

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 extremamente manuais que além de repetitivas (chatas) são suscetíveis a erro visto que são feitas manualmente. Vamos aprender como criar um código que você possa resolver esse problema sem nem tocar no mouse 😊. Aprenda como **fazer uma uma automação com integração web** com os conceitos abaixo:

Importando dados
de bases .csv

Jupyter Notebook

Importando
bibliotecas

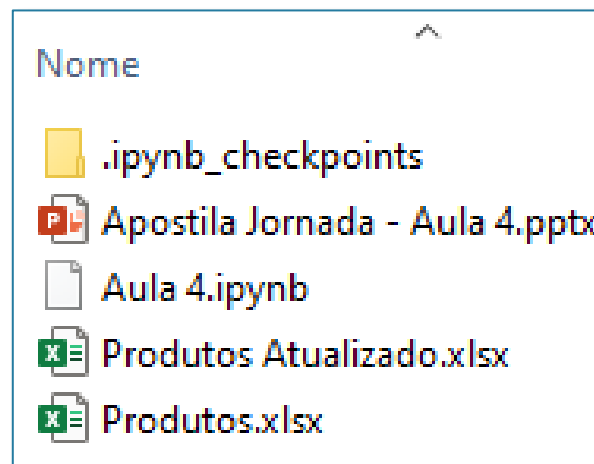
Webdriver

Usando Selenium

Após todos esses conhecimentos, seremos capazes de transformar um processo extremamente operacional

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

Produtos	Preço Base Original	Moeda	Cotação	Preço Base Reais	Percentual Preço Base	Preço Final
Câmera Canon	999,99	Dólar	5	4999,95	1,4	6999,93
Carro Renault	4500	Euro	6	27000	2	54000
Notebook Dell	899,99	Dólar	5	4499,95	1,7	7649,915
iPhone	799	Dólar	5	3995	1,7	6791,5
Carro Fiat	3000	Euro	6	18000	1,9	34200
Celular Xiaomi	480,48	Dólar	5	2402,4	2	4804,8
Joia 20g	20	Ouro	350	7000	1,15	8050



Introdução

Entendendo o problema

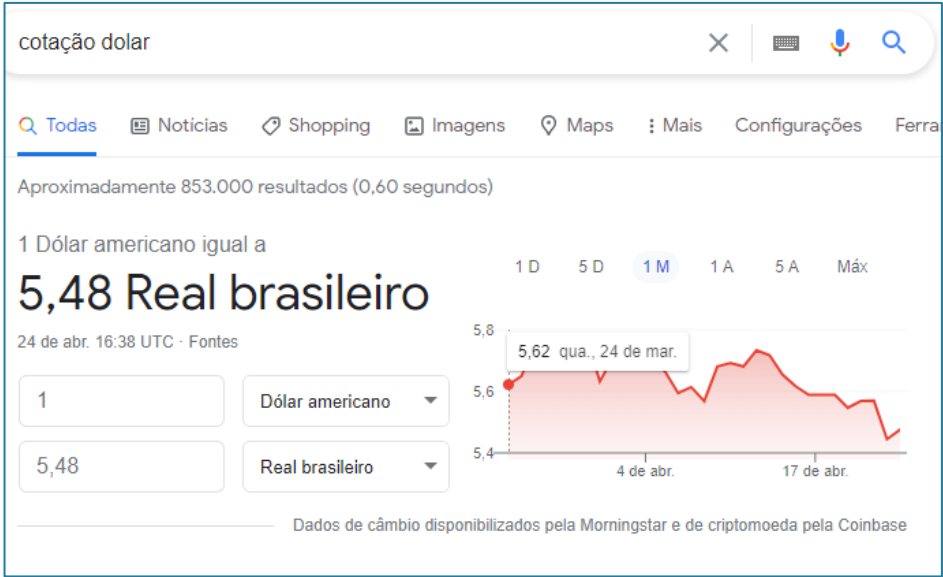
Nossa empresa fictícia, comercializamos produtos que são diretamente influenciados por moedas estrangeiras (Dólar, Euro) e ouro.

Nosso trabalho como analista de vendas é diariamente, atualizar a cotação das moedas 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, atualizar nossa planilha.

Produtos	Preço Base Original	Moeda	Cotação	Preço Base Reais	Percentual Preço Base	Preço Final
Câmera Canon	999,99	Dólar	5	4999,95	1,4	6999,93
Carro Renault	4500	Euro	6	27000	2	54000
Notebook Dell	899,99	Dólar	5	4499,95	1,7	7649,915
iPhone	799	Dólar	5	3995	1,7	6791,5
Carro Fiat	3000	Euro	6	18000	1,9	34200
Celular Xiaomi	480,48	Dólar	5	2402,4	2	4804,8
Joia 20g	20	Ouro	350	7000	1,15	8050



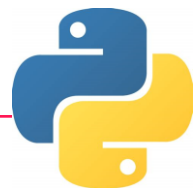
Website "melhorcâmbio.com". The header shows the logo "melhorcâmbio.com". Below the logo, there is a green banner with the text "Encontre a casa de câmbio com a melhor cotação e negocie para pagar ainda menos!". Below the banner, there are two input fields: "MOEDA..." with a dollar sign icon and "CIDADE..." 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" in red. 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.



