

Forme

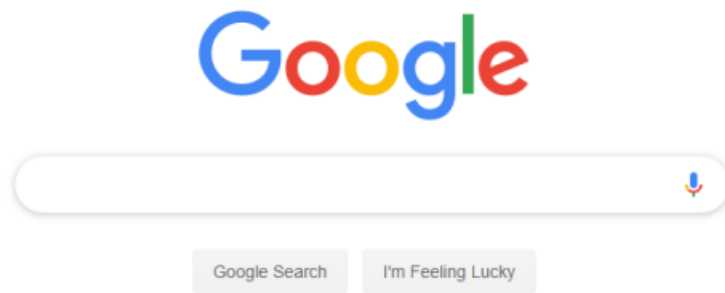
HTML dokument, sam po sebi, ne može mnogo toga da obezbedi korisniku, sem pukog pregleda informacija. Često postoji potreba da se određene informacije preuzmu od korisnika i proslede na dalju obradu bilo da je reč o obradi na serveru od strane neke serverske tehnologije ili obradu u lokalu korišćenjem klijentskih skripti. U takvim situacijama na scenu stupaju HTML forme.

ŠTA SU HTML FORME?

Formulari, odnosno forme, predstavljaju odličan alat za sakupljanje informacija od posetilaca web sajta. Formulari dozvoljavaju korisnicima da pošalju komentare i pitanja, zatraže neku informaciju, prijave se za newsletter, popune online aplikaciju ili unesu informacije za plaćanje kako bi kupili neki proizvod.

Termin forma, odnosno formular, potiče od pojma koji je korišćen da označi štampani dokument koji sadrži polja (prazne prostore) za upis podataka. HTML je preuzeo taj koncept i prilagodio forme digitalnom funkcionisanju.

Verovatno se najpoznatija forma na webu nalazi na početnoj stranici sajta Google



Forma na slici dobro je poznata svima, a funkcioniše vrlo jednostavno. Pomoću nje se reči za pretragu unose u jedno tekstualno polje, a zaim se, pomoću dugmeta koje je na slici označeno kao *Google Search*, prosleđuju na dalju obradu.

Kako funkcionišu HTML forme?

Sama po sebi, HTML forma ne pruža nikakvu funkcionalnost, već samo neku vrstu šablona za prikupljanje informacija. Kada korisnik popuni formu unutar nekog HTML dokumenta, prikupljeni podaci se prosleđuju do skripte koja obavlja obradu prosleđenih podataka. Takav proces je ilustrovan slikom ispod.



Prikazana su dva dokumenta. Dokument *contact.html* je HTML dokument koji sadrži HTML formu. Korisnik u takvu formu unosi podatke, nakon čega se oni prosleđuju dokumentu *contact_send.php*, koji sadrži kod za obradu prosleđenih podataka.

Forme su svojevrsan primer elementa koji se može koristiti za komunikaciju frontend i backend delova jedne web aplikacije. Iako se podaci koje je korisnik uneo unutar forme mogu obaviti i na klijentskom (frontend) delu aplikacije, najčešće se takav posao obavlja na serverskom delu od strane nekog od najpopularnijih backend jezika (PHP, Java, C#, Python...)

Posao frontend programera uglavnom ne podrazumeva obradu podataka koje je korisnik uneo u formu. Frontend programer se brine o tome da forma izgleda onako kako je dizajnom predviđeno, da sadrži sve neophodne elemente i da korisniku na jednostavan i razumljiv način omogući unos traženih podataka.

Kako se kreiraju HTML forme?

HTML forma kreira se korišćenjem *form* elementa. *Form* element gradi se upotrebom otvarajućeg `<form>` i zatvarajućeg `</form>` taga.

Element *form* može da sadrži attribute koji određuju način njegovog funkcionisanja. Atributi *form* elementa prikazani su na slici ispod.

Atribut	Vrednost	Opis
action	URL	Definiše lokaciju na koju je potrebno proslediti podatke forme
autocomplete	On/off	Definiše da li će forma imati autocomplete funkcionalnost, po kojoj će browser automatski popunjavati vrednosti polja na osnovu prethodnih korisničkih unosa;podrazumevana vrednost je <i>on</i>
method	get post	Definiše HTTP medou koja će se koristiti za prosleđivanje podataka forme
name		Definiše naziv forme
novalidate	novalidate	Definiše da podaci forme neće biti validirani prilikom prosleđivanja
target	_blank _self _parent _top	Definiše gde će se prikazati podaci koji se dobiju kao odgovor nakon prosleđivanja forme

Od svih prikazanih atributa, najviše se koriste *action* i *method*.

