

Gde pisati JavaScript?

Pre početka samog čina pisanja JavaScript koda, potrebno je znati gde je takav kôd 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 kôd 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 kôd

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

```
<script>
    //JavaScript 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:

```
<script type="text/javascript">
    //JavaScript code
</script>
```

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.

Sledeći primer ilustruje script element sa JavaScript kodom u strukturi jedne HTML stranice:

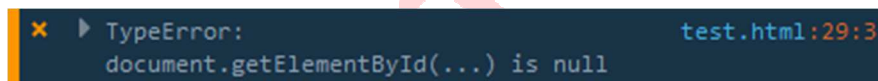
```
<!DOCTYPE html>
<html>
<head>
<title>First page</title>
</head>
<body>

    <script type="text/javascript">
        alert('Hello!');
    </script>

</body>
</html>
```

U prikazanom primeru, script element je jedini element unutar body dela. Ipak, u praksi će HTML dokument sadržati brojne elemente i 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 (slika 1.4).



Slika 2.1. Greška prilikom izvršavanja JavaScript koda

Greška sa slike 2.1. jasno stavlja do znanja da JavaScript kôd 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 kôd 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:

```
<head>
    <script type="text/javascript" >
        document.write("JavaScript - HEAD");
    </script>
</head>

<body>
    <h1>Some text 1</h1>

    <script type="text/javascript" >
```

```

        document.write("JavaScript - BODY 1");
    </script>

    <p>Some text 2</p>

    <script type="text/javascript" >
        document.write("JavaScript - BODY 2");
    </script>

    <p>End of Document</p>
</body>

```

Primer ilustruje HTML dokument sa tri `script` elementa. Iz primera se može videti da je `script` element moguće definisati i unutar `head` sekcije. Ipak, potrebno je znati da će takav kôd biti izvršen pre parsiranja `body` dela, pa samim tim i pre prikaza stranice, što će na kraju produžiti vreme koje je potrebno da protekne pre nego što korisnik unutar svog pregledača ugleda sadržaj HTML dokumenta.

Eksterni JavaScript kôd

JavaScript kôd 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 kôd.

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

Kako bi se JS kôd 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 kôd 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:

```

<head>
    <title>Page 1</title>
</head>
<body>
    <script type="text/javascript" src="popup.js"></script>
</body>

```

U primeru je unutar `body` dela definisan `script` element sa `src` atributom, čija vrednost ukazuje na putanju do fajla koji sadrži JavaScript kôd.

Takav fajl može imati sledeći sadržaj:

```
alert('Hello');
```

Kao što se može primetiti, eksterni fajl sa JavaScript kodom poseduje samo jednu liniju koda, i pri tome, kôd nije potrebno smeštati unutar bilo kakvih blokova.

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 na to da će, pored .html fajla, morati da učita još jedan dodatni fajl sa JavaScript kodom.

U takvim situacijama bolje je JavaScript kôd 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 kôd lako će keširati browser, te na taj način nije neophodno svaki put iznova njega učitavati,
- korišćenje eksternih JavaScript fajlova sa drugih servera.

Pitanje

Osnovni HTML element koji se koristi za definisanje kako internog, tako i eksternog JavaScripta, jeste:

- **script**
- style
- body
- js

Objašnjenje:

Ukoliko je potrebno definisati interni JavaScript kôd, koristi se HTML element `script`, između čijih otvarajućih i zatvarajućih tagova se definiše kôd. Isti element se koristi i za referenciranje eksternog JavaScripta, što se obavlja uz pomoć atributa `src`.

Inline JavaScript – implementacija JavaScript koda kroz proceduru za obradu događaja

Poslednji način na koji je moguće uključiti JavaScript kôd u stranicu veoma je sličan uključivanju linijske stilizacije u HTML dokument. Naime, na HTML elementima je moguće definisati slušač događaja (*event listener*, *engleski.*), koji zatim aktivira određeni JavaScript kôd 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 kôd. Na ovaj način, kada korisnik klikne na ovakav `p` element, pojaviće se da dva popup prozora, jedan za drugim.

Napomena

Linijski (engl. inline) JavaScript kôd 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.

Rezime

- Postoje tri načina za uključivanje JavaScript koda u jedan HTML dokument, te tako postoje i tri vrste JavaScripta: interni, eksterni i linijski.
- Interni JavaScript je onaj JavaScript kôd koji se piše direktno unutar HTML dokumenata.
- Da bi se JavaScript kôd implementirao direktno unutar HTML dokumenta, potrebno je naglasiti njegov početak i kraj, navođenjem tagova `<script>` i `</script>`.
- Jedan HTML dokument nije ograničen na jedan `script` element sa JavaScript kodom, pa je tako u jednom HTML dokumentu moguće imati proizvoljan broj `script` elemenata.
- Eksterni JavaScript definiše se unutar zasebnih fajlova sa ekstenzijom `.js`.
- Na frontendu, razdvojenost se postiže smeštanjem koda web sajtova u različite fajlove; HTML kôd se smešta u fajlove sa `.html` ekstenzijom, stilizacija u fajlove sa `.css` ekstenzijom, a programska logika u `.js` fajlove.
- Kako bi se JavaScript kôd iz zasebnog `.js` fajla dodao nekom HTML dokumentu, koristi se `script` element i njegov atribut `src`.
- Smeštanje JavaScript koda u zasebne fajlove veoma je korisno kada je jedan isti JavaScript kôd potrebno koristiti unutar većeg broja HTML dokumenata.
- Korišćenje eksternih JavaScript fajlova nije dobro rešenje ukoliko postoji mala količina JavaScript koda koji se koristi samo na jednoj stranici.
- Linijski, odnosno inline JavaScript kôd definiše se na samim HTML elementima, korišćenjem specijalnih atributa, kojima je moguće definisati JavaScript kôd koji će se aktivirati kada dođe do pojave nekog događaja.