## Alapfogalmak

A PHP (Hypertext Preprocessor) egy általános szerveroldali - objektumorientált - szkriptnyelv dinamikus weblapok készítésére. Az első szkriptnyelvek egyike, amely külső fájl használata helyett HTML oldalba ágyazható (1995). Képes **dinamikus weboldal**ak létrehozására, lehetőséget nyújt az **adatbázissal való kommunikáció**ra, illetve web-alapú **szoftver-alkalmazások fejlesztés**ére.[[1]](#footnote-0) A felsoroltakon kívül a szerver oldali fájlok kezelését, sütik küldését, fogadását, jogosultsági hierarchia kialakítását, titkosítást (pl. jelszavak) is lehetővé teszi.

A PHP függvények kimenete jellemzően HTML kód, ami a böngésző számára értelmezhető. *Mivel a kód átalakítása még betöltés előtt megtörténik, ezért a felhasználók számára láthatatlan marad az oldalon lévő PHP kód*. Alapvetően ez teszi elég biztonságossá az adatbázis vagy egyéb titkos információk elérését.

A nyelv szintaxiskészletében megtalálhatók a C, a Java és a Perl nyelvi elemek is, és hasonlóan a Pythonhoz ez is egy *gyengén típusos nyelv*. Ettől függetlenül akad bőven különleges, PHP specifikus elem is.[[2]](#footnote-1)

A PHP a weboldal back-end részéért felel. A **back-end** (ellentétben a front-enddel) egy adott rendszer alsóbb, a **tényleges feldolgozást végző réteg**e. A back-end réteg **feladata** a front-end réteg felől érkező **adatok feldolgozása**, ill. a keletkezett eredmény a front-end számára történő **visszajuttatása**.[[3]](#footnote-2)  
A feldolgozáshoz *szükség van egy szerverre*, pl. ilyen az *Apache HTTP Server* (röviden Apache) *nyílt forráskódú webkiszolgáló alkalmazás*.

## Miért PHP?

A legtöbb ma használt szoftverrel és operációs rendszerrel kompatibilis (ezáltal költséghatékony), elterjedt a használata (Facebook, Twitter, Wikipédia, főbb CMS , mint a WordPress is, ezen alapul...). Nyílt forráskódú, ingyenes, egyszerűen tanulható, jól dokumentált, rugalmas.[[4]](#footnote-3)

## Miért ne?

Lassú, kevésbé praktikus debuggolás, nehezen előrejelezhető memóriahasználat.

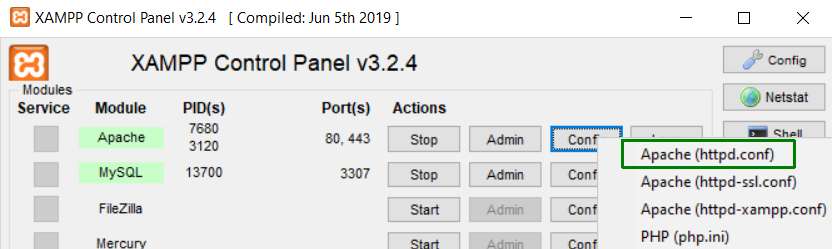
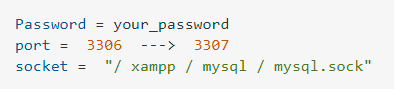
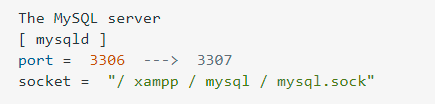
## Hasznos oldalak

<https://www.php.net/>

<https://www.tutorialspoint.com/php/php_tutorial.pdf>

<https://www.w3schools.com/php/default.asp>

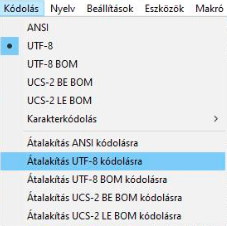
## Fejlesztői környezet telepítése (XAMPP)

