

WEBFEJLESZTÉS 2. – BEMENET, ŰRLAPOK

Horváth Győző

Egyetemi adjunktus

1117 Budapest,

Pázmány Péter sétány 1/C, 2.420

Tel: (1) 372-2500/1816

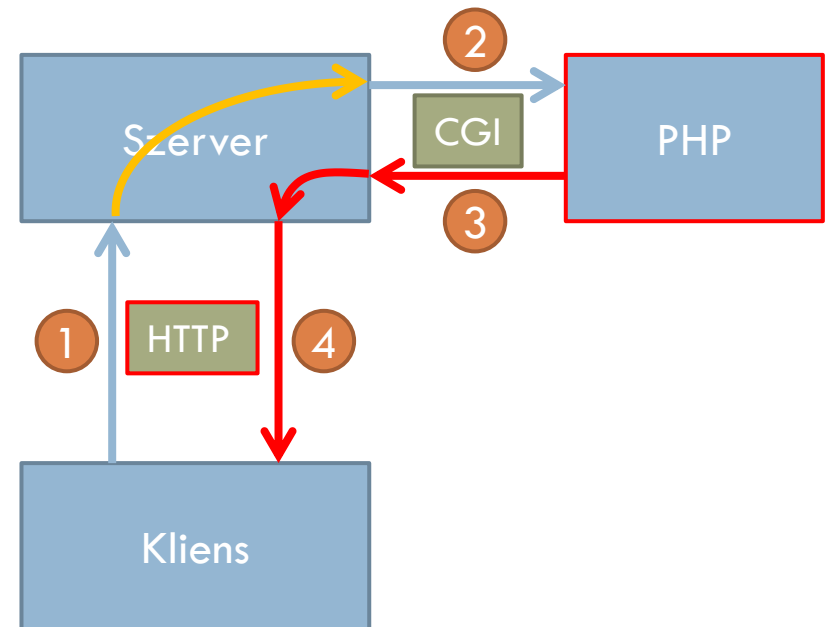
2

Ismétlés

Ismétlés

3

- Dinamikus szerveroldali webprogramozás: program állítja elő a tartalmat (HTML)
- PHP nyelv
 - ▣ Adatszerkezetek
 - ▣ Programozási tételek
- Kimenet (HTML)



HTTP kérés és válasz

4

```
GET / HTTP/1.1
Host: webprogramozas.inf.elte.hu
User-Agent: Mozilla/5.0 (Windows NT 6.1; rv:19.0) Gecko/20100101 Firefox/19.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: hu-hu,hu;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Cookie: __utma=159741371.1255432552.1308200517.1308200517.1308200517.1.
__utma=32143338.2145495546.1326
__utmz=32143338.1361177845.24.1
|utmctr=(not%20provided)
Connection: keep-alive
```

Kérés

```
HTTP/1.1 200 OK
Date: Wed, 03 Apr 2013 07:11:56 GMT
Server: Apache/2.2.10 (Linux/SUSE)
Last-Modified: Wed, 20 Feb 2013 08:39:44 GMT
ETag: "fe8438-6d6-4d623e65e9400"
Accept-Ranges: bytes
Content-Length: 1750
Content-Type: text/html

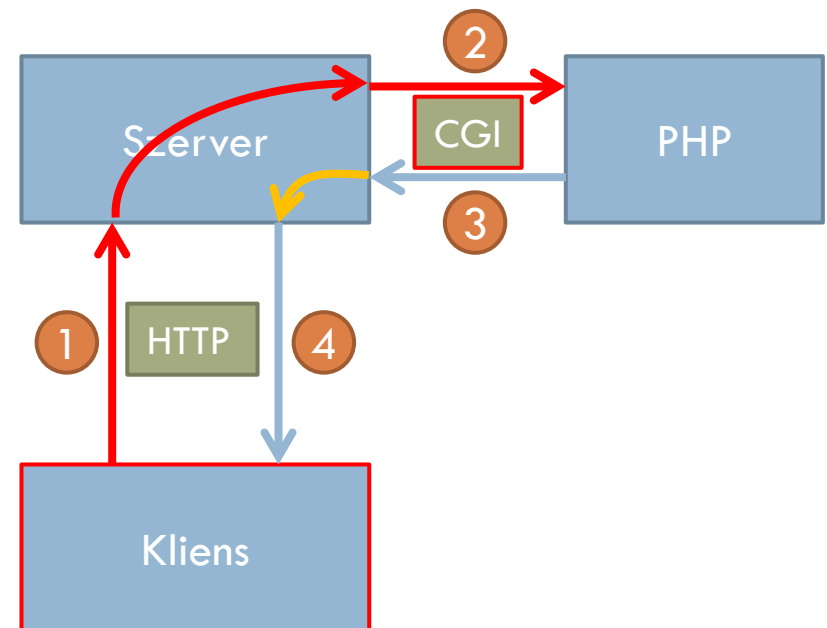
<!DOCTYPE html>
<html>
    ...
</html>
```

Válasz

Vázlat

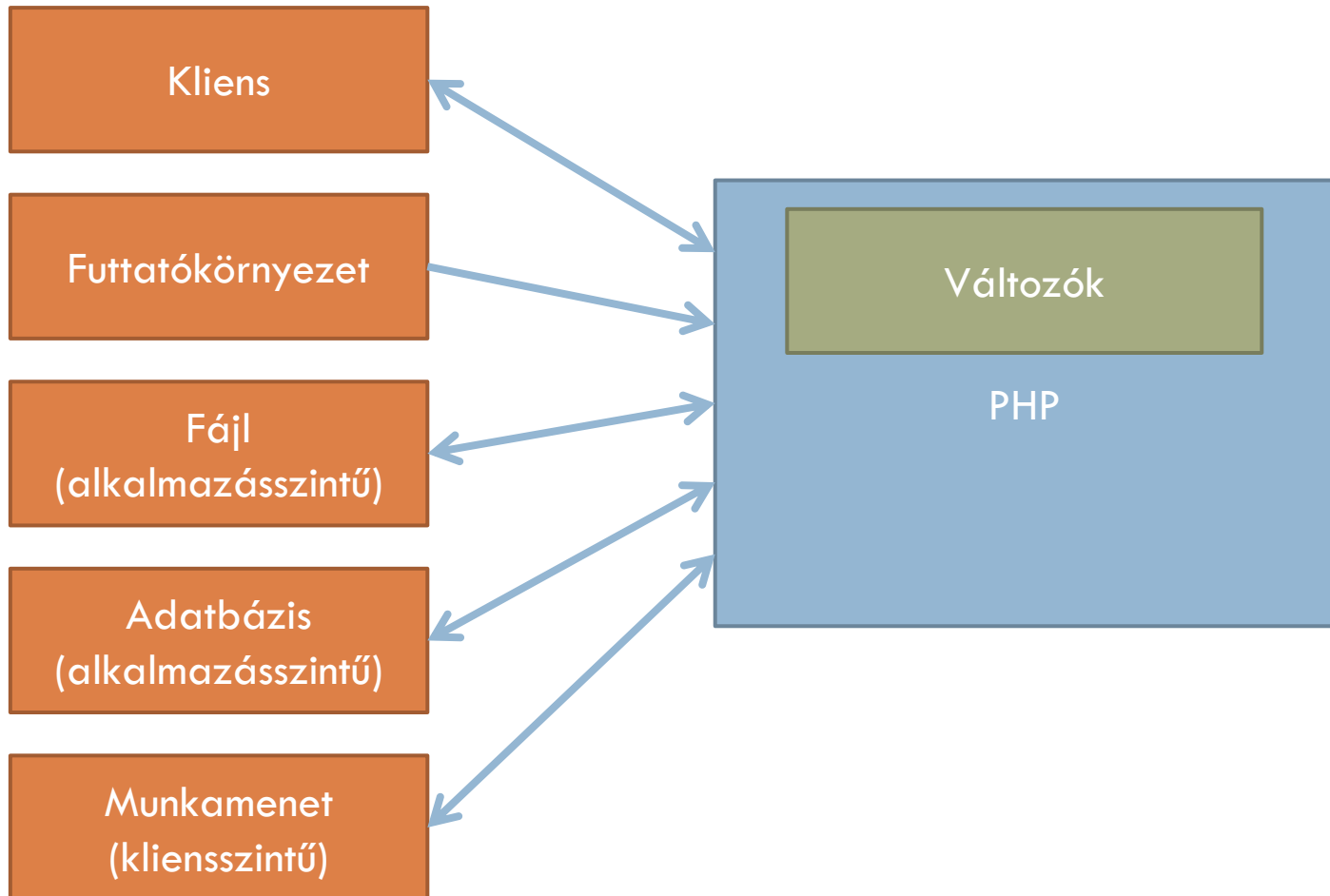
5

- CGI (PHP) programok bemenete
- CGI interfész
- Kliensről érkező adatok
- Űrlapok feldolgozása



