

# Clock

The 24-hour clock is the convention of time keeping in which the day runs from midnight to midnight and is divided into 24 hours, indicated by the hours passed since midnight, from 0 to 23. This system is the most commonly used time notation in the world today, and is the international standard ([ISO 8601](#)) notation for time of day.



24-hour clock that indicates the time 23:30:45

A time of day is written in the 24-hour notation in the form `hh:mm:ss` (for example, `01:23:45`), where `hh` (00 to 23) is the number of full hours that have passed since midnight, `mm` (00 to 59) is the number of full minutes that have passed since the last full hour, and `ss` (00 to 59) is the number of seconds since the last full minute. A leading zero is added for numbers under 10. In the 24-hour time notation, the day begins at midnight, `00:00:00`, and the last second of the day falls at `23:59:59`.

## Assignment

Define a class `Clock` that can be used to create instances of 24-hour clocks. The objects of the class `Clock` need to support at least the following methods:

- An initialization method that has three optional parameters: `hours`, `minutes` and `seconds`. These parameters respectively give the number of full hours  $h$  ( $0 \leq h < 24$ ) that have passed since midnight, the number of full minutes  $m$  ( $0 \leq m < 60$ ) that have passed since the last full hour, and the number of seconds  $s$  ( $0 \leq s < 60$ ) since the last full minute. All parameters have zero as their default value. The initial time of the clock needs to be set based on the given values that are passed as arguments to the function.
- A method `__repr__` that returns a string representation of the object, corresponding to the Python expression that can be used to create a 24-hours clock that has the same time settings as the current clock. The string that is returned must have the format `Clock(h, m, s)`, where  $h$  ( $0 \leq h < 24$ ) represents the number of full hours since midnight,  $m$  ( $0 \leq m < 60$ ) represents the number of full minutes since the last full hour, and  $s$  ( $0 \leq s < 60$ ) represents the number of seconds since the last full minute.
- A method `__str__` that returns a string representation of the object in the format `hh:mm:ss` as described in the introduction.
- A method `set` that has three optional parameters: `hours`, `minutes` and `seconds`. These parameters have the same semantics and default values as the parameters of the initialization method. The new time of the clock needs to be set based on the given values that are passed as arguments to the function.
- A method `forward` that has three optional parameters: `hours`, `minutes` and `seconds`. The values

passed as arguments to these parameters respectively indicate the number of hours, minutes and seconds that clock needs to be set forward. In doing so, the method must take into account that after the last second of the day (23:59:59), the clock resets to 00:00:00.

## Example

```
>>> clock = Clock()
>>> clock
Clock(0, 0, 0)
>>> print(clock)
00:00:00

>>> clock.set(hours=7, minutes=5, seconds=41)
>>> clock
Clock(7, 5, 41)
>>> print(clock)
07:05:41

>>> clock.forward(minutes=57, seconds=39)
>>> print(clock)
08:03:20

>>> clock.forward(hours=20, minutes=8, seconds=48)
>>> print(clock)
04:12:08
```

De 24-uursklok is een conventie die gebruikt wordt om de tijd aan te duiden, waarbij een dag loopt vanaf middernacht tot middernacht en wordt onderverdeeld in 24 uren die worden aangegeven door het aantal uren dat verstreken is sinds middernacht: van 0 tot en met 23. Dit systeem is de meest gebruikte tijdsnotatie ter wereld, en is ook de internationale standaardnotatie ([ISO 8601](#)) voor een tijdstip binnen een dag.



24-uursklok die de tijd 23:30:45 aangeeft

Een tijdstip binnen een dag wordt in de 24-uursnotatie geschreven onder de vorm uu:mm:ss (bijvoorbeeld, 01:23:45), waarbij uu (00 tot 23) het aantal volledige uren is die verstreken zijn sinds middernacht, mm (00 tot 59) het aantal volledige minuten is die verstreken zijn sinds het laatste volledige uur, en ss (00 tot 59) het aantal seconden is die verstreken zijn sinds de laatste volledige minuut. Bij getallen kleiner dan 10 wordt een voorloopnul toegevoegd. In de 24-uursnotatie begint een dag om middernacht, 00:00:00, en valt de laatste seconde van de dag op 23:59:59.

## Opgave

Definieer een klasse `Klok` waarmee instanties van 24-uursklokken kunnen aangemaakt worden.

De objecten van de klasse `Klok` moeten ondersteuning bieden aan de volgende methoden:

- Een initialisatiemethode die drie optionele parameters heeft: uren, minuten en seconden. Deze parameters stellen respectievelijk het aantal volledige uren  $u$  ( $0 \leq u < 24$ ) sinds mindernacht, het aantal volledige minuten  $m$  ( $0 \leq m < 60$ ) sinds het laatste volledige uur, en het aantal seconden  $s$  ( $0 \leq s < 60$ ) sinds de laatste volledige minuut voor. Alle parameters hebben standaardwaarde nul. De initiële tijd van de klok moet ingesteld worden op basis van de waarden die als argument aan de functie doorgegeven worden.
- Een methode `__repr__` die een stringvoorstelling van het object teruggeeft, die overeenkomt met de Python expressie die kan gebruikt worden om 24-uursklok aan te maken die dezelfde tijdsinstelling heeft als de huidige klok. De string die wordt teruggegeven moet dus het formaat `Klok(u, m, s)` hebben, waarbij  $u$  ( $0 \leq u < 24$ ) het aantal volledige uren sinds mindernacht voorstelt,  $m$  ( $0 \leq m < 60$ ) het aantal volledige minuten sinds het laatste volledige uur voorstelt, en  $s$  ( $0 \leq s < 60$ ) het aantal seconden sinds de laatste volledige minuut voorstelt.
- Een methode `__str__` die een stringvoorstelling van het object teruggeeft in het formaat `uu:mm:ss` zoals beschreven in de inleiding.
- Een methode `instellen` die drie optionele parameters heeft: uren, minuten en seconden. Deze parameters hebben dezelfde betekenis en standaardwaarde als de parameters van de initialisatiemethode. De nieuwe tijd van de klok moet ingesteld worden op basis van de waarden die als argument aan de functie doorgegeven worden.
- Een methode `vooruit` die drie optionele parameters heeft: uren, minuten en seconden. De waarden die als argument aan deze parameters doorgegeven worden, geven respectievelijk het aantal uren, minuten en seconden aan die de klok moet vooruit gezet worden. Hierbij moet rekening gehouden worden dat na de laatste seconde van de dag (23:59:59), de klok terugspringt op 00:00:00.

## Voorbeeld

```
>>> klok = Klok()
>>> klok
Klok(0, 0, 0)
>>> print(klok)
00:00:00
```

```
>>> klok.instellen(uren=7, minuten=5, seconden=41)
>>> klok
Klok(7, 5, 41)
>>> print(klok)
07:05:41
```

```
>>> klok.vooruit(minuten=57, seconden=39)
>>> print(klok)
08:03:20
```

```
>>> klok.vooruit(uren=20, minuten=8, seconden=48)
>>> print(klok)
04:12:08
```