

Software Librea

-

WEB GUNE DINAMIKOEN GARAPENA: PHP eta MySQL

GORKA JULIO



AURKIBIDEA

1. SARRERA.....	3
2. OINARRIZKO KONTZEPTUAK.....	3
2.1. Aldagaiak.....	3
2.2. Datu motak.....	4
2.3. Eragileak.....	4
2.4. Kontrol-egiturak.....	6
2.5. Funtzioak.....	7
2.6. Objektuak (object).....	8
2.7. Include eta require.....	8
3. ESTEKA INTERESGARRIAK.....	9
4. BIBLIOGRAFIA.....	9

1. SARRERA

PHP zerbitzari aldeko programazio lengoia da eta bere ezaugarri nagusia plataformarekiko independentzia da.

Zerbitzari aldeko programazio lengoaiak zerbitzarian exekutatzeko dira eta ondoren orria bezeroari bidaltzen zaio Internet bidez. Zerbitzarian exekutatzeko diren orriek datu-baseak atzitu ditzakete, sare konexioak eta beste ataza batzuk exekutatu bezeroak ikusiko duen orria sortzeko. Bezeroak, PHPren exekuzioaren ondorioz sortuko den HTML orri bat jasoko du.

PHP, zerbitzari aldeko scripten programaziorako lengoia bat da, baina ez bakarrik, asko erabiltzen diren beste lengoaiak ASP edo JSP dira eta antzeko ezaugarriak dituzte. Hala ere, hauek baino garatuagoa eta erabiliagoa da gaur egun oraindik ere.

PHP erabat ondo egokitzen da LAMP (Linux Apache MySQL PHP) inguruneetara. Horiekin batera, webgune eta aplikazio dinamikoko sortzeko ingurune egokienetako bat osatzen dute.

2. OINARRIZKO KONTZEPTUAK

PHP lengoaiaren oinarrizko kontzeptu eta ezaugarriak aztertzen joango gara orain. PHP programazio lengoia honen fitxategi luzapena .php da. Bertsioaren arabera ere izan daitekeen arren .php5 adibidez, ohikoena zenbakirik gabekoa da.

Ondoko kode-zatian PHP kode simple bat ikus daiteke. *echo* funtzioak kaxotx artean dagoena pantailarazten du eta `<?php ?>` mugatzailea erabiltzen du kodea non dagoen definitzeko. Webgunea ikusten duenak, php kodea interpretatuta ikusiko du, kasu honetan "Kaixo Mundua" esaldia.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>PHP proba</title>
  </head>
  <body>
    <?php
      echo 'Kaixo Mundua';
      /* Hau iruzkin bat da. Lerro bakarrekoa balitz // erabiliz egin zitekeen */
    ?>
  </body>
</html>
```

PHP lengoaiak, kodean existitu daitekeen edozein hutsune-mota (hutsune normalak, tabuladoreak eta lerro-bukaerak) ez du kontutan edukiko. Puntu eta koma bidez adierazten da lerro bukaera.

2.1. Aldagaiak

Gainontzeko lengoaietan ez bezala, PHPn ez da beharrezkoa aldagaiak definitzea edo erazagutzea erabili baino lehen: kodean agertzen den lehenengo momentuan, une horretan bertan aldagaia definituta geratzen da. Perl-en gertatzen den bezala PHPko aldagai guztiek aurretik '\$' ikurra eraman behar dute.

