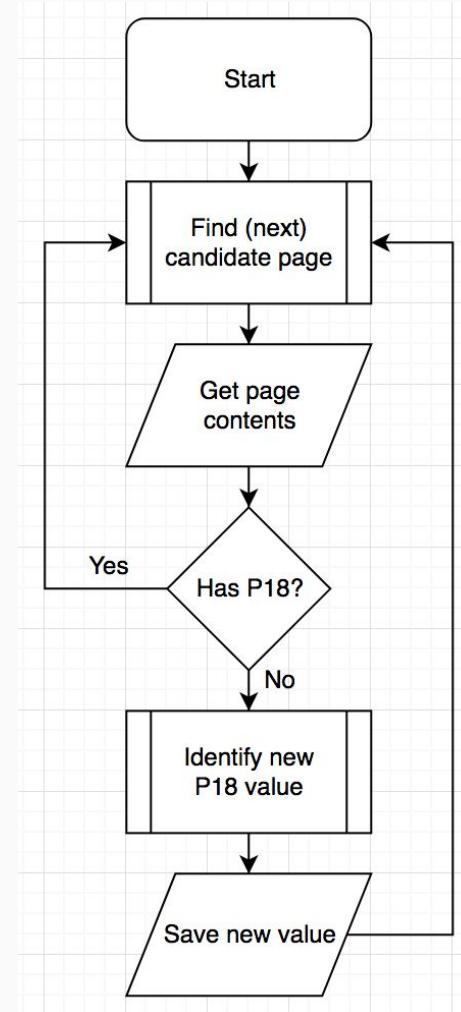
Flow charts

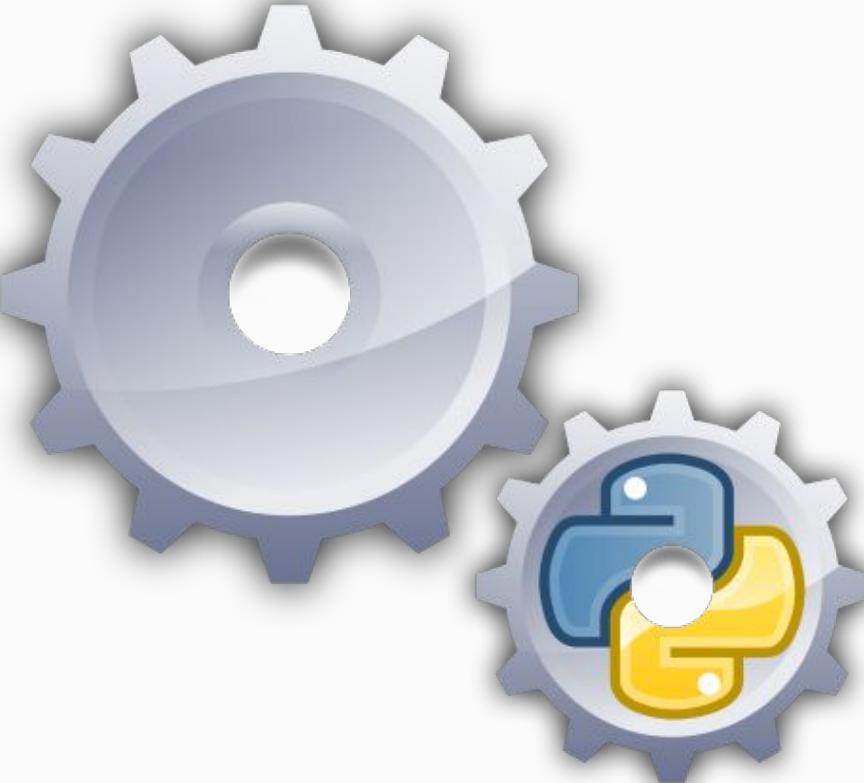
Write out the code logic in a series of steps

Identify main questions/if statements

Create online at e.g., www.draw.io

ANSI/ISO Shape	Name
→	Flowline (Arrowhead) ^[15]
○	Terminal ^[14]
□	Process ^[15]
◇	Decision ^[15]
平行四边形	Input/Output ^[15]
注释	Annotation ^[14] (Comment) ^[15]
平行四边形	Predefined Process ^[14]
○	On-page Connector ^[14]
盾形	Off-page Connector ^[14]





Pywikibot

Python framework to edit
MediaWiki content

'pip install pywikibot'

or git pull

<https://github.com/wikimedia/pywikibot.git>

can use python2, but better to
use python3 now!

