

```
85     if self.context_index == -1:
86         if symbol == key.UP and not self.active_index == 0:
87             self.menu_labels[self.active_index].color = [255, 255, 255, 255]
88             self.active_index -= 1
89             self.mags_dt = self.get_act_color_mag()
90         elif symbol == key.DOWN and not self.active_index == 3:
91             self.menu_labels[self.active_index].color = [255, 255, 255, 255]
92             self.active_index += 1
93             self.mags_dt = self.get_act_color_mag()
94         elif symbol == key.ENTER:
95             if self.active_index == 3:
96                 pyglet.app.exit()
97             else:
98                 self.context_index = self.active_index
99             elif symbol == key.ESCAPE:
100                 if self.context_index == -1:
101                     pyglet.app.exit()
102                 else:
103                     self.context_index = -1
104             elif self.context_index == 1:
105                 if symbol == key.ESCAPE:
106                     self.context_index = -1
107                 else:
108                     self.cur_game.on_key_press(symbol, modifiers)
109             else:
110                 if symbol == key.ESCAPE:
111                     self.context_index = -1
```

E-BOOK

# Python Avançado: Como pegar dados de Sites

VAROS

# Python Avançado: Como pegar dados de Sites

Seja muito bem-vindo ao meu guia de Python Avançado: Como pegar dados de Sites. Meu nome é Brenno Sullivan, eu sou Head de dados da VAROS, programador python e professor do Código.py.

Extrair dados de sites é muito útil para diversas áreas e funções e no mercado financeiro não seria diferente. Fazer isso pode te ajudar em várias situações como ter acessos a informações em tempo real, monitorar o sentimento do mercado, automatizar processos e muito mais.

Você pode, por exemplo, coletar dados de notícias, cotações de ações, preços de commodities e indicadores econômicos. Dessa forma, você consegue tomar suas próprias decisões de investimento ou ajudar um analista da empresa em que você trabalha a fazer isso.

Também é possível extrair dados de mídias sociais, fóruns e blogs para avaliar o sentimento do mercado em relação a determinados ativos ou eventos econômicos.

Enfim, independente do seu objetivo, extrair informações de sites na internet vai ser muito útil para você e para isso, nós vamos usar uma técnica chamada web scraping.

Web scraping nada mais é do que a gente utilizar a programação para conseguir acessar um site e a partir deste site coletar todos os dados que a gente precisa.

Vamos começar?

Para te ensinar como fazer isso, nada melhor do que trazer uma aplicação prática do mercado financeiro. Vamos extrair os dados históricos do mini contrato de dólar direto do site da Bolsa de Valores.

Como eu falei, a gente vai pegar esses dados diretamente do site da B3 utilizando a técnica de web scraping. Na verdade, vamos trabalhar com o site da BM&F, a antiga B3, mas dá no mesmo. A URL que vamos extrair os dados será a seguinte: <https://www2.bmf.com.br/pages/portal/bmfbovespa/boletim1/SistemaPregao1.asp>

Deste link vamos extrair todos os dados que a gente precisa do mini dólar, mas nele você também encontra dados de outras mercadorias como mini índice, câmbio, contratos futuros de commodities e outras coisas.

Para fazer isso manualmente, basta selecionar de qual data a gente gostaria de extrair as informações e, em seguida, selecionar a mercadoria.

