WEBFEJLESZTÉS 2. – PHP NYELVI ALAPOK

Horváth Győző

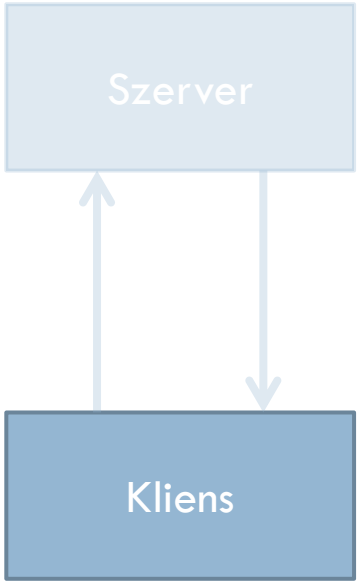
Egyetemi adjunktus

1117 Budapest,
Pázmány Péter sétány 1/C, 2.420
Tel: (1) 372-2500/1816

Szerveroldali dinamizmus

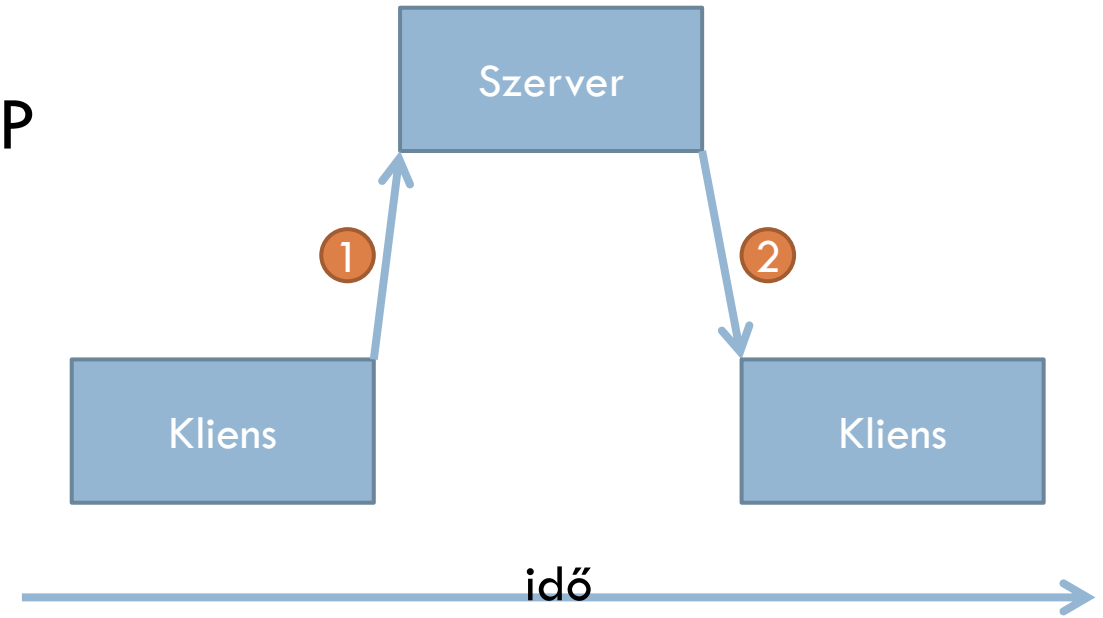
Kliensoldali webprogramozás

- Kliens-szerver architektúra
- Statikus kliens
- Dinamikus kliens
- Nem érdekes a szerver, csak az onnan érkező tartalom
- Nem is kell szerver, dolgozhatunk lokálisan
- Programozás a böngészőben
- JavaScript, DOM, stb.



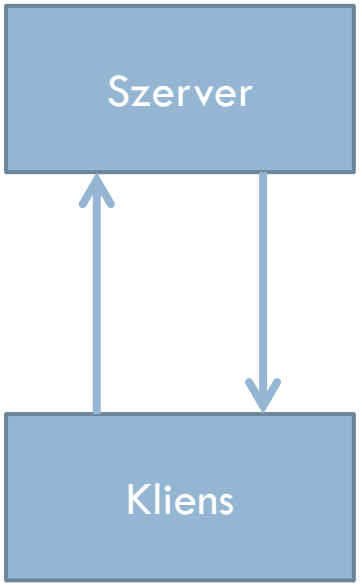
Kliens-szerver architektúra

- Érdekes a szerver is, sőt most vele foglalkozunk
- Két komponens: szerver és kliens
  - mindkettőre szükség van
  - térben és időben elválhatnak
- Csatorna: hálózat
- Kommunikáció: HTTP



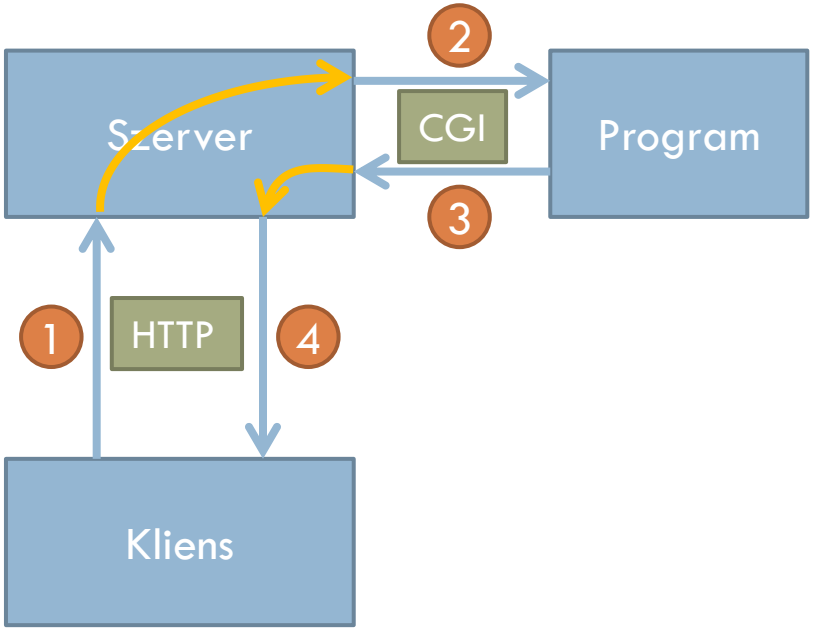
Statikus állományok

- Kérés pillanatában a szerveren megtalálható az a tartalom, amely leküldésre kerül a válaszban
- Fájlkiszolgálás
- Kiterjesztés alapján
  - .html
  - .jpg, .png, .gif
  - .css
  - .js