... sorry, lots of examples here
are python2, just change
`print "text" -> print("text")`

Set up the pywikibot user configuration file

Initial definitions
(python2)

Language

Need a file with this name in your code directory!

```
user-config.py
1 # -*- coding: utf-8 -*-
2 from __future__ import unicode_literals
3
4 # Default mediawiki family
5 family = 'wikipedia'
6
7 # Default mediawiki language
8 mylang = 'pt'
9
10 # Username for each wiki
11 usernames['wikipedia']['pt'] = u'Mike Peel'
12 usernames['wikidata']['wikidata'] = u'Mike Peel'
13 |
```

MediaWiki family

Usernames

Starting the code and connecting to ptwiki and Wikidata

Initial definitions

Import modules

Connect!

```
example.py      x
1 #!/usr/bin/env python
2 # -*- coding: utf-8 -*-
3 # Example pywikibot code description here
4 # Mike Peel    17-Sep-2018      v1 - start
5
6 # Import modules
7 import pywikibot
8 from pywikibot import pagegenerators
9 from pywikibot.data import api
10 import numpy as np
11 import requests
12
13 # You may need to enforce the use of utf-8
14 import sys
15 reload(sys)
16 sys.setdefaultencoding('UTF8')
17
18 # Connect to ptwiki
19 ptwiki = pywikibot.Site('pt', 'wikipedia')
20 # and then to wikidata
21 ptwiki_repo = ptwiki.data_repository()
```

Documentation!

Python2:
use utf-8

Python3:
remove
lines 14-16!

Doing a test edit to the Portuguese Wikipedia

Defines the function

Making the edit

Running the function

```
def editarticle(page):
    text = page.get()
    text = text + "\nThis is a test edit"
    page.text = text
    try:
        page.save("Saving test edit")
        return 1
    except:
        print "That didn't work!"
        return 0

# Page must exist already!
page = pywikibot.Page(ptwiki, 'Usuário(a):Mike_Peel/teste')
test = editarticle(page)
print test
```

Loads the page text

Saving the edit

Loads the page

python example.py

Replacing text in the Portuguese Wikipedia

```
32 def editarticle2(page):
33     text = page.get()
34     text = text.replace('This is a test edit','Isto é uma edição de teste')
35     page.text = text
36     try:
37         page.save("Saving test edit")
38         return 1
39     except:
40         print "That didn't work!"
41         return 0
42
43 page = pywikibot.Page(ptwiki, 'Usuário(a):Mike_Peel/teste')
44 test = editarticle2(page)
45 print test
```

Printing the QID, label, sitelink and a claim from Wikidata

Defines the function

26

```
def printwikidata(wd_item):  
    qid = wd_item.title()  
    print qid
```

Print the QID

Fetch the Wikidata item

30

```
    item_dict = wd_item.get()
```

Print a label

34

```
try:  
    print 'Name: ' + item_dict['labels']['en']
```

```
except:  
    print 'No English label!'
```

```
try:  
    print 'ptwiki article: ' + item_dict['sitelinks']['ptwiki']
```

```
except:  
    print 'No Portuguese article!'
```

```
try:  
    print item_dict['claims']['P31']
```

```
except:  
    print 'No P31'
```

Print a sitelink

38

Print a claim

The output from the previous slide

Q511405

Name: Sky Polarization Observatory

No Portuguese article!

```
[Claim.fromJSON(DataSite("wikidata", "wikidata"), {u'type': u'statement', u'mainsnak': {u'datatype': u'wikibase-item', u'datavalue': {u'type': u'wikibase-entityid', u'value': {u'entity-type': u'item', u'numeric-id': 184356}}, u'property': u'P31', u'snaktype': u'value'}, u'id': u'Q511405$2228cb  
b0-4fab-4319-9890-4b2d5764dc27', u'rank': u'normal'}), Claim.fromJSON(DataSite("wikidata", "wikida  
ta"), {u'type': u'statement', u'mainsnak': {u'datatype': u'wikibase-item', u'datavalue': {u'type':  
u'wikibase-entityid', u'value': {u'entity-type': u'item', u'numeric-id': 33093130}}}, u'property':  
u'P31', u'snaktype': u'value'}, u'id': u'Q511405$b40c9ccf-4e9f-fe0b-cfe1-d983199084f2', u'rank':  
u'normal'})]
```

Printing the property values correctly

Loop over claims

43

try:

44

45

for claim in item_dict['claims']['P31']:

p31_value = claim.getTarget()

p31_item_dict = p31_value.get()

print 'P31 value: ' + p31_value.title()

print 'P31 label: ' + p31_item_dict['labels']['en']

46

47

48

except:

49

print "That didn't work!"

