



# *INTENSIVÃO DE PYTHON* {#}

100% ONLINE & GRATUITO

## Apostila Completa Aula 3

Aprenda como automatizar processos que  
tenham interface com a internet  
Impressionador do absoluto zero!



Parte 1

# Introdução

# O que vamos aprender

Na quarta aula da Semana do Python, você vai aprender a criar um código para automação de processos. No dia a dia das empresas, é muito comum que existam operações manuais que além de extremamente repetitivas (chatas) são suscetíveis a erro visto que são feitas manualmente. Vamos aprender como criar um código com o qual você possa resolver esse problema sem nem tocar no mouse 😊. Aprenda como **fazer uma automação com integração web** com os conceitos abaixo:

Importando bases  
de dados do Excel

Jupyter Notebook

Importando  
bibliotecas

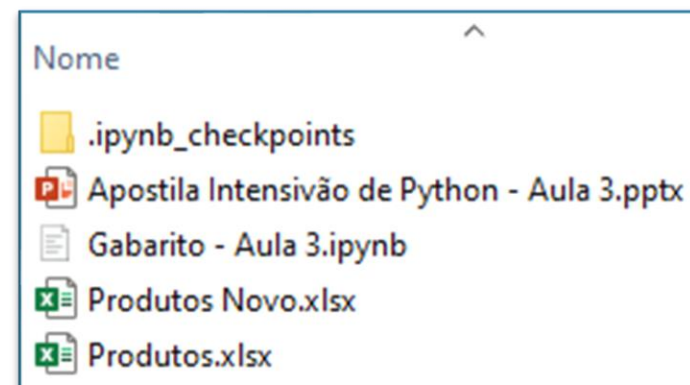
Webdriver

Usando Selenium

Após todos esse aprendizado, seremos capazes de transformar um processo extremamente repetitivo

... em processo automático e sem erros! Tudo graças a você! 😊

	A	B	C	D	E	F	G
1	Produtos	Preço Original	Moeda	Cotação	Preço de Compra	Margem	Preço de Venda
2	Câmera Canon	999,99	Dólar	5	4999,95	1,4	6999,93
3	Carro Renault	4500	Euro	6	27000	2	54000
4	Notebook Dell	899,99	Dólar	5	4499,95	1,7	7649,915
5	iPhone	799	Dólar	5	3995	1,7	6791,5
6	Carro Fiat	3000	Euro	6	18000	1,9	34200
7	Celular Xiaomi	480,48	Dólar	5	2402,4	2	4804,8
8	Joia 20g	20	Ouro	350	7000	1,15	8050



# Introdução

## Entendendo o problema

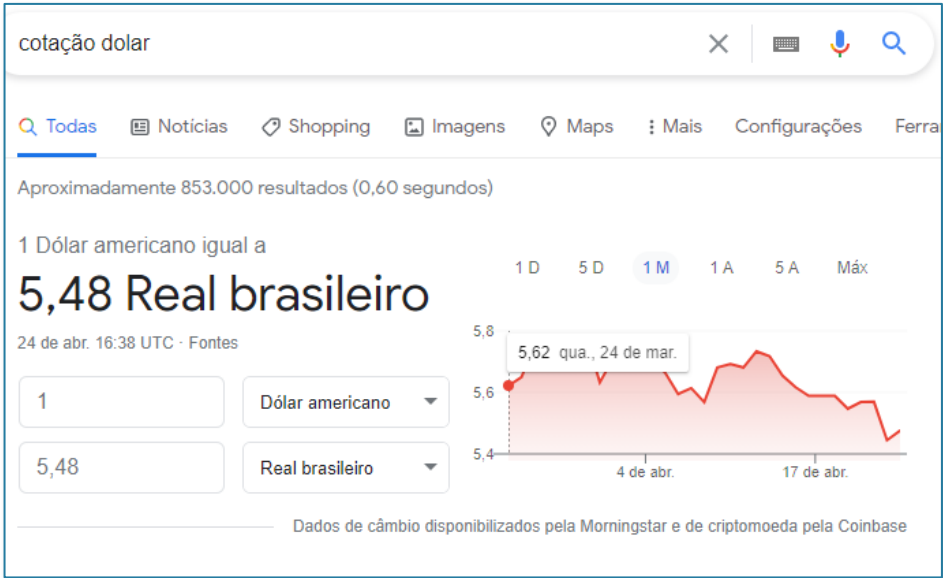
Na nossa empresa fictícia comercializamos produtos que são diretamente influenciados por moedas estrangeiras (Dólar, Euro) e ouro.

Nosso trabalho como analista de vendas é atualizar a cotação das moedas diariamente para termos no novo preço de venda.

Apesar de simples, é uma tarefa chata e que não agrega muito valor.

Para essa atualização precisamos acessar um site de cotações, pegar o valor da cotação e atualizar nossa planilha.

	A	B	C	D	E	F	G
1	Produtos	Preço Original	Moeda	Cotação	Preço de Compra	Margem	Preço de Venda
2	Câmera Canon	999,99	Dólar	5	4999,95	1,4	6999,93
3	Carro Renault	4500	Euro	6	27000	2	54000
4	Notebook Dell	899,99	Dólar	5	4499,95	1,7	7649,915
5	IPhone	799	Dólar	5	3995	1,7	6791,5
6	Carro Fiat	3000	Euro	6	18000	1,9	34200
7	Celular Xiaomi	480,48	Dólar	5	2402,4	2	4804,8
8	Joia 20g	20	Ouro	350	7000	1,15	8050



Website "melhorcâmbio.com". The header shows the logo "melhorcâmbio.com". Below the logo, there is a green banner with the text "Encontre a casa de câmbio com a melhor cotação e negocie para pagar ainda menos!". Below the banner, there are two input fields: "MOEDA..." with a dollar sign icon and "CIDADE..." with a location pin icon. Below these fields, there are two buttons: "Quero comprar" (green) and "Quero vender" (blue). Below the buttons, there is a link "Quero fazer uma remessa online". At the bottom, it says "ACESSO RÁPIDO".

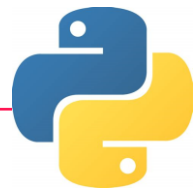


# Entendendo a solução final

Nossa solução final será:

- 1) Pesquisar as cotações das moedas e ouro;
- 2) Buscar a informação no site;
- 3) Armazenar as informações;
- 4) Preencher a cotação em uma planilha excel.