Produtos	Preço Base Original	Moeda	Cotação	Preço Base Reais	Percentual Preço Base	Preço Final
Câmera Canon	999,99	Dólar	5,48	5479,9452	1,4	7671,92328
Carro Renault	4500	Euro	6,00	27000	2	54000
Notebook Dell	899,99	Dólar	5,48	4931,9452	1,7	8384,30684
IPhone	799	Dólar	5,48	4378,52	1,7	7443,484
Carro Fiat	3000	Euro	6,00	18000	1,9	34200
Celular Xiaomi	480,48	Dólar	5,48	2633,0304	2	5266,0608
Joia 20g	20	Ouro	350,00	7000	1,15	8050



Parte 2

Importando o Selenium

Importando o Selenium

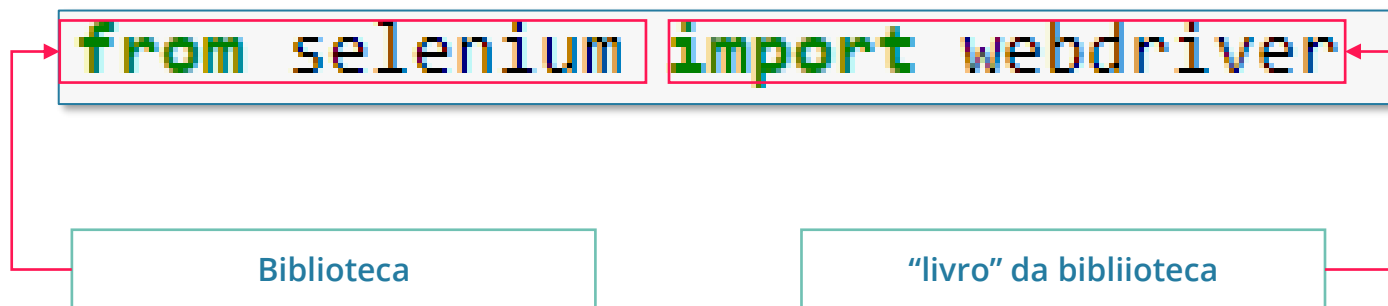
Como vimos na aula 1 da Semana do Python, vamos usar bibliotecas que nos facilitem importar dados de planilhas Excel, arquivos .csv, etc.

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 que estamos definindo qual a biblioteca e vamos dizer assim o “livro” dessa biblioteca.



Importando o webdriver

Como assim, um livro de uma biblioteca?

Antes de entendermos no python, imagine uma biblioteca de fato.

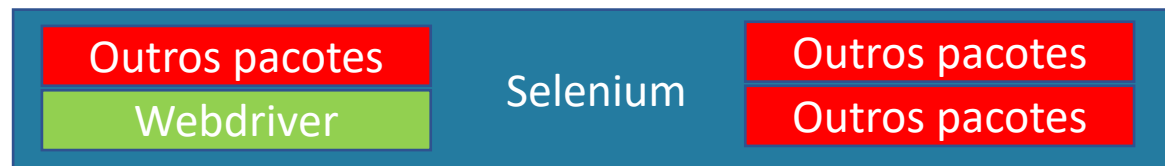
Grande, correto? Muitas das informações ali, são úteis mas não naquele momento...

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 **webdriver** (vamos explicar melhor o que é, e para que serve mais a frente).

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, 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 para a interface com a internet. Ela funciona como um robô que clica, insere dados, etc em páginas WEB como se você estivesse fazendo.

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

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



Selenium – webdriver (1/3)

Vamos para nosso código. O momento agora é criar um código que nos permita acessar o site 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 instalado 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 **nav**. Ela nos permitirá armazenar e transitar nos diferentes campos que encontraremos nas páginas WEB.

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

```
from selenium import webdriver
```

```
nav = webdriver.Chrome()
```

Variável auxiliar para
execução do Selenium

Indica que o navegador
utilizado que será
utilizado é o Google
Chrome

Interface com uma página na web

Selenium – webdriver (2/3)

Antes de continuarmos vamos instalar o webdriver.

Passo1: Entre no [link](#) para outros navegadores acessem os links ao lado.

Passo2: Baixe a última versão.

