The persian-calendar package

Javad Razavian*

Version 0.80 (2026/11/15)

Abstract

persian-calendar converts dates between the Persian (Jalali, Solar Hijri) and Gregorian calendars inside LaTeX. It prints dates with Persian month names, gives day numbers you can do arithmetic on, and provides Persian month, weekday and year-animal names. It is built on the pgfcalendar package. The conversions work with any engine; printing the Persian names needs a Unicode engine and a Persian font, for example XeLaTeX with xepersian.

Contents

1 What the package does and does not do 1

2 Loading the package 2

3 Quick start 2

4 Date conversion 2

5 Day numbers and date arithmetic 3

6 Names 4

7 Calendars with TikZ 5

8 A year planner 9

9 A yearly to-do list 11

10 Recipes 12

11 How it works 13

12 Limitations and known issues 14

13 Version history 14

1 What the package does and does not do

The package is a date converter and a set of name tables. It provides:

- Gregorian → Jalali and Jalali → Gregorian conversion, printed as text or as numbers;
- Julian day numbers for Jalali dates, so you can add days, subtract dates or loop over days;

*Source code and bug reports: https://github.com/javadr/persian-calendar

- Persian names of the Jalali and Gregorian months and of the weekdays;
- the animal name of a year in the twelve-animal cycle;
- Jalali month grids and year calendars drawn with the TikZ `calendar` library (section 7);
- a two-page year planner in the manner of yplan (section 8);
- a yearly to-do list with Jalali week numbers, scaled to any paper size (section 9).

It has no holidays.

Requirements. The pgf bundle (which contains pgfcalendar) is all the conversions need, and they work with pdfLaTeX, XeLaTeX and LuaLaTeX. Only the names (and so `\pdate`, `\gdate` and the `\pcal…name` macros) are Persian text: they are stored as UTF-8 in the source file, so printing them needs an engine and packages that handle UTF-8 Persian letters, together with a font that has them. xepersian with XeLaTeX is what this manual uses, but any setup that can typeset Persian will do. Without one, use the numeric macros `\pisodate` and `\gisodate`, which print only digits.

2 Loading the package

TeX Live and MiKTeX install the package from CTAN, so nothing needs installing by hand.

Load the package in the preamble. The examples here use xepersian for the Persian text and load persian-calendar before it:

```
\usepackage{persian-calendar}
\usepackage{xepersian}
\settextfont{Parisa}
```

The `tikz` option. The Jalali calendars of section 7 need TikZ and its `calendar` library. The option `tikz` loads both:

```
\usepackage[tikz]{persian-calendar}
```

Without it the package loads only pgf and pgfcalendar, so a document that only converts dates loads faster. Loading tikz or the library yourself as well does no harm.

3 Quick start

```
\pdate{2015-03-21}          % Gregorian date in, Jalali text out
\gdate{1394-01-01}          % Jalali date in, Gregorian text out
\pisodate{2015-03-21}       % Gregorian in, Jalali as DD/MM/YYYY
\gisodate{1394-01-01}       % Jalali in, Gregorian as DD/MM/YYYY
```

which print:

`\pdate{2015-03-21}` ⇒ ۱۳۹۴ فروردین ۱
`\gdate{1394-01-01}` ⇒ ۲۰۱۵ مارس ۲۱
`\pisodate{2015-03-21}` ⇒ 01/01/1394
`\gisodate{1394-01-01}` ⇒ 21/03/2015

Today's date, in the Jalali calendar, is `\pdate{\year-\month-\day}`, which prints ۲۴ آبان ۱۴۰۵.

4 Date conversion

All four macros take one argument, a date written `Y-M-D`: a year, a month number and a day number joined by hyphens. The month and day may have a leading zero or not (`2015-3-21` and `2015-03-21` are the same).

`\pdate{Y-M-D}`

The argument is a *Gregorian* date. Prints the *Jalali* date as "day month-name year", for example ۱ فروردین ۱۴۰۳.

`\gdate{Y-M-D}`

The argument is a *Jalali* date. Prints the *Gregorian* date as "day month-name year", using the Persian transliteration of the Gregorian month, for example ۲۰ مارس ۲۰۲۴.