## Exemplo:

### WDO Mini Opção sobre USD

MERCADO FUTURO									ATUALIZADO EM: 14/06/2024
Dados	Vol.	Dados							
VENCTO	PREÇO ABERT.	PREÇO MÍN.	PREÇO MÁX.	PREÇO MÉD.	ÚLT. PREÇO	AJUSTE	VAR. PTOS.	ÚLT. OF. COMPRA	ÚLT. OF. VENDA
N24	5.385,000	5.352,000	5.394,500	5.374,097	5.382,500	5.386,0560	7,3120+	5.389,500	5.354,00C
Q24	5.403,000	5.370,000	5.411,500	5.391,957	5.403,000	5.404,1380	7,1040+	5.400,000	5.388,00C
U24	5.417,000	5.395,000	5.425,000	5.414,919	5.416,000	5.421,0020	7,0790+	5.416,000	5.425,00C
V24	0,000	0,000	0,000	0,000	0,000	5.437,0450	6,7970+	5.412,500	5.436,50C
X24	0,000	0,000	0,000	0,000	0,000	5.455,6960	6,6240+	5.419,500	5.474,50C
Z24	0,000	0,000	0,000	0,000	0,000	5.470,6530	6,8450+	5.432,500	5.508,50C
F25	0,000	0,000	0,000	0,000	0,000	5.487,7230	6,7560+	5.442,500	5.491,00C
G25	0,000	0,000	0,000	0,000	0,000	5.507,1470	6,5070+	0,000	0,00C
H25	0,000	0,000	0,000	0,000	0,000	5.525,9750	5,9670+	0,000	0,00C
J25	0,000	0,000	0,000	0,000	0,000	5.545,7870	6,1520+	0,000	0,00C
K25	0,000	0,000	0,000	0,000	0,000	5.563,1120	5,9480+	0,000	0,00C
M25	0,000	0,000	0,000	0,000	0,000	5.586,3430	5,0670+	0,000	0,00C
N25	0,000	0,000	0,000	0,000	0,000	5.609,3700	5,8370+	0,000	0,00C
V25	0,000	0,000	0,000	0,000	0,000	5.693,6870	5,8570+	0,000	0,00C
F26	0,000	0,000	0,000	0,000	0,000	5.774,7230	5,9140+	0,000	0,00C
J26	0,000	0,000	0,000	0,000	0,000	5.863,5960	4,8340+	0,000	0,00C
N26	0,000	0,000	0,000	0,000	0,000	5.948,9860	2,1860+	0,000	0,00C
V26	0,000	0,000	0,000	0,000	0,000	6.051,8130	0,1630-	0,000	0,00C
F27	0,000	0,000	0,000	0,000	0,000	6.143,5290	1,4800-	0,000	0,00C
J27	0,000	0,000	0,000	0,000	0,000	6.244,4240	3,6910-	0,000	0,00C
N27	0,000	0,000	0,000	0,000	0,000	6.354,4730	7,5700-	0,000	0,00C
F28	0,000	0,000	0,000	0,000	0,000	6.577,6720	14,6550-	0,000	0,00C
N28	0,000	0,000	0,000	0,000	0,000	6.818,2960	14,2510-	0,000	0,00C
F29	0,000	0,000	0,000	0,000	0,000	7.043,5550	15,4430-	0,000	0,00C
MERCADO DE OPÇÕES S/DISPONÍVEL - COMPRA									
Dados	Vol.	Dados							
SÉRIE/CÓDIGOS	PREÇO ABERT.	PREÇO MÍN.	PREÇO MÁX.	PREÇO MÉD.	ÚLT. PREÇO	VAR. PTOS.	PRÊMIO DE REFERÊNCIA	ÚLT. OF. COMPRA	ÚLT. OF. VENDA
NS8Q N24C005100	292,101	292,101	292,101	292,101	292,101	0,0000	0,000	0,000	0,00C
NS8R N24C005125	0,000	0,000	0,000	0,000	0,000	0,0000	0,000	0,000	0,00C
NS8S N24C005150	0,000	0,000	0,000	0,000	0,000	0,0000	0,000	0,000	0,00C
NS8V N24C005200	0,000	0,000	0,000	0,000	0,000	0,0000	0,000	0,000	0,00C
NS8W N24C005225	0,000	0,000	0,000	0,000	0,000	0,0000	0,000	0,000	0,00C
NS89 N24C005325	93,896	93,896	93,896	93,896	93,896	0,0000	0,000	0,000	0,00C
NS93 N24C005400	40,000	40,000	40,000	40,000	40,000	40,0000	0,000	0,000	0,00C
NS96 N24C005475	0,000	0,000	0,000	0,000	0,000	0,0000	0,000	0,000	0,00C

É claro que não queremos fazer isso para uma única data. Queremos construir um histórico de dados. Então, seria impossível – ou, pelo menos, seria burrice – fazer isso manualmente.

