# Cowsay

The computer program cowsay generates ASCII pictures of a cow with a message.

```
+----------------------------+
| Moo may represent an idea, |
|  but only the cow knows.   |
+----------------------------+
        \   ^__^
         \  (oo)\_______
            (__)\       )\/\
                ||----w |
                ||     ||
```

It can also generate pictures using pre-made images of other animals such as Tux the Penguin, the Linux mascot.

```
+----------------------------+
| Moo may represent an idea, |
|  but only the cow knows.   |
+----------------------------+
   \
    \
        .--.
       |^_^ |
       |:_/ |
      //   \ \
     (|     | )
    /'\_   _/`\
    \___)=(___/
```

The computer program uses pre-made images that are stored in text files with the extension .cow, enabling it to produce different variants of "cows". Each such text file also contains two occurrences of a pair of braces ({}) enabling the program to produce different kinds of eyes. The program initially was a kind of *inside joke* within hacker culture, but has been around long enough that its use has become rather widespread.

## Assignment

Define a class Cowsay that can be used to generate ASCII pictures of a "cow" with a message. The objects of this class must at least support the following methods:

- An initialization method __init__ that takes a series (list or tuple) of strings as its argument. The given series of strings are the successive lines of the message said by the "cow". After initialization of the object, the message is never changed anymore.
- A method __repr__ that retuns a string representation of the object. This string representation contains the given series of strings that are centered across the width of the longest line of the message, with an additional space to the left and to the right. Use the string method center to center the lines of the message. A rectangular box must be drawn around the centered lines of the message. Determine the format of the box from the examples given below.
- A method setEyes that can be used to set a specific pair of eyes for the "cow". The method takes a single argument. If the argument passed to the method is not a two-character string,

an AssertionError must be raised with the message invalid eyes. In generating the ASCII picture, the first character of the argument must be used to replace the first pair of braces in the .cow file containing the pre-made image, and the second character to replace the second pair of braces. If an ASCII picture is generated for the object before this method is called a first time on the object, by default the lowercase letter o must be used twice to represent the eyes.

- A method setImage that can be used to set a specific pre-made image that is used when generating an ASCII picture for the current object. The method takes the location of a .cow file. If an ASCII picture is generated for the object before this method is called a first time on the object, by default the pre-made image must be used that is stored in the text file cow.cow. The source code may assume that this file is located in the current directory.
- A method __str__ that retuns a string containing the generated ASCII picture of the object. This string representation is a multiline string, whose first few lines correspond to the string representation as generated by the method __repr__, followed by the lines of the last pre-made image set for the object (or the pre-made image stored in the file cow.cow in case no explicit image was set for the object). In the lines containing the pre-made image, the eyes must be replaced by the last pair of eyes set for the object (or twice the lowercase letter o in case no eyes were set explicitly for the object).

## Example

In the following interactive session we assume that the text files cow.cow and tux.cow are in the current directory. Note that a docstring is nothing but an ordinary string, which requires that each backslash that occurs literally in the docstring must be repeated twice. Click here to convert the following interactive session into a docstring.

```
>>> quote = Cowsay(['Moo may represent an idea,', 'but only the cow knows.'])
>>> quote
+--------------------------+
| Moo may represent an idea, |
|  but only the cow knows.   |
+--------------------------+
>>> print(quote)
+--------------------------+
| Moo may represent an idea, |
|  but only the cow knows.   |
+--------------------------+
        \   ^__^
         \  (oo)\_______
            (__)\       )\/\
                ||----w |
                ||     ||
>>> quote.setEyes(33)
Traceback (most recent call last):
AssertionError: invalid eyes
>>> quote.setEyes('***')
Traceback (most recent call last):
AssertionError: invalid eyes
>>> quote.setEyes('^^')
>>> print(quote)
+--------------------------+
| Moo may represent an idea, |
|  but only the cow knows.   |
+--------------------------+
```

```
        \   ^__^
         \  (^^)\_______
            (__)\       )\/\
                ||----w |
                ||     ||
>>> quote.setImage('tux.cow')
>>> print(quote)
+----------------------------+
| Moo may represent an idea, |
|  but only the cow knows.   |
+----------------------------+
   \
    \
        .--.
       |^_^ |
       |:_/ |
      //   \ \
     (|     | )
    /'\_   _/`\
    \___)=(___/
Traceback (most recent call last):
AssertionError: invalid eyes
```

Het computerprogramma cowsay genereert ASCII-afbeeldingen van een koe met een bijhorend bericht.

```
+----------------------------+
| Moo may represent an idea, |
|  but only the cow knows.   |
+----------------------------+
        \   ^__^
         \  (oo)\_______
            (__)\       )\/\
                ||----w |
                ||     ||
```

Daarnaast kan het ook afbeeldingen genereren door gebruik te maken van voorgevormde tekeningen van andere dieren zoals Tux de Pinguïn, de mascotte van het Linux besturingssysteem.

```
+----------------------------+
| Moo may represent an idea, |
|  but only the cow knows.   |
+----------------------------+
   \
    \
        .--.
       |^_^ |
       |:_/ |
      //   \ \
     (|     | )
    /'\_   _/`\
    \___)=(___/
```

Het computerprogramma maakt gebruik van voorgevormde tekeningen die worden opgeslaan in tekstbestanden die gebruik maken van de extensie .cow, waardoor het programma afbeeldingen kan genereren van andere soorten "koeien". Elk tekstbestand bevat ook twee voorkomens van een paar accolades ({}) waardoor er ook verschillende soorten "ogen" kunnen gegenereerd

worden. Het programma was oorspronkelijk een soort *inside joke* binnen de wereld van de hackers, maar het bestaat al zo lang dat het gebruik ervan ondertussen wijd verspreid is.