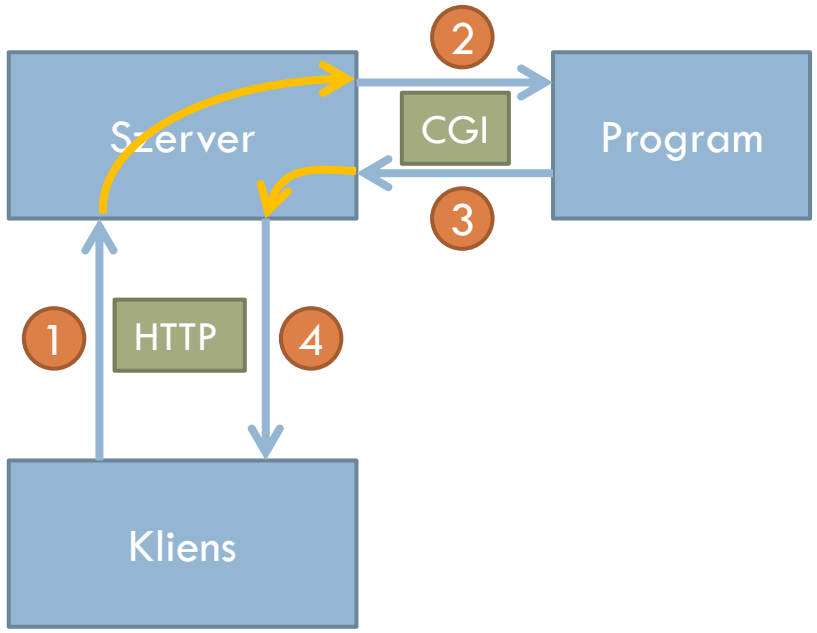
Szerveroldali webprogramozás

- Az állományok nem egyszerűen visszaküldésre kerülnek
- Szerver egy programnak adja át a vezérlést
- A leküldendő tartalmat egy program állítja elő.
- A program kimenetét a szerver elküldi a kliensnek.



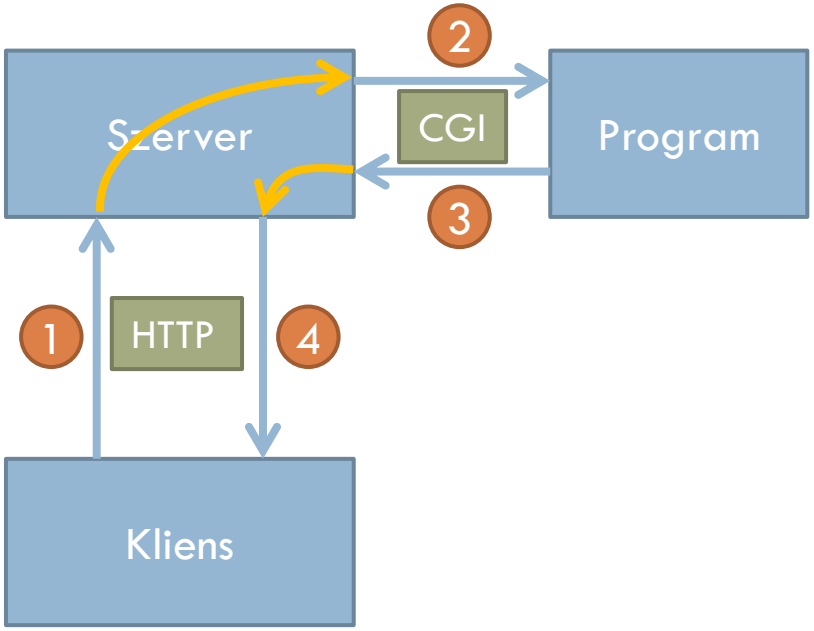
Hogyan indul el a program?

- Mi alapján dönti el a webszerver?
  - útvonal: cgi-bin könyvtár
    - futtatható állományok csak ezen belül
  - kiterjesztés
    - .cgi
    - .php



Milyen program lehet?

- Bármilyen program lehet
- Bináris
  - C++
  - Freepascal
- Szkript
  - Shell szkript
  - Perl
  - PHP
  - Python



Szerveroldali webprogramozás

- Program célja, kimenete
  - HTML generálás (általában)
  - HTTP protokoll betartásával
- Program helyességét a generált tartalom, a megkapott oldal forrása alapján ellenőrizhetjük.

HTTP protokoll

HTTP protokoll

- Kérés-válasz alapú protokoll a kliens és szerver között
- Mindig a kliens kezdeményez
- Kliens: kérés
  - kérést küld a 80-as TCP portra
  - jellemzően böngésző (hivatkozások, formok)
- Szerver: válasz
- TCP/IP réteg feletti protokoll

HTTP protokoll

- W3C szabvány
- RFC-k
  - 0.9 verzió (1991)
  - 1.0 verzió (1996) RFC 1945
  - 1.1 verzió (1999) RFC 2086, RFC2616
- http://tools.ietf.org/html/rfc2616
- http://www.w3.org/Protocols/rfc2616/rfc2616.html

HTTP kérés


```
METÓDUS ERŐFORRÁS VERZIÓ
FEJLÉC: ÉRTÉK
FEJLÉC: ÉRTÉK
FEJLÉC: ÉRTÉK

ÜZENETTEST (opcionális)
```

HTTP kérés példa


```
GET /index.html HTTP/1.1
Host: webprogramozas.inf.elte.hu
```

```
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=159741371.1255432553.1308299517.1308299517.1308299517.1;
__utma=32143338.2145495546.1326532899.1361177845.1362134456.25;
__utmz=32143338.1361177845.24.12.utmcsr=google|utmccn=(organic)|utmcmd=organic
|utmctr=(not%20provided)
Connection: keep-alive
```

HTTP kérés – metódusok

- GET
  - Megadott erőforrás letöltése
- POST
  - Feldolgozandó adat felküldése
- HEAD
  - Ld. GET, de csak a válasz fejléceket kéri le
- PUT
  - Feltölt a megadott erőforrást
- DELETE
  - Törli a megadott erőforrást