`\pisodate{Y-M-D}`

Like `\pdate` (Gregorian in, Jalali out) but prints numbers: day, month and year, each day and month padded to two digits and separated by a slash, so 01/01/1403. Note the order: this is day/month/year, not the ISO order year-month-day.

`\gisodate{Y-M-D}`

Like `\gdate` (Jalali in, Gregorian out) but prints numbers in the same day/month/year form, for example 20/03/2024.

Leap years. Esfand, the last Jalali month, has 30 days in a leap year:
`\pdate{2025-03-20}` ⇒ ۳۰ اسفند ۱۴۰۳
`\gdate{1403-12-30}` ⇒ ۲۰ مارس ۲۰۲۵
`\gdate{1404-12-30}` ⇒ ۲۱ مارس ۲۰۲۶ The last line is an invalid date, since 1404 is not a leap year; the package does not check this (section 12).

Wrap the numeric forms. Digits in Persian text are reordered by the bidirectional algorithm. To keep `\pisodate` and `\gisodate` in day/month/year order inside Persian text, put them in an `\lr` box, as this document does.

5 Day numbers and date arithmetic

The conversion macros above cannot add days. For that, the package works with Julian day numbers as defined by pgfcalendar: whole days counted from a fixed origin, so subtracting two numbers gives the number of days between two dates.

`\pcaldatetojulian{Y-M-D}{\count}`

Stores the day number of the *Jalali* date `Y-M-D` in the count register `\count`. Append `+N` to the date to add `N` days first: `1394-01-01+10` is ten days after 1 Farvardin 1394.

`\pcaljuliantodate{\count}{\y}{\m}{\d}`

Converts the day number in `\count` back to a *Jalali* date and puts the year, month and day into the macros `\y`, `\m` and `\d` (you may choose other names).

```
\newcount\jd
\pcaldatetojulian{1394-01-01+10}\jd
\pcaljuliantodate\jd\y\m\d
\number\y-\number\m-\number\d
```

prints 1394-1-11.

Read \y, \m, \d immediately. After `\pcaljuliantodate` the three macros are aliases of the package's internal result registers, not copies. Any later conversion (`\pdate`, `\gisodate`, and so on) changes them. If you need the values later, store them in your own counters first: `\newcount\myyear \myyear=\y`.

`\pcaldaysbetween{Y-M-D}{Y-M-D}{\count}`

Stores the number of days from the first Jalali date to the second in `\count`; it is negative when the second date is earlier. Both dates may end in an offset, as with `\pcaldatetojulian`.

```
\newcount\n
\pcaldaysbetween{1403-01-01}{1404-01-01}{\n}
\number\n
```

prints 366 (1403 is a leap year).

Offsets across a year end. Day arithmetic knows about leap years: 1403-12-29 plus one day is 1403-12-30, because 1403 is a leap year; plus two days is 1404-1-1.

Counts used internally. `\pcaldatetojulian` uses `\count1`, `\count2`, `\count3` and `\count8` as scratch registers. Do not keep values in those.

6 Names

`\pcalmonthname{n}`

Persian month name, `n` from 1 to 12. An argument outside this range prints nothing.

`\pcalmonthname{1}` ⇒ فروردین

All twelve, in order:

1	فروردین	4	تیر	7	مهر	10	دی
2	اردیبهشت	5	مرداد	8	آبان	11	بهمن
3	خرداد	6	شهریور	9	آذر	12	اسفند

`\pcalgregorianmonthname{n}`

Persian transliteration of the Gregorian month name, `n` from 1 to 12.

`\pcalgregorianmonthname{3}` ⇒ مارس

`\pcalweekdayname{n}`

Weekday name, `n` from 0 to 6, where **0 is Saturday** (the first day of the Persian week) and 6 is Friday.

0 شنبه 1 یکشنبه 2 دوشنبه 3 سه‌شنبه 4 چهارشنبه 5 پنجشنبه 6 جمعه

`\pcalweekdayshortname{n}`

One-letter weekday abbreviation, same numbering.

0 ش 1 ی 2 د 3 س 4 چ 5 پ 6 ج

`\pcalyearname{year}`

Name of a Jalali year in the twelve-animal cycle. `year` must be between 0 and 2099; other values print nothing.

