

Vježba 2. –JS, tipovi, funkcije, petlje

Upute za rješavanje lab. vježbi

Današnje vježbe nisu vezane uz prethodne dvije ali ih svakako uključite u svoj projekt za vrijeme rješavanja ili preuzmite predložak s prošlih.

Kako bi uspješno položili vježbu potrebno je riješiti **barem tri zadatka** dok preostale ako ne stignete možete dovršiti kod kuće do sljedećeg termina vježbi. Sve zadatke možete rješavati u jednoj datoteci **zadaci.js** i koristeći se komentarima (**/* */**) isključiti dijelove koda. U svakom zadatku pronaći ćete mali *hint* koji vam može pomoći u rješavanju zadataka. Za vrijeme rješavanja dozvoljeno je koristiti se naredbama i petljama koje nismo spominjali na predavanjima, pod uvjetom da razumijete vlastiti *kôd* i samostalno rješavate.

Postavljanje JS datoteke u projekt

Preuzmite svoj projekt i u mapu **js** kreirajte datoteku **zadaci.js**. Unutar datoteke **index.html** pozovite datoteku **zadaci.js** sljedećom naredbom:

```
...  
    <script src="/js/zadaci.js"></script>  
  </body>  
</html>
```

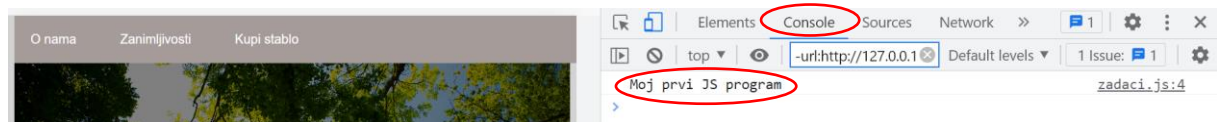
Provjera ispravnosti

U datoteci **zadaci.js** ispišite jednostavnu poruku u skočnom prozoru

```
alert("Moj prvi JS program!");
```

i istu poruku ispišite u konzoli. Poruku u konzoli provjerite na stranici opcijom **Desni klik -> Inspect Element** ili tipkom **F12**.

```
console.log("Moj prvi JS program");
```



Zadaci

1. Napisati **void** funkciju koji računa aritmetičku sredinu tri prirodna broja koja unosi korisnik putem skočnog prozora (**prompt**), a rezultat se ispisuje u konzoli.

[FUNCTIONS](#)

[IF – ELSE IF –ELSE](#)

2. Napisati **void** funkciju koji računa zbroj, razliku, umnožak i količnik dvaju brojeva. Vrijednosti brojeva i znak za operaciju (+, -, /, *) prosljeđuju se kroz poziv funkcije kao argument, a rezultat se ispisuje u konzoli.

[FUNCTIONS](#)

[OPERATORS](#)

3. Napisati funkciju koja kao **povratnu** vrijednost vraća prvu znamenku unesenog deseteroznamenkastog broja npr. za broj 1254192518 funkcija vraća 1 i rezultat ispisuje u konzoli.

[FUNCTIONS](#)

[OPERATORS](#)

[WHILE](#)

4. Napisati **void** funkciju koji unosi mjesec i ispisuje koje je godišnje doba u skočnom prozoru (**alert**). Za 1., 2. i 3. mjesec ispisuje zima, za 4., 5. i 6. ispisuje proljeće, za 7., 8. i 9. mjesec program ispisuje ljeto, a za 10., 11. i 12. ispisuje jesen.

[FUNCTIONS](#)

[SWITCH](#)

[IF – ELSE IF –ELSE](#)

5. Napisati funkciju koja provjerava broj pojavljivanja znamenke **z** u prirodnom broju **n**. Funkcija za unose korisnika npr. $n = 2955$, $z = 5$, vraća poruku „U broju 2955 imamo 2 pojavljivanja broja 5.“, te ispisuje u konzoli.

[FUNCTIONS](#)

[OPERATORS](#)

[WHILE](#)

Predavanje vježbe i praćenje

Nakon što ste riješili vježbu i o tome obavijestili asistenta, kreirajte komprimiranu datoteku (.zip) i predajte u svoju grupu.

Na Moodleu je dostupna poveznica o evidenciji vježbi.

Vježba nije položena ako:

- a) za vrijeme lab. vježbi niste riješili dio obveznih zadataka (barem tri zadatka),
- b) ste riješili vježbu ali niste pratili zadanu strukturu,
- c) ako ste predali vježbu nakon roka od tjedan dana,
- d) ako ste predali u pogrešnu grupu,
- e) ako ste predali nepotpunu vježbu (zaboravili postaviti određene mape ili datoteke),
- f) ako ste prepisali ili plagirali,
- g) izostali,
- h) ako ste neuspješno obranili i slično.

Vježba je položena ako:

- a) ste riješili dio zadatka, barem tri po izboru, za vrijeme lab. vježbi i ovisno o vremenu preostala dva zadatka u zadanom roku od tjedan dana,
- b) ste uspješno riješili sve zadatke i predali na Moodle.

Jednom predana vježba nakon roka ne smije se izmjenjivati. Nakon predane vježbe, asistent može usmeno, pismeno ili na računalu ispitati studenta (kratki zadaci, pojašnjenje koda, izmjena postojećeg koda i slično). U slučaju neuspješne obrane, student može obraniti vježbu u tjednu nadoknada prije kolokvija.