



MANUAL

AUTOMATIZAR TAREAS

CLASE 2

Bienvenidos a la inmersión de la Semana de Python en la Práctica

¡Es un placer tenerte aquí con nosotros en esta inmersión!

Este folleto pretende presentar de forma sencilla y directa todo el contenido que se vio en la clase en vivo.

Aquí en Daxus Latam creemos firmemente en el aprendizaje basado en proyectos y por eso, cada día de inmersión trae un nuevo proyecto a desarrollar.

Recuerde siempre en nuestra metodología ORA:

OBSERVAR: Durante la clase en vivo, simplemente siga lo que se está haciendo y tome nota de las explicaciones.

REPETIR: luego, rehaz el proyecto con el material en mano y si tienes alguna dificultad, repasa la clase. Todas las clases se graban y están disponibles en YouTube durante la semana de inmersión.

APLICAR: Ahora que has observado y repetido, es hora de aplicar lo aprendido. Crear nuevos proyectos con los conocimientos adquiridos.

¿Qué aprenderemos?

- › Instalar y usar bibliotecas
- › Obtener automáticamente datos de stock
- › Generar estadísticas sobre estos datos
- › Automatiza el control del mouse y el teclado con **pyautogui**
- › Utilice cadenas de varias líneas
- › Enviar correo electrónico automáticamente

Yahoo Finanzas

Sitio que proporciona noticias financieras, incluidas **cotizaciones de acciones**.

Yahoo Finanzas

[Haz clic aquí para acceder](#)

Obteniendo acciones de forma automática

Instalación de yfinance y la biblioteca matplotlib

TERMINAL

```
!pip install yfinance
```

```
!pip install matplotlib
```

Obteniendo datos para una acción

ENTRADA

```
import yfinance

ticker = input("Ingrese el código de acción: ")
dt_inicial = input("Introduzca la fecha de inicio (aaaa-mm-dd): ")
dt_final = input("Introduzca la fecha de finalización (aaaa-mm-dd): ")
datos = yfinance.Ticker(ticker)
tabla = datos.history(start=dt_inicial, end=dt_final)
tabla
```

SALIDA

	Open	High	Low	Close	Volume	Dividends	Stock Splits
Date							
2020-01-02 00:00:00-03:00	11.171028	11.240596	11.097800	11.240596	37774500	0.0	0.0
2020-01-03 00:00:00-03:00	11.306501	11.438313	11.149060	11.149060	71595600	0.0	0.0
2020-01-06 00:00:00-03:00	11.141739	11.328472	10.965991	11.280873	81844000	0.0	0.0
2020-01-07 00:00:00-03:00	11.284532	11.306500	11.156382	11.236934	32822000	0.0	0.0
2020-01-08 00:00:00-03:00	11.236934	11.266226	11.072170	11.167367	48215600	0.0	0.0
...
2020-12-22 00:00:00-03:00	9.959295	10.058155	9.904372	9.988587	46513200	0.0	0.0
2020-12-23 00:00:00-03:00	10.043507	10.343751	10.014215	10.233906	49038900	0.0	0.0
2020-12-28 00:00:00-03:00	10.384027	10.442611	10.318120	10.318120	36313300	0.0	0.0
2020-12-29 00:00:00-03:00	10.391352	10.409660	10.248554	10.351076	29891100	0.0	0.0
2020-12-30 00:00:00-03:00	10.380369	10.431630	10.325446	10.376707	36968200	0.0	0.0

Seleccionar solo la columna Cierre (Close)