```
$ald_1 = 123;  
$ald_2 = 'kaixo';  
$ald_3 = $ald_1 * 2;
```

PHPn aldagaiak oso ahul definituak daude (*mixed* mota). Aldagai bera denboran zehar behin baino gehiagotan erabili daiteke mota ezberdinetako datuak esleitzeko.

Hala ere, aldagai-mota kasu batzuetan garrantzitsua da (eragiketak egiteko edo funtzioei deiak egiteko). Kasu hauetan, PHP lengoaiak beharrezko aldaketak automatikoki egiten ditu. Bestela ere, mota behartu daiteke aldagaiaren aurretik parentesien artean mota jarritz.

Izendapenak garrantzia du PHPn:

- Aldagaietan, maiuskulak eta minuskulak GARRANTZIA dute. Honela, *\$NireAld* aldagaia eta *\$nireald* aldagai ezberdinak dira.
- Funtzio eta hitz erreserbatuen izenetan aldiz, EZ du GARRANTZIRIK. *PRINT()* funtzioak, *print()* funtzioari erreferentzia egiten du.

2.2. Datu motak

PHP lengoaiak jarraian ohiko oinarrizko datu-motekin lan egin dezake: osokoak (int, integer), koma higikorrezko zenbakiak (float, double, real), bektoreak (array), kateak (strings), objektuak (object), Boulearrak (booleans)... Bektoreen adibide batekin, hauetako batzuk erakutsiko ditugu:

Bektoreak (array)

Bektoreak dimentsio bakarreko edo anitzekoak izan daitezke. Bektoreak definitzeko eta atzitzeko bere izena eta makoen artean bere indizea adierazi behar da. Bektore bereko elementuak, mota ezberdinekoak izan daitezke (gomendatzen ez bada ere). Adibidez [1,"kaixo!!",5.5] adierazteko:

```
$NireBek[0] = 1; //Osokoa  
$NireBek[1] = "kaixo!!"; //Katea  
$NireBek[2] = 5.5; //Koma higikorrekia
```

Bektore bat definitzean, indizea ez badugu jartzen, elementua taulan definitu gabe dagoen hurrengo posizioari esleitzen zaio (eta makoak mantentzen baditugu, aldagaia berridazten ari gara!). Aurreko adibidearekin jarraituz:

```
$NireBek[] = 8.5 // $NireArray[3] = 8.5
```

2.3. Eragileak

PHP lengoaian existitzen diren eragile ezberdinak ikusiko ditugu mota arabera sailkatuta:

Eragile Aritmetikoak

Betiko eragile aritmetikoak ditugu:

Eragilea	Izena	Emaitza
\$a + \$b	Batuketa	\$a eta \$b -ren batuketaren emaitza
\$a - \$b	Kenketa	\$a eta \$b -ren kenketaren emaitza

Eragilea	Izena	Emaitza
$a * b$	Biderketa	a eta b -ren biderketaren emaitza
a / b	Zatiketa	a eta b -ren zatiketaren emaitza
$a \% b$	Hondarra	a / b -ren zatiketaren hondarra

Automatikoki gehitu eta kentzeko $a++$ ($a = a + 1$ eragiketaren baliokidea) egiturak erabil daitezke.

Eragile logikoak

Eragile logikoek, eragileek duten balioen arabera eragiketak egiten dituzte.

Eragilea	Izena	Emaitza
a and b $a \&\& b$	ETA	Egia baldin a eta b egia badira
a or b $a b$	EDO	Egia baldin a edo b egia badira
a xor b	OR esklusiboa	Egia baldin a edo b egia badira, baina ez biak
$! a$	EZ	Egia baldin a faltsua bada

Esleipena, berdintasuna eta identitatea

PHP lengoian, balioak esleitzeko eta aldagaien arteko balioak konparatzeko 3 eragile daude:

Eragilea	Izena	Emaitza
$a = b$	Esleipena	Espresio edo aldagai baten balioa esleitzen du, b -tik a -ra.
$a == b$	Berdintasuna	Bi aldagaien balioa berdina den ala ez begiratzen du
$a === b$	Identitatea	Aldagaien balioa eta mota, berdinak diren ala ez begiratzen du

Konparaketak

Bi aldagai konparatu eta gero egiazkoa (1) ala faltsua (0) itzultzen dute.

Eragilea	Izena	Emaitza
$a != b$	Ezberdin	Egia baldin a eta b aldagaiek balio ezberdina

Eragilea	Izena	Eraitza
		badute
$\$a \neq \b	Ez identitatea	Egia baldin $\$a$ eta $\$b$ aldagaiek balio ezberdina badute edo mota ezberdinekoak badira
$\$a < \b	Txikiago	Egia baldin $\$a$ $\$b$ baino txikiagoa bada
$\$a \geq \b	Handiago edo berdin	Egia baldin $\$a$ $\$b$ baino handiagoa edo berdina bada

2.4. Kontrol-egiturak

if ... elseif ... else

if baldintza, kontrol-egitura basikoena da. *if* sententzia bati, baldintza betetzen ez denean aginduak exekutatzeko kodea gehitu diezaiokegu *else* sententzia erabiliz:

