

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

U5LNXQHN3 : I don't think arrow helps here

U5NMSURQA : 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...

U5NMSURQA : and often is already installed as a dependency of another package

U5NMSURQA : 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

U1UFZTD4J : whats the cleanest/shortest way to check if any value in a tuple is in another tuple?

U1UFZTD4J : for example:```

```
for i in tup1:
    for j in tup2:
        if i == j:
            return True
return False
```
```

U1UFZTD4J : I have a feeling there are some built in's I can use...

U5NMSURQA : <@U1BP42MRS> they needed an offset in months (see thread)

U5NMSURQA : ```&gt;&gt;&gt; start = datetime(2016, 12, 29)

&gt;&gt;&gt; end = datetime.today()

&gt;&gt;&gt; (end - start).total\_seconds() / (3600 \* 24 \* 30)

6.065205603584491

&gt;&gt;&gt; delta = relativedelta(end, start)

&gt;&gt;&gt; pprint(vars(delta))

```
{'_has_time': 1,
'day': None,
'days': 30,
'hour': None,
'hours': 22,
'leapdays': 0,
'microsecond': None,
'microseconds': 924491,
'minute': None,
'minutes': 56,
'month': None,
'months': 5,
'second': None,
'seconds': 52,
'weekday': None,
'year': None,
'years': 0}
>>>
```
```

Using seconds is just wrong :confused:

U1BP42MRS : <@U5NMSURQA> then convert back

U5NMSURQA : you can't count how many months passed if you only have unix timestamp 1 and unix timestamp 2

U29163YQH : ```[x for x in tup1 if x in tup2]

```

U1BP42MRS : `any(x in y for x in (1, 2, 3, 4))`

U1BP42MRS : I guess you said any value, so `any`

U1BP42MRS : the perk of `any` over the list is it will short circuit itself on the first found one

U1UFZTD4J : ya.. that works. Is it possible without an explicit loop? (im getting picky, but im challenging myself)

U1BP42MRS : since at least 1 is true

U1BP42MRS : probably operator