`\pcalyearname{1403}` ⇒ نهنگ

The twelve animals in order, starting from year 1380:

1380	مار	1383	میمون	1386	خوک	1389	پلنگ	1392	مار	1395	میمون
1381	اسب	1384	مرغ	1387	موش	1390	خرگوش	1393	اسب	1396	مرغ
1382	گوسفند	1385	سگ	1388	گاو	1391	نهنگ	1394	گوسفند	1397	سگ

\pcalweekday{Y-M-D}{\count}

Stores the weekday of a Jalali date in `\count`, numbered like `\pcalweekdayname`: 0 is Saturday, 6 is Friday. As with `\pcaldatetojulian`, the date may end in an offset in days, and macros in it are expanded.

```
\newcount\wday
\pcalweekday{1403-01-01}{\wday}      % \wday = 4, Wednesday
\pcalweekday{1403-01-01+10}{\wday}   % ten days later: 0, Saturday
```

\pcaldateweekdayname{Y-M-D}

Prints the weekday name of a Jalali date: `\pcalweekdayname` of `\pcalweekday`.

`\pcaldateweekdayname{1403-01-01}` ⇒ چهارشنبه

7 Calendars with TikZ

The TikZ `calendar` library draws calendars with `\calendar`, but it only knows Gregorian months. The package adds a few keys that make it lay out Jalali months instead. They need tikz and the library, which the `tikz` option loads (section 2):

```
\usepackage[tikz]{persian-calendar}
...
\tikz\calendar[jalali dates=1405-01-01 to 1405-03-last,
               jalali week list, day xshift=-2.2em, day yshift=2.5ex,
               jalali month label above centered]
  if (Friday) [red] if (jalali equals=1-13) [blue];
```

فروردین 1405

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

اردیبهشت 1405

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

خرداد 1405

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

`jalali dates=Y-M-D to Y-M-D`

The range to draw, as Jalali dates. The day can be `last`, the last day of that month (29 or 30 for Esfand, depending on the leap year). It sets the library's `dates` key, so the two must not be mixed.

`jalali`

Switches the calendar to Jalali dates. The day number and the month label print the Jalali day and the Persian month name with the year (change them with `day text` and `month text`). Inside the calendar, the test `day of month=`n means the Jalali day, so the library's `month label …` styles fire on the first of each Jalali month. For every day, `\pcalcurrentyear`, `\pcalcurrentmonth`, `\pcalcurrentday` hold the Jalali date and `\pcalcurrentweekday` the weekday counted from Saturday = 0 (like `\pcalweekdayname`), so they can be used in `day code`.

`jalali week list`

Like the library's `week list`, but for Jalali months: `jalali` is included, a week runs from Saturday to Friday and a new month starts a new block. With the default positive `day xshift` Saturday is on the left; a negative `day xshift`, as above, puts it on the right.

`jalali month label above centered`

Like the library's `month label above centered`, which places the label with a fixed `1.5ex` offset and so leaves it slightly off centre. This one measures a two-digit day (`30`) in the current font and adds the inner sep of the day nodes, so the label is centred over the week at any font size. It assumes the library's default `every day` anchor (`base east`).

`jalali week columns`

The other way round, as in `calxxxx-yyyy`: one row per weekday, Saturday on top, and a new column after each Friday, so a month is four to six columns wide. A negative `day xshift` puts the first week on the right. `jalali day boxes` completes the first and last columns, `jalali weekday initials` puts the initials in a column beside the month, and `jalali month label above centered` centres over the month's own number of columns. The sample `persian-calendar-years.tex` uses it to print a range of years, nine to an A4 page: set `\FirstYear` and `\LastYear` at its top.

`jalali day boxes`

Draws every day as a box one day step wide (`day xshift`) and high (`day yshift`), so the boxes join into a grid. With `jalali week list`, empty boxes fill the days before the first and after the last day of each month (or of the range), so every month is a full rectangle of whole weeks. The text height and depth are fixed in `ex`, so all rows are level at any font size. `jalali month label above centered` then centres over the boxes. To fill some days, append to their `every day` style, as in the example below; a plain `fill=` in `if (…)` does not reach the box.

`jalali weekday initials`

Adds a row of weekday initials (ش, ی, …, ج, from `\pcalweekdayshortname`) one row above each month, in boxes when `jalali day boxes` is on. It works with `jalali week list`, and the month label moves up above it. Style the initials with `every jalali weekday`, for example `every jalali weekday/.style={font=\bfseries}`.

