

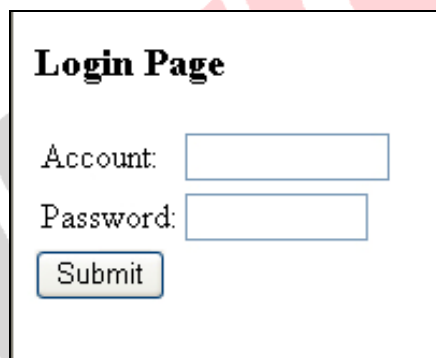
# Procesiranje formi

## Dizajniranje forme

Dizajniranje i procesiranje formi je veoma važan segment PHP programiranja. Forme omogućavaju interakciju korisnika sa nekom veb stranom i prikupljanje podataka neophodnih za prilagođavanje izgleda i sadržaja neke veb strane pojedinom korisniku. Forme su primarni mehanizam za prikupljanje podataka koje PHP program treba da procesira kako bi generisao neku informaciju, ažurirao bazu ili odgovorio na korisnikov upit.

Najčešći primer forme su forme za prijavljivanje korisnika (*login*). Dizajniranje ovakvih formi obično podrazumeva kreiranje dve strane. Prva strana (*login.php*) sadrži samu formu u okviru koje korisnik prosleđuje svoje korisničko ime i šifru. Druga strana (*welcome.php*) preuzima informacije prosledjene od strane forme i proverava ih. Ukoliko su podaci tačni prikazuje se pozdravna poruka korisniku. Ukoliko nisu, korisnik se vraća na *login.php* stranu.

Strana *login.php* bi mogla da izgleda ovako:



**Login Page**

Account:

Password:

Da bi XHTML kontrole bile prikazane na veb strani, neophodno je da se nalaze unutar `<form>` i `</form>` oznaka. Ove oznake moraju da okruže sve kontrole jedne forme, ali je moguće da sadrže i druge HTML oznake ili tekst. Sledi HTML kod strane *login.php*.

```
<html>
<body>
<h3>Login Page</h3>
<form name="Login" action="welcome.php" method="post">
  <table border="0">
    <tr>
      <td>Account: </td>
      <td><input type="text" name="Account" size="10"></td>
    </tr>
    <tr>
      <td>Password: </td>
      <td><input type="password" name="Password" size="10"></td>
    </tr>
  </table>
  <input type="submit" name="SubmitButton" value="Submit">
</form>
```

```
</form>
</body>
</html>
```

Oznaka `<form>` sadrži tri atributa važna za procesiranje formi:

- atribut *Name* – odnosi se na ime cele forme. Nije uvek neophodno imenovati formu, ali se to smatra dobrom programerskom praksom,
- atribut *Action* – vrednost ovog atributa predstavlja URL programa ili strane koji porcesiraju podatke unete u formu,
- atribut *Method* – određuje način na koji se podaci šalju. Može da ima vrednosti *GET* ili *POST*.

**GET metod** slanja podataka podrazumeva da se podaci sa forme dodaju na URL strane kojoj se šalju (URL naveden u atributu *Action*). Ovi podaci se odvajaju od URL pomoću znaka pitanja (?). Metod *GET* se sve manje koristi kao metod slanja podataka, s obzirom na to da postoji ograničenje u broju karaktera koji mogu biti poslani na ovaj način. Takođe, podaci poslani *GET* metodom su vidljivi unutar *browser*-a, pa je na taj način ugrožena privatnost podataka.

**POST metod** šalje podatke određenoj strani kao odvojeni tok podataka. Oni nisu vidljivi *browser*-u i ne postoji ograničenje u broju karaktera. Kad god je moguće, treba koristiti metod *POST* za slanje podataka.

## Polja forme

Forme se sastoje od polja ili kontrola, pomoću kojih korisnik unosi podatke, vrši izbor i prosleđuje podatke. Kontrole se unose u formu pomoću oznake `<input>`. Ova oznaka ima atribut *type* koji određuje vrstu polja (kontrola). Moguće vrednosti atributa *type* su:

- *"text"* – predstavlja tekstualno polje
- *"password"* – predstavlja tekstualno polje koje je maskirano nekim karakterom kako uneti tekst ne bi bio vidljiv u *browser*-u
- *"file"* – tekstualno polje koje uz sebe ima dugme *"Browse"* kako bi korisniku bilo omogućeno ubacivanje lokacije nekog fajla sa lokalnog diska
- *"checkbox"* – polja čija vrednost može, ali ne mora, biti izabrana
- *"radio"* – polja koja omogućavaju izbor tačno jedne vrednosti iz grupe
- *"hidden"* – polje nevidljivo korisniku koje omogućava prosleđivanje neke vrednosti
- *"submit"* – dugme čijim se pritiskom podaci sa forme šalju ka serveru
- *"reset"* – dugme koje poništava sve unete podatke

Pored kontrola definisanih oznakom `<input>`, postoje i kontrole definisane sledećim oznakama:

- `<SELECT>` i `<OPTION>` - omogućavaju izbor jedne ili više vrednosti od navedenih
- `<TEXTAREA>` - prikazuje tekst u posebnom prostoru sa klizačima

Forma na strani *login.php* sadrži jedno tekstualno polje, jedno *password* polje i dugme za potvrdu unosa *Submit*. Oznake za tekstualno i *password* polje izgledaju ovako:

```
<input type="text" name="Account" size="10">
```

```
<input type="password" name="Password" size="10">
```

Atribut *type* određuje koja vrsta kontrole će biti prikazana. Atribut *size* određuje veličinu ovih polja predstavljenu brojem karaktera. Atribut *name* je neophodan za svako polje forme s obzirom na to da se vrednosti ovog atributa, tj. imena dodeljena vrednostima koriste u procesiranju forme.

