



Pendulum

Live de python # 137

SOBRE O CANAL



O canal é mantido por uma campanha financiamento coletivo

<https://apoia.se/livedepython>

ou

<https://picpay.me/dunossauro>

Aciássio Araújo, Ademar Peixoto, Adão Oliveira, Alan Soder, Alberto Tavares, Alessandro Lopes, Alex Lima, Alexandre Tsuno, Alexandre Villares, Alexandre Santos, Alexandre Sá, Alexandre Harano, Alysson Oliveira, Amanda Magalhães, Amaziles Carvalho, Andre Rodrigues, Antonio Ribeiro, Bernardo Fontes, Bianca Rosa, Bruno Rocha, Bruno Oliveira, Caio Nascimento, Carlos Cardoso, Carlos Santos, Cleiton Souza, Davi Ramos, Davi Lima, Diego Guimarães, Dilenon Delfino, Douglas Bastos, Edson Braga, Eduardo Nunes, Eduardo Pazzi, Elias Soares, Emerson Lara, Eugenio Mazzini, Everton Alves, Fabrício Coelho, Fernando Lanfranchi, Fernando Vier, Filipe Cruz, Flavkaze , Flávia Batista, Franklin Silva, Fábio Serrão, Gabriel Simonetto, Gabriela Santiago, Gabrielly Andrade, Geandreson Costa, Gladson Menezes, Hélio Neto, Isaac Ferreira, Israel Fabiano, Ivan Vicente, Johnny Tardin, João Lugão, Jonatas Leon, Jonatas Oliveira, Jonatas Oliveira, Jones Leite, Jones Lourenço, Jones Leite, Jorge Plautz, José Prado, Jovan Costa, Juan Gutierrez, Leonardo Cruz, Leonardo Galani, Lucas Mendes, Lucas Valino, Lucas Neris, Lucas Polo, Luciano Ratamero, Luiz Lima, Luiz Bruno, Maiquel Leonel, Marcela Campos, Marcello Benigno, Marcos Jesus, Maria Clara, Mateus Braga, Melisa Campagnaro, Otavio Carneiro, Patric Lacouth, Patrick Gomes, Paulo Tadei, Paulo Gonçalves, Rafael Dias, Rafael Peixoto, Reinaldo Silva, Renne Rocha, Ricardo Schalch, Rodrigo Vaccari, Rodrigo Ferreira, Régis Tomkiel, Samanta Cicilia, Thiago Araujo, Thiago Bueno, Tiago Cordeiro, Tyrone Damasceno, Valdir Junior, Vicente Marcal, Victor Orsolan, Victor Geraldo, Vinícius Ferreira, Vinícius Bastos, Vítor Zanoni, William Oliveira, Willian Lopes, Willian Rosa, Willian Lopes, Wladimir Falcão,



OBRIGADO A VOCÊ





0. Sobre o Pendulum

Sobre a biblioteca

1. Timezones

Trabalhando com regiões

2. Períodos

Trabalhando com tempos diferentes

3. Localização

Trabalhando com tempo em outras línguas

4. API Fluída

Simplificando chamadas

5. Testes

Facilitando para o não uso de mocks

Pendulum

Facilitando a vida
com datetimes

Pendulum, por pendulum



Instâncias de `datetime` são suficientes para casos básicos, mas quando você enfrenta casos de uso mais complexos, elas geralmente mostram limitações e não são tão intuitivas para trabalhar.

Pendulum fornece uma API mais limpa e fácil de usar, ao mesmo tempo que conta com a biblioteca padrão. Portanto, ainda é `datetime`, mas melhor.