With either key, a new month starts at least three rows (four with initials) below the previous one, however small `month yshift` is, so an unfinished last week never runs into the next label.

```
\tikz\calendar[jalali dates=1405-01-01 to 1405-02-last,
  jalali week list, jalali day boxes, jalali weekday initials,
  day xshift=-0.9cm, day yshift=0.7cm,
  jalali month label above centered]
  if (Friday) [red, every day/.append style={fill=red!10}];
```

فروردین 1405

ج	پ	چ	س	د	ی	ش
7	6	5	4	3	2	1
14	13	12	11	10	9	8
21	20	19	18	17	16	15
28	27	26	25	24	23	22
				31	30	29

اردیبهشت 1405

ج	پ	چ	س	د	ی	ش
4	3	2	1			
11	10	9	8	7	6	5
18	17	16	15	14	13	12
25	24	23	22	21	20	19
	31	30	29	28	27	26

`jalali equals=M-D`

A date test that matches one Jalali month and day in every year, for example `jalali equals=1-1` for Nowruz. It can be used in `if (…)` and with `\pgfcalendarifdate`, also outside `jalali` calendars, as can `jalali day of month=`n.

Other layouts. `day list downward`, `day list upward`, `day list right` and `day list left` work together with `jalali`, because they only test `day of month=1`. The library's `week list` and `month list` use the Gregorian weekday and day, so use `jalali week list` instead.

A year by season. Draw one calendar per month with `\foreach` and place each with `at`: the month gives the column within its season, the season the row. The minus sign in `-\col` puts the later months further left, so the page reads right to left; drop it, and the minus of `day xshift`, for left to right. The horizontal step must be wider than seven days, the vertical one taller than six weeks and the label. `1405-\m-01` works because the hyphens are written out and `\m` is only read as a number; a single macro holding the whole date would not. All months are in the same year, so `month text` replaces the default label (month and year) with the month name alone; put it after `jalali week list`, which sets its own.

```
\begin{tikzpicture}
\foreach \m in {1,...,12} {
  \pgfmathtruncatemacro\col{mod(\m-1,3)}% month within the season
```

```
  \pgfmathtruncatemacro\row{div(\m-1,3)}% season
  \calendar at (-\col*5.4cm, -\row*4.6cm)
    [jalali dates=1405-\m-01 to 1405-\m-last, jalali week list,
     day xshift=-0.7cm, day yshift=0.5cm,
     jalali month label above centered,
     month text={\pcalmonthname{\pcalcurrentmonth}}]
    if (Friday) [red];
}
\end{tikzpicture}
```

خرداد

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

اردیبهشت

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

فروردین

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

شهریور

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

مرداد

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

تیر

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

آذر

6	5	4	3	2	1	
13	12	11	10	9	8	7
20	19	18	17	16	15	14
27	26	25	24	23	22	21
				30	29	28

آبان

1						
8	7	6	5	4	3	2
15	14	13	12	11	10	9
22	21	20	19	18	17	16
29	28	27	26	25	24	23
						30

مهر

3	2	1				
10	9	8	7	6	5	4
17	16	15	14	13	12	11
24	23	22	21	20	19	18
	30	29	28	27	26	25

اسفند

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

بهمن

2	1					
9	8	7	6	5	4	3
16	15	14	13	12	11	10
23	22	21	20	19	18	17
30	29	28	27	26	25	24

دی

4	3	2	1			
11	10	9	8	7	6	5
18	17	16	15	14	13	12
25	24	23	22	21	20	19
		30	29	28	27	26

Calendars for a range of years. The calxxxx-yyyy document on CTAN prints Gregorian calendars for a range of years, nine years to a page, each year as four rows of three months and each month as a column per week and a row per weekday. `\pcalyear` and `\pcalyears` draw the same layout for Jalali years, with `jalali week columns`. Like the keys above they need TikZ and its `calendar` library (the `tikz` option); without them they stop with an error that says so.

`\pcalyear{year}`

Draws one Jalali year as a `tikzpicture`: the year as title, then four rows of seasons, three months each, right to left, with the Fridays in red. It is sized for nine years to an A4 page, 0.3 cm per day.

