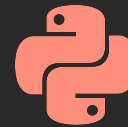




Manipulando data e horários com DateTime

Live de Python #203



1. Datas

Manipulando as datas

2. timedelta

Variações de tempo

3. Horários

Manipulando horários

4. timezone

O tempo consciente

5. Datas e horários

Juntado tudo

6. Dicas e coisas adicionais

Libs de terceiros, indicação de materiais, testes etc...



picpay.me/dunossauro



apoia.se/livedepython



pix.dunossauro@gmail.com



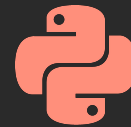
Ajude o projeto <3



Acássio Anjos, Ademar Peixoto, Alexandre Harano, Alexandre Souza, Alexandre Takahashi, Alexandre Villares, Alex Lima, Alynne Ferreira, Alysso Oliveira, Ana Carneiro, Ana Padovan, Andre Azevedo, André Rocha, Aquiles Coutinho, Arnaldo Turque, Bloquearsites Farewall, Bruno Barcellos, Bruno Freitas, Bruno Guizi, Bruno Oliveira, Bruno Ramos, Caio Nascimento, César Almeida, Christiano Moraes, Clara Battesini, Daniel Freitas, Daniel Haas, Danilo Segura, Dartz Dartz, David Kwast, Delton Porfiro, Dhyeives Rodovalho, Diego Farias, Diego Guimarães, Dilenon Delfino, Dino Aguilar, Douglas Bastos, Douglas Braga, Douglas Martins, Douglas Zickuhr, Eli Júnior, Emerson Rafael, Érico Andrei, Eugenio Mazzini, Euripedes Borges, Evandro Avellar, Everton Silva, Fabio Barros, Fábio Barros, Fabio Castro, Fábio Thomaz, Felipe Rodrigues, Fernanda Prado, Fernando Rozas, Fernando Sousa, Flávio Meira, Flavkaze Flavkaze, Franklin Silva, Gabriel Barbosa, Gabriel Simonetto, Geandreson Costa, Guilherme Felitti, Guilherme Gall, Guilherme Ostrock, Gustavo Dettenborn, Gustavo Suto, Heitor Fernandes, Henrique Junqueira, Igor Taconi, Ismael Ventura, Israel Gomes, Italo Silva, Jair Andrade, Jairo Lenfers, Janael Pinheiro, Jean Marcio, João Lugão, Johnny Tardin, Jonatas Leon, Jonatas Oliveira, Jônatas Silva, Jorge Plautz, Jose Mazolini, Juan Gutierrez, Juliana Machado, Julio Silva, Kaio Peixoto, Kaneson Alves, Leandro Botassio, Leandro Miranda, Leonardo Cruz, Leonardo Mello, Leonardo Nazareth, Lucas Adorno, Lucas Mello, Lucas Mendes, Lucas Oliveira, Lucas Polo, Lucas Teixeira, Lucas Valino, Luciano Silva, Luciano Teixeira, Luiz Junior, Luiz Lima, Luiz Paula, Maiquel Leonel, Marcelino Pinheiro, Marcelo Matte, Márcio Martignoni, Marco Mello, Marcos Gomes, Marco Yamada, Maria Clara, Marina Passos, Mario Deus, Mateus Lisboa, Matheus Silva, Matheus Vian, Mirian Batista, Murilo Andrade, Murilo Cunha, Murilo Viana, Natan Cervinski, Nicolas Teodosio, Osvaldo Neto, Otávio Barradas, Patricia Minamizawa, Patrick Felipe, Paulo Braga, Paulo Tadei, Pedro Duarte, Pedro Henrique, Pedro Pereira, Peterson Santos, Priscila Santos, Rafael Lopes, Rafael Rodrigues, Rafael Romão, Ramayana Menezes, Reinaldo Silva, Renan Moura, Renato Veirich, Riverfount Riverfount, Robson Maciel, Rodrigo Brandao, Rodrigo Ferreira, Rodrigo Freire, Rodrigo Junior, Rodrigo Vaccari, Rodrigo Vieira, Rogério Sousa, Ronaldo Silva, Rui Jr, Samanta Cicilia, Sara Selis, Thalysso Bogéa, Thiago Araujo, Thiago Borges, Thiago Bueno, Thiago Curvelo, Thiago Moraes, Thiago Oliveira, Thiago S, Thiago Souza, Tiago Minuzzi, Tony Dias, Victor Wildner, Vinícius Bastos, Vinicius Figueiredo, Vítor Gomes, Vitor Luz, Vlademir Souza, Vladimir Lemos, Walter Reis, Wellington Abreu, Wesley Mendes, William Alves, Willian Lopes, Wilson Neto, Yury Barros