<https://pendulum.eustace.io/faq/>

Pendulum, com exemplos



```
1  from pendulum import parse, datetime
2
3  # Datetime tradicional
4  dt = datetime(2020, 9, 7)
5
6  # Parser simplificado
7  simple_parsed = parse('20-09-7', strict=False)
8
9  # Parser comum
10 normal_parsed = parse('1996-12-19T16:39:57-08:00')
11
12 # Ano bissexto
13 normal_parsed.is_leap_year()
```

Pendulum, com exemplos

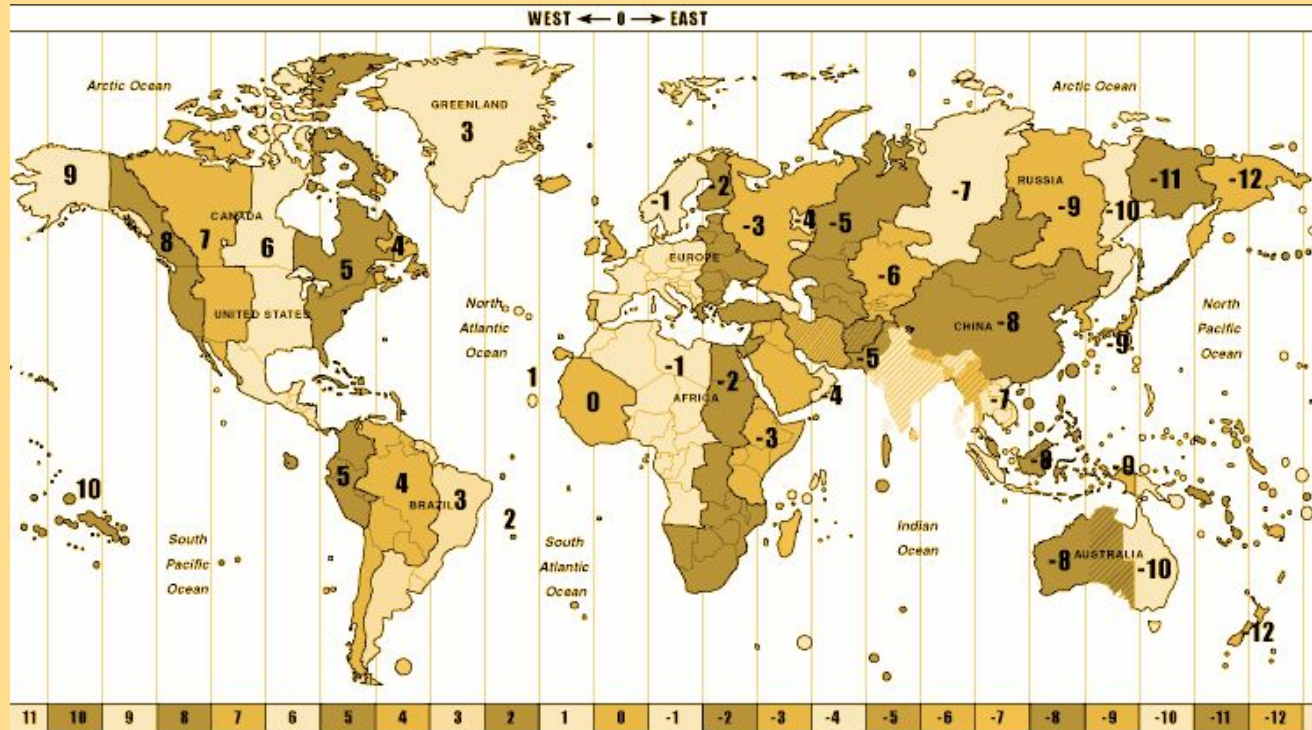


```
1 from pendulum import now
2
3 # Com TZ
4 n = now('UTC')
5
6 # Delta
7 n_plus_10 = n.add(days=10)
8
9 # Delta
10 n_sub_10 = n.subtract(days=10)
11
12 # 24 * 10 = 240 horas
13 n.diff(n_plus_10).in_hours() # 24
14
15 # 24 * 10 = 240 horas
16 n.diff(n_sub_10).in_hours() # 24
```


timez
ones

Simplificando
conversões

timezones



0 objeto timezone



```
1  from pendulum import timezone, now
2
3  # Pega a data e hora atual
4  n = now('UTC')
5
6  # Objeto timezone
7  tz_paris = timezone('Europe/Paris')
8
9  # Hora e data atual em Paris
10 converted_paris = tz_paris.convert(n)
```

