

PHP ŰRLAPKEZELÉS

Az interaktív weboldalak lehetővé teszik a felhasználók számára adatok beküldését. Az adatok feldolgozása a szerver oldalon történik meg.

Űrlapadatok elérése

A felhasználó által, egy űrlap segítségével elküldött adatokhoz három szuperglobális tömb segítségével férhetünk hozzá:

A `$_GET` tömb tartalmazza a GET metódussal, az URL részeként küldött adatokat.

A `$_POST` tömb tartalmazza a POST metódussal, a háttérben küldött adatokat.

A `$_REQUEST` tömbben megtaláljuk a GET és a POST metódussal küldött, valamint a sütikben tárolt adatokat (`$_COOKIE`) egyaránt.

Az adatok feldolgozásához ismerni kell az űrlapelemek neveit. Ezeket kulcsként használva az éppen használt szuperglobális tömbre juthatunk a beérkező adatokhoz. Pl. Ha

```
<input type="text" name="urlapelem_neve" value="Szöveg">
```

akkor

```
echo $_GET['urlapelem_neve']; vagy  
echo $_POST['urlapelem_neve']; vagy  
echo $_REQUEST['urlapelem_neve'];
```

\$ GET vagy \$ POST

\$_GET tömb használata

A GET metódus használata kerülendő, ha érzékeny vagy személyes adatok kerülnek továbbításra. Az elküldött adatok az URL-ben megjelennek. Senki sem szeretné adatait esetleg jelszavát egy keresőben a kedvenc oldala linkjében.

A GET metódus nem használható nagy mennyiségű adat továbbítására sem.

Hasznos lehet viszont, kereső funkció megvalósítása esetén. Ekkor az oldal URL-jét el lehet eltárolni könyvjelzőként.

A `$_GET` szuperglobális tömbben tárolt adatok nem csak űrlapból származhatnak. A HTML kódban megadott URL-ekben is továbbíthatunk paramétereket.

```
fajlneve.php?parameter1=ertek1&parameter2=ertek2
```

Az űrlapok többségénél POST metódust használunk.

Segítségével nagy mennyiségű adat továbbítható a háttérben.

isset() függvény

Az `isset()` függvény megvizsgálja, hogy létezik-e a paramétereként átadott tetszőleges típusú változó.

Az `isset()` függvény szintaxisa:

logikai = `isset` (tetszőleges változó) ;

1. paraméter: a vizsgált változó.

Visszatérési érték: Boolean

TRUE: ha a vizsgált változó létezik és nem NULL.

FALSE: ha a vizsgált változó nem létezik vagy NULL. Ha egy tömb nem létező elemével végzünk műveletet, akkor futási idejű hiba lép fel.

A `$_POST`, `$_GET` vagy `$_REQUEST` szuperglobális tömbök esetén különösen fontos, hogy ellenőrizzük a használni kívánt input elemekhez tartozó tömbelemek létezését. Sosem lehetünk abban biztosak, hogy a várt adatok elküldése valóban elküldésre kerültek.

Példa:

```
<body>
<h2>isset() függvény használata</h2>
<?php
    // Az űrlapot elküldték ?
    $_TEST =array("kulcs1"=>"szöveg1","kulcs2"=>"szöveg2");
    echo '<br><br>$_TEST["kulcs1"] : ' ; var_dump($_TEST["kulcs1"]);
    echo '<br>isset($_TEST["kulcs1"]): ' ; var_dump(isset($_TEST["kulcs1"]));

    echo '<br><br>$_TEST["kulcs2"] : ' ; var_dump($_TEST["kulcs2"]);
    echo '<br>isset($_TEST["kulcs2"]): ' ; var_dump(isset($_TEST["kulcs2"]));

    echo '<br><br>$_TEST["kulcs3"] : ' ; var_dump($_TEST["kulcs3"]);
    echo '<br>isset($_TEST["kulcs3"]): ' ; var_dump(isset($_TEST["kulcs3"]));

    $_TEST["kulcs3"] = "";
    echo '<br><br>$_TEST["kulcs3"] : ' ; var_dump($_TEST["kulcs3"]);
    echo '<br>isset($_TEST["kulcs3"]): ' ; var_dump(isset($_TEST["kulcs3"]));

    $_TEST["kulcs3"] = "szöveg3";
    echo '<br><br>$_TEST["kulcs3"] : ' ; var_dump($_TEST["kulcs3"]);
    echo '<br>isset($_TEST["kulcs3"]): ' ; var_dump(isset($_TEST["kulcs3"]));

    $_TEST["kulcs3"] = NULL;
    echo '<br><br>$_TEST["kulcs3"] : ' ; var_dump($_TEST["kulcs3"]);
    echo '<br>isset($_TEST["kulcs3"]): ' ; var_dump(isset($_TEST["kulcs3"]));
    ?>
</body>
```

