

Web Garapena Zerbitzari-Ingurunean

1.SARRERA

Dagoeneko Jakin beharko zenukeena

Jarraitu aurretik, honako hauen oinarritzko ulermena izan beharko zenuke:

- [HTML](#)
- [CSS](#)
- [JavaScript](#)

Zer da PHP?

- PHP "PHP: Hypertext Preprocessor" akronimoa da
- PHP oso erabilia eta kode irekiko script-lengoaia da
- PHP scriptak zerbitzarian exekutatzen dira
- PHP doakoa da deskargatzeko eta erabiltzeko

PHP hizkuntza harrigarria eta ezaguna da!

Nahikoa indartsua da sareko blog-sistema handienaren oinarrian egoteko (WordPress)!

Sakona nahikoa da sare sozial handiak exekutatzeko!

Nahikoa erraza da hasiberriaren lehen zerbitzariaren alboko hizkuntza izatea ere!

Zer da PHP fitxategi bat?

- PHP fitxategiek testua, HTML, CSS, JavaScript eta PHP kodea izan ditzakete
- PHP kodea zerbitzarian exekutatzeko da, eta emaitza nabigatzaileari itzultzen da HTML arrunt gisa
- PHP fitxategiek luzapena dute ".php"



Zer egin dezake PHP-k?

- PHP-k orrialdeko eduki dinamikoa sor dezake
- PHP zerbitzarian fitxategiak sortu, ireki, irakurri, idatzi, ezabatu eta itxi ditzake
- PHP-k inprimakiaren datuak bil ditzake
- PHP cookieak bidali eta jaso ditzake
- PHP-k zure datu-baseko datuak gehitu, ezabatu eta alda ditzake
- PHP erabil daiteke erabiltzaileen sarbidea kontrolatzeko
- PHP-k datuak enkriptatu ditzake

PHPrekin ez zara HTML irteerara mugatzen. Irudiak edo PDF fitxategiak atera ditzakezu. Edozein testu ere atera dezakezu, hala nola XHTML eta XML.

Zergatik PHP?

- PHP hainbat plataformatan exekutatzen da (Windows, Linux, Unix, Mac OS X, etab.)
- PHP bateragarria da gaur egun erabiltzen diren ia zerbitzari guztiekin (Apache, IIS, etab.)
- PHP-k datu-base ugari onartzen ditu
- PHP doakoa da. Deskargatu PHP baliabide ofizialetik: www.php.net
- PHP ikasteko erraza da eta zerbitzariaren aldetik modu eraginkorrean exekutatzen da

2. INSTALAZIOA

XAMPP

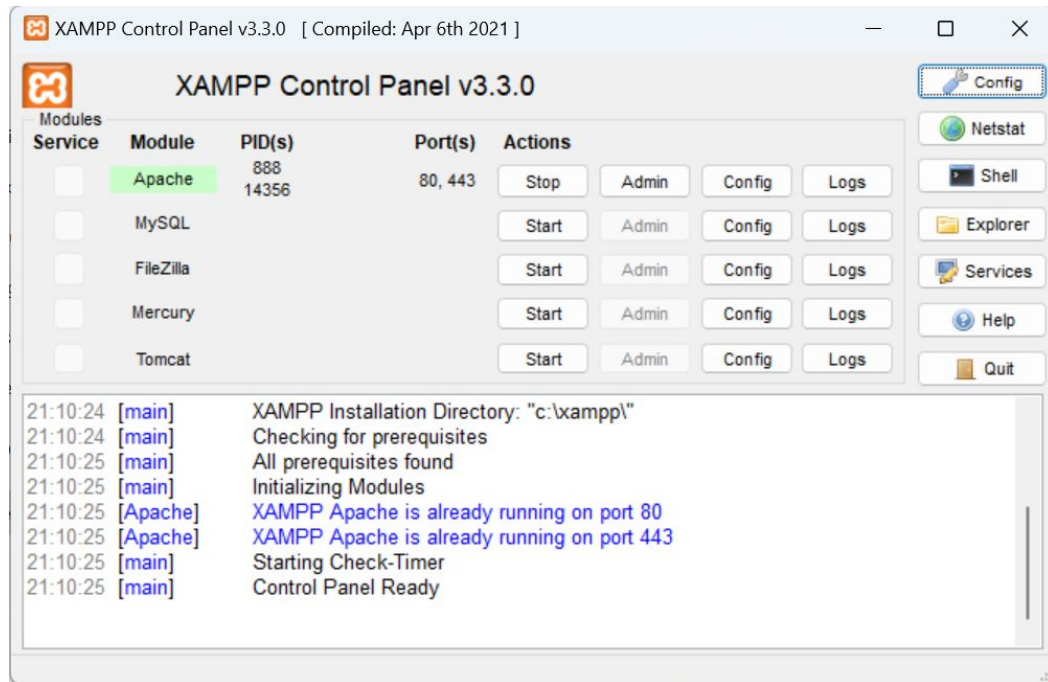
XAMPP garapen-ingurune ezagunena da PHPrekin

