[majidh1/JalaliDatePicker: A lightweight and simple jalali datepicker. powerful javascript datepicker with no dependencies](https://github.com/majidh1/JalaliDatePicker)  
  
The jdatetime package is a Python library that provides a Jalali (Persian) calendar implementation based on Python's standard datetime module. It allows for seamless conversion between Gregorian and Jalali dates and times.

**📦 Package Information**

* **Author**:Milad Rastia
* **License**:Python Software Foundation Licens
* **Latest Version**:5.2.0 (released on January 26, 2025 citeturn0search1
* **Python Compatibility**:Supports Python versions 3.9 and abov citeturn0search0
* **Installation**:Install via pip



pip install jdatetime

```

---

### 🔧 Key Features

Provides `jdatetime.date` and `jdatetime.datetime` classes that mirror the functionality of Python's standard `date` and `datetime` classe.

Includes methods for converting between Gregorian and Jalali date:

`fromgregorian(`

`togregorian(`

Supports localization, allowing the setting of the locale to Persian (Farsi) using `jdatetime.set\_locale(jdatetime.FA\_LOCALE).

Offers timezone-aware date and time handlin.

---

### 📘 Documentation & Resources

- \*\*PyPI Page\*: [https://pypi.org/project/jdatetime/](https://pypi.org/project/jdatetim/)

- \*\*GitHub Repository\*: [https://github.com/slashmili/jdatetime](https://github.com/slashmili/jdatetie)

- \*\*Official Documentation\*: [https://pythonhosted.org/jdatetime/](https://pythonhosted.org/jdatetim/)

---

### 🛠️ Example Usae

Here's how you can use `jdatetime` to convert a Gregorian date to a Jalali dte:



```python

import jdatetime

# Convert Gregorian date to Jalali

gregorian\_date = jdatetime.date(2025, 4, 14)

jalali\_date = gregorian\_date.togregorian()

