O AUTORIMA

Laurence Lars Svekis je stručnjak za inovativnu tehnologiju sa znanjem iz različitih oblasti i stvarnim iskustvom u razvoju veb stranica; učestvovao je od 1999. godine u različitim projektima veb razvoja, kako velikim, tako i malim. On je vrhunski instruktor kurseva od 2015. godine i voli da „oživljava“ ideje na Internetu. Podučavanje i pomaganje drugima je bila neverovatna prilika za njega, jer uživa da deli znanje sa drugima. Ima entuzijazam za edukaciju i želju da pomogne drugima da iskuse radost razvoja aplikacija i veb dizajna.

Maaike van Putten je softverski programer i trener koji je ljubitelj razvoja softvera i pomaže drugima da pređu na sledeći nivo u svojoj karijeri. Neki od njenih omiljenih programskih jezika su JavaScript, Java i Python. Učestvuje kao programer u projektima razvoja softvera i kao trener za različite teme, od IT-a za neznalice do naprednih tema za više softverske programere. Osim toga, voli da kreira sadržaj na Internetu za različite platforme kako bi pomogla široj publici.

Rob Percival je veoma cenjen veb programer i Udemy instruktor, sa više od 1,7 miliona učenika. Više od 500.000 učenika je položilo Robov kurs „Complete Web DeveloperCourse 2.0“ i kurseve za Android i iOS programere.

O RECENZENTU

Chris Minnick je poznati autor, bloger, trener, govornik i veb programer. Njegova kompanija „WatzThis?“ posvećena je pronalaženju boljih načina da početnici ovladaju računarskim i programskim veštinama.

Chris je full-stack programer više od 25 godina i trener više od 10 godina. Predaje veb razvoj, ReactJS i napredni JavaScript u mnogim velikim svetskim kompanijama, ali i u javnim bibliotekama, u zajedničkim kancelarijskim prostorima i na okupljanjima.

Minnick je autor i koautor više od deset tehničkih knjiga za odrasle i decu, uključujući i „React JS Foundations, HTML and CSS for Dummies“, „Coding with JavaScript for Dummies“, „JavaScript for Kids“, „Adventures in Coding“ i „Writing Computer Code“.

Pridružite se Discord prostoru naše knjige

Pridružite se Discord radnom prostoru za knjigu na mesečnoj sesiji „Ask me anything“ sa autorima: https://packt.link/JSBook.

Predgovor

JavaScript je neverovatan multifunkcionalni jezik koji se mnogo koristi, između ostalog, za razvoj veb stranica (). Svaka interakcija koja se dešava na veb stranicama je JavaScript u akciji. U stvari, svi savremeni pregledači razumeju JavaScript, a ubrzo ćete ga razumeti i vi.

U ovoj knjizi se bavimo svime što je potrebno da znate da biste kreirali JavaScript aplikacije i koristili JavaScript na veb stranicama. Kada pročitate ovu knjigu u celosti, moći ćete da kreirate interaktivne veb stranice, dinamične aplikacije i još mnogo štošta, dok budete napredovali na svom profesionalnom JavaScript „putovanju"!

Za koga je ova knjiga

Da biste započeli rad pomoću ove knjige, nije vam potrebno iskustvo u JavaScriptu. Međutim, ako imate neko iskustvo u kodiranju, verovatno da će vam knjiga i vežbe biti lakši. Osnovno poznavanje HTML-a i CSS-a bilo bi od koristi. Ako ste početnik programer, upućujemo vam dobrodošlicu u svet programiranja. U početku JavaScript može vam se učiniti teškim, ali mi ćemo vam biti vodič.

Šta obuhvata ova knjiga

U Poglavlju 1, „Početak rada u JavaScriptu", obuhvaćene su neke osnove JavaScript jezika koje ćete morati da znate da biste razumeli ostatak knjige.

U Poglavlju 2, „Osnove JavaScripta", razmotrene su osnove, kao što su promenljive, tipovi podataka i operatori.

U Poglavlju 3, „Više vrednosti JavaScripta", objašnjeno je kako da uskladištite više vrednosti u jednoj promenljivoj pomoću nizova i objekata.

U Poglavlju 4, „Logički iskazi“, počinje prava zabava: koristićete logičke iskaze koji će automatski donositi odluke!

U Poglavlju 5, „Petlje“, objašnjena je situacija u kojoj je potrebno ponavljanje bloka koda, za koje koristite petlje. Koristićete različite tipove petlji, kao što su for i while.

U Poglavlju 6, „Funkcije“, predstavljen je veoma koristan blok za ponavljanje isečaka koda - funkcije! To vam omogućava da u bilo kojem trenutku u skriptu pozovete određeni blok koda da uradi nešto automatski. Blok će vam pomoći da ne ponavljate pisanje istog koda, što je jedan od osnovnih principa pisanja jasnog koda.

U Poglavlju 7, „Klase“, nastavljeno je razmatranje gradivnih blokova JavaScripta koji pomažu da bolje strukturirate aplikacije. Već ste videli kako se kreiraju objekti, a pomoću klasa ćete naučiti kako da kreirate šablon za objekte, koji možete ponovo da koristite kad god vam zatreba određeni tip objekta.

U Poglavlju 8, „Ugrađeni JavaScript metodi“, razmotrene su neke odlične ugrađene funkcije. Funkcije su nešto što možete sami da napišete, ali često ćete koristiti ugrađene JavaScript funkcije kad god treba da izvršite uobičajene zadatke, kao što je provera da li je nešto broj ili nije.

U Poglavlju 9, „Objektni model dokumenta“, „zaronićete“ u objektni model pregledača i objektni model dokumenta (document object model – DOM). Oni će mnogo obogatiti načine na koje možete da koristite JavaScript. Naučićete šta je DOM i kako možete da utičete na njega pomoću JavaScripta da biste promenili vaše veb stranice.

