**Forme**

**00:14:39**

Jedinica: 5 od 30

**+Rezime**

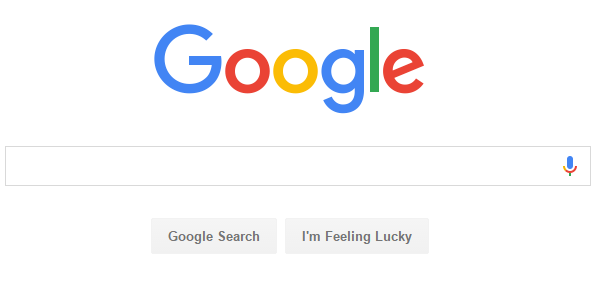
HTML dokument, sam po sebi, ne može mnogo toga da obezbedi korisniku, sem pukog pregleda informacija. Često postoji potreba da se određene informacije preuzmu od korisnika i proslede na dalju obradu, bilo da je reč o obradi na serveru, od strane neke serverske tehnologije ili obradi u lokalu, korišćenjem 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 veb-sajta. Formulari dozvoljavaju korisnicima da pošalju komentare i pitanja, zatraže neku informaciju, prijave se za Newsletter, popune onlajn 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 (prazane prostore) za upis podataka. HTML je preuzeo taj koncept i prilagodio forme digitalnom funkcionisanju.

Verovatno najpoznatija forma na vebu nalazi se na početnoj stranici sajta Google (slika 5.1).

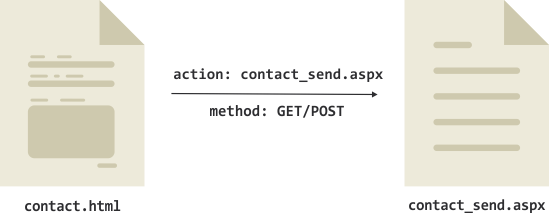


*Slika 5.1 – HTML forma na sajtu Google*

Forma na slici 5.1 dobro je poznata svima, a funkcioniše vrlo jednostavno. Pomoću nje se reči za pretragu unose u jedno tekstualno polje, a zatim se, pomoću submit 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 ilustrovan je slikom 5.2.



*Slika 5.2 – Prosleđivanje podataka forme na obradu*

Na slici 5.2 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.aspx*, koji sadrži serversku skriptu za obradu prosleđenih podataka.

**Kako se kreiraju HTML forme?**

HTML forma se kreira korišćenjem form elementa. Form element se gradi upotrebom otvarajućeg <form> i zatvarajućeg </form> taga, kao u sledećem primeru:

|  |  |
| --- | --- |
| 1 | <form></form> |

Element form može da sadrži atribute koji određuju način njegovog funkcionisanja. Atributi form elementa prikazani su tabelom 5.1.

|  |  |  |
| --- | --- | --- |
| **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 on |
| enctype | application/x-www-form-urlencoded multipart/form-data text/plain | definiše na koji način će podaci biti enkodovani prilikom prosleđivanja serveru; ova opcija ima efekta samo ukoliko se za slanje koristi metoda [POST](https://www.link-elearning.com/linkdl/opisPojma.php?id=144336) |
| method | get post | definiše HTTP metodu koja će se koristiti za prosleđivanje podataka forme |
| name | Text | definiše naziv forme |
| novalidate | Novalidate | definiše da podaci forme neće biti [validirani](https://www.link-elearning.com/linkdl/opisPojma.php?id=144289) prilikom prosleđivanja |
| target | \_blank \_self \_parent \_top | definiše gde će se prikazati podaci koji se dobiju kao odgovor nakon prosleđivanja forme |

*Tabela 5.1 – Atributi form elementa*

Atributi čije se vrednosti najčešće definišu prilikom razvoja web aplikacija su action i method.