50

51

return 0

Get the target contents

The output from the previous slides

```
Q511405
Name: Sky Polarization Observatory
No Portuguese article!
[Claim.fromJSON(DataSite("wikidata", "wikidata"), {"type": "statement", "mainsnak": {"datatype": "wikibase-item", "datavalue": {"type": "wikibase-entityid", "value": {"entity-type": "item", "numeric-id": 184356}}, "property": "P31", "snaktype": "value"}, "id": "Q511405$2228cb
b0-4fab-4319-9890-4b2d5764dc27", "rank": "normal"}), Claim.fromJSON(DataSite("wikidata", "wikida
ta"), {"type": "statement", "mainsnak": {"datatype": "wikibase-item", "datavalue": {"type": "wikibase-entityid", "value": {"entity-type": "item", "numeric-id": 33093130}}, "property": "P31", "snaktype": "value"}, "id": "Q511405$b40c9ccf-4e9f-fe0b-cfe1-d983199084f2", "rank": "normal"})]
P31 value: Q184356
P31 label: radio telescope
P31 value: Q33093130
P31 label: cosmic microwave background experiment
```

How to generate lists of pages to edit - using a sparql query

Sparql
query

```
69 sparql = "SELECT ?item WHERE { ?item wdt:P31 wd:Q184356 } LIMIT 10"
70 generator = pagegenerators.WikidataSPARQLPageGenerator(sparql, site=ptwiki_repo)
71 for page in generator:
72     printwikidata(page)
```

Loop over
results

Run the
query

How to generate lists of pages to edit - using categories and template uses

```
99 targetcat = 'Categoria:Telescópios'
100 cat = pywikibot.Category(ptwiki, targetcat)
101 subcats = pagegenerators.SubCategoriesPageGenerator(cat, recurse=False);
102 for subcat in subcats:
103     print subcat.title()
104
105 pages = pagegenerators.CategorizedPageGenerator(cat, recurse=False);
106 for page in pages:
107     print page.title()
108
109 template = pywikibot.Page(ptwiki, 'Predefinição:Info/Telescópio')
110 targets = template.embeddedin()
111 for target in targets:
112     print target.title()
113
114 targets = pagegenerators.RandomPageGenerator(total=10, site=ptwiki, namespaces='14')
115 for target in targets:
116     print target.title()
```

Search for existing Wikidata items

```
109 # From https://gist.github.com/ettorerizza/7eaebbd731781b6007d9bdd9ddd22713
110 def search_entities(site, itemtitle):
111     params = { 'action' : 'wbsearchentities',
112                'format' : 'json',
113                'language' : 'en',
114                'type' : 'item',
115                'search': itemtitle}
116     request = api.Request(site=site, parameters=params)
117     return request.submit()
118
119 wikidataEntries = search_entities(ptwiki_repo, "Neuromat")
120 if wikidataEntries['search'] != []:
121     results = wikidataEntries['search']
122     numresults = len(results)
123     for i in range(0,numresults):
124         qid = results[i]['id']
125         label = results[i]['label']
126         print qid + " - " + label
```

Doing a test edit to Wikidata - adding 'instance of' = 'sandbox'

Get the item

```
21 def editwikidata(wd_item, propertyid, value):  
22     qid = wd_item.title()  
23     print qid  
24     item_dict = wd_item.get()  
25  
26     claim_target = pywikibot.ItemPage(ptwiki_repo, value)  
27     newclaim = pywikibot.Claim(ptwiki_repo, propertyid)  
28     newclaim.setTarget(claim_target)  
29     print newclaim  
30     wd_item.addClaim(newclaim, summary=u'Adding test claim')  
31  
32     return 0  
33  
34 testqid = 'Q4115189' # Wikidata sandbox  
35 testproperty = 'P31' # instance of  
36 testvalue = 'Q3938' # Sandbox  
37 wd_item = pywikibot.ItemPage(ptwiki_repo, testqid)  
38 print editwikidata(wd_item, testproperty, testvalue)
```

Get the target item

Create a claim

Set claim target

Save the claim

QID and property ID to edit

Manually checking the edit before saving it

```
print newclaim
text = raw_input("Save? ")
if text == 'y':
    wd_item.addClaim(newclaim, summary=u'Adding test claim')
```

Q4115189

```
Claim.fromJSON(DataSite("wikidata", "wikidata"), {u'type': u'statement', u'mainsnak': {u'datatype': u'wikibase-item', u'datavalue': {u'type': u'wikibase-entityid', u'value': {u'entity-type': u'item', u'numeric-id': 3938}}}, u'property': 'P31', u'snaktype': u'value'}, u'rank': u'normal'})
```