XAMPP Apache banaketa guztiz doakoa eta instalatzeko erraza da eta MariaDB, PHP eta Perl ditu. Instalatzeko eta erabiltzeko oso erraza izateko diseinatu da XAMPPren instalazio-paketea.

[Deskargatu](#)

XAMPP

C:\xampp\xampp-control → aginte panela kargatzeko exekutagarria



Apache Config → httpd.conf
apacheren konfigurazio fitxategia

DocumentRoot "C:/xampp/htdocs"
adierzten du zein den fitxategien erroa.
Alda daiteke baina ez da gomendagarria

VisualStudio Code

Garapen ingurunea php-rekin lan egiteko garapen ingurunea.



[Deskargatu](#)

3. PHP Oinarrizko sintaxia

PHP script bat dokumentuko edozein lekutan jar daiteke.

PHP script bat honela hasten da `<?php` eta honekin amaitzen da `?>`:

```
<?php
// PHP code goes here
?>
```

PHP fitxategien luzapen lehenetsia " `.php`" da.

PHP fitxategi batek normalean HTML etiketak eta PHP scripting kode batzuk ditu.

Jarraian, PHP fitxategi soil baten adibidea dugu, PHP script bat duena, " `echo`"Hello World!" web orri batean:

ADIBIDEA

```
<!DOCTYPE html>
<html>
<body>

<h1>My first PHP page</h1>

<?php
echo "Hello World!";
?>

</body>
</html>
```

Letra handiak eta letra txikiak

PHPn, gako-hitzak (adibidez if, else, while, echo, etab.), klaseak, funtzioak eta erabiltzaileak definitutako funtzioak ez dira maiuskulak eta minuskulak bereizten.

Oharra: Hala ere; aldagai-izen guztiek maiuskulak eta minuskulak bereizten dituzte!

```
<!DOCTYPE html>
<html>
<body>

<?php
$color = "red";
echo "My car is " . $color . "<br>";
Echo "My house is " . $COLOR . "<br>";
ECHO "My boat is " . $coLOR . "<br>";
?>

</body>
</html>
```

Iruzkinak

Lerro bakarreko iruzkina egiteko

- `//`
- `#`

Lerro askotako iruzkinak egiteko

- `/* ... */`

Aldagaiak

- PHPn, aldagai bat zeinuarekin hasten da \$, eta ondoren aldagaiaren izenaz:
- Aldagaien arauak:
 - Aldagai bat zeinuarekin hasten da \$, aldagaiaren izenaren ondoren
 - Aldagaiaren izenak letra batekin edo azpimarra karaktere batekin hasi behar du
 - Aldagaiaren izena ezin da zenbaki batekin hasi
 - Aldagai-izen batek karaktere alfa-zenbakizkoak eta azpimarra soilik izan ditzake (Az, 0-9 eta _)
 - Aldagaien izenek maiuskulak bereizten dituzte (\$ageeta \$AGEbi aldagai ezberdin dira)

Adibidez:

```
$txt = "W3Schools.com";
```

```
$x = 15.5;
```

ECHO ETA PRINT

echo eta print gutxi gorabehera berdinak dira. Biak erabiltzen dira datuak pantailara ateratzeko.

