

WEB PROGRAMOZÁS

3.ELŐADÁS

Űrlapok

2

Globális és környezeti változók

Globális és környezeti változók

3

- A globális változók azok a változók, amelyeket a program legfelső szintjén, azaz a függvényeken kívül vezetünk be
- A függvényeken belül használt változók lokálisak
- A globális változókat **global** kulcsszóval kell deklarálni a függvényekben

```
$a = 1; /* globális hatáskör */  
  
function Test() {  
    echo $a; /* egy helyi változót vár */  
}  
  
Test();
```

```
$a = 1;  
$b = 2;  
  
function Osszead() {  
    global $a, $b;  
    $b = $a + $b;  
}  
  
Osszead();  
echo $b;
```

Globális és környezeti változók

4

- Globális változók elérésének másik módja a PHP által definiált speciális `$GLOBALS` tömb használata
- Például, ha van egy **`$valtozo`** nevű változó a globális hatókörben, akkor a függvény törzsében erre a **`$GLOBALS["valtozo"]`** névvel hivatkozhatunk
- Példa:

```
$a = 1;
$b = 2;

function Osszead() {
    $GLOBALS['b'] = $GLOBALS['a'] + $GLOBALS['b'];
}

Osszead();
echo $b;
```

Globális és környezeti változók

5

- A `$GLOBALS` asszociatív tömb, ahol a globális változó neve jelenti a kulcsot, és a változó értéke a tömbelem értéke
- A `$GLOBALS` tömb minden hatáskörben létezik, mivel a `$GLOBALS` egy *szuperglobális* változó

Globális és környezeti változók

6

- Ezeket a változókat környezeti változóknak nevezzük és a PHP program tetszőleges pontján használhatjuk a **global** kulcsszó használata nélkül
- Fontosabb változók:
 - ▣ \$GLOBALS: tartalmaz valamennyi globális hatókörrel definiált változót
 - ▣ \$_SERVER: A HTTP szervertől érkező változókkal feltöltött asszociatív tömb
 - ▣ \$_GET: GET módszer által szolgáltatott adatokat tartalmazó asszociatív tömb
 - ▣ \$_POST: POST módszer által szolgáltatott adatokat tartalmazó asszociatív tömb
 - ▣ \$_FILES: POST módszerrel feltöltött fájlokról tartalmaz információkat
 - ▣ \$_SESSION: Az aktuális szkripthez tartozó session változókat tartalmazó asszociatív tömb
 - ▣ \$_REQUEST: tartalmazza a \$_GET, \$_POST és \$_COOKIE tömböket

Globális változók kiírása

7

- Példa a \$GLOBALS kiírására:

```
<?php
    $user1 = "Judit";
    $user2 = "István";
    $user3 = "János";
    foreach ( $GLOBALS as $kulcs=>$ertek ){
        print "\$GLOBALS[\"$kulcs\"] == $ertek<br>";
    }
?>
```

- Eredmény:

```
$GLOBALS["GLOBALS"] = Array
$GLOBALS["_POST"] = Array
$GLOBALS["_GET"] = Array
$GLOBALS["_COOKIE"] = Array
$GLOBALS["_FILES"] = Array
$GLOBALS["user1"] = Judit
$GLOBALS["user2"] = István
$GLOBALS["user3"] = János
$GLOBALS["ertek"] = János
$GLOBALS["kulcs"] = ertek
```


Űrlap (Form)

9

- Különböző adatok elküldésére használjuk
- A űrlapból érkező adatokat a **\$_GET**/**\$_POST** és a **\$_FILES** tömbökön keresztül érjük el
- Egy egyszerű űrlap:

```
<form method="get" action="feldolgoz.php">  
    <input type="text" name="nev">  
    <input type="text" name="kor">  
    <input type="submit">  
</form>
```

Űrlap (Form)

10

- **Action:** azt közli a szerverrel, hogy melyik oldalra kell váltania akkor, amikor a felhasználó az űrlap küldő gombjára kattint
- **Method:** azt a módot szabályozza, ahogy az információk elküldésre kerülnek a szerver számára
 - ▣ **GET:** a felhasználó által az űrlapmezőkbe gépelt értékeket az URL-hez kapcsolja

Példa:

www.szerver.ro/feldolgoz.php?nev=Kis+Istvan&kor=23

- ▣ **POST:** az elküldött információk nem az URL útvonalban, hanem a HTTP kérés törzsében kerülnek továbbításra

Űrlapmezők és a PHP

11

- Az űrlap adatainak elküldése után azok a `$_GET` vagy `$_POST` asszociatív tömbbe kerülnek
- Az előbbi példa esetén a szövegbeviteli mezőbe írt karakterláncok a `$_GET["nev"]` valamint a `$_GET[„kor”]` tömbbelembe kerülnek
- Tehát az elküldött adatok a `$_GET` vagy `$_POST` globális tömbökön keresztül érhetők el

Űrlapok különféle feldolgozási módja

12

- A feldolgozó szkript külön fájlban van:

```
<html>
  <head><title>Űrlap</title></head>
  <body>
    <form method="post" action="feldolgoz.php">
      Név: <input type="text" name="nev" value="">
      <br><br>
      E-mail: <input type="text" name="email" value="">
      <br><br>
      <input type="submit" value="Elküldés">
    </form>
  </body>
</html>
```

feldolgoz.php

```
<?php
    print "Név: " . $_POST["nev"];
    print "<br><br>";
    print "E-mail: " . $_POST["email"];
?>
```