PHP programok adatai

6



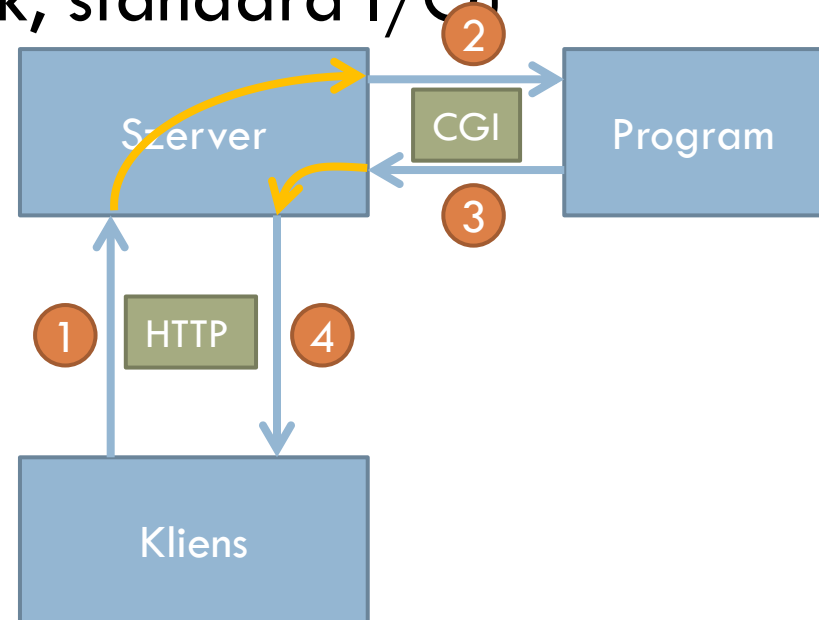
7 Common Gateway Interface

Szerveroldali input és output, Common Gateway Interface

Common Gateway Interface (CGI)

8

- Azt határozza meg, hogy egy webservert hogyan indíthat el egy programot és milyen módon cserél adatot vele.
- **Indítás:** bináris állomány a kért erőforrás
- **Adatok** (környezeti változók, standard I/O)
 - Kérés körülményei
 - URL
 - HTTP fejléc
 - HTTP üzenettörzs



Kérés körülményei és HTTP fejlécek

9

- Kérés körülményei → környezeti változók
 - ▣ REQUEST_METHOD: HTTP kérés metódusa
 - ▣ REMOTE_ADDRESS: kliens IP címe
 - ▣ CONTENT_LENGTH: üzenettest hossza bájtban
- HTTP fejlécek → környezeti változók
 - ▣ HTTP_*
 - ▣ Pl. Accept fejléc → HTTP_ACCEPT környezeti változó

```
[REMOTE_ADDR] => 188.142.184.197
[REMOTE_PORT] => 49323
[REQUEST_METHOD] => GET
[HTTP_ACCEPT] => text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
[HTTP_ACCEPT_LANGUAGE] => hu-hu,hu;q=0.8,en-US;q=0.5,en;q=0.3
```

URL

10

□ Kérés URL-je

```
<scheme>://<host>:<port>/<path>?<query>#<fragment>
```

□ Ezek leképezése környezeti változókra

- <scheme> → SERVER_PROTOCOL
- <host> → SERVER_NAME
- <port> → SERVER_PORT
- <path> → SCRIPT_NAME (PATH_INFO)
- <query> → QUERY_STRING

HTTP üzenettörzs

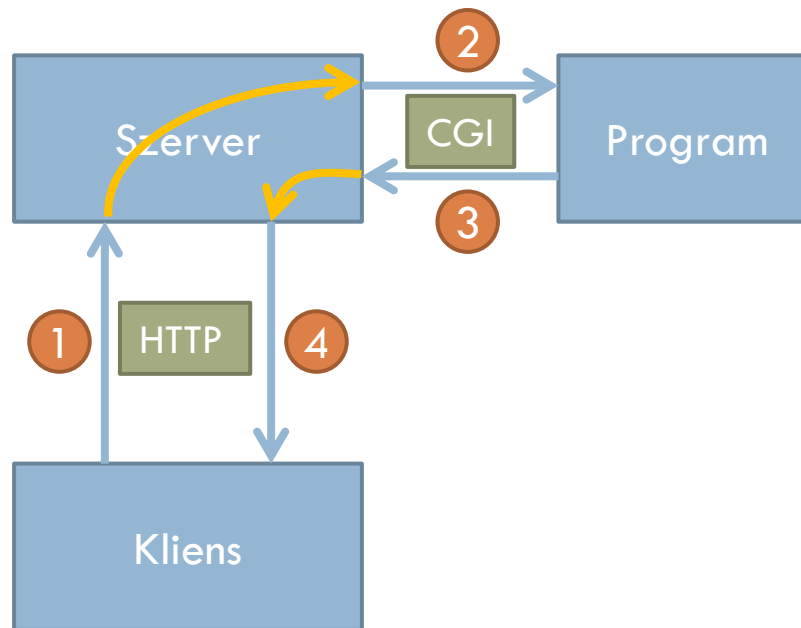
11

- **Standard bemenet**en jelenik meg
- Környezeti változó
 - `CONTENT_LENGTH`: adatmennyiség hossza

A program eredménye

12

- A program eredményének a standard outputon kell megjelennie
- Ezt továbbítja a webserverver a kliens felé



Környezeti változók – példa

13

```
[HTTP_HOST] => webprogramozas.inf.elte.hu
[HTTP_USER_AGENT] => Mozilla/5.0 (Windows NT 6.1; WOW64; rv:19.0) Gecko/20100101 Firefox/19.0
[HTTP_ACCEPT] => text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
[HTTP_ACCEPT_LANGUAGE] => hu-hu,hu;q=0.8,en-US;q=0.5,en;q=0.3
[HTTP_ACCEPT_ENCODING] => gzip, deflate
[HTTP_REFERER] => http://webprogramozas.inf.elte.hu/~gyozke/wf2/temp/
[HTTP_COOKIE] => WACID=1263304129000A11327866; __utma=32143338.519822639.1361532995.1362426295.1363347264.4;
__utms=32143338.1362426295.3.3.utmcsr=google|utmccn=(organic)|utmcmd=organic|utmctr=(not%20provided)
[HTTP_CONNECTION] => keep-alive
[PATH] => /usr/sbin:/bin:/usr/bin:/sbin
[SERVER_SIGNATURE] => <address>Apache/2.2.10 (Linux/SUSE) Server at webprogramozas.inf.elte.hu Port
80</address>
[SERVER_SOFTWARE] => Apache/2.2.10 (Linux/SUSE)
[SERVER_NAME] => webprogramozas.inf.elte.hu
[SERVER_ADDR] => 157.181.161.8
[SERVER_PORT] => 80
[REMOTE_ADDR] => 188.142.184.197
[DOCUMENT_ROOT] => /srv/www/webprog
[SERVER_ADMIN] => root@webprogramozas.inf.elte.hu
[SCRIPT_FILENAME] => /home/gyozke/public_html/wf2/temp/getpost.php
[REMOTE_PORT] => 49323
[GATEWAY_INTERFACE] => CGI/1.1
[SERVER_PROTOCOL] => HTTP/1.1
[REQUEST_METHOD] => GET
[QUERY_STRING] =>
[REQUEST_URI] => /~gyozke/wf2/temp/getpost.php
[SCRIPT_NAME] => /~gyozke/wf2/temp/getpost.php
[PHP_SELF] => /~gyozke/wf2/temp/getpost.php
[REQUEST_TIME] => 1365626159
```

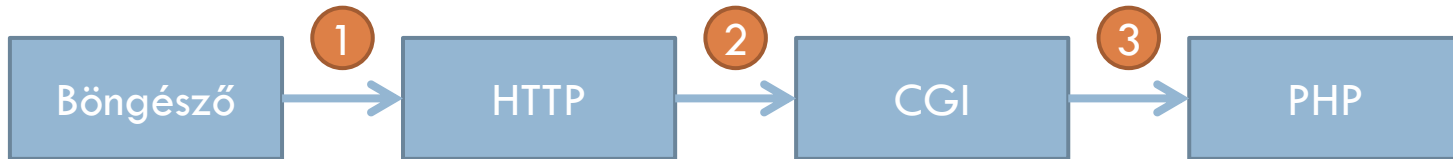
CGI összefoglalás

14

- HTTP kérés részei kerülnek átadásra a programnak
 - környezeti változók
 - standard bemenet
- Információ hol érkezhethet
 - **QUERY_STRING**
 - **HTTP üzenettörzs**