`\pcalyears{first}{last}`

Draws the years from `first` to `last`, three to a row, on as many pages as they need. The sample `persian-calendar-years.tex`, which comes with the package, is the Jalali counterpart of calxxxx-yyyy: it calls `\pcalyears` with `\FirstYear` and `\LastYear`, set at its top to `1405` and `1413`.

Styles. Four TikZ styles change the look; set them with `\tikzset`:

- `every pcal year`: the picture of a year (default font `\tiny`);
- `every pcal year title`: the year at the top (default font `\footnotesize\bfseries`);
- `every pcal month`: added to the options of each month's `\calendar`, after the package's own;
- `pcal friday`: the Fridays, `red` by default.

The first three are empty; the fonts above are set before them. For example, this draws the Fridays in blue:

```
\tikzset{pcal friday/.style={blue}}
```

This is `\pcalyear{1405}`:



8 A year planner

The yplan package on CTAN prints a vertical year planner: all the days of a year on two A4 pages, six months to a page, with a column per month and a row per day. `\pcalplanner` does

the same for a Jalali year. It is an ordinary `tabular`, so it needs neither TikZ nor any package beyond the ones that print Persian.

`\pcalplanner{year}`

`\pcalplanner*{year}` Typesets Jalali `year` on two pages: Farvardin to Shahrivar, then Mehr to Esfand. Each page starts with `\clearpage` and has page style `empty`. Above the table is the year, followed by `\pcalplannerleapyear` ("سال کبیسه") in a leap year. Down the side are the weekday names; the rows are aligned on the weekday, so a row is the same weekday in every month. Weeks run Saturday to Friday and a double rule follows every Friday. As in yplan, only the cells that hold a day are ruled, and each page has as many rows as its months need (at most 37). The six month columns share the line width that the weekday column and the rules leave, so the page margins set their width. Under xepersian the table runs right to left: the weekday column is on the right, then Farvardin. The year may be a macro.

```
\documentclass{article}
\usepackage[a4paper,margin=1.2cm]{geometry}
\usepackage{persian-calendar}
\usepackage{xepersian}
\settextfont{Parisa}
\begin{document}
\pcalplanner{1405}
\end{document}
```

The sample `persian-calendar-planner.tex`, which comes with the package, prints such planners with the Fridays shaded and a few notes.

Week numbers. `\pcalplanner` prints the number of the week in each Saturday cell, on the far side of the cell from the day (the left under xepersian), and in the cell of 1 Farvardin when that is not a Saturday; `\pcalplanner*` leaves the week numbers out. There is no official rule for numbering the weeks of the Jalali year, so the package moves ISO 8601 to the Persian week. ISO makes week 1 the week that holds the year's first Thursday, the middle day of a Monday-to-Sunday week. The middle day of a Saturday-to-Friday week is Tuesday, so here week 1 is the week that holds the first Tuesday of the year, that is the week holding 4 Farvardin. As in ISO, every week runs from Saturday to Friday, and a year has 52 weeks, or 53 when 1 Farvardin is a Tuesday, or a Monday in a leap year. 1–3 Farvardin belong to the last week of the old year when 1 Farvardin is a Wednesday, Thursday or Friday: 1 Farvardin 1403, a Wednesday, is in week 53 of 1402. The last days of Esfand belong to week 1 of the new year when the next 1 Farvardin is a Sunday, Monday or Tuesday. `\pcalplannerweeknumber` prints the number, which it takes as its argument, in `\scriptsize`; redefine it to change the look.

Adjusting the layout. As in yplan, these can be redefined with `\renewcommand`:

- `\pcalplannerdaystrut`: sets the height of the day rows; `\rule{0pt}{12pt}` by default. Make it taller or shorter to fill the page.
- `\pcalplannermonthstrut`: the same for the row of month names.
- `\pcalplannerleapyear`: the leap-year label.