Űrlapok különféle feldolgozási módja

13

- A feldolgozást ugyanabban a fájlban végezzük el, ahol az űrlap kódja található:

feldolgoz2.php

```
<html>
<head><title>Űrlap és feldolgozó (2 az 1-ben!)</title></head>
<body>
<?php
    print "Név: " . $_POST["nev"];
    print "<br><br>";
    print "E-mail: " . $_POST["email"];
?>

<br><br>
<form method="post" action="feldolgoz2.php">
    Név: <input type="text" name="nev" value="">
    <br><br>
    E-mail: <input type="text" name="email" value="">
    <br><br>
    <input type="submit" value="Elküldés">
</form>
</body>
</html>
```

Eredmény:

Név: Kis Istvan

E-mail: kis@alma.ro

Név:

E-mail:

Űrlapok különféle feldolgozási módja

14

□ Változó létezésének vizsgálata

```
<?php
if (isset($_POST["nev"]) && isset($_POST["email"])){
    $adatok = "Név: " . $_POST["nev"] . "<br><br>E-mail: " . $_POST["email"];
}
else{
    $adatok = "";
}
print $adatok;
?>
```

- **isset(\$valtozo)** függvény: egy változó létezését vizsgálja, amely TRUE értéket ad vissza, ha a változó létezik, FALSE-t ha nem
- **\$PHP_SELF**: aktuális állományt jelöli

<form action="<?php print \$PHP_SELF?>" method="POST">

Beviteli mezők

15

- Szövegmező: egy ilyen elembe egysoros szöveget írhatunk

```
<input type="text" name="azonosító" value="szöveg">
```

- Elérése: `$_POST['azonosító']`
- Kezdőérték: `value` mező

- Jelszóbeviteli mező (password)

```
<input type="password" name="azonosító" value="szöveg">
```

- Paramétereinek funkciója ugyanaz, mint a szövegbeviteli mező esetén.

Beviteli mezők

16

□ Szövegterület:

```
<textarea name="azonosító" rows="5" cols="20">Szöveg</textarea>
```

- Elérése: `$_POST['azonosító']`
- Kezdőérték: tag-ek közé írt szöveg
- **rows**: sorok, **cols**: szélesség



Szöveg

Beviteli mezők

17

□ Szövegterület (Példa):

```
<html>
<head><title></title></head>
<body>
<form method="POST" action="szovegterulet.php">
Melyek a kedvenc webhelyeid?
<textarea name="WebHelyek" cols="50" rows="5">
http://
http://
http://
http://
</textarea>
<br><br><br>
<input type="submit" value="Elküld">
</form>
</body>
</html>
```

szovegterulet.php

```
<html>
<head><title></title></head>
<body>
A kedvenc webhelyeid:
<?php
echo $_POST['WebHelyek'];
?>
</body>
</html>
```

Beviteli mezők

18

□ Szövegterület (Példa):

Melyek a kedvenc webhelyeid?

```
http://honline.ro  
http://sapiencia.ro  
http://erdely.ma  
http://transindex.ro|
```

Elküld

Eredmény:

A kedvenc webhelyeid: `http://honline.ro http://sapiencia.ro http://erdely.ma http://transindex.ro`

Beviteli mezők

19

□ Rádiógombok (Radio):

`<input type="radio" name="azonosító" value="1">`

`<input type="radio" name="azonosító" value="1" checked>`

- Elérése: `isset($_POST['azonosító'])`
- Kezdőérték: ha **checked**, akkor ki van választva
- **value**: a csoportban való megkülönböztetésre használjuk
- Egy csoportba tartozó rádiógombok összekötése érdekében ugyanazt a nevet kell adni a csoport minden gombjának

□ Példa:

`<input name="Kerdes1" type="radio" value="Portó">`

`<input name="Kerdes1" type="radio" value="Lisszabon">`

`<input name="Kerdes1" type="radio" value="Madrid">`

Beviteli mezők

20

□ Rádiógombok (Radio):

- Ha az egyik elem ki van választva, akkor az űrlapelem name paraméterével megegyező kulcsú **\$_POST** tömbelembe kerül a kiválasztott űrlapelem value paramétere

□ Példa:

```
<input name="Kerdes1" type="radio" value="Portó">
```

```
<input name="Kerdes1" type="radio" value="Lisszabon"  
checked="checked">
```

```
<input name="Kerdes1" type="radio" value="Madrid">
```

⇒ `$_POST["Kerdes1"] = "Lisszabon"`

- Ha nincs alapértelmezett érték beállítva, nem jön létre a megfelelő **\$_POST** változó (pld. `$_POST[„Kerdes1"] = null` lesz)

Beviteli mezők

21

□ Rádiógombok (Példa):