**Tudo isso automaticamente!!** Apenas rodando o código que vamos criar.



	A	B	C	D	E	F	G
1	Produtos	Preço Original	Moeda	Cotação	Preço de Compra	Margem	Preço de Venda
2	Câmera Canon	999,99	Dólar	5,48	5476,359236	1,4	7666,90293
3	Carro Renault	4500	Euro	6,62	29812,02608	2	59624,05217
4	Notebook Dell	899,99	Dólar	5,48	4928,717836	1,7	8378,820321
5	iPhone	799	Dólar	5,48	4375,654786	1,7	7438,613136
6	Carro Fiat	3000	Euro	6,62	19874,68406	1,9	37761,8997
7	Celular Xiaomi	480,48	Dólar	5,48	2631,307399	2	5262,614797
8	Joia 20g	20	Ouro	312,81	6256,2	1,15	7194,63



Parte 2

# Importando o Selenium

# Importando o Selenium

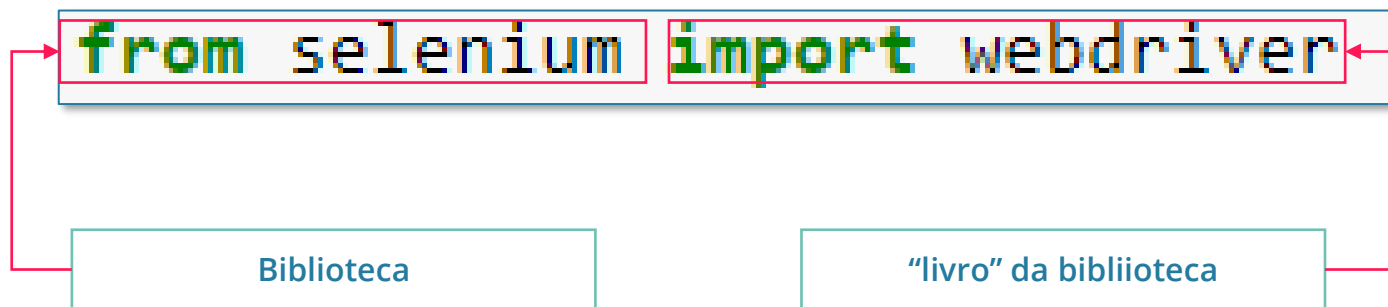
Como vimos na aula 1 da Semana do Python, vamos usar bibliotecas que nos facilitem importar dados de planilhas Excel, arquivos .csv, etc.

No entanto, além do pandas iremos importar o **selenium**.

Antes de entendermos no detalhe o que elas fazem e para que servem, vamos nos atentar a uma diferença na hora da importação.

Podemos perceber que na primeira linha importamos o selenium utilizando a estrutura **from** antes do import.

Essa estrutura significa dizer “dentro da biblioteca selenium importe o livro webdriver”.



# Importando o webdriver

Como assim, um livro de uma biblioteca?

Antes de entendermos o código, imagine uma biblioteca de fato.

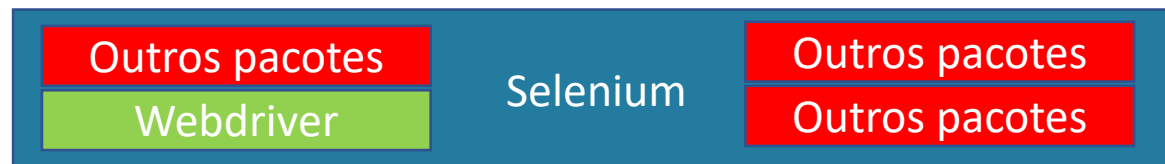
Grande, correto? Muitas das informações ali, são úteis mas não naquele momento...

Possivelmente só vamos conseguir ler 1, 2 ou 3 livros de uma vez. Não faz sentido alugar toda a biblioteca, apenas o que vamos precisar.

Aqui é exatamente a mesma coisa! Não temos que importar toda uma biblioteca se apenas uma parte dessa biblioteca nos interessa.

No nosso caso, o que nos interessa é o **webdriver**.

Isso torna nosso código mais simples e rápido!





# Instalando o Selenium

Para essa aula estamos usando o Jupyter e nele já possuímos uma série de pacotes “pré-instalados” mas as vezes precisamos instalar pacotes adicionais.

O Python, possui um “instalador embutido” que se chama **pip**.

Por ele, é possível, instalar e desinstalar pacotes.

Caso você queira saber se o selenium está instalado basta usar o comando abaixo em uma das células do Jupyter:

**pip freeze**

Caso não encontre na lista o selenium, use o comando abaixo para instalar:

**pip install -U selenium**

```
pip freeze
argh==0.26.2
asn1crypto==1.3.0
astroid==2.4.2
astropy==4.0.1.post1
atomicwrites==1.4.0
attrs==19.3.0
autopep8 @ file:///tmp/build/80754af9/autopep8_1592412889138/work
Babel==2.8.0
backcall==0.2.0
backports.functools-lru-cache==1.6.1
scipy @ file:///C:/ci/scipy_1592916963468/work
seaborn==0.10.1
selenium==3.141.0
Send2Trash==1.5.0
simplegeneric==0.8.1
singledispatch==3.4.0.3
sip==4.19.13
six==1.15.0
snowballstemmer==2.0.0
sortedcollections==1.2.1
sortedcontainers==2.2.2
soupsieve==2.0.1
Sphinx @ file:///tmp/build/80754af9/sphinx_1594223420021/work
```

Comando pip freeze

Lista de pacotes instalados no Python

Parte 3

# Interface com uma página na web

# Interface com uma página na web

# Selenium

Conforme explicamos anteriormente o **selenium** é uma biblioteca assim como o pandas.

Essa biblioteca é muito utilizada como interface com o navegador. Ela funciona como um robô que clica, insere dados, etc em páginas WEB: interagindo com os sites como se fosse você.

Muito útil para processos repetitivos como este que temos aqui.

Aqui temos links de documentação para aqueles que gostariam de se aprofundar no tema e funcionalidades:

<https://selenium-python.readthedocs.io/>

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

<https://www.selenium.dev/documentation/en/>



# Selenium – webdriver (1/3)

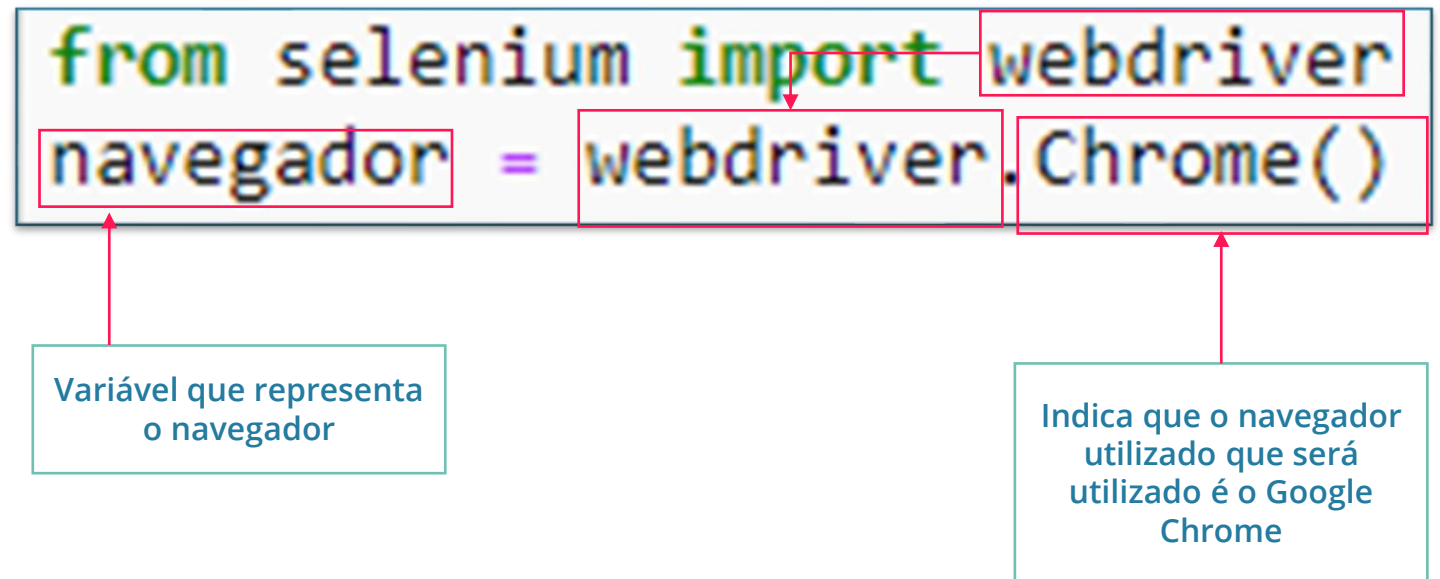
Vamos para nosso código. Devemos criar um código que nos permita acessar o site, e buscar os dados da nossa base.