É por isso que faremos esse trabalho de forma automática com o Python. Para isso, o que precisaremos fazer é definir no python de qual data gostaríamos de extrair os dados e o python fará o “trabalho de corno” de acessar este site diversas vezes, em várias datas diferentes e extrair os dados para nós.

Para isso, a primeira coisa que você vai fazer é importar alguns pacotes que serão necessários para essa tarefa. São eles: pandas, selenium, webdriver\_manager, time e date time.



```
import pandas as pd
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from webdriver_manager.chrome import ChromeDriverManager
from selenium.webdriver.chrome.options import Options
import time
from datetime import datetime
```

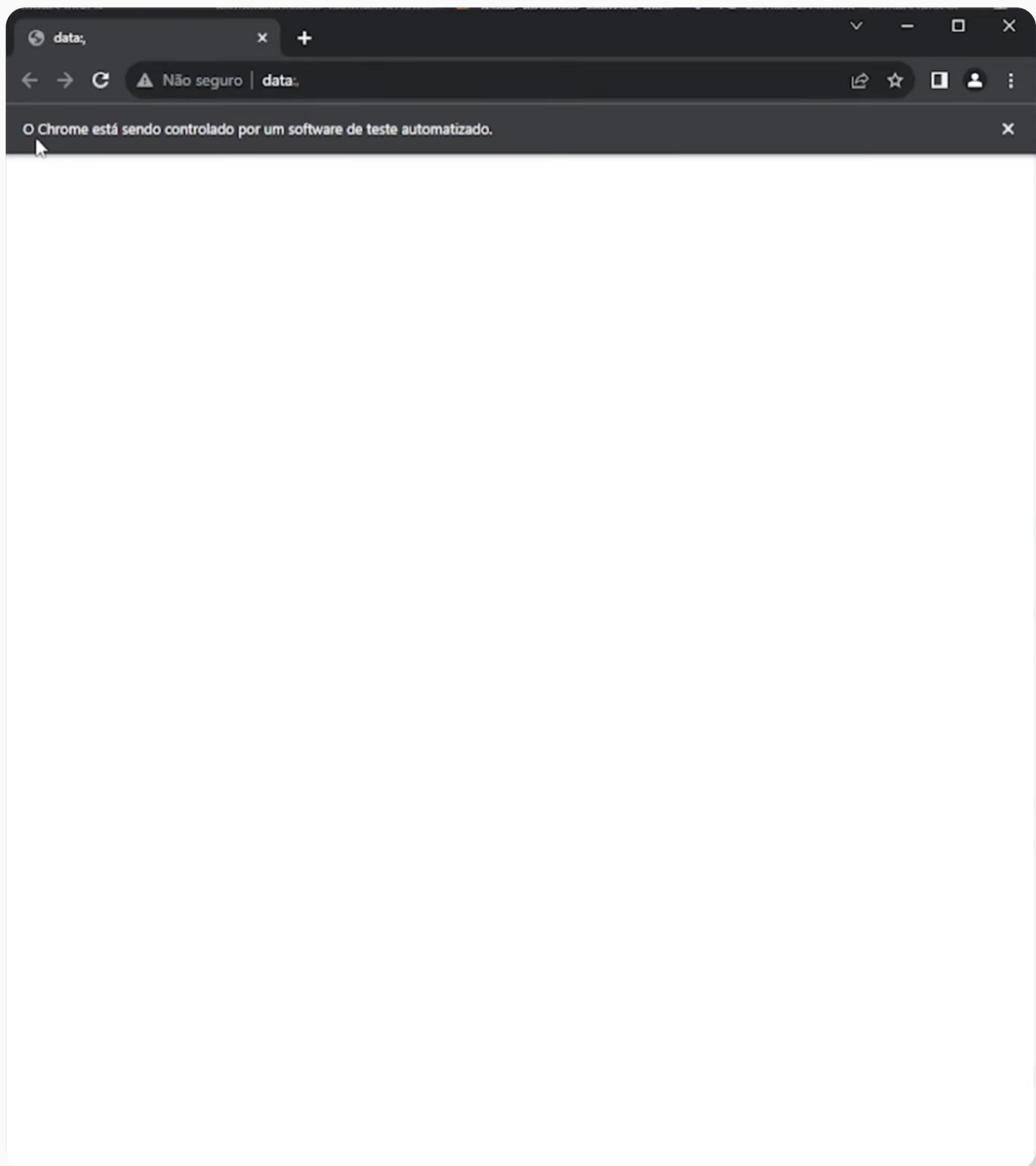
O pandas é o pacote do python usado para fazer análise de dados. Já o selenium e o webdriver\_manager serão usados para o web scraping em si. Ainda vamos usar outros pacotes auxiliares como o time, para forçar um tempo de espera no código e o datetime para trabalhar com datas.