U Poglavlju 10, „Manipulacija dinamičkim elementima pomoću DOM-a“, prikazano je kako se dinamički manipuliše elementima DOM-a, što će vam omogućiti da kreirate moderno korisničko iskustvo (doživljaj za korisnika). Možete da promenite vaš veb sajt kao odgovor na ponašanje korisnika, kao što je klik na neko dugme.

Poglavlje 11, „Interaktivni osluškivači sadržaja i događaja“, podiže vaše odgovore korisnicima na viši nivo. Na primer, naučićete kako da reagujete na događaje, kao što su kursorovo napuštanje polja za unos i kretanje miša korisnika.

U Poglavlju 12, „JavaScript na srednjem nivou“, razmatrane su teme koje će vam biti potrebne da biste napisali JavaScript kod na srednjem nivou, kao što su regularni izrazi, rekurzija i debagovanje, da biste poboljšali performanse vašeg koda.

U Poglavlju 13, „Konkurentnost“, predstavljena je tema konkurentnog i asinhronog programiranja koje omogućava kodu da izvršava više zadataka u isto vreme i da bude zaista fleksibilan.

Poglavlje 14, „HTML5, Canvas i JavaScript“, fokusirano je na HTML5 i JavaScript. Videćete mnogo i HTML-a i JavaScripta u prethodnim poglavljima, ali ovde ćete se fokusirati na funkcije specifične za HTML5, kao što je element canvas.

U Poglavlju 15, „Sledeći koraci“, istražujete korake koje biste mogli da preduzmete nakon što naučite sve osnovne funkcije JavaScripta i možete da pišete sjajne programe, koristeći JavaScript. Pogledaćete neke od poznatih JavaScript biblioteka i razvojnih radnih okvira (frameworks), kao što su Angular, React i Vue, i pogledaćete Node.js da biste uvideli kako se backend može napisati u JavaScriptu.

Izvucite maksimum iz ove knjige

Prethodno iskustvo u kodiranju će vam biti od pomoći, ali definitivno nije neophodno. Ako imate računar sa uređivačem teksta (kao što su Notepad ili TextEdit, a ne Word!) i pregledač, možete početi rad pomoću ove knjige. Podstičemo vas da uradite vežbe i projekte i da neprestano eksperimentišete dok budete čitali poglavlja da biste bili sigurni da razumete svaki koncept pre nego što krenete dalje.

Preuzimanje datoteka primera koda

Paket koda za ovu knjigu sa nalazi i na GitHubu na adresi https://github.com/PacktPublishing/JavaScript-from-Beginner-to-Professional. Na raspolaganju su vam i drugi paketi kodova iz našeg bogatog kataloga knjiga i video-zapisa dostupnih na adresi https://github.com/PacktPublishing/.

Preuzmite slike u boji za ovu knjigu

Takođe smo vam obezbedili PDF datoteku koja sadrži slike u boji ekrana/dijagrama upotrebljenih u ovoj knjizi. Tu datoteku možete da preuzmete sa adrese https://static.packt-cdn.com/downloads/9781800562523_ColorImages.pdf.

Korišćene konvencije

U ovoj knjizi se koristi niz konvencija.

`Kod u tekstu` - Označava kodne reči u tekstu, nazive tabela baza podataka, nazive direktorijuma, nazive datoteka, ekstenzije datoteka, nazive putanja, lažne URL adrese, korisnički unos i Twitter postove. Evo primera: „Takođe morate da ukažete pregledaču koji dokument koristite u deklaraciji `<!DOCTYPE>`.“

Blok koda je postavljen na sledeći način:

```
<html>
  <script type="text/javascript">
    alert("Hi there!");
  </script>
</html>
```

Svi unosi ili ispisi komandne linije napisani su na sledeći način:

```
console.log("Hello world!")
```

Podebljana slova - Novi termini, važne reči ili reči koje vidite na ekranu - na primer, u menijima ili okvirima za dijalog, biće prikazani u tekstu **podebljanim slovima**. Na primer: „Ako kliknete desnim tasterom miša i izaberete **Inspect** na macOS sistemima, videćete da će biti prikazan ekran, koji je sličan onom na sledećem snimku ekrana."



Upozorenja ili važne napomene se prikazuju ovako.



Saveti i trikovi se prikazuju ovako.

Stupite u kontakt

Povratne informacije naših čitalaca su uvek dobrodošle.

Opšte povratne informacije: Ako imate pitanja o bilo kojem aspektu ove knjige, pošaljite nam e-mail na adresu kombib@gmail.com.

Štamparske greške - Iako smo preduzeli sve mere da bismo obezbedili tačnost sadržaja, greške su moguće. Ako pronađete neku grešku u ovoj knjizi, bili bismo zahvalni ako biste nam to javili. Otvorite stranicu http://www.packtpub.com/support/errata i popunite obrazac.

Piraterija - Ako na Internetu pronađete ilegalne kopije naših knjiga, u bilo kojoj formi, molimo vas da nas o tome obavestite i da nam pošaljete adresu lokacije ili naziv veb sajta. Pošaljite nam poruku na adresu copyright@packt.com i link ka sumnjivom materijalu.

Ako ste zainteresovani da postanete autor - Ako postoji tema za koju ste stručni i zainteresovani ste za pisanje ili doprinos knjizi, posetite stranicu http://authors.packtpub.com.

Podelite svoje mišljenje

Bili bismo vam zahvalni ako biste nam poslali svoje mišljenje o knjizi „JavaScript od početnika do profesionalca“ nakon što je pročitate! Posetite veb stranicu ove knjige na našem sajtu i napišite komentar.

https://packt.link/r/1800562527

Vaša recenzija je važna i nama i tehničkoj zajednici i pomoći će nam da isporučujemo sadržaje visokog kvaliteta.

1

Početak rada u JavaScriptu

Izgleda da ste odlučili da počnete da učite JavaScript. To je odličan izbor! JavaScript je programski jezik koji se može koristiti i na serverskoj i na klijentskoj strani aplikacija. Serverska strana aplikacije je backend logika koja obično funkcioniše na računarima u centrima podataka i komunicira sa bazom podataka, dok je klijentska strana ono što funkcioniše na uređaju korisnika, često u pregledaču za JavaScript.