1. <https://www.apachefriends.org/hu/index.html> (megfelelő op. rendszer kiválasztása után letölti...)
2. C:\xampp\xampp-control.exe (Windows) elindítása (érdemes asztalra/tálcára tenni).
3. Apache (webkiszolgáló alkalmazás) és MySQL ( SQL-alapú relációs adatbázis-kezelő szerver) elindítása
4.   
   Ha szükséges (nem működik), beállítjátok a port-ot a httpd.conf fájlban (Listen kulcsszóra való keresés után 80 helyett 8080 beállítása)
5. Első php fájl létrehozása a *C:\xampp\htdocs mappában* (alapbeállításként ez a mappa szerepel).
6.   
   Ha véletlen a 3306 port is foglalt, akkor
   1. állítsd le a szervert, majd
   2. a C:\xampp/mysql/bin/my.ini fájlban állítsd át mindkét helyen a portot 3307-re:  
        
      
   3. indítsd újra a szervert![[5]](#footnote-4)
7. Blokkolt port esetén próbáljuk ki az [alábbi](https://www.webucator.com/blog/2016/05/fixing-a-blocked-port-in-xampp/) lépéseket.

## Elnevezések, írásmód

* alapvetően nem case-sensitive, **de a változók igen**!
* camelCase (esetleg snake\_case) írásmód használatos: változók, függvények nevei,
* CSUPA\_NAGY\_BETU, esetleg CSUPANAGYBETU írásmód: konstansok (automatikusan globális)
* PascalCase írásmód: osztály neve

## Szöveg kódolása



## Alapszintaxis

<?php

echo '<p>Ezt a PHP írta.</p>';

?>

## Komment

//comment

#comment

/\*több

soros

komment\*/

## Kiíratás

echo 'sz';

echo('sz');

print 'sz';

print('sz');

print\_r($tomb1);

var\_dump($GLOBALS);

## 

## Konstans

Általánosan:

define ( string $name , [mixed](https://www.php.net/manual/en/language.types.declarations.php#language.types.declarations.mixed) $value , bool $case\_insensitive = false ) : bool[[6]](#footnote-5) - a visszatérési érték jelentése: sikeres true esetén

Konkrétan:

define("GREETING","Hello you! How are you today?");

echo constant("GREETING");[[7]](#footnote-6)

## Változók

A változót inicializáláskor is “$” jellel jelöljük, és hivatkozásakor is. A konkatenáció (összefűzés) jele a pont.

### Szöveg

Érdemes kívül használni a “-t, belül az ‘ jelet.

$szoveg1 = 'szoveg1';

$szoveg2 = "szoveg2";

echo "A két szöveg együtt: ".$szoveg1."+".$szoveg2;

$szoveg8 = "Szép";

$szoveg8 .=" napot!"; - jelentése += Javaban

echo $szoveg8;

### Egész

32 bites előjeles számok 32 bites rendszerekben, 64 bitesek 64 bites rendszerekben, ha a határokat eléri bárhol, akkor automatikusan float lesz a szám tárolása.

$egesz = 2;

### Tört

### ***Keressétek meg a double határát!*** *Segítség:*

echo pow(2.0, ?);

$tort1 = 3.56;

$tort2 = 3.44;

echo "<br>A két szám összege: ".$tort1+$tort2; - az eredmény egész szám kinézetre, de marad a double típus.

Ugyanúgy a 4 \* 2.5 eredménye 10, de az eredmény double-ként van tárolva a 2.5 szorzó tényező miatt.

### Logikai típus

$logikai1 = true;

$logikai2 = false;

echo "<br>A két logikai érték: ".$logikai1." ".$logikai2; - csak az igaz jelenik meg!