<p>Mi Portugália fővárosa?</p> <p><input type="radio"/> Portó</p> <p><input type="radio"/> Lisszabon</p> <p><input type="radio"/> Madrid</p> <p>Tipp elküldése</p>	<p>Tévedsz, nem Portó a helyes válasz!</p> <p>Mi Portugália fővárosa?</p> <p><input type="radio"/> Portó</p> <p><input type="radio"/> Lisszabon</p> <p><input type="radio"/> Madrid</p> <p>Tipp elküldése</p>	<p>Eltaláltad, Lisszabon a helyes válasz!</p> <p>Mi Portugália fővárosa?</p> <p><input type="radio"/> Portó</p> <p><input type="radio"/> Lisszabon</p> <p><input type="radio"/> Madrid</p> <p>Tipp elküldése</p>
--	---	--

```
<?php
if (isset($_POST['elkuldott'])) {
    if ($_POST['kerdes1'] == "Lisszabon") {
        echo "Eltaláltad, $_POST[kerdes1] a helyes válasz!<hr>";
    }
    if ($_POST['kerdes1'] != "Lisszabon") {
        echo "Tévedsz, nem $_POST[kerdes1] a helyes válasz!<hr>";
    }
}
?>
```

Beviteli mezők

22

□ Jelölőnégyzet (checkbox):

`<input type="checkbox" name="azonosító" value="1">`

`<input type="checkbox" name="azonosító" value="1" checked>`

- Elérése: `isset($_POST['azonosító'])`
- Kezdőérték: ha **checked**, akkor ki van választva
- **value**: a csoportban való megkülönböztetésre használjuk

□ Példa

```
<form method="POST" action="jelolonegyzet.php">
  <p> Írtál már, PHP programot?
    <input type="checkbox" name="Valasz">
  </p>
  <input type="submit" value="Elküld">
</form>
```

Írtál már, PHP programot? ☐

Elküld

Notice: Undefined index: Valasz in

jelolonegyzet.php

```
<?php
echo $_POST['Valasz'];
?>
```

Írtál már, PHP programot? ☒

Elküld

on

Beviteli mezők

23

□ Példa (folytatás):

```
<form method="POST" action="jelolonegyzet.php">
  <p> Írtál már, PHP programot?
    <input type="checkbox" name="Valasz">
  </p>
  <input type="submit" value="Elküld">
</form>
```

Írtál már, PHP programot? ☐

Elküld

Érték:

Jelölőnégyzetet nem jelölted be!

```
if (empty($_POST['Valasz'])) {
    $_POST['Valasz'] = "";
}
echo "Érték: " . $_POST['Valasz'] . " <br /><br />";

if ($_POST['Valasz'] == 'on') {
    echo"<b>Jelölőnégyzetet bejelölted!</b><br />";
} else {
    echo"<b>Jelölőnégyzetet nem jelölted be!</b><br />";
}
```

Beviteli mezők

- **Jelölőnégyzet (checkbox):** csoportos használat esetén a name paraméterben jelezni kell, hogy tömbként kívánjuk továbbítani az értékeket
- **Példa:**

```
<form method="POST" action="jelolonegyzetek.php">  
  <input type="checkbox" name="nyelv[]" value="html" /> HTML  
  <input type="checkbox" name="nyelv[]" value="css" /> CSS  
  <input type="checkbox" name="nyelv[]" value="php" /> PHP  
  <br>  
  <input type="submit" value="Elkuld">  
</form>
```

```
<?php  
foreach ($_POST["nyelv"] as $nyelv){  
    print $nyelv . "<br>";  
}  
?>
```

```
html  
css  
php
```


Beviteli mezők

25

□ Listadobozok:

□ Egy elem kiválasztása

- Elérés: `$_POST['Arkategoria']`

```
<select name="Arkategoria">
  <option>1 millió Ft alatt</option>
  <option>1-2 millió Ft</option>
  <option>2-5 millió Ft</option>
  <option>5 millió Ft felett</option>
</select>
```

□ Több elem kiválasztása

- Elérés: `$_POST['MotorMeret'][index]`

```
<select name="MotorMeret[]" multiple>
  <option>1,0 L</option>
  <option>1,4 L</option>
  <option>1,6 L</option>
</select>
```

Beviteli mezők

26

- **Rejtett Űrlapmezők:** információk rejtett átadását teszik lehetővé.

```
<input type="hidden" name="azonosító" value="szöveg">
```

- Nem jeleníthetők meg

- Példa:

```
<?php
$Uzenet1="Ez egy láthatatlan üzenet";
echo "<form method='GET' action='valami.php'>";
echo "<input type='hidden' name='Rejtett1' value=' $Uzenet1'>";
echo "<input type='submit' value='Elküld'>";
echo "</form>";
?>
```

Űrlapokból származó értékek felhasználása

27

□ Példa:

Pénzeszsák Kereskedelmi Bank Rt. hitelkérő űrlap

Keresztnév: Vezetéknév: Életkor:

Cím:

Csikszereda

Mennyi a jelenlegi havi jövedelme? ▼

Mennyi hitelre lenne szüksége?

