

Trabalhando com Datas e Horas Python



Trabalhando com Datas

Vamos importar as bibliotecas necessárias

```
[44] # Biblioteca para recursos de Data
import datetime
# Biblioteca para recursos cálculos de fuso horário
import pytz
```

Vamos gerar a Data e Hora de agora

```
[65] # Identificando agora [ Dia e Hora ]
Agora = datetime.datetime.now()
print( Agora )

2021-04-25 21:51:17.137997
```

Vamos gerar apenas a Data

```
[67] # identificando a data de hoje
Hoje = datetime.date.today()
print( Hoje )

2021-04-25
```

Vamos quebrar o Ano, Mês e dia

```
[70] # identificando o Ano
print( 'Ano:', Hoje.year )

# identificando o Mes
print( 'Mes:', Hoje.month )

# identificando o Dia
print( 'Dia:', Hoje.day )

Ano: 2021
Mes: 4
Dia: 25
```

Vamos criar uma data passando parâmetros

```
[72] # Agrupando valores para virar uma Data
Dia = 1
Mes = 1
Ano = 2021

Data_Agrupada = datetime.datetime(Ano, Mes, Dia)
print( Data_Agrupada )

2021-01-01 00:00:00
```

Identificar o mês de uma data

```
[77] # Identificando o Mês de uma Data
      Identificando_Mes = Hoje.strftime('%B')
      print( Identificando_Mes )
```

April

Identificar o dia da semana de uma data

```
[78] # Identificando o dia da Semana
      Identificar_Dia_Semana = Hoje.strftime('%A')
      print( Identificar_Dia_Semana )
```

Sunday

Identificar o dia da semana de uma data mas valor numérico

```
[79] # Identificando o dia da Semana valor numerico
      # 0 Domingo, 1 Segunda ...
      Dia_Semana_Valor = Hoje.strftime('%w')
      print( Dia_Semana_Valor )
```

0

Identificar os dias do ano

```
[81] # Identificando a quantidade dias do ano
      Dias_do_Ano = Hoje.strftime('%j')
      print( 'Já se passaram:', Dias_do_Ano, 'dias' )
```

Já se passaram: 115 dias

Identificar as semanas do ano

```
[84] # Identificando o numero da semana do ano
      Semana_do_Ano = Hoje.strftime('%U')
      print( 'Já se passaram:', Semana_do_Ano, 'Semanas' )
```

Já se passaram: 17 Semanas

Identificar horário do dia

```
[91] # Identificando se é Manhã ou Tarde
      # AM - Manhã, PM - Tarde
      Horario_do_Dia = Hoje.strftime('%p')
      print( Horario_do_Dia )
```

AM

Ajustando a posição da Data

```
[96] # formatando o horário

Dia_Mes_Ano = '%d/%m/%y'
Mes_Dia_Ano = '%m/%d/%y'
Ano_Mes_Dia = '%y/%m/%d'

print('Dia/Mes/Ano:', Hoje.strftime( Dia_Mes_Ano ))
print('Mes/Dia/Ano:', Hoje.strftime( Mes_Dia_Ano ))
print('Ano/Mes/Dia:', Hoje.strftime( Ano_Mes_Dia ))
```

```
Dia/Mes/Ano: 25/04/21
Mes/Dia/Ano: 04/25/21
Ano/Mes/Dia: 21/04/25
```

Ajustando ao fuso horário

```
[99] # Formatando para horários locais mundiais

Formato = '%d/%m/%Y, %H:%M:%S'

# Gerando o horário de Nova Iorque
Local_Pais = pytz.timezone('America/New_York')
Horario_Local_NovaIork = datetime.datetime.now(Local_Pais)
print('Nova Iorque:', Horario_Local_NovaIork.strftime(Formato) )

# Gerando o horário de Londres
Local_Pais_2 = pytz.timezone('Europe/London')
Horario_Local_Londres = datetime.datetime.now(Local_Pais_2)
print('Londres:', Horario_Local_Londres.strftime(Formato))

# Gerando o horário do Japão
Local_Pais_3 = pytz.timezone('Asia/Tokyo')
Horario_Local_Japao = datetime.datetime.now(Local_Pais_3)
print('Japão:', Horario_Local_Japao.strftime(Formato))
```

```
Nova Iorque: 25/04/2021, 18:10:20
Londres: 25/04/2021, 23:10:20
Japão: 26/04/2021, 07:10:20
```