Obrigado você



E seus objetos

Datas

Date



Date é um **objeto** que manipula datas no Python. É um objeto que se refere a dias, meses e anos

```
1  from datetime import date
2
3  # Uma data específica
4  date(day=7, month=7, year=2007)
5
6  # 0 dia de hoje
7  date.today()
```

Aritmética das datas



As datas podem ser comparadas entre si. Qual data é maior, menor igual ou diferente

```
1  from datetime import date
2
3  sete_sete_sete = date(day=7, month=7, year=2007)
4  hoje = date.today()
5
6  hoje > sete_sete_sete # True
7  hoje >= sete_sete_sete # True
8  hoje < sete_sete_sete # False
9  hoje <= sete_sete_sete # False
10 hoje == sete_sete_sete # False
11 hoje != sete_sete_sete # True
```

Atributos do objeto date



Você pode verificar individualmente os atributos de data, mês e ano:

```
1  from datetime import date
2  hoje = date.today()
3
4  hoje.day    # 23
5  hoje.month  # 5
6  hoje.year   # 2022
```


Atributos do objeto date



Porém, são imutáveis

```
1 from datetime import date
2 hoje = date.today()
3
4 hoje.day = 7
5 # AttributeError: attribute 'day' of 'datetime.date' objects is not writable
```

date.replace()



Se precisarmos alterar algum valor da data, devemos chamar o método `replace`:

```
1  from datetime import date
2  hoje = date.today() # 23 de maio de 2022 (23/05/2022)
3
4  # Replace retorna um novo objeto
5  ontem = hoje.replace(day=22)
6  ano_passado = hoje.replace(year=2021)
7  mes_passado = hoje.replace(month=4)
```

Conversões para strings



Se quisermos converter as datas para strings, seja para enviar para internet, salvar em um arquivo, usar no banco de dados e etc.. temos a seguinte tabela.

Caractere	O que significa	Exemplo
%d	Dia	01
%a	Dia da semana abreviado	Seg
%A	Dia da semana completo	Segunda
%m	Mês	01
%b	Mês abreviado	Jan
%B	Mês completo	Janeiro
%Y	Ano	2022

Tabela completa: <https://docs.python.org/pt-br/3/library/datetime.html#strftime-and-strptime-format-codes>

Conversão com strftime



Para converter, usamos o método **strftime** e passamos uma string com a formatação que desejamos:

```
1  from datetime import date
2
3  hoje = date.today()
4
5  hoje.strftime('%d-%m-%Y') # 23-05-2022
6
7  hoje.strftime('%d de %B de %Y')
8  # 23 de May de 2022
9
10 hoje.strftime('%A, %d de %B de %Y')
11 # Monday, 23 de May de 2022
```

Localização de datas

Caso seu sistema não seja em inglês, a localização do sistema deve ser feita. Podemos fazer isso usando a biblioteca nativa **locale**.

```
1  from datetime import date
2  import locale
3
4  hoje = date.today()
5
6  # Localização para o idioma do SO
7  locale.setlocale(locale.LC_ALL, '')
8
9  hoje.strftime('%d de %B de %Y')
10 # 23 de maio de 2022
11
12 hoje.strftime('%A, %d de %B de %Y')
13 # segunda, 23 de maio de 2022
```