☒ 1 millió forintos hitelcsomagunk kamata 8,0 %

☐ 5 millió forintos hitelcsomagunk kamata 11,5 %

☐ 10 millió forintos hitelcsomagunk kamata 15,0 %

Űrlapokból származó értékek felhasználása

28

□ Példa:

Szöveg mezők

```
Keresztnév:  
<input name="Keresztnev" type="text">  
Vezetéknév:  
<input name="Vezeteknev" type="text">  
Életkor:  
<input name="Eletkor" type="text" size="3"><br><br>  
Cím:  
<textarea name="Cim" rows="4" cols="40">  
</textarea>
```

Lista

```
Mennyi a jelenlegi havi jövedelme?  
<select name="Jovedelem">  
<option value=0>100 000 forintnál kevesebb </option>  
<option value=100000>100 000 - 200 000 Ft</option>  
<option value=200000>200 000 - 300 000 Ft</option>  
<option value=300000>300 000 forintnál több</option>  
</select>
```

Rádiógombok

```
Mennyi hitelre lenne szüksége?<br><br>  
<input name="Hitel" type="radio" value="1000000">  
1 millió forintos hitelcsomagunk kamata 8,0 %  
<br>  
<input name="Hitel" type="radio" value="5000000">  
5 millió forintos hitelcsomagunk kamata 11,5 %  
<br>  
<input name="Hitel" type="radio" value="10000000">  
10 millió forintos hitelcsomagunk kamata 15,0 %
```

Űrlapokból származó értékek felhasználása

29

□ Példa:

PHP szkript

```
<?php
$JovedelemHatar = $_POST['Jovedelem'] * 12 / 5;
$KorHatar = ($_POST['Eletkor'] / 10 - ($_POST['Eletkor'] % 10) / 10) - 1;
$HitelHatar = $JovedelemHatar * $KorHatar;
echo "Igényelt hitelösszeg: $_POST[Hitel]<br>";
echo "Teljesíthető hitelösszeg: $HitelHatar<br><br>";
if ($_POST['Hitel'] <= $HitelHatar) echo "Kedves $_POST[Keresztnev] $_POST[Vezeteknev],
if ($_POST['Hitel'] > $HitelHatar) echo "Kedves $_POST[Keresztnev] $_POST[Vezeteknev], e
?>
```

Űrlapokból származó értékek felhasználása

30

□ 2. Példa:

Roncsderbi autókölcsönző

Keresztnév: Vezetéknév: Életkor:

Cím:

qwew

Rendelkezik érvényes jogosítvánnyal? ☒

Űrlapokból származó értékek felhasználása

31

□ 2. Példa:

```
<form method="post" action="auto.php">
<input type="hidden" name="elkuldott" value="true">
Keresztnév: <input name="keresztnev" type="text">
Vezetéknév: <input name="vezeteknev" type="text">
Életkor: <input name="eletkor" type="text"size="3"><br><br>
Cím:
<textarea name="cim" rows=4 cols=40>
</textarea>
<br><br>
Rendelkezik érvényes jogosítvánnyal?
<input name="jogositvany" type="checkbox">
<br><br>
<input type="submit" value="Kérelem elküldése">
</form>
```

Űrlapokból származó értékek felhasználása

32

□ 2. Példa:

```
<?php
if (isset($_POST['elkuldott'])) {
    if ($_POST['eletkor'] > 20 and $_POST['jogositvany'] == "on") {
        echo ("Kölcsönzési igényét elfogadtuk.<hr>");
    }
    if ($_POST['eletkor'] < 21 or $_POST['jogositvany'] == "") {
        echo ("Sajnos nem áll módunkban autót kölcsönözni Önnek.<hr>");
    }
}
?>
```


Dinamikus oldal készítése

33

□ Példa:

```
<form method="POST" action="dinamikus.php">
  <input type="hidden" name="elkuldott" value="true">
  Hány gyermeke van?
  <input name="szam" type="text">
  <br>
  <br>
  <input type="submit" value="Szám elküldése">
  <br>
</form>
```

Hány gyermeke van?

Gépelje be a(z) 1. gyermeke nevét:

Nyomja meg a gombot a továbblépéshez!

Dinamikus oldal készítése

34

```
<?php
if (isset($_POST['elkuldott'])) {
    echo "<form method='POST' action='dinamikus.php'>";
    for ($szamlalo = 0; $szamlalo < $_POST['szam']; $szamlalo++) {
        $eltolas = $szamlalo + 1;
        echo "<br>Gépelje be a(z) $eltolas. gyermeke nevét:<br>";
        echo "<input name='gyerek[]' type='text'>";
    }
    echo"<br>Nyomja meg a gombot a továbblépéshez!<br>";
    echo "<input type='submit' value='Következő'>";
    echo "<input type='hidden' name='elkuldott01' value='true'></form>";
} else {
    if (isset($_POST['elkuldott01'])) {
        $szaml = 0;
        echo "Gyermekei neve:";
        do {
            $gyerekek_neve = $_POST['gyerek'][$szaml];
            echo"<br><b>$gyerekek_neve</b>";
            $ures_ell = $gyerekek_neve;
            $szaml = $szaml + 1;
        } while ($ures_ell != "");
        if ($szaml == 1) {
            echo "Nem megfelelő";
        }
    }
}
?>
```

Űrlap ellenőrzés

35

- Üres mezők ellenőrzése: **empty** függvény
- Példa:

* required field.