Action atribut definiše **kome** će podaci forme biti prosleđeni. To je uglavnom adresa do fajla sa serverskom skriptom, napisanom nekim od serverskih jezika. Ukoliko se izostavi ovaj podatak, browser će 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 – podaci forme se prosleđuju serveru kroz URL. Informacije koje se prosleđuju na ovaj način su transparentne i podložne napadima hakera. Pošto URL može da ima maksimum 8.192 karaktera, ovaj metod nije podesan za duže formulare. Takođe, može doći do [transliteracije](https://www.link-elearning.com/linkdl/opisPojma.php?id=144337)ili [transkripcije](https://www.link-elearning.com/linkdl/opisPojma.php?id=144338) i neki karakteri se mogu promeniti ili izgubiti.
* POST – podaci se pakuju unutar poruke [HTTP zahteva](https://www.link-elearning.com/linkdl/opisPojma.php?id=144339). Podaci nisu šifrovani i stoga su (iako sigurniji nego u GET metodu) ipak podložni napadima hakera, tako da, ako sakupljamo personalne informacije kao što su korisnička imena, lozinke ili brojevi kreditnih kartica, moramo osigurati bezbednu konekciju do sigurnog servera.

Uzimajući u obzir opisane atribute koje form element može imati, može se napisati sledeći HTML kod:

|  |  |
| --- | --- |
| 1  2 | <form action="script.aspx" method="post" name="demo\_form" id="demo\_form" enctype="multipart/form-data" target="\_blank">  </form> |

U primeru je kreirana forma koja podatke prosleđuje skripti koja se nalazi unutar fajla *script.aspx*, kao metod slanja, koristi se HTTP metoda POST, naziv i id forme su *demo\_form*, a rezultat koji se bude dobio od servera na zahtev upućen ovom formom biće prikazan u novoj stranici.

**Elementi HTML forme**

Prikazani form tagovi su samo 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 su zaduženi za prikupljanje i prosleđivanje podataka. Svi oni će biti prikazani 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 prikazane su u tabeli 5.2.

|  |  |
| --- | --- |
| **Atribut** | **Opis** |
| text | element za unos teksta |
| password | element za unos lozinki |
| radio | element za selektovanje jedne od više ponuđenih opcija |
| submit | element za prosleđivanje podataka forme |
| checkbox | checkbox, kontrola koja može biti u dva stanja: on i off |
| color | kontrola za odabir boje (color picker) |
| datetime-local | kontrola za odabir datuma i vremena |
| email | element za unos email adrese |
| file | dugme koje aktivira prozor za odabir fajla |
| hidden | skriveni element |
| month | element za odabir meseca i godine |
| number | element za unos isključivo brojeva |
| range | slider element za odabir brojčane vrednosti |
| search | element za unos teksta za pretragu |
| tel | element za unos telefonskog broja |
| time | kontrola za unos vremena |
| url | element za unos URL adrese |
| week | element za unos nedelje i godine |

*Tabela 5.2 – Atributi form elementa*

|  |
| --- |
| **Napomena**    Narandžastom bojom obeleženi su atributi koji su se pojavili u HTML5 jeziku. |

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

**input (text)**

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

|  |  |
| --- | --- |
| 1 | <input type="text" name="color" > |

Pored atributa type, u primeru je navedena vrednost i 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 s 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 kontrola prikazana je sledećim primerom:

|  |  |
| --- | --- |
| 1  2 | <label for="color">Color: </label>  <input type="text" name="color" id="color"> |

Prikazani kod proizvodi efekat kao na slici 5.3.

https://www.link-elearning.com/linkdl/coursefiles/1166/PH5JC_05_03.jpg

*Slika 5.3 – Label i input elementi na stranici*

Bitno je primetiti da label element poseduje atribut for, kojim se vezuje za određeni input element. Vrednost atributa for, label elementa, mora se poklopiti sa vrednošću id atributa input elementa, na koji se label element odnosi.

Element input poznaje atribut sa nazivom placeholder, kojim se može definisati kratak tekst, koji će opisati vrednost koja se očekuje da bude uneta u tekstualnu kontrolu. Sledeći primer ilustruje upotrebu tog atributa:

|  |  |
| --- | --- |
| 1  2 | <label for="color">Color: </label>  <input type="text" name="color" id="color" placeholder="enter color..."> |

Ovoga puta, prikazani kod će imati efekat kao na slici 5.4.

https://www.link-elearning.com/linkdl/coursefiles/1166/PH5JC_05_04.jpg

*Slika 5.4 – Input element sa placeholder vrednošću*

**input (password)**

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

|  |  |
| --- | --- |
| 1 | <input type="password" name="pass" id="pass"> |

Prikazani kod, proizvodi efekat kao na slici 5.5.

https://www.link-elearning.com/linkdl/coursefiles/1166/PH5JC_05_05.jpg

*Slika 5.5 – Input kontrola za unos lozinki*

Sa slike 5.5 može se videti da nema teksta pored kontrole, jer nije kreiran label tag. Ipak, tekst se može jednostavno dodati postavljanjem label taga sa atributom for="pass".