print(jalali\_date) # Output: 1404/01/25

``



To convert a Jalali date to a Gregorian dte:



```python

import jdatetime

# Convert Jalali date to Gregorian

jalali\_date = jdatetime.date(1404, 1, 25)

gregorian\_date = jalali\_date.togregorian()

print(gregorian\_date) # Output: 2025-04-14

``



--

For more detailed information and advanced usage, refer to the official documentation linked abve.

برای استفاده از دیتاپیکر فارسی در تمپلیت Django، به‌طور کامل مراحل زیر را دنبال کن تا تاریخ‌ها به‌درستی در قالب فارسی (شمسی) نمایش داده شوند و مقایسه تاریخ‌ها نیز به‌طور دقیق انجام گیرد:

### 1. تنظیمات HTML و جاوا اسکریپت در تمپلیت

ابتدا مطمئن شو که پکیج‌هایی که به دیتاپیکر فارسی نیاز داری در تمپلیت لود شده‌اند. به‌علاوه، برای استفاده از دیتاپیکر فارسی، باید فایل‌های CSS و JavaScript مناسب را در تمپلیت خود بارگذاری کنی.

در فایل تمپلیت (.html)، این کدها را اضافه کن:

<!-- لود کردن CSS و JS دیتاپیکر فارسی -->

<link href="{% static 'admin/css/jalalidatepicker.min.css' %}" rel="stylesheet">

<script src="{% static 'admin/js/datepicker-fa.js' %}"></script>

<script src="{% static 'admin/js/django\_jalali.min.js' %}"></script>

<script src="{% static 'admin/js/jalalidatepicker.min.js' %}"></script>

<!-- فرم تاریخ با دیتاپیکر فارسی -->

<div class="col-md-2">

<label for="date\_from" class="form-label">{% trans "از تاریخ" %}</label>

<input type="text" name="date\_from" id="date\_from" value="{{ date\_from|default:'' }}" class="form-control jalali-datepicker" placeholder="1404/01/01" autocomplete="off">

</div>

<div class="col-md-2">

<label for="date\_to" class="form-label">{% trans "تا تاریخ" %}</label>

<input type="text" name="date\_to" id="date\_to" value="{{ date\_to|default:'' }}" class="form-control jalali-datepicker" placeholder="1404/12/29" autocomplete="off">

</div>

<!-- اسکریپت برای فعال‌سازی دیتاپیکر -->

<script>

$(document).ready(function () {

$('.jalali-datepicker').jalaliDatepicker({

format: 'YYYY/MM/DD', // فرمت تاریخ شمسی

altFormat: 'YYYY/MM/DD', // فرمت دیگری برای ذخیره تاریخ

altField: '#hidden\_date\_field' // اگر نیاز به ذخیره تاریخ به صورت میلادی داشتید

});

});

</script>

### 2. مقایسه تاریخ‌ها در تمپلیت

اگر قصد داری تاریخ‌های شمسی وارد شده را در داخل تمپلیت مقایسه کنی، باید از ویو یا جاوا اسکریپت استفاده کنی. برای مقایسه تاریخ‌ها باید آنها را به فرمت میلادی تبدیل کرده و سپس مقایسه کنی.

#### 2.1. مقایسه تاریخ‌ها در ویو (Django View)

در ویو، می‌توانی تاریخ‌ها را با استفاده از توابع parse\_jalali\_date که قبلاً تعریف کرده‌ایم تبدیل کنی و مقایسه انجام دهی:

from django.shortcuts import render

from .utils import parse\_jalali\_date

def budget\_period\_filter\_view(request):

date\_from = request.GET.get('date\_from', '')

date\_to = request.GET.get('date\_to', '')

# تبدیل تاریخ‌های وارد شده به میلادی

if date\_from:

try:

date\_from = parse\_jalali\_date(date\_from)

except ValidationError:

date\_from = None # در صورتی که تاریخ نامعتبر باشد

if date\_to:

try:

date\_to = parse\_jalali\_date(date\_to)

except ValidationError:

date\_to = None

# اعمال فیلتر در queryset

queryset = BudgetPeriod.objects.all()

if date\_from:

queryset = queryset.filter(start\_date\_\_gte=date\_from)

if date\_to:

queryset = queryset.filter(end\_date\_\_lte=date\_to)

return render(request, 'your\_template.html', {

'budget\_periods': queryset,

'date\_from': date\_from,

'date\_to': date\_to,

})

#### 2.2. مقایسه تاریخ‌ها با استفاده از جاوا اسکریپت (در تمپلیت)

اگر می‌خواهی مقایسه تاریخ‌ها را در داخل خود تمپلیت انجام دهی، می‌توانی از جاوا اسکریپت برای تبدیل تاریخ‌ها به میلادی استفاده کنی، سپس آنها را مقایسه کنی:

<script>

$(document).ready(function () {

// تاریخ شمسی وارد شده توسط کاربر

