

Stardate

A **stardate** is a fictional system of time measurement developed for the television and film series Star Trek. In the series, use of this data system is commonly heard at the beginning of a voiceover log entry.

While the general idea resembles the [Julian day](#) currently used by astronomers, writers and producers have selected numbers using different methods over the years, some more arbitrary than others. This makes it impossible to convert all stardates into equivalent calendar dates, especially since stardates were originally intended to disguise the precise era of Star Trek. The following excerpt from the *Star Trek Guide* (April 17, 1967, p. 25) instructs writers for the [original Star Trek TV series](#) on how to select stardates for their scripts.

We invented "Stardate" to avoid continually mentioning Star Trek's century (actually, about two hundred years from now), and getting into arguments about whether this or that would have developed by then. Pick any combination of four numbers plus a percentage point, use it as your story's stardate. For example, 1313.5 is twelve o'clock noon of one day and 1314.5 would be noon of the next day. Each percentage point is roughly equivalent to one-tenth of one day. The progression of stardates in your script should remain constant but don't worry about whether or not there is a progression from other scripts. Stardates are a mathematical formula which varies depending on location in the galaxy, velocity of travel, and other factors, can vary widely from episode to episode.

According to creator [Gene Roddenberry](#), the stardate has initially been devised to prevent that events in the television series could be pinpointed to a specific year. He gave the following explanation about this:

"This time system adjusts for shifts in relative time which occur due to the vessel's speed and space warp capability. It has little relationship to Earth's time as we know it. One hour aboard the USS Enterprise at different times may equal as little as three Earth hours. The stardates specified in the log entry must be computed against the speed of the vessel, the space warp, and its position within our galaxy, in order to give a meaningful reading."

Roddenberry later told that he did not fully understand this explanation himself, but that others

considered it a logical explanation:

"I'm not quite sure what I meant by that explanation, but a lot of people have indicated it makes sense. If so, I've been lucky again, and I'd just as soon forget the whole thing before I'm asked any further questions about it."

Because there's no clear reasoning behind the stardate, throughout the series and films several examples of events are found that happen later in time but have a lower stardate than something that happened earlier.

Assignment

In this assignment we use two different methods to express dates in our era as stardates. The **old method** roughly corresponds to the stardates used in the first six Star Trek motion pictures. The method represents a date in our era in the format `YYMM.DD`, where `YY` corresponds to the year minus 1900, `MM` to the month (01-12) and `DD` to the day of the month (01-31). The values are always expressed with two digits, using a leading zero if necessary. As a result, September 8, 1966 is represented using the old method as stardate `6609.08`. Another digit must be added for years after 1999, such that September 11, 2015 is represented using the old method as stardate `11509.11`.

Starting from the motion picture [Star Trek XI](#) a **new method** was used to convert dates from our era into stardates. Following this new method, stardates are expressed in the format `YYYY.xx`, where `YYYY` is the actual for-digit year, and `.xx` expresses the fraction of the year that has passed to two decimal places (i.e., hundredths of a year). The last two digits may thus be determined by calculating the number of days since January 1 of the given year (no days have passed on January 1, one day has passed on January 2, and so on) times one hundred and dividing this result by the total number of days in the year (taking into account leap years). If the decimal places are left out, this results in an integer between 0 and 99 (including the boundaries). As a result, January 1, 2015 corresponds to stardate `2015.00` and July 4, 2015 to stardate `2015.50` (halfway through the year 2015).

Define a class `Stardate` that can be used to represent stardates in Python. This class must at least support the following methods:

- An initialisation method that takes a date as its argument (a `datetime.date` object). The given data represents the date from our era that corresponds with the stardate represented by the object. In case no date is passed to the initialisation method, the object represents that stardate that corresponds to the current date.
- A method `__repr__` that returns a string representation of the stardate, corresponding the Python statement that needs to be used to create a stardate that is equivalent with the current object. This representation must be formatted as `Stardate(date)`, where `date` is the string representation of the corresponding date from our era that is returned from the built-in function `repr`.
- A method that `__str__` returns a string representation of the stardate, that either corresponds to the representation according to the old method or de the new method as described above (also see the description of the method `switch` below). Immediately after initialisation of the object, a representation according to the new method must be returned.
- A method `switch` that takes no arguments. If this method is called, the method `__str__` should switch between the old method and the new method for the representation of the stardate.

This means that if the old method was used before calling the method `switch`, the new method must be used after calling it, and *vice versa*.