Eredmény:

isset() függvény használata

```
$_TEST["kulcs1"] : string(8) "szöveg1"  
isset($_TEST["kulcs1"]) : bool(true)
```

```
$_TEST["kulcs2"] : string(8) "szöveg2"  
isset($_TEST["kulcs2"]) : bool(true)
```

```
$_TEST["kulcs3"] : NULL  
isset($_TEST["kulcs3"]) : bool(false)
```

```
$_TEST["kulcs3"] : string(0) ""  
isset($_TEST["kulcs3"]) : bool(true)
```

```
$_TEST["kulcs3"] : string(8) "szöveg3"  
isset($_TEST["kulcs3"]) : bool(true)
```

```
$_TEST["kulcs3"] : NULL  
isset($_TEST["kulcs3"]) : bool(false)
```

Úrlap feldolgozása

Első lépésben ellenőrizni kell, hogy a felhasználó valóban elküldte a vizsgált űrlapot. Egy-egy weboldal megnyitható rendszerint a rajta található űrlap beküldése nélkül is. Egy weboldal több űrlapot is tartalmazhat. A vizsgálatot az adott űrlap saját elküldési gombjának (submit) vizsgálatával végezhetjük: `if (isset($_GET['kuld']))` Ugyanilyen módon hivatkozhatunk az űrlap egyéb elemeire (a 'kuld' helyett 'urlapelem_nev'), megvizsgálva, hogy a látogató kitöltötte-e: ennek fejében változhat, kerülhet generálásra az adott oldal.

Most mi nem ezt a formulát használjuk, hanem létrehozunk egy általánosan használható `feldolgoz.php`-t, amelynek az űrlap küldi a beírt adatokat. Az űrlapról GET-tel vagy POST-tal küldött szuperglobálisként megjelenő tömb egységesítve egy `$tomb` nevű változóba kerül, amelynek elemeit aztán asszociatív módon egy `foreach` ciklus segítségével érjük el, annak kulcs és érték párjaival, kikerülve, hogy a változó űrlapok elemeire névvel kelljen hivatkozni.

Forráskód (feldolgoz.php):

```
<body>  
<?php  
    $tomb = ((isset($_GET) && ($_GET==true)) ? $_GET :  
            (isset($_POST) && ($_POST==true) ? $_POST : ""));  
    $sorszam=0;  
    foreach ($tomb as $urlapelem_neve => $ertek)  
    {  
        $sorszam++;  
        echo "A(z) $urlapelem_neve nevű $sorszam. paraméter értéke: $ertek<br/>";  
    }  
?>  
</body>
```

GET metódus esetén:

```
<body>
  <h2>$ _GET tömb használata</h2>
  <form action="feldolgoz.php" method="GET">
    <p><label for=nev>Név: </label>
    <input type=text name=nev></p>
    <p><label for=jelszo>Jelszó: </label>
    <input type=password name=jelszo></p>
    <input type=hidden name=azonosito value="visitor033">
    <p><label for=nem>Nem: </label><br>
    <input type=radio name=nem value=ferfi>Férfi<br>
    <input type=radio name=nem value=no checked>Nő</p>
    <p><label for=ettkor>Életkor: </label>
    <input type=number name=ettkor></p>
    <p><label for=hobbi>Hobbik: </label><br>
    <input type=checkbox name=hobbi value=szamitastechnika>Számítástechnika<br>
    <input type=checkbox name=hobbi value=sport>Sport<br>
    <input type=checkbox name=hobbi value=olvasas>Olvasás<br>
    <input type=checkbox name=hobbi value=tv disabled>TV</p>
    <p><label for=vezettseg>Végzettség: </label><br>
    <select name=vezettseg>
      <option selected>Általános iskola
      <option>Középiskola
      <option>Főiskola
      <option>Egyetem
    </select></p>
    <input type="submit" name="kuld" value="Elküldés">
  </form>
</body>
```

