Python średnio zaawansowany

Dzień 4

Blok nr 2:

Akwizycja danych

AGENDA

- Web scraping
- Narzędzia

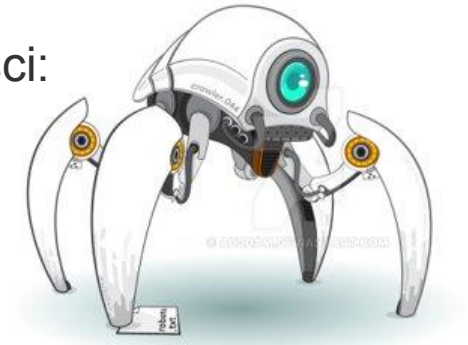
Web scraping

Web scraping

Web scraping to metoda automatycznego pozyskiwania danych ze stron internetowych przez boty (web crawler, harvester). Pozyskane dane mogą być przechowywane w różnych formatach, bazach danych, arkuszach kalkulacyjnych w celu późniejszej analizy.

Proces web scrapingu składa się z dwóch aktywności:

1. Pobranie strony internetowej
2. Wydobycie danych z dokumentu



Web scraping vs API

Dostęp:

- aplikacja internetowa nie oferuje dostępu poprzez API
- API nie udostępnia pełnego zestawu danych
- API może wymagać płatnego dostępu
- aktywność wynikająca z korzystania z API jest monitorowana

Dodatkowo:

- dokument HTML musi zostać sparsowany w celu ekstrakcji danych

Złota zasada web scrapingu

Nie korzystaj z web scrapingu jeżeli aplikacja internetowa oferuje dostęp poprzez API

Charakterystyka web scrapingu

- Kod odpowiedzialny za web scraping generuje requesty (symuluje aktywność użytkownika)
- Serwer aplikacji odpowiada na żądania crawlera
- Jeżeli crawler generuje ruch 1000 zapytań na minutę w niektórych przypadkach może to negatywnie wpłynąć na wydajność strony (nawet spowodować **tymczasową niedostępność zasobu**!)

Utrudnienia w web scrapingu

- Osoba odpowiedzialna za administrację serwerem aplikacji może zablokować ruch przychodzący z maszyny na której został uruchomiony crawler jeżeli zbyt często scrapujemy zawartość (jesteśmy traktowani jako intruz)

- Regulamin korzystania z usług strony internetowej może przewidywać **wyciągnięcie konsekwencji w przypadku korzystania z botów**

Rozwiązanie: jeśli piszesz bota, zapisuj ściągnięte strony i na nich lokalnie udoskonalaj kod żeby minimalizować ryzyko wykrycia

Utrudnienia w web scrapingu

- Scrapowanie stron korzystających z Javascript jest mocno utrudnione (dociągają kolejne zasoby)
- W przypadku zmiany struktury HTML na stronie konieczna jest zmiana kodu crawlera ☹
- Aplikacje sprawdzają nagłówki, zwłaszcza pola **User-Agent**
- Aplikacje sprawdzają obecność i zawartość cookies

Utrudnienia w web scrapingu

- Sprawdź zawartość pliku robots.txt, np. https://www.lotto.pl/robots.txt
- Kod HTML nie musi być poprawnie zbudowany
- CAPTCHA

- Badaj timeout
- Jesteś świadom różnic w aktywności przeglądarki i skryptu, np. czy myślisz o ściąganiu obrazków 1x1 piksel?