Example

```
>>> import datetime
>>> date = Stardate(datetime.date(2015, 9, 11))
>>> repr(date)
'Stardate(datetime.date(2015, 9, 11))'
>>> date
Stardate(datetime.date(2015, 9, 11))
>>> str(date)
'2015.69'
>>> print(date)
2015.69
>>> date.switch()
>>> date
Stardate(datetime.date(2015, 9, 11))
>>> print(date)
11509.11
>>> date.switch()
>>> date
Stardate(datetime.date(2015, 9, 11))
>>> print(date)
2015.69

>>> today = Stardate()
>>> today
Stardate(datetime.date(2017, 6, 10))
>>> print(today)
2017.44

>>> print(Stardate(datetime.date(2015, 1, 1)))
2015.00
>>> print(Stardate(datetime.date(2015, 7, 4)))
2015.50
>>> print(Stardate(datetime.date(2015, 7, 5)))
2015.50
>>> print(Stardate(datetime.date(2015, 7, 6)))
2015.50
>>> print(Stardate(datetime.date(2015, 7, 7)))
2015.51
>>> print(Stardate(datetime.date(2015, 12, 31)))
2015.99
```

Resources

- **Onion R (1967)**. Pages from the Official Star Trek Writers' Guide. *Slate Magazine*. [🔗](#)

Een **sterrendatum** (Engels: *stardate*) is een fictieve tijdseenheid die ontwikkeld werd voor de Star Trek televisieseries en films. Bij aanvang van bijna elke aflevering hoort men een commentaarstem het logboek inspreken, waarbij de sterrendatum van de aflevering vermeldt wordt.

Alhoewel het algemene idee achter de sterrendatum sterk lijkt op de [juliaanse dag](#) zoals die door sterrenkundigen gebruikt wordt, hebben schrijvers en producenten van de Star Trek televisieseries en films doorheen de jaren verschillende methoden gebruikt om sterrendatums aan afleveringen of gebeurtenissen toe te kennen, sommige al meer willekeurig dan andere. Dit maakt het onmogelijk om alle sterrendatums om te zetten naar onze tijdrekening, vooral omdat de sterrendatum oorspronkelijk ingevoerd werd om het precieze tijdperk waarin Star Trek zich afspeelt vaag te houden. Het volgende uittreksel uit de *Star Trek Guide* (17 april 1967, p. 25) geeft instructies aan de schrijvers van [de oorspronkelijke Star Trek televisieserie](#) over hoe ze sterrendatums moeten kiezen voor hun scripts.

We invented "Stardate" to avoid continually mentioning Star Trek's century (actually, about two hundred years from now), and getting into arguments about whether this or that would have developed by then. Pick any combination of four numbers plus a percentage point, use it as your story's stardate. For example, 1313.5 is twelve o'clock noon of one day and 1314.5 would be noon of the next day. Each percentage point is roughly equivalent to one-tenth of one day. The progression of stardates in your script should remain constant but don't worry about whether or not there is a progression from other scripts. Stardates are a mathematical formula which varies depending on location in the galaxy, velocity of travel, and other factors, can vary widely from episode to episode.

Volgens bedenker [Gene Roddenberry](#) werd de sterrendatum bedacht om te voorkomen dat de gebeurtenissen in de televisieserie naar een specifiek jaar konden herleid worden. Later gaf hij de volgende uitleg:

"This time system adjusts for shifts in relative time which occur due to the vessel's speed and space warp capability. It has little relationship to Earth's time as we know it. One hour aboard the USS Enterprise at different times may equal as little as three Earth hours. The stardates specified in the log entry must be computed against the speed of the vessel, the space warp, and its position within our galaxy, in order to give a meaningful reading."

Later verklaarde Roddenberry dat hij die uitleg zelf niet helemaal begreep, maar dat anderen het een logische verklaring vonden:

"I'm not quite sure what I meant by that explanation, but a lot of people have indicated it makes sense. If so, I've been lucky again, and I'd just as soon forget the whole thing"

before I'm asked any further questions about it."

Omdat er niet echt een duidelijke redenering achter de sterrendatum zit, is het in de loop van de series en films wel vaker voorgekomen dat gebeurtenissen die later in de tijd gebeurden toch een lagere sterrendatum hadden dan iets wat eerder gebeurde.

Opgave

In deze opgave maken we gebruik van twee methoden om een dag uit onze tijdsrekening uit te drukken als een sterrendatum. De **oude methode** correspondeert ongeveer met de sterrendatums die gebruikt werden in de eerste zes Star Trek films. Hierbij wordt een datum uit onze tijdsrekening voorgesteld in het formaat YYMM.DD, waarbij YY staat voor het jaar min 1900, MM voor de maand (01-12) en DD voor de dag van de maand (01-31). De waarden worden altijd uitgedrukt met twee cijfers door eventueel gebruik te maken van een voorloopnul. Zo wordt bijvoorbeeld 8 september 1966 volgens de oude methode voorgesteld door de sterrendatum 6609.08. Voor datums later dan het jaar 1999 moet er een extra cijfers voor de jaren gebruikt worden, waardoor bijvoorbeeld 11 september 2015 volgens de oude methode voorgesteld wordt door de sterrendatum 11509.11.