Chrome:	https://sites.google.com/a/chromium.org/chromedriver/downloads
Edge:	https://developer.microsoft.com/en-us/microsoft-edge/tools/webdriver/
Firefox:	https://github.com/mozilla/geckodriver/releases
Safari:	https://webkit.org/blog/6900/webdriver-support-in-safari-10/

Current Releases

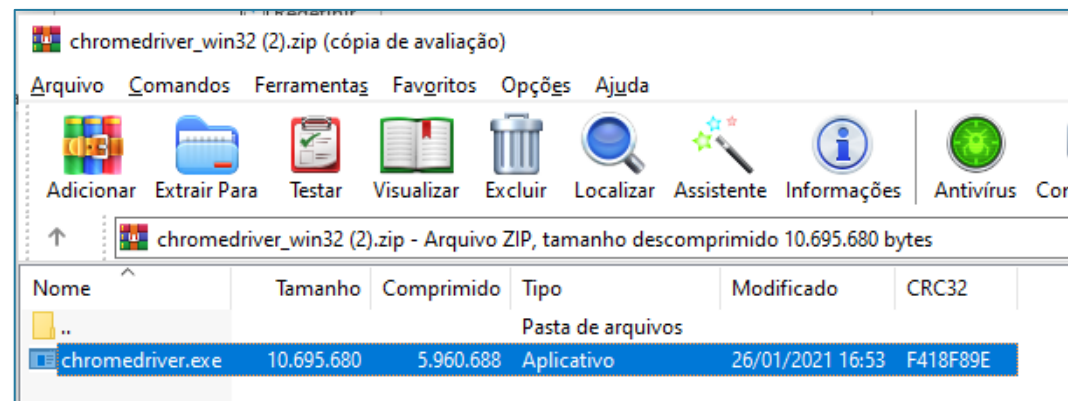
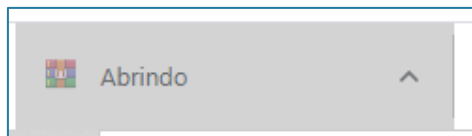
- If you are using Chrome version 89, please download [ChromeDriver 89.0.4389.23](#)
- If you are using Chrome version 88, please download [ChromeDriver 88.0.4324.96](#)
- If you are using Chrome version 87, please download [ChromeDriver 87.0.4280.88](#)
- For older version of Chrome, please see below for the version of ChromeDriver that supports it.

Name	Last modified
Parent Directory	
chromedriver_linux64.zip	2021-01-28 17:30:52
chromedriver_mac64.zip	2021-01-28 17:30:53
chromedriver_mac64_m1.zip	2021-01-28 17:30:55
chromedriver_win32.zip	2021-01-28 17:30:57
notes.txt	2021-01-28 17:31:00

Escolha seu sistema operacional.

