

Uvod u JavaScript jezik

HTML, CSS i JavaScript su osnovne tehnologije frontend razvoja. Ovi jezici se nazivaju osnovni, zato što su web pregledači u stanju da direktno razumeju njihov kod.

HTML i CSS su opisni jezici koji se koriste za to da definišu sadržaj stranice i njenu vizuelnu komponentu. Reč je o jezicima koji se ne mogu nazvati programskim. Njima nedostaju različiti, veoma značajni jezički elementi, koji programskim jezicima omogućavaju formulisanje programske logike – tipove, strukture, nizove, operatore, petlje, grananja itd.

Za razliku od dva upravo spomenuta jezika, JavaScript je programski jezik koji se može koristiti prilikom kreiranja web sajtova.

ŠTA JE JAVASCRIPT?

JavaScript je programski jezik web stranica. On je namenjen klijentskom programiranju na webu, odnosno reč je o jeziku koji dozvoljava formulisanje programske logike prilikom razvoja web sajtova.

Ovaj jezik je moguće koristiti za realizaciju različitih ponašanja. U osnovi je to jezik kojim je web sajtove moguće učiniti dinamičnijim. Parcijalno ažuriranje sadržaja web stranica, animiranje galerija slika, interaktivne mape, animacija 2D ili 3D grafike, web igre, samo su neki od pojmova koji gotovo sigurno podrazumevaju upotrebu JavaScript jezika.

Izvorni kod napisan JavaScript jezikom izvršava web pregledač. S obzirom na to da postoji veliki broj pregledača i da je implementacija jezgra za izvršavanje JavaScript koda prepuštena njihovim proizvođačima, jasno je da je neophodno postojanje određenog standarda kada su u pitanju sintaksna pravila jezika. U slučaju JavaScript jezika, reč je o standardu koji se naziva ECMAScript.

PRVI JAVASCRIPT PRIMER

Nakon uvodnog izlaganja o JavaScript jeziku, vreme je da se vidi jedan praktičan primer koji ilustruje jednu od mogućnosti JavaScript jezika

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>First JS Example</title> </head>
<body>

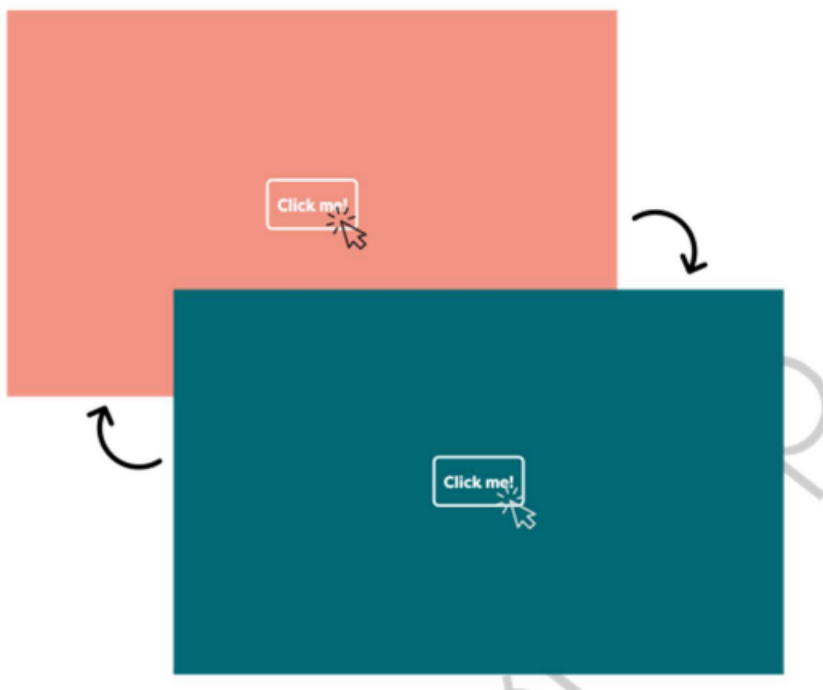
  <div id="main-container">
    <button id="button" type="button">Click Me!</button>
  </div>
```