Action atribut definiše **kome** će podaci forme biti prosleđeni. To je uglavnom adresa do fajla sa serverskim skriptom, napisanom nekim od serverskih jezika. Takav podatak je moguće i izostaviti, a u toj situaciji će browser podrazumevati da se za obradu forme koristi ista ona strana na kojoj se forma nalazi.

Atribut *method* definiše način na koji će podaci biti prosleđeni i može imati dve vrednosti:

- **GET** – Prilikom korišćenja GET metoda podaci forme se prosleđuju serveru kroz URL adresu, koja je sastavni deo zaglavlja HTTP zahteva (GET metoda ne prosleđuje telo)
- **POST** – Kada se za prosleđivanje forme koristi POST metoda, podaci se smeštaju unutar tela HTTP zahteva.

GET metodu najbolje je koristiti u situacijama koje ne zahtevaju kreiranje novih podataka. Na primer, ukoliko je na sajtu potrebno implementirati pretragu, forma sa poljem za unos kriterijuma, podatke bi prosleđivala korišćenjem GET metode.

Sa druge strane, forma za kreiranje korisničkog naloga na nekom sajtu podatke uglavnom prosleđuje korišćenjem POST metode.

Elementi forme

Prikazani *form* tagovi samos u kontejneri za elemente koji se mogu naći unutar HTML forme. Tako *form* element može da sadrži jedan ili više sledećih elemenata:

- `<input>`
- `<textarea>`
- `<button>`
- `<select>`
- `<fieldset>`
- `<label>`

Navedeni elementi zaduženi su za prikupljanje i prosleđivanje podataka. Svi oni će biti opisani u nastavku ove lekcije.

`<input>` element

Element *input* je najznačajniji element forme. Može se pojaviti u nekoliko različitih varijacija, i to u zavisnosti od vrednosti njegovog *type* atributa. Različite varijante *input* elementa prikazani su u tabeli ispod.

Atribut	Opis
<code>text</code>	Element za unos teksta
<code>password</code>	Element za unos lozinki
<code>radio</code>	Element za selektovanje jedne od više ponuđenih opcija
<code>submit</code>	Element za prosleđivanje podataka forme
<code>checkbox</code>	Checkbox, kontrola koja može imati dva stanja: on i off
<code>color</code>	Kontrola za odabir boje (color picker)
<code>datetime-local</code>	Kontrola za odabir datuma i vremena
<code>email</code>	Element za unos email adrese
<code>file</code>	Dugme koje aktivira prozor za odabir fajla
<code>hidden</code>	Skriveni element
<code>month</code>	Element za odabir meseca i godine
<code>number</code>	Element za unos isključivo brojeva
<code>range</code>	Slider element za odabir brojčane vrednosti
<code>search</code>	Element za unos teksta za pretragu
<code>tel</code>	Element za unos telefonskog broja
<code>time</code>	Kontrola za unos vremena
<code>url</code>	Element za unos URL adrese
<code>week</code>	Element za unos nedelje i godine

Narandžastom bojom obeleženi su atributi koji su se pojavili u HTML5 jeziku.

Element *input* kreira se korišćenjem samozatvarajućeg taga, jer ne može posedovati nikakav sadržaj, već isključivo atribut. U nastavku će biti prikazane neke od najznačajnijih varijanti input HTML elemenata.

input (text)

Input element za unos jednostavnog teksta kreira se postavljanjem vrednosti *type* atributa na *text*, kao u sledećem primeru:

```
<input type="text" name="color" >
```

Pored atributa *type*, u primeru je navedena i vrednost za atribut *name*. To je veoma bitan podatak za skriptu koja će obraditi podatke ovakve forme, jer će na osnovu vrednosti *name* atributa podaci biti identifikovani.

U kombinaciji sa elementima forme za unos teksta, često se koristi i element *label*, čija uloga je da identifikuje polje za unos i obezbedi neku bližu informaciju o takvom polju. Kombinacija *label* i *input* prikazana je sledećim primerom:

```
<label for="color">Color:</label>
```

```
<input type="text" name="color" id="color">
```

Prikazani kod proizvodi efekat kao na slici ispod.