Vanaf de film [Star Trek XI](#) werd een **nieuwe methode** in gebruik genomen om datums uit onze jaarrekening voor te stellen als een sterrendatum. Hierbij worden datums uitgedrukt in het formaat YYYY.xx, waarbij YYYY staat voor het jaartal uitgedrukt met vier cijfers en .xx voor gepasseerde deel van het jaar uitgedrukt als twee decimale cijfers (aantal honderdsten van een jaar). De laatste twee cijfers kunnen dus bepaald worden door het aantal verstreken dagen te bepalen sinds 1 januari van het jaar (op 1 januari zijn er geen dagen verstreken, op 2 januari is er één dag verstreken, enzoverder), en dit aantal te vermenigvuldigen met honderd en te delen door het totaal aantal dagen van het jaar (rekening houdend met schrikkeljaren). Als men de decimale cijfers buiten beschouwing laat, dan levert dit een waarde op tussen 0 en 99 (grenzen inbegrepen). Zo komt 1 januari 2015 overeen met sterrendatum 2015.00 en 4 juli 2015 met sterrendatum 2015.50 (halverwege het jaar 2015).

Definieer een klasse `Sterrendatum` waarmee sterrendatums kunnen voorgesteld worden in Python. Deze klasse moet ondersteuning bieden aan de volgende methoden:

- Een initialisatiemethode waaraan een datum kan doorgegeven worden (een `datetime.date` object). De gegeven datum stelt de datum in onze tijdsrekening voor die overeenkomt met de sterrendatum die door het object moet voorgesteld worden. Indien geen datum wordt doorgegeven aan de initialisatiemethode, dan stelt het object de sterrendatum voor die correspondeert met de datum van vandaag.
- Een methode `__repr__` die een stringvoorstelling van de sterrendatum teruggeeft, die overeenkomt met het Python statement dat moet gebruikt worden om een sterrendatum aan te maken die samenvalt met het huidige object. Deze voorstelling moet het formaat `Sterrendatum(datum)` hebben, waarbij `datum` de stringvoorstelling is van de corresponderende datum uit onze tijdsrekening die men bekomt aan de hand van de ingebouwde functie `repr`.
- Een methode `__str__` die een stringvoorstelling van de sterrendatum teruggeeft, die ofwel overeenkomt met de voorstelling volgens de oude of de nieuwe methode zoals hierboven beschreven (zie ook de omschrijving van de methode `wissel` hieronder). Onmiddellijk na initialisatie van het object moet de voorstelling volgens de nieuwe methode teruggegeven worden.

- Een methode `wissel` waaraan geen argumenten mogen doorgegeven worden. Als deze methode wordt aangeroepen, dan moet de methode `__str__` schakelen tussen de oude en de nieuwe methode voor de voorstelling van de sterrendatum. Dat wil zeggen dat als voorheen de oude methode gebruikt werd, na het aanroepen van de methode `wissel` de nieuwe methode moet gebruikt worden, en omgekeerd.

Voorbeeld

```
>>> import datetime
>>> datum = Sterrendatum(datetime.date(2015, 9, 11))
>>> repr(datum)
'Sterrendatum(datetime.date(2015, 9, 11))'
>>> datum
Sterrendatum(datetime.date(2015, 9, 11))
>>> str(datum)
'2015.69'
>>> print(datum)
2015.69
>>> datum.wissel()
>>> datum
Sterrendatum(datetime.date(2015, 9, 11))
>>> print(datum)
11509.11
>>> datum.wissel()
>>> datum
Sterrendatum(datetime.date(2015, 9, 11))
>>> print(datum)
2015.69

>>> vandaag = Sterrendatum()
>>> vandaag
Sterrendatum(datetime.date(2017, 6, 10))
>>> print(vandaag)
2017.44

>>> print(Sterrendatum(datetime.date(2015, 1, 1)))
2015.00
>>> print(Sterrendatum(datetime.date(2015, 7, 4)))
2015.50
>>> print(Sterrendatum(datetime.date(2015, 7, 5)))
2015.50
>>> print(Sterrendatum(datetime.date(2015, 7, 6)))
2015.50
>>> print(Sterrendatum(datetime.date(2015, 7, 7)))
2015.51
>>> print(Sterrendatum(datetime.date(2015, 12, 31)))
2015.99
```

Bronnen

- **Onion R (1967)**. Pages from the Official Star Trek Writers' Guide. *Slate Magazine*. [🔗](#)