Valores possíveis para locale



As variações de locale servem para qualquer localidade. Porém, se manifestam de maneira diferente em sistemas operacionais diferentes.

```
1  import locale
2
3  # Mostra as abreviações de localizações
4  locale.locale_alias
5
6  # Abreviações de localizações no windows
7  locale.windows_locale
```

Valores possíveis para locale



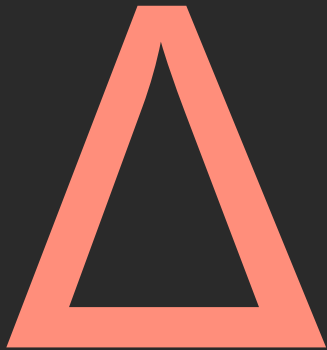
As
ma

**Vale a pena uma live sobre
localização de sistemas?**

es.



```
3 # Mostre as abreviações de localizações
4 locale.locale_alias
5
6 # Abreviações de localizações no windows
7 locale.windows_locale
```



Timedelta

Variações de tempo/espço Δ



Se quisermos manipular as datas usando o tempo/espço. Usando Δ .

A ideia é conseguirmos fazer contas usando um objeto chamado **timedelta**.

```
1  from datetime import date, timedelta
2
3  hoje = date.today()
4
5  ontem = hoje - timedelta(days=1)
6  amanha = hoje + timedelta(days=1)
7
8  semana_que_vem = hoje + timedelta(days=7)
9
10 mes_passado = hoje - timedelta(month=1)
```

Time

Manipulando o
tempo

Time



Time é o objeto responsável por atribuições de horas, minutos, segundos e microssegundos. De forma independente os dias.

```
1  from datetime import time
2
3  hora_da_live = time(22, 0)
4  # datetime.time(22, 0)
5
6  time(
7      hour=22, minute=0,
8      second=3, microsecond=384
9  )
```

Também conta com rica comparação



```
1  from datetime import time
2
3  time(22, 0) == time(23, 0)  # False
4  time(22, 0) != time(23, 0)  # True
5  time(22, 0) > time(23, 0)   # False
6  time(22, 0) >= time(23, 0)  # False
7  time(22, 0) < time(23, 0)   # True
8  time(22, 0) <= time(23, 0)  # True
```

Alterações em horários



Os **objetos time NÃO são compatíveis com timedeltas**.

Isso tem uma razão de ser. Na lógica do relógio não podemos alterar uma hora para negativa sem ser outro dia. Mas como a data não está sendo considerada. Não é possível de se fazer essa operação. A única forma é usar `.replace()`, caso exista e necessidade.

```
1  from datetime import time
2
3  t = time(22,0)
4  t.replace(hour=2)
5  # datetime.time(2, 0)
```

Parser do formato ISO



```
1  from datetime import time
2
3  time.fromisoformat('22:00')
4  # datetime.time(22, 0)
5
6  time.fromisoformat('22:00:14')
7  # datetime.time(22, 0, 14)
8
9  time.fromisoformat('22:40:13:000843')
10 # datetime.time(22, 40, 13, 843)
```

Conversão para ISO



```
1  from datetime import time
2
3  t = time(22, 22, 22, 22)
4  t.isoformat() # '22:22:22.000022'
5
6  t.isoformat(timespec='auto')      # '22:22:22.000022'
7  t.isoformat(timespec='minutes')   # '22:22'
8  t.isoformat(timespec='seconds')   # '22:22:22'
```

Formatação



Caractere	O que significa	Exemplo
%H	Hora	00 - 23
%I	Hora do relógio de ponteiro	00 - 12
%p	Equivalente de localidade	AM / PM
%M	Minutos	00 - 59
%S	Segundos	00-59
%f	Microsegundos	000000 - 999999

```
1 from datetime import time
2
3 t = time(22, 0)
4 t.strftime('%I:%M:%S %p')
5 # '10:00:00 PM'
```