Color:

input (password)

Kada je potrebno obezbediti korisniku unos lozinke, koristi se *input* element čija je vrednost *type* atributa *password*, baš kao u sledećem primeru:

```
<input type="password" name="pass" id="pass">
```

Prikazani kod proizvodi efekat kao na slici ispod.

Sa slike se može videti da nema teksta pored input elementa, jer u kodu ne postoji *label* element. Sa slike se još može videti i to da ovaj tip *input* elementa sakriva karaktere koji se unose.

input (radio)

HTML omogućava i kreiranje radio button elementa, i to kao input elemenata sa *radio* vrednošću *type* atributa. Korišćenjem radio button elementa korisnik ne unosi nikakve vrednosti, već bira jednu od unapred ponuđenih. I ovo je kontrola koja se najčešće koristi u kombinaciji sa *label* elementom, a HTML kod koji ilustruje kreiranje radio buttona je sledeći:

```
<input name="country" type="radio" id="serbia" value="serbia">  
<label for="serbia">Serbia</label>  
  
<input name="country" type="radio" id="romania" value="romania">  
<label for="romania">Romania</label>  
  
<input name="country" type="radio" id="ukraine" value="ukraine">  
<label for="ukraine">Ukraine</label>
```

Prikazani kod proizvešće efekat kao na slici:



Za kreiranje svake stavke, odnosno mogućeg izbora, koristi se zaseban *input* element sa tipom *radio*. Da bi browser znao da više različitih input tagova kreiraju jednu grupu, neophodno je postaviti identične vrednosti *name* atributa. U primeru je vrednost *name* atributa *country* i na taj način je moguće odabrati samo jednu državu od ponuđene tri. U protivnom, odnosno, da vrednosti *name* atributa nisu iste, svaki radio button ponašao bi se kao zasebna celina i ne bi bio postignut željeni efekat.

U primeru je za svaki *input* element definisana i vrednost *value* atributa. To je atribut pomoću koga se definiše vrednost koja će biti prosleđena na obradu.

Za identifikaciju radio button elementa koristi se *label* element na isti način na koji je korišćen i sa *text input* elementom. Vrednost atributa *for* *label* elementa poklapa se sa vrednošću *id* atributa pripadajuće radio button kontrole.

Ukoliko je potrebno da neki radio button bude čekiran prilikom učitavanja stranice, koristi se boolean atribut *checked*. Na primer, ukoliko je potrebno da nakon učitavanja stranice čekirani radio button bude onaj koji ukazuje na državu sa nazivom *Romania*, dovoljno je uraditi sledeće:

```
<input name="country" type="radio" id="romania" value="romania" checked>
```

Input (checkbox)

U prethodnom pasusu prikazani su HTML elementi za kreiranje radio tipa elementa i tehnike kojima se postiže kreiranje grupe takvih elemenata, iz kojih se može odabrati samo jedna stavka. HTML poseduje i element koji dozvoljava kreiranje checkbox input elementa. To je element koji može imati dva stanja: *on/off*, odnosno *čekirano/nečekirano* (*true/false*). Za razliku od radio input elementa, checkbox se uglavnom upotrebljava samostalno, kada je od korisnika potrebno dobiti jednostavan da/ne odgovor na pitanje.

Sledeći primer ilustruje kreiranje checkbox kontrole korišćenjem *input* elementa:

```
<input type="checkbox" name="question">
```

```
<label for="question">I like HTML</label>
```

Prikazani kod proizvešće rezultat kao na slici ispod.



U prethodnom primeru, pored *input checkbox* elementa, definisan je i jedan *label* element koji prikazuje tekst korisniku. *Label* i *input checkbox* elementi međusobno su povezani identičnim vrednostima za *for* atribut. Na taj način dovoljno je kliknuti na površinu *label* elementa (na tekst labele) i klik će biti prosleđen *checkbox* elementu.