[Próbáld ki ezeket is!](https://www.php.net/manual/en/function.boolval.php)

## Lista létrehozása (és kiíratása)[[8]](#footnote-7)

$a = array("red", "green", "blue");

print\_r($a);

$tomb0 = array(1,2,3,4,5);

$tomb1 = array();

$tomb2[0] = "Ady Endre";

$tomb2[1] = "József Attila";

$tomb3 = ["Milan Kundera", "Hermann Hesse"];

Hozzáfűzés kétféleképpen:

$tomb2[] = "Tóth Árpád";

array\_push($tomb2, "Tóth Árpád");

Lista elemszáma:

$elemSzam = count($tomb2);

Listák összeillesztése:

$tomb4 = array\_merge($tomb2, $tomb3);

### Asszociatív tömb (dictionary)

$b = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");

print\_r($b);

$tomb1["elso"]="Ady";

$tomb1["masodik"]="József";

Módosítás és létrehozás ugyanúgy:

$tomb1["masodik"]="Karinthy";

### Tömb mint konstans

define('ANIMALS', array(

'dog',

'cat',

'bird'

));

echo ANIMALS[1];

## Operátorok

//+-\*/%, && , ||

echo (8/3)."<br>";

//matematikai programcsalád

echo floor(8/3)."<br>";

Konkatenáció jele: . (pont)

Típus lekérdezése

echo gettype($hamis);

## Változók életciklusa[[9]](#footnote-8)

Ami eltér a megszokottól:

Ha a függvény lokális változóját nem szeretnénk törölni életciklusa után, akkor a változó előtt ezt **static** kulcsszóval jelöljük.

## Vezérlési szerkezetek

### Elágazás\_1:

if ($hamis) {

...

}

elseif ($feltetel){

...

}

else {

...

}

### Elágazás\_2:

$logikai = true;

echo ($logikai ? 'Ez igaz.' : 'Ez hamis.');

### Elágazás\_3:

switch ($valtozo) {

case $value1:

...

break;

case $value2:

...

break;

default:

...

break;

}

### Ciklus\_1:

for ($index1 = 0; $index1 < count($tomb); $index1++) {

...

}

### Ciklus\_2:

foreach ($tomb2 as $value) {

echo $value." ";

}

### Ciklus\_3 (asszociatív tömb bejárása):

foreach ($tomb0 as $kulcs => $ertek) {

echo $kulcs. ": ".$ertek." <br>";

}

### Ciklus\_4:

$index1 = 0;

while ($index1<10){

echo $index1." ";

$index1++;

}

### Ciklus\_5:

$x = 1;

do {

echo "A szám: $x <br>";

$x++;

} while ($x <= 5);

## Űrlapoknál adatküldés (action.php-ban:)

if(isset($\_POST['szamol'])){

$v = (int) $\_POST['v'];

$a = (int) $\_POST['A'];

$w = $a\*$v;

echo "<p>Az adott keresztmetszeten időegység alatt átáramló folyadék térfogata: ".$w."</p>";

}

A fenti kódban a szamol a submit gomb **name** értéke.

Az isset függvény azt ellenőrzi, hogy be van-e állítva a paraméterül átadott változó értéke. (Beírták-e az adatot, majd megnyomták-e a gombot.)

Ha a küldés módja GET, akkor csak ilyen módon tudod feldolgozni (isset($\_GET['szamol'])...), és fordítva.

1. <https://www.tutorialspoint.com/php/php_tutorial.pdf> [↑](#footnote-ref-0)
2. <https://techterms.com/definition/php> [↑](#footnote-ref-1)
3. <https://pcforum.hu/szotar/?term=backend> [↑](#footnote-ref-2)
4. <https://www.mantralabsglobal.com/blog/7-reasons-php-getting-popular/> [↑](#footnote-ref-3)
5. <https://stackoverflow.com/questions/43421403/mysql-wont-start-post-3306-in-use> [↑](#footnote-ref-4)
6. <https://www.php.net/manual/en/function.define.php> [↑](#footnote-ref-5)
7. <https://www.w3schools.com/php/func_misc_constant.asp> [↑](#footnote-ref-6)
8. <https://www.w3schools.com/php/phptryit.asp?filename=tryphp_func_var_print_r> [↑](#footnote-ref-7)
9. <https://www.w3schools.com/php/php_variables_scope.asp> [↑](#footnote-ref-8)