



PRAKTIKUM IZ PROGRAMIRANJA 3

VEŽBE 5

Lazar Vasović, Luka Marković,
Andreja Živić, Jovan Janićijević



Biblioteka **webbrowser** se koristi za kontrolu veb pretraživača

```
import webbrowser
```

```
webbrowser.open("https://imi.pmf.kg.ac.rs/")
```

```
#otvara željeni url u podrazumevanom pretraživaču
```

```
browser=webbrowser.get("C:/Mozilla Firefox/firefox.exe %s")
```

```
#vraća kontroler za zadatu putanju do pretraživača
```

```
#nakon putanje mora se postaviti %s
```

```
#moraju se koristiti / umesto \\
```

```
browser.open("https://imi.pmf.kg.ac.rs/")
```



-Zahtevom(Request) se serveru može proslediti bilo kakav upit, odnosno od servera možemo tražiti podatke, možemo poslati podatke serveru ili ih menjati u zavisnosti od tipa zahteva koji šaljemo(dva osnovna tipa su GET i POST).

-U svakom od navedenih slučajeva server će poslati odgovor(Response) na zahtev, čak i u slučaju da nije u mogućnosti da uradi ono što smo od njega tražili.

-Svaki Response sadrži statusni kod koji određuje da li se zahtev uspešno poslao ili je došlo do nekakve greške i ako jeste do kakve greške je došlo

-Takođe pored statusnog koda odgovor sadrži i podatke koje server treba da nam pošalje u slučaju da je to potrebno

Kod	Značenje
200	OK
400	Bad request
403	Forbidden
404	Not Found
500	Internal Server Error



Biblioteka **requests** koristi se za slanje zahteva nekom server

-pip install requests

```
import requests  
res = requests.request('GET', 'https://imi.pmf.kg.ac.rs/')  
#res sadrži sve informacije vezane za odgovor servera  
#funkcija kao argumente redom prima metod zahteva i url redom
```

```
res.status_code  
#vraća status respons-a, iz statusa se može zaključiti da li je došlo  
do bilo kakve greške prilikom slanja zahteva ili je dobijen nekakav  
odgovor od servera
```

```
res.text  
#ukoliko nije došlo do greške sadrži tekst odgovora servera
```

Napisati program koji čuva veb stranicu u fajl

```
import requests
res = requests.request('GET', 'https://imi.pmf.kg.ac.rs/')

if (res.status_code == requests.codes.ok):
    f = open("index.html", 'wb')
    #potrebno je fajl otvoriti kao write binary zbog unicode
    #encoding-a

    #iter_content() kao povratnu vrednost vraća komade fajla
    #određene veličine u bajtovima koju prosledjujemo kao
    #parametar
    for chunk in res.iter_content(100000):
        f.write(chunk)

f.close()
```



Za parsiranje HTML fajla koristi se BeautifulSoup modul koji omogućava pronalaženje specifičnih delova veb stranice

-pip install beautifulsoup4

```
import requests, bs4
```

```
#pravljenje BeautifulSoup objekta od html fajla
```

```
html = open("index.html")
```

```
bsHtml = bs4.BeautifulSoup(html, features='html.parser')
```

```
html.close()
```

```
#pravljenje BeautifulSoup objekta od veb stranice
```

```
res = requests.request('GET', "https://imi.pmf.kg.ac.rs/")
```

```
if (res.status_code == requests.codes.ok):
```

```
    soup = bs4.BeautifulSoup(res.text, features='html.parser')
```



Ispisati naslov stranice i pronaći sve linkove na stranici IMI-ja

```
import requests, bs4

res = requests.request('GET', "https://imi.pmf.kg.ac.rs/")
if (res.status_code == requests.codes.ok):
    soup = bs4.BeautifulSoup(res.text, features='html.parser')

    #naslov veb stranice
    print(soup.title.text)

    #vraća sve tagove 'a' sa svim njihovim atributima i
    vrednostima
    linkovi = soup.find_all('a')
    for link in linkovi:
        print(link.get('href'))
        print(link.get_text()) #vraća text taga
```



Beautiful Soup objekat sadrži metodu `select()` koja služi za dopremanje HTML elemenata preko CSS selektora(id, ime klase, ime taga)

```
soup.select('tag')
```

```
soup.select('#id')
```

```
soup.select('.class')
```

```
soup.select('tag1 tag2')
```

#vraća sve elemente `<tag2>` unutar elemenata `<tag1>`

```
soup.select('tag1 > tag2')
```

#vraća sve elemente `<tag2>` unutar elemenata `<tag1>` bez drugih elemenata između

```
soup.select('tag[atribut]')
```

#svi elementi `<tag>` sa atributom `atribut` koji ima bilo koju vrednost

```
soup.select('tag[atribut="vrednost"]')
```

#svi elementi `<tag>` sa atributom `atribut` koji ima odgovarajuću vrednost

Različiti paterni se mogu kombinovati prilikom pozivanja select() metode

```
soup.select('.class tag')  
#svi elementi <tag> unutar klase .class
```

```
soup.select('tag > #id')  
#svi elementi sa odredjenim id-om koji su direktno unutar <tag>
```

```
soup.select('tag[atribut='vrednost'] > .class')
```

Napisati program koji učitava parametar pretrage po google-u kao string. Pretraga se vrši preko linka 'http://google.com/search?q=' + parametar. Rezultat pretrage je veb stranica koja sadrži linkove ka željenim stranicama. Linkovi su vrednosti atributa *href* elemenata sa tagom `<a>` koji se nalaze unutar elementa čija je klasa *r*. Izdvojiti prvih 10 linkova pretrage i otvoriti ih u browseru.



```
import requests, bs4, webbrowser

search = input()

res=requests.request('GET','http://google.com/search?q='+search)

if (res.status_code == requests.codes.ok):
    soup = bs4.BeautifulSoup(res.text, features='html.parser')
    for element in soup.select('.r a')[:10]:
        webbrowser.open(element.get('href'))
```