ENTRADA

```
cierre = tabla.Close
cierre
```

SALIDA

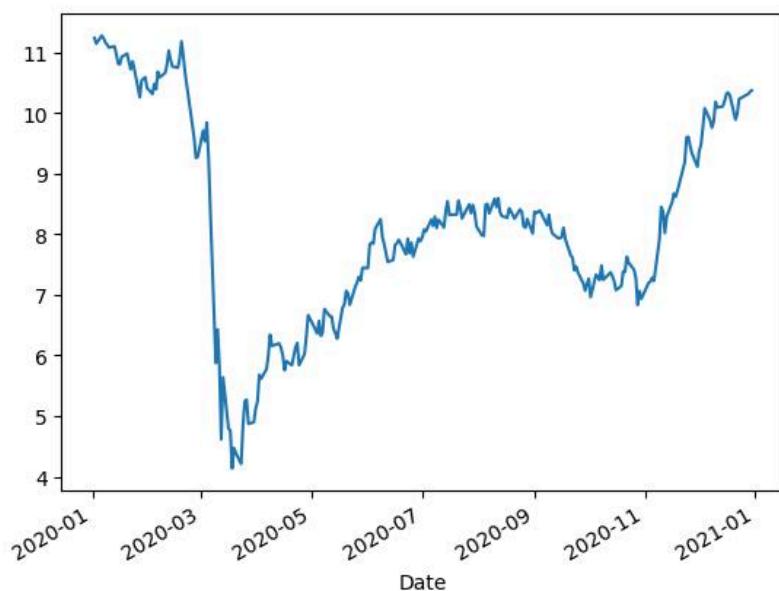
```
Date
2020-01-02 00:00:00-03:00    11.240596
2020-01-03 00:00:00-03:00    11.149060
2020-01-06 00:00:00-03:00    11.280873
2020-01-07 00:00:00-03:00    11.236934
2020-01-08 00:00:00-03:00    11.167367
...
2020-12-22 00:00:00-03:00    9.988587
2020-12-23 00:00:00-03:00    10.233906
2020-12-28 00:00:00-03:00    10.318120
2020-12-29 00:00:00-03:00    10.351076
2020-12-30 00:00:00-03:00    10.376707
Name: Close, Length: 248, dtype: float64
```

Generar un gráfico lineal (utiliza la biblioteca matplotlib)**ENTRADA**

```
cierre.plot()
```

SALIDA

```
<Axes: xlabel='Date'>
```



Generando estadísticas

A menudo necesitamos convertir entre tipos de datos y en Python esto es hecho de una manera muy sencilla.

ENTRADA

```
maxima = round(cierre.max(), 2)
minima = round(cierre.min(), 2)
valor_medio = round(cierre.mean(), 2)

print(maxima)
print(minima)
print(valor_medio)
```

SALIDA

```
11.28
4.13
8.17
```

Enviar correo electrónico automáticamente

Instalación de la biblioteca pyautogui

TERMINAL

```
!pip install pyautogui
```

Coordenadas en pantalla

Las coordenadas utilizadas en el código variarán según el monitor y la resolución de pantalla que estés utilizando. Utilice el código a continuación para mapear sus coordenadas.

Obtener las coordenadas en la pantalla

ENTRADA

```
import time
import pyautogui

time.sleep(5)
print(pyautogui.position())
```

Código

```
import pyautogui
import pyperclip
import webbrowser
from time import sleep

destinatario = "tu_email@gmail.com"
asunto = "Proyecto de análisis 2020"

mensaje = f"""
Buen día,
```

A continuación se presenta el análisis de la acción. {ticker} del periodo solicitado: {dt_inicial} a {dt_fin}:

```
Cotización máxima: USD {maxima}
Cotización mínima: USD {minima}
Valor medio: USD {valor_medio}
"""

# establecer una pausa entre las acciones de pyautogui
pyautogui.PAUSE = 3

# abre el navegador en gmail
webbrowser.open("www.gmail.com")
sleep(3)
```

Continuación

```
# Haz clic en el botón "Escribir"  
pyautogui.click(x=2503, y=314)  
  
# Complete para  
pyperclip.copy(destinatario)  
pyautogui.hotkey("ctrl", "v")  
pyautogui.hotkey("tab")  
  
# Complete el asunto  
pyperclip.copy(asunto)  
pyautogui.hotkey("ctrl", "v")  
pyautogui.hotkey("tab")  
  
# Llenar el cuerpo de e-mail  
pyperclip.copy(mensaje)  
pyautogui.hotkey("ctrl", "v")  
  
# Haga clic en el botón enviar  
pyautogui.click(x=4056, y=1360)  
  
# Cerrar la pestaña  
pyautogui.hotkey("ctrl", "f4")  
  
print("¡E-mail enviado exitosamente!")
```



SEMANA DE
python)
EN LA PRÁCTICA

Daxus
LATAM