0 tempo consciente

timez
one

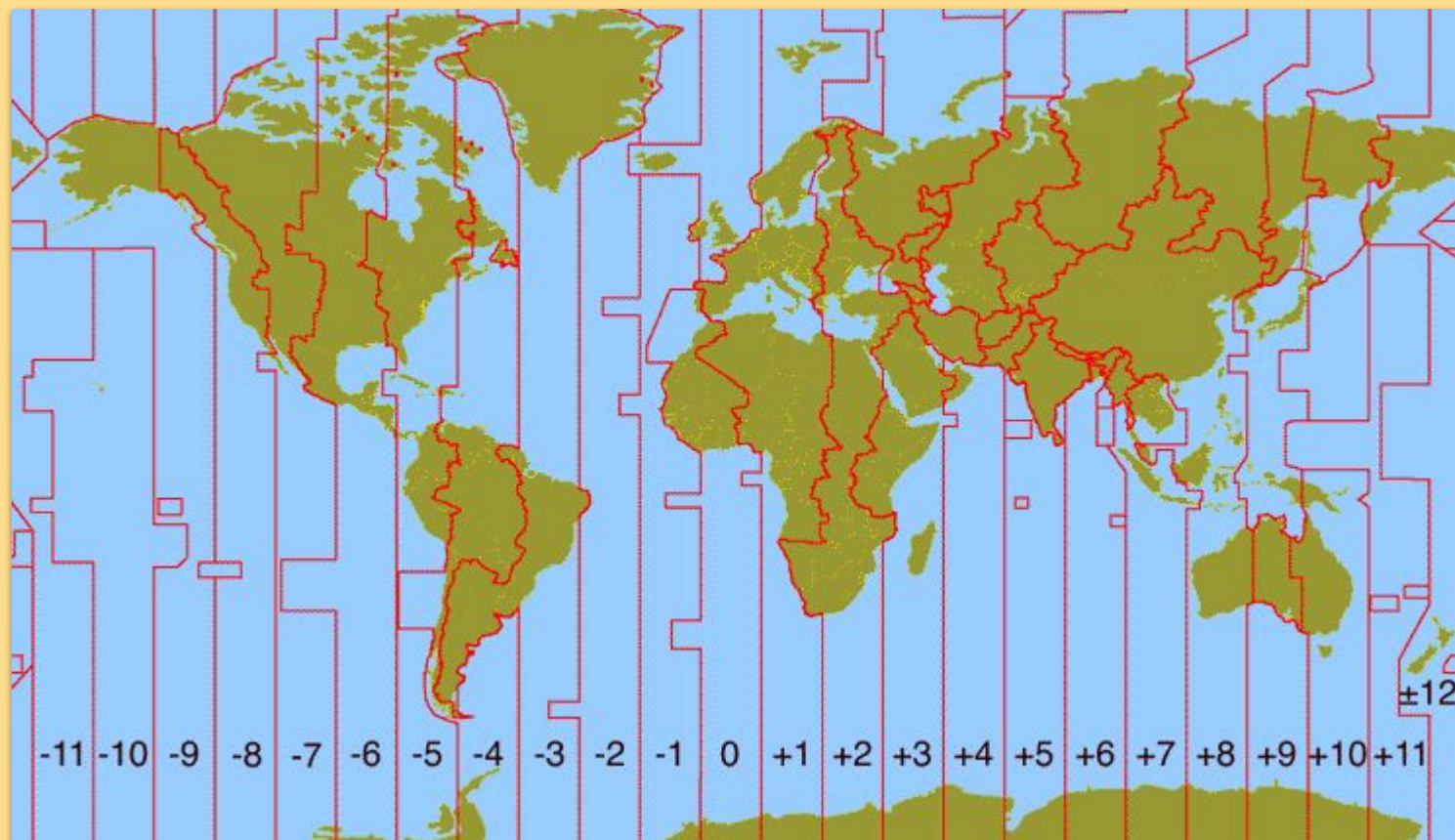
Ingênuo vs Consciente



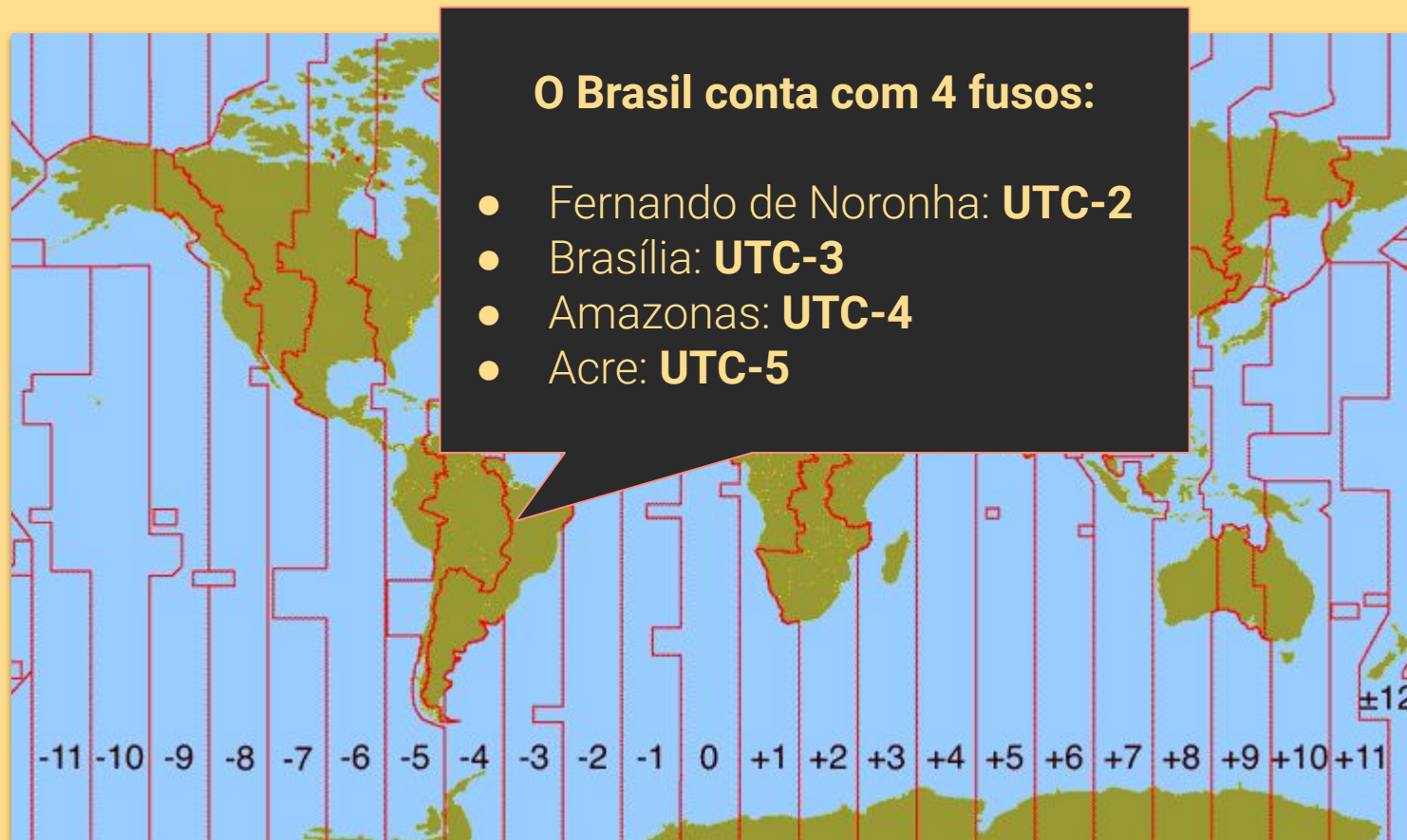
Uma derivação importante quando estamos falando sobre horários é que horários podem ser **Ingênuos** e **conscientes**.

