Extraer Contenido HTML



a traves del codigo que te presento a continuación necesito que me envies hasta el contenedor de la carrera de ingenieria en sistemas computacionales, esta es la url https://itp.itpachuca.edu.mx/sistemas.php



Para extraer información específica de una página web, como el contenedor de la carrera de Ingeniería en Sistemas Computacionales desde la URL proporcionada, puedes usar la biblioteca `requests` junto con `BeautifulSoup` en Python. Aquí te muestro un ejemplo básico de cómo hacerlo:

```
python
 import requests
 from bs4 import BeautifulSoup
 # URL de la página web
 url = "https://itp.itpachuca.edu.mx/sistemas.php"
 # Realizar la solicitud HTTP
 response = requests.get(url)
 # Verificar que la solicitud fue exitosa
 if response.status code == 200:
     # Parsear el contenido HTML
     soup = BeautifulSoup(response.text, 'html.parser')
     # Buscar el contenedor de la carrera de Ingeniería en Sistemas Computacionales
     # Nota: Necesitarás inspeccionar el HTML de la página para encontrar el selector
 adecuado.
     # Aquí se asume que el contenedor tiene una clase específica o un ID.
     # Ejemplo de búsqueda por clase (ajusta el nombre de la clase según el HTML real)
     contenedor = soup.find('div', class_='nombre-de-clase-aqui')
     if contenedor:
         # Imprimir el contenido del contenedor
         print(contenedor.prettify())
     else:
         print("Contenedor no encontrado.")
 else:
     print("Error al acceder a la página:", response.status code)
```

Pasos a seguir:

1. Instala las bibliotecas necesarias si no las tienes ya:

```
pip install requests beautifulsoup4
```



- 2. **Inspecciona el HTML de la página** para identificar cómo está estructurado el contenedor de la carrera de Ingeniería en Sistemas Computacionales. Puedes hacer esto en las herramientas de desarrollador de tu navegador.
- 3. **Ajusta el selector** (`find()` en el ejemplo) de acuerdo con la estructura del HTML que encuentres. Por ejemplo, si el contenedor tiene un ID en lugar de una clase, deberás usar `soup.find('div', id='id-del-contenedor')`.
- 4. **Ejecuta el script** y debería mostrarte el contenido del contenedor.

Si necesitas ayuda para identificar el selector adecuado o tienes algún problema con el código, ¡avísame!



from selenium import webdriver from selenium.webdriver.chrome.service import Service from selenium.webdriver.common.by import By from selenium.webdriver.chrome.options import Options from webdriver_manager.chrome import ChromeDriverManager import time

Configura las opciones de Chrome (opcional) chrome_options = Options() chrome_options.add_argument("--headless") # Ejecutar en modo headless (sin interfaz gráfica)

Configura el driver de Chrome from selenium import webdriver

from selenium.webdriver.common.by import By

from selenium.webdriver.chrome.service import Service

from selenium.webdriver.chrome.options import Options

from selenium.webdriver.support.ui import WebDriverWait

from selenium.webdriver.support import expected_conditions as EC

from webdriver_manager.chrome import ChromeDriverManager

Configuración de opciones para el navegador

chrome_options = Options()

chrome_options.add_argument("--start-maximized") # Abre el navegador en pantalla completa

Configuración del controlador

service = Service(ChromeDriverManager().install())



```
# Inicializa el navegador
driver = webdriver.Chrome(service=service, options=chrome_options)
try:
  # Paso 1: Abre Wikipedia
  driver.get("https://itp.itpachuca.edu.mx/")
  # Espera a que el cuadro de búsqueda esté presente
  search_box = WebDriverWait(driver, 10).until(
    EC.presence_of_element_located((By.ID, "searchInput"))
  )
  search_box.send_keys("TecNM")
  search_box.submit()
  # Espera a que aparezcan los resultados
  pachuca_link = WebDriverWait(driver, 10).until(
    EC.presence_of_element_located((By.PARTIAL_LINK_TEXT, "Instituto Tecnológico de Pachuca"))
  )
  pachuca_link.click()
  # Espera a que la página del Instituto Tecnológico de Pachuca se cargue
  oferta_academica_link = WebDriverWait(driver, 10).until(
    EC.presence_of_element_located((By.LINK_TEXT, "Oferta académica"))
  )
  oferta_academica_link.click()
finally:
```



```
# Mantén el navegador abierto
input("Presiona Enter para cerrar el navegador...")
driver.quit()
```



Tu código usa Selenium para automatizar la interacción con una página web. Parece que quieres buscar información sobre la oferta académica en el sitio del Instituto Tecnológico de Pachuca. Sin embargo, hay algunos aspectos a ajustar y mejorar. Por ejemplo:

- 1. No es necesario importar `webdriver` y `By` más de una vez.
- 2. El `search_box` y la búsqueda de "TecNM" parecen irrelevantes para la URL que has proporcionado (https://itp.itpachuca.edu.mx/sistemas.php). En lugar de buscar, deberías simplemente ir directamente a la página de interés.