Se você se lembra do nosso passo anterior, vai perceber que estamos usando o webdriver que importamos anteriormente.

O webdriver possui uma particularidade. Ele precisa ser baixado, extraído e colocado dentro da **MESMA pasta** que o Python possui seu executável. O próximo slide será só sobre isso, então não se preocupe.

Voltando para nosso código, podemos perceber que criamos uma variável **navegador**. Ela nos permitirá interagir com os diferentes elementos que encontraremos nas páginas WEB.

Outro ponto é que vemos **.Chrome()**. Isso nos indica que acessaremos essas páginas via Google Chrome.



# Interface com uma página na web

## Selenium – webdriver (2/3)

Antes de mais nada precisamos baixar o ChromeDriver, e o ChromeDriver precisa ser da mesma versão que o Chrome instalado no seu computador.

**Passo 1:** Abra o Chrome e acesse o endereço `chrome://settings/help`

**Passo 2:** Veja a versão do seu Google Chrome. Só os primeiros dígitos antes do primeiro ponto importam. Então, já que o navegador do exemplo é versão 97.0.4692.71, vamos baixar qualquer ChromeDriver que seja versão 97.

**Passo 3:** Acesse o [link](#) para baixar o driver

**Passo 4:** Escolha a versão mais próxima da sua versão, todas elas servem, mas versões mais próximas apresentam maior compatibilidade.

**Passo 5:** Escolha seu sistema operacional. No nosso caso Windows. Não existe chromedriver 64bits para Windows, então usaremos o win32.

Google Chrome

O Chrome está atualizado  
Versão 97.0.4692.71 (Versão oficial) 64 bits

Ajuda com o Chrome

Informar um problema

**ChromeDriver 97.0.4692.71**

Supports Chrome version 97

- Resolved issue 3878: Better handling of...

For more details, please see the [release](#)

**ChromeDriver 97.0.4692.36**

Supports Chrome version 97

- Resolved issue 3878: Better handling of...

For more details, please see the [release](#)

**ChromeDriver 97.0.4692.20**

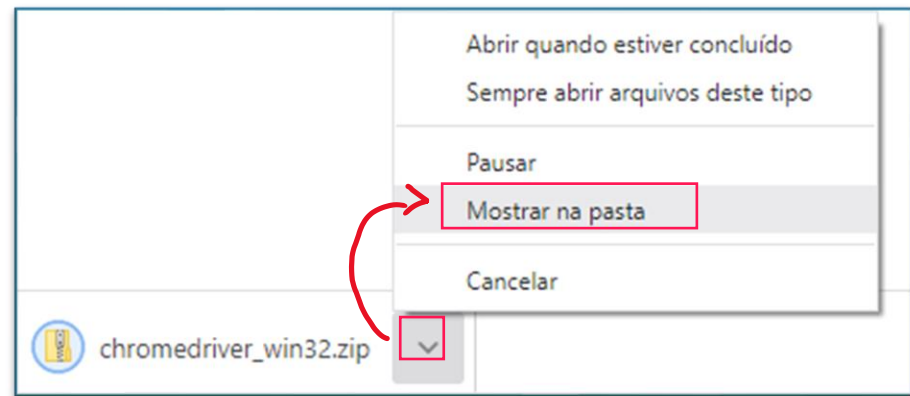
Parent Directory

<a href="#">chromedriver_linux64.zip</a>	2022-01-05 05:45:10	9.52MB
<a href="#">chromedriver_mac64.zip</a>	2022-01-05 05:45:13	7.89MB
<a href="#">chromedriver_mac64_m1.zip</a>	2022-01-05 05:45:15	7.48MB
<a href="#">chromedriver_win32.zip</a>	2022-01-05 05:45:17	5.89MB
<a href="#">notes.txt</a>	2022-01-05 05:45:22	0.00MB

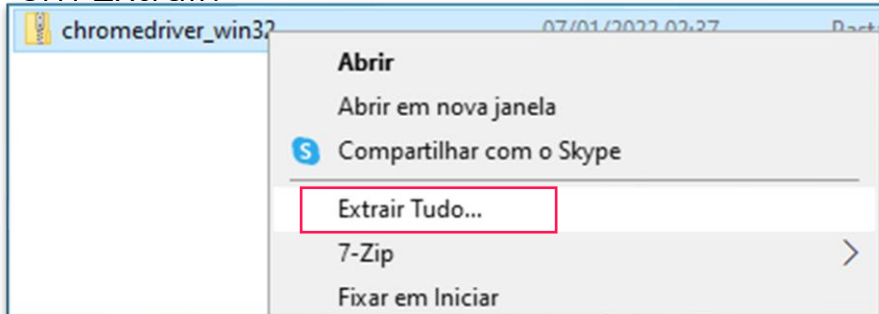
# Interface com uma página na web

## Selenium – webdriver (3/3)

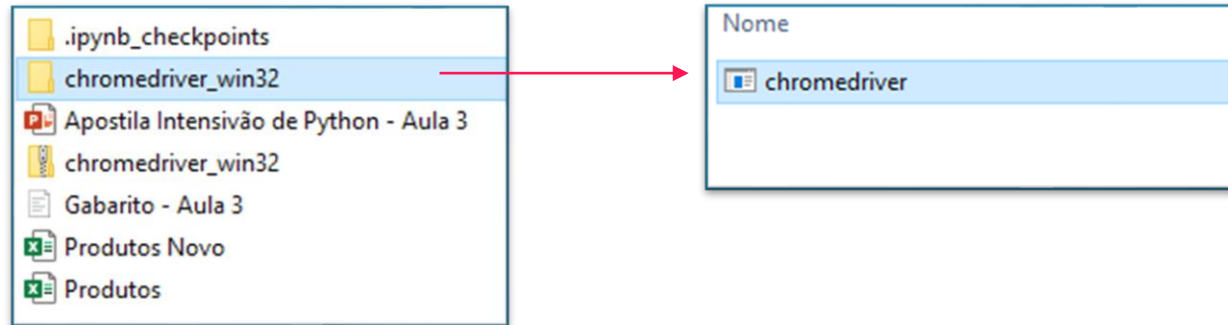
**Passo 6:** Abra o local do arquivo baixado.