var dateFrom = $('#date\_from').val();

var dateTo = $('#date\_to').val();

// تبدیل تاریخ‌ها به میلادی

function convertToGregorian(jalaliDate) {

var parts = jalaliDate.split('/');

var year = parseInt(parts[0]);

var month = parseInt(parts[1]);

var day = parseInt(parts[2]);

// استفاده از jdatetime برای تبدیل تاریخ شمسی به میلادی

var jDate = new jdatetime.date(year, month, day);

var gregorianDate = jDate.togregorian();

return new Date(gregorianDate.year, gregorianDate.month - 1, gregorianDate.day);

}

var dateFromGregorian = convertToGregorian(dateFrom);

var dateToGregorian = convertToGregorian(dateTo);

// مقایسه تاریخ‌ها

if (dateFromGregorian > dateToGregorian) {

alert("تاریخ شروع نمی‌تواند بعد از تاریخ پایان باشد.");

}

});

</script>

### 3. خلاصه

* استفاده از دیتاپیکر فارسی با پکیج jalalidatepicker در تمپلیت.
* تبدیل تاریخ شمسی به میلادی در ویو یا با استفاده از جاوا اسکریپت.
* مقایسه تاریخ‌ها در تمپلیت با جاوا اسکریپت یا در ویو با استفاده از parse\_jalali\_date.

این کدها باید به‌طور کامل تاریخ‌ها را با فرمت شمسی در تمپلیت نمایش دهند و همچنین امکان مقایسه تاریخ‌ها را فراهم کنند.

با توجه به درخواست شما برای اطلاعات درباره پکیج jdatetime که در پروژه‌تان برای مدیریت تاریخ استفاده می‌کنید، و با استناد به توضیحات ارائه‌شده در مورد کپی‌رایت و لایسنس (متعلق به Milad Rastian و تحت لایسنس Python)، در ادامه جزئیات کامل این پکیج را ارائه می‌دهم. اطلاعات بر اساس منابع معتبر و به‌روز تهیه شده است تا نیاز شما در زمینه استفاده از این پکیج در سیستم بودجه‌ریزی (مرتبط با BudgetPeriodListView و دیت‌پیکر شمسی) برآورده شود.

**اطلاعات پکیج** jdatetime

**۱. توضیحات کلی**

* **نام پکیج**: jdatetime
* **توضیحات**: jdatetime یک پیاده‌سازی از تقویم جلالی (شمسی) برای پایتون است که بر اساس ماژول استاندارد datetime پایتون ساخته شده است. این پکیج امکان کار با تاریخ‌های شمسی را فراهم می‌کند و روش‌های مشابه datetime را ارائه می‌دهد، با قابلیت تبدیل تاریخ‌های شمسی به میلادی و برعکس.
* **هدف**: مناسب برای پروژه‌هایی که نیاز به مدیریت تاریخ‌های شمسی دارند، مانند سیستم‌های ایرانی یا برنامه‌هایی که با کاربران فارسی‌زبان کار می‌کنند (مثل سیستم بودجه‌ریزی شما).
* **لایسنس**: Python Software Foundation License (لایسنس پایتون)، که یک لایسنس متن‌باز و آزاد است و امکان استفاده، اصلاح و توزیع را فراهم می‌کند.
* **کپی‌رایت**: متعلق به Milad Rastian (eslashmili@gmail.com) از سال ۲۰۱۰ تا ۲۰۱۸.

**۲. ویژگی‌ها**

* **تطابق با** datetime: تمامی متدهای استاندارد ماژول datetime پایتون (مانند now(), today(), strftime()) در jdatetime پشتیبانی می‌شوند.
* **متدهای اضافی**:
  + fromgregorian(\*\*kw): تبدیل تاریخ میلادی به شمسی. مثال:

python

jdatetime.date.fromgregorian(year=2025, month=4, day=14) # خروجی: تاریخ شمسی معادل

* + togregorian(): تبدیل تاریخ شمسی به میلادی. مثال:

python

jdatetime.date(1404, 1, 25).togregorian() # خروجی: تاریخ میلادی معادل

* + isleap(): بررسی سال کبیسه در تقویم شمسی.
  + پشتیبانی از locale برای نمایش تاریخ به زبان فارسی (با تنظیم jdatetime.FA\_LOCALE).
* **پشتیبانی از لوکال**:
  + با تنظیم لوکال به fa\_IR، تاریخ‌ها با نام‌های فارسی (مثل «چهارشنبه»، «اردیبهشت») نمایش داده می‌شوند:

python

import jdatetime

import locale

locale.setlocale(locale.LC\_ALL, jdatetime.FA\_LOCALE)

print(jdatetime.datetime.now().strftime("%A, %d %B %Y")) # مثال: چهارشنبه, ۲۵ فروردین ۱۴۰۴

* **پشتیبانی از تردها**: امکان تنظیم لوکال پیش‌فرض برای هر ترد (thread) یا greenlet.

**۳. نصب**

* **دستور نصب**:

bash

pip install jdatetime

* **آخرین نسخه** (تا تاریخ فعلی، ۱۴ آوریل ۲۰۲۵): نسخه ۵.۲.۰ (منتشرشده در ۲۶ ژانویه ۲۰۲۵).
* **نیازمندی‌ها**:
  + پایتون نسخه ۳.۹ یا بالاتر.
  + سیستم‌عامل: مستقل از سیستم‌عامل (OS Independent).

**۴. کاربردها**

* **توسعه‌دهندگان**: برای برنامه‌هایی که نیاز به تاریخ شمسی دارند (مثل سیستم‌های مالی، مدیریت بودجه، یا تقویم‌های ایرانی).
