

Unpredictable birthdays

Birthdays always have been so predictable: someone who was born on October 5th is doomed to celebrate his anniversary each year on October 5th. That's why we have devised some alternative ways to celebrate your birthday. For example, if you were born on the 5th day of the month, which was a Wednesday, you could celebrate every 5th day of the month that also falls on a Wednesday. Or you might celebrate every hundredth day that you are alive. Moreover, these alternatives also solve the problem that people born on February 29th can only celebrate their birthdays once every four years.



Humpty Dumpty wearing the cravat he received as an unbirthday present from the White King and Queen. From the book *Through the Looking-Glass* by Lewis Carroll, with illustrations by John Tenniel.

If you can not get enough of it to celebrate your birthday, then this might be something for you. An **unbirthday** is celebrated on any of the 364 (365 in leap years) days in which it is not a person's anniversary. It is a neologism coined by Lewis Carroll in his *Through the Looking-Glass*, giving rise to [The Unbirthday Song](#) in the 1951 Disney animated feature film *Alice in Wonderland*.

Assignment

The first part of your task is to implement four functions that can be used to determine whether or not someone is celebrating his (alternative) birthday on a given date. This is indicated by the Boolean value returned by these functions. All functions take two dates (`datetime.date` objects): the date of birth of a person and the date of which must be determined whether it is an (alternative) anniversary.

- A function `birthday` that celebrates a birthday on the given date according to the usual definition of anniversaries.
- A function `sameweekday` that celebrates a birthday on the given date if it falls on the same day of the month and the same weekday as the given date of birth.
- A function `hundredday` that celebrates a birthday on the given date if the number of days since the given date of birth is a multiple of 100.
- A function `unbirthday` that celebrates a birthday on the given date if someone born on the given date of birth does not celebrate his anniversary.

In addition, you must also write a function `birthdays` that must return a tuple of all (alternative)

birthdays that fall within the period between given start and end dates (including both dates). The birthdays must be listed in chronological order. The function takes as an argument the date of birth of the person whose birthdays must be determined.

The function `birthdays` also has two optional parameters `start` and `end`, that can be used to pass the start and/or end dates to the function. If no start date is explicitly passed when calling the function, the given date of birth must be used as the default value. If no end date is explicitly passed when calling the function, today's date must be used as the default value.

The function `birthdays` also has an optional parameter `birthday`, that can be used to pass a function that must be used to determine whether or not a given date is an (alternative) birthday. This function must have the same two parameters as the first four functions you have implemented for this assignment (to which a date of birth and another date must be passed respectively) and must return a Boolean value that indicates whether or not the given date is a birthday for someone born on the given date of birth. If no value is explicitly passed to the parameter `birthday` when calling the function `birthdays`, the function must return the list of anniversaries that fall within the period.

Example

```
>>> from datetime import date
```

```
>>> birthday(date(1969, 10, 5), date(2015, 10, 5))
```

```
True
```

```
>>> birthday(date(1969, 10, 5), date(1989, 10, 4))
```

```
False
```

```
>>> sameweekday(date(1969, 10, 5), date(2002, 5, 5))
```

```
True
```

```
>>> sameweekday(date(1969, 10, 5), date(1989, 10, 4))
```

```
False
```

```
>>> hundredday(date(1969, 10, 5), date(1975, 10, 14))
```

```
True
```

```
>>> hundredday(date(1969, 10, 5), date(1989, 10, 4))
```

```
False
```

```
>>> unbirthday(date(1969, 10, 5), date(2015, 10, 5))
```

```
False
```

```
>>> unbirthday(date(1969, 10, 5), date(1989, 10, 4))
```

```
True
```

```
>>> birthdays(date(1969, 10, 5), end=date(1972, 1, 1))
```

```
(datetime.date(1969, 10, 5), datetime.date(1970, 10, 5), datetime.date(1971, 10, 5))
```

```
>>> birthdays(date(1969, 10, 5), birthday=sameweekday, start=date(2014, 1, 1))
```

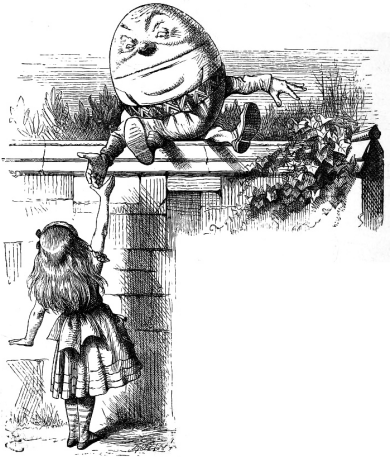
```
(datetime.date(2014, 1, 5), datetime.date(2014, 10, 5), datetime.date(2015, 4, 5), datetime.date(2015, 7, 5))
```

```
>>> birthdays(date(1969, 10, 5), start=date(1975, 1, 1), end=date(1976, 1, 1), birthday=hundredday)
```

```
(datetime.date(1975, 3, 28), datetime.date(1975, 7, 6), datetime.date(1975, 10, 14))
```

Verjaardagen zijn zo voorspelbaar: als je geboren bent op 5 oktober, dan ben je voor de rest van je leven gedoemd om ook op 5 oktober je verjaardag te vieren. Daarom hebben we enkele alternatieve manieren bedacht om je geboortedag te vieren. Als je op de vijfde van de maand geboren bent en dat was een woensdag, dan zou je bijvoorbeeld elke vijfde van de maand kunnen vieren als die ook op een woensdag valt. Of je zou bijvoorbeeld elke honderdste dag dat

je leeft kunnen vieren. Bovendien lossen deze alternatieven ook het probleem op dat mensen die op 29 februari geboren zijn anders maar om de vier jaar hun verjaardag kunnen vieren.