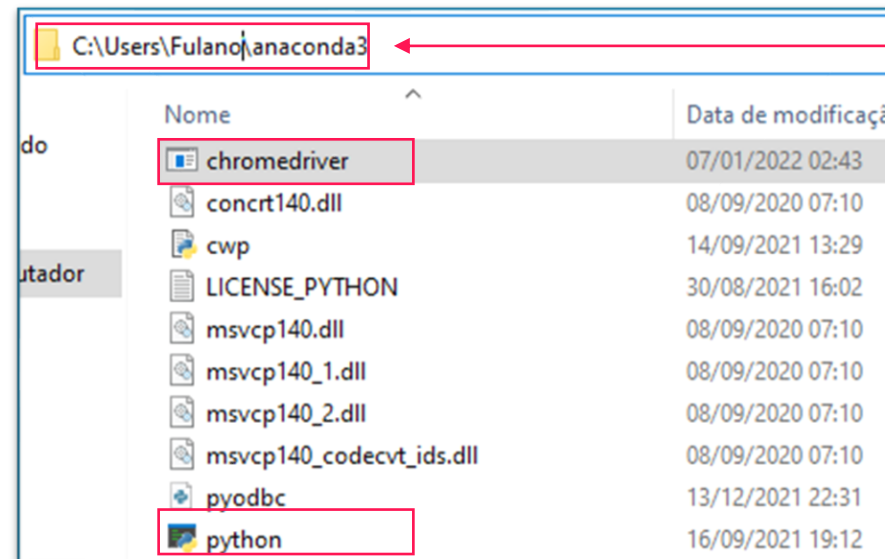
**Passo 7:** Clique com o botão direito no arquivo baixado e em seguida clique em Extrair Tudo... Abrirá uma nova janela. Clique em Extrair.



**Passo 8:** Será criada uma pasta que contém o arquivo chromedriver.exe



**Passo 9:** Coloque o arquivo **chromedriver.exe** na mesma pasta que o arquivo **python.exe**



Pasta que contém o arquivo python.exe. ATENÇÃO!!! Este caminho pode variar!!!



Parte 4

# Buscando as cotações na Web

# Buscando as cotações na Web

## Cotação do Dólar(1/8)

Agora que já importamos nossas bibliotecas, vamos começar a utilizar o Selenium para acessarmos os sites que nos fornecerão as cotações que precisamos.

Essencialmente o que faremos será escrever códigos que reproduzam o que nós mesmos faríamos se estivéssemos realizando essa tarefa manualmente. Ou seja:

- 1) Entrar no site que estamos buscando (<https://google.com.br>);
- 2) No campo de pesquisa, digitar “cotação dólar”;
- 3) Apertar ENTER para Pesquisar;
- 4) Pegar a cotação indicada.

The image displays four sequential screenshots of a web browser, each with a red number in the top right corner indicating a step in the process:

- Step 1:** The address bar shows the URL `https://google.com.br|`. A dropdown menu is visible below it, listing several suggestions, including `https://google.com.br`.
- Step 2:** The Google homepage is shown. The search bar contains the text `cotação dólar`.
- Step 3:** The same search bar is shown, but with a black arrow pointing to the search button and the word **ENTER** below it, indicating the action to be taken.
- Step 4:** The search results page is displayed. The search bar still contains `cotação dólar`. The results show approximately 853,000 results. The first result is a snippet from `span.DFIfde.SwHCTb` showing the exchange rate **5,48 Real brasileiro** for 1 Dólar americano. Below this, there are two input fields: one with the value `1` and a dropdown menu set to `Dólar americano`, and another with the value `5,48` and a dropdown menu set to `Real brasileiro`.

# Buscando as cotações na Web

## Cotação do Dólar (2/8)

Como vimos, a primeira etapa é entrar no site que vamos buscar as cotações.

Essa será nossa primeira linha de código.

### 1) Entrar no site que estamos buscando

Nosso código começará com duas funções:

- 1) Abrir no novo navegador do Google Chrome;
- 2) Buscar o site indicado.

Como podemos ver, para realizar essa tarefa, vamos utilizar o método **.get(<site>)**.

Perceba que a nova página possui um cabeçalho indicando que está sendo rodada por uma automação.

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

Vamos para o passo de número 2!

```
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.by import By
```

```
navegador = webdriver.Chrome()
navegador.get("https://www.google.com/")
```

Método que realiza o  
acesso à página

Site que desejamos  
entrar

Variável que interage com o navegador

🌐 https://google.com.br|

🌐 https://google.com.br

# Buscando as cotações na Web

## Cotação do Dólar (3/8)

A segunda etapa é selecionar a caixa de busca e digitar as informações que desejamos buscar. No nosso caso, “cotação dólar”

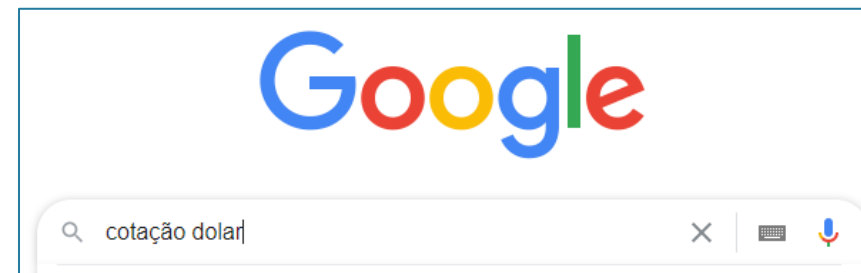
- 1) ~~Entrar no site que estamos buscando (<https://google.com.br>);~~
- 2) No campo de pesquisa, digitar “cotação dólar”;

Mas como? Vamos fazer uma pausa e entender como sites são construídos. Em geral, o que vemos ao olharmos para um site se trata de um código HTML.

Sem entrar muitos detalhes, o que significa isso?

Entenda que cada coisa que existe na página possui um código que “marca sua posição” na página.

Ou seja, se encontrarmos esse código, acharemos o campo de pesquisa. Para acharmos esse campo, existem diversas formas a mais simples e que exigirá ZERO conhecimento de HTML é usando o **Inspecionar**. Segue o passo a passo ☺ :



1

A screenshot of a right-click context menu. The 'Inspecionar' option at the bottom is highlighted with a mouse cursor. Other options like 'Recortar', 'Copiar', and 'Colar' are also visible.

Com o botão direito do mouse abra as opções e clique em Inspecionar;

2

A screenshot of the Chrome Developer Tools 'Elements' panel. A red box highlights the '<input type="text"' element in the HTML tree. A mouse cursor is pointing at the highlighted element.

Clique no botão indicado abaixo;

3

A screenshot of the Chrome Developer Tools 'Properties' panel. It shows details for the selected input field, including its name 'Pesquisar' and role 'combobox'. A mouse cursor is pointing at the input field in the background.