* **ادمین‌های سیستم**: برای اسکریپت‌هایی که تاریخ‌های شمسی را پردازش می‌کنند.
* **پروژه شما**: در BudgetPeriodListView، این پکیج می‌تواند برای تبدیل تاریخ‌های شمسی واردشده توسط دیت‌پیکر (مثل 1404/01/01) به فرمت میلادی برای ذخیره در پایگاه داده یا فیلتر کردن استفاده شود.

**۵. مشخصات فنی**

* **نویسنده**: Milad Rastian (eslashmili@gmail.com).
* **صفحه اصلی**: <https://github.com/slashmili/python-jalali>
* **مستندات**: <https://pythonhosted.org/jdatetime/>
* **تگ‌ها**: Jalali, Persian, Calendar, Datetime
* **وضعیت توسعه**: فعال و پایدار (Healthy Maintenance). حداقل یک نسخه جدید در سه ماه گذشته منتشر شده است.
* **محبوبیت**: بیش از ۶۶۹۰ دانلود هفتگی، جزو ۱۰٪ پکیج‌های پرطرفدار PyPI.
* **امنیت**: بدون آسیب‌پذیری شناخته‌شده تا فوریه ۲۰۲۵.

**۶. استفاده در پروژه شما**

با توجه به اینکه در BudgetPeriodListView از دیت‌پیکر شمسی برای فیلدهای date\_from و date\_to استفاده می‌کنید، jdatetime می‌تواند به این صورت در پروژه شما مفید باشد:

* **تبدیل تاریخ‌ها**:
  + تاریخ‌های شمسی واردشده توسط کاربر (مثل 1404/01/01) را با jdatetime.date.fromgregorian یا strptime پردازش کنید:

python

import jdatetime

date\_str = "1404/01/01"

jalali\_date = jdatetime.datetime.strptime(date\_str, "%Y/%m/%d").date()

gregorian\_date = jalali\_date.togregorian() # برای ذخیره در پایگاه داده

* **نمایش تاریخ‌ها**:
  + تاریخ‌های ذخیره‌شده (میلادی) را به شمسی تبدیل کنید و در دیت‌پیکر نمایش دهید:

python

from datetime import date

gregorian\_date = date(2025, 4, 14)

jalali\_date = jdatetime.date.fromgregorian(date=gregorian\_date)

date\_str = jalali\_date.strftime("%Y/%m/%d") # خروجی: 1404/01/25

* **یکپارچگی با دیت‌پیکر**:
  + فرمت خروجی jdatetime (YYYY/MM/DD) با دیت‌پیکر شمسی (persian-datepicker) سازگار است، بنابراین می‌توانید از آن برای پر کردن فیلدها استفاده کنید:

html

<input type="text" name="date\_from" id="date\_from" value="{{ date\_from|default:jalali\_date.strftime('%Y/%m/%d') }}" class="form-control jalali-datepicker">

**۷. مزایا و معایب**

* **مزایا**:
  + سازگاری کامل با datetime استاندارد پایتون.
  + پشتیبانی از تبدیل دوطرفه شمسی-میلادی.
  + متن‌باز و رایگان با لایسنس آزاد.
  + مستندات قابل‌قبول و جامعه فعال.
* **معایب**:
  + مستندات ممکن است برای کاربران جدید کمی پیچیده باشد (برخی کاربران آن را «غیر دوستانه» توصیف کرده‌اند).
  + عدم پشتیبانی از برخی فیلترهای خاص (مثل \_\_month در جنگو) در نسخه‌های قدیمی‌تر.

**۸. ارتباط با دیت‌پیکر**

با توجه به مشکل قبلی شما (عدم کارکرد jalali-datepicker)، jdatetime می‌تواند به‌عنوان مکمل برای پردازش تاریخ‌های شمسی در سمت سرور استفاده شود، در حالی که دیت‌پیکر (persian-datepicker) برای رابط کاربری است. برای اطمینان از یکپارچگی:

* **فرمت تاریخ**: دیت‌پیکر باید تاریخ‌ها را در فرمت YYYY/MM/DD تولید کند (که با تنظیم format: 'YYYY/MM/DD' انجام شده است).
* **پردازش در ویو**: تاریخ‌های دریافتی از فرم را با jdatetime اعتبارسنجی و تبدیل کنید:

python

from Tanbakhsystem.utils import parse\_jalali\_date

def parse\_jalali\_date(date\_str):

try:

return jdatetime.datetime.strptime(date\_str, "%Y/%m/%d").date().togregorian()

except ValueError:

raise ValueError("فرمت تاریخ نامعتبر است (مثال: 1404/01/01)")

**۹. وضعیت پروژه**

* **ریپازیتوری گیت‌هاب**: <https://github.com/slashmili/python-jalali>
* **ستاره‌ها**: بیش از ۳۵۴ ستاره در گیت‌هاب.
* **مشارکت**: امکان مشارکت از طریق فورک و ارسال Pull Request.
* **آخرین به‌روزرسانی**: نسخه ۵.۲.۰ در ژانویه ۲۰۲۵، با بهبودهایی در پشتیبانی از Python 3.13 و رفع باگ‌های کوچک.

**۱۰. مثال‌های کاربردی**

* **دریافت تاریخ فعلی شمسی**:

python

import jdatetime

print(jdatetime.date.today()) # مثال: 1404/01/25

* **تبدیل تاریخ میلادی به شمسی**:

python

from datetime import date

gregorian\_date = date(2025, 4, 14)

jalali\_date = jdatetime.date.fromgregorian(date=gregorian\_date)