Sa slike 5.5 se još može videti i to da ovaj tip input elementa sakriva karaktere koji se unose, ali ne garantuje, sam po sebi, sigurnost prilikom slanja podataka.

**input (Radio)**

HTML omogućava i kreiranje Radio Button kontrola, i to kao input elementa sa radio vrednošću type atributa. Korišćenjem Radio Button kontrole, 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:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | <input name="country" type="radio" id="Serbia" value="Serbia" checked="checked">  <label for="Serbia">Serbia</label>    <br />    <input name="country" type="radio" id="Romania" value="Romania">  <label for="Romania">Romania</label>    <br />    <input name="country" type="radio" id="Ukraine" value="Ukraine">  <label for="Ukraine">Ukraine</label> |

Prikazani kod proizvešće efekat kao na slici 5.6.

**

*Slika 5.6 – Grupa input kontrola tipa radio*

Za kreiranje svake stavke, odnosno mogućeg izbora, koristi se zaseban input element sa tipom radio. Da bi browser znao da više različitih input tagova kreira jednu grupu, neophodno je postaviti identične vrednosti name atributa. U primeru je vrednost name atributa *county*, 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 bi se ponašao 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 kontrola na strani, koriste se label elementi, na isti način na koji su korišćeni i sa text input elementima. Vrednost atributa for, label elementa se poklapa sa vrednošću id atributa pripadajuće Radio Button kontrole.

Ukoliko je potrebno da neki Radio Button bude čekiran prilikom učitavanja stranice, koristi se [boolean](https://www.link-elearning.com/linkdl/opisPojma.php?id=144290) 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:

|  |  |
| --- | --- |
| 1 | <input name="country" type="radio" id="Romania" value="Romania" checked> |

**input (checkbox)**

U prethodnom pasusu prikazan je HTML element za kreiranje radio-kontrole, i tehnike kojima se postiže kreiranje grupe takvih kontrola, iz koje se može odabrati samo jedna. HTML poseduje i element koji dozvoljava kreiranje checkbox kontrole. To je kontrola koja može imati dva stanja: *on/off*, odnosno *true/false*, odnosno *čekirano/nečekirano*. Za razliku od radio-kontrola, checkbox se uglavnom upotrebljava samostalno, kada je od korisnika potrebo dobiti jednostavan da/ne odgovor na pitanje.

Sledeći primer ilustruje kreiranje checkbox kontrole korišćenjem input elementa:

|  |  |
| --- | --- |
| 1  2 | <input type="checkbox" name="question" value="Like" id="question">  <label for="question">I like HTML</label> |

Prikazani kod proizvešće rezultat kao na slici 5.7.

https://www.link-elearning.com/linkdl/coursefiles/1166/PH5JC_05_07.jpg

*Slika 5.7 – Input element checkbox tipa*

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

Input checkbox element poseduje atribut value, koji definiše vrednost koja će biti prosleđena zajedno sa formom u slučaju da je checkbox čekiran.

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:

|  |  |
| --- | --- |
| 1 | <input type="checkbox" name="question" value="Like" id="question" checked> |

**input (submit)**

Input element može se pojaviti u još 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:

|  |  |
| --- | --- |
| 1 | <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.

https://www.link-elearning.com/linkdl/coursefiles/1166/PH5JC_05_08.jpg

*Slika 5.8 – Input element submit tipa*

**input (hidden)**

Jedan od input elemenata je i onaj koji nema svoji vizualnu reprezentaciju na stranici, a dobija se postavljanjem tipa input elementa na hidden. Često se naziva i skriveno polje. Iako možda pomisao na sakriveno polje unutar forme deluje kontradiktorno, ovakva kontrola je moguća i često je u upotrebi. Sakriveno polje, kao što mu naziv kaže, *nije vidljivo* za korisnika, ali se može koristiti da privremeno sačuva neke podatke. Kada se kaže da ovakvo polje nije vidljivo, prevashodno se misli na prikaz na stranici. Polje tipa hidden svakako je vidljivo u kodu strane, a što je još značajnije, vrednost polja hidden se prosleđuje formi, kao i vrednost bilo kojeg drugog polja.