Conversão semântica



```
1  from pendulum import timezone, now
2
3  # Pega a data e hora atual
4  n = now( 'UTC' )
5
6  # Conversão "semantica"
7  converted_sp = n.in_tz( 'America/Sao_Paulo' )
```

Períodos

Trabalhando
diferenças entre
datetimes

Atributos entre períodos



```
1  from pendulum import now
2
3  n = now('UTC')
4
5  n_after_year = n.add(years=1, months=3, days=7)
6
7  # Usando Diff
8  diff = n.diff(n_after_year)
9
10 diff.in_days()      # 463 dias
11 diff.in_weeks()     # 66 semanas
12 diff.in_months()    # 15 meses
13 diff.in_years()     # 1 ano
```

Objeto Period



```
1  from pendulum import period, now, tomorrow
2
3  dt1 = now( )
4  dt2 = tomorrow( )
5
6  period = period(dt1, dt2)
7
8  period.in_words( ) # '4 hours 50 minutes 35 seconds'
9  period.in_minutes( ) # 290 minutos
```

Parseando períodos



```
1  from pendulum import parse
2
3  period_5_years = parse('2000-01-01T00:00:00Z/2005-01-01T00:00:00Z')
4  period_5_days = parse('2000-01-01T00:00:00Z/2000-01-06T00:00:00Z')
```


Iterando em períodos



```
1  from pendulum import parse
2
3  period_5_years = parse('2000-01-01T00:00:00Z/2005-01-01T00:00:00Z')
4  period_5_days = parse('2000-01-01T00:00:00Z/2000-01-06T00:00:00Z')
5
6  for year in period_5_years.range('years'):
7      print(year)  # Printar 5 datetimes
8
9  for day in period_5_days.range('days'):
10     print(year)  # Printar 5 datetimes
```

Trabalhando com
línguas diferentes

Locali
zação

Formatação com localização



```
1  from pendulum import datetime
2
3  dt = datetime(2020, 1, 1)
4
5  # Português: quarta-feira 01 janeiro 2020
6  dt.format('dddd DD MMMM YYYY', locale='pt-br')
7
8  # Alemão: Mittwoch 01 Januar 2020
9  dt.format('dddd DD MMMM YYYY', locale='de')
```

Escolhendo localização global



```
1  from pendulum import now, set_locale
2
3  set_locale('pt-br') # Seta para tudo
4
5  dt = now()
6  dt.add(years=1).diff_for_humans()      # em 1 ano
7  dt.add(days=7, months=3).diff_for_humans() # em 3 meses
```

API Fluida

Simplificando
chamadas

Simplicidade na API



```
1  from pendulum import now
2
3  dt = now( )
4
5  # Todas as operações retornam novos objetos
6  dt.set(year=1993) # Datas e horários
7  dt.on(1994, 3, 2) # on para datas
8  dt.at(22, 22, 22) # at para horários
9  dt.in_tz('America/Toronto')
```

Simplicidade na API



```
1  from pendulum import now, yesterday
2
3  n = now()
4  y = yesterday()
5
6  # Valem todos os operadores booleanos
7  n == y  # False
8  n > y   # True
9
10 y.is_past()  # True
11 n.is_leap_year()  # True (obs 2020)
12 n.start_of()  # zero horários - início
13 n.end_of()    # zero horários - fim
```

Simplificando
Mocks

Testes

Gerenciador de contexto para testes



```
1  from pendulum import test, now, yesterday
2
3  dt = yesterday( )
4
5  with test(dt):
6      assert dt == now( )
```

Sem gerenciamento (bom para fixtures)



```
1  from pendulum import set_test_now, now, yesterday
2  from time import sleep
3  dt = yesterday()
4
5  set_test_now(dt) # Inicia o mock
6
7  n1 = now()
8  sleep(3)
9  n2 = now()
10
11 assert n1 == n2
12
13 set_test_now(dt) # Para o mock
```

SOBRE O CANAL



O canal é mantido por uma campanha financiamento coletivo

<https://apoia.se/livedepython>

ou

<https://picpay.me/dunossauro>