Humpty Dumpty draagt de stropdas die hij als geschenk voor zijn onjaardag gekregen heeft van de Witte Koning en Koningin. Uit het boek *Through the Looking-Glass* van Lewis Carroll, met illustraties van John Tenniel.

Als je er niet genoeg van kunt krijgen om je geboortedag te vieren, dan is dit misschien wel iets voor jou. Een **onjaardag** wordt gevierd op elk van de 364 (of 365 in het geval van een schrikkeljaar) dagen waarop een persoon niet zijn verjaardag viert. Het is een neologisme dat werd uitgevonden door Lewis Carroll voor zijn boek *Through the Looking-Glass*, en gaf aanleiding tot [The Unbirthday Song](#) in de Disney animatiefilm *Alice in Wonderland* uit 1951.

Opgave

Je opdracht bestaat er eerst en vooral in om vier functies te schrijven waarmee kan bepaald worden of iemand op een gegeven datum zijn (alternatieve) verjaardag viert. Dit wordt aangegeven door de Booleaanse waarde die door de functies teruggegeven wordt. Aan deze functies moeten telkens twee datums doorgegeven (datetime.date objecten): de geboortedatum van de persoon en de datum waarvan moet bepaald worden of het al dan niet een verjaardag is.

- Een functie verjaardag waarmee kan bepaald worden of op de gegeven datum iemand met de gegeven geboortedatum zijn (gewone) verjaardag viert.
- Een functie verweekdag die de gegeven datum als een verjaardag beschouwt als die op dezelfde dag van de maand en op dezelfde weekdag valt als de gegeven geboortedatum.
- Een functie verhonderddag die de gegeven datum als een verjaardag beschouwt als het aantal dagen sinds de gegeven geboortedatum een veelvoud is van 100.
- Een functie onjaardag die de gegeven datum als een verjaardag beschouwt als die niet samenvalt met de (gewone) verjaardag van iemand met de gegeven geboortedatum.

Schrijf nu ook nog een functie verjaardagen die een tuple teruggeeft met alle (alternatieve) verjaardagen die vallen tussen een startdatum en een einddatum (inclusief deze datums zelf). De verjaardagen moeten daarbij in chronologische volgorde opgelijst worden. Aan de functie moet de geboortedatum doorgegeven worden van de persoon waarvan de verjaardagen moeten bepaald worden.

De functie verjaardagen heeft ook nog twee optionele parameters start en einde waaraan respectievelijk de startdatum en de einddatum kunnen doorgegeven worden. Als de startdatum

niet expliciet wordt opgegeven bij het aanroepen van de functie, dan moet de geboortedatum als startwaarde gebruikt worden. Als de einddatum niet expliciet wordt opgegeven bij het aanroepen van de functie, dan moet de datum van vandaag als eindwaarde gebruikt worden.

De functie `verjaardagen` heeft ook nog een optionele parameter `verjaardag`, waaraan een functie kan doorgegeven worden die moet gebruikt worden om de verjaardagen te bepalen. Deze functie moet dezelfde twee parameters hebben als de eerste vier functies die je voor deze opgave geschreven (waaraan respectievelijk een geboortedatum en een datum moeten doorgegeven worden) en moet een Booleaanse waarde teruggeven die aangeeft of de gegeven datum al dan niet een verjaardag is voor iemand met de gegeven geboortedatum. Als er bij het aanroepen van de functie `verjaardagen` niet expliciet een waarde wordt doorgegeven aan de parameter `verjaardag`, dan moet de functie een lijst met alle gewone verjaardagen teruggeven die binnen de periode vallen.

Voorbeeld

```
>>> from datetime import date
```

```
>>> verjaardag(date(1969, 10, 5), date(2015, 10, 5))
```

```
True
```

```
>>> verjaardag(date(1969, 10, 5), date(1989, 10, 4))
```

```
False
```

```
>>> verweekdag(date(1969, 10, 5), date(2002, 5, 5))
```

```
True
```

```
>>> verweekdag(date(1969, 10, 5), date(1989, 10, 4))
```

```
False
```

```
>>> verhonderddag(date(1969, 10, 5), date(1975, 10, 14))
```

```
True
```

```
>>> verhonderddag(date(1969, 10, 5), date(1989, 10, 4))
```

```
False
```

```
>>> onjaardag(date(1969, 10, 5), date(2015, 10, 5))
```

```
False
```

```
>>> onjaardag(date(1969, 10, 5), date(1989, 10, 4))
```

```
True
```

```
>>> verjaardagen(date(1969, 10, 5), einde=date(1972, 1, 1))
```

```
(datetime.date(1969, 10, 5), datetime.date(1970, 10, 5), datetime.date(1971, 10, 5))
```

```
>>> verjaardagen(date(1969, 10, 5), verjaardag=verweekdag, start=date(2014, 1, 1))
```

```
(datetime.date(2014, 1, 5), datetime.date(2014, 10, 5), datetime.date(2015, 4, 5), datetime.date(2015, 7, 5))
```

```
>>> verjaardagen(date(1969, 10, 5), start=date(1975, 1, 1), einde=date(1976, 1, 1), verjaardag=verhonderddag)
```

```
(datetime.date(1975, 3, 28), datetime.date(1975, 7, 6), datetime.date(1975, 10, 14))
```