Save? █

Creating a new Wikidata entry

```
wikidata_newitem.py •
1 #!/usr/bin/python
2 # -*- coding: utf-8 -*-
3 # Create new Wikidata items
4 # Started 25 August 2018 by Mike Peel
5 from __future__ import unicode_literals
6
7 import pywikibot
8 import numpy as np
9 import time
10 import string
11 from pywikibot import pagegenerators
12 import urllib
13
14 commons = pywikibot.Site('commons', 'commons')
15 repo = commons.data_repository() # this is a DataSite object
16
17 def newitem(category, items):
18     new_item = pywikibot.ItemPage(repo)
19     new_item.editLabels(labels={"en":category}, summary="Creating item")
20     candidate_item = pywikibot.ItemPage(repo, new_item.getID())
21     candidate_item = pywikibot.ItemPage(repo, new_item)
22     print candidate_item
23
24     data = {'sitelinks': [{ 'site': 'commonswiki', 'title': category}]}
25     candidate_item.editEntity(data, summary=u'Add commons sitelink')
26
27     for item in items:
28         claim = pywikibot.Claim(repo, item[0])
29         claim.setTarget(pywikibot.ItemPage(repo, item[1]))
30         try:
31             candidate_item.addClaim(claim, summary=u'Setting '+item[0]+' value')
32         except:
33             print "That didn't work"
34
35
36 category = 'Category:1546 in Finland'
37 items = [['P31','Q4167836'],['P971','Q65701'],['P971','Q33']]
38 test = newitem(category, items)
```

Scrape a website

```
89 def parsesite(url):
90     try:
91         r = requests.get(url)
92         websitetext = r.text
93     except:
94         print 'Problem fetching page!'
95         return 0
96     # print websitetext
97     split = websitetext.split("<h1 style='display:none'>")
98     i = 0
99     for item in split:
100         i+=1
101         # Skip the top part
102         if i > 2:
103             # print item
104             print 'Title: ' + item.split('</h1>')[0].strip() + '\n'
105             print 'Museum: ' + item.split("strong>Museu:</strong><span itemprop='publisher'>")[1].split("</span>")
106                                         )[0].strip() + "\n"
107     return 0
108 parsesite('http://www.museusdoestado.rj.gov.br/sisgam/index.php?pagina=1&operador=or&busca=a%20b%20c%20d%20e%20f
%20g%20h%20i%20j%20k%20l%20m%20n%20o%20p%20q%20r%20s%20t%20u%20v%20w%20x%20y%20z&museu=todos&qresultados=40')
```

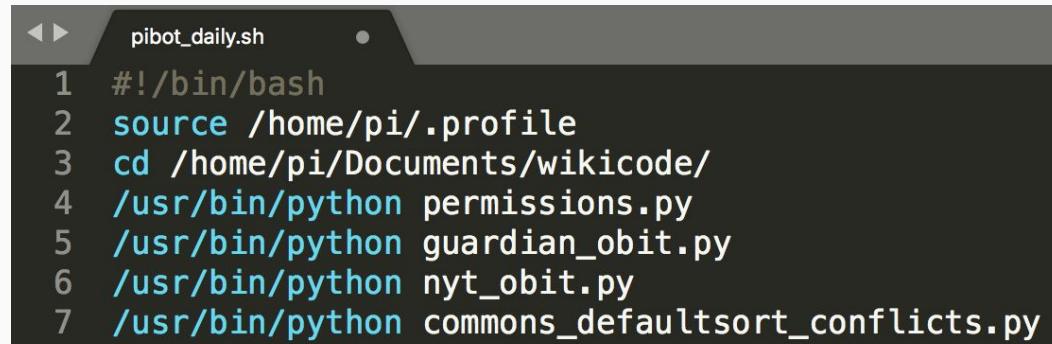
Cron jobs to automatically run the bot on a set schedule

```
pi@raspberrypi3:~ $ crontab -e
GNU nano 2.7.4                                         File: /tmp/crontab.JaMTUg/crontab

SHELL=/bin/bash

TERM=xterm
PYTHONIOENCODING=UTF-8
LANG=en_US.UTF-8
LC_ALL=en_US.UTF-8
# /etc/crontab: system-wide crontab

# m h dom mon dow user  command
0 8 * * * /home/pi/Documents/wikicode/pibot_daily.sh
0 9 1 * * /home/pi/Documents/wikicode/pibot_monthly.sh
```