HTTP kérés – metódusok

- TRACE
  - Visszaküldi a kapott kérést
- OPTIONS
  - A szerver által támogatott HTTP metódusok listája
- CONNECT
  - Kérést transzparens tunnellé alakítja (SSL-hez kell)

HTTP válasz


```
VERZIÓ STÁTUSZKÓD INDOKLÁS
FEJLÉC: ÉRTÉK
FEJLÉC: ÉRTÉK
FEJLÉC: ÉRTÉK

ÜZENETTEST (opcionális)
```

HTTP válasz – példa


```
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>
```

HTTP válasz – státuszsor

- Státuszkód
  - 1xx: Informatív (kérés megkapva)
  - 2xx: Siker (kérés megérkezett, elfogadva)
    - 200 OK
  - 3xx: Átirányítás (további műveletre van szükség)
  - 4xx: Kliens hiba (kérés hibás, nem teljesíthető)
    - 404 Not found
    - 404 Nem található
  - 5xx: Szerver hiba (nem tudja teljesíteni a kérést)
    - 500 Internal Server Error

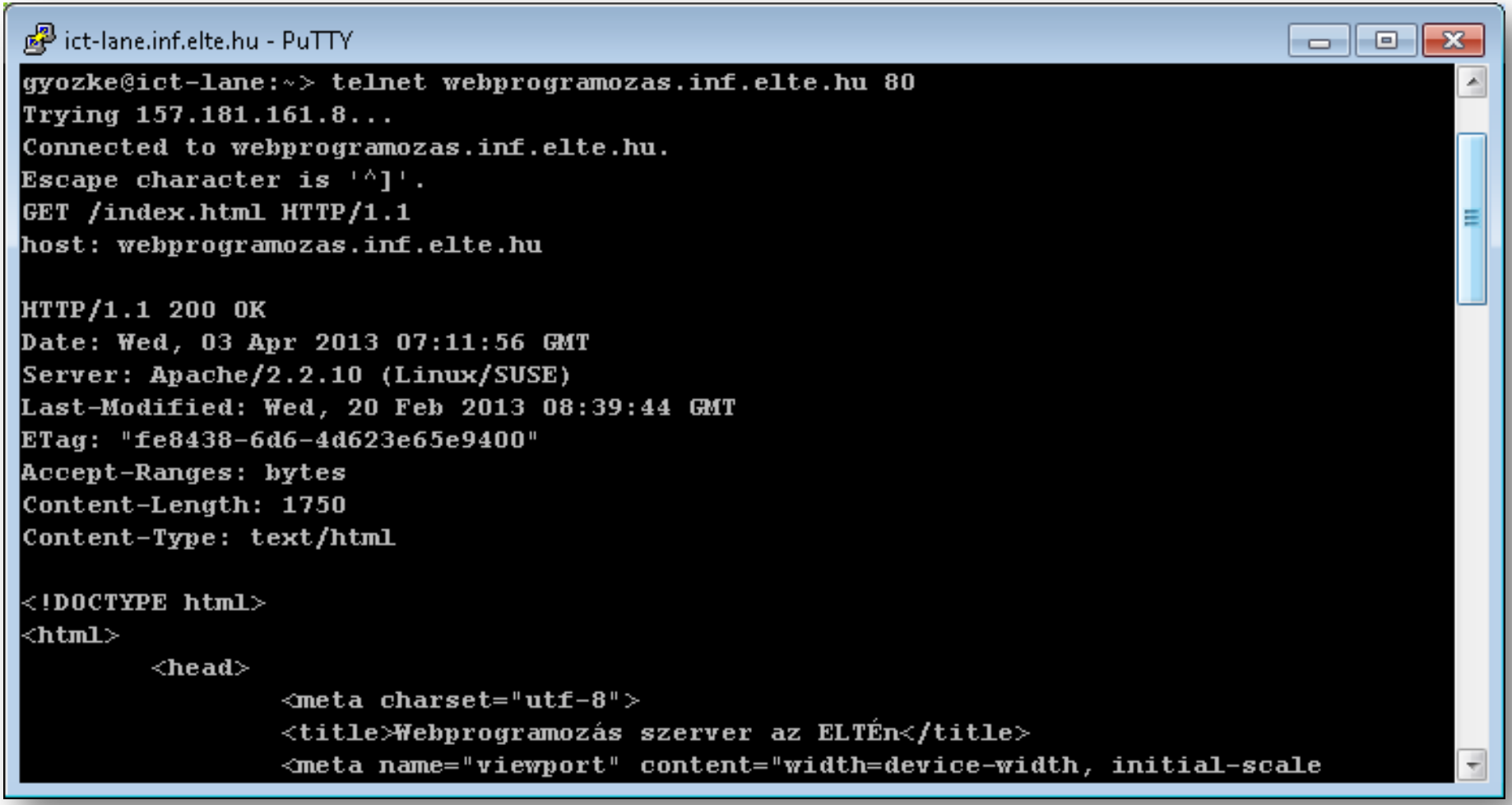
HTTP – eszközök

- telnet
  - TCP alapú, kétirányú, általánosan elérhető, nyolcbites byte-alapú kommunikációs protokoll
  - Kérést tudunk vele küldeni tetszőleges TCP portra