Sledeći primer ilustruje kreiranje jednog skrivenog polja.

|  |  |
| --- | --- |
| 1 | <input type="hidden" name="hiddenField" id="hiddenField" value="x"> |

Skriveno polje se najčešće koristi da se u njega, prilikom učitavanja stranice, upišu neki podaci, koji će biti prosleđeni skripti na obradu. Na primer, zamislite formular za sugestije i komentare na nekom sajtu koji prodaje proizvode. Ukoliko je korisnik ulogovan prilikom pristupa formi, sajt može u sakriveno polje upisati njegov identifikacioni broj. Takav podatak za korisnika nema nikakvu važnost, pa je korišćenje hidden polja za čuvanje ovakve vrednosti idealan izbor. Na taj način, kada korisnik unese podatke i prosledi formu, prosleđuje se i identifikacioni broj.

**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 (slika 5.9). Sledeći kod ilustruje način na koji se kreira input numberelement:

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

Pored standardnih atributa, koji su već razmatrani u dosadašnjem toku kursa, input kontrola 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 i na slici 5.9, koja ilustruje efekat koji prikazani kod ima na strani.

https://www.link-elearning.com/linkdl/coursefiles/1166/PH5JC_05_09.jpg

*Slika 5.9 – Input element number tipa*

Kao što je već rečeno, input element tipa number, na stranici se renderuje kao kontrola kod koje se vrednost bira korišćenjem Up i Down tastera sa desne strane elementa.

**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:

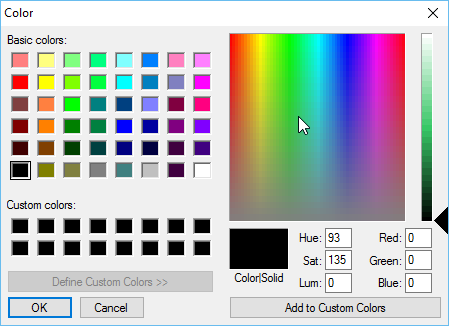
|  |  |
| --- | --- |
| 1 | <input type="color" name="favcolor"> |

Na stranici se ovaj element renderuje kao na slici 5.10.

https://www.link-elearning.com/linkdl/coursefiles/1166/PH5JC_05_10.jpg

*Slika 5.10 – Input element color tipa*

Klikom na dugme, dobija se prozor za odabir boje, kao na slici 5.11.



*Slika 5.11 – Color picker koji se prikazuje aktiviranjem input color elementa*

**input (range)**

HTML5 donosi i kontrolu koja omogućava odabir brojčane vrednosti u definisanom opsegu, koja se najčešće na stanici renderuje kao [slider kontrola](https://www.link-elearning.com/linkdl/opisPojma.php?id=144293). Takva kontrola se kreira korišćenjem input elementa tipa range, kao u sledećem primeru:

|  |  |
| --- | --- |
| 1 | <input type="range" name="points" min="0" max="10"> |

Efekat koji prikazani kod proizvodi je kao na slici 5.12.

https://www.link-elearning.com/linkdl/coursefiles/1166/PH5JC_05_12.jpg

*Slika 5.12 – Input element range tipa*

|  |
| --- |
| **Napomena**    *Sve do sada prikazane HTML5 input kontrole svoju pravu moć ispoljavaju u kombinaciji sa logikom koja se može definisati korišćenjem jezika JavaScript. To će biti predmet nekog od narednih modula.* |

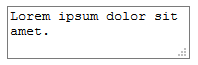
**<textarea> element**

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:

|  |  |
| --- | --- |
| 1 | <textarea name="description" id="description">Lorem ipsum dolor sit amet.</textarea> |

Efekat koji prikazani kod proizvodi je kao na slici 5.13.