Eredmény:

A(z) nev nevű 1. paraméter értéke: Gipsz Jakab
A(z) jelszo nevű 2. paraméter értéke: nemtom
A(z) azonosito nevű 3. paraméter értéke: visitor033
A(z) nem nevű 4. paraméter értéke: ferfi
A(z) ettkor nevű 5. paraméter értéke: 15
A(z) hobbi nevű 6. paraméter értéke: olvasas
A(z) vezettseg nevű 7. paraméter értéke: általános iskola
A(z) kuld nevű 8. paraméter értéke: Elküldés

Készítsünk egy másik fájlt, amely most POST metódussal küldi el ugyanoda az adatokat!

Milyen különbséget fedezhetünk fel az elküldés során? Miért?

Látunk-e valamilyen változást az előző eredményéhez képest? Miért?

Az elküldés során nem csak egy másik php fájlunk küldhetjük el az űrlap adatait, hanem önmagának, ugyanazon fájlunk is. Mivel azonban már paramétereket kapott, új értékkel ezért nem ugyanazt a megjelenő felületet fogjuk látni, mint a paraméter nélküli esetben. A továbbiakban ezekre látunk példát.

Több submit kezelése

Ha egy űrlap többféleképp is feldolgoztató, akkor elhelyezhetünk rajta több submit elemet, és attól függően választunk a feldolgozást végző programrészek között, hogy éppen submit elem neve található a \$_POST vagy \$_GET szuperglobális tömbben.

Egy weboldalon több űrlap is elhelyezhető. Ekkor szintén submit elem neve alapján tudjuk a feldolgozást végző programrészt.

Egy űrlap két Submit (\$_GET):

```
<body>
  <?php
    // Az űrlapot az 1. nyomógommbal küldték?
    if (isset($_GET['kuld1'])) {
      echo "Az 1. gombhoz tartozó tevékenységek végrehajtása...";
      echo '<br> a $_GET["kuld1"] mező értéke: <b>'.$_GET['kuld1'].'<b>' ;
    }
    // Az űrlapot az 2. nyomógommbal küldték?
    if (isset($_GET['kuld2'])) {
      echo "Az 2. gombhoz tartozó tevékenységek végrehajtása...";
      echo '<br> a $_GET["kuld2"] mező értéke: <b>'.$_GET['kuld2'].'<b>' ;
    }
  ?>
  <h2>2 elküldő gomb kezelése ($_GET)</h2>
  <form action="ennek_a_fajlnak_a_neve.php" method="GET">
    <input type="submit" name="kuld1" value="1. submit">
    <input type="submit" name="kuld2" value="2. submit">
  </form>
</body>
```

Egyik eredmény:

Az 1. gombhoz tartozó tevékenységek
végrehajtása...

a \$_GET["kuld1"] mező értéke: 1. submit

2 elküldő gomb kezelése (\$_GET)

Egy űrlap két Submit (\$_POST):

```
<body>
  <?php
    // Az űrlapot az 1. nyomógommbal küldték?
    if (isset($_POST['kuld1'])) {
      echo "Az 1. gombhoz tartozó tevékenységek végrehajtása...";
      echo ' <br> a $_POST["kuld1"] mező értéke: <b>'.$_POST['kuld1'].'<b>' ;
    }
    // Az űrlapot az 2. nyomógommbal küldték?
    if (isset($_POST['kuld2'])) {
      echo "Az 2. gombhoz tartozó tevékenységek végrehajtása...";
      echo ' <br> a $_POST["kuld2"] mező értéke: <b>'.$_POST['kuld2'].'<b>' ;
    }
  ?>
  <h2>2 elküldő gomb kezelése ($_POST)</h2>
  <form action="ennek_a_fajlnak_a_neve.php" method="POST">
    <input type="submit" name="kuld1" value="1. submit">
    <input type="submit" name="kuld2" value="2. submit">
  </form>
</body>
```

Egyik eredmény:

Az 2. gombhoz tartozó tevékenységek

végrehajtása...