`\pcalplannernote{M-D}{text}`

`\pcalplannernote{Y-M-D}{text}` Adds a note to a day, as on a wall calendar: `\pcalplanner` puts `\pcalplannermark` (a star) after the day in its cell and lists the notes of

each page under its table, in date order and grouped by month: each month once, followed by its days and their notes, as in "Farvardin: 1) … 13) … Khordad: 14) …". A date without a year (`M-D`) is for every year, which suits holidays; one with a year (`Y-M-D`) only for that year. A day may have several notes; it is listed once, with its texts joined by a comma, and the notes for every year come first. Give the notes before `\pcalplanner`, for example in the preamble. A note for 30 Esfand appears only in a leap year.

```
\pcalplannernote{1-1}{Nowruz}
\pcalplannernote{1-13}{Nature Day}
\pcalplannernote{1405-7-1}{First day of autumn 1405}
```

These macros set the look of the list; redefine them, and `\pcalplannermark`, to change it:

- `\pcalplannernotesfont`: the font of the list, `\footnotesize`;
- `\pcalplannernotemonth{year}{month}`: the heading of a month, its name in bold and a colon;
- `\pcalplannernoteday{year}{month}{day}`: the label of a day, `day)`;
- `\pcalplannernotemonthsep`, `\pcalplannernotesep`, `\pcalplannernotetextsep`: the space between months (`\qquad`), between days (`\quad`) and between the texts of one day (a Persian comma).

The notes take room under the table: with many of them, make `\pcalplannerdaystrut` shorter so that each page still fits.

`\pcalplannercell{year}{month}{day}{weekday}`

Typesets one day cell. `\pcalplanner` calls it for every day with the Jalali year, month and day, and the weekday numbered as `\pcalweekdayname` does (0 is Saturday, 6 is Friday). By default it prints the day number. Redefine it to colour, mark or annotate days; empty cells do not call it. This one shades the Fridays and prints 1 Farvardin in red:

```
\usepackage[table]{xcolor}
\providecommand\UseMathForPositioningText{}
...
\renewcommand\pcalplannercell[4]{%
  \ifnum#4=6 \cellcolor{gray!20}\fi
  \ifnum\numexpr#2*100+#3=101 \textcolor{red}{#3}\else#3\fi}
```

`\cellcolor` comes from colortbl, which the `table` option of xcolor loads. The `\providecommand` line works around a bug in bidi (section 12). Use `\textcolor` rather than `\color` in a cell: `\color` at the start of a cell moves the number down a line. The cell is a paragraph column, so longer text, such as a holiday name, wraps inside it.

9 A yearly to-do list

`\pcaltodo[options]{year}`

Prints a yearly to-do list for Jalali `year`: two pages for each month, a page for the first half and a page for the rest. A page has a column for each day, with rows to write in. The head of a day column shows, from the top, the Gregorian day of the year, the Gregorian month in three letters and the Gregorian day, all in the Latin font; then the Jalali week number, the weekday as one letter and the Jalali day of the year, and under them the Jalali day. The week

number is drawn in a circle on each Saturday (and on 1 Farvardin), by the rule of `\pcalplanner` (section 8); an arrow runs from it to the next week's number. Above the days are the month and year, and the range of Gregorian months. It needs TikZ (the `tikz` option), and xepersian for the Persian text.

```
\documentclass{article}
\usepackage[a5paper,margin=0pt]{geometry}
\usepackage[tikz]{persian-calendar}
\usepackage{xepersian}
\settextfont{Parisa}
\setlatintextfont{Linux Libertine}
\pagestyle{empty}
\begin{document}
\pcaltodo[line color=teal, columns={Next week, Task|event, Purchase}]{1405}
\end{document}
```

`line color=`***colour*** the colour of all lines; `brown` by default.

`columns={`***a,b,...***`}` the titles of the columns to the left of the day columns, one column per title and as many as you like. The first title is next to the days and the others follow to the left; each title is written from top to bottom in its column. There are no columns by default, so all the space is yours.

Page size. The form is designed for A5 (14.8 cm by 20.98 cm). It is scaled, in proportion, to fit the text area, with its text and lines, so any paper works: set it and the margins with geometry. `a5paper,margin=0pt` gives the original size. When the page has other proportions, the form is centred. Each form ends with `\newpage`.

Look. `\pcaltodoweekday{n}` prints the weekday (0 is Saturday) and `\pcaltodogregmonth{m}` the Gregorian month; redefine them to change those. The week number is set by `\pcalplannerweeknumber`.

10 Recipes

A list of consecutive days. Typeset the 5 days after 1 Farvardin 1403 with a loop. Each date is converted and printed right away, because the result macros are overwritten by the next conversion:

```
\newcount\jd
\foreach \x in {0,...,4}{%
  \pcaldatetojulian{1403-01-01+\x}\jd
  \pcaljuliantodate\jd\y\m\d
  \number\y-\number\m-\number\d\par
}
```

1403-1-1
1403-1-2
1403-1-3
1403-1-4
1403-1-5

Using `\x` inside `\pcaldatetojulian` works, because that macro expands its argument.