Verovatno ste koristili funkcionalnost napisanu u JavaScriptu. Ako ste koristili veb pregledače, kao što su Chrome, Firefox, Safari ili Edge, definitivno ste koristili takvu funkcionalnost. JavaScript je svuda na Vebu. Ako uđete na veb stranicu i ona od vas zatraži da prihvatite kolačiće i da kliknete na OK, iskačući prozor nestaje. To je JavaScript u akciji. A ako želite da se krećete po veb sajtu i otvori se podmeni, to je još JavaScripta u akciji. Često se, kada filtrirate proizvode u veb prodavnici, koristi JavaScript. A šta je sa četovima koji počinju da komuniciraju sa vama nakon što na nekom veb sajtu provedete određeni broj sekundi? Pogodili ste ako kažete to je JavaScript!

Skoro svaka interakcija koju imate sa veb stranicama se dešava zbog JavaScripta – na primer, dugmad na koju kliknete, rođendanske čestitke koje kreirate i proračuni koje izvršavate. Za sve što zahteva više od statične veb stranice potreban je JavaScript.

U ovom poglavlju će biti obrađene sledeće teme:

- zašto bi trebalo da naučite JavaScript
- podešavanje vašeg okruženja
- kako pregledač razume JavaScript
- upotreba konzole pregledača

- dodavanje JavaScripta na veb stranicu
- pisanje JavaScript koda



Napomena: Odgovori na vežbu, projekat i kviz za samoproveru se nalaze u „Dodatku".

Zašto bi trebalo da naučite JavaScript?

Postoji mnogo razloga zbog kojih biste želeli da naučite JavaScript. JavaScript potiče iz 1995. godine i često se smatra najčešće korišćenim programskim jezikom, zato što ga veb pregledači podržavaju i razumeju. Sve što vam je potrebno za interpretaciju JavaScripta je već instalirano na vašem računaru ako imate veb pregledač i uređivač teksta. Međutim, postoje bolja podešavanja, o kojima će biti reči kasnije u ovom poglavlju.

JavaScript je odličan programski jezik za početnike, a većina naprednih softverskih programera će bar razumeti nešto JavaScripta, jer će u nekom trenutku naići na njega. JavaScript je odličan izbor za početnike zbog više razloga. Prvi razlog je što možete početi da kreirate zaista odlične aplikacije koristeći JavaScript ranije nego što možete da zamislite. Nakon što pročitate Poglavlje 5, „Petlje", moći ćete da pišete prilično složene skriptove koji komuniciraju sa korisnicima. Kada pročitate ovu knjigu u celosti, moći ćete da pišete dinamične veb stranice koje će izvršavati razne zadatke.

JavaScript se može koristiti za pisanje mnogo različitih tipova aplikacija i skriptova. Može se koristiti za programiranje za veb pregledač, ali i logički sloj koda aplikacije koji ne možete da vidite (kao što je komunikacija sa bazom podataka) može se programirati u JavaScriptu, zajedno sa igrama i skriptovima za automatizaciju, i za mnoštvo drugih namena. JavaScript se takođe može koristiti za različite stilove programiranja - pod tim podrazumevamo načine strukturiranja i pisanja koda. Kako biste to uradili zavisi od svrhe vašeg skripta. Ako nikada ranije niste pisali kod, možda niste sasvim razumeli ove koncepte (njihovo razumevanje nije sasvim neophodno u ovoj fazi), ali JavaScript se može koristiti za (polu) objektno-orijentisano, funkcionalno i proceduralno programiranje, tj. za drugačije paradigme programiranja.

Postoji mnoštvo biblioteka i radnih okvira koje možete koristiti kada naučite osnove JavaScripta. Ove biblioteke i radni okviri će zaista poboljšati funkcionisanje vašeg softvera i olakšati i omogućiti izvršavanje više zadataka za kraće vremena. Primeri ovih odličnih biblioteka i radnog okvira su React, Vue.js, jQuery, Angular i Node.js. Ne brinite o njima za sada; samo ih posmatrajte kao bonus za kasnije. Neke od njih ćemo ukratko razmotriti na samom kraju ove knjige.

Na kraju, pomenućemo JavaScript zajednicu. JavaScript je veoma popularan programski jezik i mnogi ljudi ga koriste. Ne postoji problem za koji ne možete da pronađete rešenje na Internetu.

Zajednica JavaScripta je ogromna. Popularni forum „Stack Overflow" sadrži mnogo pomoći za sve vrste problema kodiranja i ima ogroman odeljak o JavaScriptu. Često ćete naići na taj forum dok guglate po Internetu, tražeći probleme, savete i trikove.

Ako je JavaScript vaš prvi jezik u programiranju, novi ste u celoj softverskoj zajednici i spremni ste da uživate. Softverski programeri, bez obzira na jezik koji koriste, vole da pomažu jedni drugima. Postoje forumi i tutorijali na Internetu, pa možete pronaći odgovore na skoro sva vaša pitanja. Kao početniku, može vam biti teško da razumete sve odgovore. Budite strpljivi, nastavite da se trudite i učite i ubrzo ćete razumeti JavaScript.

Podešavanje vašeg okruženja

Postoji mnogo načina na koje možete da podesite JavaScript okruženje za kodiranje. Za početak, vaš računar verovatno već ima sve osnovno što vam je potrebno da biste pisali JavaScript kod. Preporučujemo vam da malo olakšate sebi život i da koristite IDE.

Integrisano razvojno okruženje

Integrisano razvojno okruženje (Integrated Development Environment - IDE) je posebna aplikacija koja se koristi za pisanje, izvršavanje i debagovanje koda. Možete ga jednostavno otvoriti kao bilo koji program. Na primer, da biste napisali tekstualni dokument, potrebno je da otvorite program, da izaberete odgovarajuću datoteku i da počnete da pišete tekst. Kodiranje je slično. Otvarate IDE i pišete kod. IDE često ima posebno dugme za izvršavanje koda. Pritiskom na ovo dugme pokrenuće se kod iz IDE-a. Za JavaScript možda ćete morati u određenim slučajevima otvoriti vaš pregledač ručno.