- **Ingênuo** é um horário qualquer, que **não carrega informações sobre timezone**
- **Consciente** é um horário que **contém o timezone**

Tempo Universal Coordenado – UTC



Tempo universal Coordenado



Usando timezone



```
1  from datetime import time, timedelta, timezone
2
3  tz_fnt = timezone(timedelta(hours=-2))
4  tz_brt = timezone(timedelta(hours=-3))
5  tz_amt = timezone(timedelta(hours=-4))
6  tz_act = timezone(timedelta(hours=-4))
7
8  horario_da_live = time(22, 0, tzinfo=tz_brt)
9
10 t = time(22, 0, tzinfo=timezone(timedelta(hours=-3)))
```

Campos adicionais de strftime



Caractere	O que significa	Exemplo
%z	Diferença do UTC	-0300
%Z	Nome do fuso	Atributo no do fuso

```
1 from datetime import time, timedelta, timezone
2
3 tz_fnt = timezone(timedelta(hours=-2))
4
5 t = time(22, 0, tzinfo=tz_fnt)
6 t.strftime('%I:%M:%S %p - UTC:%z - %Z')
7 # '10:00:00 PM - UTC:-0200 - Noronha'
```

date
time

Juntando tudo

datetime



O objeto datetime representa os objetos **date** e **time**. Pode ser usado de forma ingênua e consciente e conta com alguns métodos legais para descobrirmos.

```
1  from datetime import datetime, timedelta, timezone
2
3  hora_da_live = datetime(
4      year=2022,
5      month=5,
6      day=23,
7      hour=22,
8      minute=0,
9      second=0,
10     tzinfo=timezone(timedelta(hours=-3))
11 )
12
13 agora = datetime.now()
```


Definições



Os campos obrigatórios são referentes a **date**. E a resposta padrão sempre voltará com hora e minutos em 0.

```
1  from datetime import datetime
2
3  datetime(2022, 5, 23)
4  # datetime.datetime(2022, 5, 23, 0, 0)
```

Operações



o objeto datetime conta com toda a aritmética disponível nos time deltas.

```
1  from datetime import datetime, timedelta
2
3  dn = datetime.now()
4
5  ontem_esse_horario = dn - timedelta(days=1)
6  amanha_esse_horario = dn + timedelta(days=1)
7
8  delta = dn - ontem_esse_horario
9  # datetime.timedelta(days=1)
```

Operações podem gerar dados



```
1  from datetime import datetime, timedelta
2
3  meu_aniversario = datetime(1993, 6, 3)
4
5  meu_aniversario - datetime.now()
6  # timedelta(days=-10582, seconds=30880, microseconds=993473)
```

Conversões de timezone



— □ ×

```
1  from datetime import time, timedelta, timezone
2
3  tz_fnt = timezone(timedelta(hours=-2))
4  tz_brt = timezone(timedelta(hours=-3))
5
6  live_dt = datetime(2022, 5, 23, 22, 0, tzinfo=tz_brt)
7  live_dt_fnt = live_dt.astimezone(tz_fnt)
8  live_dt_fnt.time() # datetime.time(23, 0)
```

Formatação



O datetime conta com todos os formatadores de date, time e timezone

```
1 import locale
2 from datetime import datetime, timedelta, timezone
3
4 live_dt = datetime(2022, 5, 23, 22, 0)
5 live_dt.strftime(
6     '%d(%a|%A) de(%b|%B|%m) de %Y. Às %H:%M:%S %p'
7 )
8 # '23(Mon|Monday) de(May|May|05) de 2022. Às 22:00 PM'
9
10 locale.setlocale(locale.LC_ALL, '')
11 # 'pt_BR.UTF-8'
12
13 live_dt.strftime(
14     '%d(%a|%A) de(%b|%B|%m) de %Y. Às %H:%M:%S %p'
15 )
16 # '23(seg|segunda) de(mai|maio|05) de 2022. Às 22:00:00 '
```