Name: *

E-mail: *

Website:

Comment:

Gender: ☐ Female ☐ Male *

```
<?php
$nameErr = $emailErr = $genderErr = $websiteErr = "";
$name = $email = $gender = $comment = $website = "";

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if (empty($_POST["name"])) {
        $nameErr = "Name is required";
    } else {
        $name = test_input($_POST["name"]);
    }

    if (empty($_POST["email"])) {
        $emailErr = "Email is required";
    } else {
        $email = test_input($_POST["email"]);
    }
}
```

Űrlap ellenőrzés

36

□ Példa (folytatás):

```
function test_input($data) {  
    $data = trim($data);  
    $data = stripslashes($data);  
    $data = htmlspecialchars($data);  
    return $data;  
}
```

```
<p><span class="error">* required field.</span></p>  
<form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]); ?>">  
    Name: <input type="text" name="name">  
    <span class="error">* <?php echo $nameErr; ?></span>  
    <br><br>  
    E-mail: <input type="text" name="email">  
    <span class="error">* <?php echo $emailErr; ?></span>  
    <br><br>  
    Website: <input type="text" name="website">  
    <span class="error"><?php echo $websiteErr; ?></span>
```

Űrlap ellenőrzés

37

- Reguláris kifejezés: egy szintaktikai szabályok szerint leírt minta, amely segítségével szövegek, adatok feldolgozásakor, egyező részhalmazok kereshetőek ki – karaktereket, számokat és helyettesítő karaktereket tartalmaz
- Metakarakterek:
 - \ - az utána következő metakarakterre illeszkedik pl.:
"\"*" csillag karakterre, "\"\" visszaperjelre
 - . (pont) - bármely, kivéve az új sor karakter
 - ^ - a minta kezdete
 - \$ - a minta vége

Úrlap ellenőrzés

38

- [] - karakter osztály, a szögletes zárójelek közötti karakterek valamelyikeire bármilyen sorrendben illeszkedik
- () - csoportosítás, a kerek zárójelek közötti karakterekre pontos sorrendben illeszkedik
- [**^ab@**] - (tagadás a karakter osztályban) – bármi, kivéve a, b vagy @
- [**a-fA-F**] - ASCII kis és nagy betűkre illeszkedik a-tól f-ig

Úrlap ellenőrzés

39

□ Ismétlések:

- * - a minta 0 vagy többszöri ismétlése
- + - a minta 1 vagy többszöri ismétlése

□ Általános karakterosztályok:

- \d - decimális szám
- \D - nem decimális szám, minden más
- \s - szóköz karakter
- \S - bármely más karakter a szóközön kívül
- \w - bármely szó
- \W - bármi, ami nem szó

Űrlap ellenőrzés

40

- Példák reguláris kifejezésekre:
 - ▣ `go*gle` minta illeszkedik a `ggle-e`, a `gogle-re`, a `google-re`
 - ▣ `[abc]-re` illeszkedik „a”, „b”, vagy „c”
 - ▣ `[a-z]-re` illeszkedik bármelyik kisbetű
 - ▣ `[^abc]` illeszkedik bármilyen karakterre, amelyik nem „a”, „b”, vagy „c”
 - ▣ `[^a-z]` illeszkedik bármilyen egyedülálló karakterre, ami nem kisbetű

Űrlap ellenőrzés

41

- Reguláris kifejezések kiértékelése: `preg_match` függvény
- `Preg_match(minta, kifejezés)`
- Példa: betűket és fehér karaktereket fogadunk el

```
$name = test_input($_POST["name"]);  
// check if name only contains letters and whitespace  
if (!preg_match("/^[a-zA-Z ]*$/", $name)) {  
    $nameErr = "Only letters and white space allowed";  
}
```

Űrlap ellenőrzés

42

□ Űrlap mezőinek megtartása:

* required field.

Name: *

E-mail: * Invalid email format

Website:

Comment:

Gender: ☐ Female ☒ Male *

```
<form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]); ?>">
  Name: <input type="text" name="name" value="<?php echo $name; ?>">
  <span class="error">* <?php echo $nameErr; ?></span>
  <br><br>
  E-mail: <input type="text" name="email" value="<?php echo $email; ?>">
  <span class="error">* <?php echo $emailErr; ?></span>
```

WordPress leckék 1.

43

- Egyik legnépszerűbb és legismertebb tartalom-kezelő rendszer
- Alapvetően blog-készítésre tervezték, de tetszőleges weboldal készítésére is alkalmas
- Adminisztrációja és karbantartása kezdők számára is egyszerű
- Rengeteg funkciót kínál, ami még bővíthető (pluginek)
- A kinézet könnyen változtatható, a weben vannak ingyenes „témák” (themes)
- A rendszer ingyenesen letölthető az alábbi címekről:
 - ▣ www.wordpress.org – angol változat
 - ▣ <http://hu.wordpress.org/> - magyar változat

WordPress telepítés (lokális gépre)

44

- A telepítés lépései:
 - ▣ Tömörítsd ki a csomagot egy üres könyvtárba
 - ▣ Nyisd meg a *wp-config-sample.php* fájlt egy szerkesztővel és töltsd ki az adatbázis kapcsolattal kapcsolatos részeket
 - Létre kell hozni egy új adatbázist (*phpmyadmin* segítségével), majd azt a nevet kell megadni a konfigurációs fájlban

```
// ** MySQL beállítások - Ezeket a szolgáltatótól lehet beszerezni ** //  
/** Adatbázis neve */  
define('DB_NAME', 'wordpresshu');  
  
/** MySQL felhasználónév */  
define('DB_USER', 'root');  
  
/** MySQL jelszó. */  
define('DB_PASSWORD', '');  
  
/** MySQL kiszolgáló neve */  
define('DB_HOST', 'localhost');
```

WordPress telepítés (lokális gépre)

45

- A telepítés lépései (folytatás):
 - ▣ Mentsd el a fájlt, *wp-config.php* néven
 - ▣ Nyisd meg a </wp-admin/install.php> fájlt a böngésződben. Ez be fogja állítani azokat a táblákat, amire szüksége lesz
 - ▣ A telepítéshez az alábbi ablak jelenik meg, ahol beállítható a honlap alap adatai (Honlap név, admin felhasználónév és jelszó, email cím)

Alapadatok

Meg kell adni a következő információkat! Nem kell aggódni, ezek később bármikor megváltoztathatóak.