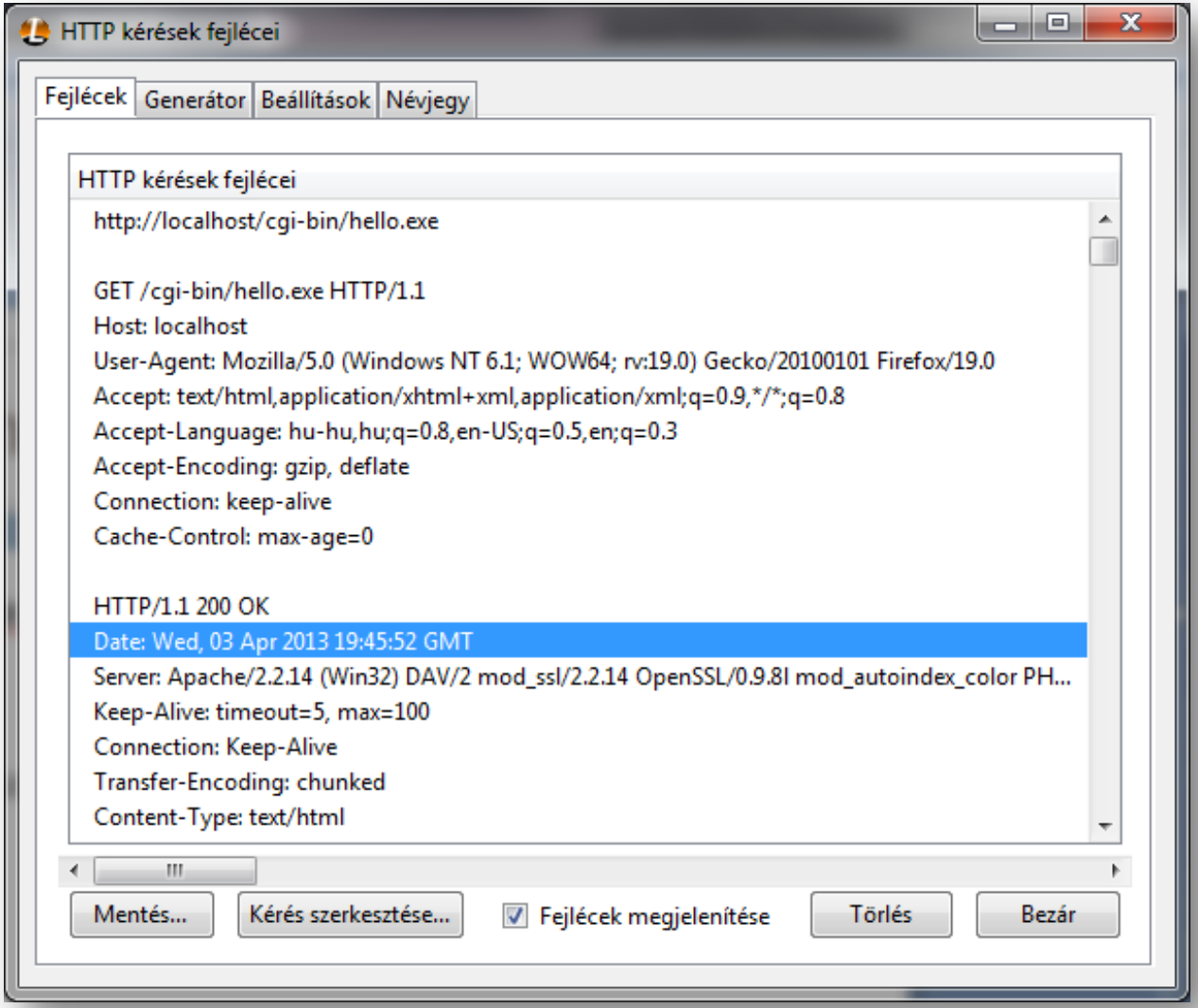
```
> telnet webprogramozas.inf.elte.hu 80
HTTP kérés sorai (beírva, bemásolva)

HTTP válasz sorai
```

- Böngésző kiegészítők
  - Live HTTP headers
  - HTTPFox

HTTP – eszközök példa



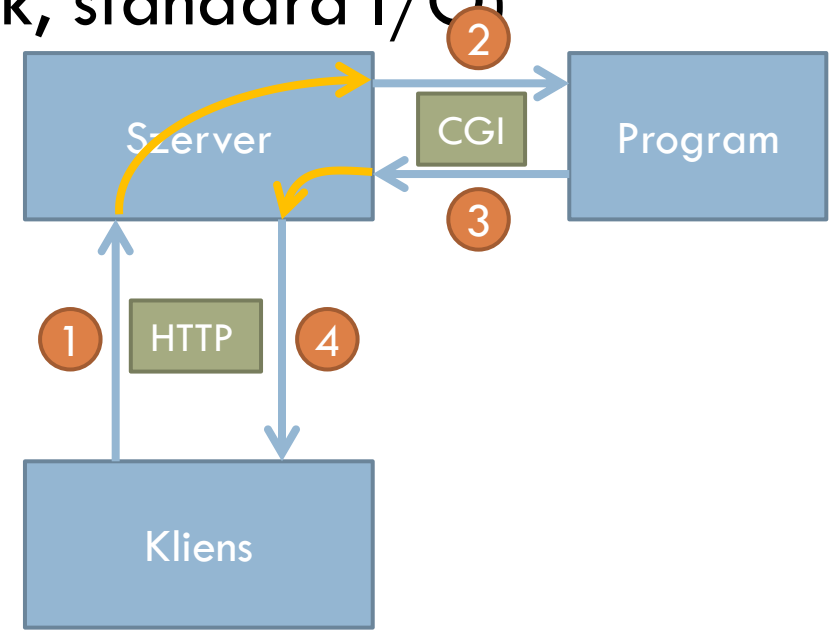
HTTP – eszközök példa



Common Gateway Interface

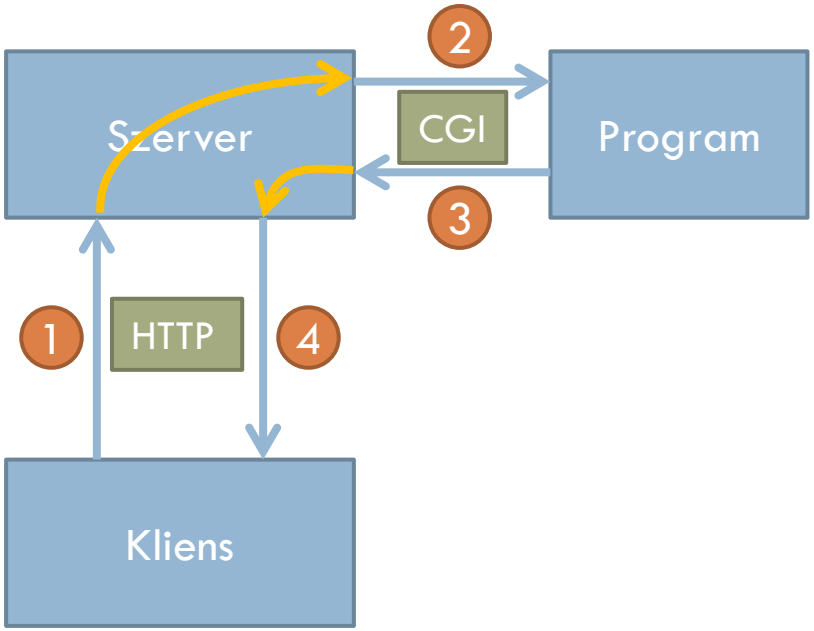
Common Gateway Interface (CGI)

