



Programimi për www server

Pjesa e parë

Prof. Asoc. Dr. Ermir Rogova

Objektivat

- Hyrje në PHP
- Funkzionet themelore

PHP8

- PHP është gjuha më e njohur e skriptimit nga ana e serverit për krijimin e faqeve dinamike në internet
- PHP është shkurtesë për PHP Hypertext Preprocessor.
- Është një shumë e popullarizuar gjuhë “open source” për server për të shkruar faqet e internetit të krijuara në mënyrë dinamike.
- Fillimisht u krijua nga Rasmus Lerdorf në 1994. Fillimisht ishte e njohur si Personal Home Page.
- Skriptet PHP ekzekutohen në server dhe rezultati dërgohet në shfletuesin e internetit si HTML.
- PHP mund të integrohet lehtësisht me sistemet e bazave të të dhënave të njohura, përfshirë MySQL, PostgreSQL, Oracle, Microsoft SQL Server, Sybase, etj.
- Versioni aktual (2025) i PHP është 8.
- PHP është gjuhë shumë e fuqishme por edhe e lehtë për tu mësuar dhe përdorur.

Çfarë mund të bëjmë me PHP

- Ka shumë më shumë gjëra që mund të bëni me PHP.
 - Ju mund të gjeneroni faqe dhe fajllë në mënyrë dinamike.
 - Ju mund të krijoni, hapni, lexoni, shkruani dhe mbyllni fajllë në server.
 - Ju mund të grumbulloni të dhëna nga një formular në internet si informacioni i përdoruesit, email, nr i telefonit, etj.
 - Ju mund t'u dërgoni email përdoruesve të faqes suaj të internetit.
 - Ju mund të dërgoni dhe të merrni cookies për të ndjekur vizitorin e faqes tuaj të internetit.
 - Ju mund të ruani, fshini dhe modifikoni informacionin në bazën tuaj të të dhënave.
 - Ju mund të kufizoni hyrjen e paautorizuar në faqen tuaj të internetit.
 - Ju mund të enkriptoni të dhëna për transmetim të sigurt përmes internetit.
- Lista nuk përfundon këtu, ka shumë gjëra të tjera interesante që mund të bëni me PHP.

Avantazhet e PHP

- Lehtë për tu mësuar:
 - PHP është e lehtë për tu mësuar dhe përdorur. Për programuesit fillestarë që sapo kanë filluar zhvillimin e faqeve për internet, PHP shpesh konsiderohet si zgjedhja e preferuar e gjuhës për të mësuar.
- Burim i hapur:
 - PHP është një projekt me burim të hapur. Është zhvilluar dhe mirëmbajtur nga një komunitet botëror i zhvilluesve që kanë vënë kodin burimor të tij në dispozicion për ta shkarkuar dhe përdorur lirisht.
- Portabiliteti:
 - PHP ekzekutohet në platforma të ndryshme si Microsoft Windows, Linux, Mac OS, etj. dhe është kompatibel me pothuajse të gjithë serverat e përdorur sot si Apache, IIS, etj.
- Performanca e shpejtë:
 - Skriptet e shkruara në PHP zakonisht ekzekutohen më shpejt sesa ato të shkruara në gjuhë të tjera të skriptimit si ASP, Ruby, Python, Java, etj.
- Komuniteti i gjerë:
 - Megjese PHP mbështetet nga komuniteti në mbarë botën, gjetja e ndihmës ose dokumentacionit në lidhje me PHP në internet është jashtëzakonisht e lehtë

Popullariteti

- Disa webfaqe të mëdha si Facebook, Yahoo, Flickr dhe Wikipedia janë ndërtuar duke përdorur PHP.
- Shumica e sistemeve kryesore të menaxhimit të përmbajtjes (CMS), si WordPress, Drupal, Joomla dhe Magento janë të ndërtuara në PHP.

Çka na duhet?

- Skripti PHP ekzekutohet në një web server që ekzekuton PHP.
- Pra, para se të filloni të shkruani ndonjë program PHP keni nevojë për
 - Web Serveri Apache
 - Motori PHP (PHP engine)
 - Serveri i bazës së të dhënave MySQL
- Ju mund t'i instaloni ato individualisht ose të zgjidhni një paketë të para-konfiguruar për sistemin tuaj operativ si Linux dhe Windows.
- Paketa të popullarizuara të para-konfiguruar janë XAMPP dhe Laragon.