Nesse caso Windows

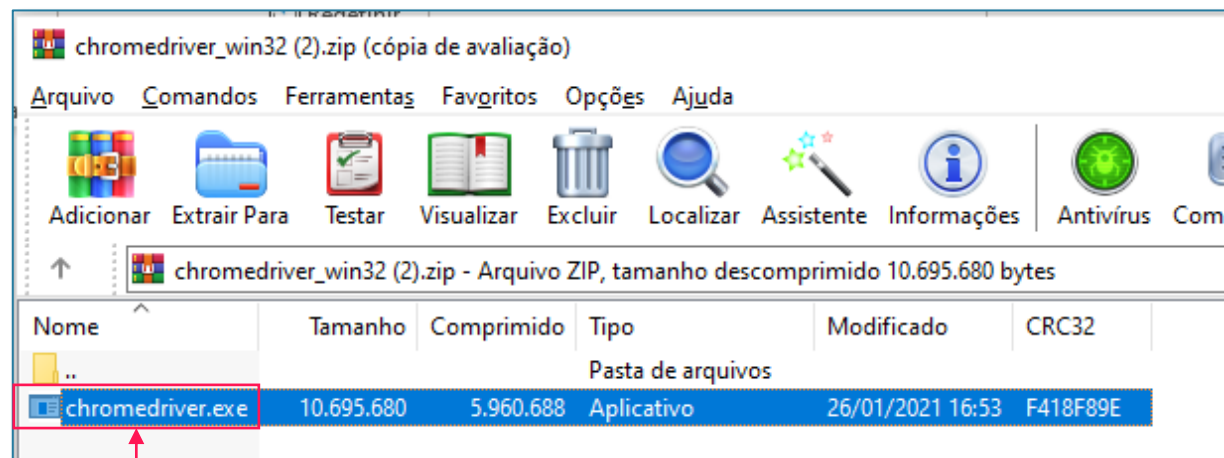
Passo3: Abra o arquivo baixado



Interface com uma página na web

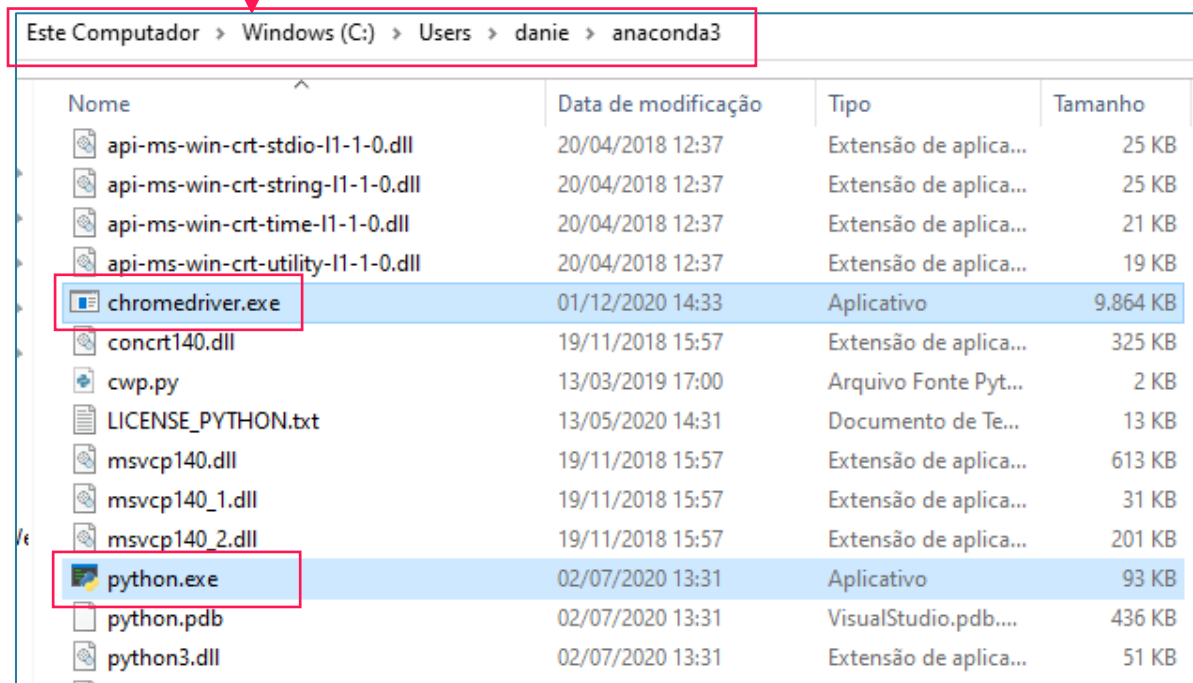
Selenium – webdriver (3/3)

Passo4: Coloque o arquivo **chromedriver.exe** na mesma pasta que o arquivo **python.exe**



Extraia o arquivo
chromedriver.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 manual. 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.

1

2

3

4

ENTER

15

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

Nossa primeira linha de código terá 2 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 desejado)**.