print(jalali\_date) # مثال: 1404/01/25

* **اعتبارسنجی تاریخ ورودی**:

python

date\_str = "1404/01/01"

try:

jalali\_date = jdatetime.datetime.strptime(date\_str, "%Y/%m/%d").date()

print("تاریخ معتبر:", jalali\_date)

except ValueError:

print("فرمت تاریخ نامعتبر است")

**یکپارچگی با پروژه شما**

* **نیاز شما**: استفاده از تاریخ شمسی در BudgetPeriodListView برای فیلتر کردن دوره‌های بودجه با دیت‌پیکر.
* **راه‌حل**:
  + از jdatetime برای پردازش تاریخ‌های ورودی (date\_from و date\_to) در ویو استفاده کنید.
  + دیت‌پیکر (persian-datepicker) را با فرمت YYYY/MM/DD تنظیم کنید تا با jdatetime سازگار باشد.
  + تاریخ‌های ذخیره‌شده در پایگاه داده (میلادی) را با jdatetime.date.fromgregorian به شمسی تبدیل کنید و در تمپلیت نمایش دهید.
* **نمونه کد در ویو**:

python

from django.views.generic import ListView

from Tanbakhsystem.utils import parse\_jalali\_date

import jdatetime

class BudgetPeriodListView(ListView):

model = BudgetPeriod

template\_name = 'budgets/budget/budgetperiod\_list.html'

context\_object\_name = 'budget\_periods'

def get\_queryset(self):

queryset = super().get\_queryset()

date\_from = self.request.GET.get('date\_from')

date\_to = self.request.GET.get('date\_to')

if date\_from:

try:

queryset = queryset.filter(start\_date\_\_gte=parse\_jalali\_date(date\_from))

except ValueError:

messages.error(self.request, "فرمت تاریخ شروع نامعتبر است")

if date\_to:

try:

queryset = queryset.filter(end\_date\_\_lte=parse\_jalali\_date(date\_to))

except ValueError:

messages.error(self.request, "فرمت تاریخ پایان نامعتبر است")

return queryset

def get\_context\_data(self, \*\*kwargs):

context = super().get\_context\_data(\*\*kwargs)

context['date\_from'] = self.request.GET.get('date\_from', '')

context['date\_to'] = self.request.GET.get('date\_to', '')

return context

* **نمونه تابع** parse\_jalali\_date:

python

def parse\_jalali\_date(date\_str):

return jdatetime.datetime.strptime(date\_str, "%Y/%m/%d").date().togregorian()

**رفع مشکل دیت‌پیکر**

با توجه به مشکل قبلی شما (عدم کارکرد jalali-datepicker)، اگر هنوز دیت‌پیکر کار نمی‌کند، اطمینان حاصل کنید که:

1. **jQuery لود شده است**:

html

<script src="{% static 'js/jquery.min.js' %}"></script>

1. **فایل‌های دیت‌پیکر**:

html

<link rel="stylesheet" href="{% static 'persian-datepicker/persian-datepicker.min.css' %}">

<script src="{% static 'persian-date/persian-date.min.js' %}"></script>

<script src="{% static 'persian-datepicker/persian-datepicker.min.js' %}"></script>

1. **اسکریپت اصلاح‌شده**:

html

<script>

document.addEventListener('DOMContentLoaded', function () {

$('.jalali-datepicker').persianDatepicker({

format: 'YYYY/MM/DD',

autoClose: true,

persianDigit: true,

initialValue: true,

minDate: new persianDate([1380, 1, 1]).valueOf(),

maxDate: new persianDate([1450, 12, 29]).valueOf()

});

});

</script>

**نتیجه‌گیری**

* **پکیج** jdatetime:
  + یک ابزار قدرتمند برای مدیریت تاریخ‌های شمسی، با لایسنس آزاد (Python License).
  + نویسنده: Milad Rastian.
  + نسخه فعلی: ۵.۲.۰ (ژانویه ۲۰۲۵).
  + مناسب برای تبدیل و پردازش تاریخ‌ها در پروژه شما.
* **کاربرد در پروژه**:
  + پردازش تاریخ‌های شمسی از دیت‌پیکر و تبدیل آن‌ها به میلادی برای پایگاه داده.
  + نمایش تاریخ‌های شمسی در رابط کاربری.
* **رفع مشکل دیت‌پیکر**:
  + اطمینان از لود jQuery و فایل‌های persian-datepicker.
  + تنظیم فرمت تاریخ برای سازگاری با jdatetime.

لطفاً موارد زیر را تأیید کنید:

* آیا دیت‌پیکر با کدهای ارائه‌شده کار می‌کند؟
* آیا نیاز به مثال یا توضیح بیشتری درباره استفاده از jdatetime در بخش خاصی (مثل ویو، تمپلیت، یا اعتبارسنجی) دارید؟
* آیا نکته خاصی (مثل ویژگی‌های پیشرفته‌تر jdatetime یا ادغام با جنگو) مدنظرتان است؟

اگر مشکلی باقی مانده یا جزئیات بیشتری نیاز دارید، بفرمایید تا دقیق‌تر بررسی کنم.