Não se preocupe se você nunca usou esses pacotes. Conforme formos avançando no nosso código, eu te explicarei melhor sobre como vamos usar cada um deles.



```
driver = webdriver.Chrome(service=Service(ChromeDriverManager().install()))
```

Com essa linha de código, o meu robô no python é capaz de abrir o Google Chrome. Basta executar o código que uma nova janela do Chrome será aberta.



Em seguida, vamos definir a data de início para a coleta dos dados. Neste exemplo, comecei a minha base histórica pela **data 12/04/2023**, mas você pode usar a data que você quiser.



```
data_contratos = '12/04/2023'
```

Com a data definida, a gente precisa da URL onde o dado que queremos extrair está localizado. Para encontrá-la, é só você acessar aquela URL principal e seguir o passo a passo manual que eu te ensinei no início deste guia.

<https://www2.bmf.com.br/pages/portal/bmfbovespa/boletim1/SistemaPregao1.asp?pagetype=pop&caminho=Resumo%20Estat%EDstico%20-%20Sistema%20Preg%E3o&Data=14/06/2024&Mercadoria=WDO>

Pronto. É essa URL que vamos pedir para o nosso robô acessar e trazer as informações. Perceba que uma parte dela é destinada a identificar a data que nós solicitamos os dados.

<https://www2.bmf.com.br/pages/portal/bmfbovespa/boletim1/SistemaPregao1.asp?pagetype=pop&caminho=Resumo%20Estat%EDstico%20-%20Sistema%20Preg%E3o&Data=14/06/2024&Mercadoria=WDO>

Agora que descobrimos isso, o que precisaremos fazer no nosso código é, ao invés de usar uma data fixa nesta URL, usar a variável **data\_contratos** que criamos ali em cima.

Assim, no nosso código, a URL ficará assim:



```
url = f'''  
https://www2.bmf.com.br/pages/portal/bmfbovespa/boletim1/SistemaPregao1.asp?  
pagetype=pop&caminho=Resumo%20Estat%EDstico%20-%20Sistema%20Preg%E3o&Data=  
{data_contratos}&Mercadoria=WDO'''
```



Feito isso, basta solicitar que o nosso robô acesse este site com a seguinte linha de código:



```
driver.get(url)
```



Assim, esse será o resultado do nosso código até agora: um robô que abre o Google Chrome na página que solicitamos.

**Resumo Estatístico - Sistema Pre...**

O Chrome está sendo controlado por um software de teste automatizado.

Sistema Pregão  
RETROATIVO: 12/04/2023 MERCADORIA: Selecionar a Mercadoria

Para visualizar a informação desejada clique no botão  . Para ocultá-la clique no botão  .