Convertendo uma Data para inteiro

```
[102] # Convertendo Data numero inteiro
Data_Inteira = Hoje.toordinal()
print( Data_Inteira )
```

```
737905
```

Convertendo uma String para Data

```
[100] # Convertendo string para Data
Data_01 = 'Wednesday, June 6, 2018'
Data_02 = '6/6/18'
Data_03 = '06-06-2018'

# Define dates as datetime objects
Data_01_Ajustada = datetime.datetime.strptime(Data_01,
                                                '%A, %B %d, %Y')
Data_02_Ajustada = datetime.datetime.strptime(Data_02, '%m/%d/%y')
Data_03_Ajustada = datetime.datetime.strptime(Data_03, '%m-%d-%Y')

print( Data_01_Ajustada )
print( Data_02_Ajustada )
print( Data_03_Ajustada )

2018-06-06 00:00:00
2018-06-06 00:00:00
2018-06-06 00:00:00
```

Operando com datas

```
[103] # Atribuindo datas futuras
Futuro = datetime.date.fromordinal(Hoje.toordinal()+45)
print( Futuro )

# Atribuindo datas passadas
Passado = datetime.date.fromordinal(Hoje.toordinal()-55)
print( Passado )

# Calculando intervalo de duas datas
Intervalo = Futuro - Passado

print( Intervalo.days, ' dias' )

2021-06-09
2021-03-01
100 dias
```

Funções com Data

```
[104] # Criando uma regra de dias da Semana
def Dias_Semana(Numero_Semana):
    Dias = ['Segunda', 'Terça', 'Quarta', 'Quinta',
            'Sexta', 'Sabado', 'Domingo']
    return Dias[Numero_Semana]

Numero_Semana = int ( Hoje.weekday() )

print('Hoje é', Dias_Semana(Numero_Semana) )

Hoje é Domingo
```

Trabalhando com Horas

Extrair Horas de uma data completa

```
[107] # Extraíndo apenas a hora
      print('Hora: ', Agora.hour)

      # # Extraíndo apenas os minutos
      print('Minutos: ', Agora.minute)
```

```
Hora: 21
Minutos: 51
```

Converter para hora temporal

```
[126] # Converter um horario para horario temporal
      Hora_Temporal = datetime.datetime.timestamp(Agora)

      print('Hora Temporal:', Hora_Temporal)
```

```
Hora Temporal: 1619387477.137997
```

Converter de hora temporal para hora padrão

```
[115] # Convertendo hora temporal
      Hora_Temporal_Numericc = 1619387477.137997

      #convert timestamp to datetime object
      Converter = datetime.datetime.fromtimestamp(Hora_Temporal_Numericc)

      print('Hora convertida', Converter)
```

```
Hora convertida 2021-04-25 21:51:17.137997
```

Converter String em Horário

```
[117] # Converter String em Hora
      # Ajustando a horario com 12 horas
      Hora_Formatada = 'Apr 25 2021 9:40AM'
      Ajustando = datetime.datetime.strptime(Hora_Formatada,
                                             '%b %d %Y %I:%M%p')

      print('Horário:', Ajustando.time() )
```

```
Horário: 09:40:00
```

Converter String em Horário

```
[119] # Converter String em Hora
      # Ajustando a horario com 24 horas

      Hora_Formatada_2 = '2018-06-29 12:15:27.243860'
      Ajustando_2 = datetime.datetime.strptime(Hora_Formatada_2,
                                             '%Y-%m-%d %H:%M:%S.%f')

      print('Horário:', Ajustando_2.time() )
```

```
Horário: 12:15:27.243860
```

Biblioteca 'Parse'

