

Friday the 13th

culture. Where this superstition comes from is not immediately known, and although there are numerous reports in various religions and customs, it seems that this superstition, is only about 100 years old, at most.

The combination of Friday and the number thirteen is also not as universal as you might think. In Belgium, the Netherlands and England, for example, Friday the thirteenth is indeed seen as an unlucky day. In Greece, Spain and Latin America, however, Tuesday the thirteenth is an unlucky day. Furthermore, in Italy everyone is extra careful on Friday the seventeenth.

Assignment

Write a function `unluckydays` that has one required argument and three optional arguments. The mandatory argument specifies a start date and the optional arguments indicate an end date, a day number and a weekday number, respectively. The function must return the number of days between the start date and the end date (including the endpoints) that have a daynumber that falls on the specified weekday. If only the mandatory argument is given, the number of Fridays the thirteenth between the start date and today is calculated. If the start date is later than the end date, there are no days between the start date and end date, and as a result, the function logically returns the value 0.

Preparation

In this assignment you will have to make use of the data types `date` and `timedelta` that are defined in the `datetime` module of the [Python Standard Library](#). Before starting to work on the actual assignment, you should first have a look at how Python responds when you execute the following sequence of instructions in an interactive Python session:

- ```
>>> from datetime import date
>>> birthday = date(1983, 1, 14)
>>> d = date.today() - birthday
>>> type(d)
>>> d.days
```
- ```
>>> from datetime import timedelta
>>> birthday + timedelta(1)
>>> day1 = birthday + timedelta(1)
>>> day1
>>> day2 = day1 + timedelta(1)
>>> day2
```
- ```
>>> today = date.today()
>>> today
>>> today.weekday()
>>> tomorrow = today + timedelta(1)
>>> tomorrow.weekday()
>>> tomorrow.day
```

Make sure that you understand why the different results are generated.

## Example

```
>>> from datetime import date
>>> unluckydays(date(2012, 1, 1), date(2012, 12, 31))
3
>>> unluckydays(date(2012, 1, 1), date(2012, 12, 31), 14, 5)
3
>>> unluckydays(date(2012, 1, 1), date(2012, 12, 31), 17, 4)
2
>>> unluckydays(date(2012, 1, 1), date(2012, 12, 31), 13, 1)
2
>>> unluckydays(
... startdate=date(2012, 1, 1),
... daynumber=1,
... weekdaynumber=0,
... enddate=date(2012, 12, 31)
...)
1
```

Vrijdag de dertiende wordt in onze cultuur algemeen gezien als een ongeluksdag. Vanwaar dit bijgeloof komt, is niet direct geweten en hoewel er allerlei verklaringen zijn in verschillende godsdiensten en gebruiken, lijkt het er op dat dit bijgeloof nog maar hooguit een dikke 100 jaar oud is.

Ook is de combinatie van vrijdag en het getal dertien niet zo universeel als je wel zou denken. In onder andere België, Nederland en Engeland wordt inderdaad vrijdag de dertiende als ongeluksdag gezien. In Griekenland, Spanje en Latijns-Amerika is het echter dinsdag de dertiende dat een ongeluksdag zou moeten zijn. In Italië is iedereen dan weer extra voorzichtig op vrijdag de zeventiende.

## Opgave

Schrijf een functie ongeluksdagen die één verplicht argument heeft en drie optionele argumenten. Het verplichte argument geeft een startdatum en de optionele argumenten geven respectievelijk een einddatum, een dagnummer en een weekdagnummer. De functie moet als resultaat het aantal dagen tussen de begindatum en de einddatum (inclusief de eindpunten) teruggeven die een dagnummer hebben dat op de opgegeven weekday valt. Als enkel het verplichte argument wordt meegegeven, wordt het aantal vrijdag de dertiende tussen de begindatum en vandaag berekend. Indien de begindatum recenter is dan de einddatum, dan liggen er geen dagen tussen de begindatum en de einddatum, en moet de functie logischerwijs de waarde 0 als resultaat teruggeven.

## Vorbereiding

In deze opgave moet je gebruik maken van de gegevenstypes `date` en `timedelta` die gedefinieerd worden in de module `datetime` van de [Python Standard Library](#). Voor je aan de eigenlijke opgave begint, kan je best eerst nagaan hoe Python reageert als je achtereenvolgens de volgende instructies uitvoert binnen een interactieve Python sessie:

```
1. >>> from datetime import date
 >>> geboortedatum = date(1983, 1, 14)
 >>> d = date.today() - geboortedatum
 >>> type(d)
```

```
>>> d.days
```

```
2. >>> from datetime import timedelta
>>> geboortedatum + timedelta(1)
>>> dag1 = geboortedatum + timedelta(1)
>>> dag1
>>> dag2 = dag1 + timedelta(1)
>>> dag2
```

```
3. >>> vandaag = date.today()
>>> vandaag
>>> vandaag.weekday()
>>> morgen = vandaag + timedelta(1)
>>> morgen.weekday()
>>> morgen.day
```

Zorg er zeker voor dat je begrijpt waarom de verschillende resultaten gegeneerd worden.

## Voorbeeld

```
>>> from datetime import date
>>> ongeluksdagen(date(2012, 1, 1), date(2012, 12, 31))
3
>>> ongeluksdagen(date(2012, 1, 1), date(2012, 12, 31), 14, 5)
3
>>> ongeluksdagen(date(2012, 1, 1), date(2012, 12, 31), 17, 4)
2
>>> ongeluksdagen(date(2012, 1, 1), date(2012, 12, 31), 13, 1)
2
>>> ongeluksdagen(
... startdatum=date(2012, 1, 1),
... dagnummer=1,
... weekdagnummer=0,
... einddatum=date(2012, 12, 31)
...)
1
```