

# Sprawozdanie nr 03

## Temat: Web Scraping w Pythonie

---

### Teoria:

**Zadaniem wirtualnego środowiska jest** stworzenie przestrzeni, w której możemy zainstalować konkretne pakiety dla konkretnego projektu.

Gdy masz projekt stworzony jest dla starszej wersji poszczególnej biblioteki, aktualizacja wersji wiąże się z potencjalnym ryzykiem uszkodzenia funkcjonalności w projekcie. Gdybyśmy mieli możliwość skorzystania jedynie z systemowej biblioteki musielibyśmy albo wszystkie funkcjonalności stale aktualizować do wymagań najnowszych wersji (co w przypadku dużych projektów może być czasochłonne) albo stale pracować na starszych wersjach lub manewrować pomiędzy wersjami, co z kolei, wiązałoby się z dużym nakładem pracy.

W takim wypadku przydaje się wirtualne środowisko. Zaletą takiego rozwiązania jest możliwość utworzenia własnego środowiska, dla każdego projektu osobno, dzięki czemu w naszych nowych projektach możemy pracować na najnowszych wersjach pakietów, a nasze starsze aplikacje dalej mogą działać i być budowane na starszych wersjach.

**Web scraping** jest procesem wyciągania danych ze stron internetowych w celu późniejszej analizy.

Wykorzystywany jest on, np. do porównywania cen na różnych portalach aukcyjnych.

Proces ten przeprowadzany jest przeważnie w sposób zautomatyzowany.

---

### Przebieg zadania:

Po stworzeniu folderu na projekt i przejściu do niego, żeby utworzyć wirtualne środowisko wykorzystujemy komendę: `py -m venv venv`

Gdzie pierwsze venv oznacza nazwę modułu, a drugie nazwę folderu wirtualnego środowiska. Nazwałem go venv według ogólnie przyjętego standardu.

```
D:\Projekt>cd "Sprawozdanie 3"
D:\Projekt\Sprawozdanie 3>mkdir webscrapper
D:\Projekt\Sprawozdanie 3>cd webscrapper
D:\Projekt\Sprawozdanie 3\webscrapper>py -m venv venv
```

Po chwili czekania, środowisko jest gotowe. Żeby się do niego dostać należy uruchomić skrypt `activate.bat` znajdujący się w folderze `Scripts` wewnątrz `venv`:

```
D:\Projekt\Sprawozdanie 3\webscrapper>venv\Scripts\activate.bat
(venv) D:\Projekt\Sprawozdanie 3\webscrapper>
```


























W tym momencie jesteśmy gotowi do pracy z tym środowiskiem, tj. możemy już np. dodawać pakiety do naszego wirtualnego środowiska. Możemy zobaczyć, że nasze środowisko jest aktywne, po wyświetlającej się nazwie katalogu „venv”, w nawiasach, po lewej stronie.

Wraz z projektem dodatkowo załączam plik *requirements.txt* w celu odtworzenia wirtualnego środowiska. Znajdują się w nim użyte pakiety oraz ich wersje, w schemacie umożliwiającym proste odtworzenie konfiguracji.

### Skrypt webscrapper.py

Skrypt ten ma za zadanie wyświetlać kilkanaście najbliższych meczy z gry Counter-Strike: Global Offensive za pośrednictwem strony: <https://www.gosugamers.net/counterstrike/matches>

Schemat wyświetlania meczy na stronie wygląda następująco:

 D13	vs.	 Renewal	MESA Pro Series #3   Main Event   Playoffs	LIVE		1x2
 VISU	vs.	 Enhanced	ED Mental Plays Invitational	LIVE		1x2
 Aura	vs.	 OE	Rix GG UKCS Cup   MAY   Playoffs	LIVE		1x2
 Horus eSpo	vs.	 YD	Rix GG UKCS Cup   MAY   Playoffs	LIVE		1x2
 Apeks	vs.	 Sangal	Elisa Invitational Summer 2021   Regional Swiss Stage   Rou...	LIVE		1x2  Live stats
 SINNERS	vs.	 Brute	Sazka eLEAGUE Spring 2021   Main Event   Group Stage	LIVE		1x2
 IQUE	vs.	 CalleNC	ED Mental Plays Invitational	LIVE		1x2
 One Tap	vs.	 UDP	Romanian Esports League Season 3   Main Event   Group St...	IN 25 MINUTES		1x2