- Azt határozza meg, hogy egy webszerver 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



A program eredménye

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



CGI példaprogramok

Példa – C++


```cpp
#include <iostream>
using namespace std;
int main()
{
    cout << "Content-Type: text/html" << endl;
    cout << endl;

    cout << "<!doctype html>" << endl;
    cout << "<html>" << endl;
    cout << "    <head>" << endl;
    cout << "        <meta charset=\"utf-8\">" << endl;
    cout << "        <title></title>" << endl;
    cout << "    </head>" << endl;
    cout << "    <body>" << endl;
    cout << "        <p>Hello vilag!</p>" << endl;
    cout << "    </body>" << endl;
    cout << "</html>" << endl;

    return 0;
}
```

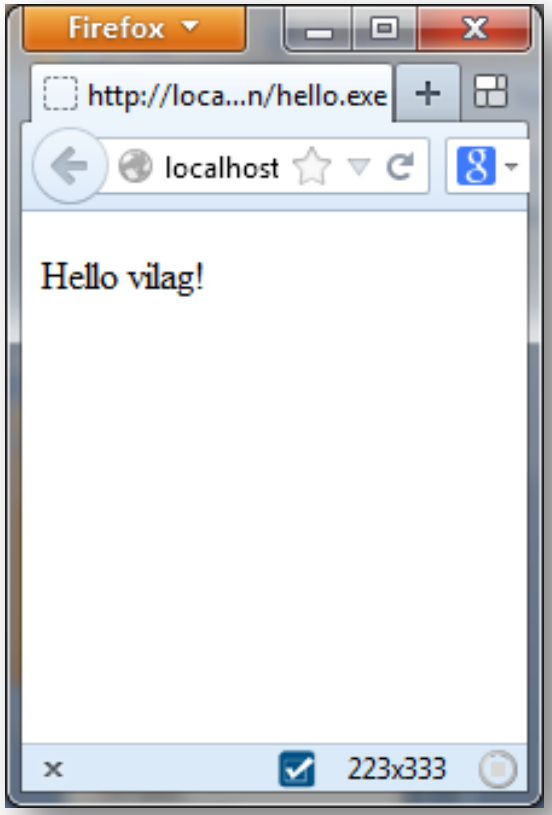
Példa – C++

- Egyszerűen futtatva

```
Content-Type: text/html

<!doctype html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
    </head>
    <body>
        <p>Hello vilag!</p>
    </body>
</html>
```

Példa – C++

- Böngészőből futtatva
  - http://localhost/cgi-bin/hello.exe

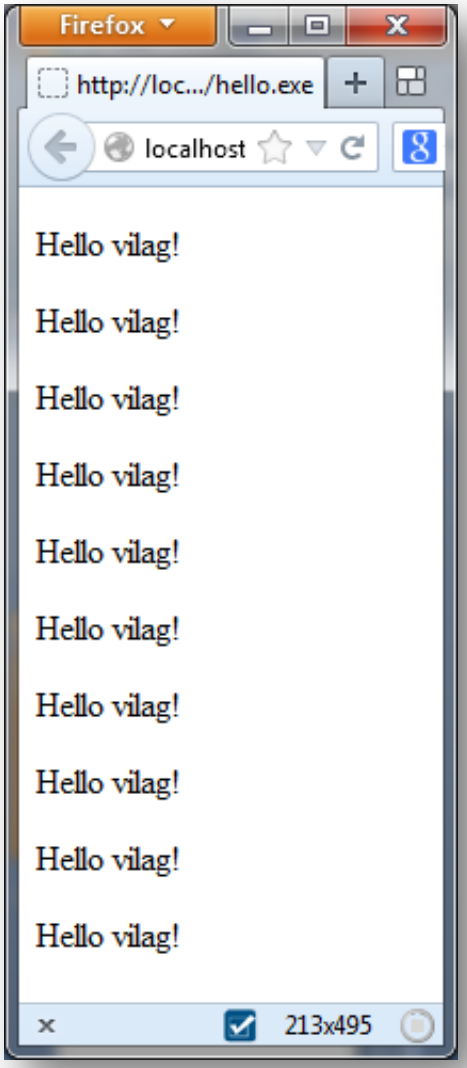


Példa – C++


```cpp
int main()
{
    cout << "Content-Type: text/html" << endl;
    cout << endl;

    cout << "<!doctype html>" << endl;
    cout << "<html>" << endl;
    cout << "    <head>" << endl;
    cout << "        <meta charset=\"utf-8\">" << endl;
    cout << "        <title></title>" << endl;
    cout << "    </head>" << endl;
    cout << "    <body>" << endl;

    for (int i = 1; i<=10; i++) {
        cout << "        <p>Hello vilag!</p>" << endl;
    }

    cout << "    </body>" << endl;
    cout << "</html>" << endl;
    return 0;
}
```

Példa – C++

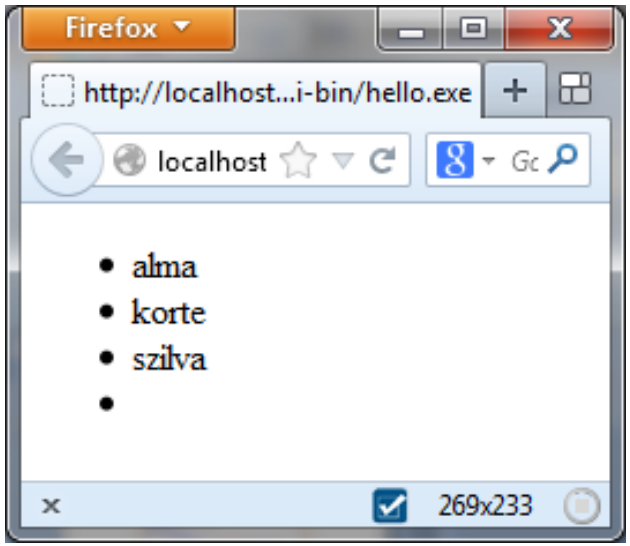

```
Content-Type: text/html

<!doctype html>
<html>
    <head>
        <meta charset="utf-8">
        <title></title>
    </head>
    <body>
        <p>Hello vilag!</p>
        <p>Hello vilag!</p>
        <p>Hello vilag!</p>
        <p>Hello vilag!</p>
        <p>Hello vilag!</p>
        <p>Hello vilag!</p>
        <p>Hello vilag!</p>
        <p>Hello vilag!</p>
        <p>Hello vilag!</p>
        <p>Hello vilag!</p>
    </body>
</html>
```