Kliensről érkező adatok

15



16

Böngésző és HTTP

Kérésindítási lehetőségek

17

- Cím beírása (GET)
- Linkre kattintás (GET)
- Űrlap elküldése (GET, POST)
- JavaScript (GET, POST)
 - ▣ oldal újratöltés (`location`, GET)
 - ▣ űrlap elküldés(`submit()` metódus, GET, POST)
 - ▣ AJAX kérés (bármilyen HTTP metódus)

Címsor és hivatkozások

18

- A megadott URL kerül elküldésre

```
<a href="http://server.hu/index.php?adat=ELTE">Valami</a>
```

- Általános formája

```
<scheme>://<host>:<port>/<path>?<query>#<fragment>
```

- Például

```
http://localhost/cgi-bin/cgi.exe  
http://webprogramozas.inf.elte.hu/~gyozke/getpost.php  
http://webprogramozas.inf.elte.hu/~gyozke/getpost.php?ELTE  
http://webprogramozas.inf.elte.hu/~gyozke/getpost.php?adat=ELTE  
https://tools.ietf.org/html/rfc3986#section-3.4  
http://www.w3.org/TR/REC-html40/interact/forms.html#h-17.13
```

<form> elem

19

□ Attribútumok

- action: szerveroldali erőforrás megjelölése
- method: HTTP metódus
- enctype: a formadatok kódolása a küldés során

□ Enctype

- application/x-www-form-urlencoded (alapértelmezett)
- multipart/form-data (fájlfeltöltés, csak POST esetén)
- text/plain (levélküldés)

Űrlap elküldése

20

- Elküldés
 - submit gomb megnyomása
 - JavaScript: `form.submit()` meghívása
- Elküldhető elemek kiválasztása
 - name attribútum (nagyon fontos)
 - nem disabled
 - bejelölt checkbox
 - bejelölt radio
 - kiválasztott select

Űrlap elküldése (folytatás)

21

- Kérési adatok összeállítása
 - Elküldhető formelemelekből név=érték párokat készít
 - ezeket & jellel fűzi össze
- Kérési adatok kódolása
 - Id. enctype attribútum
 - → Kérésszöveg (Query String)

Űrlap elküldése (folytatás)

22

- A kódolt kérési adatok elküldése
 - GET
 - az action-ben megadott URL mögé fűzi egy ?-lel
 - A kérésszöveg az URL adatokat tárolására szolgáló `<query>` részben jelenik meg
 - limitált méret az URL mérete miatt (2 kB)
 - POST
 - a HTTP kérés **üzenettörzs**ében utazik a kérésszöveg
 - sok adat küldhető
- <http://www.w3.org/TR/REC-html40/interact/forms.html#h-17.13>

Űrlap elküldése – példa

23

□ Űrlap:

```
<form action="http://localhost/cgi-bin/cgi.exe" method="post">  
  <input type="text" name="alma" value="piros">  
  <input type="password" value="kek">  
  <input type="hidden" name="jelszo" value="titkos">  
  <input type="checkbox" name="check1" value="ertek1" checked>  
  <input type="checkbox" name="check2" checked>  
  <input type="submit">  
</form>
```

□ Query string

```
alma=piros&jelszo=titkos&check1=ertek1&check2=on
```

Űrlapelemek

24

- Szöveges elemek
 - `<input>` type: text, password, hidden
 - `<textarea>`
 - mindig elküldésre kerülnek
 - akár üres szöveggel is
 - textareánál vigyázni kell: minden elküldésre kerül, ami a `<textarea></textarea>` elem között van

Űrlapelemek

25

- checkbox, radio
 - csak bejelölve kerül elküldésre
 - ha nincs value → „on” érték
- select
 - csak bejelölve kerül elküldésre
 - ha nincs value → option szövege az érték
 - size>1 esetén ha nincs kiválasztva → nem küldi el
 - multiple esetén → ugyanazon névvel több név=érték pár (pl. sel=alma&sel=szilva)

Űrlapelemek

26

- submit
 - ha van neve → elküldésre kerül az értékével (felirat)
- file
 - `enctype="multipart/form-data"`
 - `method="POST"`

Űrlapok

27

□ GET

- beírás, link: megfelelő URL „kézi” előállítása

 - <http://pelda.com/index.php?alma=piros&korte=sarga>

- form: automatikusan állnak elő a név=érték párok

 - `method="get", action="http://pelda.com/index.php"`

 - <http://pelda.com/index.php?alma=piros&korte=sarga>

□ POST

- form: automatikusan állnak elő a név=érték párok

- a kéreyszöveg az üzenettörzsben utazik

HTTP kérés

28

□ URL

```
<scheme>://<host>:<port>/<path>?<query>#<fragment>
```

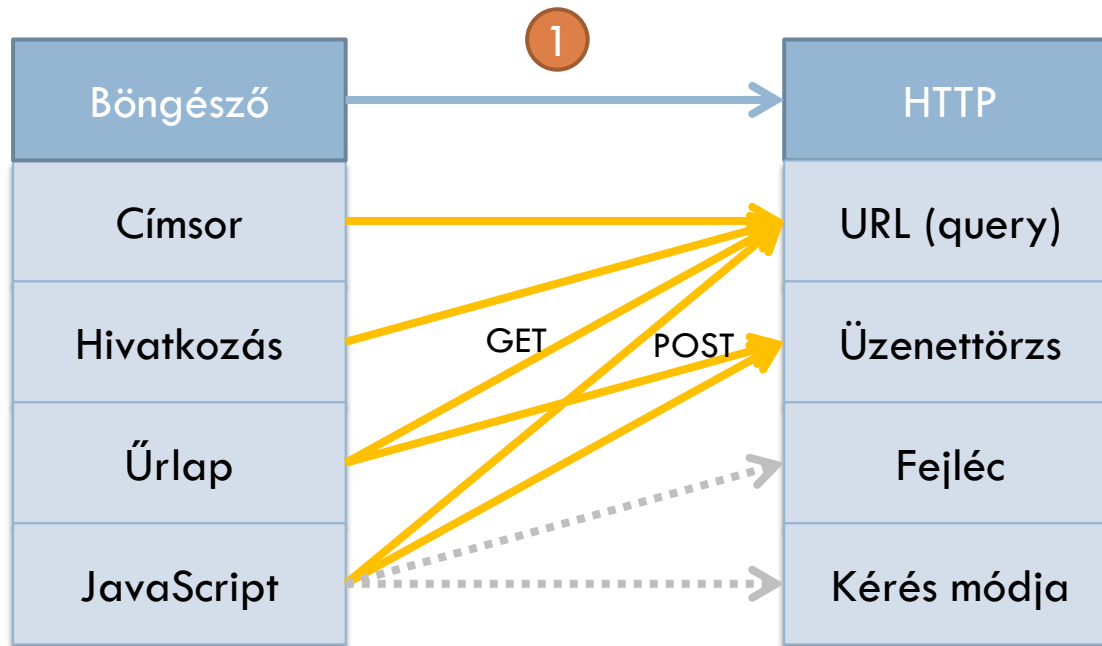
□ HTTP kérés

```
METÓDUS /<path>?<query> VERZIÓ  
Host: <host>  
FEJLÉC: ÉRTÉK  
FEJLÉC: ÉRTÉK  
FEJLÉC: ÉRTÉK  
  
ÜZENETTEST
```

Összefoglalás

29

- Kérésszöveg
 - URL <query> részében
 - Üzenettörzs



30

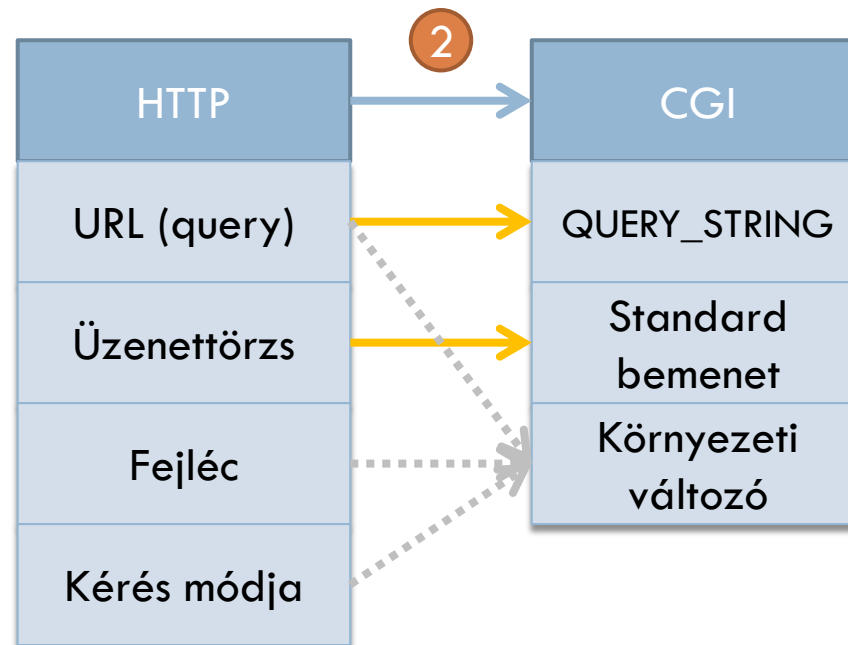
HTTP és CGI

HTTP kérés leképezése

31

□ Szabad paraméterek

- URL <query> része → QUERY_STRING
- Üzenettörzs → Standard bemenet



32

CGI programok

- CGI kommunikációja
 - ▣ környezeti változók
 - ▣ standard bemenet és kimenet
- Programoknak tudnia kell
 - ▣ környezeti változók értékeit lekérdezni
 - ▣ olvasni a standard bemenetről
 - ▣ írni a standard kimenetre