Shembull i skriptës PHP

```
<?php
// Display greeting message
echo "Hello, world!";
?>
```

Standalone

```
<!DOCTYPE HTML>
<html>
<head>
  <title>PHP Application</title>
</head>
<body>
  <?php
  // Display greeting message
  echo 'Hello World!';
  ?>
</body>
</html>
```

embedded within a HTML page

Sintaksa standarde PHP

- Një skriptë PHP fillon me tagun `<?php` dhe mbaron me tagun `?>`
- Tagjet `<?php` dhe `?>` në shembullin vijues thjesht i thotë motorit PHP të trajtojë bllokun e kodit të përmbledhur si kod PHP, në vend se HTML i thjeshtë
- Çdo deklaratë në PHP përfundon me pikëpresje (;)
 - kjo i tregon motorit PHP se është arritur fundi i deklaratës aktuale

```
<?php
// Some code to be executed
echo "Hello, world!";
?>
```

Integrimi i PHP në HTML

- Fajllët PHP janë fajllë me tekst të thjeshtë me prapashtesë .php
- Brenda një fajlli PHP mund të shkruani HTML siç bëni në faqet e rregullta HTML si dhe të vendosni kodet PHP për ekzekutimin nga ana e serverit.
- Shembulli para dy sllajdeve tregoi se si mund të vendosni kodet PHP brenda HTML për të krijuar faqe dinamike të formuara mirë.
- Nëse shikoni kodin burimor të faqes që rezulton në shfletuesin tuaj, ndryshimi i vetëm që do të shihni është kodi PHP `<? php echo "Përshëndetje, botë!"; ?>` është zëvendësuar me outputin " Përshëndetje, botë!".
- Çfarë ndodhi këtu është: kur ekzekutoni këtë kod, motori PHP ekzekutoi udhëzimet midis etiketave `<? Php...?>` dhe pjesën tjetër të gjërave e le ashtu siç është.
- Në fund web serveri dërgon prodhimin përfundimtar në shfletuesin tuaj, i cili është plotësisht në HTML.

Komentet në PHP (1)

- Një koment është thjesht një tekst që është injoruar nga motori PHP.
- Qëllimi i komenteve është ta bëjnë kodin më të lexueshëm.
- Mund të ndihmojë zhvilluesin tjetër (ose ju në të ardhmen kur redaktoni kodin burimor) të kuptojë se çfarë po përpiqeshit të bënit me PHP.
- PHP mbështet komente me një rresht si dhe me shumë rreshta.

Komentet në PHP (2)

- Për të shkruar një koment me një rresht ose filloni rreshtin ose me dy thyesa (//) ose me një simbol hash (#).

```
<?php
// This is a single line comment
# This is also a single line comment
echo "Hello, world!";
?>
```

- Sidoqoftë, për të shkruar komente me shumë rreshta, fillojeni komentin me një thyesë të ndjekur nga një yll (/ *) dhe përfundojeni komentin me një yll të ndjekur nga një thyesë (* /)

```
<?php
/* This is a multiple line comment block
that spans across more than
one line */
echo "Hello, world!";
?>
```

Ndjeshmëria në shkronja

- Emrat e variablave në PHP janë të ndjeshëm në shkronja të mëdha/vogla
 - Si rezultat, variablat \$color, \$Color dhe \$COLOR trajtohen si tri variabla të ndryshme.
- Sidoqoftë, fjalët kyçe, emrat e funksioneve dhe klasave nuk janë të ndjeshme në shkronja të mëdha/vogla.
 - Si rezultat thirrja e gettype() ose GETTYPE() prodhon të njëjtin rezultat.

Variablat në PHP

- Variablat përdoren për të ruajtur të dhëna, si stringu, numra, etj.
- Vlerat e variablave mund të ndryshojnë gjatë ekzekutimit të skriptës.
- Disa gjëra të rëndësishme për të ditur rreth variablave:
 - Në PHP, një variabël nuk ka nevojë të deklarohet përpara se t'i jepet një vlerë.
 - PHP automatikisht shndërron variablën në llojin e saktë të të dhënave, varësisht nga vlera e saj.
 - Operatori i caktimit (=) përdoret për të caktuar vlerën e një ndryshoreje.
 - Në PHP variabla mund të deklarohet si: \$variabla = vlera;

Variablat – shembull

```
<?php
// Declaring variables
$txt = "Hello World!";
$number = 10;

// Displaying variables value
echo $txt; // Output: Hello World!
echo $number; // Output: 10
?>
```