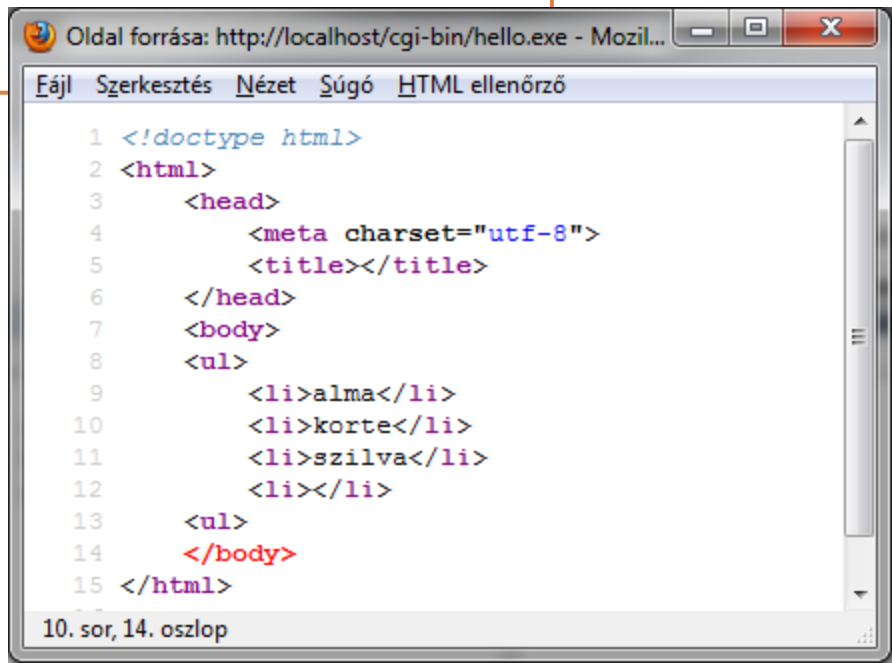


Példa – C++


```
cout << "    <ul>" << endl;
ifstream f("lista.txt");
while (!f.eof()) {
    string sor;
    getline(f, sor);
    cout << "        <li>" << sor << "</li>" << endl;
}
f.close();
cout << "    <ul>" << endl;
```

```
alma
korte
szilva
```





PHP nyelvi alapok

PHP

- PHP
  - Personal Home Page
  - PHP: Hypertext Preprocesszor
- Jellemzői
  - nyílt forráskódú
  - általános célú
  - szkriptnyelv
  - HTML-be ágyazható

PHP hivatkozások

- Főoldal
  - http://www.php.net/
- Referencia
  - http://www.php.net/manual/en/
  - függvényreferencia
  - nyelvi referencia
  - kódolási tanácsok
  - telepítés

PHP használata

- Általános célú szkriptnyelv, nincsen a webhez kötve
- Szerveroldali dinamikus webprogramozás
  - CGI: cgi-bin mappában
  - Szerver modul: tetszőleges helyen

```
http://szerver/peldak/pelda.php
```

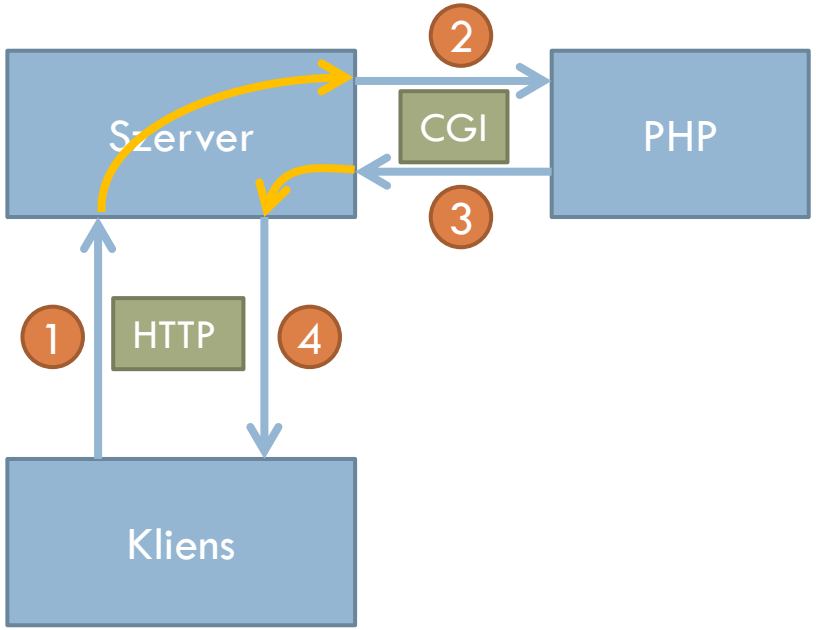
- CLI – Command Line Interfész
  - PHP szkriptek parancssori futtatása

```
> php pelda.php
```

- PHP-GTK
  - Grafikus asztali programok írása

PHP a webprogramozásban

- .php kiterjesztésű állomány kérése esetén webszerver kikeresi az állományt
- Átadja a PHP értelmezőnek
- A program kimenetét a böngésző a kliens felé továbbítja




```
http://szerver/peldak/pelda.php
```

PHP mint programozási nyelv

- Gyengén típusos
  - változók típusa a benne tárolt értéktől függ
  - automatikus típuskonverziók
- Értelmezett
  - PHP értelmező
- Szkriptnyelv
- Sok minden igaz, amit JavaScriptnél tanultunk
  - szintaxis
  - vezérlési szerkezetek, operátorok
  - viselkedés

További jellemzők

- Kis- és nagybetű érzékeny
- Utasításokat pontosvessző zárja le
- Objektumorientált nyelv (OOP)
- C alapú szintaxis (ismerős lesz)
- Nincs főprogram

Megjegyzés


```
<?php
//egysoros megjegyzés

# Perl szintaktikájú egysoros
megjegyzés

/*
többsoros
megjegyzés
*/
?>
```