Međutim, IDE radi više od toga; obično ima isticanje sintakse. To znači da će određeni elementi u vašem kodu imati određenu boju i lako ćete videti kada nešto „krene naopako". Još jedna odlična funkcija je funkcija automatskih predloga u kojoj vam uređivač pomaže da koristite opcije koje imate na mestu na kojem kodirate. To se često naziva dovršavanje koda. Mnogi IDE-ovi imaju posebne plaginove, tako da možete učiniti upotrebu drugih alatki intuitivnijom i možete im dodati funkcije - na primer, aktivno ponovno učitavanje datoteka (hod reload) u pregledaču.

Postoji mnogo IDE-ova, a razlikuju se po onome što imaju da ponude. Koristimo Visual Studio Code u celoj knjizi, ali to je naš lični izbor. Drugi popularni IDE-ovi u vreme pisanja ove knjige su Atom, Sublime Text i WebStorm.

Stalno se pojavljuje mnogo novih IDE-ova, tako da postoji velika verovatnoća da najpopularniji IDE u trenutku kada čitate ovu knjigu nije na ovoj listi. Postoji mnogo drugih opcija. Možete brzo pretražiti JavaScript IDE-ove na Vebu. Postoji nekoliko „stvari" na koje treba da obratite pažnju kada birate IDE. Uverite se da podržava isticanje sintakse, debagovanje i dovršavanje koda za JavaScript.

Veb pregledač

Takođe će vam biti potreban pregledač. Većina pregledača je savršena za JavaScript, ali je bolje da ne koristite Internet Explorer, koji ne podržava najnovije JavaScript funkcije. Dve dobre opcije bile bi Chrome i Firefox. Podržavaju najnovije JavaScript funkcije i dostupni su korisni plaginovi.

Dodatne alatke

Postoji mnogo dodatnih „stvari" koje možete da koristite tokom kodiranja - na primer, plaginove za pregledač koji će vam pomoći da otklonite greške ili pojednostavite nešto. U ovom trenutku vam nije potreban ni jedan od njih, ali obratite pažnju kad god naiđete na alatku za koju su drugi veoma zainteresovani.

Uređivač na Internetu

Možda nemate pristup računaru, već možda samo tabletu, ili ne možete ništa da instalirate na svom laptopu. Postoje i odlični uređivači na Internetu za ove scenarije. Nećemo imenovati ni jedan, jer se brzo razvijaju i verovatno će biti zastareli kada ovu knjigu budete čitali. Međutim, ako na Vebu pretražite `online JavaScript IDE`, naći ćete mnogo opcija na Internetu u kojima možete jednostavno da počnete da kodirate JavaScript i da pritisnete dugme da biste ga pokrenuli.

Kako pregledač razume JavaScript?

JavaScript je interpretirani jezik, što znači da ga računar razume dok ga izvršava. Neki jezici se obrađuju pre izvršavanja, a to se zove kompajliranje, ali JavaScript to ne radi. Računar može samo da interpretira JavaScript „u hodu". „Mašina" koja razume JavaScript će se ovde zvati interpreter.

Veb stranica nije samo JavaScript. Veb stranice su napisane na tri jezika: na HTML-u, CSS-u i JavaScriptu.

HTML utvrđuje šta se nalazi na stranici; sadržaj stranice je u HTML-u. Ako na stranici postoji pasus, HTML stranice sadrže pasus. A ako postoji naslov, to znači da je HTML korišćen za dodavanje naslova i drugih „stvari". HTML se sastoji od elemenata koji se nazivaju i tagovi. Oni utvrđuju šta se nalazi na stranici. Evo malog primera koji će kreirati veb stranicu sa tekstom `Hello World`:

```
<html>
  <body>
    Hello world!
  </body>
</html>
```

Ne brinite ako nikada niste videli HTML. U Poglavlju 9, „Objektni model dokumenta", postoji kratak kurs o njemu.

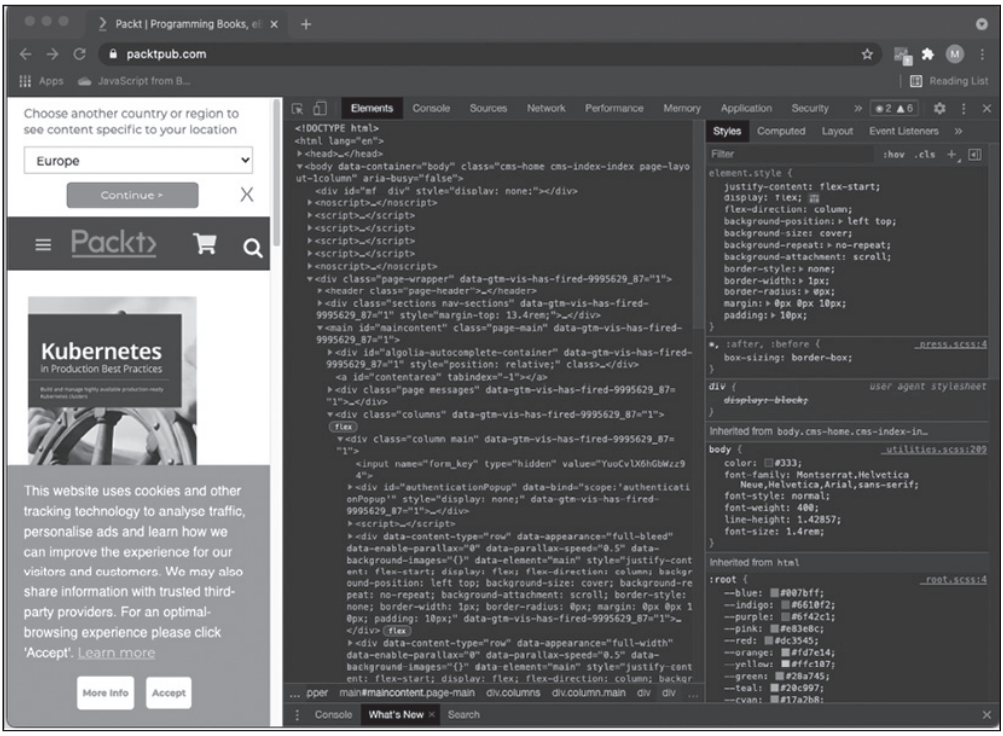
CSS je izgled veb stranice. Na primer, ako je tekst plav, to znači da je obojen pomoću CSS-a. Veličinu fonta, porodicu fontova i poziciju na stranici određuje CSS. JavaScript je poslednji deo „slagalice", koji definiše šta veb stranica može da uradi i kako može da komunicira sa korisnikom ili backendom.