Baš kao i kod *input radio* elementa, i *checkbox* element može se na stranici pojaviti u čekiranom početnom stanju korišćenjem atributa *checked*. Sledeći kod to ilustruje:

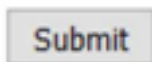
```
<input type="checkbox" name="question" checked>
```

input (submit)

Input element može se pojaviti još u jednom obliku. To je oblik koji dozvoljava kreiranje kontrole za prosleđivanje podataka forme. Tako nešto se postiže postavljanjem vrednosti atributa *type* na *submit*, kao u sledećem primeru:

```
<input type="submit" value="Submit">
```

Atribut *value* definiše tekst koji će biti prikazan unutar dugmeta na strani. Tako je efekat koji proizvodi navedeni kod prikazan na slici ispod.



input (number)

HTML5 donosi dosta novih input elemenata, među kojima je i input element tipa *number*. Ovaj element omogućava unos brojčane vrednosti korišćenjem kontrole koja poseduje Up i Down tastere za odabir brojčane vrednosti. Sledeći kod ilustruje način na koji se kreira input *number* element:

```
<input type="number" name="quantity" min="1" max="5">
```

Pored standardnih atributa, koji su već razmatrani u dosadašnjem toku kursa, *input* element tipa *number* poznaje i dva karakteristična atributa: *min* i *max*. Njima se definiše opseg vrednosti koje se mogu odabrati korišćenjem ovakve kontrole. To se može videti na slici ispod.



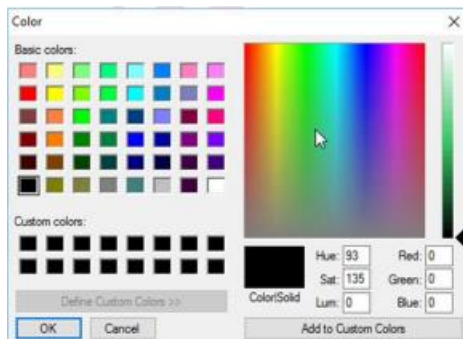
input (color)

Još jedan *input* element koji predstavlja novinu u HTML5 jeziku jeste *input* element tipa *color*. On omogućava odabir boje korišćenjem ugrađenog color pickera operativnog sistema. Sledeći kod ilustruje kreiranje ovog elementa:

```
<input type="color" name="favcolor">
```



Klikom na dugme dobija se prozor za odabir boje kao na slici ispod



input (range)

HTML5 donosi i element koji omogućava odabir bročane vrednosti u definisanom opsegu koja se najčešće na stranici renderuje kao slider kontrola. Takva kontrola se kreira korišćenjem input elementa tipa *range*, kao u sledećem primeru:

```
<input type="range" name="points" min="0" max="10">
```

Efekat je prikazan na slici ispod.



<textarea> element

Da bi se korisniku omogućio unos višelinijskog teksta, koristi se element *textarea*. Za razliku od *input text* elementa, *textarea* se ne kreira korišćenjem samozatvarajućeg taga, te zahteva otvarajući i zatvarajući tag, između kojih je moguće uneti tekst koji će se prikazati u kontroli prilikom učitavanja strane. Ukoliko takav tekst korisnik ne obriše, biće prosleđen zajedno sa podacima.

Primer kreiranja *textarea* elementa je sledeći:

```
<textarea name="komentar" placeholder="Unesi komentar..."></textarea>
```

<button> element

Nešto ranije u ovoj lekciji prikazan je element za kreiranje dugmeta za prosleđivanje podataka forme. Naravno, reč je o *input* elementu tipa *submit*. HTML poznaje još jedan element slične namene. Reč je o *button* elementu.

Element *button* se koristi kako bi se na strani kreiralo dugme. Ipak, za razliku od *input submit* elementa, *button* element se ne kreira korišćenjem samozatvarajućeg taga, odnosno poseduje svoj početak i kraj. To omogućava ovom elementu da prihvati sadržaj proizvoljnog oblika i da se tako *button* element kreira korišćenjem različitih elemenata. Upravo tako nešto ilustruje sledeći primer:

```
<button type="button">
```

```

```

```
Finish
```

```
</button>
```

U primeru je prikazan jedan *button* element koji u sebi sadrži sliku i tekst. Na *button* elementu definisana je i vrednost *type* atributa. Kao vrednost postavljeno je „button” što ovaj element proglašava običnim dugmetom, bez specijalne namene. Pored ove vrednosti, mogu se koristiti i vrednosti *submit* i *reset*, čime se postiže kreiranje dugmića specijalnih namena. Tip *submit* definiše da je reč o dugmetu koje će aktivirati prosleđivanje podataka forme, a tip *reset* da je reč o dugmetu koje će resetovati podatke forme na početne vrednosti.