Honlap neve

Felhasználónév
A felhasználói nevek csak alfanumerikus karaktereket, szóközt, aláhúzást, kötőjelet, pontot és az @ jelet tartalmazhatnak.

Jelszó, kétszer
Amennyiben ez a mező üresen marad, a jelszó automatikusan fog generálódni.

Gyenge

Tipp: A jelszó legalább 7 karakter hosszúságú legyen. Annak érdekében, hogy erősebb legyen, célszerű kis- és nagybetűket, számokat és szimbólumokat kombinálni pl.: " ? \$ % ^ &). Ékezetes betűket ne használjunk!

E-mail címe
Mielőtt folytatnánk, figyelmesen ellenőrizni kell még egyszer az e-mail címet!

Adatvédelem ☐ Engedélyezzük a honlap tartalmának indexelését a keresőrobotok számára.

WordPress telepítése

WordPress telepítés (lokális gépre)

46

- A telepítés lépései (folytatás):
 - ▣ A telepítő ezután a bejelentkező oldalra fog vinni. Lépj be az *admin* felhasználó névvel és a jelszóval, amit a telepítés közben generált.

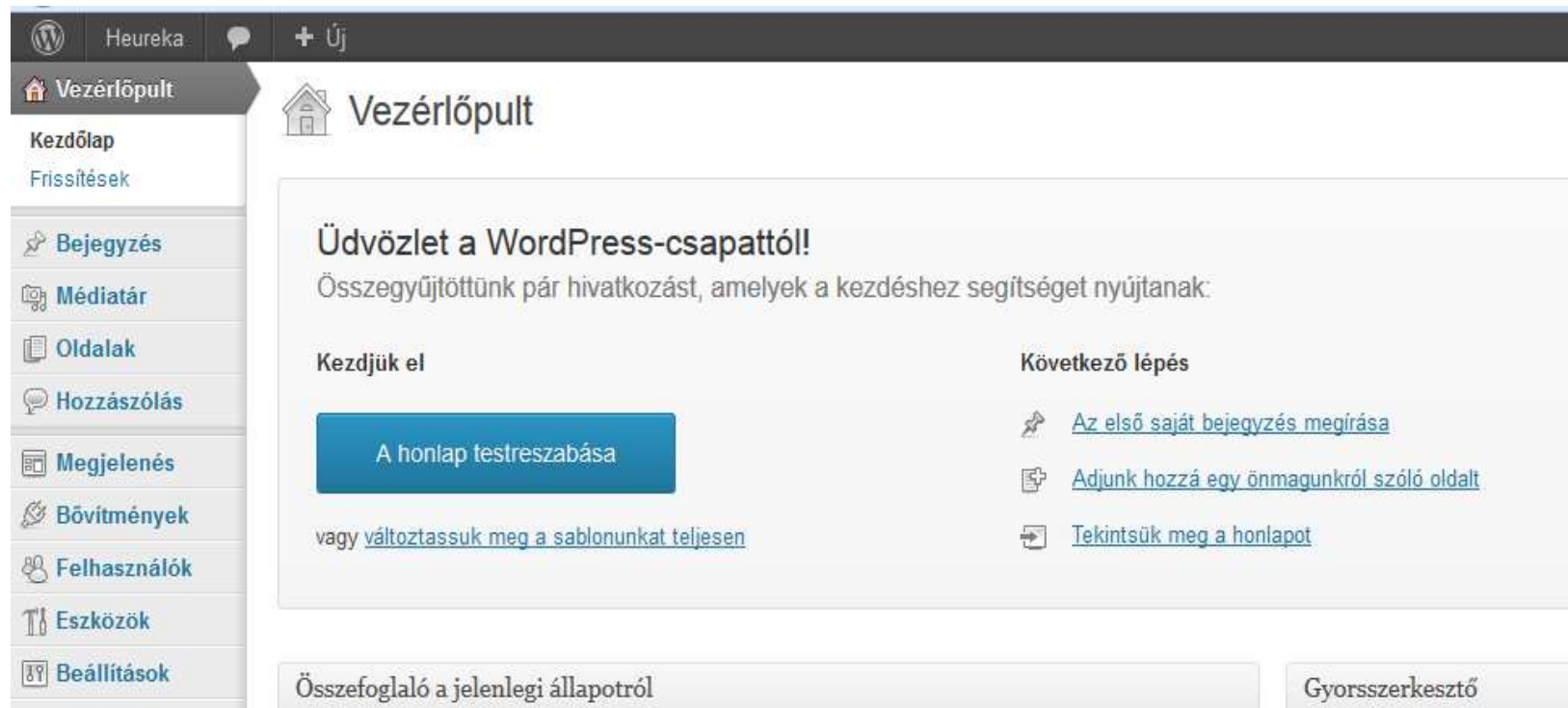


The image shows the WordPress login interface. At the top is the WordPress logo, consisting of a circular 'W' icon and the word 'WORDPRESS' in a blue serif font. Below the logo is a white rectangular box containing the login form. Inside this box, there are two text input fields: the first is labeled 'Felhasználónév' (Username) and the second is labeled 'Jelszó' (Password). Below the password field is a checkbox labeled 'Emlékezzem rám' (Remember me). To the right of the checkbox is a blue button with the text 'Bejelentkezés' (Log in). Below the login box, there are two links: 'Elfelejtett jelszó?' (Lost your password?) and '← Vissza (Heureka)' (← Back (Heureka)).