Clique no campo;

Continua

# Buscando as cotações na Web

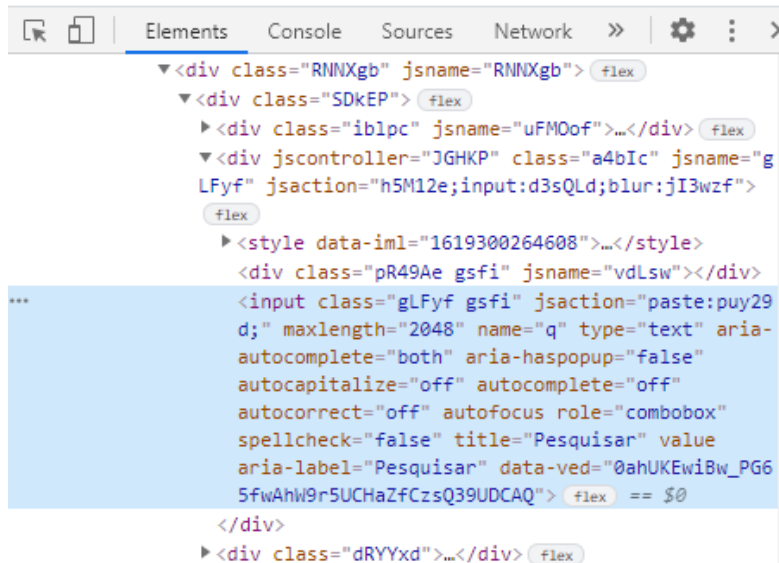
## Cotação do Dólar (4/8)

A segunda etapa é selecionar a caixa de busca e digitar as informações que desejamos buscar. No nosso caso, “cotação dólar”

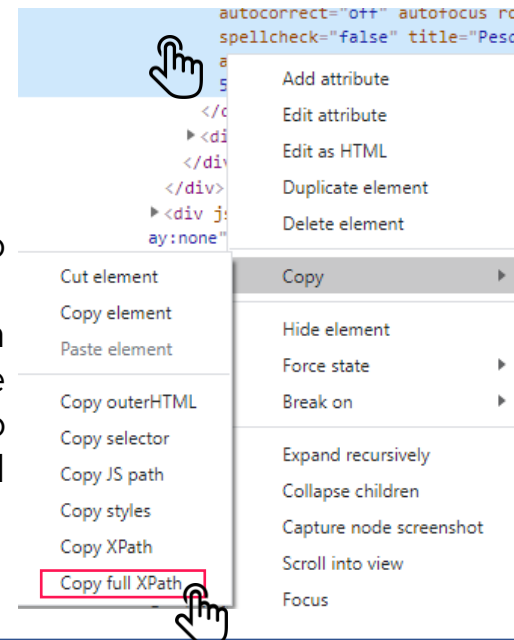
- 1) Entrar no site que estamos buscando (<https://google.com.br>);
- 2) No campo de pesquisa, digitar “cotação dólar”;



4 Linha de código que indica o campo selecionado



5 Novamente com o botão direito selecione a opção “Copy” e depois a opção Copy full XPath



Pronto! Agora temos o código que indica onde fica nosso campo de pesquisa do google.

**Ele está armazenado no CTRL+C do seu PC.** Basta agora usarmos agora no nosso código 😊

## Buscando as cotações na Web

# Cotação do Dólar (5/8)

Acabamos de clicar no campo **XPATH**. Mas o que é o XPATH? Como falamos anteriormente é a posição do site. Vamos dar uma olhada nesse código que ficou armazenado no CTRL+C do nosso computador:

```
/html/body/div[1]/div[3]/form/div[1]/div[1]/div[1]/div/div[2]/input
```

Agora que temos o caminho, precisamos explicar para o Selenium que ele irá buscar pelo XPATH e não por outro método. Portanto, o método que usaremos será o:

```
.find_element(By.XPATH, '/html/body/div[1]/div[3]/form/div[1]/div[1]/div[1]/div/div[2]/input')
```

```
# Passo 1: Pegar a cotação do Dólar  
navegador.find_element(By.XPATH, '/html/body/div[1]/div[3]/form/div[1]/div[1]/div[1]/div/div[2]/input')
```

Utilizando a nossa variável **navegador** e o **método de busca no caminho indicado**, conseguimos acessar o campo de busca.

No entanto, falta uma etapa nesse processo. Digitar **“cotação do dólar”**. Para isso usaremos outro método:

```
.send_keys()
```

```
# Passo 1: Pegar a cotação do Dólar  
navegador.find_element(By.XPATH,  
    '/html/body/div[1]/div[3]/form/div[1]/div[1]/div[1]/div/div[2]/input').send_keys("cotação dólar")
```

Esse método nos permitirá **após a seleção do campo de pesquisa** digitar “cotação dólar”.

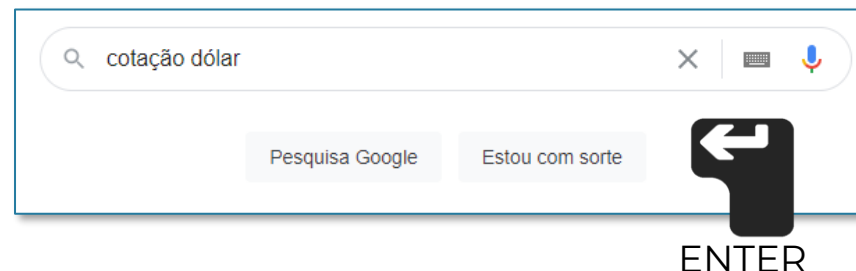


# Buscando as cotações na Web

## Cotação do Dólar (6/8)

Para a terceira etapa usaremos o mesmo conceito utilizado anteriormente, mas mudando um pouco nosso código anterior.

- 1) ~~Entrar no site que estamos buscando (<https://google.com.br>);~~
- 2) ~~No campo de pesquisa, digitar “cotação dólar”;~~
- 3) Apertar ENTER para Pesquisar;



```
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.by import By

navegador = webdriver.Chrome()
navegador.get("https://www.google.com/")
# Passo 1: Pegar a cotação do Dólar
navegador.find_element(By.XPATH, '/html/body/div[1]/div[3]/form/div[1]/div[1]/div[1]/div/div[2]/input').send_keys("cotação dólar")
navegador.find_element(By.XPATH, '/html/body/div[1]/div[3]/form/div[1]/div[1]/div[1]/div/div[2]/input').send_keys(Keys.ENTER)
```

Local permanece o mesmo,  
então nenhuma mudança no  
código até aqui

Para textos, o `.send_keys`, nos permite enviar todo o texto com a utilização das aspas `" "`. Para o uso de uma tecla simulando que uma pessoa está pressionando-a no teclado, precisamos adicionar o complemento `Keys`, e após isso a tecla desejada. Nesse caso `ENTER`.

# Buscando as cotações na Web

## Cotação do Dólar (7/8)

Para a quarta etapa usaremos o mesmo conceito utilizado anteriormente na localização do campo de busca do Google.

- 1) Entrar no site que estamos buscando (<https://google.com.br>);
- 2) No campo de pesquisa, digitar “cotação dólar”;
- 3) Apertar ENTER para Pesquisar;
- 4) Pegar a cotação indicada.

Após o carregamento da página, usaremos o **INSPECIONAR** novamente para localizarmos o valor indicado na imagem ao lado.