Konventat e emërtimeve për variablat në PHP

- Këto janë rregullat për emërtimin e një variable në PHP:
 - Të gjitha variablat në PHP fillojnë me një shenjë \$, e ndjekur nga emri i variablës.
 - Emri i një variable duhet të fillojë me një shkronjë ose me _.
 - Emri i një variable nuk mund të fillojë me një numër.
 - Emri i një variable në PHP mund të përmbajë vetëm karaktere alfa-numerike dhe nënvizime (Az, 0-9 dhe _).
 - Emri i një variable nuk mund të përmbajë hapësira

Declarata echo

- Deklarata echo mund të prodhojë një ose më shumë stringje.
- Në terma të përgjithshëm, deklarata echo mund të shfaqë gjithçka që mund të shfaqet në shfletues, të tilla si tekst, numrat, vlerat e variablave, rezultatet e shprehjeve etj.
- Meqenëse echo është një konstrukt gjuhësor jo në të vërtetë një funksion mund ta përdorni pa kllapa p.sh. echo ose echo () .

Paraqitja e stringut

- Shembulli i mëposhtëm do t'ju tregojë se si të shfaqni një varg teksti me deklaratën echo:

```
<?php  
// Displaying string of text  
echo "Hello World!";  
?>
```

- Outputi i kodit të mësipërm PHP do të duket kështu:
 - Hello World!

Paraqitja e kodit HTML

- Shembulli i mëposhtëm do t'ju tregojë se si të shfaqni kodin HTML duke përdorur deklaratën echo:

```
<?php
// Displaying HTML code
echo "<h4>This is a simple heading.</h4>";
echo "<h4 style='color: red;'>This is heading with style.</h4>";
?>
```
- Outputi i kodit të mësipërm PHP do të duket diçka si kjo:
- This is a simple heading.
- **This is heading with style.**

Paraqitja e variablave

- Shembulli i mëposhtëm do t'ju tregojë se si të shfaqni variablat duke përdorur deklaratën echo:

```
<?php
// Defining variables
$txt = "Hello World!";
$num = 123456789;
$colors = array("Red", "Green", "Blue");

// Displaying variables
echo $txt;
echo "<br>";
echo $num;
echo "<br>";
echo $colors[0];
?>
```

- Outputi i kodit të mësipërm PHP do të duket diçka si kjo:
 - Hello World!
 - 123456789
 - Red

Stringjet në PHP

- Një string është një sekuencë e shkronjave, numrave, karaktereve speciale dhe vlerave aritmetike ose kombinimi i të gjithave këtyre
- Mënyra më e thjeshtë për të krijuar një string është vendosja e karaktereve në thonjëza, si kjo:
 - `$my_string = 'Hello World';`
- Mund të përdorni edhe thonjëza të dyfishta (") .Por, thonjëzat e vetme dhe të dyfishtë funksionojnë në mënyra të ndryshme.
- Stringjet e mbyllura në thonjëza të vetme trajtohen pothuajse fjalë për fjalë, ndërsa stringjet e kufizuara nga thonjëzat e dyfishta zëvendësojnë variablat me paraqitjet e vlerave të tyre.

Stringjet – shembull

```
<?php
$my_str = 'World';
echo "Hello, $my_str!<br>";           // Displays: Hello World!
echo 'Hello, $my_str!<br>';         // Displays: Hello, $my_str!

echo '<pre>Hello\tWorld!</pre>';     // Displays: Hello\tWorld!
echo "<pre>Hello\tWorld!</pre>";     // Displays: Hello  World!
echo 'I\'ll be back';               // Displays: I'll be back
?>
```

- Outputi do të duket kështu:
 - Hello, World!
 - Hello, \$my_str!
 - Hello\tWorld!
 - Hello World!
 - I'll be back

Manipulimi i stringjeve

- PHP ofron shumë funksione të integruara për manipulimin e vargjeve si
 - Llogaritja e gjatësisë e një stringu,
 - Gjetja e nën-stringjeve ose karaktereve,
 - Zëvendësimi një pjesë të një stringu me karaktere të ndryshme,
 - Ndarja e stringut dhe shumë të tjera
- Shembuj të disa prej këtyre funksioneve do të jepen gjatë ushtrimeve