Adatok tipikus beolvasása

34

- REQUEST_METHOD → GET vagy POST
- GET
 - ▣ QUERY_STRING
- POST
 - ▣ CONTENT_LENGTH hosszan
 - ▣ standard inputról olvasni
 - ▣ QUERY_STRING is töltve lehet (URL függő)

Példa – C++ program

35

- Környezeti változó olvasása
 - `getenv(str)`
- Standard bemenetről olvasás karakterenként
 - `fgetc(stdin)`
- Szövegműveletek

Példa – C++ program

36

```
request_method = getenv("REQUEST_METHOD");
content_length_str = getenv("CONTENT_LENGTH");
content_length = atoi(content_length_str);
query_string_get = getenv("QUERY_STRING");
string query_string_post = "";

if (strcmp(request_method, "POST") == 0) {
    int db = 0;
    char c;
    while (db < content_length && EOF != (c = fgetc(stdin))) {
        query_string_post += c;
        db++;
    }
}
```


Leképezés

38

- Szabad paraméterek
 - ▣ QUERY_STRING → \$_GET
 - ▣ Standard bemenet → \$_POST
- Egyéb adatok
 - ▣ Környezeti változók → \$_SERVER
- Szuperglobális tömbök

```
//kéresszöveg:  
alma=piros&korte=sarga  
$_GET['alma']; // "piros"  
$_GET['korte']; // "sarga"
```

```
//kéresszöveg:  
alma=piros&korte=sarga  
$_POST['alma']; // "piros"  
$_POST['korte']; // "sarga"
```

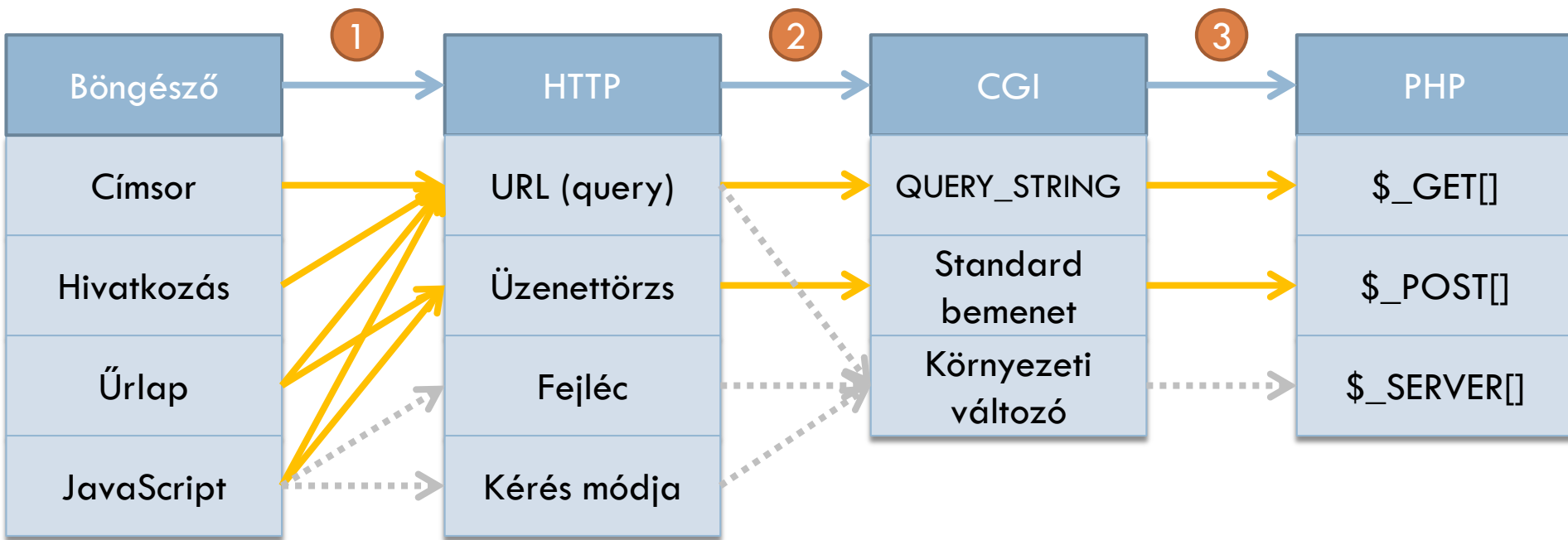
Szuperglobális tömbök

39

- `$_GET`: `QUERY_STRING`ben érkező név=érték pároknak megfelelő tömb, ahol a név az a kulcs.
- `$_POST`: a standard inputon érkező név=érték pároknak megfelelő tömb, ahol a név az a kulcs.
- `$_SERVER`: az összes környezeti változót tartalmazza
- `$_FILES`: a feltöltött fájlok helyét tartalmazza
- `$_COOKIES`: a kliensről érkező sütik jelennek meg benne

Összefoglalás

40



GET példa

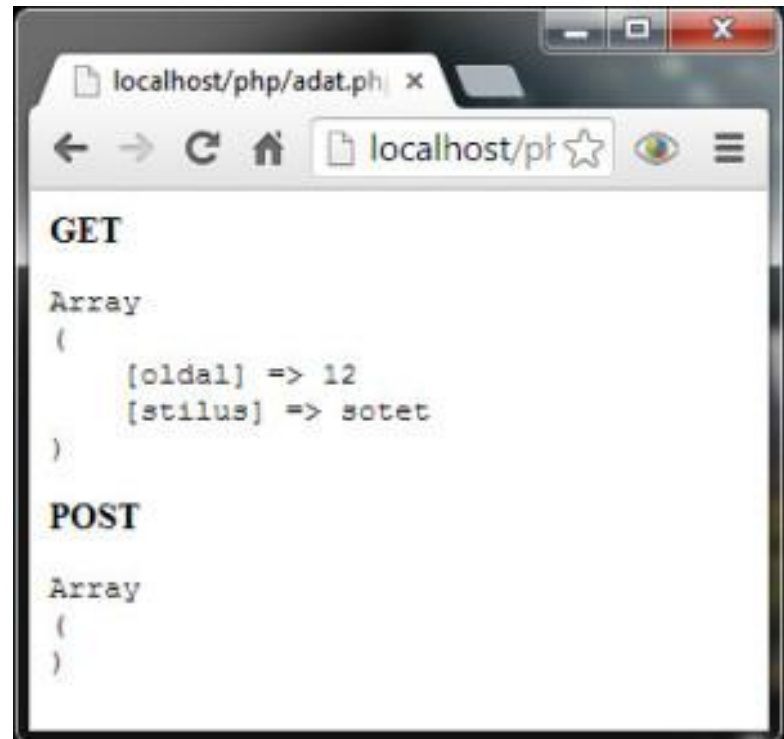
41

- Kliensen egy link

```
<a href="adat.php?oldal=12&stilus=sotet">Példa hivatkozás</a>
```

- PHP kód

```
function kiirTomb($tomb) {  
    echo '<pre>';  
    print_r($tomb);  
    echo '</pre>';  
}  
kiirTomb($_GET);  
kiirTomb($_POST);
```