Aldeak txikiak dira:

- echo ez du itzulera-baliorik eta print 1-eko balioa du, beraz, esamoldeetan erabil daiteke.
- echo parametro anitz har ditzake (halako erabilera arraroa den arren), print argumentu bat har dezake.
- echo baino apur bat azkarragoa da print.

Operadore aritmetikoak

Operator	Name	Example	Result
+	Addition	$\$x + \y	Sum of $\$x$ and $\$y$
-	Subtraction	$\$x - \y	Difference of $\$x$ and $\$y$
*	Multiplication	$\$x * \y	Product of $\$x$ and $\$y$
/	Division	$\$x / \y	Quotient of $\$x$ and $\$y$
%	Modulus	$\$x \% \y	Remainder of $\$x$ divided by $\$y$
**	Exponentiation	$\$x ** \y	Result of raising $\$x$ to the $\$y$ 'th power

Beste operadore batzuk

Assignment	Same as...	Description
<code>x = y</code>	<code>x = y</code>	The left operand gets set to the value of the expression on the right
<code>x += y</code>	<code>x = x + y</code>	Addition
<code>x -= y</code>	<code>x = x - y</code>	Subtraction
<code>x *= y</code>	<code>x = x * y</code>	Multiplication
<code>x /= y</code>	<code>x = x / y</code>	Division

Operator	Name	Description	
<code>++\$x</code>	Pre-increment	Increments \$x by one, then returns \$x	
<code>\$x++</code>	Post-increment	Returns \$x, then increments \$x by one	
<code>--\$x</code>	Pre-decrement	Decrements \$x by one, then returns \$x	
Operator	Name	Example	Result
<code>and</code>	And	<code>\$x and \$y</code>	True if both \$x and \$y are true
<code>or</code>	Or	<code>\$x or \$y</code>	True if either \$x or \$y is true
<code>xor</code>	Xor	<code>\$x xor \$y</code>	True if either \$x or \$y is true, but not both

IF..ELSE..ELSEIF

PHPn baldintzapeko adierazpen hauek ditugu:

- if adierazpena - koderen bat exekutatzen du baldintza bat egia bada
- if...else adierazpena - kode bat exekutatzen du baldintza bat egia bada eta beste kode bat baldintza hori faltsua bada
- if...elseif...else adierazpena - kode desberdinak exekutatzen ditu bi baldintza baino gehiagotarako

SINTAXIA →

```
if (condition) {  
    code to be executed if this condition is true;  
} elseif (condition) {  
    code to be executed if first condition is false and this condition is true;  
} else {  
    code to be executed if all conditions are false;  
}
```

Switch

Adierazpena switch baldintza ezberdinetan oinarritutako ekintza desberdinak egiteko erabiltzen da.

```
switch (n) {  
    case label1:  
        code to be executed if  
n=label1;  
        break;  
    case label2:  
        code to be executed if  
n=label2;  
        break;  
    case label3:  
        code to be executed if  
n=label3;  
        break;  
    ...  
    default:  
        code to be executed if n is  
different from all labels;  
}
```

ADIBIDEA

```
<?php  
$favcolor = "red";  
  
switch ($favcolor) {  
    case "red":  
        echo "Your favorite color is red!";  
        break;  
    case "blue":  
        echo "Your favorite color is blue!";  
        break;  
    case "green":  
        echo "Your favorite color is green!";  
        break;  
    default:  
        echo "Your favorite color is neither red, blue, nor  
green!";  
}  
?>
```

Ariketak 1 eta Ariketak 2

Errepikapenak While

`while` errepikapenak kode bloke bat exekutatzen du, zehaztutako baldintza egiazkoa den bitartean.

```
while (condition is true) {  
    code to be executed;  
}
```