**WDO Mini Opção sobre USD**

ATUALIZADO EM: 12/04/2023

**MERCADO FUTURO**

VENCTO	Dados Vol. <input checked="" type="checkbox"/>		Dados <input type="checkbox"/>							
	PREÇO ABERT.	PREÇO MÍN.	PREÇO MÁX.	PREÇO MÉD.	ÚLT. PREÇO	AJUSTE	VAR. PTOS.	ÚLT. OF. COMPRA	ÚLT. OF. VENDA	
K23	5.026,500	4.927,500	5.028,000	4.960,978	4.933,500	4.949,5210	70,0870-	4.930,500	4.933,000	
M23	5.056,000	4.962,000	5.057,000	4.987,794	4.962,000	4.980,1650	70,4410-	4.960,000	4.966,500	
N23	5.052,000	4.990,000	5.052,000	5.003,803	4.990,000	5.007,7270	70,9460-	4.984,000	4.998,500	
Q23	0,000	0,000	0,000	0,000	0,000	0,034,2340	71,3190-	4.990,000	0,000	
U23	0,000	0,000	0,000	0,000	0,000	0,065,3330	71,5960-	0,000	0,000	
V23	0,000	0,000	0,000	0,000	0,000	0,088,7210	71,3980-	0,000	0,000	
Z23	0,000	0,000	0,000	0,000	0,000	0,137,0140	0,0000	0,000	0,000	
F24	0,000	0,000	0,000	0,000	0,000	0,157,6170	70,6270-	0,000	0,000	
G24	0,000	0,000	0,000	0,000	0,000	0,184,1230	70,9350-	0,000	0,000	
H24	0,000	0,000	0,000	0,000	0,000	0,204,2320	69,6300-	0,000	0,000	
V24	5.348,500	5.348,500	5.348,500	5.348,500	5.348,500	5.376,6890	68,0600-	0,000	0,000	
F25	0,000	0,000	0,000	0,000	0,000	0,456,2780	64,7000-	0,000	0,000	

**MERCADO DE OPÇÕES S/DISPONÍVEL - COMPRA**

SÉRIE/CÓDIGOS	Dados Vol. <input checked="" type="checkbox"/>		Dados <input type="checkbox"/>							
	PREÇO ABERT.	PREÇO MÍN.	PREÇO MÁX.	PREÇO MÉD.	ÚLT. PREÇO	VAR. PTOS.	PRÊMIO DE REFERÊNCIA	ÚLT. OF. COMPRA	ÚLT. OF. VENDA	
KR8G K23C004950	61,047	61,047	61,047	61,047	61,047	0,0000	0,000	0,000	0,000	
KR8K K23C005025	0,000	0,000	0,000	0,000	0,000	0,0000	0,000	0,000	0,000	
KR8R K23C005175	0,000	0,000	0,000	0,000	0,000	0,0000	0,000	0,000	0,000	
KR8V K23C005250	0,000	0,000	0,000	0,000	0,000	0,0000	0,000	0,000	0,000	
KR91 K23C005400	0,000	0,000	0,000	0,000	0,000	0,0000	0,000	0,000	0,000	
KR94 K23C005475	0,000	0,000	0,000	0,000	0,000	0,0000	0,000	0,000	0,000	
MR8T M23C005325	0,000	0,000	0,000	0,000	0,000	0,0000	0,000	0,000	0,000	
FS81 F24C006850	0,000	0,000	0,000	0,000	0,000	0,0000	0,000	0,000	0,000	

**MERCADO DE OPÇÕES S/DISPONÍVEL - VENDA**

SÉRIE/CÓDIGOS	Dados Vol. <input checked="" type="checkbox"/>		Dados <input type="checkbox"/>							
	PREÇO ABERT.	PREÇO MÍN.	PREÇO MÁX.	PREÇO MÉD.	ÚLT. PREÇO	VAR. PTOS.	PRÊMIO DE REFERÊNCIA	ÚLT. OF. COMPRA	ÚLT. OF. VENDA	
KRD8 K23P004800	0,000	0,000	0,000	0,000	0,000	0,0000	0,000	0,000	0,000	
KRDG K23P004950	42,642	42,642	65,484	61,654	62,025	0,0000	0,000	0,000	0,000	
KRDK K23P005025	117,613	117,613	117,613	117,613	117,613	0,0000	0,000	0,000	0,000	
KRDN K23P005100	0,000	0,000	0,000	0,000	0,000	0,0000	0,000	0,000	0,000	
MRD7 M23P004875	66,271	66,271	71,271	68,771	71,271	0,0000	0,000	0,000	0,000	
MRDD M23P005000	121.090	121.090	121.090	121.090	121.090	0,0000	0,000	0,000	0,000	