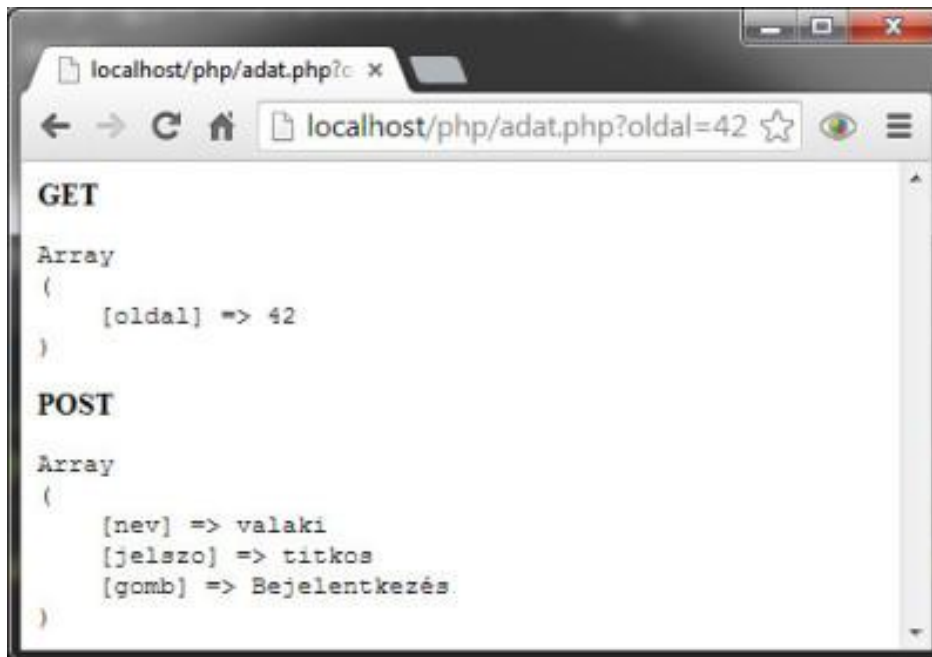


POST példa

42

□ Űrlap

```
<form action="adat.php?oldal=42" method="post">  
  Név: <input type="text" name="nev" value="valaki"> <br>  
  Jelszó: <input type="password" name="jelszo" value="titkos"> <br>  
  <input type="submit" name="gomb" value="Bejelentkezés">  
</form>
```



Bemeneti adatok feldolgozása

Példa

44

- Kör kerületének kiszámítása
- Input: sugár
 - ▣ Kötelező
- Output: kerület
 - ▣ Valid oldalon jelenjen meg

Elvárt kimenet

45

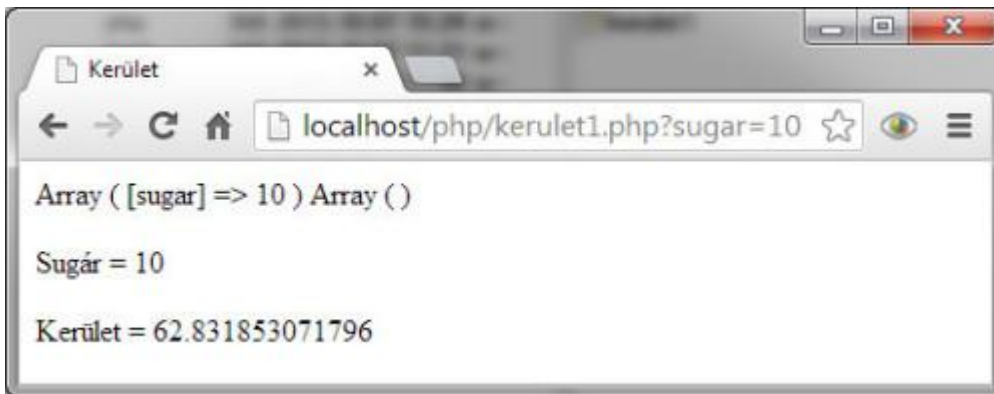
```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Kerület</title>
  </head>
  <body>
    <p>Sugár = 10</p>
    <p>Kerület = 62.83</p>
  </body>
</html>
```

1. megoldás

46

- Az URL-ben adjuk át paraméterként

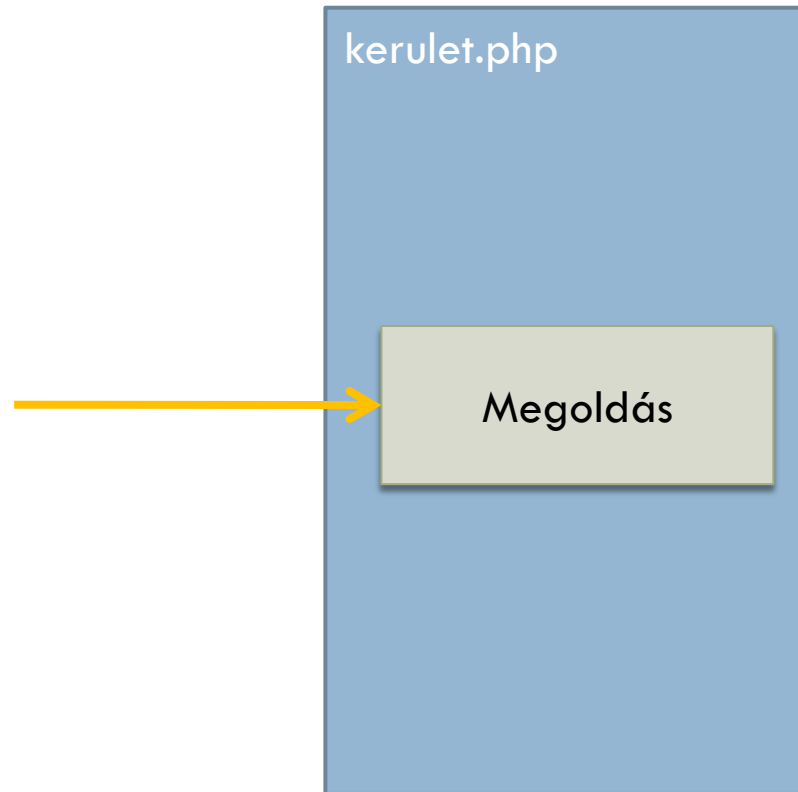
`http://pelda.hu/kerulet.php?sugar=10`



```
<?php  
print_r($_GET);  
print_r($_POST);  
?>  
<!doctype html>  
<html>  
  <head>  
    <meta charset="utf-8">  
    <title>Kerület</title>  
  </head>  
  <body>  
    <?php  
    //beolvasás  
    $sugar = $_GET['sugar'];  
    //feldolgozás  
    $pi = pi();  
    $ker = 2 * $sugar * $pi;  
    //kiiras  
    echo "<p>Sugár = {$sugar}</p>";  
    echo "<p>Kerület = {$ker}</p>";  
    ?>  
  </body>  
</html>
```

1. megoldás

47



2. megoldás

48

- Feldolgozás kivétele a kiírásból
- Kiírás és PHP logika szétválasztása

```
<?php
print_r($_GET);
print_r($_POST);
//beolvasás
$sugar = $_GET['sugar'];
//feldolgozás
$pi = pi();
$ker = 2 * $sugar * $pi;
//kiiras
?>
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Kerulet</title>
  </head>
  <body>
    <?php
echo "<p>Sugár = {$sugar}</p>";
echo "<p>Kerulet = {$ker}</p>";
?>
  </body>
</html>
```


3. megoldás

49

- Kiírás csiszolása
- Csak a valóban dinamikus tartalom legyen PHP-ban

```
<?php
print_r($_GET);
print_r($_POST);
//beolvasás
$sugar = $_GET['sugar'];
//feldolgozás
$pi = pi();
$ker = 2 * $sugar * $pi;
//kiiras
?>
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Kerület</title>
  </head>
  <body>
    <p>Sugár = <?php echo $sugar; ?></p>
    <p>Kerület = <?php echo $ker; ?></p>
  </body>
</html>
```

HTML sablon (alternatív szintaxis)

50

□ Kiírás

```
<?php echo $változó; ?>
```

□ Elágazás

```
<?php if (felt) : ?>  
HTML sablon kód  
<?php else : ?>  
HTML sablon kód  
<?php endif; ?>
```