ADIBIDEA

Beheko adibidean 1etik 5era bitarteko zenbakiak erakusten dira:

```
<?php  
$x = 1;  
  
while($x <= 5) {  
    echo "The number is: $x <br>";  
    $x++;  
}  
?>
```

Errepikapenak do..while

`do...while` errepikapenak beti exekutatzen du kode blokea behin, gero baldintza egiaztatuko du eta errepikapena errepikatuko du zehaztutako baldintza egia den bitartean.

```
do {  
    code to be executed;  
} while (condition is true);
```

ADIBIDEA

Beheko adibideak lehenik `$x` aldagai bat ezartzen du 1 (`$x = 1`). Orduan, `do while` errepikapenak irteeraren bat idatziko du, eta, ondoren, `$x` aldagaia 1arekin handituko da. Ondoren, baldintza egiaztatzen da (`$x 5` baino txikiagoa edo berdina al da?), eta errepikapenak exekutatzen jarraituko du. `$x 5` baino txikiagoa edo berdina da:

```
<?php  
$x = 1;  
  
do {  
    echo "The number is: $x <br>";  
    $x++;  
} while ($x <= 5);  
?>
```

Errepikapenak For

for errepikapena erabiltzen da aldeztu aurretik scriptak zenbat aldiz exekutatu behar duen dakizunean.

```
for (init counter; test counter; increment counter) {  
    code to be executed for each iteration;  
}
```

ADIBIDEA

Behoko adibidean 0tik 10era bitarteko zenbakiak bistaratzen dira:

```
<?php  
for ($x = 0; $x <= 10; $x++) {  
    echo "The number is: $x <br>";  
}  
?>
```

Errepikapenak foreach

`foreach` errepikapena matrizeetan bakarrik funtzionatzen du, eta array bateko gako/balio bikote bakoitzean errepikapena egiteko erabiltzen da.

```
foreach ($array as $value) {  
    code to be executed;  
}
```

ADIBIDEA

Hurrengo adibidean emandako matrizearen balioak aterako dira (\$koloreak):

```
<?php  
$colors = array("red", "green", "blue", "yellow");  
  
foreach ($colors as $value) {  
    echo "$value <br>";  
}  
?>
```

Ariketak 3

Array-ak

Array bat aldagai berezi bat da, aldi berean balio bat baino gehiago eduki ditzakeena.

Array batek balio asko eduki ditzake izen bakar baten azpian, eta balioak atzi ditzakezu indize-zenbaki bati erreferentzia eginez.

Sortu Array bat

PHPn, `array()` funtzioa array bat sortzeko erabiltzen da:

```
array();
```

PHPn, hiru matrize mota daude:

- Indexed arrays - Zenbakizko indizea duten matrizeak
- Array asoziatiboak - gako izendun matrizeak
- Dimentsio anitzeko matrizeak - Array bat edo gehiago dituzten matrizeak

Array baten luzeera

Funtzioa count() array baten luzera (elementu kopurua) itzultzeko erabiltzen da:

ADIBIDEA

```
<?php  
$cars = array("Volvo", "BMW", "Toyota");  
echo count($cars);  
?>
```

Array-ak: Indexatutako matrizeak

Bi modu daude indexatutako matrizeak sortzeko:

Indizea automatikoki esleitu daiteke (indizea 0tik hasten da beti), honela:

```
$cars = array("Volvo", "BMW", "Toyota");
```

edo indizea eskuz esleitu daiteke:

```
$cars[0] = "Volvo";  
$cars[1] = "BMW";  
$cars[2] = "Toyota";
```

Array-ak: Errepikapenak

Indexatutako array baten balio guztiak arakatu eta inprimatzeko, for errepikapen bat erabil dezakezu, honela:

ADIBIDEA

```
<?php
$cars = array("Volvo", "BMW", "Toyota");
$arrlength = count($cars);
for($x = 0; $x < $arrlength; $x++) {
    echo $cars[$x];
    echo "<br>";
}
?>
```

Array asoziatiboak

Array asoziatiboak zuk esleitzen dituzun gako izendatuak erabiltzen dituzten matrizeak dira.