Oznaka za dugme za potvrdu je:

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

Tekst koji je dodeljen kao vrednost atributu *value* će biti ispisan na dugmetu.

## Procesiranje formi

Informacije koje korisnik unosi u formu se nakon pritiska dugmeta za potvrdu prosleđuju strani koja odgovara vrednosti *ACTION* atributa. Podaci se prenose do servera kao parovi imena svake kontrole forme i vrednosti koje su unešene u kontrolu od strane korisnika. Svaki par se razdvaja od sledećeg pomoću znaka &. Tako bi nakon popunjavanja forme za logovanje, tekst koji se prosleđuje ka serveru mogao da bude:

```
Account=Pera&Password=xyz&SubmitButton=Submit
```

Ukoliko u imenu neke kontrole ili u podatku koji je korisnik uneo postoje prazna mesta, ona se zamenjuju znakom +.

Za preuzimanje ovih vrednosti, PHP koristi tri tipa promenljivih:

- `$_GET[]`
- `$_POST[]`
- `$_REQUEST[]`

Ove promenljive ustvari predstavljaju asocijativne nizove vrednosti i vidljive su u celom skriptu.

Sledeći primer prikazuje jednu stranu sa formom koja će biti procesirana PHP-om:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD/XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml11-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
  <title>A Web Page</title>
</head>
<body>

  <form action="process.php" method="post">

    First Name: <input type="text" name="FName"/>
    Last Name: <input type="text" name="LName"/>
    City:      <input type="text" name="City"/>
    State:     <input type="text" name="State"/>
```

```

Message:    <textarea name="Message" cols="30" rows="5"></textarea>

<input type="submit" name="submit" value="Submit Data"/>

</form>

</body>
</html>

```

Ova forma sadrži četiri tekst polja, jednu *textarea* kontrolu i dugme za potvrdu. Kada korisnik potvrdi unos, podaci se šalju strani *process.php* na obradu. S obzirom na to da forma koristi POST metod, svi podaci će biti prosleđeni *\$\_POST* promenljivoj i to na sledeći način:

```

$_POST["FName"] = Marko
$_POST["LName"] = Markovic
$_POST["City"] = Novi Sad
$_POST["State"] = Srbija
$_POST["Message"] = Ja sam PHP programer.

```

Strana *process.php* bi mogla da izgleda ovako:

```

<?php

echo "Your First Name is: " . $_POST["FName"] . "<br/>";
echo "Your Last Name is: " . $_POST["LName"] . "<br/>";
echo "Your City is: " . $_POST["City"] . "<br/>";
echo "Your State is: " . $_POST["State"] . "<br/>";
echo "<br/>";
echo "Your Message is: " . $_POST["Message"];

?>

```

Ova strana preuzima podatke i koristi *echo* naredbu da ih ispiše u prozoru čitača. Da forma koristi *GET* metod, umesto *\$\_POST* promenljive bila bi korišćena *\$\_GET* promenljiva. Niz *\$\_REQUEST* može biti korišćen i sa *POST* i sa *GET* metodom.

Uobičajena je praksa da se PHP skript odvoji u posebnu stranu, međutim, moguće je i kombinovati HTML i PHP skript na istoj strani. Tada bi prethodni primer izgledao ovako:

```

<?php
if ($_POST['submit'] == "Submit Data")
{
    echo "Your First Name is: " . $_POST["FName"] . "<br/>";
    echo "Your Last Name is: " . $_POST["LName"] . "<br/>";
    echo "Your City is: " . $_POST["City"] . "<br/>";
    echo "Your State is: " . $_POST["State"] . "<br/>";
    echo "<br/>";
    echo "Your Message is: " . $_POST["Message"];
}
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD/XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml11-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

```

```

<head>
  <title>A Web Page</title>
</head>
<body>

<form action="form_process.php" method="post">

First Name: <input type="text" name="FName"/>
Last Name: <input type="text" name="LName"/>
City:      <input type="text" name="City"/>
State:     <input type="text" name="State"/>

Message:   <textarea name="Message" cols="30" rows="5"></textarea>

<input type="submit" name="submit" value="Submit Data"/>

</form>

</body>
</html>

```

Naredba *IF* na početku skripta proverava da li je dugme za potvrdu pritisnuto i ako jeste, ispisuje podatke unutar strane. Takođe, a obzirom na to što je PHP skript na istoj strani kao i sama forma, vrednost atributa ACTION je promenjena, pa strana sama sebi šalje podatke.

Nakon pritiska dugmeta *Submit*, tj. nakon slanja podataka strani na obradu, sama forma se ponovo prikazuje i to sa praznim poljima. Često je potrebno unešene podatke prikazati i unutar forme kako bi korisnik eventualno uneo neke promene (posebno ako neka polja nisu popunjena kako treba). Pošto se uneti podaci nalaze u nekoj od promenljivih *\$\_POST* ili *\$\_GET*, čitanjem odgovarajućih vrednosti moguće je popuniti polja forme, kao u sledećem primeru:

```

<!DOCTYPE html PUBLIC "-//W3C//DTD/XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml11-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
  <title>A Web Page</title>
</head>
<body>

<form action="process.php" method="post">

First Name: <input type="text" name="FName" value="<?php echo
$_POST['FName']?>" />
Last Name: <input type="text" name="LName" value="<?php echo
$_POST['LName']?>" />

<input type="submit" name="submit" value="Submit Data"/>

</form>

</body>
</html>

```