WordPress telepítés (lokális gépre)

47

- A bejelentkezés után a WordPress vezérlőpultjába lépünk be, amely segítségével az oldalunk testreszabható



A honlap testreszabása – alap elemek

48

- „A honlap testreszabása” gombra klikkelve, a Vezérlőpult nézetet vált és a bal oldali sávban tudjuk elvégezni az egyes műveleteket.

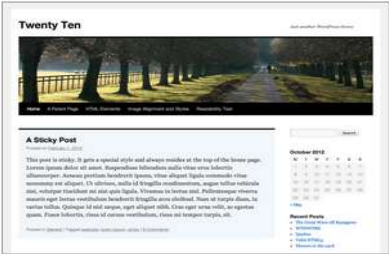


Sablon cseréje és telepítése

49

- A sablon meghatározza a honlapunk kinézetének minden elemét
- Cseréje a Vezérlőpult, *Megjelenés* almenüjének *Sablonok* menüpontjának kiválasztásával történik

Sablonkezelő
Sablonok telepítése



Twenty Ten

This post is styled to get a special style and always resides at the top of the front page.

A Sticky Post

This post is sticky. It gets a special style and always resides at the top of the front page.

October 2012

M	T	W	T	F	S	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

Recent Posts

- Post Title
- Post Title
- Post Title
- Post Title
- Post Title

Aktuális sablon

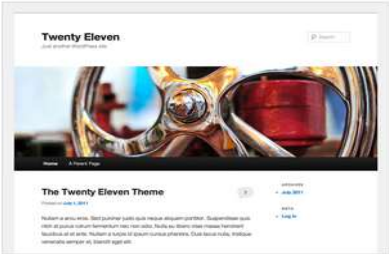
Twenty Ten

Szerző: [the WordPress team](#) | 1.6 verzió

The 2010 theme for WordPress is stylish, customizable, simple, and readable — make it yours with a custom menu, header image, and background. Twenty Ten supports six widgetized areas (two in the sidebar, four in the footer) and featured images (thumbnails for gallery posts and custom header images for posts and pages). It includes stylesheets for print and the admin Visual Editor, special styles for posts in the “Asides” and “Gallery” categories, and has an optional one-column page template that removes the sidebar.


[Testreszabás](#) [BEÁLLÍTÁSOK](#) [Widgetek](#) [Menük](#) [Fejleszt](#) [Háttér](#)