*Slika 5.13 – Textarea element*

**<button> element**

Nešto ranije u ovoj lekciji prikazan je element za kreiranje kontrole 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 submitelementa, 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 kontrola kreira korišćenjem različitih elemenata. Upravo tako nešto ilustruje sledeći primer:

|  |  |
| --- | --- |
| 1  2  3  4 | <button type="button">     <img src="images/button.gif" alt="button" width="10" height="10">     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 je postavljen tekst *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 sa specijalnim namenama. 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 jeden 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.

|  |  |
| --- | --- |
| 1  2  3  4  5 | <select name="country" id="country">  <option value="Serbia">Serbia</option>  <option value="Romania" selected="selected">Romania</option>  <option value="Ukraine">Ukraine</option>  </select> |

Prikazani kod proizvodi efekat kao na slici 5.14.



*Slika 5.14 – Element select*

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

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

|  |
| --- |
| **Napomena**    Prilikom navođenja boolean atributa selected i nešto ranije atributa checked, može se primetiti izvesna razlika. Kada je u pasusu o input radio-kontroli definisan checked atribut, to je učinjeno direktnim navođenjem njegove vrednosti. S druge strane, prilikom navođenja selected atributa, definisani su i njegov naziv i vrednost. O čemu je reč?    Zapravo, oba pristupa su ispravna, i u prvom slučaju je primenjena tehnika koja se naziva minimizacija atributa, koja je dozvoljena samo u HTML varijanti jezika, ali ne i u xHTML varijanti. Tako su sledeće dve linije potpuno analogne:    <option value="Romania" selected="selected">Romania</option>    I    <option value="Romania" selected>Romania</option> |

Element select omogućava i selekciju više vrednosti korišćenjem atributa multiple (*boolean atribut*). Sledeći primer to ilustruje:

|  |  |
| --- | --- |
| 1  2  3  4  5 | <select name="country" id="country" multiple="multiple">  <option value="Serbia">Serbia</option>  <option value="Romania" selected="selected">Romania</option>  <option value="Ukraine">Ukraine</option>  </select> |

Efekat koji proizvodi prikazani kod ilustrovan je slikom 5.15.



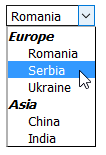
*Slika 5.15 – Select element sa multiple atributom*

Na slici 5.15 vidi se da select element sa multiple atributom ima nešto drugačiji izgled na stranici, koji više odgovara [list kontroli](https://www.link-elearning.com/linkdl/opisPojma.php?id=144291). Naravno, razlog tome jeste omogućavanje korisniku da selektuje više stavki. Selekcija više stavki se postiže uz pomoć Control tastera na PC-ju i Command tastera na [Mac](https://www.link-elearning.com/linkdl/opisPojma.php?id=144340) računarima.

Još jedan HTML element koji je moguće koristiti unutar select elementa jeste element optgroup. Ovaj element omogućava grupisanje srodnih stavki unutar drop down kontrole. Primer upotrebe ovakvog elementa je sledeći:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | <select>    <optgroup label="Europe">      <option value="romania">Romania</option>      <option value="serbia">Serbia</option>      <option value="ukraine">Ukraine</option>    </optgroup>    <optgroup label="Asia">      <option value="chine">China</option>      <option value="india">India</option>    </optgroup>  </select> |

Prikazani kod proizvodi efekat kao na slici 5.16



*Slika 5.16 – Select element sa dve grupe kreirane korišćenjem optgroup elementa*

Tag <optgroup> dozvoljava definisanje dva atributa. Atributom label definiše se naziv grupe koji će biti prikazan u drop down listi, dok atribut disabled (boolean atribut) definiše da li će selektovanje stavki grupe biti onemogućeno.

**Izbacite element koji nije input tag:**

 textarea

 text

 password

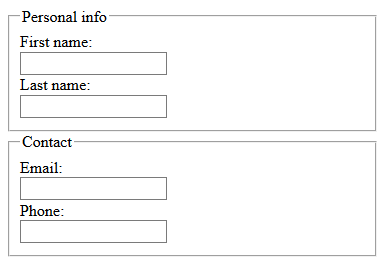
 radio button

**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:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | <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 će na stranici proizvesti rezultat kao na slici 5.17.



*Slika 5.17 – Izgled forme sa definisanim regionima korišćenjem fieldset elementa*

Element fieldset kreira jasno izdvoje

ne grupe kontrola koja se nalaze unutar forme. Grupe kontrola se na stranici unutar browsera jasno izdvajaju dodavanjem okvira. Pritom se naziv grupe definiše elementom legend.