Array elkartu bat sortzeko bi modu daude:

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

edo:

```
$age['Peter'] = "35";
```

```
$age['Ben'] = "37";
```

```
$age['Joe'] = "43";
```

Array-ak: Errepikapen asoziatiboak

Array asoziatibo baten balio guztiak arakatu eta inprimatzeko, foreach errepikapen bat erabil dezakezu, honela:

ADIBIDEA

```
<?php
$page = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");

foreach($page as $x => $x_value) {
    echo "Key=" . $x . ", Value=" . $x_value;
    echo "<br>";
}
?>
```

Array-ak: funtzio erabilienak

- `count ()` → Funtzioa `count()` array baten luzera (elementu kopurua) itzultzeko erabiltzen da.
- Ordenatzeko:
 - `sort()`- ordenatu matrizeak goranzko ordenan
 - `rsort()`- ordenatu matrizeak beheranzko ordenan
 - `asort()`- ordenatu matrize elkartuak goranzko ordenan, balioaren arabera
 - `ksort()`- ordenatu matrize elkartuak goranzko ordenan, gakoaren arabera
 - `arsort()`- ordenatu matrize elkartuak beheranzko ordenan, balioaren arabera
 - `krsort()`- ordenatu matrize elkartuak beheranzko ordenan, gakoaren arabera
- Bilaketak:
 - `in_array(var1,array)` → bilatzen du `var1` array-an
- `unset (array)` → ezabatu array-a osorik edo elementu bat
- `var_dump(var1)` → aldagai bati buruzko informazioa erakusten du

Ariketa 4

Dimentsio anitzeko array-ak

PHP-k bi, hiru, lau, bost edo gehiagoko sakoneko dimentsio anitzeko matrizeak onartzen ditu. Hala ere, hiru maila baino gehiagoko arrayak kudeatzea zaila da jende gehientzat.

Bi dimentsioko array baten adibidea

```
$cars = array (  
    array("Volvo",22,18),  
    array("BMW",15,13),  
    array("Saab",5,2),  
    array("Land Rover",17,15)  
);
```

ADIBIDEA: Array-a arakatu

```
<?php  
for ($row = 0; $row < 4; $row++)  
{  
    echo "<p><b>Row number  
$row</b></p>";  
    echo "<ul>";  
    for ($col = 0; $col < 3;  
$col++) {  
        echo  
        "<li>".$cars[$row][$col]."</li>"  
        ;  
    }  
    echo "</ul>";  
}  
?>
```

Ariketa 5

Funtzioak

- PHP-k 1000 funtzio baino gehiago ditu barneratuta, eta horrez gain, zure funtzio pertsonalizatuak sor ditzakezu.
 - [hemen kontsultatu ditzakezu](#)
- PHP integratutako funtzioez gain, zure funtzioak sortzea posible da.
 - Funtzioa programa batean behin eta berriz erabil daitekeen adierazpen bloke bat da.
 - Funtzio bat ez da automatikoki exekutatu orrialde bat kargatzen denean.
 - Funtzio bat funtzioari dei baten bidez exekutatu da.

ADIBIDEA

```
<?php
function mezua() {
    echo "Kaixo Mundua!";
}

mezua(); // funtzioari deia
?>
```

Funtzio argudioak

Informazioa funtzioetara pasa daiteke argumentuen bidez. Argumentu bat aldagai bat bezalakoa da.

Argumentuak funtzioaren izenaren ondoren zehazten dira, parentesi artean. Nahi adina argumentu gehi ditzakezu, koma batekin bereizi.