Elérhető sablonok




Twenty Eleven

The Twenty Eleven Theme



Twenty Thirteen

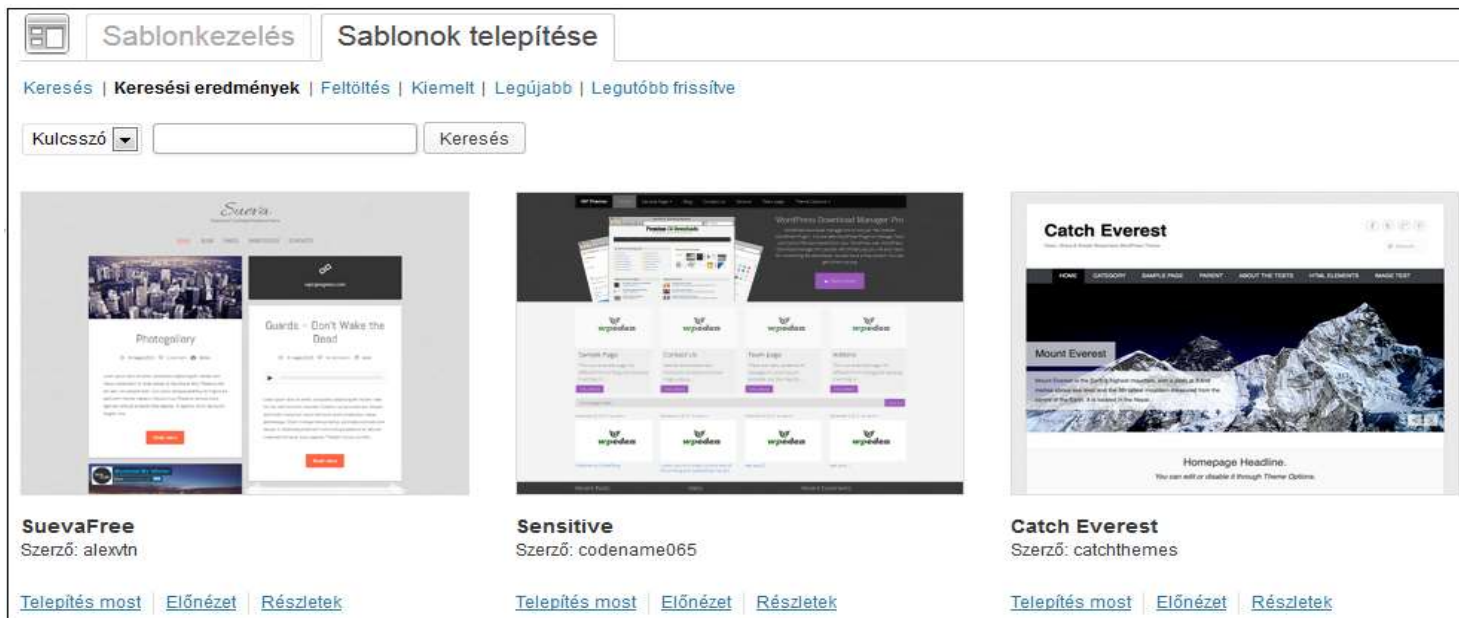


Twenty Twelve

Sablon cseréje és telepítése

50

- Új sablont a *Sablonok telepítése* fülre kattintva tudunk telepíteni
- A letöltött sablonok a elérhetőek lesznek a sablonkezelőből és a tárhelyünkön a *wp-content/themes/* mappában kerülnek tárolásra



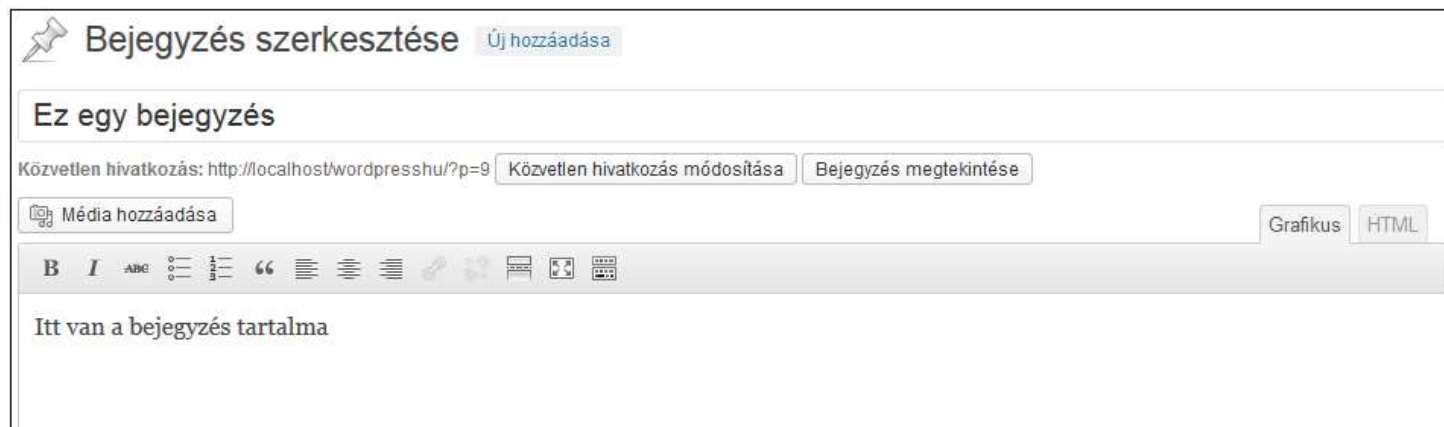
The screenshot displays the WordPress 'Sablonok telepítése' (Theme Installation) interface. At the top, there are tabs for 'Sablonkezelés' and 'Sablonok telepítése'. Below the tabs is a search bar with a 'Keresés' button. The main content area shows three theme preview cards:

- SuevaFree**
Szerző: alexvtn
Links: [Telepítés most](#), [Előnézet](#), [Részletek](#)
- Sensitive**
Szerző: codename065
Links: [Telepítés most](#), [Előnézet](#), [Részletek](#)
- Catch Everest**
Szerző: catchthemes
Links: [Telepítés most](#), [Előnézet](#), [Részletek](#)

Bejegyzések létrehozása és szerkesztése

51

- WordPress-ben alapú honlapokon a tartalmak oldalak és bejegyzések formájában jelenhetnek meg
- Ezek létrehozása és szerkesztése a Vezérlőpult *Bejegyzések*, illetve *Oldalak* menüpontjából lehetséges
- Új bejegyzéseket az *Új hozzáadása* menüpont kiválasztásával tudunk létrehozni. A bejegyzésnek nevet kell adni, majd a szerkesztőmezőben a tartalmát kell létrehozni



Bejegyzések létrehozása és szerkesztése

52


- A bejegyzés elkészítése után a *Közzététel* gombra klikkelve, elérhetővé válik a bejegyzésünk a közvetlen linken
- A későbbi módosítások alkalmával a Frissítés gombra klikkelve kerülnek megőrzésre a módosítások.

Közzététel

Mentés vázlatkéntElőnézet

Állapot: **Vázlat** Szerkesztés

Nyilvánosság: **Nyilvános** Szerkesztés

 **Azonnal közzétenni** Szerkesztés

Lomtárba helyezésKözzététel

Bejegyzések létrehozása és szerkesztése

53


- A bejegyzés elkészítése után a *Közzététel* gombra klikkelve, elérhetővé válik a bejegyzésünk a közvetlen linken
- A későbbi módosítások alkalmával a Frissítés gombra klikkelve kerülnek megőrzésre a módosítások.

Közzététel

Mentés vázlatkéntElőnézet

Állapot: **Vázlat** [Szerkesztés](#)

Nyilvánosság: **Nyilvános** [Szerkesztés](#)

 **Azonnal közzétenni** [Szerkesztés](#)

[Lomtárba helyezés](#)Közzététel