a \$_POST["kuld2"] mező értéke: **2. submit**

2 elküldő gomb kezelése (\$_POST)

1. submit

2. submit

Két űrlap különböző metódusokkal:

```
<body>
<?php
    // Az űrlapot az 1. nyomógommbal küldték?
    if (isset($_GET['kuld1'])) {
        echo "Az 1. űrlaphoz tartozó tevékenységek végrehajtása...";
        echo ' <br> a $_GET["kuld1"] mező értéke: <b>' . $_GET['kuld1'] . '<b>' ;
    }
    // Az űrlapot az 2. nyomógommbal küldték?
    if (isset($_POST['kuld2'])) {
        echo "A 2. űrlaphoz tartozó tevékenységek végrehajtása...";
        echo ' <br> a $_POST["kuld2"] mező értéke: <b>' . $_POST['kuld2'] . '<b>' ;
    }
?>
<h2>2 elküldő gomb kezelése ($_GET)</h2>
<form action="ennek_a_fajlnak_a_neve.php" method="GET"> <!--get módszer!!!-->
    <input type="submit" name="kuld1" value="1. submit">
</form>
<form action="ennek_a_fajlnak_a_neve.php" method="POST"> <!--post módszer!!!-->
    <input type="submit" name="kuld2" value="2. submit">
</form>
</body>
```

Eredmény1:

Az 1. űrlaphoz tartozó tevékenységek
végrehajtása...
a \$_GET["kuld1"] mező értéke: 1. submit

2 form és 2 elküldő gomb

| |
|-----------|
| 1. submit |
| 2. submit |

Eredmény2:

A 2. űrlaphoz tartozó tevékenységek végrehajtása...
a \$_POST["kuld2"] mező értéke: 2. submit

2 form és 2 elküldő gomb

| |
|-----------|
| 1. submit |
| 2. submit |

Szöveges beviteli mező

Szöveges adatbevitelnél a HTML5 űrlapelem többségénél a value jellemző értékét találjuk a \$_POST vagy \$_GET tömbben.

A példakód működését külön böngésző ablakban is érdemes kipróbálni, és megfigyelni megfigyelni miként változik az URL: >> Számok bevitele

Kereső mező (\$_GET):

```
<body>
  <?php
    // Az űrlapot elküldték ?
    if (isset($_GET["kuld"])) {
      // Létezik a 'szoveg' nevű mező?
      if (isset($_GET["szoveg"])) {
        if ( $_GET["szoveg"] > "" ) {
          echo "A keresett kifejezés: " . $_GET["szoveg"] . "<br>";
        }
      }
    }
  ?>
  <h2>Kereső mező ($_GET)</h2>
  <form action="ennek_a_fajlnak_a_neve.php" method="GET">
    <label for="szoveg">Keresett kifejezés:</label>
    <input type="text" id="szoveg" name="szoveg" value="" >
    <input type="submit" name="kuld" value="Keress!">
  </form>
</body>
```

Eredmény:

Kereső mező (\$_GET)

Keresett kifejezés:

Jelszó bevitele

A jelszavakat, személyes és mások számára nem publikus adatokat minden esetben POST módszerrel kell bekérni!

A GET módszer használatakor az URL-ben olvasható a megadott jelszó is.

Jelszó - \$_POST tömb használata:

```
<body>
<?php
    $nev = ""; $jelszo = "";
    // Az űrlapot elküldték ?
    if (isset($_POST["kuld"])) {
        // Létezik a 'nev' nevű mező?
        if (isset($_POST["nev"])) {
            if ( $_POST["nev"] > "" ) {$nev = $_POST["nev"];}
        }
        // Létezik a 'jelszo' nevű mező?
        if (isset($_POST["jelszo"])) {
            if ( $_POST["jelszo"] > "" ) {$jelszo = $_POST["jelszo"];}
        }
        // A beérkező adatok megjelenítése
        echo"<h2>Ezt küldte a szervernek:</h2>
            <br><b>Név:</b> $nev
            <br><b>Jelszó:</b> $jelszo
            <br><hr><br>";
    }
    ?>
    <h2>Jelszó - $_POST tömb használata</h2>
    <form action="ennek_a_fajlnak_a_neve.php" method="POST" >
        <label for="nev">Név:</label>
        <input type="text" id="nev" name="nev" value="" > <br>
        <label for="jelszo">Jelszó:</label>
        <input type="password" id="jelszo" name="jelszo" value="" >
        <input type="submit" name="kuld">
    </form>
</body>
```