Declare registers once. `\newcount` inside a loop allocates a new register at every iteration and eventually fails with "No room for a new `\count`". Declare your counters before the loop.

11 How it works

The conversions are plain TeX register arithmetic, with no external program and no Lua. Two internal macros do all the work. Every public macro calls one of them and then prints or copies the result registers.

`\pcal@gregoriantojalali{Y-M-D}` Gregorian to Jalali. It first decides whether the Gregorian year is a leap year, builds the month-length tables of both calendars, and counts the days from the start of the Gregorian year to the date. It then walks the Jalali month lengths until that day count is used up. The Jalali leap year rule is the common *33-year cycle*: eight leap years in every 33.

`\pcal@jalalitogregorian{Y-M-D}` Jalali to Gregorian. It computes the number of days since a fixed epoch from the year (365 days per year, plus 8 leap days per 33 years, plus the leap days of the partial cycle), adds the days of the preceding months, and then splits that total into 400-year, 100-year, 4-year and 1-year Gregorian blocks.

Result registers. Jalali results are in `\pcal@y`, `\pcal@m`, `\pcal@d`. Gregorian results are in `\pcal@gy`, `\pcal@gm`, `\pcal@gd`. The leap flags are `\ifpcal@leap` (Gregorian) and `\ifpcal@kabiseh` (Jalali). The month lengths are held in registers whose names end in roman numerals (`\pcal@latini` to `\pcal@latinxii` and `\pcal@persiani` to `\pcal@persianxiii`). Dey appears twice: the first slot is the part of Dey left in the Gregorian year, the last is the whole month, which is why there are thirteen.

Julian day bridge. `\pcaldatetojulian` converts the Jalali date to Gregorian with the Jalali-to-Gregorian macro above and hands the result to `\pgfcalendardatetojulian`. `\pcaljuliantodate` does the reverse: `\pgfcalendarjuliantodate` gives a Gregorian date, which the Gregorian-to-Jalali macro turns into the Jalali date.

Year names. The twelve animals repeat every 12 years. A second, three-step shift is applied once per century, so that year 0 starts the cycle at a different animal than year 100. This is why the macro stops at 2099.

No register leaks. All registers are allocated once when the package loads, so any number of conversions can be made in one document. A test in the repository (`leaks.lvt`) checks this.

TikZ calendars. The keys of section 7 are defined with `\pgfkeys` when the package loads, so they need no TikZ until they are used. `jalali dates` converts both ends to Gregorian and passes them to the library's `dates` key; `last` becomes the first day of the next Jalali month with a `+-1` offset. `jalali` puts a call that sets `\pcalcurrentyear` and the others in front of every day, ahead of the library's own per-day code, and redefines the `day of month` test for that calendar only. The conversions are run inside a group, because pgfcalendar walks the days with `\loop` and the conversions use `\loop` too.

The year planner. For each page `\pcalplanner` converts only the 1st of each of its six months (and of the month after) to a day number, through `\pcaldatetojulian`'s machinery; their differences are the month lengths, and the weekday of each 1st gives the month's first row. The rows are then written into a token register, one cell per month with a call to `\pcalplannercell`

in each cell that holds a day, together with the `\cline`s under them, and the register is put into the `tabular`. The rows are built by recursion rather than `\loop`, for the reason above.