Narzędzia

Narzędzia deweloperskie

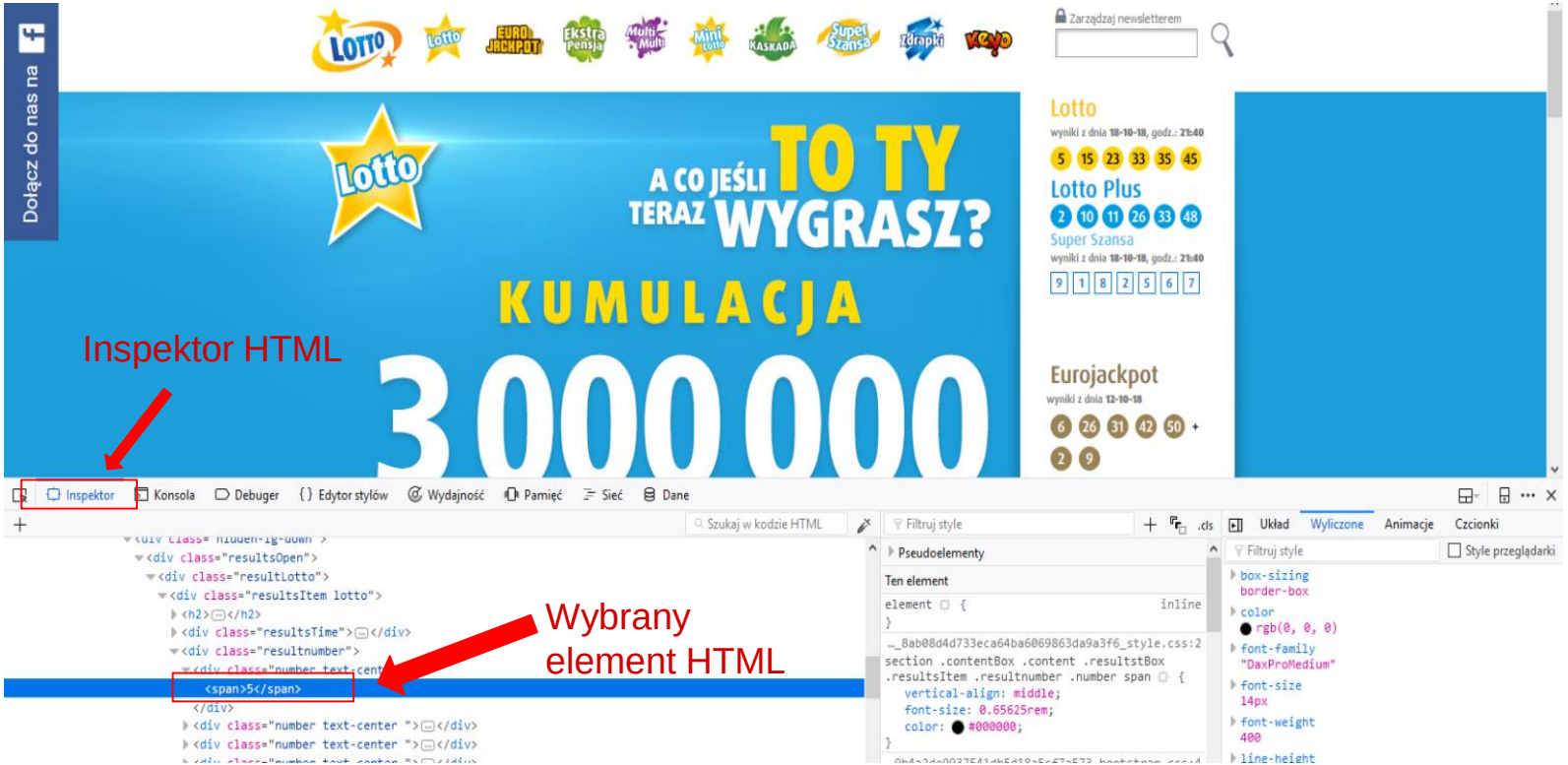
Przeglądarki internetowe oferują funkcjonalność inspektora elementów HTML aktualnie przeglądanej, kompletnej, strony internetowej.
Pozwala on analizować i edytować widoczny kod.