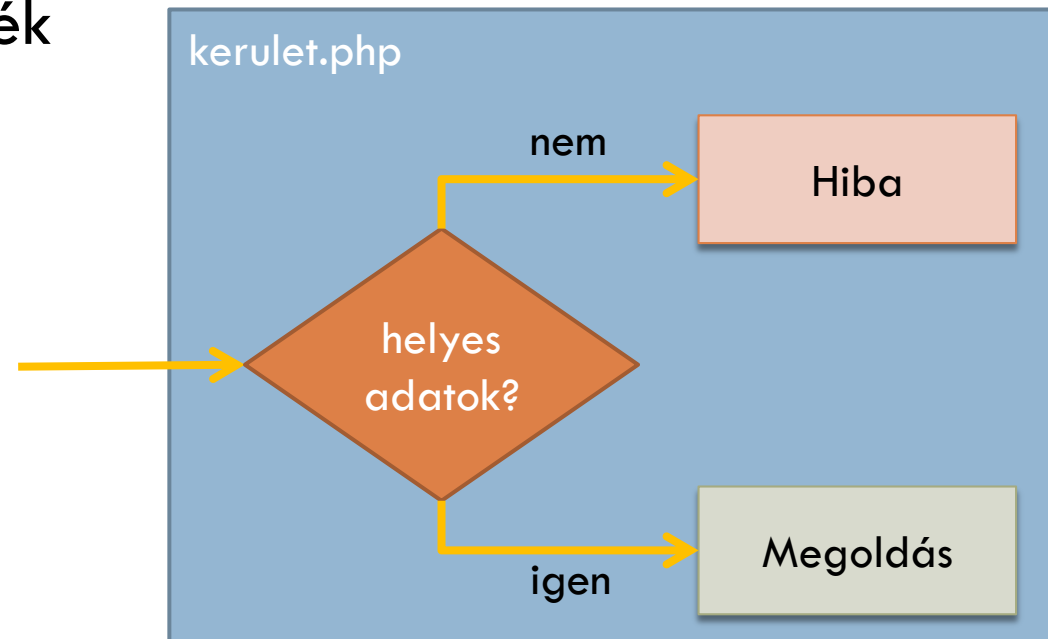
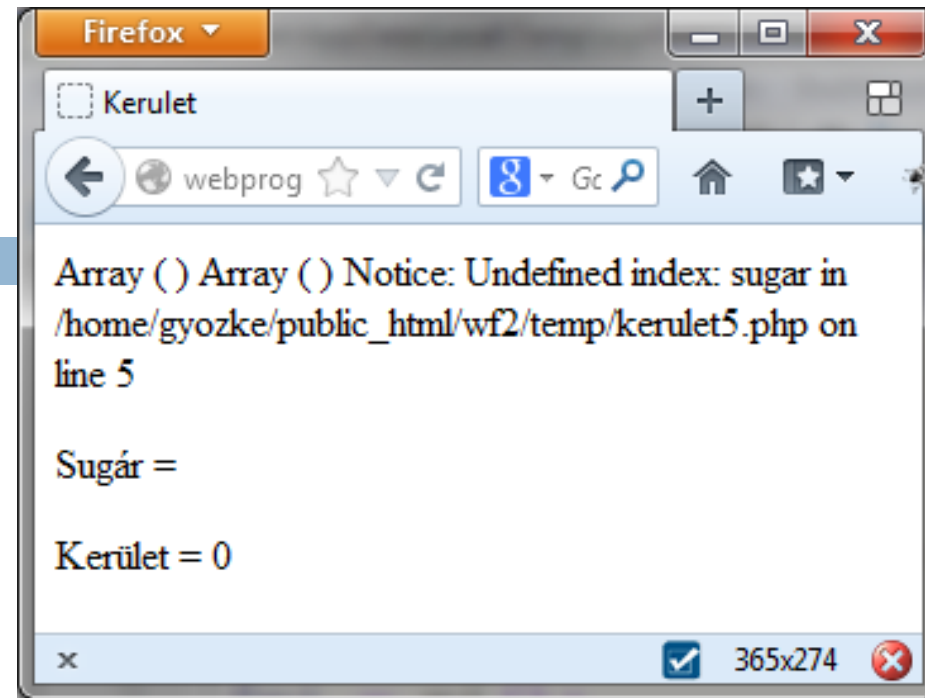
□ Ciklus

```
<?php foreach ($tomb as $valt) : ?>  
HTML sablon kód  
<?php endforeach; ?>
```

Input hiánya

51

- <http://pelda.hu/kerulet.php>
- Kezeleni kell az input hiányát
 - ▣ Alapértelmezett érték
 - ▣ Hibaüzenet



4. megoldás

52

□ Alapértelmezett érték

```
//beolvasás  
$sugar = 0;  
if (isset($_GET['sugar'])) {  
    $sugar = $_GET['sugar'];  
}
```

```
//beolvasás  
$sugar = (isset($_GET['sugar'])  
    ? $_GET['sugar']  
    : 0);
```

5. megoldás

53

- Hibaüzenet
- Kimenet

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Kerulet</title>
  </head>
  <body>
    <?php if ($hiba) : ?>
      <p><?php echo $hiba; ?></p>
    <?php else : ?>
      <p>Sugár = <?php echo $sugar; ?></p>
      <p>Kerület = <?php echo $ker; ?></p>
    <?php endif; ?>
  </body>
</html>
```

6. megoldás

54

- Hibaüzenet
- Beolvasás
- Ellenőrzés
 - ▣ megadták?
 - ▣ kitöltötték?
 - ▣ szám?
- Több input esetén
 - ▣ hibalista (PHP tömb)
 - ▣ hibalista (HTML ul)

```
//beolvasás
$sugar = null;
if (isset($_GET['sugar'])) {
    $sugar = $_GET['sugar'];
}
//előfeltétel ellenőrzése
$hiba = '';
if (is_null($sugar)) {
    $hiba = 'Sugár hiányzik!';
}
else if ($sugar == '') {
    $hiba = 'Sugár üres!';
}
else if (!is_numeric($sugar)) {
    $hiba = 'Sugár nem szám!';
}
if (!$hiba) {
    //feldolgozás
    $pi = pi();
    $ker = 2 * $sugar * $pi;
}
```

56

Ürlapok

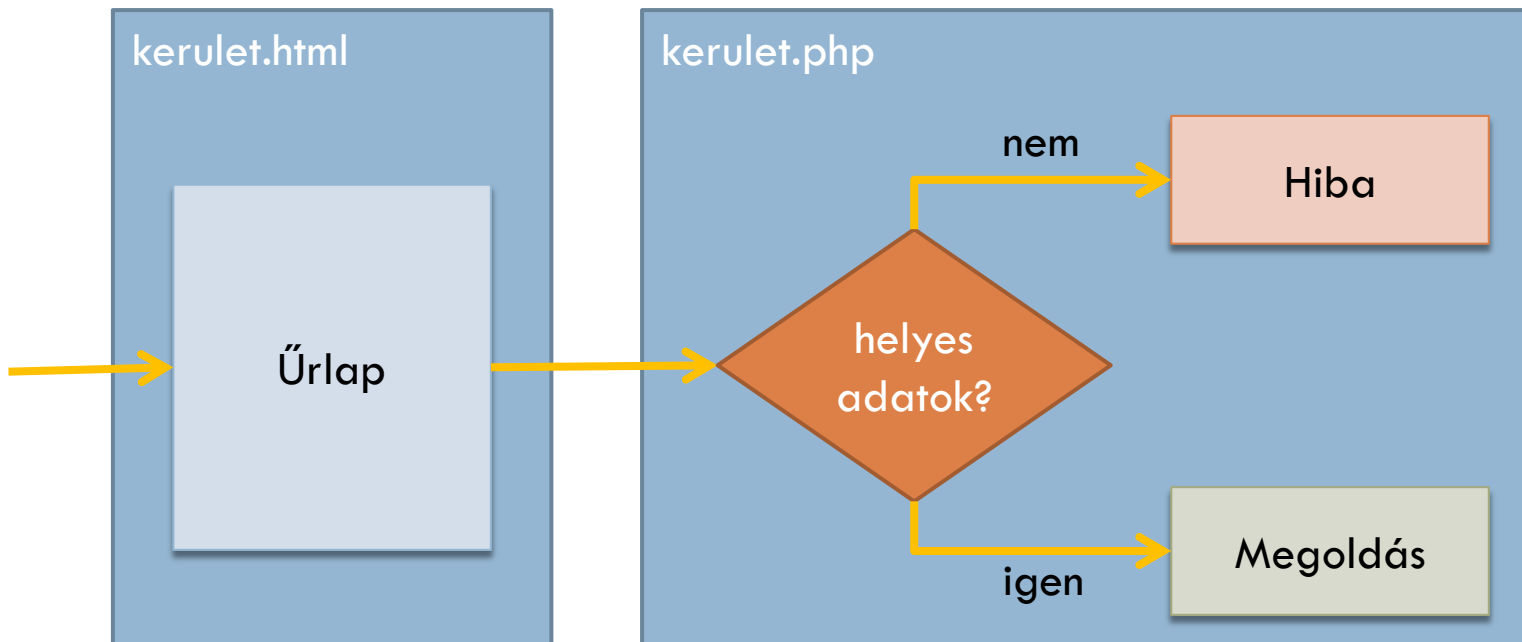
Űrlapok

57

- Előző megoldás nem felhasználóbarát
 - ▣ böngésző címsora
 - ▣ URL
- Input megadás a felhasználó részéről általában űrlapokon keresztül történik

Tervezés

58



Ürlap

59

- kerulet.html
- statikus oldal

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Kerulet</title>
  </head>
  <body>
    <form action="kerulet.php" method="get">
      Sugár: <input type="text" name="sugar">
      <input type="submit">
    </form>
  </body>
</html>
```