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
```

```
nav = webdriver.Chrome()
```

```
# pesquisar cotação dolar
nav.get("https://www.google.com/")
```

Nos auxilia a comunicar
com o site desejado

Site que desejamos
entrar

Variável auxiliar para execução do Selenium

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 include 'Recortar', 'Copiar', 'Colar', etc.

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

2

Clique no botão indicado abaixo;

A screenshot of the Chrome Developer Tools 'Elements' tab. A red box highlights the '<html itemscope itemtype="http://schema.R">' tag. A mouse cursor is pointing at the box.

3

Clique no campo;

A screenshot of the Chrome Developer Tools 'Accessibility' tab. It shows details for an 'input.gLfyf.gsfi' element, including its name 'Pesquisar' and role 'combobox'. A mouse cursor is pointing at the search input field in the background browser window.

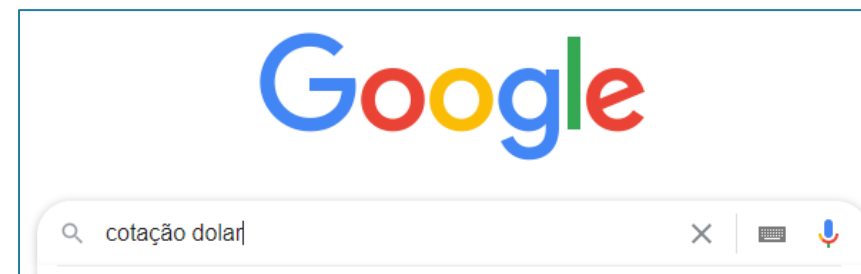
Continua

Buscando as cotações na Web

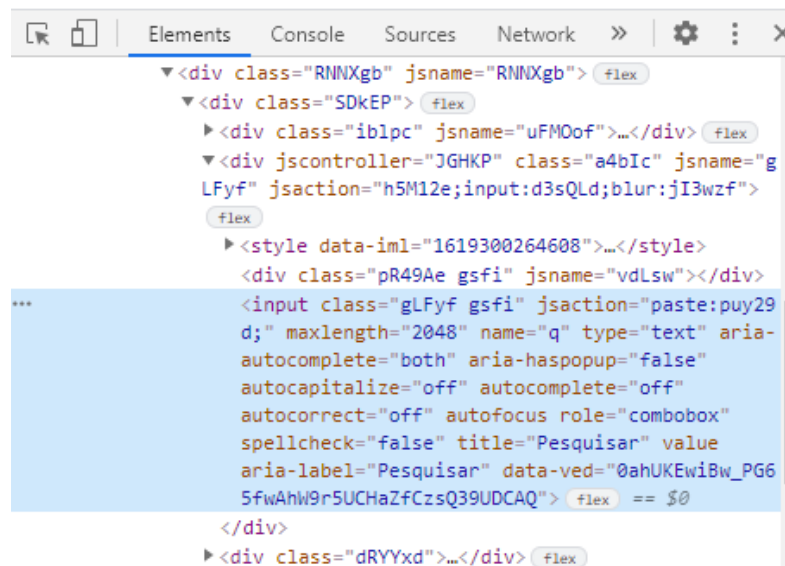
Cotação do Dólar (4/8)

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”;

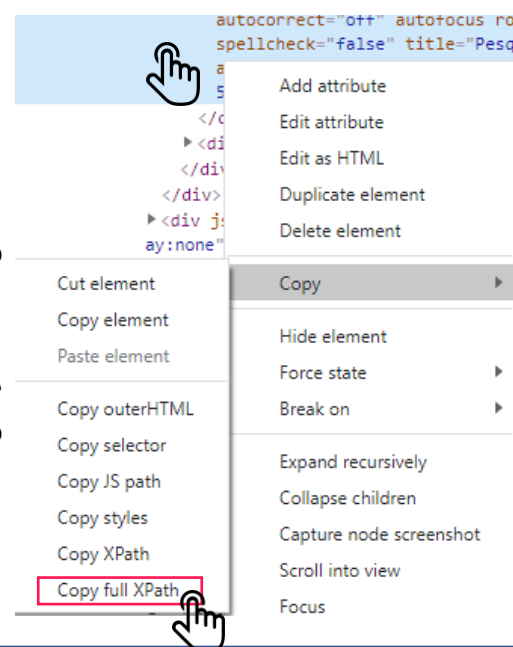


4 Linha de código que indica o campo selecionado



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()`

```
# pesquisar cotação dolar
nav.get("https://www.google.com/")
nav.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 **nav** 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()`

```
# pesquisar cotação dolar
nav.get("https://www.google.com/")
nav.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")
```

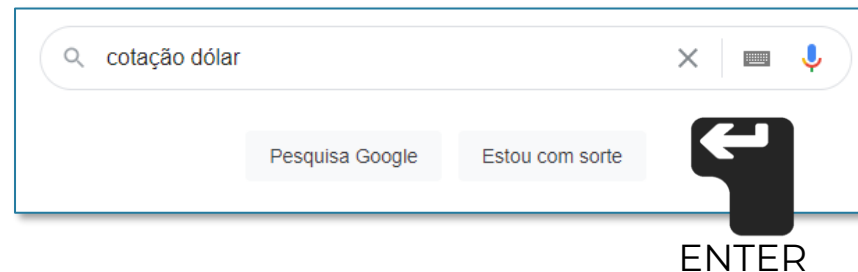
Esse método nos permitirá **após a seleção do campo de pesquisa** digitar “cotação dólar”.

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

nav = webdriver.Chrome()

# pesquisar cotação dolar
nav.get("https://www.google.com/")
nav.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")
nav.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 "", 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

nav = webdriver.Chrome()
```

```
# pesquisar cotação dolar
nav.get("https://www.google.com/")
nav.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")
nav.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 = nav.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 a variável que recebe o valor da cotação.

Podemos ver que temos 2 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_dólar)** 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

nav = webdriver.Chrome()

# pesquisar cotação dolar
nav.get("https://www.google.com/")
nav.find_element_by_xpath('/html/body/div[1]/div')
nav.find_element_by_xpath('/html/body/div[1]/div')
cotacao_dolar = nav.find_element_by_xpath('//*[@@')
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.