<select> element

Nešto ranije prikazan je element koji omogućava korisniku odabir jedne od više ponuđenih opcija. To je bio *input* element tipa *radio*. HTML poznaje još jedan element sa istom namenom, ali u nešto drugačijem obliku. Reč je o elementu *select*.

Select element omogućava kreiranje drop down kontrole u HTML dokumentu. *Select* element se kreira korišćenjem `<select>` i `</select>` tagova, dok se stavke koje će biti ponuđene korisniku kreiraju korišćenjem `<option>` i `</option>` tagova.

Sledeći primer ilustruje kreiranje drop down kontrole.

```
<select name="drzava">
```

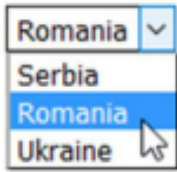
```
<option value="Serbia">Serbia</option>
```

```
<option value="Romania" selected>Romania</option>
```

```
<option value="Ukraine">Ukraine</option>
```

```
</select>
```

Prikazani kod proizvodi efekat kao na slici ispod.



Kao što je već rečeno, svaki *option* element unutar *select* elementa predstavlja jednu vrednost koju korisnik može da odabere. Svaki *option* element mora da sadrži i *value* atribut, koji definiše vrednost stavke koja će biti prosleđena na obradu.

Slično kao i kod *input radio* elementa i kod *option* elementa moguće je odabrati stavku koja će biti selektovana kada se stranica učita. To se postiže korišćenjem atributa *selected*.

Grupisanje podataka forme

Srodni elementi HTML forme mogu se grupisati korišćenjem elementa *fieldset*. Tako grupisani elementi biće jasno izdvojeni i na samoj stranici. Sledeći primer to ilustruje:

```
<form action="script.php" method="post" name="demo_form">
<fieldset>
  <legend>Personal info</legend>
  <label for="name">First name:</label><br>
  <input type="text" name="name" id="name"><br>

  <label for="surname">Last name:</label><br>
  <input type="text" name="surname" id="surname"><br>
</fieldset>

<fieldset>
  <legend>Contact</legend>
  <label for="email">Email:</label><br>
  <input type="text" name="email" id="email"><br>

  <label for="phone">Phone:</label><br>
  <input type="text" name="phone" id="phone">
</fieldset>
</form>
```

U primeru je kreirana forma sa dva *fieldset* regiona. Svaki *fieldset* element kao podelement poseduje element *legend*, unutar koga se definiše naslov grupe. Prikazani primer na stranici će proizvesti rezultat kao na slici ispod.

Personal info

First name:

Last name:

Contact

Email:

Phone:

Element *fieldset* kreira jasno izdvojene grupe kontrola koje se nalaze unutar forme. Grupe kontrola se na stranici unutar browsera jasno izdvajaju dodavanjem okvira. Pritom se naziv grupe definiše elementom *legend*.

Napomena

Na današnjem webu, forme su jedan od elemenata web sajtova i web aplikacija koji se može susresti u najrazličitijim oblicima. kada se govori o njihovom izgledu. HTML jezik neobezbeđuje gotovo nikakve mogućnosti za uticanje na izgled formi i njihovih elemenata. Tako nešto obavlja se korišćenjem jezika CSS i biće posebno objašnjeno u jednoj od narednih lekcija.

Rezime

- Forme predstavljaju alat za sakupljanje informacija od posetilaca web sajta.
- Prikupljeni podaci forme prosleđuju se do skripte koja obavlja obradu takvih podataka.
- HTML forma kreira se korišćenjem form elementa.
- Element form može da sadrži attribute koji određuju način njegovog funkcionisanja.
- Atribut action definiše kome će podaci forme biti prosleđeni.
- Atribut method definiše način na koji će podaci biti prosleđeni.
- Element input je najznačajniji element forme i može se pojaviti u nekoliko različitih varijacija, i to u zavisnosti od vrednosti njegovog type atributa.
- Unos jednostavnog teksta unutar forme postiže se korišćenjem input elementa sa vrednošću type atributa text.
- Unos višelinijanskog teksta unutar forme postiže se korišćenjem elementa textarea.
- Prosleđivanje podataka forme postiže se korišćenjem input elementa tipa submit.

X

Nenad Dmitrović