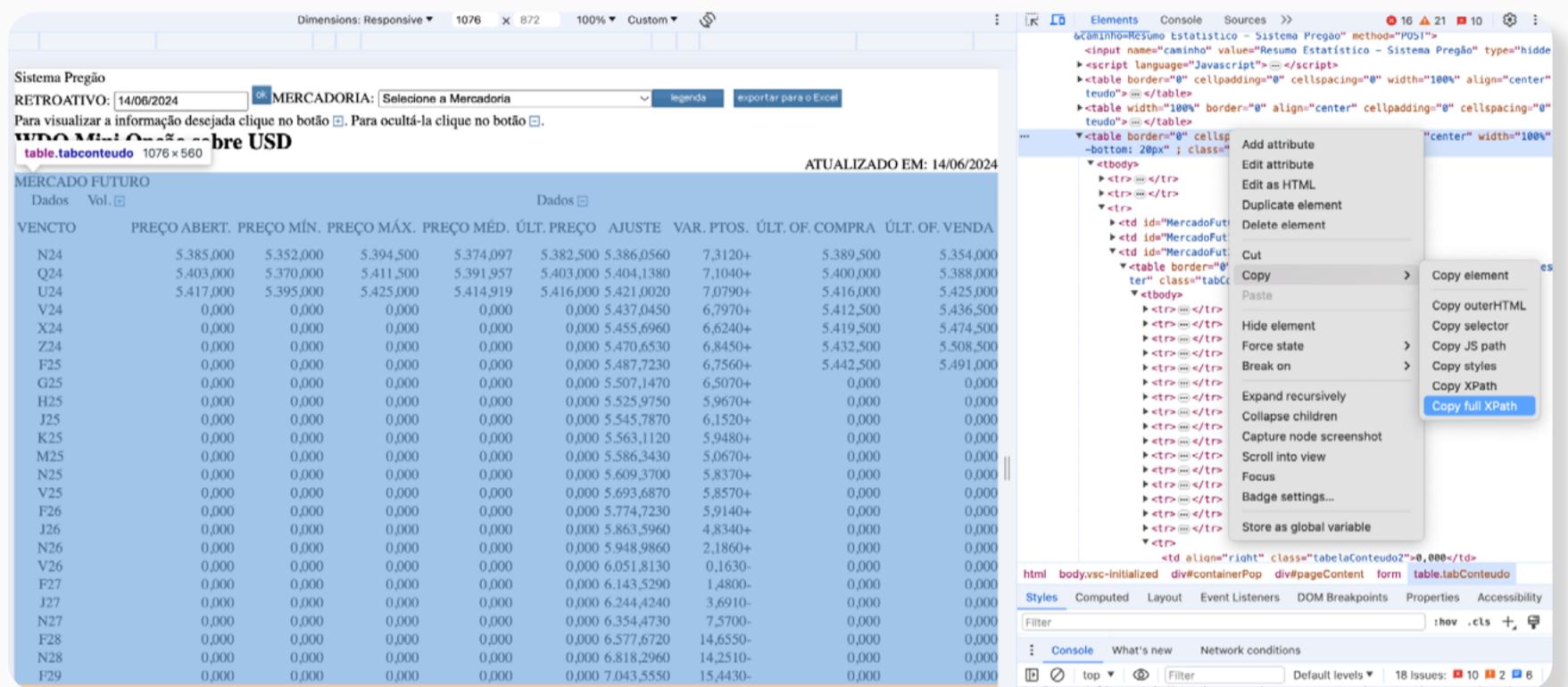
O que a gente precisa descobrir agora é como passar para o robô as informações que queremos dentro desta página.