Eredmény:

Jelszó - \$_POST tömb használata

Név:

Jelszó:

Szövegbevitel plusz

Minimális védelem

A felhasználó által küldött adatokkal mindig óvatosan kell bánni. Ha nem tároljuk, csak megjelenítjük azokat, akkor elegendő a HTML címkék hatástalanítása a htmlspecialchars() függvénnyel, tároláskor az addslashes() függvény használata az ajánlott minimum. Sztringkezelő függvényekkel tovább szűrhetjük a feldolgozás előtt álló adatokat.

Pl. \$nev = htmlspecialchars(\$_POST["nev"])

Szöveg visszaírása

Gyakran előfordul, hogy a felhasználó elsőre hibásan kitöltött űrlapot küld. Annak érdekében, hogy ne kelljen minden újra begépelnie az ismert adatokat érdemes az ismételtén kiküldött űrlapba visszaírni.

Pl. <input type="text" id="nev" name="nev" value="<?php echo \$nev ?>">

Szövegbevitel plusz:

```
<body>
<?php
    $nev = ""; $jelszo = "";
    // Az űrlapot elküldték ?
    if (isset($_POST["kuld"])) {
        // Létezik a 'nev' nevű mező?
        if (isset($_POST["nev"])) {
            if ( $_POST["nev"] > "" ) {
                $nev = htmlspecialchars($_POST["nev"]);
            }
        }
        // Létezik a 'jelszo' nevű mező?
        if (isset($_POST["jelszo"])) {
            if ( $_POST["jelszo"] > "" ) {
                $jelszo = htmlspecialchars($_POST["jelszo"]);
            }
        }
        // A beérkező adatok megjelenítése
        echo "<h2>Ezt küldte a szervernek:</h2>";
        echo "<br><b>Név:</b> $nev";
        echo "<br><b>Jelszó:</b> $jelszo";
        echo "<br><hr><br>";
    }
?>
<h2>Szövegbevitel plusz</h2>
<form action="ennek_a_fajlnak_a_neve.php" method="POST" >
    <label for="nev">Név:</label>
    <input type="text" id="nev" name="nev" value="<?php echo $nev ?>">
    <label for="jelszo">Jelszó:</label>
    <input type="password" id="jelszo" name="jelszo" value="" >
    <input type="submit" name="kuld">
</form>
</body>
```

Eredmény:

Szövegbevitel plusz

Név:

Jelszó:

Textarea kezelése

Többsoros szövegbeviteli mező esetén a korábban megismert módon, az elem nevét kulcsként használva \$_POST vagy \$_GET tömbben találjuk az elküldött adatokat.

Visszaíráskor azonban az adatokat nem a value jellemzőben, hanem a nyitó és záró címke között kell elhelyeznünk.

Példa:

```
<body>
  <?php
    $Szoveg = "";
    // Az űrlapot elküldték ?
    if (isset($_POST["kuld"])) {
      // Létezik a 'elem_nev' nevű mező?
      if (isset($_POST["elem_nev"])) {
        // Van benne valami?
        if ( $_POST["elem_nev"] > "" )
          { $Szoveg = htmlspecialchars($_POST["elem_nev"]); }
      }
      // A beérkező adatok megjelenítése
      echo"<h2>Ezt küldte a szervernek:</h2>
        <br><b>Szöveg:</b> $Szoveg
        <br><hr><br>";
    }
  ?>
  <h2>Textarea használata</h2>
  <form action="ennek_a_fajlnak_a_neve.php" method="POST" >
    <label for="textarea_id">Szöveg:</label>
    <textarea id="textarea_id" name="elem_nev"><?php echo $Szoveg ?></textarea>
    <input type="submit" name="kuld">
  </form>
</body>
```

Ezt küldte a szervernek:

Eredmény:

Szöveg: Próbaszöveg

Textarea használata

Szöveg:

Jelölőnégyzet kezelése

A jelölőnégyzet value jellemzőjének értéke kerül elküldésre abban az esetben, ha ki van választva. Kötelező és rendszerint elegendő is a \$_POST vagy \$_GET tömbben a hozzátartozó tömbelem létezését megvizsgálni.