```
<script type="text/javascript">
document.getElementById ("button") .addEventListener ("click", toggleBgColor);
var counter = 0;
function toggleBgColor () (
counter++;

if (counter & 2 == 1) {
document.getElementById ("main-
container").style.backgroundColor = "#026873";
}
else document.getElementById ("main-
container"). style.backgroundColor = "#F29484";
</ script>
</body>
</html>
```

Struktura prikazanog HTML dokumenta veoma je jednostavna. On poseduje jedan div element unutar koga se nalazi jedan button. Ovakvom dokumentu biće dodat JavaScript kod, koji će omogućiti promenu boje pozadine div elementa, klikom na button element.

JavaScript kod je napisan između specijalnih HTML tagova za unos skript koda `<script>` `</script>`. Ovakav script element sa kodom dovoljno je postaviti kao poslednji element unutar body elementa.



Klikom na dugme *Click me!* Pozadina div elementa naizmenično se menja između dve predefinisane boje.

Prikazani primer je kratka demonstracija samo jedne od mogućnosti JavaScript jezika. Unutar prikazanog koda iskorišćen je veliki broj različitih JavaScript jezičkih elemenata, koji su, sasvim razumljivo, u ovom trenutku nama nepoznati. Stoga, kako biste u potpunosti mogli da razumete kako funkcioniše JavaScript kod kojim se dobija prikazano ponašanje neophodno je upoznati se sa osnovnim postulatima JavaScript jezika, što će biti predmet lekcija koje slede.

Za početak, neophodno je dati odgovor na dva veoma važna pitanja:

- *Gde pisati JavaScript kod?*
- *Kako izgleda osnovna leksička struktura JavaScript jezika?*

Gde pisati JavaScript?

Pre početka samog čina pisanja JavaScript koda, potrebno je znati gde je takav kod moguće smestiti. Lekcija pred vama odgovoriće na ovo pitanje. Biće prikazano nekoliko različitih načina za implementaciju JavaScript koda u HTML.

Slično kao i kod CSS-a, i JavaScript kod se može pisati na nekoliko različitih lokacija.

Generalno, postoje tri načina za uključivanje JavaScript koda u jedan HTML dokument:

- Implementacija JavaScript koda između tagova `<script>` i `</script>`
- Uključivanje JavaScript koda sa eksternog izvora
- Implementacija JavaScript koda kroz proceduru za obradu događaja

U nastavku će biti demonstrirani upravo navedeni različiti načini za implementaciju JavaScript koda u HTML dokument.

INTERNI JAVASCRIPT KOD

Da bi se JavaScript kod implementirao direktno unutar HTML dokumenta, potrebno je naglasiti njegov početak i kraj, navođenjem tagova:

```
<script>
// Java Script code
</script>
```

Nekada, u starijim verzijama HTML jezika, prilikom definisanja *script* elementa bilo je neophodno navesti i *type* atribut sa vrednošću *text/javascript* kako bi se naglasilo da je reč o kodu koji je napisan JavaScript jezikom.

Navođenje *type* atributa na *script* elementu danas nije neophodno, pošto je JavaScript podrazumevani skriptni jezik weba.

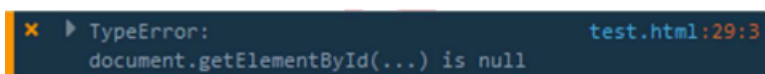
POZICIONIRANJE SCRIPT ELEMENTA

Prilikom korišćenja *script* elementa za integraciju JavaScript koda, programer ima potpunu slobodu kada je u pitanju njegovo pozicioniranje unutar dokumenta. Ipak, veoma je bitno znati da pozicioniranje *script* elementa može uticati na ponašanje JavaScript koda.

Obično se *script* element sa JavaScript kodom smešta na kraj *body* elementa, ali to nije pravilo.

Obzirom da HTML dokument sadrži brojne elemente, u takvoj situaciji je veoma bitno gde se piše *script* element sa JavaScript kodom. Naime, browser obavlja čitanje stranice od vrha ka dnu. Onoga trenutka kada prilikom parsiranja stranice browser dođe do JavaScript koda, on će obaviti njegovo prevođenje i izvršavanje.

Za demonstriranje efekta postavljanja JavaScript koda na različite pozicije unutar HTML dokumenta može poslužiti primer iz prethodne lekcije. Ukoliko se u ovom primeru *script* element sa JavaScript kodom navede na početku *body* dela, pre ostalih HTML elemenata, prilikom njegovog izvršavanja doći će do greške.