Założeniem skryptu jest zachowanie w tablicy:

- nazw grających zespołów łącznie z ich odpowiadającymi państwami
- nazwy turnieju, w którym grają
- Daty meczu (lub w przypadku gry na żywo – napis „LIVE”)

```
D13 from Mongolia  
vs.  
Renewal from Mongolia  
On tournament: MESA Pro Series #3 | Main Event | Playoffs  
On day: Live
```

Rekord 0 z tablicy wyjścia – mecz 1 na stronie

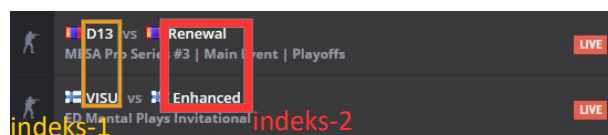
```
One Tap from Romania  
vs.  
UDP from Romania  
On tournament: Romanian Esports League Season 3 | Main Event | Group Stage  
On day: 16/05/2021 16:00
```

Rekord 7 z tablicy wyjścia – mecz 8 na stronie

Warto dodać, że czas na stronie wyświetlany jest relatywnie do aktualnej godziny. Jako że skrypt zakładał zachowanie dokładnej daty rozpoczęcia meczu, nie zmienia on tej daty na godzinę relatywną, jak to się dzieje na stronie.

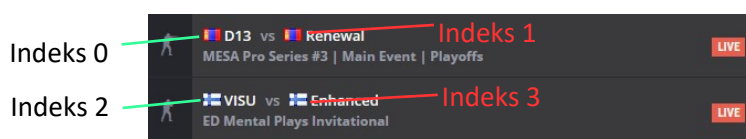
Skrypt dzieli się na 5 głównych funkcji i 1 dodatkową wyświetlającą zawartość listy meczy w wygodnej do przeglądania formie. Do głównych funkcji należą:

- findTeams(soup, team\_i) – Jej zadaniem jest uzyskanie wszystkich drużyn wyświetlających się po lewej stronie zakładek i zwrócenie listy tych drużyn:



Jako argumenty funkcja ta przyjmuje: obiekt BeautifulSoup z zawartością strony oraz indeks tablicy. Indeks tablicy pobierany jest na wejściu, ponieważ funkcja ta jest wywoływana dwukrotnie dla nazw drużyn po lewej, a następnie po prawej stronie (indeks 1, indeks 2).

- findNations(soup) – Jej zadaniem jest uzyskanie państw, z których pochodzą wszystkie drużyny. W odróżnieniu od pierwszej funkcji tutaj nie trzeba rozróżniać lewej i prawej strony (inny podział blokowy na stronie). Funkcja ta zwraca pełną listę państw dla obu stron, gdzie liczby 0 oraz liczby parzyste są indeksami państw drużyn z lewej strony, a liczby nieparzyste zaczynając od 1 są indeksami państw drużyn z prawej strony. Jako jedyny argument funkcja ta przyjmuje: obiekt BeautifulSoup.



- findMatches(soup) – Zadaniem tej funkcji jest zebranie nazw meczy w zakładkach i zwrócenie listy nazw. Ponownie jedynym argumentem funkcji jest obiekt BeautifulSoup.
- findDates(soup) – Ostatnia funkcja zbierająca dane ze strony, jej zadaniem jest zebranie dat rozpoczęcia meczy lub gdy mecz jest na żywo, zwrócenie tekstu: „LIVE”. Argumentem funkcji jest obiekt BeautifulSoup.
- combine(t1, t2, n, m, d) – Funkcja scalająca wszystkie zebrane dane w jedną spójną listę, gdzie jeden wpis zawiera informacje o jednym meczu w następującym formacie:

*[drużyna\_1] from [kraj\_drużyny\_1]  
vs.  
[drużyna\_2] from [kraj\_drużyny\_2]  
On tournament: [nazwa\_turnieju]  
On day: [data\_rozpoczęcia lub napis LIVE]*

Funkcja jako argumenty przyjmuje kolejno: t1 – drużyny z lewej strony zakładek, t2 – drużyny z prawej strony zakładek, n – kraje drużyn, m – nazwy turniej, d – data rozpoczęcia meczu lub napis „LIVE” dla meczu trwającego.

Funkcja w pętli dokonuje odpowiedniego formatowania (podanego wyżej) i zwraca listę meczy ze wszystkimi danymi.