```
if (espresioa) {
    egia_komandoak
} else {
    faltsu_komandoak
}
```

Espresioa egiazkoa bada, *egia_komandoak* exekutatu da. Faltsua bada aldiz, *faltsu_komandoak*. Bi kasuetan, *if* egituraren ondoren datozen aginduak exekutatu dira.

while eta do .. while

while eta *do .. while* kontrol-egituretan zehazten den baldintza betetzen den bitartean exekutatu behar diren begiztak (egituraren gorputzean jartzen dena) zehazten dira. *while* begizta baten egitura hau da:

```
while (espresioa) {
    aginduak
}
```

Eraitza, *aginduak* sententzien errepikapena izango da *espresioa* egia den bitartean. *do .. while* egitura, *while* baten aldaketa bezala ikus daiteke, non baldintza begiztaren aurrean egiaztatu ordez, bukaeran egiten den.

```
do {
    aginduak
} while (espresioa);
```

for

for egiturak, PHP lengoaiak duen begiztarik konplikatuena da. Bere egitura, C lengoaiaren berdina da.

```
for (espresio1; espresio2; espresio3) {
    aginduak
}
```

- espresio1* begiztaren hasieratzea da. Orokorrean, hasierako balio bat ematen du aldagai bati edo batzuei (komaz bananduta). Behin bakarrik exekutatu da, hasieran, programaren exekuzioa begiztara iristen denean.

- *espresio2* begiztaren baldintza da. *Espresio2* egiazkoa den bitartean, begiztaren iterazioa jarraituko du. Iterazio bakoitzaren hasieran, baldintza ebaluatzen da, ez bada betetzen begizta amaitzen da eta *for* aginduaren ondoren dagoen kodearen exekuzioarekin jarraitzen da.
- *espresio3* begiztaren urratsa da. Iterazio bakoitzaren bukaeran exekutatzen da eta orokorrean, aldagai baten balioa aldatzen du (komaz bananduta bat baino gehiago baldin badago). Adibidea:

```
$faktoriala5 = 1;
for ($i = 2; $i <= 5; $i++) {
    $ faktoriala5 *= $i;
}
```

foreach

foreach aldagai bati, lista baten balio guztiak esleitzen dizkion iterazio desberdinetan. Bere sintaxia ondokoa da.

```
foreach (array as $aldagaia) {
    aginduak
}
```

Begiztako iterazio bakoitzean, *\$aldagaia* aldagaian *array*-ko elementu bat gordetzen da, lehenengotik hasita eta goranzko ordena batean. Adibidez:

```
$a = array (1, 2, 3, 17);
foreach ($a as $v) {
    print "\$a-ren uneko balioa: $v.\n";
}
```

switch

Baldintzako egitura da beste lengoaia batzuen *case* egituraren antzeko. Bere sintaxia hau da:

```
switch (aldagaia) {
    case balioa1:
        agindua1
    case balioa2:
        agindua2
    ...
    case balioaN:
        aginduaN
    default:
        aginduakBeste
}
```

Fluxua goitik behera lineala da, *case* egiturako balio bakoitzarekin konparatzen da eta baldintza betetzen duen kodea exekutatzen du. Baldintzarik betetzen ez bada, *default* klausulako kodea exekutatzen da. Egitura honek, *case* bakoitzeko *if* bat izatearen berdina litzateke, bat bestearen atzetik. *break* batez *switch*-etik ateratzea behartu daiteke.

2.5. Funtzioak