Kada koristite JavaScript, pre ili kasnije ćete naići na termin **ECMAScript**. Ovo je specifikacija ili standardizacija za JavaScript jezik. Trenutni standard je **ECMAScript 6** (koji se naziva i **ES6**). Pregledači koriste ovu specifikaciju da podrže JavaScript, ali koriste i **objektni model dokumenta (DOM)**, koji ćete videti kasnije. JavaScript ima mnogo implementacija koje se mogu neznatno razlikovati, ali ECMAScript se može smatrati osnovnom specifikacijom koju će JavaScript implementacija definitivno koristiti.

Upotreba konzole pregledača

Veb pregledači imaju ugrađenu opciju da bi „videli" kod koji omogućava veb stranicu na kojoj se nalazite. Ako pritisnete F12 na Windows računaru dok ste u veb pregledaču ili kliknete desnim tasterom miša i izaberete **Inspect** na macOS sistemima, biće prikazan ekran koji je sličan onom na sledećem snimku ekrana.

Možda će to izgledati malo drugačije u pregledaču na vašem računaru, ali klik desnim tasterom miša i odabir opcije **Inspect** generalno mogu da funkcionišu.



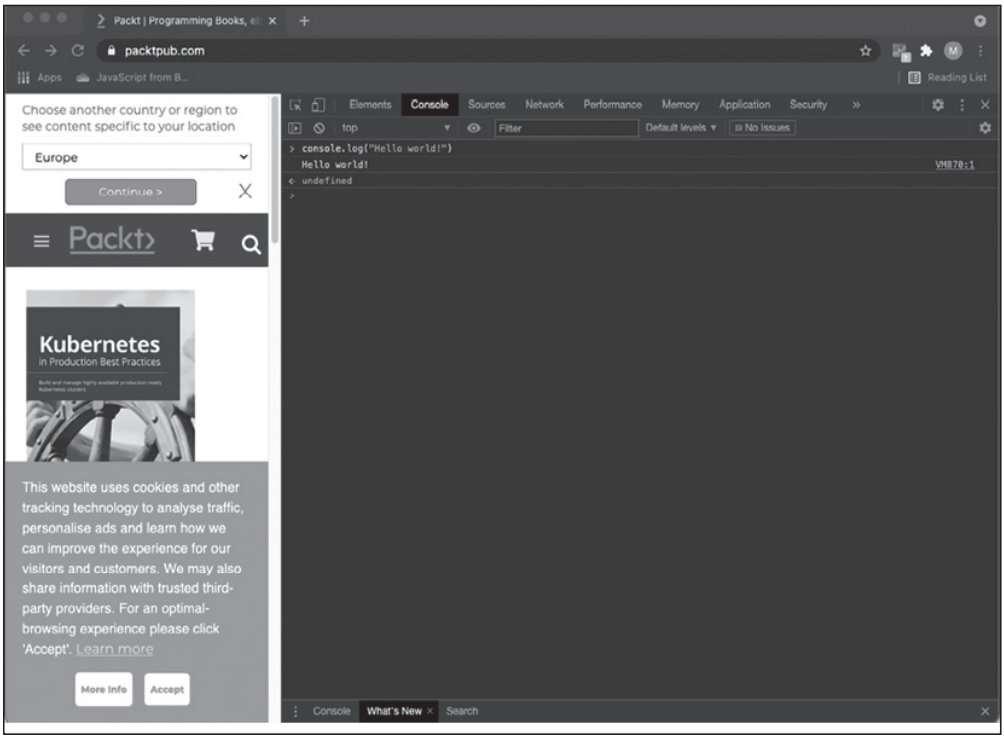
Slika 1.1 Konzola pregledača na veb sajtu Packt

Ovaj snimak ekrana sadrži više kartica na vrhu. Sada gledamo kartice elemenata koje sadrže sav HTML i CSS (sećate li se njih?). Ako kliknete na karticu konzole, naći ćete na dnu panela mesto na kojem možete direktno da umetnete neki kod. Možda ćete videti neka upozorenja ili poruke o grešci na ovoj kartici - one nisu neuobičajene i ne brinite o njima ukoliko stranica funkcioniše.

Programeri koriste konzolu da bi evidentirali šta se dešava i da bi izvršili debagovanje. Debagovanje je pronalaženje problema kada aplikacija ne prikazuje željeno ponašanje. Konzola daje neke uvide šta se dešava ako evidentirate razumne poruke. Ovo je, zapravo, prva komanda koju ćete naučiti:

```
console.log("Hello world!");
```

Kada kliknete na ovu karticu konzole, unesite ovaj prvi JavaScript kod, a zatim pritisnite *Enter*. Biće prikazan ispis koda u konzoli. Izgledaće kao na sledećem snimku ekrana.



Slika 1.2 JavaScript u konzoli pregledača

Koristićete iskaz `console.log()` u velikom delu cele ove knjige u vašem kodu da biste testirali vaše isečke koda i videli rezultate. Postoje i drugi metodi konzole – na primer, metod `console.table()`, koji kreira tabelu kada se uneti podaci mogu predstaviti u obliku tabele. Drugi metod konzole je `console.error()`, koji će evidentirati unete podatke, ali pomoću stila koji ukazuje na grešku.

