

# Calendar

Set up a monthly calendar for a given month in a given year. It is not allowed to use any modules from *The Python Standard Library*.

February 2012						
Su	Mo	Tu	We	Th	Fr	Sa
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29			

## Assignment

1. Write a function `leapyear` that prints a Boolean value for a given year  $y$  (that is given to the function as an integer), indicating whether  $y$  is a leap year or not. Leap years are years that can be divided by 4. Keep in mind that only last years of a century that are divisible by 400 (such as 2000) are leap years, whilst if the last year of a century isn't divisible by 400 (like 1700, 1800, 1900) it is not a leap year.
2. Write a function `weekday` that indicates for a given day  $d$ , month  $m$  and year  $y$  the day of the week this date occurs on. This can be calculated as follows:

```
\[
\begin{array}{rcl}
y_0 & = & y - \frac{(14 - m)}{12} \\
x & = & y_0 + \frac{y_0}{4} - \frac{y_0}{100} + \frac{y_0}{400} \\
m_0 & = & m + 12 \left( \frac{14 - m}{12} \right) - 2 \\
d_0 & = & (d + x + \frac{31m_0}{12}) \mod 7
\end{array}
\]
```

Here, all fractures represent whole divisions (quotient) and `mod` stands for the modulo operator (rest after a whole division). The value  $d_0$  must be printed by the function, where 0 represents Sunday, 1 is Monday, 2 is Tuesday, and so on. The arguments  $d$ ,  $m$  and  $y$  must be given to the function as optional integer arguments. As standard value, you use the values that correspond with 2 February 2012.

3. Use the functions `leapyear` and `weekday` to write a function `calendar`, that prints the monthly calendar for a given month of a given year (both are given to the function as integer arguments). The first line of the calendar contains the English name of the month (in lowercase letters), followed by a space and the year, the whole text is centered over 20 positions. The next line contains consecutive week days, abbreviated to two lowercase letters, starting from Sunday and separated by spaces. After that, all days follow on different lines, neatly lined out under the correct day of the week they are on. The days are outlined on the right over two positions, with a space between the different days, in the same way it is done for the week days in the heading. The output of the calendar doesn't contain any empty lines, and there are no spaces at the end of a line.

## Example

```

>>> leapyear(2000)
True
>>> leapyear(2011)
False
>>> leapyear(2012)
True
>>> leapyear(2100)
False
>>> weekday(2, 8, 1953)
0
>>> weekday(21, 1, 2012)
6
>>> weekday()
4
>>> weekday(y=2011)
3
>>> calendar(1, 2012)
  January 2012
Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31
>>> calendar(2, 2012)
  February 2012
Su Mo Tu We Th Fr Sa
     1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29
>>> calendar(2, 2013)
  February 2013
Su Mo Tu We Th Fr Sa
     1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28

```

Stel een maandkalender op voor een gegeven maand in een gegeven jaar. Hierbij is het niet toegelaten om gebruik te maken van modules uit *The Python Standard Library*.

februari 2012						
zo	ma	di	wo	do	vr	za
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29			

## Opgave

1. Schrijf een functie `schrikkeljaar` die voor een gegeven jaar `$j$` (dat als integer argument aan de functie moet doorgegeven worden) een Booleaanse waarde teruggeeft die aangeeft of `$j$` een schrikkeljaar is of niet. Schrikkeljaren zijn jaren die deelbaar zijn door 4. Hou

echter rekening met het feit dat enkel eeuwjaar die deelbaar zijn door 400 (zoals 2000) schrikkeljaar zijn, terwijl eeuwjaar die niet deelbaar zijn door 400 (zoals 1700, 1800, 1900) geen schrikkeljaar zijn.

- Schrijf een functie `weekdag` die voor een gegeven dag `$d`, maand `$m` en jaar `$j` aangeeft op welke weekday deze dag valt. Dit kan op de volgende manier berekend worden:

```
\[
\begin{array}{rcl}
j_0 & = & j - \frac{(14 - m)}{12} \\
x & = & j_0 + \frac{j_0}{4} - \frac{j_0}{100} + \frac{j_0}{400} \\
m_0 & = & m + 12 \left( \frac{14 - m}{12} \right) - 2 \\
d_0 & = & (d + x + \frac{31m_0}{12}) \pmod{7}
\end{array}
\]
```

Hierbij stellen alle breuken gehele delingen voor (quotient) en staat `mod` voor de modulo operator (rest na gehele deling). De waarde `$d_0` moet door de functie teruggegeven worden, waarbij de waarde 0 staat voor zondag, 1 voor maandag, 2 voor dinsdag, enzoverder. De argumenten `$d`, `$m` en `$j` moeten als optionele integer argumenten aan de functie doorgegeven worden. Als standaardwaarden gebruik je de waarden die overeenkomen met 2 februari 2012.

- Gebruik de functies `schrikkeljaar` en `weekdag` om een functie `kalender` te schrijven, die de maandkalender voor een gegeven maand in een gegeven jaar (beide worden als integer argumenten aan de functie doorgegeven) uitschrijft. De eerste regel van de kalender bevat de nederlandsstalige naam van de maand (met kleine letters), gevolgd door een spatie en het jaartal, waarbij de volledige tekst gecentreerd wordt uitgeschreven over 20 posities. De volgende regel bevat de opeenvolgende weekdays, telkens afgekort tot twee kleine letters, startend met zondag en gescheiden door spaties. Daarna volgen op verschillende regels alle dagen van de maand, netjes uitgelijnd onder de juiste weekday waarop deze dagen vallen. De dagen worden telkens rechts uitgelijnd over twee posities uitgeschreven, met een spatie tussen de verschillende dagen zoals dat ook voor de weekdays in de hoofding gebruikt werd. De uitvoer van de kalender bevat geen lege regels, en er staan nooit spaties op het einde van een regel.

## Voorbeeld

```
>>> schrikkeljaar(2000)
True
>>> schrikkeljaar(2011)
False
>>> schrikkeljaar(2012)
True
>>> schrikkeljaar(2100)
False
>>> weekdag(2, 8, 1953)
0
>>> weekdag(21, 1, 2012)
6
>>> weekdag()
4
>>> weekdag(j=2011)
3
>>> kalender(1, 2012)
    januari 2012
```

zo ma di wo do vr za  
1 2 3 4 5 6 7  
8 9 10 11 12 13 14  
15 16 17 18 19 20 21  
22 23 24 25 26 27 28  
29 30 31

>>> kalender(2, 2012)

februari 2012

zo ma di wo do vr za

1 2 3 4  
5 6 7 8 9 10 11  
12 13 14 15 16 17 18  
19 20 21 22 23 24 25  
26 27 28 29

>>> kalender(2, 2013)

februari 2013

zo ma di wo do vr za

1 2  
3 4 5 6 7 8 9  
10 11 12 13 14 15 16  
17 18 19 20 21 22 23  
24 25 26 27 28