Operatorët në PHP

- Operatorët janë simbole që i thonë procesorit PHP të kryejë veprime të caktuara.
- Për shembull, simboli i mbledhjes (+) është një operator që i thotë PHP-së të mbledh dy variabla ose vlera, ndërsa simboli më i madh se (>) është një operator që i thotë PHP-së të krahasojë dy vlera.
- PHP përdor llojet e mëposhtëm të operatorëve:
 - Aritmetikore
 - Caktues
 - Krahasues
 - Rritja dhe zbritja
 - Llogjikë
 - String
 - Varg
 - Anija kozmike (PHP7 e mbi)

Operatorët aritmetikorë

Operator	Description	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 $\$y$

Operatorët caktues

Operator	Description	Example	Is The Same As
=	Assign	$\$x = \y	$\$x = \y
+=	Add and assign	$\$x += \y	$\$x = \$x + \$y$
-=	Subtract and assign	$\$x -= \y	$\$x = \$x - \$y$
*=	Multiply and assign	$\$x *= \y	$\$x = \$x * \$y$
/=	Divide and assign quotient	$\$x /= \y	$\$x = \$x / \$y$
%=	Divide and assign modulus	$\$x \% = \y	$\$x = \$x \% \$y$

Operatorët krahasues

Operator	Name	Example	Result
==	Equal	$\$x == \y	True if $\$x$ is equal to $\$y$
===	Identical	$\$x === \y	True if $\$x$ is equal to $\$y$, and they are of the same type
!=	Not equal	$\$x != \y	True if $\$x$ is not equal to $\$y$
<>	Not equal	$\$x <> \y	True if $\$x$ is not equal to $\$y$
!==	Not identical	$\$x !== \y	True if $\$x$ is not equal to $\$y$, or they are not of the same type
<	Less than	$\$x < \y	True if $\$x$ is less than $\$y$
>	Greater than	$\$x > \y	True if $\$x$ is greater than $\$y$
>=	Greater than or equal to	$\$x >= \y	True if $\$x$ is greater than or equal to $\$y$
<=	Less than or equal to	$\$x <= \y	True if $\$x$ is less than or equal to $\$y$

Rritja dhe zvoglimi

Operator	Name	Effect
$++\$x$	Pre-increment	Increments $\$x$ by one, then returns $\$x$
$\$x++$	Post-increment	Returns $\$x$, then increments $\$x$ by one
$--\$x$	Pre-decrement	Decrements $\$x$ by one, then returns $\$x$
$\$x--$	Post-decrement	Returns $\$x$, then decrements $\$x$ by one

Operatorët Ilogjikë

Operator	Name	Example	Result
and	And	$\$x$ and $\$y$	True if both $\$x$ and $\$y$ are true
or	Or	$\$x$ or $\$y$	True if either $\$x$ or $\$y$ is true
xor	Xor	$\$x$ xor $\$y$	True if either $\$x$ or $\$y$ is true, but not both
&&	And	$\$x$ && $\$y$	True if both $\$x$ and $\$y$ are true
	Or	$\$x$ $\$y$	True if either $\$x$ or $\$y$ is true
!	Not	! $\$x$	True if $\$x$ is not true

Operatorët e stringjeve

Operator	Description	Example	Result
.	Concatenation	\$str1 . \$str2	Concatenation of \$str1 and \$str2
.=	Concatenation assignment	\$str1 .= \$str2	Appends the \$str2 to the \$str1

Operatorët e vargjeve

Operator	Name	Example	Result
+	Union	$\$x + \y	Union of $\$x$ and $\$y$
==	Equality	$\$x == \y	True if $\$x$ and $\$y$ have the same key/value pairs
===	Identity	$\$x === \y	True if $\$x$ and $\$y$ have the same key/value pairs in the same order and of the same types
!=	Inequality	$\$x != \y	True if $\$x$ is not equal to $\$y$
<>	Inequality	$\$x <> \y	True if $\$x$ is not equal to $\$y$
!==	Non-identity	$\$x !== \y	True if $\$x$ is not identical to $\$y$

Operatori anija kozmike (spaceship)

- PHP 7 prezantoi një operator të ri, anija kozmike ($\lt;=>$) i cili mund të përdoret për krahasimin e dy shprehjeve. Njihet gjithashtu si operator krahasimi i kombinuar.
- Operatori i anijes kozmike kthen 0 nëse të dy operandët janë të barabartë, 1 nëse e majta është më e madhe dhe -1 nëse e djathta është më e madhe. Në thelb ofron krahasim trepalësh.