```
function izena ($arg_1, $arg_2, ..., $arg_n) {
    aginduak
}
```

```

return $irteera;
}

```

Parametroak balio bezala pasatzen dira, *\$arg_1, ... , \$arg_n* aldagaietan kopia lokalak egiten dira eta kopia lokal horien gainean lan egiten da, honela funtziotik ateratzerakoan, balio originalak ez dira aldatu. *return* agindua aukerazkoa da eta funtzioak irteera-balio bat bueltatzeko balio du. Funtzioaren kodean behin baino gehiagotan atera daiteke eta beti funtzioaren exekuzio-amaiera ematen du. Funtzio baten gorputzean, edozein PHP agindu egon daiteke, baita beste funtzio edo klase batzuen definizioa ere.

2.6. Objektuak (object)

Nahiz eta *Objektuei Zuzendutako Lengoia* purua ez izan, PHP5 lengoaiak aurrera pausu handiak eman ditu alor horretan, objektuak eta klaseak onartzeaz gain mota honetako lengoaien ezaugarri asko barneratu ditu eta nahiko garatua dago. Adibide bat:

```

class Pertsona {
    public $izena;
    public $abizena;

    public function __construct($izena, $abizena = "") {
        $this->izena = $izena;
        $this->abizena = $abizena;
    }

    public function agurra() {
        return "Kaixo, nire izena ".$this->izena." ".$this->abizena." da.";
    }

    static public function staticAgurra($izena, $abizena) {
        return "Kaixo, nire izena ".$izena." ".$abizena." da.";
    }
}
$egilea = new Person('Joseba', 'Sarrionandia');
echo $egilea->agurra();
echo Person::staticAgurra('Joseba', 'Sarrionandia');
//Biek 'Kaixo, nire izena Joseba Sarrionandia da.' pantailaratuko dute.

```

2.7. Include eta require

require("fitxategia"); klausularen bitartez *fitxategiaren* kodea ordezkatzen da exekutatu baino lehen. *Fitxategia* fitxategi lokala edo URL bat izan daiteke. C lengoaiako *#include*-ren antzekoa da. Ordezkapen hau behin bakarrik egiten da, *.php* fitxategiaren edukia aurreprozesatzen ari den bitartean eta exekutatu baino lehen.

Include("fitxategia"); klausula ere fitxategiaren edukiarekin ordezkatzen da, baina ordezkapena hasieran egin ordez, exekuzio garaian egiten da, programaren fluxua programako lerro horretara iristen denean.

include_once eta *require_once* funtzioak ere existitzen dira. Aukera horiek fitxategi bat behin bakarrik prozesatuko dela ziurtatzen dute, nahiz eta aurreragoko *include*-tan berriz azaldu.

3. ESTEKA INTERESGARRIAK

- Gune ofiziala: <http://www.php.net>
- PHP klaseen biltegia: <http://www.phpclasses.org>
- PHPBuilder, php baliabideak: <http://www.phpbuilder.com>
- PHP gazteleraz: <http://www.programacion.com/php/tutorial/php/>
- PHP baliabide bilduma: <http://php.resourceindex.com>
- Apache web zerbitzaria: <http://www.apache.org>
- MySQL datu basea: <http://www.mysql.com>
- *PHP* + *MySql* + *Apache* + *phpMyAdmin*:
<http://www.apachefriends.org/en/xampp.html>

4. BIBLIOGRAFIA

- Desarrollo Web con PHP 6, Apache y MySQL (Wrox (anaya Multimedia)) - Timothy Boronczyk, Elizabeth Naramore, Jason Gerner, Yann Le Scouarnec, Jeremy Stolz, Michael K. Glass. ISBN 8441526222 (2010)
- PHP y MySQL (O'reilly (anaya Multimedia)) - Michele E. Davis, Jon A. Phillips. ISBN 8441523894 (2008).