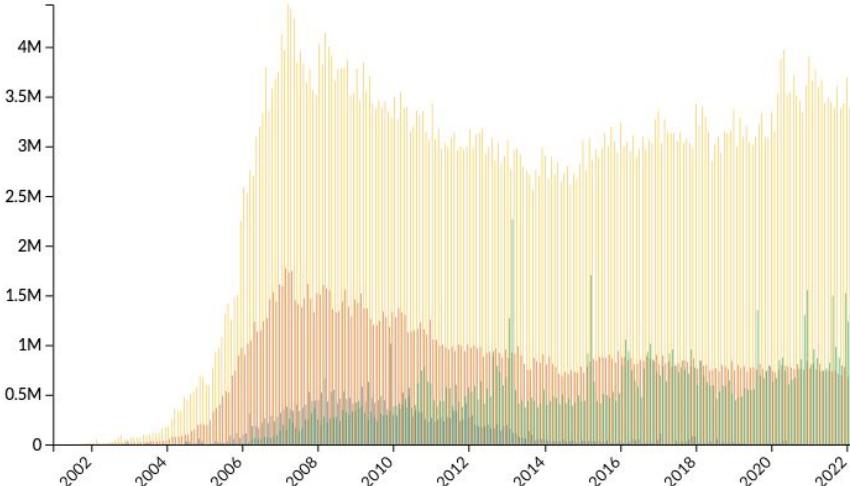
A screenshot of a terminal window showing the contents of a file named "pibot_daily.sh". The window has a dark background with light-colored text. At the top, there are navigation icons for a file browser. The file content is as follows:

```
1 #!/bin/bash
2 source /home/pi/.profile
3 cd /home/pi/Documents/wikicode/
4 /usr/bin/python permissions.py
5 /usr/bin/python guardian_obit.py
6 /usr/bin/python nyt_obit.py
7 /usr/bin/python commons_defaultsort_conflicts.py
```

Statistics of humans vs. bots over time

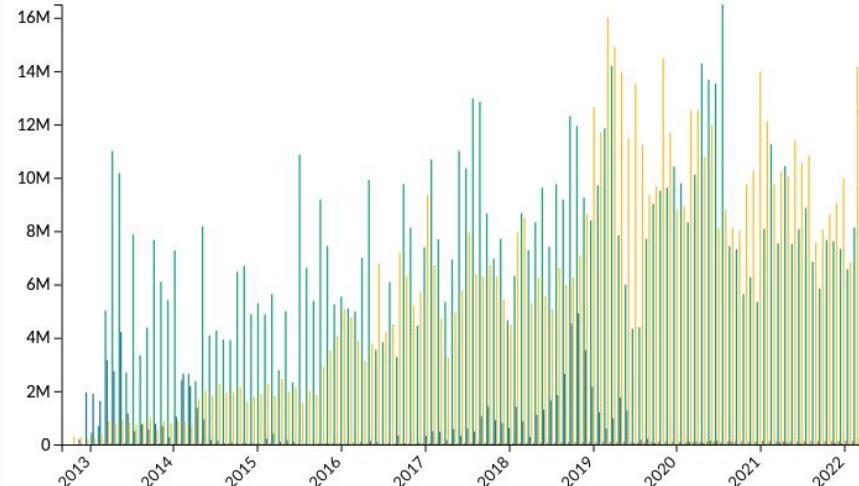
Edits

Anonymous Group bot Name bot User



Edits

Anonymous Group bot Name bot User



https://stats.wikimedia.org/#/en.wikipedia.org/contributing/edits/normal|bar|all|editor_type~anonymous*group-bot*user|monthly

https://stats.wikimedia.org/#/wikidata.org/contributing/edits/normal|bar|2012-07-23~2022-05-27|editor_type~anonymous*group-bot*user|monthly

Useful links

- <https://github.com/mpeel/wikicode/> - pi bot + semi-automated scripts
- https://meta.wikimedia.org/wiki/User:Pi_bot - list of scripts + descriptions
- <https://github.com/mpeel/wikicode/blob/master/example.py>
- https://www.wikidata.org/wiki/Wikidata:Bot_requests - requests for bots to do things
- https://doc.wikimedia.org/pywikibot/master/api_ref/pywikibot.html
- <https://phabricator.wikimedia.org/T276329> - see 'Tasks' for some intro tasks
(WMF participates in Outreachy + Google Summer of Code each year!)
- <https://www.google.com/> - good place to look for example code ;-)
- <https://stackoverflow.com/> - where you'll actually find example code
- https://www.wikidata.org/wiki/User_talk:Mike_Peel
Feel free to ask me questions! (Both now, and later online!)

Open discussion

- What makes sense / doesn't make sense so far?
- How do the practicalities match up with your ideas?
- What are your next steps?