Recurso rápido para converter strings

'dateutil.parser' - Este módulo oferece um analisador de string de data / hora genérico que é capaz de analisar os formatos mais conhecidos para representar uma data e / ou hora.

```
[131] # Função para todos tipos de datas de string
      from dateutil.parser import parse
```

```
Formatos_Datas = [
    '2021-04-25 09:12:32.243860',
    'Apr 25 2021 9:40PM',
    'Apr 25 2021 at 9:40AM',
    'April 25, 2021, 19:29:55',
    'Sun, 25/04/2021, 11:22PM',
    'Sun, 25 April, 2021',
    '2021-04-25T10:12:45Z',
    '2021-04-25 17:08:00.586525+00:00',
    '2021-04-25 17:08:00.586525+05:00',
    'Sunday , 25th April, 2021 at 4:30pm'
]
```

```
for Consulta_Data in Formatos_Datas:
    print('Parsing: ' + Consulta_Data)
    Data = parse(Consulta_Data)
    print(Data.date())
    print(Data.time())
    print(Data.tzinfo)
    print('----- \o/ -----')
```

[133] Convetendo: 2021-04-25 09:12:32.243860	Convetendo: Sun, 25 April, 2021
2021-04-25	2021-04-25
09:12:32.243860	00:00:00
None	None
----- próximo -----	----- próximo -----
Convetendo: Apr 25 2021 9:40PM	Convetendo: 2021-04-25T10:12:45Z
2021-04-25	2021-04-25
21:40:00	10:12:45
None	tzlocal()
----- próximo -----	----- próximo -----
Convetendo: Apr 25 2021 at 9:40AM	Convetendo: 2021-04-25 17:08:00.586525
2021-04-25	2021-04-25
09:40:00	17:08:00.586525
None	tzlocal()
----- próximo -----	----- próximo -----
Convetendo: April 25, 2021, 19:29:55	Convetendo: 2021-04-25 17:08:00.586525
2021-04-25	2021-04-25
19:29:55	17:08:00.586525
None	tzoffset(None, 18000)
----- próximo -----	----- próximo -----
Convetendo: Sun, 25/04/2021, 11:22PM	Convetendo: Sunday , 25th April, 2021
2021-04-25	2021-04-25
23:22:00	16:30:00
None	None
----- próximo -----	

Final

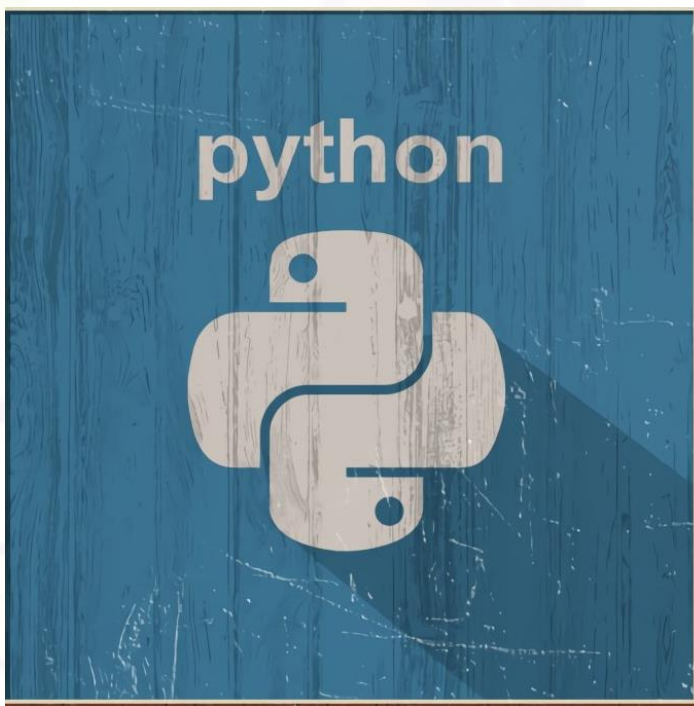
Esse guia é sobre como tratar diferentes formas de datas e horário em python.

Guia da documentação das bibliotecas caso queira mais informação

<https://dateutil.readthedocs.io/en/stable/parser.html>

<https://pypi.org/project/pytz/>

<https://docs.python.org/pt-br/3/library/datetime.html>



Odemir Depieri Jr

Software Engineer Sr
Tech Lead
Specialization AI