Példánkban először vizsgáljuk a létezését, és ezt követően értékét.

```
if ((isset($_POST["check1"])) && ($_POST["check1"]=='Válasz1')) {... }
```

A && operátor esetén a műveletek feldolgozása balról jobbra halad, ami esetünkben fontos!

Checkbox használata:

```
<body>
  <?php
    $ch1 = ""; $ch2 = "";
    // Létezik a 'check1' nevű mező?
    // értéke = Válasz1?
    if ((isset($_POST["check1"])) && ($_POST["check1"]=='Válasz1')) {
      $ch1 = " checked ";
      echo "<br> check1 kiválasztva";
    }
    // Létezik a 'check2' nevű mező?
    // értéke = Válasz2?
    if ((isset($_POST["check2"])) && ($_POST["check2"]=='Válasz2')) {
      $ch2 = " checked ";
      echo "<br>check2 kiválasztva";
    }
  ?>
  <h2>Checkbox használata</h2>
  <form action="ennek_a_fajlnak_a_neve.php" method="POST" >
    <input type="checkbox" name="check1" id="check1" value="Válasz1"
      <?php echo $ch1 ?> >
    <label for="check1">Válasz 1.</label><br>
    <input type="checkbox" name="check2" id="check2" value="Válasz2"
      <?php echo $ch2 ?> >
    <label for="check2">Válasz 2.</label><br>
    <input type="submit" name="kuld">
  </form>
</body>
```

Eredmény:

check1 kiválasztva

Checkbox használata

☒ Válasz 1.

☐ Válasz 2.

Küldés

Kiválasztó gomb kezelése

Kiválasztó gombból rendszerint több kerül egy csoportba. Az egy csoportba tartozó radio gombok neve azonos.

Egy időben a csoport egy eleme lehet aktív. Ennek value jellemzőjének értéke kerül elküldésre.

Elegendő a \$_POST vagy \$_GET tömbben a csoporthoz tartozó tömbelem létezését egyszer megvizsgálni. Ennek értékéből kiderül, hogy melyik elem van kiválasztva.

Radio gomb használata:

```
<body>
<?php
    $ch1 = ""; $ch2 = "";
    // Létezik a 'radio_csoport'?
    if (isset($_POST["radio_csoport"])) {
        // értéke = Válasz1?
        if ($_POST["radio_csoport"]=="Válasz1") {
            $ch1 = " checked ";
            echo "<br> radio1 kiválasztva";
        }
        // értéke = Válasz2?
        if ($_POST["radio_csoport"]=="Válasz2") {
            $ch2 = " checked ";
            echo "<br> radio2 kiválasztva";
        }
    }
?>
<h2>Radio gomb használata</h2>
<form action="ennek_a_fajlnak_a_neve.php" method="POST" >
    <input type="radio" name="radio_csoport" id="radio1" value="Válasz1"
    <?php echo $ch1 ?> >
    <label for="radio1">Válasz 1.</label><br>
    <input type="radio" name="radio_csoport" id="radio2" value="Válasz2"
    <?php echo $ch2 ?> >
    <label for="radio2">Válasz 2.</label><br>
    <input type="submit" name="kuld">
</form>
</body>
```

Eredmény:

radio2 kiválasztva

Radio gomb használata

- ☐ Válasz 1.
- ☒ Válasz 2.

Küldés

Lenyíló lista kezelése

Lenyíló lista kezelése, ha csak egy választást engedélyezünk, akkor a kiválasztó gombnál megismert módon történik.

Itt is a kiválasztott elem value jellemzőjének értéke kerül elküldésre. Ha ez nincs megadva, akkor a felirata.