## Opgave

Schrijf een klasse Cowsay waarmee ASCII-afbeeldingen kunnen gegenereerd worden van een "koe" met een bijhorend bericht. De objecten van deze klasse moeten minstens de volgende methoden hebben:

- Een initialisatiemethode __init__ waaraan een reeks (lijst of tuple) van strings moet doorgegeven worden. Deze strings vormen de opeenvolgende regels van het bericht dat door de "koe" moet uitgesproken worden. Na initialisatie van het object wordt dit bericht niet meer gewijzigd.
- Een methode __repr__ die een stringvoorstelling van het object teruggeeft. Deze stringvoorstelling bestaat uit de reeks strings die gecentreerd worden uitgeschreven over de breedte van de langste string uit het bericht, met telkens nog een extra spatie links en rechts. Gebruik de stringmethode center om de strings te centreren. Rond de gecentreerde regels van het bericht moet een rechthoekig kader getekend worden. Leidt de opmaak van dit kader af uit onderstaande voorbeelden.
- Een methode veranderOgen waarmee de ogen van de "koe" kunnen ingesteld worden. Aan deze methode moet één enkel argument doorgegeven worden. Indien het opgegeven argument geen string is die bestaat uit twee karakters, dan moet een AssertionError opgeworpen worden met de boodschap ongeldige ogen. Bij het genereren van een ASCII-afbeelding moet het eerste karakter gebruikt worden ter vervanging van het eerste paar accolades in het .cow bestand met de voorgevormde tekening, en het tweede karakter ter vervanging van het tweede paar accolades. Indien er een ASCII-afbeelding gegenereerd wordt van het object voordat deze methode voor de eerste keer wordt aangeroepen, dan moet standaard twee keer de kleine letter o als oog gebruikt worden.
- Een methode veranderTekening waarmee een voorgevormde tekening kan ingesteld worden die moet gebruikt worden om een ASCII-afbeelding van het object te genereren. Aan deze methode moet de locatie van een .cow bestand doorgegeven worden. Indien er een ASCII-afbeelding gegenereerd wordt van het object voordat deze methode voor de eerste keer wordt aangeroepen, dan moet standaard gebruik gemaakt worden van de voorgevormde tekening in het bestand cow.cow. De broncode mag ervan uitgaan dat dit bestand zich in de huidige directory bevindt.
- Een methode __str__ die een stringvoorstelling van het object teruggeeft die de gegenereerde ASCII-afbeelding voorstelt. Deze stringvoorstelling bestaat uit een aantal opeenvolgende regels. De eerste regels daarvan komen overeen met de stringvoorstelling zoals die gegenereerd wordt door de methode __repr__, gevolgd door regels van de laatst ingestelde tekening (of de tekening uit cow.cow indien er nog niet expliciet een tekening werd ingesteld voor het object). In de regels van de tekening moeten de ogen vervangen worden door de laatste ingestelde ogen (of tweemaal door de kleine letter o indien er nog geen expliciete ogen werden ingesteld voor het object).

## Voorbeeld

Bij onderstaande interactieve sessie gaan we ervan uit dat de bestanden cow.cow en tux.cow zich in de huidige directory bevinden. Merk op dat een docstring een gewone string is, waardoor elke letterlijke backslash die voorkomt in de docstring moet verdubbeld worden. Klik hier om

onderstaande interactieve sessie om te vormen tot een docstring.

```
>>> spreuk = Cowsay(['Moo may represent an idea,', 'but only the cow knows.'])
>>> spreuk
+----------------------------+
| Moo may represent an idea, |
|  but only the cow knows.   |
+----------------------------+
>>> print(spreuk)
+----------------------------+
| Moo may represent an idea, |
|  but only the cow knows.   |
+----------------------------+
        \   ^__^
         \  (oo)\_______
            (__)\       )\/\
                ||----w |
                ||     ||
>>> spreuk.veranderOgen(33)
Traceback (most recent call last):
AssertionError: ongeldige ogen
>>> spreuk.veranderOgen('***')
Traceback (most recent call last):
AssertionError: ongeldige ogen
>>> spreuk.veranderOgen('^^')
>>> print(spreuk)
+----------------------------+
| Moo may represent an idea, |
|  but only the cow knows.   |
+----------------------------+
        \   ^__^
         \  (^^)\_______
            (__)\       )\/\
                ||----w |
                ||     ||
>>> spreuk.veranderTekening('tux.cow')
>>> print(spreuk)
+----------------------------+
| Moo may represent an idea, |
|  but only the cow knows.   |
+----------------------------+
   \
    \
        .--.
       |^_^ |
       |:_/ |
      //   \ \
     (|     | )
    /'\_   _/`\
    \___)=(___/
>>> print(spreuk)
Traceback (most recent call last):
AssertionError: ongeldige ogen
```