Com o código **XPATH** localizados vamos usar o mesmo método usado anterior.

```
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.by import By
```

```
navegador = webdriver.Chrome()
navegador.get("https://www.google.com/")
```

*# Passo 1: Pegar a cotação do Dólar*

```
navegador.find_element(By.XPATH,
    '/html/body/div[1]/div[3]/form/div[1]/div[1]/div[1]/div/div[2]/input').send_keys("cotação dólar")
navegador.find_element(By.XPATH,
    '/html/body/div[1]/div[3]/form/div[1]/div[1]/div[1]/div/div[2]/input').send_keys(Keys.ENTER)
cotacao_dolar = navegador.find_element(By.XPATH,
    '//*[@id="knowledge-currency__updatable-data-column"]/div[1]/div[2]/span[1]').get_attribute("data-value")
print(cotacao_dolar)
```

```
<div class="dDoNo ikb48b vk_bk gsrt gzfeS">
  <span class="DFlfde SwHCTb" data-precision="2" data-value="5.476414">5,48</span> == $0
  <span class="MWvIVe" data-mid="/m/03385m" data-name="Real brasileiro">Real brasileiro</span>
</div>
```



Usamos o parâmetro **'data-value'** pois como podemos ver no código HTML, esse é o atributo que recebe o valor da cotação.

Podemos ver que temos duas diferenças importantes:

- 1) Criação de uma variável `cotacao_dolar` que armazenará o valor coletado pelo Selenium.
- 2) Método `get_attribute('data-value')` que nos permite agora ao invés de dar informações, coletar.

## Buscando as cotações na Web

# Cotação do Dólar (8/8)

Pronto! Conseguimos nossa cotação do dólar.

Podemos usar o `print(cotacao_dolar)` para exibirmos o resultado e garantir que tudo funcionou☺.

Perceba que o valor printado, não é 5,48 como exibido e sim 5,476414 que é o valor armazenado na variável `data-value` apresentada no slide anterior.

Vamos agora para a cotação do EURO!

```
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.by import By

navegador = webdriver.Chrome()
navegador.get("https://www.google.com/")

# Passo 1: Pegar a cotação do Dólar
navegador.find_element(By.XPATH,
    '/html/body/div[1]/div[3]/form/div[1]/div[1]
navegador.find_element(By.XPATH,
    '/html/body/div[1]/div[3]/form/div[1]/div[1]
cotacao_dolar = navegador.find_element(By.XPATH,
    '//*[@id="knowledge-currency__updatable-data
print(cotacao_dolar)
```

5.476414

# Buscando as cotações na Web

## Cotação do Euro

Talvez, você tenha pensado:

“ Só trocar dólar por EURO. Não?!”

Você tem total razão. Só precisamos fazer uma checagem antes:

A página para cotação do EURO do Google é IGUAL a de dólar?

Vamos checar...

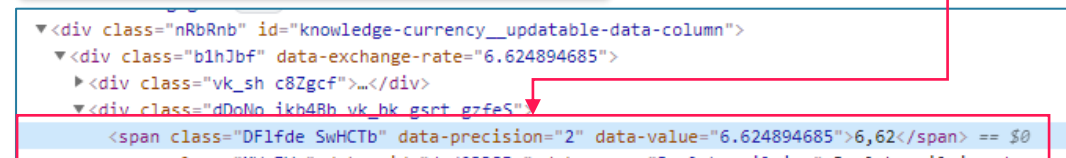
Conforme podemos ver nas imagens ao lado, a estrutura é exatamente a mesma e a variável **data-value** permanece a mesma.

Perfeito, então podemos copiar o código do dólar e apenas avaliar o que é particular do dólar e substituir por euro.

```
# Passo 2: Pegar a cotação do Euro
navegador.get("https://www.google.com/")
navegador.find_element(By.XPATH,
    '/html/body/div[1]/div[3]/form/div[1]/div[1]/div[1]/div/div[2]/input').send_keys("cotação euro")
navegador.find_element(By.XPATH,
    '/html/body/div[1]/div[3]/form/div[1]/div[1]/div[1]/div/div[2]/input').send_keys(Keys.ENTER)
cotacao_euro = navegador.find_element(By.XPATH,
    '//*[@id="knowledge-currency__updatable-data-column"]/div[1]/div[2]/span[1]').get_attribute("data-value")
print(cotacao_euro)
```



Mesma estrutura da página de cotação do dólar. Logo, poderemos usar essencialmente o mesmo código usado anteriormente.



Substituição do termo dólar por euro

Criação de uma nova variável cotacao\_euro

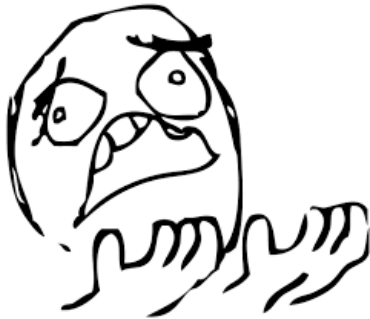
# Buscando as cotações na Web

## Cotação do Ouro (1/5)

O Euro foi moleza.. Ouro é só fazer a mesma coisa...

Vamos checar!!!!

É ..... diferente !!!! ☹ ☹ ☹ ☹ ☹



MAS... Sou impressionador ☺



**CHALLENGE ACCEPTED**

cotação ouro

Todas Notícias Shopping Imagens Vídeos Mais Configurações Ferramentas


Aproximadamente 3.720.000 resultados (0,40 segundos)

**Anúncio** · <https://www.btgpactualdigital.com/> ▾

**Cotação do Ouro - Invista em Ouro**

Proteja seu patrimônio. Reduza a exposição à volatilidade do mercado investindo em **ouro**.  
Proteja sua carteira diversificando investimentos com **Ouro**. Comece com apenas 500 reais.

O valor do grama do **ouro** hoje está **cotado** em R\$ 312,81.



<https://www.melhorcambio.com> > ouro-hoje

**Ouro Hoje: Cotação atualizada do grama do ouro. Valor e ...**

🔍 Sobre trechos em destaque • 🗉 Feedback

As pessoas também perguntam



# Buscando as cotações na Web

## Cotação do Ouro (2/5)

Memes a parte, podemos ver que nossa página é levemente diferente das páginas anteriores.

Aqui, temos duas opções:

- 1) Criar um código para a página de pesquisa;
- 2) Buscarmos outra página que vá nos oferecer essa informação.

Como o objetivo aqui é aprender, vamos com a segunda opção ☺

<https://www.melhorcambio.com/>

Antes de irmos para o código, vamos entender os caminhos que precisamos seguir:

- 1) Acessar o site: <https://www.melhorcambio.com/>;
- 2) Localizar o ícone que representa a cotação do ouro;
- 3) Mudar de aba do navegador para a nova aba aberta;
- 4) Coleta a informação da cotação.

The image displays four sequential screenshots of the **melhorcambio.com** website, illustrating the steps to find the gold price. Each screenshot is marked with a red box and a number:

- 1**: The homepage of **melhorcambio.com**. It features the company logo, a tagline "Encontre a casa de câmbio com a melhor cotação e negocie para pagar ainda menos!", and search filters for "MOEDA..." and "CIDADE...".
- 2**: A navigation menu with icons for "Ouro Hoje" (highlighted with a red box) and "Petróleo Hoje".
- 3**: A browser window showing the "Ouro Hoje" page. A red arrow indicates the transition from the menu to this page. The page shows a search bar for "Ouro (g)" with the value "1,00".
- 4**: A detailed view of the "Ouro Hoje" page. It shows the "VALOR DO GRAMA DO OURO" as "R\$ 312,81", which is highlighted with a red box.