Lenyíló lista használata:

```
<body>
  <?php
    $sel1 = ""; $sel2 = ""; $sel3 = "";
    // Létezik a 'radio_csoport'?
    if (isset($_POST["select1"])) {
      // értéke = választas1?
      if ($_POST["select1"]=="valasztas1") {
        $sel1 = " selected ";
        echo "<br> 1. kiválasztva";
      }
      // értéke = választas2?
      if ($_POST["select1"]=="valasztas2") {
        $sel2 = " selected ";
        echo "<br> 2. kiválasztva";
      }
      // értéke = választas3?
      if ($_POST["select1"]=="valasztas3") {
        $sel3 = " selected ";
        echo "<br> 3. kiválasztva";
      }
    }
  ?>
  <h2>Kiválasztó lista használata</h2>
  <form action="ennek_a_fajlnak_a_neve.php" method="POST" >
    <select name="select1" size="3">
      <option value="valasztas1" <?php echo $sel1 ?> >1.</option>
      <option value="valasztas2" <?php echo $sel2 ?> >2.</option>
      <option value="valasztas3" <?php echo $sel3 ?> >3.</option>
    </select>
    <input type="submit" name="kuld">
  </form>
</body>
```

3. kiválasztva

Kiválasztó lista használata



1.

2.

3.

Multiselect lista kezelése

Ha egy lenyíló listában több elem is kiválasztható, akkor egy név kevés, tömböt kell használni.

A HTML kódban a név mögött elhelyezett szögletes zárójelet használunk

```
<select name="select1[]" multiple="multiple">
```

A \$_POST tömb listánkhoz tartozó tömbelemének tartalma, maga is tömb lesz és ennek megfelelően kell kezelni.

```
Pl. foreach ($_POST['select1'] as $selectedElem) {...}
```

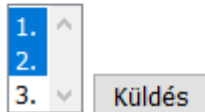
Lenyíló lista kezelése:

```
<body>
  <?php
    $sel1 = ""; $sel2 = ""; $sel3 = "";
    if (isset($_POST["select1"])) {
      foreach ($_POST['select1'] as $selectedElem)
      {
        if ($selectedElem=="valasztas1") {
          $sel1 = " selected ";
          echo "<br> 1. kiválasztva";
        }
        if ($selectedElem=="valasztas2") {
          $sel2 = " selected ";
          echo "<br> 2. kiválasztva";
        }
        if ($selectedElem=="valasztas3") {
          $sel3 = " selected ";
          echo "<br> 3. kiválasztva";
        }
      }
    }
  ?>
  <h2>Multiselect lista használata</h2>
  <form action="ennek_a_fajlnak_a_neve.php" method="POST" >
    <select name="select1[]" size="3" multiple="multiple">
      <option value="valasztas1" <?php echo $sel1 ?> >1.</option>
      <option value="valasztas2" <?php echo $sel2 ?> >2.</option>
      <option value="valasztas3" <?php echo $sel3 ?> >3.</option>
    </select>
    <input type="submit" name="kuld">
  </form>
</body>
```

Eredmény:

1. kiválasztva
2. kiválasztva

Multiselect lista használata



Képes elküldőgomb kezelése

Képes elküldőgombok esetén, a kattintás pillanatában érvényes egérkoordináták kerülnek elküldésre.

Ezek eléréséhez az elem nevét „_x”-el vagy „_y”-al kell kiegészíteni:

`$_POST['elemnev_x']`

`$_POST['elemnev_y']`

Példa:

```
<body>
  <?php
    // Az űrlapot elküldték ?
    if ((isset($_POST['kuld_x'])) && (isset($_POST['kuld_y']))) {
      echo "Az űrlap adatainak feldolgozása.... <br>";
      echo "X:".htmlspecialchars($_POST['kuld_x'])."<br>";
      echo "Y:".htmlspecialchars($_POST['kuld_y'])."<br>";
    }
  ?>
  <h2>Képes elküldő gomb használata </h2>
  <form action="ennek_a_fajlnak_a_neve.php" method="POST">
    <input type="image" name="kuld" src="gor.gif" width="48" height="48">
  </form>
</body>
```

Eredmény:

Az űrlap adatainak feldolgozása....

X:18

Y:16

Képes elküldő gomb használata



Fájlbevitel kezelése

Ha a HTML kódban egy input elem type jellemzőjének file értéket adunk, akkor lehetővé válik fájlok feltöltése. Ekkor az űrlap jellemzőinek beállításánál a következőkre figyeljünk:

```
method="post"
enctype="multipart/form-data"
```

A megjelenő nyomógombra kattintva a felhasználó saját számítógépén kitallózhatja a feltölteni kívánt fájlt, ami az űrlap elküldésével a szerver egy átmeneti könyvtárba kerül.