Deklaratat kushtëzuese në PHP

- Si shumica e gjuhëve të programimit, PHP gjithashtu ju lejon të shkruani kod që kryen veprime të ndryshme bazuar në rezultatet e një testi logjik ose krahasues në kohën e ekzekutimit.
- Kjo do të thotë, ju mund të krijoni kushte prove në formën e shprehjeve që rezultojnë si të vërteta ose të pavërteta dhe bazuar në këto rezultate mund të kryeni veprime të caktuara.
- Ka disa deklarata në PHP që mund t'i përdorni për të marrë vendime:
 - Deklarata if
 - Deklarata if ... else
 - Deklarata if ... elseif else
 - Deklarata switch ... case
- Ne do të shqyrtojmë secilën nga këto deklarata në sllajdet e ardhshme.

Deklarata if

- Deklarata if përdoret për të ekzekutuar një bllok kodi vetëm nëse kushti i specifikuar rezulton i vërtetë.
- Kjo është deklarata më e thjeshtë kushtëzuese në PHP dhe mund të shkruhet si:

```
if(condition){  
    // Code to be executed  
}
```

- Shembulli i mëposhtëm do të japë "Kalo një fundjavë të këndshme!" nëse dita aktuale është e premte:

```
<?php  
$d = date("D");  
if($d == "Fri"){  
    echo "Kalo një fundjavë të këndshme!";  
}  
?>
```

Deklarata if ... else

- Ju mund ta përmirësoni procesin e vendimmarrjes duke siguruar një zgjedhje alternative përmes shtimit të një deklarate else në deklaratën if.
- Deklarata if ... else ju lejon të ekzekutoni një bllok kodi nëse kushti i specifikuar vlerësohet i vërtetë dhe një bllok tjetër kodi nëse vlerësohet i pavërtetë.

```
if(condition){  
    // Code to be executed if condition is true  
} else{  
    // Code to be executed if condition is false  
}
```

- Shembulli i mëposhtëm do të japë "Kalo një fundjavë të këndshme!" nëse dita e tanishme është e premte, përndryshe do të dalë "Kalo një ditë të mbarë!"

```
<?php  
$d = date("D");  
if($d == "Fri"){  
    echo "Kalo një fundjavë të këndshme!";  
} else{  
    echo "Kalo një ditë të mbarë!";  
}  
?>
```

Deklarata if...elseif...else

- Deklarata if...elseif...else përdoret për të kombinuar deklarata të shumta if...else

```
if(kushti1){  
    // Kodi që ekzekutohet nëse kushti1 është i vërtetë  
} elseif(kushti2){  
    // Kodi që ekzekutohet nëse kushti1 është i pavërtetë  
    dhe kushti2 i vërtetë  
} else{  
    // Kodi që ekzekutohet nëse kushti1 dhe kushti2 janë të pavërtetë  
}
```

- Shembulli i mëposhtëm do të japë "Kalo një fundjavë të këndshme!" nëse dita aktuale është e Premte, dhe "Kalofshi një të Dielë të këndshme!" nëse dita aktuale është e Dielë, përndryshe do të dalë "Kalo një ditë të mbarë!"

```
<?php  
$d = date("D");  
if($d == "Fri"){  
    echo "Have a nice weekend!";  
} elseif($d == "Sun"){  
    echo "Have a nice Sunday!";  
} else{  
    echo "Have a nice day!";  
}  
?>
```

Opeatori ternar

- Operatori ternar ofron një mënyrë të shkurtër për të shkruar deklaratat if ... else.
- Operatori ternar përfaqësohet nga simboli i pikëpyetjes (?) dhe duhen tre operandë:
 - një kusht për të kontrolluar,
 - një rezultat për të vërtetë, dhe
 - një rezultat për false.
- Për të kuptuar se si funksionon ky operator, merrni parasysht shembullin në vijim

Operatori ternar – shembull

```
<?php
if($age < 18){
    echo 'Child';    // Display Child if age is less than 18
} else{
    echo 'Adult';    // Display Adult if age is greater than or equal to 18
}
?>
```

- Duke përdorur operatorin ternar, i njëjti kod mund të shkruhet në një mënyrë më kompakte:

```
<?php echo ($age < 18) ? 'Child' : 'Adult'; ?>
```

Deklaratat switch...case

- Deklarata switch...case është një alternativë ndaj deklaratës if...elseif...else, e cila bën gati të njëjtën gjë.
- Deklarata switch...case teston një variabël ndaj një serie vlerash derisa të gjejë një përputhje dhe më pas ekzekuton bllokun e kodit që i korrespondon asaj përputhjeje.