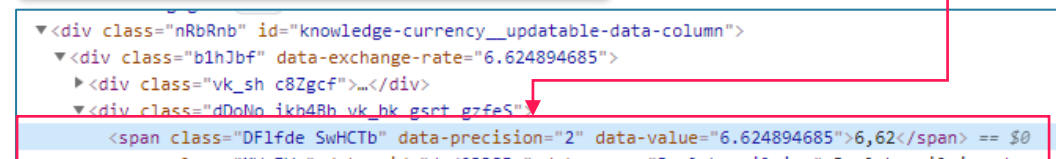
Perfeito, então podemos copiar o código do dólar e apenas avaliar o que é particular do dólar e substituir por euro.

```
# pesquisar cotação euro
nav.get("https://www.google.com/")
nav.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")
nav.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 = nav.find_element_by_xpath('//*[@id="knowledge-currency__updatable-data-column"]/div[1]/div[2]/span[1]').get_attribute('data-value')
print(cotacao_euro)
```

criação de uma nova variável 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

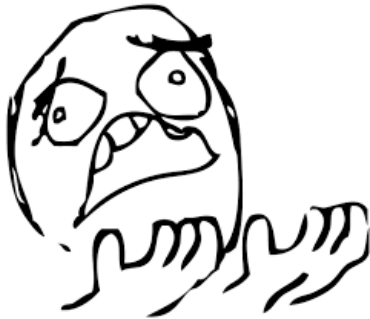
Buscando as cotações na Web

Cotação do Ouro (1/7)

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 (2/7)

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

Aqui, temos duas opções:

- 1) Criar um código para página 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 logo, the 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 "Ouro (g) 1,00" and "VALOR DO GRAMA DO OURO".
- 4**: A close-up of the "Ouro Hoje" page showing the price: "R\$ 312,81".

Buscando as cotações na Web

Cotação do Ouro (3/7)

Para nossa primeira etapa, usaremos o mesmo código utilizado anteriormente.

1) Acessar o site: <https://www.melhorcambio.com/>;

Mas aqui vamos adicionar uma nova linha que será usada apenas no passo 3, mas precisamos estar preparados.

Essa linha terá como objetivo armazenar em uma variável que a aba que estamos é a aba original. Por enquanto, é tudo que precisamos saber.

Usaremos aqui um novo método para essa operação:

window_handles

The diagram illustrates the Selenium code used to access the website and identify the original browser tab. It includes a code snippet with annotations:

```
# pegar cotacao ouro
nav.get("https://www.melhorcambio.com/")
aba_original = nav.window_handles[0]
```

Annotations:

- Como feito nas cotações anteriores nos permite acessar a página indicada** (points to the `nav.get()` line)
- Variável que armazenará que a página abaixo se trata da aba "original"** (points to the `aba_original` variable)
- Método do Selenium que nos permite "identificar" uma aba do navegador** (points to the `nav.window_handles` property)

Below the code, a screenshot of the **melhorcâmbio.com** website is shown, featuring the logo, a tagline, and input fields for currency and city.

Buscando as cotações na Web

Cotação do Ouro (4/7)

Segunda etapa será muito parecida com o que aprendemos anteriormente. Usando os métodos de busca do Selenium, precisamos identificar onde se encontra o link para a cotação do ouro.

- 1) Acessar o site: <https://www.melhorcambio.com/>;
- 2) Localizar o ícone que representa a cotação do ouro;

A diferença nesse caso é que agora vamos usar um novo método para interagir com a página. O método **click()**. Que como o nome já indica bem nos permite **clicar** no ícone identificado.

```
# pegar cotacao ouro
nav.get("https://www.melhorcambio.com/")
aba_original = nav.window_handles[0]
nav.find_element_by_xpath('//*[@id="commodity-hoje"]/tbody/tr[2]/td[2]/a/img').click()
```

XPATh que permite
localizar o ícone

O método .click() irá
permitir clicar no
ícone indicado pelo
caminho XPATh.



Buscando as cotações na Web

Cotação do Ouro (5/7)

A terceira etapa é essencialmente uma novidade em relação a tudo que vimos até agora.

- 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;

Podemos perceber que ao clicarmos no ícone indicado, uma nova aba é automaticamente aberta.