Aquí tienes una versión simplificada y corregida de tu script, que navega directamente a la página de Ingeniería en Sistemas Computacionales:

```
python
 from selenium import webdriver
 from selenium.webdriver.chrome.service import Service
 from selenium.webdriver.chrome.options import Options
 from selenium.webdriver.common.by import By
 from selenium.webdriver.support.ui import WebDriverWait
 from selenium.webdriver.support import expected_conditions as EC
 from webdriver_manager.chrome import ChromeDriverManager
 # Configuración de opciones para el navegador
 chrome options = Options()
 chrome_options.add_argument("--start-maximized") # Abre el navegador en pantalla completa
 # Configuración del controlador
 service = Service(ChromeDriverManager().install())
 # Inicializa el navegador
 driver = webdriver.Chrome(service=service, options=chrome options)
     # Paso 1: Abre la página de Ingeniería en Sistemas Computacionales
     driver.get("https://itp.itpachuca.edu.mx/sistemas.php")
     # Espera a que el contenido deseado esté presente
     # Ajusta el selector según la estructura real de la página
     contenedor = WebDriverWait(driver, 10).until(
         EC.presence_of_element_located((By.ID, "contenedor-carrera"))  # Cambia el ID
 según sea necesario
     # Imprime el contenido del contenedor
     print(contenedor.get_attribute('outerHTML'))
 finally:
     # Mantén el navegador abierto para depuración
     input("Presiona Enter para cerrar el navegador...")
     driver.quit()
```

Puntos clave:

- Revisa el HTML de la página de Ingeniería en Sistemas Computacionales para asegurarte de que el `ID` o el selector que usas es correcto (`"contenedor-carrera"` en el ejemplo es solo un marcador de posición).
- 2. **Asegúrate de tener el controlador de Chrome actualizado** con `webdriver_manager`, que ya está configurado en tu script.

Si el selector que usas no coincide con el HTML de la página, puedes ajustarlo usando las herramientas de desarrollador del navegador para encontrar el identificador adecuado. ¡Avísame si necesitas más ayuda con la configuración o el análisis de la página!



de la pagina itp.itpachuca.edu.mx usando webscraping accesa a la pagina de ingenieria en sistemas computacionales



Para acceder a la página de Ingeniería en Sistemas Computacionales de `itp.itpachuca.edu.mx` usando web scraping con Selenium, puedes seguir los pasos a continuación. Asegúrate de inspeccionar la página web para encontrar los identificadores correctos de los elementos. A continuación te muestro un ejemplo general de cómo hacerlo:

```
python
 from selenium import webdriver
 from selenium.webdriver.chrome.service import Service
 from selenium.webdriver.chrome.options import Options
 from selenium.webdriver.common.by import By
 from selenium.webdriver.support.ui import WebDriverWait
 from selenium.webdriver.support import expected conditions as EC
 from webdriver_manager.chrome import ChromeDriverManager
 # Configuración de opciones para el navegador
 chrome options = Options()
 chrome options.add argument("--headless") # Ejecutar en modo headless (sin interfaz
 gráfica)
 chrome options.add argument("--start-maximized") # Abre el navegador en pantalla completa
 # Configuración del controlador
 service = Service(ChromeDriverManager().install())
 # Inicializa el navegador
 driver = webdriver.Chrome(service=service, options=chrome options)
     # Paso 1: Abre la página principal del Instituto Tecnológico de Pachuca
     driver.get("https://itp.itpachuca.edu.mx/")
     # Espera a que el menú o enlace de "Oferta Académica" esté presente
     oferta_academica_link = WebDriverWait(driver, 10).until(
         EC.presence_of_element_located((By.LINK_TEXT, "Oferta académica"))
     oferta_academica_link.click()
     # Espera a que el enlace de la carrera de Ingeniería en Sistemas Computacionales esté
 presente
     ingenieria sistemas link = WebDriverWait(driver, 10).until(
         EC.presence_of_element_located((By.LINK_TEXT, "Ingeniería en Sistemas
 Computacionales"))
     ingenieria_sistemas_link.click()
```



```
# Aquí puedes extraer la información deseada
# Por ejemplo, imprimir el HTML de la página de Ingeniería en Sistemas Computacionales
contenido = driver.page_source
print(contenido)

finally:
    # Cierra el navegador
driver.quit()
```

Puntos importantes:

- 1. **Configuración de opciones**: Usa `--headless` para ejecutar el navegador sin interfaz gráfica. Puedes quitarlo si prefieres ver el navegador en acción.
- 2. **Espera explícita**: `WebDriverWait` se usa para esperar hasta que los elementos sean visibles y estén presentes en el DOM. Esto es importante para evitar errores de sincronización.
- 3. **Selección de elementos**: Asegúrate de usar el texto del enlace correcto en `**By.LINK_TEXT**` o el selector apropiado para encontrar el enlace deseado. Puedes ajustar los selectores basados en la estructura HTML real de la página.
- 4. **Extraer información**: `driver.page_source` te proporciona el HTML completo de la página después de que se haya cargado. Puedes usar BeautifulSoup para analizar este HTML si necesitas extraer información específica.

