U0NRYQNAZ : i've personally started writing my own markdown filters/extensions for things that I need.

U4BMZ90T0 : Not sure if anyone can help, but have a quick question regarding <#C5XHHMXHB|os_windows> , powershell more specifically.

U4BMZ90T0 : <https://pythondev.slack.com/archives/C5XHHMXHB/p1498673610887626>

U1BP42MRS : I'd be surprised if that's the case... but is it not working for you?

U4BMZ90T0 : provide**

U5LNXQHN3 : <@U0NRYQNAZ> That sounds useful, especially since I've found that both Markdown and reST each only do about 80% of what I want... a different 80%... but unfortunately I only need to do this one piece of documentation then I'll probably never need it again, so it's hard to justify the time expense

U2BS4M1RV : <@U4BMZ90T0> I believe powershell requires a path be stated to any executable not in the search path. But, a ./myscript.py is sufficient.

U2BS4M1RV : Or, in this case, .exe

U2BS4M1RV : <@U1UFZTD4J> I showed the lead engineer Sentry. System looks cool, but he thinks we would hit the 20k transactions per day fairly quickly. Do you know of similar systems without limits?

U5LNXQHN3 : "Fixed" my documentation problem by hand-editing the HTML and CSS, and planning to use highlight.js to highlight the code blocks. Felt like the static site generators were just giving me extra hoops to jump through, in the end.

U5CRANMFV : Hi, all.How do I calculate the difference between two dates is best?
The output should only be in months.
The only way I could.
```
In [25]: startDay
Out[25]: datetime.datetime(2017, 6, 1, 0, 0)



In [26]: endDay
Out[26]: datetime.datetime(2017, 1, 1, 0, 0)



In [27]: startDay - endDay
Out[27]: datetime.timedelta(151)



In [28]: different = startDay - endDay



In [29]: different.days//30
Out[29]: 5
```

U5LNXQHN3 : How do you define months in this context?

U5CRANMFV : <@U5LNXQHN3> sorry?

U5LNXQHN3 : A month is not a fixed unit of time (at least not in the cultures I know of)

U5CRANMFV : I need to know how many months have passed between two dates

U5LNXQHN3 : If you want the intuitive answer then I think you will just have to check the difference in years and months individually

U5CRANMFV : for example I started it today - June 2017, and the end - June 2016. The difference - 12 months

U5CRANMFV : I think the best way is to pass the Python...

U5LNXQHN3 : Does this work? `((endDay.year - startDay.year) * 12) + (endDay.month - startDay.month)`

U5CRANMFV : works

U5CRANMFV : Thank you. You are a genius.

U5LNXQHN3 : You can't use timedelta for this because a span of 30 days is over a month if the span starts on Feb 1st and under a month if the span starts on Jan 1st

U5LNXQHN3 : no problem

U5CRANMFV : exactly...

U1UFZTD4J : <@U2BS4M1RV> if you self host, there is no limit

U1UFZTD4J : I would recommend that. With docker its really not all that hard to host it

U1UFZTD4J : thats what I do. All similar systems will have some type of limit. There is also bugsnag, and rollbar, but they all have limits

U1UFZTD4J : self hosting is the only way to have no limits. Oh, I guess google cloud has one called stack-driver that is just a flat monthly fee + minimal data charges

U5NMSURAQ : use `dateutil`

U1BP42MRS : or `maya` or `arrow`, etc

U5LNXQHN3 : That's cheating :wink:

U5NMSURAQ :
```
&gt;&gt;&gt; from datetime import datetime
&gt;&gt;&gt; from dateutil.relativedelta import relativedelta
&gt;&gt;&gt; from pprint import pprint
&gt;&gt;&gt; start = datetime(2017, 1, 1)
&gt;&gt;&gt; end = datetime.today()
&gt;&gt;&gt; delta = relativedelta(end, start)
&gt;&gt;&gt; pprint(vars(delta))
{'_has_time': 1,
 'day': None,
 'days': 27,
 'hour': None,
 'hours': 22,
 'leapdays': 0,
 'microsecond': None,
 'microseconds': 510107,
 'minute': None,
 'minutes': 53,
 'month': None,
 'months': 5,
 'second': None,
 'seconds': 28,
 'weekday': None,
 'year': None,
 'years': 0}
```

U2BS4M1RV : He may have misread that then, I'll look at it. Thanks.

U5LNXQHN3 : I don't think arrow helps here

U5NMSURAQ : if you don't need to do anything complex, dateutil is nice and small

U1BP42MRS : Just use a unix timestamp and do maths in second offsets :wink: Of course, then there is the problem of leap seconds...

U5NMSURAQ : and often is already installed as a dependency of another package

U5NMSURAQ : arrow is cool, but it's for really complex stuff

U5LNXQHN3 : just don't use dateutil for parsing, because it often guesses wrong

U0NRYQNAZ : this way i was able to style it differently in css.

U5LNXQHN3 : <@U0NRYQNAZ> Looks like a good approach. Something I'll consider if I find myself doing this again