Típusok

- Négy elemi típus
  - logikai
  - egész
  - lebegőpontos
  - szöveg
- Két összetett típus
  - tömb
  - osztály
- Speciális típusok
  - erőforrás
  - NULL
  - callbacks

Literálok


```
//Logikai
true
false
TRUE
FALSE

//Egész
12      //decimális
-34
0123    //oktális
0x0F    //hexadecimális
0b0101 //bináris

//Lebegőpontos
3.1415
5.6e12
-7E-2
```

```
//Például
$l = true;
$i = -23;
$d = 23.65;
```

Szövegliterál

- aposztróf
- macskaköröm
- heredoc
- nowdoc
- Változók behelyettesítése

```php
$a = 12;

$s1 = 'alma\t{$a} alma';  //alma\t{$a} alma
$s1 = 'Több
sor is lehet benne';

$s2 = "alma\t{$a} alma";  //alma   12 alma
$s2 = "Ez egy
több soros szöveg";

$s3 = <<<EOT              //behelyettesít
Több soros {$a}
szöveg
EOT;

$s4 = <<<'EOT'            //nem helyettesít be
Ez is lehet {$a}
több soros.
EOT;
```

Változók

- $valtozo
- Hatókör
  - globális változók
  - függvényen belül lokális változók
- Szuperglobális változók (ld. függvények)
- Változó változók
  - $a = 'hello';
  - $$a = 12; //$hello === 12

Kiírás

- Ami nincs PHP blokkban, automatikusan kiíródik
- Kiírás
  - echo
  - var_dump()
  - print_r()

Típusokkal kapcsolatos fontos függvények

- Típusbeállítás
  - cast
    - (int)$a
  - settype()
- Típuslekérdezés
  - gettype()
  - is_integer()
  - is_float()
  - is_numeric()
  - is_string()
  - is_bool()
  - …

Kiírások, típusműveletek


```php
$l = true;
$i = -23;
$d = 23.65;
$s = 'alma';

echo $l;  //1
echo $i;  //-23
echo $d;  //23.65
echo $s;  //alma

var_dump($l); //bool(true)
var_dump($i); //int(-23)
var_dump($d); //float(23.65)
var_dump($s); //string(4) "alma"

print_r($l);  //1
print_r($i);  //-23
print_r($d);  //23.65
print_r($s);  //alma
```

```php
//Típuslekérdezés
echo gettype($l); //'boolean'
echo gettype($i); //'integer'
echo gettype($d); //'double'
echo gettype($s); //'string'

//Típusbeállítás
$sd1 = (string)$d;
$sd2 = $d;
settype($sd2, 'string');
echo gettype($sd1); //'string'
echo gettype($sd2); //'string'
```

Típusokkal kapcsolatos fontos függvények

- Speciális függvények
  - isset()
  - is_null()
  - empty()
- Automatikus konverziók
  - http://www.php.net/manual/en/language.types.type-juggling.php
- Típusok összehasonlítása
  - http://www.php.net/manual/en/types.comparisons.php

Összehasonlítás

Expression	gettype()	empty()	is_null()	isset()	boolean : *if($x)*
$x = "";	string	**TRUE**	**FALSE**	**TRUE**	**FALSE**
$x = null	NULL	**TRUE**	**TRUE**	**FALSE**	**FALSE**
var $x;	NULL	**TRUE**	**TRUE**	**FALSE**	**FALSE**
$x is undefined	NULL	**TRUE**	**TRUE**	**FALSE**	**FALSE**
$x = array();	array	**TRUE**	**FALSE**	**TRUE**	**FALSE**
$x = false;	boolean	**TRUE**	**FALSE**	**TRUE**	**FALSE**
$x = true;	boolean	**FALSE**	**FALSE**	**TRUE**	**TRUE**
$x = 1;	integer	**FALSE**	**FALSE**	**TRUE**	**TRUE**
$x = 42;	integer	**FALSE**	**FALSE**	**TRUE**	**TRUE**
$x = 0;	integer	**TRUE**	**FALSE**	**TRUE**	**FALSE**
$x = -1;	integer	**FALSE**	**FALSE**	**TRUE**	**TRUE**
$x = "1";	string	**FALSE**	**FALSE**	**TRUE**	**TRUE**
$x = "0";	string	**TRUE**	**FALSE**	**TRUE**	**FALSE**
$x = "-1";	string	**FALSE**	**FALSE**	**TRUE**	**TRUE**
$x = "php";	string	**FALSE**	**FALSE**	**TRUE**	**TRUE**
$x = "true";	string	**FALSE**	**FALSE**	**TRUE**	**TRUE**
$x = "false";	string	**FALSE**	**FALSE**	**TRUE**	**TRUE**

Összehasonlítás

	TRUE	FALSE	1	0	-1	"1"	"0"	"-1"	NULL	array()	"php"	""
TRUE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE
FALSE	FALSE	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	TRUE	TRUE	FALSE	TRUE
1	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
0	FALSE	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE
-1	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE
"1"	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
"0"	FALSE	TRUE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE
"-1"	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE
NULL	FALSE	TRUE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	FALSE	TRUE
array()	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE
"php"	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE
""	FALSE	TRUE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	TRUE

Operátorok

- Ugyanaz, mint JavaScriptben
- Különbség
  - + kizárólag összeadás
  - szövegösszefűzés: .

```
"piros"."alma"
```

Vezérlési szerkezetek

- Ugyanaz, mint C++-ban vagy JavaScriptben
- Elágazások
  - if
  - if-else
  - switch
- Ciklusok
  - for
  - foreach (ld. tömbök)
  - while
  - do-while

```php
<?php
//Különbségek a JavaScripthez képest
for ($i = 1; $i <= n; $i++) {
  utasítások;
}

foreach ($tomb as $ertek) {
  utasítások;
}

foreach ($tomb as $kulcs => $ertek) {
  utasítások;
}
?>
```

Függvények

- Szintaxisban hasonlít a JavaScriptes függvényekre
- Változók láthatósága más
  - A globális változók nem látszódnak a függvényeken belül
  - → $GLOBALS[] tömb
  - → global kulcsszó
  - szuperglobális változók (mindenhol látszódnak)
- Érték és referencia szerinti paraméterátadás
  - mint C++-ban