Si encuentras problemas con los selectores, puedes inspeccionar la página para obtener la información precisa sobre los identificadores y ajustar el código en consecuencia. ¡Avísame si necesitas más ayuda con el scraping!



como resuelvo este problema de manera detallada explicame Traceback (most recent call last):
File "C:\codigos\M2_2Webscraping\webScrapingITP.py", line 25, in <module>
oferta_academica_link = WebDriverWait(driver, 10).until(

File "C:\codigos\M2_2Webscraping\.venv\Lib\sitepackages\selenium\webdriver\support\wait.py", line 105, in until raise TimeoutException(message, screen, stacktrace) selenium.common.exceptions.TimeoutException: Message:



El error `TimeoutException` que estás encontrando indica que el elemento que estás tratando de encontrar no se carga dentro del tiempo especificado (10 segundos en tu caso). Esto puede suceder por varias razones, como:

- 1. El elemento no está presente en la página en el momento en que se está buscando.
- 2. El selector que estás utilizando para encontrar el elemento es incorrecto.
- 3. La página web puede tener dinámicas adicionales, como la carga de contenido asíncrono que requiere más tiempo para aparecer.

Cómo Resolver el Problema Detalladamente

Aquí hay un enfoque paso a paso para resolver este problema:



1. Verificar el Selector

Asegúrate de que el selector que estás usando es correcto. Puedes hacer esto inspeccionando el HTML de la página:

- Abre la página en tu navegador.
- Haz clic derecho en el enlace o elemento que deseas seleccionar y elige "Inspeccionar" (o "Inspeccionar elemento").
- Verifica el texto del enlace o el ID/clase del elemento.

Por ejemplo, si el enlace a "Oferta Académica" no se encuentra con `By.LINK_TEXT`, intenta usar `By.PARTIAL LINK TEXT` o busca por una clase o ID específico:

2. Ajustar el Tiempo de Espera

Si el contenido se carga dinámicamente, es posible que necesites más tiempo. Intenta aumentar el tiempo de espera:

3. Verificar la Carga Dinámica

Si el contenido se carga de manera asíncrona (es decir, con JavaScript después de que la página inicial se carga), considera esperar hasta que el contenido específico esté visible. A veces, puedes usar `EC.visibility_of_element_located` en lugar de `EC.presence_of_element_located` para asegurarte de que el elemento sea visible.

4. Depurar la Página

Asegúrate de que la página se está cargando correctamente y que el enlace que buscas está realmente en la página que estás accediendo. Puedes usar `print(driver.page_source)` para ver el HTML actual y verificar si el contenido esperado está presente.

Ejemplo Modificado

Aquí tienes un código modificado para manejar mejor el problema y realizar una verificación adicional:



```
python
 from selenium import webdriver
 from selenium.webdriver.chrome.service import Service
 from selenium.webdriver.chrome.options import Options
 from selenium.webdriver.common.by import By
 from selenium.webdriver.support.ui import WebDriverWait
 from selenium.webdriver.support import expected_conditions as EC
 from webdriver manager.chrome import ChromeDriverManager
 # Configuración de opciones para el navegador
 chrome options = Options()
 chrome_options.add_argument("--headless") # Ejecutar en modo headless (sin interfaz
 chrome_options.add_argument("--start-maximized") # Abre el navegador en pantalla completa
 # Configuración del controlador
 service = Service(ChromeDriverManager().install())
 # Inicializa el navegador
 driver = webdriver.Chrome(service=service, options=chrome options)
     # Paso 1: Abre la página principal del Instituto Tecnológico de Pachuca
     driver.get("https://itp.itpachuca.edu.mx/")
     # Espera a que el enlace de "Oferta Académica" esté presente y visible
     oferta academica link = WebDriverWait(driver, 20).until(
         EC.visibility_of_element_located((By.LINK_TEXT, "Oferta académica"))
     oferta academica link.click()
     # Espera a que el enlace de Ingeniería en Sistemas Computacionales esté presente y
 visible
     ingenieria sistemas_link = WebDriverWait(driver, 20).until(
         EC.visibility_of_element_located((By.LINK_TEXT, "Ingeniería en Sistemas
 Computacionales"))
     ingenieria sistemas link.click()
     # Imprime el HTML de la página de Ingeniería en Sistemas Computacionales
     contenido = driver.page_source
     print(contenido)
 finally:
     # Cierra el navegador
     driver.quit()
```

Resumen

- 1. Verifica el selector.
- 2. Ajusta el tiempo de espera.
- 3. Asegúrate de que la página y el contenido se carguen correctamente.

Si sigues teniendo problemas, comparte el HTML relevante o más detalles sobre la estructura de la página para proporcionar ayuda más específica.

ChatGPT puede cometer errores. Comprueba la información importante.