Praktična vežba 1.1

Rad u konzoli:

1. Otvorite konzolu pregledača, otkucajte `4 + 10` i pritisnite *Enter*. Šta vidite kao odgovor?
2. Koristite sintaksu `console.log()`, tako što ćete staviti vrednost u zaobljene zagrade. Pokušajte da unesete svoje ime sa navodnicima (da biste ukazali na tekstualni znakovni niz, o kojem će biti reči u sledećem poglavlju).

Dodavanje JavaScripta na veb stranicu

Postoje dva načina na koja možete da povežete JavaScript sa veb stranicom. Prvi način je da ukucate JavaScript direktno u HTML između dva taga `<script>`. U HTML-u prvi tag `<script>` treba da deklariše da će sledeći skript biti izvršen. A sadržaj bi trebalo da bude unutar ovog elementa. Zatim, zatvarate skript pomoću istog taga, kome prethodi kosa crta `</script>`. Ili možete da povežete JavaScript datoteku sa HTML datotekom, koristeći tag script na vrhu HTML stranice.

Direktno u HTML-u

Evo primera kako da napišete veoma jednostavnu veb stranicu na kojoj će se pojaviti iskačući okvir sa natpisom `Hi there!`:

```
<html>
  <script type="text/javascript">
    alert("Hi there!");
  </script>
</html>
```

Ako ovaj kod sačuvate kao `.html` datoteku i otvorite tu datoteku u vašem pregledaču, dobićete nešto poput sledećeg snimka ekrana. To ćete sačuvati kao `Hi.html`.



Slika 1.3 JavaScript je kreirao ovaj iskačući prozor sa tekstom „Hi there!“.

Komanda `alert` će kreirati iskačući prozor koji sadrži poruku. Ova poruka je navedena između zagrada iza upozorenja.

Trenutno se sadržaj nalazi direktno unutar tagova `<html>`. To nije najbolja praksa. Moraćete da kreirate dva elementa unutar tagova `<html>`—`<head>` i `<body>`. U elementu `head` upisujete metapodatke i kasnije koristite ovaj deo da povežete spoljne datoteke sa vašom HTML datotekom. U „telu“ se nalazi sadržaj veb stranice.

Takođe morate da ukažete pregledaču na dokument koji koristite u deklaraciji `<!DOCTYPE>`. Pošto pišete JavaScript unutar HTML datoteke, morate da koristite `<!DOCT!YPE html>`. Evo primera:

```
<!DOCTYPE html>
<html>

<head>
  <title>This goes in the tab of your browser</title>
</head>

<body>
The content of the webpage
  <script>
    console.log("Hi there!");
  </script>
</body>

</html>
```

Ovaj primer veb stranice će prikazati tekst `The content of the webpage`. Ako pogledate u konzolu pregledača, naići ćete na iznenađenje! Takođe je izvršen JavaScript i evidentiran tekst `Hello!` u konzoli.

Praktična vežba 1.2

JavaScript na HTML stranici:

1. Otvorite uređivač koda i kreirajte HTML datoteku.
2. U okviru vaše HTML datoteke postavite HTML tagove, tip dokumenta, HTML, naslov i „telo“, a zatim dodajte tagove script.
3. Postavite neki JavaScript kod u tagovima script. Možete koristiti `log („hello world!“).`

Povezivanje spoljne datoteke sa veb stranicom

Takođe možete povezati spoljnu datoteku sa HTML datotekom. To se smatra boljom praksom, jer je kod bolje organizovan i možete izbeći veoma dugačke HTML stranice zbog JavaScripta. Osim toga, možete ponovo da koristite JavaScript na drugim veb stranicama vašeg veb sajta, bez potrebe za kopiranjem i „lepljenjem“. Ako imate isti JavaScript na 10 stranica i morate da promenite skript, morali biste da promenite samo jednu datoteku ako biste promenili skript na način koji vam prikazujemo u ovom primeru.

Zasebne JavaScript datoteke imaju sufiks .js. Prvo ćete kreirati jednu takvu datoteku, koja će se zvati ch1_alert.js. Ovo će biti njen sadržaj:

```
alert("Saying hi from a different file!");
```

Zatim ćete kreirati posebnu HTML datoteku (ponovo koristeći sufiks `.html`). I dodaćete joj sledeći sadržaj:

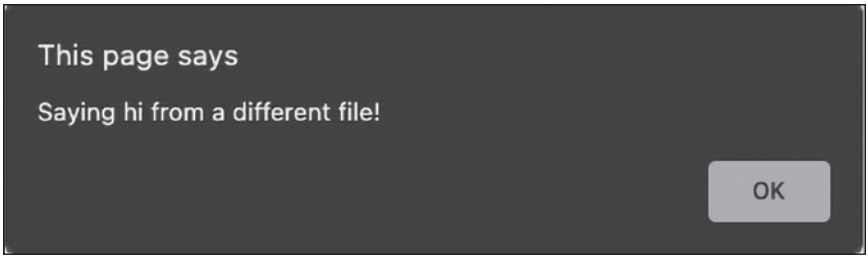
```
<html>
  <script type="text/javascript" src="ch1_alert.js"></script>
</html>
```

Pobrinite se da stavite datoteke na istu lokaciju ili da navedete putanju JavaScript datoteke u vašem HTML-u. Nazivi razlikuju velika i mala slova i trebalo bi da se tačno podudaraju.

Imate dve opcije. Možete koristiti relativnu i apsolutnu putanju. Hajde da prvo razmotrimo apsolutnu putanju, jer je nju najlakše objasniti. Vaš računar ima root. Za Linux i macOS je `/`, a za Windows je često `C:/`. Putanja datoteke koja počinje od roota je apsolutna putanja. Ovo je najlakše dodati, jer će funkcionisati na vašem uređaju. Međutim, postoji „caka“ - ako se na vašem uređaju ova fascikla veb sajta kasnije premesti na server, apsolutna putanja više neće funkcionisati.