ADIBIDEA

Ondorengo adibideak argumentu batekin (\$fname) funtzio bat du. FamilyName() funtzioari deitzen zaionean, izen bat ere pasatzen dugu (adib. Jani), eta izena funtzioaren barruan erabiltzen da, hainbat izen-abizenak ateratzen dituen, baina abizen berdina:

```
<?php
function familyName($fname) {
    echo "$fname Refsnes.<br>" ;
}

familyName("Jani");
familyName("Hege");
familyName("Stale");
familyName("Kai Jim");
familyName("Borge");
?>
```

Funtzioak bueltako balioa

Funtzioak balioak bueltatu ditzake “return” baten bitartez.

ADIBIDEA

Hurrengo adibidean zenbaki bat eta kate bat bidaltzen saiatzen gara:

```
<?php
function addNumbers( $a, $b) {
    return "$a, $b";
}
echo addNumbers(5, "5 days");
```

Funtzioak: strict

PHP 7-n, mota-deklarazioak gehitu ziren. Honek funtzio bat deklaratzearan espero den datu-mota zehazteko aukera ematen digu, eta deklarazioa gehituz gero strict , "Errore larria" botako du datu-mota bat ez datorrela.

ADIBIDEA

Hurrengo adibidean funtzioari zenbaki bat eta kate bat bidaltzen saiatzen gara, baina hemen deklarazioa gehitu dugu `strict` :

```
<?php declare(strict_types=1); // strict requirement

function addNumbers(int $a, int $b) {
    return $a + $b;
}

echo addNumbers(5, "5 days");
// since strict is enabled and "5 days" is not an integer, an error
will be thrown
?>
```

Ariketa 6

Aldagai Superglobalak

PHP-n aurrez definitutako aldagai batzuk "superglobalak" dira, hau da, beti eskuragarri daude, esparrua edozein dela ere, eta edozein funtzio, klase edo fitxategitatik atzi dezakezu ezer berezirik egin beharrik gabe.

PHP aldagai superglobalak hauek dira:

- `$GLOBALS`
- `$_SERVER`
- `$_REQUEST`
- `$_POST`
- `$_GET`
- `$_FILES`
- `$_ENV`
- `$_COOKIE`
- `$_SESSION`

\$GLOBALS

\$GLOBALS PHP script-eko edozein lekutatik aldagai globaletara sartzeko erabiltzen den PHP aldagai super globala da (funtzio edo metodoetatik ere).

PHP-k aldagai global guztiak \$GLOBALS[index] izeneko array batean gordetzen ditu . Indizeak aldagaiaren izena dauka .

Beheko adibidean \$GLOBALS aldagai super globala nola erabili erakusten da:

ADIBIDEA

```
<?php
$x = 75;
$y = 25;

function addition() {
    $GLOBALS['z'] = $GLOBALS['x'] +
    $GLOBALS['y'];
}

addition();
echo $z;
?>
```

\$_SERVER

\$_SERVER PHP aldagai super global bat da, goiburuei, bideei eta script-kokapenei buruzko informazioa biltzen duena.

[\\$_SERVER-en sartzen diren elementuak](#)

ADIBIDEA

```
<?php
echo $ SERVER['PHP_SELF'];
echo "<br>";
echo $_SERVER['SERVER_NAME'];
echo "<br>";
echo $ SERVER['HTTP_HOST'];
echo "<br>";
echo $ SERVER['HTTP_REFERER'];
echo "<br>";
echo
$ SERVER['HTTP_USER_AGENT'];
echo "<br>";
echo $_SERVER['SCRIPT_NAME'];
?>
```

\$_REQUEST

PHP \$_REQUEST PHP
aldagai super global bat
da, HTML inprimakia
bidali ondoren datuak
biltzeko erabiltzen
dena.

ADIBIDEA

```
<html>
<body>

<form method="post" action="<?php echo $_SERVER['PHP_SELF'];?>">
    Name: <input type="text" name="fname">
    <input type="submit">
</form>

<?php
if ($ SERVER["REQUEST METHOD"] == "POST") {
    // collect value of input field
    $name = $ REQUEST['fname'];
    if (empty($name)) { // isset($name) funtzioa erabili daiteke ere
        echo "Name is empty";
    } else {
        echo $name;
    }
}
?>
</body>
</html>
```

\$_POST

PHP \$_POST is a PHP super global variable which is used to collect form data after submitting an HTML form with method="post". \$_POST is also widely used to pass variables.