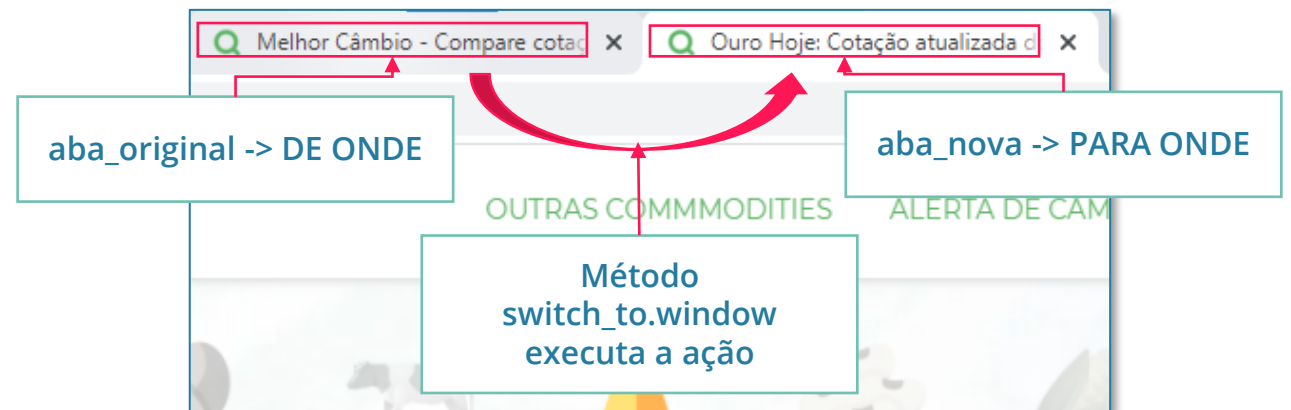
Precisamos migrar para essa nova aba. Para que isso aconteça precisamos informar DE ONDE e PARA ONDE.

Lembra que criamos uma variável `aba_original` no passo 1 sem explicar muito o motivo? Chegou a hora.

Ela será o nosso **DE ONDE** e a variável `aba_nova` que foi criada na linha de código indicada ao lado irá ser o nosso **PARA ONDE**.

Feito isso, usaremos o método `switch_to.window()` do Selenium para mudarmos de aba.

```
# pegar cotacao ouro
nav.get("https://www.melhorcambio.com/")
aba_original = nav.window_handles[0]
nav.find_element_by_xpath('//*[@id="commodity-hoje"]/tbody/
aba_nova = nav.window_handles[1]
nav.switch_to.window(aba_nova)
```



Buscando as cotações na Web

Cotação do Ouro (6/7)

Na quarta etapa iremos usar o mesmo princípio das outras aquisições de informação de páginas WEB que já fizemos anteriormente.

- 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.

No entanto, ao invés de usarmos o método find via XPATH, usaremos o método find com busca pelo ID.

Outro ponto de 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.

```
# pegar cotacao ouro
nav.get("https://www.melhorcambio.com/")
aba_original = nav.window_handles[0]
nav.find_element_by_xpath('//*[@id="commodity-hoje"]/tbody/tr[2]/td[2]/a/
aba_nova = nav.window_handles[1]
nav.switch_to.window(aba_nova)
cotacao_ouro = nav.find_element_by_id('comercial').get_attribute('value')
cotacao_ouro = cotacao_ouro.replace(",", ".")
print(cotacao_ouro)
```

Transforma
312,81 em 312.81

Usa a
variável
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 (7/7)

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

nav = webdriver.Chrome()

# pesquisar cotação dolar
nav.get("https://www.google.com/")
nav.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")
nav.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 = nav.find_element_by_xpath('//*[id="knowledge-currency__updateable-data-column"]/div[1]/div[2]/span[1]').get_attribute('text')
print(cotacao_dolar)

# pesquisar cotação euro
nav.get("https://www.google.com/")
nav.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")
nav.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 = nav.find_element_by_xpath('//*[id="knowledge-currency__updateable-data-column"]/div[1]/div[2]/span[1]').get_attribute('text')
print(cotacao_euro)

# pegar cotacao ouro
nav.get("https://www.melhorcambio.com/")
aba_original = nav.window_handles[0]
nav.find_element_by_xpath('//*[id="commodity-hoje"]/tbody/tr[2]/td[2]/a/img').click()
aba_nova = nav.window_handles[1]
nav.switch_to.window(aba_nova)
cotacao_ouro = nav.find_element_by_id('comercial').get_attribute('value')
cotacao_ouro = cotacao_ouro.replace(",", ".")
print(cotacao_ouro)

nav.quit()
```

5.476414
6.624894685
312.81

Fecha o navegador

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 Base Original	Moeda	Cotação	Preço Base Reais	Percentual Preço Base	Preço Final
2	Câmera Canon	999,99	Dólar	5,48	5479,9452	1,4	7671,923
3	Carro Renault	4500	Euro	6,00	27000	2	54000
4	Notebook Dell	899,99	Dólar	5,48	4931,9452	1,7	8384,307
5	IPhone	799	Dólar	5,48	4378,52	1,7	7443,484
6	Carro Fiat	3000	Euro	6,00	18000	1,9	34200
7	Celular Xiaomi	480,48	Dólar	5,48	2633,0304	2	5266,061
8	Joia 20g	20	Ouro	350,00	7000	1,15	8050