Feldolgozó szkript

60

□ kerulet.php (ld. korábban)

```
<?php
print_r($_GET);
print_r($_POST);
//beolvasás
$sugar = $_GET['sugar'];
//előfeltétel ellenőrzése
$hiba = '';
if (is_null($sugar)) {
    $hiba = 'Sugár hiányzik!';
}
else if ($sugar == '') {
    $hiba = 'Sugár üres!';
}
else if (!is_numeric($sugar)) {
    $hiba = 'Sugár nem szám!';
}
if (!$hiba) {
    //feldolgozás
    $pi = pi();
    $ker = 2 * $sugar * $pi;
}
//kiiras
?>
```

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Kerulet</title>
  </head>
  <body>
    <?php if ($hiba) { ?>
    <p><?php echo $hiba; ?></p>
    <?php } else { ?>
    <p>Sugár = <?php echo $sugar; ?></p>
    <p>Kerület = <?php echo $ker; ?></p>
    <?php } ?>
    <a href="kerulet.html">Vissza az úrlapra</a>
  </body>
</html>
```

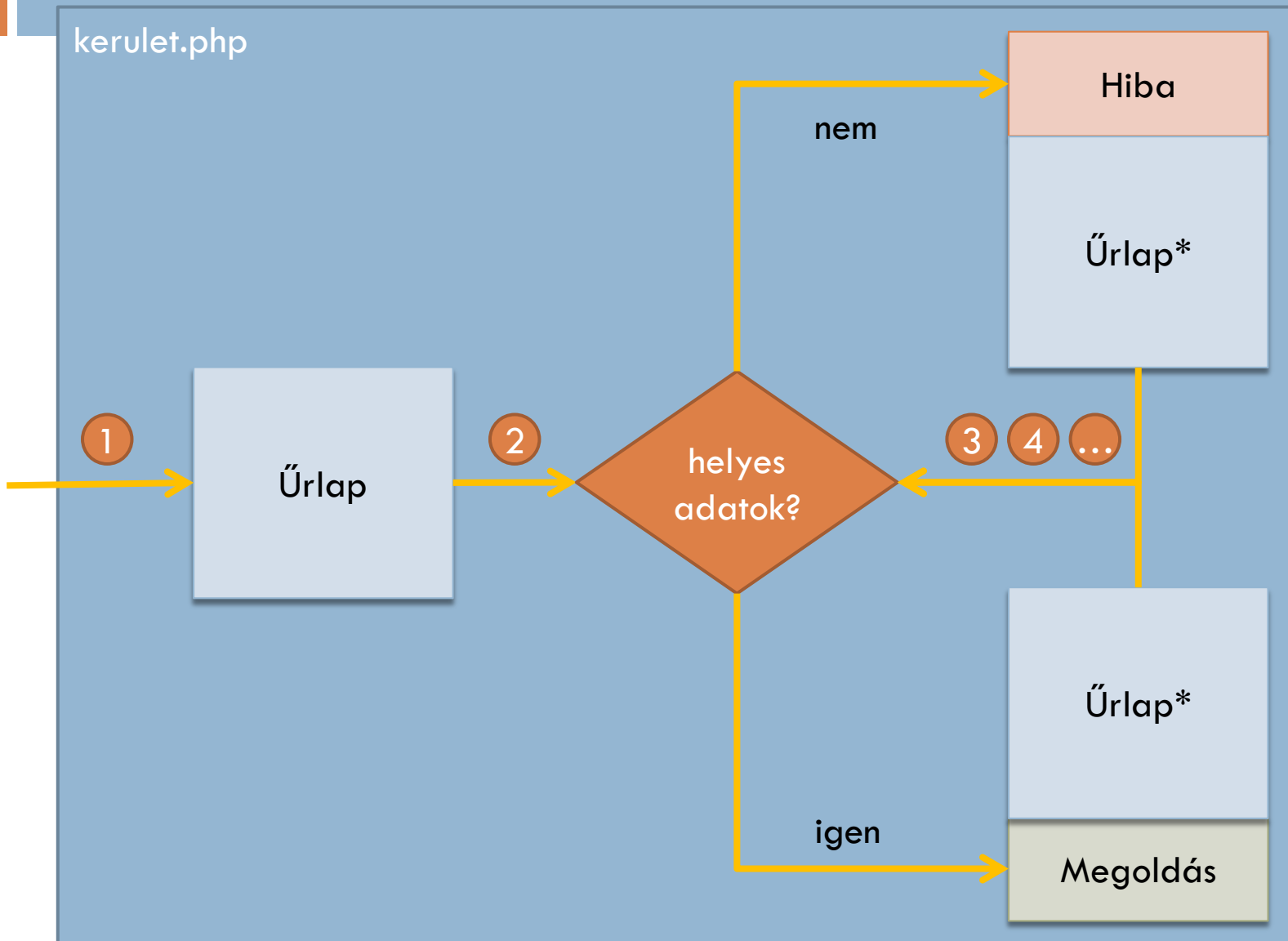
Ergonómiai megfontolások

61

- Az alkalmazás funkcionálisan teljesíti az elvárásokat
- De nem ergonomikus a használata
 - ▣ Felhasználó nem tudja, mit írt be
 - ▣ Külön kattintásra (vissza) tud
 - javítani
 - újra számolni
- Jó lenne ott jelezni a hibát, ahol az űrlap van
- Jó lenne az űrlapon jelezni a megoldást is

Tervezés – oldalak

62



Tervezés – állomány szint

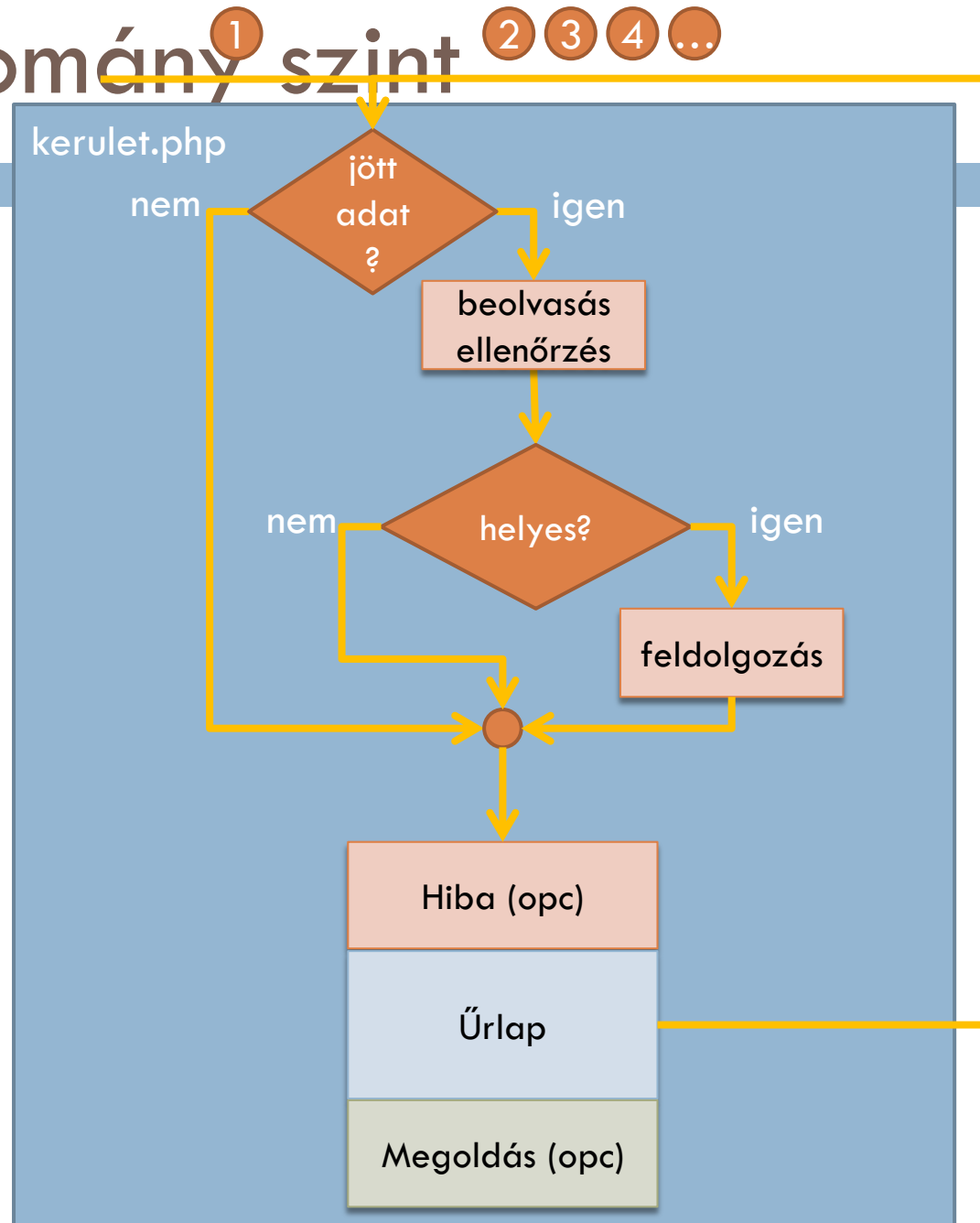
63

□ PHP kód

- 1: nincs input
 - kiírás (űrlap)
- 2, 3, 4, ...:
 - beolvasás
 - ellenőrzés
 - feldolgozás
 - kiírás (űrlap)

□ Eldöntés

- Van-e POST adat
- REQUEST_METHOD



HTML rész

64

□ POST-ra átírva!