Druga, bezbednija opcija je relativna putanja. Vi određujete putanju iz datoteke u kojoj se trenutno nalazite. Dakle, ako je datoteka u istoj fascikli, moraćete samo da unesete naziv. Ako se datoteka nalazi u fascikli „example“ koja se nalazi unutar fascikle u kojoj je vaša datoteka, moraćete da navedete `example/nameOfTheFile.js`. A ako je direktorijum gore, moraćete da navedete `../nameOfTheFile.js`.

Ako otvorite HTML datoteku, trebalo bi da dobijete rezultat kao na sledećoj slici.



Slika 1.4 Iskačući prozor koji je kreirao JavaScript u drugoj datoteci

Praktična vežba 1.3

Izvršite povezivanje sa JS JavaScript datotekom:

1. Kreirajte zasebnu datoteku `app` sa ekstenzijom `.js`.
2. Unutar `.js` datoteke dodajte neki JavaScript kod.
3. Povežite zasebnu `.js` datoteku unutar HTML datoteke koju ste kreirali u *praktičnoj vežbi 1.2*.
4. Otvorite HTML datoteku u vašem pregledaču i proverite da li je JavaScript kod pravilno izvršen.

Pisanje JavaScript koda

Dakle, sada imamo mnogo konteksta, ali kako, zapravo, pišete JavaScript kod? Treba imati na umu neke važne „stvari" – način na koji se formatira kod, upotrebu odgovarajućeg nivoa uvlačenja, upotrebu znaka tačka-zarez i dodavanje komentara. Prvo ćemo razmotriti formatiranje koda.

Formatiranje koda

Kod mora biti dobro formatiran. Ako imate dugačku datoteku sa mnogo linija koda i niste se pridržavali nekoliko osnovnih pravila formatiranja, biće teško da se razume šta ste napisali. Dakle, koja su osnovna pravila formatiranja? Za sada su najvažnija uvlačenje i znak tačka-zarez. Postoje i konvencije o imenovanju, ali one će biti obrađene za svaku sledeću temu zasebno.

Uvlačenja i razmak

Kada pišete kod, često linija koda pripada određenom bloku koda (kodu između dve vitičaste zagrade `{ ovako}`) ili nadređenom iskazu. U tom slučaju kodu u tom bloku dodajte jedno uvlačenje da biste bili sigurni da možete lako da vidite šta je deo bloka i kada počinje novi blok. Ne morate da razumete sledeći isečak koda, ali smo u njemu prikazali čitljivost sa uvlačenjema i bez njega.

Bez novih redova:

```
let status = "new"; let scared = true; if (status === "new") { console.
log("Welcome to JavaScript!");  if (scared) { console.log("Don't worry
you will be fine!"); } else { console.log("You're brave! You are going
to do great!"); } } else { console.log("Welcome back, I knew you'd like
it!"); }
```

Sa novim redovima, ali bez uvlačenja:

```
let status = "new";
let scared = true;
if (status === "new") {
console.log("Welcome to JavaScript!");
if (scared) {
console.log("Don't worry you will be fine!");
} else {
console.log("You're brave! You are going to do great!");
}
} else {
console.log("Welcome back, I knew you'd like it!");
}
```

Sa novim redovima i uvlačenjem:

```
let status = "new";
let scared = true;
if (status === "new") {
  console.log("Welcome to JavaScript!");
  if (scared) {
    console.log("Don't worry you will be fine!");
  } else {
    console.log("You're brave! You are going to do great!");
  }
} else {
  console.log("Welcome back, I knew you'd like it!");
}
```

Sada možete lako da vidite kada se blokovi koda završavaju. Ovo je mesto na kojem blok `if` ima odgovarajuću zagradu `}` na istom nivou uvlačenja. U primeru bez uvlačenja morali biste da prebrojite zagrade da biste utvrdili kada će se blok `if` završiti. Iako uvlačenje nije neophodno za rad koda, pobrinite se da ga dobro koristite. Kasnije ćete biti zahvalni sami sebi.

Znak tačka-zarez

Posle svakog iskaza treba da umetnete znak tačka-zarez. JavaScript je veoma „zahvalan“ i razumeće mnoge situacije u kojima ste zaboravili jedan znak tačka-zarez, ali naviknete se na vreme da dodajete jedan znak posle svake linije koda. Kada deklarišete blok koda, kao što su iskaz `if` ili petlja, ne bi trebalo da ga završavate znakom tačka-zarez. Koristite taj znak samo za odvojene iskaze.

Komentari koda

Pomoću komentara možete ukazati interpreteru da treba da ignoriše neke linije datoteke. Komentari neće biti izvršeni. Često je korisno da izbegnete izvršavanje nekog dela datoteke zbog sledećih razloga:

1. Ne želite da izvršite neki deo koda dok izvršavate skript, pa ga komentarišete kako bi ga interpreter ignorisao.
2. Ne želite da izvršite metapodatke, tj. dodavanje konteksta kodu, kao što su ime autora i opis onoga što datoteka obuhvata.
3. Ne želite da izvršite dodavanje komentara određenim delovima koda da biste objasnili šta se dešava ili zašto je napravljen određeni izbor.

Postoje dva načina za pisanje komentara. Možete pisati jednolinijske ili višelinijske komentare. Evo primera:

```
// I'm a single line comment
// console.log("single line comment, not logged");

/* I'm a multi-line comment. Whatever is between the slash asterisk and
the asterisk slash will not get executed.
console.log("I'm not logged, because I'm a comment");
*/
```

U prethodnom isečku koda vidite oba stila komentarisanja. Prvi je jednolinijski. To takođe može biti umetnuti komentar na kraju linije. Šta god sledi posle znakova `//` u liniji biće ignorisano. Drugi je višelinijski; piše se tako što počinje znakovima `/*` i završava se znakovima `*/`.