```
import pandas as pd
```

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

	Produtos	Preço Base Original	Moeda	Cotação	Preço Base Reais	Percentual Preço Base	Preço Final
0	Câmera Canon	999.99	Dólar	5.48	5479.9452	1.40	7671.92328
1	Carro Renault	4500.00	Euro	6.00	27000.0000	2.00	54000.00000
2	Notebook Dell	899.99	Dólar	5.48	4931.9452	1.70	8384.30684
3	IPhone	799.00	Dólar	5.48	4378.5200	1.70	7443.48400
4	Carro Fiat	3000.00	Euro	6.00	18000.0000	1.90	34200.00000
5	Celular Xiaomi	480.48	Dólar	5.48	2633.0304	2.00	5266.06080
6	Joia 20g	20.00	Ouro	350.00	7000.0000	1.15	8050.00000

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 nossa dataframe **produtos_df**.

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 Base Original	Moeda	Cotação	Preço Base Reais	Percentual Preço Base	Preço Final
0	Câmera Canon	999.99	Dólar	5.48	5479.9452	1.40	7671.92328
1	Carro Renault	4500.00	Euro	6.00	27000.0000	2.00	54000.00000
2	Notebook Dell	899.99	Dólar	5.48	4931.9452	1.70	8384.30684
3	IPhone	799.00	Dólar	5.48	4378.5200	1.70	7443.48400
4	Carro Fiat	3000.00	Euro	6.00	18000.0000	1.90	34200.00000
5	Celular Xiaomi	480.48	Dólar	5.48	2633.0304	2.00	5266.06080
6	Joia 20g	20.00	Ouro	350.00	7000.0000	1.15	8050.00000

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

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

A modificação ocorrerá no dataframe produtos_df 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.

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

	Produtos	Preço Base Original	Moeda	Cotação	Preço Base Reais	Percentual	Preço Base	Preço Final
0	Câmera Canon	999.99	Dólar	5.476414	5479.9452	1.40		7671.92328
1	Carro Renault	4500.00	Euro	6.624895	27000.0000	2.00		54000.00000
2	Notebook Dell	899.99	Dólar	5.476414	4931.9452	1.70		8384.30684
3	iPhone	799.00	Dólar	5.476414	4378.5200	1.70		7443.48400
4	Carro Fiat	3000.00	Euro	6.624895	18000.0000	1.90		34200.00000
5	Celular Xiaomi	480.48	Dólar	5.476414	2633.0304	2.00		5266.06080
6	Joia 20g	20.00	Ouro	312.810000	7000.0000	1.15		8050.00000

Novas cotações inseridas

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

Calculando os valores das colunas

Como vimos no slide anterior, as demais colunas do nosso `produto_df` não foram calculadas automaticamente.

Logo, vamos precisar via Pandas recalcular seus valores.

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.

```
produtos_df['Preço Base Reais'] = produtos_df['Cotação'] * produtos_df['Preço Base Original']  
produtos_df['Preço Final'] = produtos_df['Percentual Preço Base'] * produtos_df['Preço Base Reais']  
display(produtos_df)
```

	Produtos	Preço Base Original	Moeda	Cotação	Preço Base Reais	Percentual Preço Base	Preço Final
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

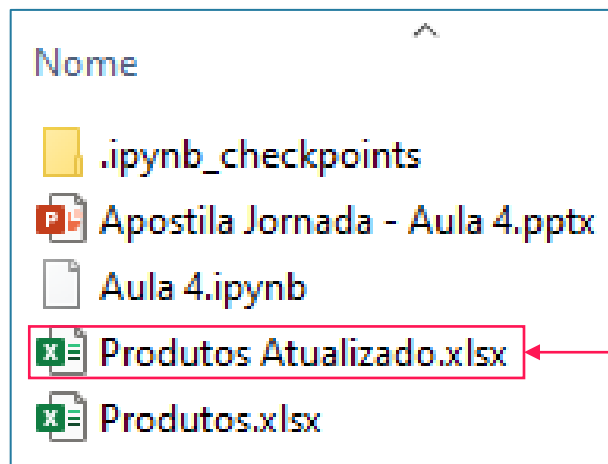
Exportando o resultado para uma planilha Excel

Exportando o resultado para uma planilha Excel

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

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

```
produtos_df.to_excel("Produtos Atualizado.xlsx", index=False)
```



Nome do novo arquivo criado

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

Produtos	Preço Base Orig	Moeda	Cotação	Preço Base Rentual	Preço	Preço Final
Câmera Câ	999,99	Dólar	5,476414	5476,359	1,4	7666,903
Carro Ren	4500	Euro	6,624895	29812,03	2	59624,05
Notebook	899,99	Dólar	5,476414	4928,718	1,7	8378,82
IPhone	799	Dólar	5,476414	4375,655	1,7	7438,613
Carro Fiat	3000	Euro	6,624895	19874,68	1,9	37761,9
Celular Xi	480,48	Dólar	5,476414	2631,307	2	5262,615
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