## Buscando as cotações na Web

# Cotação do Ouro (3/5)

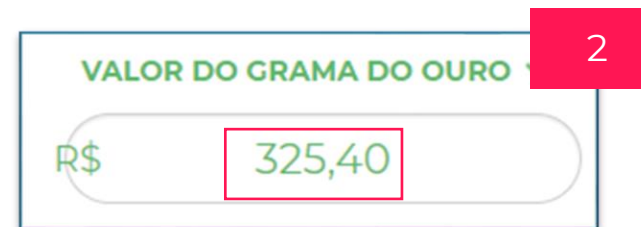
Poderíamos seguir o caminho proposto no slide anterior: acessar o site, clicar no ícone, mudar de aba e coletar a informação. O Selenium é plenamente capaz de executar isso. MAAAAS... Somos impressionadores! Sejam os espertos! Para que executar uma sequência de passos grande se podemos encurtar o caminho?

Se repararmos bem, sempre que clicamos no ícone que leva à página que possui a cotação do ouro, o link nos leva para a mesma URL: <https://www.melhorcambio.com/ouro-hoje>. Por que então acessar o site, clicar no ícone, mudar de aba e resgatar o dado se podemos acessar diretamente a página que possui o dado e recuperá-lo?

Então é isso o que faremos!

**Passo 1:** Acessar o site <https://www.melhorcambio.com/ouro-hoje>

**Passo 2:** Coletar a cotação



# Buscando as cotações na Web

## Cotação do Ouro (4/5)

**Passo 1:** Acessar o site <https://www.melhorcambio.com/ouro-hoje>

```
# Passo 3: Pegar a cotação do Ouro
navegador.get("https://www.melhorcambio.com/ouro-hoje")
```

Essa etapa é feita da mesma forma que fizemos para o Dólar e Euro. Utilizaremos o método **.get(<site>)**

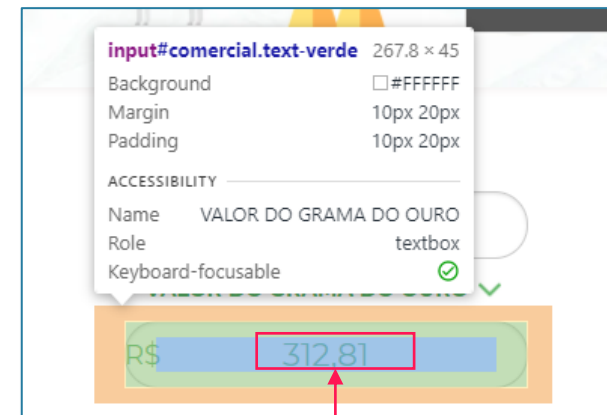
**Passo 2:** Coletar a informação da cotação

Coletaremos a informação da mesma forma que fizemos antes. Vamos acessar o elemento através do XPATH e utilizaremos o método **.get\_attribute(<nome-do-atributo>)** para acessar seu valor.

Um ponto em que precisamos prestar atenção é que o Python se utiliza de **“.”** como divisor das casas decimais. Ou seja, ao coletarmos a informação no modelo brasileiro com vírgula ao invés de ponto precisamos alterá-la para como o Python enxergará essa informação.

```
# Passo 3: Pegar a cotação do Ouro
navegador.get("https://www.melhorcambio.com/ouro-hoje")
cotacao_ouro = navegador.find_element(By.XPATH, '//*[@id="comercial"]').get_attribute("value")
cotacao_ouro = cotacao_ouro.replace(",", ".")
print(cotacao_ouro)
```

Transforma  
312,81 em 312.81



Usa o  
atributo  
value para  
coletar o  
valor

```
<label for="comercial">...</label>
<span id="sigla_moeda_verde" style="top:20px;left:-47px;">R$</span>
<input type="text" value="312,81" class="text-verde" id="comercial" calc="sim" == $0
</div>
</div>
```

# Buscando as cotações na Web

## Cotação do Ouro (5/5)

Agora que temos todas as cotações, Podemos fechar nosso navegador e partirmos para o uso do **PANDAS** que nos permitirá interagir com nosso arquivo EXCEL.

```
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.by import By

navegador.get("https://www.google.com/")

# Passo 1: Pegar a cotação do Dólar
navegador.find_element(By.XPATH,
    '/html/body/div[1]/div[3]/form/div[1]/div[1]/div[1]/div/div[2]/input').send_keys("cotação dólar")
navegador.find_element(By.XPATH,
    '/html/body/div[1]/div[3]/form/div[1]/div[1]/div[1]/div/div[2]/input').send_keys(Keys.ENTER)
cotacao_dolar = navegador.find_element(By.XPATH,
    '//*[@id="knowledge-currency__updatable-data-column"]/div[1]/div[2]/span[1]').get_attribute("data-value")
print(cotacao_dolar)

# Passo 2: Pegar a cotação do Euro
navegador.get("https://www.google.com/")
navegador.find_element(By.XPATH,
    '/html/body/div[1]/div[3]/form/div[1]/div[1]/div[1]/div/div[2]/input').send_keys("cotação euro")
navegador.find_element(By.XPATH,
    '/html/body/div[1]/div[3]/form/div[1]/div[1]/div[1]/div/div[2]/input').send_keys(Keys.ENTER)
cotacao_euro = navegador.find_element(By.XPATH,
    '//*[@id="knowledge-currency__updatable-data-column"]/div[1]/div[2]/span[1]').get_attribute("data-value")
print(cotacao_euro)

# Passo 3: Pegar a cotação do Ouro
navegador.get("https://www.melhorcambio.com/ouro-hoje")
cotacao_ouro = navegador.find_element(By.XPATH, '//*[@id="comercial"]').get_attribute("value")
cotacao_ouro = cotacao_ouro.replace(",",".")
print(cotacao_ouro)

navegador.quit()
```

Fecha o navegador

5.476414  
6.624894685  
312.81

Print das variáveis cotacao\_dolar, cotacao\_euro e cotacao\_ouro, respectivamente

Parte 5

# Importando a planilha Excel

# Importando a planilha via Pandas

Assim como vimos nas demais aulas da Semana do Python, usaremos o PANDAS para a importação da nossa base Excel.

O código ao lado nos permite realizar essa importação.

	A	B	C	D	E	F	G
1	Produtos	Preço Original	Moeda	Cotação	Preço de Compra	Margem	Preço de Venda
2	Câmera Canon	999,99	Dólar	5	4999,95	1,4	6999,93
3	Carro Renault	4500	Euro	6	27000	2	54000
4	Notebook Dell	899,99	Dólar	5	4499,95	1,7	7649,915
5	iPhone	799	Dólar	5	3995	1,7	6791,5
6	Carro Fiat	3000	Euro	6	18000	1,9	34200
7	Celular Xiaomi	480,48	Dólar	5	2402,4	2	4804,8
8	Joia 20g	20	Ouro	350	7000	1,15	8050



```
# Passo 4: Importar a lista de produtos
import pandas as pd

tabela = pd.read_excel("Produtos.xlsx")
display(tabela)
```

	Produtos	Preço Original	Moeda	Cotação	Preço de Compra	Margem	Preço de Venda
0	Câmera Canon	999.99	Dólar	5	4999.95	1.40	6999.930
1	Carro Renault	4500.00	Euro	6	27000.00	2.00	54000.000
2	Notebook Dell	899.99	Dólar	5	4499.95	1.70	7649.915
3	iPhone	799.00	Dólar	5	3995.00	1.70	6791.500
4	Carro Fiat	3000.00	Euro	6	18000.00	1.90	34200.000
5	Celular Xiaomi	480.48	Dólar	5	2402.40	2.00	4804.800
6	Joia 20g	20.00	Ouro	350	7000.00	1.15	8050.000

Parte 6

# Atualizando a planilha Excel



# Atualizando os dados da planilha com as cotações (1/2)

Aqui, usaremos as 3 variáveis criadas anteriormente para armazenar os valores das cotações.