```
<body>
  <?php if ($hiba) : ?>
    <p><?php echo $hiba; ?></p>
  <?php endif; ?>

  <form action="kerulet.php" method="post">
    Sugár: <input type="text" name="sugar">
    <input type="submit">
  </form>

  <?php if (isset($ker)) : ?>
    <p>Sugár = <?php echo $sugar; ?></p>
    <p>Kerület = <?php echo $ker; ?></p>
  <?php endif; ?>
</body>
```

Feldolgozó rész

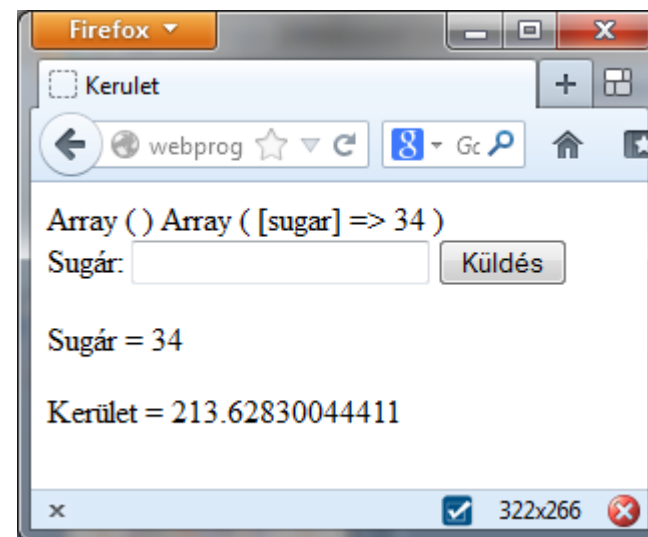
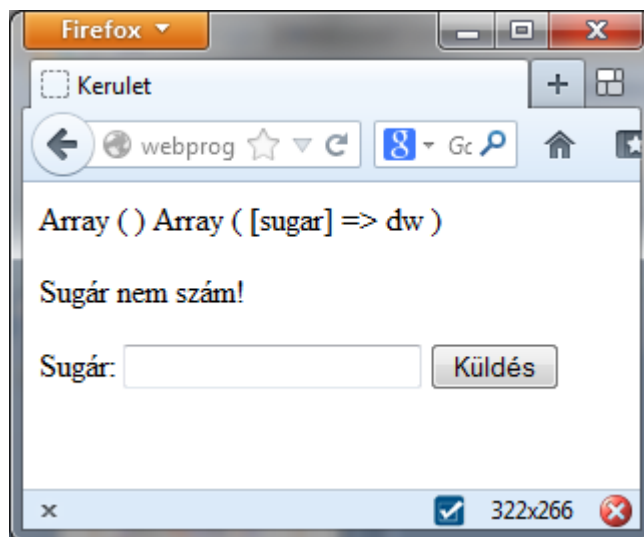
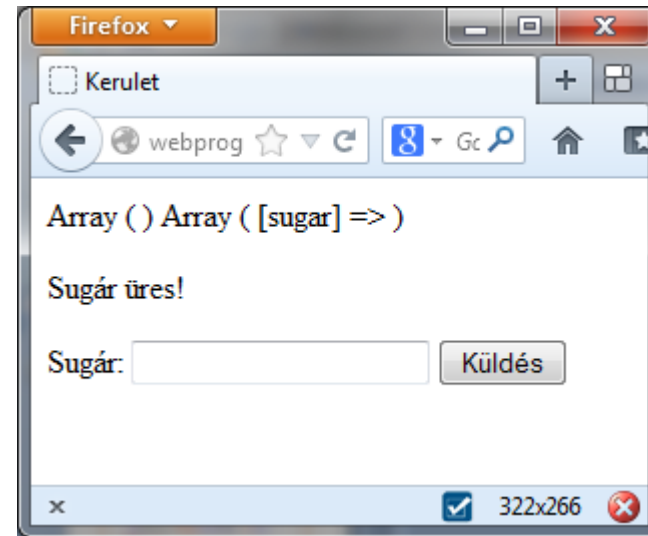
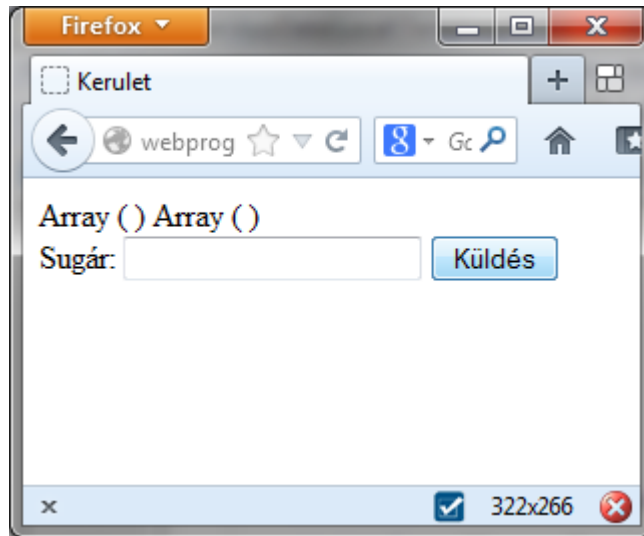
65

- Lényeg:
 - ▣ `if ($_POST) {...}`
- `$hiba` kiemelve az elágazás elé

```
$hiba = '';  
if ($_POST) {  
    //beolvasás  
    //...  
    //előfeltétel ellenőrzése  
    //...  
    if (!$hiba) {  
        //feldolgozás  
        //...  
    }  
}
```


Képernyőképek

66



Űrlap állapotartása

67

- Írjuk vissza a kitöltött adatokat, legalább hiba esetén!

```
<form action="kerulet.php" method="post">  
  Sugár: <input type="text" name="sugar"  
    value="<?php echo $sugar; ?>">  
  <input type="submit">  
</form>
```

```
$hiba = '';  
$sugar = '';  
if ($_POST) {  
  //beolvasás  
  $sugar = $_POST['sugar'];  
  //...  
}
```

Feldolgozó rész kiemelése

68

- Függvénybe zárjuk a lényegi részt
- Elkülönítjük az adatok beolvasásától és kiírásától

```
function kerulet($sugar) {
    $pi = pi();
    $ker = 2 * $sugar * $pi;
    return $ker;
}
//...
if ($_POST) {
    //...
    if (!$hiba) {
        //feldolgozás
        $ker = kerulet($sugar);
    }
}
```

Szkript szerkezete

69

- Három nagy részből áll
 - ▣ Adatszerkezetek és ezeket feldolgozó függvények
 - ▣ A kérés fogadásáért, beolvasásért, ellenőrzésért és kiírásért felelős rész
 - ▣ Megjelenítendő HTML sablonja

```
<?php
//Adatok és függvények
//-----
//Beolvasás
//Ellenőrzés
//Feldolgozás meghívása
//-----
?>
<!doctype html>
<html>
<!-- HTML kód -->
<html>
```

Összefoglalás

70

- Adatok átadása
 - URL ? utáni része → `$_GET` tömbbe kerül
 - formok, name attribútum az elemeknél
 - → `$_GET`, `$_POST` tömbökbe, ahol a name attribútum a kulcs
- Űrlapok feldolgozása
 - Önmaguknak küldik
 - `if ($_POST)` megkülönböztetni az első hívást a többitől
- Minden szkript három részből áll: adat, beolv, HTML