Függvények


```php
//Függvény általános formája
function fvnev($par1, $par2) {
  utasítások;
  return visszatérési érték;
}

//Például
function negyzet($x) {
  return $x * $x;
}
negyzet(3);     // => 9
```

```php
//Alapértelmezett érték
function udvozles($nev = 'Senki bácsi') {
  return "Hello {$nev}!";
}
udvozles();     // => "Hello Senki bácsi!"

//Referencia szerinti paraméterátadás
$szam = 41;
function novel(&$szam) {
  $szam += 1;
}
novel($szam);
echo $szam;     // => 42
```

Függvények


```php
//Globális változók elérése
$globus = 'Föld';
function zartFuggveny() {
  global $globus;
  echo "Gyönyörű a {$globus}!";
}
//vagy
function zartFuggveny() {
  echo "Gyönyörű a {$GLOBALS['globus']}!";
}
```

Összetett adatszerkezetek

Tömb

- Gyűjtemények általános objektuma
  - Összetett adatszerkezet megvalósítása
    - rekord
    - indexelt tömb
    - asszociatív tömb
    - többdimenziós tömb
    - fa, sor, verem, stb.
- Asszociatív tömb: kulcs-érték párokból áll
  - kulcs: integer vagy string
  - érték: bármilyen típusú lehet

Tömb


```php
//Üres tömb
$uresTomb = array();

//Indexelt tömb
$indTomb = array('alma', 'korte',
'szilva');
echo $indTomb[0]; //'alma'
print_r($indTomb);
/*
Array
(
    [0] => alma
    [1] => korte
    [2] => szilva
)
*/

//A tömb hosszának lekérdezése
count($indTomb);      // => 3
```

```php
//Elemek módosítás
$indTomb[1] = 13;
$indTomb[1];          //13

//Új elem beszúrása a tömb végére
$indTomb[] = 'új';
print_r($indTomb);
/*
Array
(
    [0] => alma
    [1] => 13
    [2] => szilva
    [3] => új
)
*/

//Elem törlése
unset($t[1]);
```

Asszociatív tömb


```php
//Asszociatív tömb
$asszTomb = array(
  'alma'    => 'piros',
  'korte'   => 'sarga',
  'szilva'  => 'kek',
);
echo $asszTomb['alma']; //piros
print_r($asszTomb);
/*
Array
(
    [alma] => piros
    [korte] => sarga
    [szilva] => kek
)
*/
```

```php
//Asszociatív tömb kézzel
$asszTombKezi = array();
$asszTombKezi['alma'] = 'piros';
$asszTombKezi['korte'] = 'sarga';
$asszTombKezi['szilva'] = 'kek';
print_r($asszTombKezi);
/*
Array
(
    [alma] => piros
    [korte] => sarga
    [szilva] => kek
)
*/
```

Mátrix

- Tömbök tömbje
- Lehet vegyesen használni
  - indexeltben indexelt
  - indexeltben asszociatív

```php
//Mátrix
$matrix = array(
  array(1, 2, 3),
  array(4, 5, 6),
  array(7, 8, 9),
);
```

Tömbök bejárása

- foreach
- reset(), next(), prev(), current(), key(), each()

```php
$gyumolcsok = array(
        'alma'          => 'piros',
        'korte'         => 'sárga',
        'szilva'        => 'kék',
);

foreach ($gyumolcsok as $gyumolcs => $szin) {
        echo "{$szin} {$gyumolcs}\n";
}
/*
piros alma
sárga korte
kék szilva
*/
```

```php
reset($gyumolcsok);
while (list($kulcs, $ertek) = each($gyumolcsok)) {
    echo "{$ertek} {$kulcs}\n";
}
```

Tömb, rekord

Tömb

```php
$kutyuk = array(
  'telefon',
  'fülhallgató',
  'pendrive',
  'e-könyv olvasó',
);
```

Rekord

```php
$hallgato = array(
  'nev'    => 'Mosolygó Napsugár',
  'neptun'  => 'kod123',
  'szak'    => 'Informatika BSc'
);
```

Rekordok tömbje


```
$hallgatok = array(
  array(
    'nev'    => 'Mosolygó Napsugár',
    'neptun'  => 'kod123',
    'szak'    => 'Informatika BSc',
  ),
  array(
    'nev'    => 'Kék Ibolya',
    'neptun'  => 'kod456',
    'szak'    => 'Informatika BSc',
  ),
);
```

```php
$hallgatok = array(
  array(
    'nev'     => 'Mosolygó Napsugár',
    'neptun'  => 'kod123',
    'szak'    => 'Informatika BSc',
    'targyak'  => array(
      'Programozás',
      'Webfejlesztés 2.',
      'Számítógépes alapismeretek',
    ),
  ),
  array(
    'nev'     => 'Kék Ibolya',
    'neptun'  => 'kod456',
    'szak'    => 'Informatika BSc',
    'targyak'  => array(
      'Programozás',
      'Webfejlesztés 2.',
      'Diszkrét matematika',
      'Testnevelés',
    ),
  ),
);
```

Összegzés tétel


```php
function osszegzes($tomb) {
  $s = 0;
  foreach ($tomb as $szam) {
    $s = $s + $szam;
  }
  return $s;
}

$x = array(1, 3, -2, 8);
echo 'Az összeg: ' . osszegzes($x);
```

Kimenet generálása

PHP kód elhelyezése

- .php kiterjesztésű fájl
- <?php és ?> tagek közötti részt futtatja a PHP értelmező
- Ezeken kívüli rész automatikusan kiírásra kerül
- Több PHP blokk is lehet

```
<?php
//PHP kód
?>
```

```
Szöveg
<?php
//PHP kód
?>
Szöveg
<?php
//PHP kód
?>
Szöveg
```

Példa


```
echo '<!doctype html>';
echo '<html>';
echo '  <head>';
echo '          <meta charset="utf-8">';
echo '          <title></title>';
echo '   </head>';
echo '  <body>';
echo '      <p>Hello vilag!</p>';
echo '  </body>';
echo '</html>';
```

Példa


```
echo <<<VEGE
<!doctype html>
<html>
        <head>
                <meta charset="utf-8">
                <title></title>
        </head>
        <body>

        </body>
</html>
VEGE;
```

Példa


```
<!doctype html>
<html>
        <head>
                <meta charset="utf-8">
                <title></title>
        </head>
        <body>

        </body>
</html>
```