Kiedyś: Firebug 
Dziś:

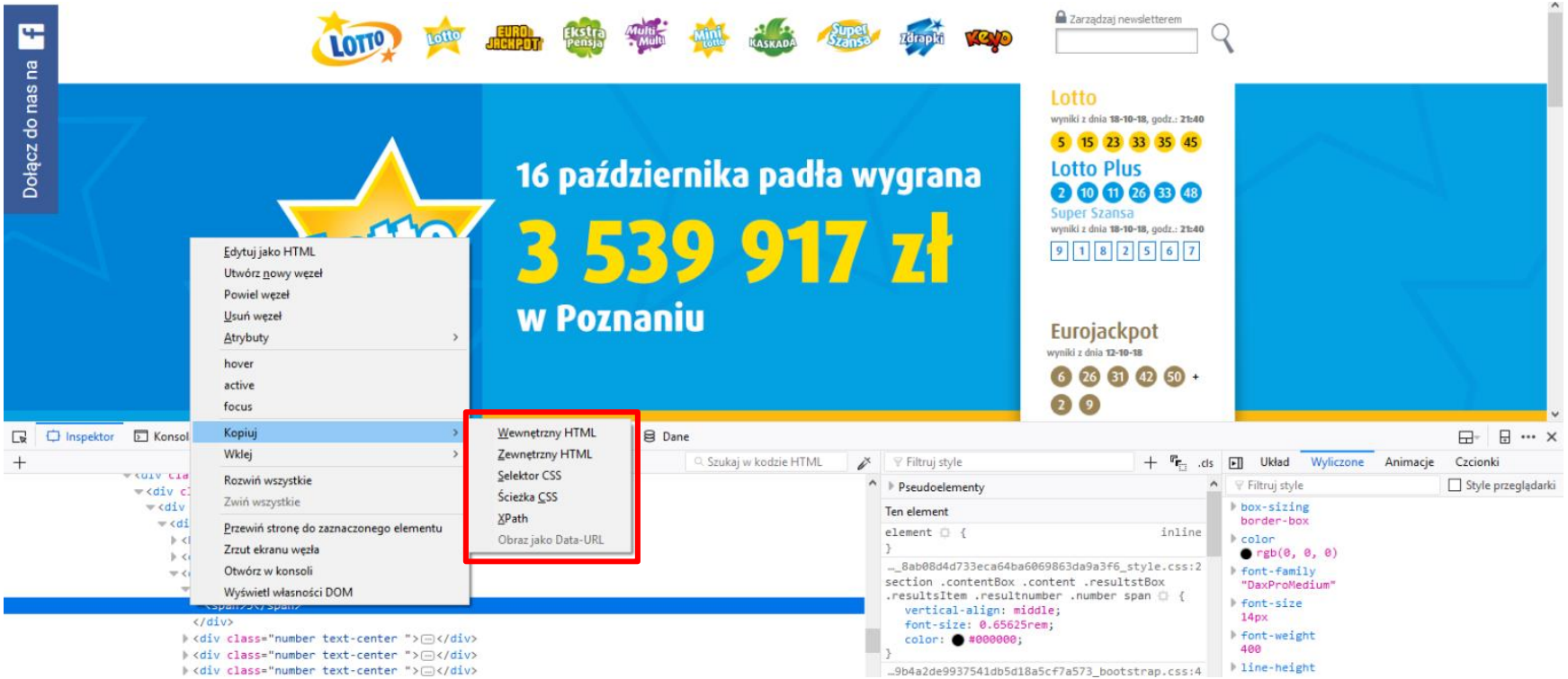
- **Firefox:**
  - F12 -> zakładka "Inspektor„
- **Chrome/Internet Explorer:**
  - F12 -> zakładka "Elementy"

Wyszukiwanie elementu HTML

Wyszukiwanie elementu HTML



Wyszukiwanie elementu HTML



Narzędzia

Scrapy:

- rozbudowany kombajn do tworzenia botów ściągających dane i dodatkowo:
  - umożliwia eksport danych w różnych formatach
  - loguje czynności
  - kompleksowo konfigurowalny (liczba botów, obsługa cookies, opóźnienia)
- średnio wysoki próg wejścia

BeautifulSoup + Requests

Dzięki!