Para isso, precisaremos inspecionar a página do navegador usando o botão direito do mouse.

Feito isso, uma janela será aberta e conseguiremos visualizar o código fonte do site. Nessa janela podemos explorar o código HTML e identificar os elementos presentes na página.

Podemos perceber que essa página é dividida em tabelas e são essas tabelas que possuem os dados que queremos extrair.

Localizada a tabela no código, basta clicarmos com o botão direito do mouse e ir em “Copy full XPath”.



Feito isso, vamos armazenar essa informação na variável **local\_tabela**.



```
local_tabela = ''
/html/boy/div/div[2]/form[1]/table[3]/tbody/tr[3]/td[3]/table
...

local_indice = ''
/html/boy/div/div[2]/form[1]/table[3]/tbody/tr[3]/td[1]/table
...
```



Com essas informações, basta usar a função `find_element` do driver, para extrairmos todas as informações de dentro da tabela. A partir disso, vamos colocar essas informações dentro do python usando a função `read_html` do pandas, chamado como `pd`. O código fica assim:



```
elemento = driver.find_element("xpath", local_tabela)
elemento_indice = driver.find_element("xpath", local_indice)

html_tabela = elemento.get_attribute('outerHTML')
html_indice = elemento_indice.get_attribute('outerHTML')

tabela = pd.read_html(html_tabela) [0]
indice = pd.read_html(html_indice) [0]

driver.quit()
```

Assim é possível armazenar essa informação posteriormente dentro de um csv, para caso você queira transportar essas informações para o Excel.

Um ponto de atenção é que esses dados não vem estruturados. Nesta etapa, as suas tabelas **tabela** e **indice** estarão assim:

0	1	2	3	4	5	6	7	8
PREÇO ABERT.	PREÇO MÍN.	PREÇO MÁX.	PREÇO MÉD.	ÚLT. PREÇO	AJUSTE	VAR. PTOS.	ÚLT. OF. COMPRA	ÚLT. OF. VENDA
5.026,500	4.927,500	5.028,000	4.960,978	4.933,500	4.949,5210	70,0870-	4.930,500	4.933,000
5.056,000	4.962,000	5.057,000	4.987,794	4.962,000	4.980,1650	70,4410-	4.960,000	4.966,500
5.052,000	4.990,000	5.052,000	5.003,803	4.990,000	5.007,7270	70,9460-	4.984,000	4.998,500
0000	0000	0000	0000	0000	5.034,2340	71,3190-	4.990,000	0000
0000	0000	0000	0000	0000	5.065,3330	71,5960-	0000	0000
0000	0000	0000	0000	0000	5.088,7210	71,3980-	0000	0000
0000	0000	0000	0000	0000	5.137,0140	00000	0000	0000
0000	0000	0000	0000	0000	5.157,6170	70,6270-	0000	0000
0000	0000	0000	0000	0000	5.184,1230	70,9350-	0000	0000
0000	0000	0000	0000	0000	5.204,2320	69,6300-	0000	0000
5.348,500	5.348,500	5.348,500	5.348,500	5.348,500	5.376,6890	68,0600-	0000	0000
0000	0000	0000	0000	0000	5.456,2780	64,7000-	0000	0000

0

0 VENCTO

1 K23

2 M23

3 N23

4 Q23

5 U23

6 V23

7 Z23

8 F24

9 G24

10 H24

11 V24

12 F25

Isso significa que precisaremos organizar os nossos dados fazendo alguns ajustes. Primeiro, traremos apenas os dados da coluna “ajuste”, que é o preço de fechamento do contrato, usando o código a seguir:



```
tabela.columns = tabela.loc [0]  
tabela = tabela ['AJUSTE']
```

A tabela  
ficará assim:

	AJUSTE
0	4.949,5210
1	4.980,1650
2	5.007,7270
3	5.034,2340
4	5.065,3330
5	5.088,7210
6	5.137,0140
7	5.157,6170
8	5.184,1230
9	5.204,2320
10	5.376,6890
11	5.456,2780

Em seguida, vamos remover a primeira linha da tabela porque ela não contém os dados de fechamento, é apenas um cabeçalho e não deveria fazer parte dos nossos dados.



```
tabela = tabela.drop(0, axes = 0 )  
indice.columns = indice.loc[0]
```



Agora, vamos tratar a tabela **indice**. Nós precisamos trazer a palavra “VENCTO” para o cabeçalho da coluna da tabela . Para isso, usaremos o seguinte código:



```
indice_dolar = indice [ 'VENCTO' ]  
indice = indice.drop(0, axis = 0)
```



Agora, vamos unir as duas tabelas, trazendo índice para a primeira coluna da tabela:



```
tabela.index = indice ['VENCTO']
tabela = tabela[tabela != "0"]
```



Dessa forma, a nossa tabela ficará assim:

```
VENCTO
K23    4.949,5210
M23    4.980,1650
N23    5.007,7270
Q23    5.034,2340
U23    5.065,3330
V23    5.088,7210
Z23    5.137,0140
F24    5.157,6170
G24    5.184,1230
H24    5.204,2320
V24    5.376,6890
F25    5.456,2780
Name: AJUSTE, dtype: object
```

Para finalizar, vamos facilitar ainda mais a visualização desses dados, transformando o código correspondente ao vencimento de cada contrato na data de vencimento de fato. Para isso, usaremos o seguinte código:



O que esse código faz é basicamente criar uma legenda, associando cada letra a um vencimento.

Depois disso, a nossa tabela ficará assim finalmente:

```
2023-05-01      4.949,5210
2023-06-01      4.980,1650
2023-07-01      5.007,7270
2023-08-01      5.034,2340
2023-09-01      5.065,3330
2023-10-01      5.088,7210
2023-12-01      5.137,0140
2024-01-01      5.157,6170
2024-02-01      5.184,1230
2024-03-01      5.204,2320
2024-10-01      5.376,6890
2025-01-01      5.456,2780
Name: AJUSTE, dtype: object
```

Pronto. Acabamos de extrair os dados do contrato de mini dólar diretamente da Bolsa de Valores usando o Python.

É assim que nós conseguimos extrair dados de um site de forma simples usando web scraping. Lembrando que esse é apenas um exemplo de extração de dados. Você pode extrair basicamente qualquer dado de qualquer site dessa forma. Alguns serão mais fáceis, outros mais difíceis, mas a lógica é a mesma.

Esse é, sem dúvidas, um conhecimento que te diferencia da maioria dos investidores e profissionais do mercado financeiro. **Detalhe: isso é menos de 1% do que o Python pode fazer por você**, se você souber aplicá-lo do jeito certo no mercado financeiro.

Ele foi um grande divisor de águas na minha vida. **Se eu não soubesse Python seria muito mais difícil** me destacar no fundo de pensão onde eu trabalhei e ganhar a confiança para gerir milhões de reais com menos de 21 anos. Resumindo: eu não estaria aqui hoje falando com você e você não estaria lendo este livro.

Esse é um dos motivos que me fez dedicar a minha carreira a ensinar outras pessoas como aplicar o Python no mercado financeiro. É por isso que o Código.py, minha formação de Python aplicado a finanças, existe.

A formação está com as inscrições fechadas no momento, mas você pode entrar na lista de espera para ser informado quando abrirmos a próxima turma. Basta [clicar aqui neste link](#) ou na imagem abaixo para ser redirecionado e se inscrever na lista de espera.

# código.py

Crie **códigos em Python**, do absoluto zero, que **investem o seu dinheiro** enquanto você dorme.

[Entrar na lista de espera →](#)

**Brenno Sullivan**

Head de Dados da Varos

**Varos 2024**  
Todos os direitos reservados.