Usando novamente o pandas vamos atualizar o valor do nosso dataframe **tabela**.

Para isso, usaremos o método **loc**, que nos auxilia a acessar o valor de uma “célula” do dataframe específica.

Nesse caso, temos 2 colunas de interesse:

- Moeda;
- Cotação;

No entanto, só temos interesse de acessar a linha referente a cotação da variável que estamos buscando.

Vamos usar primeiro apenas o exemplo da cotação do dólar para entendermos melhor a construção do código.

	Produtos	Preço Original	Moeda	Cotação	Preço de Compra	Margem	Preço de Venda
0	Câmera Canon	999.99	Dólar	5	4999.95	1.40	6999.930
1	Carro Renault	4500.00	Euro	6	27000.00	2.00	54000.000
2	Notebook Dell	899.99	Dólar	5	4499.95	1.70	7649.915
3	IPhone	799.00	Dólar	5	3995.00	1.70	6791.500
4	Carro Fiat	3000.00	Euro	6	18000.00	1.90	34200.000
5	Celular Xiaomi	480.48	Dólar	5	2402.40	2.00	4804.800
6	Joia 20g	20.00	Ouro	350	7000.00	1.15	8050.000

O valor que será utilizado para alterar a cotação será o valor contido na nossa variável. O termo float, garante que ela será inserida como um número e não como um texto

```
tabela.loc[tabela["Moeda"] == "Dólar", "Cotação"] = float(cotacao_dolar)
```

A modificação ocorrerá no dataframe tabela que criamos ao importar a planilha excel

Método .loc permitirá selecionar apenas as linhas que atendam ao requisito de ter “Dólar” na coluna Moeda

Encontradas as linhas que possuem “Dólar” na coluna “Moeda”, o Loc irá alterar a coluna “Cotação”

# Atualizando os dados da planilha com as cotações (2/2)

Para as demais cotações, basta usar a mesma linha de código mas alterando os termos necessários.

```
tabela.loc[tabela["Moeda"] == "Dólar", "Cotação"] = float(cotacao_dolar)
tabela.loc[tabela["Moeda"] == "Euro", "Cotação"] = float(cotacao_euro)
tabela.loc[tabela["Moeda"] == "Ouro", "Cotação"] = float(cotacao_ouro)
display(tabela)
```

	Produtos	Preço Original	Moeda	Cotação	Preço de Compra	Margem	Preço de Venda
0	Câmera Canon	999.99	Dólar	5.476414	4999.95	1.40	6999.930
1	Carro Renault	4500.00	Euro	6.624895	27000.00	2.00	54000.000
2	Notebook Dell	899.99	Dólar	5.476414	4499.95	1.70	7649.915
3	IPhone	799.00	Dólar	5.476414	3995.00	1.70	6791.500
4	Carro Fiat	3000.00	Euro	6.624895	18000.00	1.90	34200.000
5	Celular Xiaomi	480.48	Dólar	5.476414	2402.40	2.00	4804.800
6	Joia 20g	20.00	Ouro	312.810000	7000.00	1.15	8050.000

Novas cotações inseridas

Perceba que os valores NÃO foram alterados. Precisamos recalculá-los

# Calculando os valores das colunas

Como vimos no slide anterior, as demais colunas do nosso dataframe **tabela** não foram calculadas automaticamente.

Logo, vamos precisar recalcular seus valores via Pandas.

A imagem ao lado apresenta o código necessário para essa operação.

Perceba que não precisamos calcular linha a linha, basta indicarmos quais são as colunas envolvidas na operação e a coluna que receberá os novos valores.

```
tabela["Preço de Compra"] = tabela["Preço Original"] * tabela["Cotação"]  
tabela["Preço de Venda"] = tabela["Preço de Compra"] * tabela["Margem"]  
display(tabela)
```

	Produtos	Preço Original	Moeda	Cotação	Preço de Compra	Margem	Preço de Venda
0	Câmera Canon	999.99	Dólar	5.476414	5476.359236	1.40	7666.902930
1	Carro Renault	4500.00	Euro	6.624895	29812.026083	2.00	59624.052165
2	Notebook Dell	899.99	Dólar	5.476414	4928.717836	1.70	8378.820321
3	iPhone	799.00	Dólar	5.476414	4375.654786	1.70	7438.613136
4	Carro Fiat	3000.00	Euro	6.624895	19874.684055	1.90	37761.899704
5	Celular Xiaomi	480.48	Dólar	5.476414	2631.307399	2.00	5262.614797
6	Joia 20g	20.00	Ouro	312.810000	6256.200000	1.15	7194.630000

Novas colunas calculadas

Parte 7

# Exportando o resultado para uma planilha Excel

# Exportando o resultado para uma planilha Excel

Agora que temos nosso dataframe atualizado, basta exportá-lo novamente para nosso computador como um arquivo Excel.

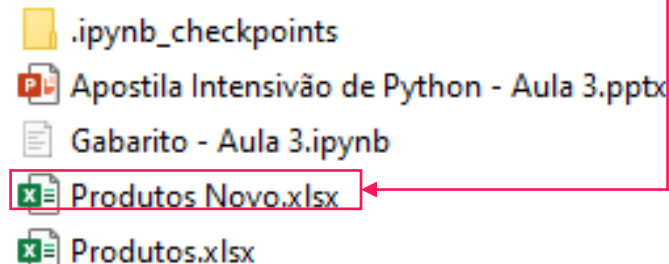
Para executar essa operação utilizaremos o método `.to_excel()`.

```
tabela.to_excel("Produtos Novo.xlsx", index=False)
```

Nome do novo arquivo criado

Indica que não deverá ser exportada a coluna com o índice das linhas

Nome



	A	B	C	D	E	F	G
1	Produtos	Preço Original	Moeda	Cotação	Preço de Compra	Margem	Preço de Venda
2	Câmera Canon	999,99	Dólar	5,476414	5476,359236	1,4	7666,90293
3	Carro Renault	4500	Euro	6,624895	29812,02608	2	59624,05217
4	Notebook Dell	899,99	Dólar	5,476414	4928,717836	1,7	8378,820321
5	iPhone	799	Dólar	5,476414	4375,654786	1,7	7438,613136
6	Carro Fiat	3000	Euro	6,624895	19874,68406	1,9	37761,8997
7	Celular Xiaomi	480,48	Dólar	5,476414	2631,307399	2	5262,614797
8	Joia 20g	20	Ouro	312,81	6256,2	1,15	7194,63

# INTENSIVÃO DE PYTHON {#}

100% ONLINE & GRATUITO

Ainda não segue a gente no Instagram e nem é inscrito no nosso canal do Youtube? Então corre lá!



@hashtagprogramacao



youtube.com/hashtag-programacao