[empty \(\) vs isset \(\) vs is_null \(\)](#)

ADIBIDEA

```
<html>
<body>

<form method="post" action="<?php echo
$ SERVER['PHP SELF'];?>">
    Name: <input type="text" name="fname">
    <input type="submit">
</form>

<?php
if ($ SERVER["REQUEST METHOD"] == "POST") {
    // collect value of input field
    $name = $ POST['fname'];
    if (empty($name)) {
        echo "Name is empty";
    } else {
        echo $name;
    }
}
?>
</body>
</html>
```

\$_GET

PHP \$_GET PHP aldagai super global bat da, inprimakiaren datuak biltzeko erabiltzen dena, metodo="get"-rekin HTML inprimakia bidali ondoren.

```
<html>

<body>

<a
href="test_get.php?subject=PHP&web=W3schools.com">Test $GET</a>

</body>

</html>
```

ADIBIDEA "test_get.php"

```
<html>
<body>

<?php
echo "Study " . $GET['subject']
. " at " . $_GET['web'];
?>

</body>
</html>
```

\$_GET VS \$_POST

GET eta POSTek array bat sortzen dute (adibidez, array(gako1 => balio1, gako2 => balio2, gako3 => balio3, ...)). Array honek gako/balio bikoteak ditu, non gakoak inprimaki-kontrolen izenak diren eta balioak erabiltzailearen sarrera-datuak diren.

GET eta POST \$_GET eta \$_POST gisa tratatzen dira. Hauek superglobalak dira, hau da, beti eskuragarri daude, esparrua edozein dela ere - eta edozein funtzio, klase edo fitxategitatik atzi dezakezu ezer berezirik egin beharrik gabe.

\$_GET URL parametroen bidez uneko script-era pasatzen diren aldagaien array bat da.

\$_POST uneko script-era HTTP POST metodoaren bidez igarotzen den aldagai sorta bat da.

Noiz erabili \$_GET

GET metodoarekin inprimaki batetik bidalitako informazioa denek ikusgai dago (aldagai-izen eta balio guztiak URLan bistaratzen dira). GETek ere mugak ditu bidali beharreko informazio kopuruan. Muga 2000 karaktere ingurukoa da. Hala ere, aldagaiak URLan bistaratzen direnez, orria laster-markak jartzea posible da. Hau erabilgarria izan daiteke kasu batzuetan.

GET erabil daiteke sentikorrak ez diren datuak bidaltzeko.

Oharra: GET INOIZ ez da erabili pasahitzak edo bestelako informazio sentikorra bidaltzeko!

Noiz erabili \$_POST

POST metodoa duen inprimaki batetik bidalitako informazioa ikusezina da besteentzat (izen/balio guztiak HTTP eskaeraren gorputzean txertatuta daude) eta ez du bidaltzeko informazio kopuruaren mugarik .

Gainera, POSTak funtzionalitate aurreratuak onartzen ditu, esate baterako, zati anitzeko sarrera bitarraren laguntza, fitxategiak zerbitzarira kargatzen dituen bitartean.

Hala ere, aldagaiak URLan bistaratzen ez direnez, ezin da orria laster-marka jarri.

FORMULARIOAK

PHP - HTML formulario
sinple bat

Beheko adibidean
HTML inprimaki soil bat
bistaratzeko da
sarrerako bi eremu eta
bidaltzeko botoi batekin:

ADIBIDEA

```
<html>
<body>

<form action="welcome.php"
method="post">
Name: <input type="text"
name="name"><br>
E-mail: <input type="text"
name="email"><br>
<input type="submit">
</form>

</body>
</html>
```

ADIBIDEA "welcome.php"

```
<html>
<body>

Welcome <?php echo
$ GET["name"]; ?><br>
Your email address is: <?php
echo $_GET["email"]; ?>

</body>
</html>
```