Parser



O datetime é o único objeto que conta com um parser de strings, o método **strptime**.

```
1  from datetime import datetime
2  datetime.strptime(
3      '23 de Maio de 2022', # String com a data
4      '%d de %B de %Y'      # Formatador
5  )
6
7  # datetime.datetime(2022, 5, 23, 0, 0)
```

Extras

Não se vive só de
embutidos

dateparser



dateparser é uma biblioteca externa que pode nos ajudar a parsear valores sem escrever os formatadores. É ótimo para sistemas que carregam dados de diversas fontes ou para web scraping.

```
1  # pip install dateparser
2  from dateparser import parse
3
4  parse('1/1/2022') # datetime(2022, 1, 1, 0, 0)
5
6  parse('1 de maio de 2022') # datetime(2022, 5, 1, 0, 0)
7
8  parse('1 de maio de 2022 22:50') # datetime(2022, 5, 1, 22, 50)
9
10 parse('22:50:00') # datetime(2022, 5, 23, 22, 50) - data de hoje
11
12 parse('1 minuto atrás') # datetime(2022, 5, 23, 15, 49)
```


Pendulum



Pendulum é uma biblioteca de datetime externa para casos onde o datetime nativo não atende todas as expectativas.

- Casos complexos de parse
- Por padrão os tipos já vem com timezone
- timezones descritivos
- Tem operações complexas:
 - Range de datetime
 - subtração e edição sem delta
- Formatadores mais simples
 - Sem uso de %
 - Conversões diretas para RFCs

Pendulum



```
1  # pip install pendulum
2  import pendulum
3
4  pendulum.now( )
5  # DateTime(2022, 5, 23, 15, 57, 58, 2242, tzinfo=Timezone('America/Sao_Paulo'))
6
7  pendulum.today( )
8  # DateTime(2022, 5, 23, 0, 0, 0, tzinfo=Timezone('America/Sao_Paulo'))
9
10 pendulum.yesterday( )
11 # DateTime(2022, 5, 22, 0, 0, 0, tzinfo=Timezone('America/Sao_Paulo'))
12
13 pendulum.duration(days=7)
14 # Duration(weeks=1, days=0)
```



Assista a live de
Python # 137

```
1 # pip install pendulum
2 import pendulum
3
4 pendulum.now()
5 # DateTime(2022, 5, 23, 0, 0, 0, tzinfo=Timezone('America/Sao_Paulo'))
6
7 pendulum.today()
8 # DateTime(2022, 5, 23, 0, 0, 0, tzinfo=Timezone('America/Sao_Paulo'))
9
10 pendulum.yesterday()
11 # DateTime(2022, 5, 22, 0, 0, 0, tzinfo=Timezone('America/Sao_Paulo'))
12
13 pendulum.duration(days=7)
14 # Duration(weeks=1, days=0)
```

Freezegun



Na hora dos testes sempre temos problemas para manipular datas e horários. Nessa hora o freezegun nos ajuda a pausar o tempo

```
1  # pip install freezegun
2  from freezegun import freeze_time
3  from datetime import datetime
4
5  with freeze_time("2012-01-14 12:00:01"):
6      dt = datetime.now()
7
8  >> dt # FakeDatetime(2012, 1, 14, 12, 0, 1)
```



picpay.me/dunossauro



apoia.se/livedepython



pix.dunossauro@gmail.com



Ajude o projeto <3



Links citados

Dateparser: <https://github.com/scrapinghub/dateparser>

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

Live de Pendulum: <https://youtu.be/iy3fGZEK8f4>

Freezegun: <https://github.com/spulec/freezegun>