12 Limitations and known issues

- **No input checking.** A month 13, day 40 or malformed string gives wrong output or a TeX error, not a package message.
- **Literal dates only.** The date argument of `\pdate`, `\gdate`, `\pisodate` and `\gisodate` is split at the hyphens as it stands; a macro holding a date is not expanded. Write `\pdate{2015-03-21}`, not `\pdate\mydate`. (`\pcaldatetojulian` does expand its date.)
- **Approximate Jalali leap rule.** The 33-year rule is an arithmetic approximation of the official calendar, which is defined by astronomical observation. The two agree for the years in ordinary use, but not necessarily for all years. For distant years, check against an authoritative table.
- **A known crash.** `\pcal@jalalitogregorian{1279-10-11}` stops with "Missing number, treated as zero"; neighbouring dates work. The cause has not been found.
- **Nested loops.** The conversions use `\loop` from the LaTeX kernel internally. A `\loop…\repeat` of your own that calls them must run its body inside a group (`\begingroup…\endgroup`), otherwise it stops after one pass. `\foreach` is not affected.
- **Shared registers.** Every conversion overwrites the result registers, as described in section 5.
- **Year names** work for 0 to 2099 only.
- **TikZ layouts.** Only `jalali week list`, `jalali week columns` and the `day list` styles follow Jalali months; the library's `week list` and `month list` stay Gregorian.
- `jalali dates` **and** `dates` set the same range, so the later one wins. The dates of `jalali dates` must be literal, like those of `\pdate`.
- **Boxes and labels assume right-aligned days.** `jalali day boxes` and `jalali month label above centered` expect the library's default `every day` anchor (`base east`). The initials are one style for all seven days, so a Friday initial cannot be coloured on its own.
- **TikZ internals.** `jalali` adds its code to the library's internal macro `\tikz@before@day`. A future TikZ that renames it would break the Jalali calendars; the regression test `tikz.lvt` would catch it.
- **Shading planner cells with xepersian.** Loading colortbl (for example with `\usepackage[table]{xcolor}`) together with xepersian stops at the first table with "Undefined control sequence `\UseMathForPositioningText`". bidi's adaptation of array uses this macro, which current versions of array no longer define. Until bidi is fixed, put `\providecommand\UseMathForPositioningText{}` in the preamble.
- **The planner is A4 portrait by default.** `\pcalplanner` does not change the page size or the margins; set them with geometry, and the row height with `\pcalplannerdaystrut`.
- **Do not redefine** `\wd`**.** `\newcount\wd` or similar replaces a TeX primitive that TikZ needs, and stray text such as "`==0pt`" appears in the pictures.

13 Version history

0.80 (2026-11-15) `\pcaltodo`: a yearly to-do list with weekday letters, Latin Gregorian dates, circled Jalali week numbers and your own titled columns, scaled to any paper size.

0.78 (2026-11-06) `\pcalplannernote`: notes in the year planner, marked in the cells and listed under each page by month, as on a wall calendar.

0.77 (2026-11-06) Week numbers in `\pcalplanner`, by ISO 8601 moved to Saturday weeks; `\pcalplanner*` leaves them out.

0.76 (2026-11-06) Manuals: the Installation section is now "Loading the package".

0.75 (2026-11-06) `\pcaldaysbetween`: the number of days between two Jalali dates.

0.74 (2026-11-06) `\pcalweekday` and `\pcaldateweekdayname`: the weekday of a Jalali date.

0.73 (2026-11-06) The `tikz` option, which loads TikZ and its `calendar` library.

0.72 (2026-11-05) `\pcalyear` and `\pcalyears`: year calendars in the manner of calxxxx-yyyy, with styles for the year, its title, the months and the Fridays.

0.71 (2026-11-05) Shorter names: `\pcal…` instead of `\pcalendar…`; the old names still work.

0.7 (2026-11-05) A year planner in the manner of yplan: `\pcalplanner`, and `\pcalplannercell` to colour or annotate its days. The sample `persian-calendar-planner.tex`.

0.62 (2026-11-04) `jalali week columns` and the sample `persian-calendar-years.tex`.

0.61 (2026-11-03) `jalali day boxes` (a grid of boxes) and `jalali weekday initials`.

0.6 (2026-11-03) Renamed from pcalendar to persian-calendar at the request of CTAN (file and package names only, the macros keep their names). Jalali month grids with the TikZ `calendar` library (section 7). License stated as LPPL 1.3c everywhere.

0.51 (2026-11-01) Documentation only: the manuals now say that only printing the Persian names needs a Unicode engine and a Persian font; the Persian manual was rewritten.

0.5 (2026-11-01) Fixed the register leak (several `\newcount` per call) and the undeclared month-length register; added this manual and `\pcalendardate`/`\pcalendarversion`.

0.4 (2023-11-07) Ready for CTAN.

0.3 (2015-08-14) Animal name of a Jalali year, 0 to 2099.

0.2 (2015-08-09) Jalali to Gregorian conversion.

0.1 (2015-07-25) Initial version.