Praktična vežba 1.4

Dodavanje komentara:

1. Dodajte novi iskaz vašem JavaScript kodu postavljanjem vrednosti promenljive. Pošto ćemo postavljenje vrednosti razmotriti u sledećem poglavlju, možete koristiti sledeću liniju:

   ```
   let a = 10;
   ```

2. Dodajte komentar na kraju iskaza koji pokazuje da ste postavili vrednost `10`.
3. Odštampajte vrednost, koristeći funkciju `console.log()`. Dodajte komentar koji objašnjava šta će ta funkcija da radi.

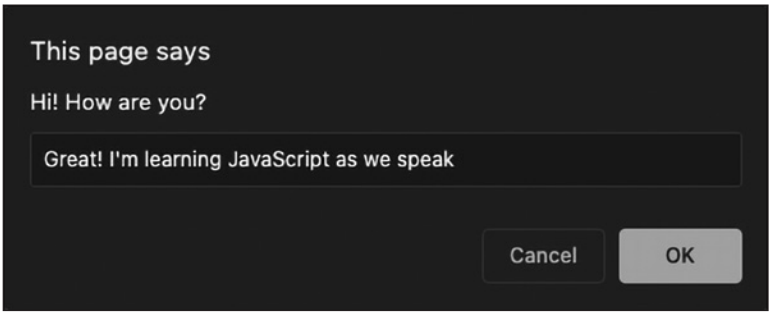
4. Na kraju vašeg JavaScript koda koristite višelinijski komentar. U stvarnom proizvodnom skriptu možete da u tom komentaru dodate kratak opis namene datoteke.

Komandna linija

Želimo da vam pokažemo i komandnu liniju. Ona funkcioniše kao upozorenje, ali prihvata korisnički unos. Ubrzo ćete naučiti kako da skladištite promenljive, a kada to naučite, možete da sačuvate rezultat ove funkcije prompt i da iskoristite taj rezultat. Samo napred i promenite `alert()` u `prompt()` u datoteci `Hi.html`, kao na sledećoj slici.

```
prompt("Hi! How are you?");
```

Zatim nastavite dalje i osvežite HTML. Biće prikazan iskačući prozor sa poljem za unos u kojem možete da unesete tekst, na sledeći način:



Slika 1.5 Stranica sa odzivnikom za korisnički unos

Vrednost koju vi (ili bilo koji drugi korisnik) unesete biće vraćena u skript i može se koristiti u vašem kodu! Ovo je odlično za dobijanje korisničkog unosa da biste oblikovali način na koji vaš kod funkcioniše.

Slučajni brojevi

U svrhu zabavnih vežbi u prethodnim poglavljima ove knjige, želeli bismo da znate kako da generišete slučajni broj u JavaScriptu. Sasvim je u redu ako još uvek ne razumete šta se dešava; samo znajte da je ovo komanda za kreiranje slučajnog broja:

```
Math.random();
```

Možete da uneste vrednost u konzolu i da vidite rezultat ako ga evidentirate:

```
console.log(Math.random());
```

Ovaj broj će biti decimalni broj između 0 i 1. Ako želite broj između 0 i 100, možete ga pomnožiti brojem 100 ovako:

```
console.log(Math.random() * 100);
```



Ne brinite - matematičke operatore ćemo razmotriti u Poglavlju 2, „Osnove JavaScripta“.

Ako ne želite da rezultat bude decimalni broj, možete primeniti funkciju `Math.floor` na njemu, koja ga zaokružuje na najbliži ceo broj:

```
console.log(Math.floor(Math.random() * 100));
```

Ne brinite ako još ne shvatate ovaj metod. On će biti detaljnije objašnjen kasnije u ovoj knjizi. U Poglavlju 8, „Ugrađeni JavaScript metodi“, detaljnije će biti reči o ugrađenim metodima. Do tada, samo nam verujte da prethodni kod generiše slučajni broj između 0 i 100.

Projekat poglavlja

Kreiranje HTML datoteke i povezane JavaScript datoteke

Kreirajte HTML datoteku i zasebnu JavaScript datoteku. Zatim se povežite sa JavaScript datotekom iz HTML datoteke.

1. U JavaScript datoteci unesite svoje ime u konzolu i dodajte višelinijski komentar vašem kodu.
2. Pokušajte da komentarišete poruku konzole u vašoj JavaScript datoteci, tako da se ništa ne prikazuje u konzoli.

Kviz za samoproveru

1. Koja je HTML sintaksa za dodavanje spoljne JavaScript datoteke?
2. Da li možete da pokrenete JavaScript u datoteci sa JS ekstenzijom u vašem pregledaču?
3. Kako se piše višelinijski komentar u JavaScriptu?
4. Koji je najbolji način da uklonite liniju koda iz izvršavanja, koju biste možda želeli da zadržite dok debagujete?

Rezime

Bravo - počeli ste da koristite JavaScript! U ovom poglavlju je razmatrano mnogo konteksta, koje ćete morati da znate pre nego što počnete da kodirate u JavaScriptu. Videli ste da možete koristiti JavaScript za mnoge svrhe, a jedan od najpopularnijih slučajeva korišćenja je na Vebu. U pregledačima se može koristiti JavaScript, jer imaju specijalan deo koji se zove interpreter, a koji može da obrađuje JavaScript. Videli ste da postoji više opcija za pisanje JavaScripta na računaru. Biće vam potreban IDE, program koji možete koristiti za pisanje i izvršavanje koda.

Dodavanje JavaScripta na veb stranicu može se izvršiti na nekoliko načina. Videli ste kako da ga uključite u element script i kako da dodate zasebnu JavaScript datoteku na stranicu. Završili smo ovo poglavlje nekim važnim opštim napomenama kako napisati dobro strukturiran i čitljiv kod i kod koji je jednostavan za održavanje i dobro dokumentovan komentarima. Takođe ste videli da možete upisivati u konzolu pomoću metoda `console.log()` i zatražiti unos od korisnika pomoću metoda `prompt()`. Na kraju, takođe ste videli da možete da generišete slučajne brojeve pomoću funkcije `Math.random()`.

U sledećem poglavlju ćete upoznati JavaScript osnovne tipove podataka i operatore kojima možete da manipulišete.