A feltöltött fájl adataihoz a \$_FILES szuperglobális tömbön keresztül férhetünk hozzá:

```
error: hibakód
name: a fájlnev
size: a fájl mérete (bájtban)
type: a fájl típusa
tmp_name: a fájl jelenlegi elérési útja a szerveren
Pl.
$_FILES["file1"]["name"]
ha
<input type="file" name="file1" >
```

Példa:

```
<body>
  <?php
    if (isset($_POST['kuld'])) {
      if ($_FILES["file"]["error"] > 0) {
        echo "Hibakód: " . $_FILES["file"]["error"] . "<br>";
      } else {
        echo "Fájlnév: " . $_FILES["file"]["name"] . "<br>";
        echo "Fájltípus: " . $_FILES["file"]["type"] . "<br>";
        echo "Fájlméret: " . ($_FILES["file"]["size"] / 1024) . " kB<br>";
        echo "Átmeneti könyvtár: " . $_FILES["file"]["tmp_name"] . "<br>";
        echo "Ha a fájl jellemzői megfelelők, akkor <br>
              már csak a kívánt könyvtárba kell másolni ..." ;
      }
    }
  ?>
  <h2>Input file használata </h2>
  <form action="ennek_a_fajlnak_a_neve.php" method="POST" enctype="multipart/form-data">
    <label for="file_id">Fájlnév:</label>
    <input type="file" name="file" id="file_id"><br>
    <input type="submit" name="kuld" value="Feltöltés">
  </form>
</body>
```

Eredmény:

Input file használata

Fájlnév: Nincs kijelölve fájl.

Fájlbevitel szűkítése

A kiválasztott fájl típusát, méretét, kiterjesztését a \$_FILES szuperglobális tömbben vizsgálva, megelőzhetjük a nem megfelelő állományok feltöltését.

Példa:

```
<body>
<?php
    if (isset($_POST['kuld'])) {
        $vart_kiterjesztes_tomb = array("gif", "jpg", "png");
        $nev_kitejesztes_tomb = explode(".", $_FILES["file"]["name"]);
        $kitejesztes = end($nev_kitejesztes_tomb);

        if (($FILES["file"]["type"] == "image/gif")
            || ($FILES["file"]["type"] == "image/jpeg")
            || ($FILES["file"]["type"] == "image/png"))
            && ($FILES["file"]["size"] < 2048)
            && in_array($kitejesztes, $vart_kiterjesztes_tomb)) {

            if ($FILES["file"]["error"] > 0) {
                echo "Hibakód: " . $FILES["file"]["error"] . "<br>";
            } else {
                echo "Fájlnév: " . $FILES["file"]["name"] . "<br>";
                echo "Ha a fájl jellemzői megfelelők, akkor <br>
                    már csak a kívánt könyvtárba kell másolni ..." ;
            }
        } else {echo "Valami nem OK!";}
    }
?>
<h2>Input file szűkítése </h2>
<form action="ennek_a_fajlnak_a_neve.php" method="POST" enctype="multipart/form-data">
    <label for="file_id">Fájlnév:</label>
    <input type="file" name="file" id="file_id"><br>
    <input type="submit" name="kuld" value="Feltöltés">
</form>
</body>
```

Eredmény:

Input file szűkítése

Fájlnév: Nincs kijelölve fájl.

Multiselect input file

A HTML5 lehetőséget ad a file típusú input elem használatakor több fájl együttes kiválasztására a multiple jellemző megadásával.

A listáknál megismert módon itt is az input elem nevét tömbnévként használjuk.

Példa:

```
<body>
  <?php
    if ((isset($_POST['kuld'])) && (isset($_FILES['file']))) {
      foreach ($_FILES['file']['name'] as $filename) {
        echo 'Fájlnév: '.$filename.'  
';
      }
    }
  ?>
  <h2>Multiselect input file </h2>
  <form action="ennek_a_fajlnak_a_neve.php" method="POST" enctype="multipart/form-data">
    <label for="file_id">Fájlnév:</label>
    <input type="file" name="file[]" id="file_id" multiple="multiple"><br>
    <input type="submit" name="kuld" value="Feltöltés">
  </form>
</body>
```

Eredmény:

Multiselect input file

Fájlnév: Nincsenek kijelölve fájlok.