```
✖ ▶ TypeError: document.getElementById(...) is null test.html:29:3
```

Greška sa slike jasno stavlja do znanja da JavaScript kod nije uspeo da pronađe traženi HTML element. Razlog je i više nego očigledan. S obzirom na to da se stranica parsira od vrha ka dnu, u trenutku izvršavanja JavaScript koda pregledač još nije izvršio parsiranje HTML elemenata, te stoga JavaScript kod nije u mogućnosti da pronađe element po id vrednosti.

Jedan HTML dokument nije ograničen na jedan *script* element sa JavaScript kodom. Tako je u jednom HTML dokumentu moguće imati proizvoljan broj *script* elemenata.

EKSTERNI JAVASCRIPT KOD

JavaScript kod moguće je pisati i izvan HTML dokumenata, unutar zasebnih fajlova sa ekstenzijom **.js**. Na taj način se postiže bolja razdvojenost različitih slojeva web aplikacija. Naime, prilikom razvoja aplikacija bilo kog tipa, nastoji se postići što bolja razdvojenost različitih slojeva aplikacije. Kada je u pitanju frontend programiranje, ti slojevi se mogu podeliti na:

- Strukturni sloj – HTML elementi
- Sloj stilizacije – CSS stilovi
- Sloj klijentske programske logike – JavaScript kod

Na frontendu svaki od navedenih slojeva poseduje svoj tip fajla. HTML kod se smešta u fajlove sa **.html** ekstenzijom, stilizacija u fajlove sa **.css** ekstenzijom, a programska logika u **.js** fajlove.

Kako bi se JS kod iz zasebnog **.js** fajla dodao nekom HTML dokumentu, koristi se već viđeni *script* element. Ovaj element poseduje atribut *src*, kojim je moguće definisati URL adresu fajla koji sadrži JavaScript kod koji je potrebno uključiti u HTML dokument. Tako se uključivanje

eksternog fajla sa JavaScript kodom u HTML dokument može obaviti veoma jednostavno, korišćenjem samo jedne linije koda.

Korišćenje eksternih JavaScript fajlova nije dobro rešenje ukoliko postoji mala količina JavaScript koda koji se koristi samo na jednoj stranici. U takvom slučaju pregledač će napraviti jedan HTTP zahtev više, s obzirom da će, pored *.html* fajla, morati da učitati još jedan dodatni fajl sa JavaScript kodom.

U takvim situacijama bolje je JavaScript kod napisati u *<script>* tagovima na onim delovima stranice gde je to potrebno. Ipak, korišćenje eksternih JavaScript fajlova može imati brojne, izuzetno značajne prednosti:

- Razdvajanje velike količine JavaScript koda od HTML-a
- Korišćenje istog JavaScript koda na više stranica
- Eksterni JavaScript kod lako će keširati browser, te na taj način nije neophodno svaki put iznova njega učitavati
- Korišćenje eksternih JavaScript fajlova sa drugih servera

INLINE JAVASCRIPT – implementacija JavaScript koda kroz proceduru za obradu događaja

Poslednji način na koji je moguće uključiti JavaScript koda u stranicu veoma je sličan uključivanju linijske stilizacije u HTML dokument. Naime, na HTML elementima je moguće definisati slušač događaja (engl. event listener), koji zatim aktivira određeni JavaScript kod kada dođe do pojave nekog događaja. Sledeći primer to ilustruje:

```
<p onclick="alert('Hello!'); alert('Hello again!')"> Click me </p>
```

U primeru, na *p* elementu, definisan je atribut **onclick**, a kao njegova vrednost je naveden JavaScript kod. Na ovaj način, kada korisnik klikne na ovakav *p* element, pojaviće se dva popup prozora, jedan za drugim.

Napomena: Linijski JavaScript kod u HTML dokumentu je potrebno maksimalno izbegavati, s obzirom na to da otežava preglednost i testiranje. Takođe, pretplatu na različite događaje moguće je postići na mnogo elegantniji način.