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

Vargjet (arrays)

- Vargjet janë variabla komplekse që na lejojnë të ruajmë më shumë se një vlerë ose një grup vlerash nën një emër të vetëm të variables.
- Ekzistojnë tre lloje të vargjeve që mund të krijoni. Këto janë:
 - Varg i indeksuar - Një varg me një çelës numerik (index)
 - Varg asociativ - Një varg ku secili çelës ka vlerën e tij specifike
 - Varg shumëdimensional - Një varg që përmban një ose më shumë vargje brenda vetes.

Vargjet e indeksuara

- Një varg i indeksuar ose numerik ruan secilin element të grupit me një indeks numerik.
- Shembujt e mëposhtëm tregojnë dy mënyra për të krijuar një koleksion të indeksuar, mënyra më e lehtë është:

```
<?php
// Define an indexed array
$colors = array("Red", "Green", "Blue");
?>
```
- Kjo është ekuivalente me shembullin e mëposhtëm, në të cilin indekset caktohen manualisht:

```
<?php
$colors[0] = "Red";
$colors[1] = "Green";
$colors[2] = "Blue";
?>
```

Vargjet asociative

- Në një varg asociativ, çelësat e caktuar për vlerat mund të jenë stringje arbitrare dhe të përcaktuara nga përdoruesi.
- Në shembullin vijues vargu përdor çelësa në vend të numrave të indeksit:

```
<?php
// Define an associative array
$ages = array("Peter"=>22, "Clark"=>32, "John"=>28);
?>
```

- Shembulli i mëposhtëm është ekuivalent me shembullin e mëparshëm, por tregon një mënyrë tjetër për krijimin e vargjeve shoqëruese:

```
<?php
$ages["Peter"] = "22";
$ages["Clark"] = "32";
$ages["John"] = "28";
?>
```

Paraqitja e strukture dhe vlerave të vargut

- Ju mund të shihni strukturën dhe vlerat e çdo vargu duke përdorur një nga dy deklaratat:
 `var_dump (vargu)` ose
 `print_r (vargu)`.
- Deklarata `print_r ()`, jep disi më pak informacion.
- Vetëm përmbajtja e vargut mund të shihet me deklaratën `echo`

Renditja e vargjeve PHP

- PHP vjen me një numër funksionesh të integruara të dizenuara posaçërisht për renditjen e elementeve të vargjeve në mënyra të ndryshme si alfabetikisht ose numerikisht në rend ngjitës ose zbritës.
- Disa nga funksionet që përdoren më shpesh për klasifikimin e vargjeve:
 - `sort()` dhe `rsort()` - Për renditjen e vargjeve të indeksuara
 - `asort()` dhe `arsort()` - Për renditjen e vargjeve asociative sipas vlerës
 - `ksort()` dhe `krsort()` - Për renditje e vargjeve asociative sipas çelësit

Unazat

- Unazat përdoren për të përsëritur ekzekutimin të njëjtit bllok kodi, derisa të plotësohet një kusht i caktuar.
- Ideja themelore prapa një unaze është të automatizoni detyrat e përsëritura brenda një programi për të kursyer kohë dhe përpjekje.
- PHP mbështet katër lloje të ndryshme unazash.
 - while - kalon nëpër një bllok kodi derisa kushti të vlerësohet i vërtetë.
 - do... while - blloku i kodit ekzekutohet një herë dhe pastaj vlerësohet kushti. Nëse kushti është i vërtetë, unaza përsëritet për sa kohë që kushti i specifikuar është i vërtetë.
 - for - kalon nëpër një bllok kodi derisa numëruesi të arrijë një numër të caktuar.
 - foreach - kalon nëpër një bllok kodi për secilin element në një grup.
- Unaza foreach funksionon posaçërisht me vargje. **SHUME E DOBISHME!**

Unaza while

- Deklarata while do të përshkojë një bllok kodi derisa gjendja në deklaratën while të vlerësohet e vërtetë.

```
while(condition){  
    // Code to be executed  
}
```

- Shembulli më poshtë përcakton një unazë që fillon me $\$i = 1$.
- Laku do të vazhdojë të ekzekutohet për sa kohë që $\$i$ është më i vogël ose i barabartë me 3.
- $\$i$ do të rritet për 1 sa herë që cikli ekzekutohet:

```
<?php  
$i = 1;  
while($i <= 3){  
    $i++;  
    echo "The number is " . $i . "<br>";  
}  
?>
```

Unaza do...while

- Unaza do...while është një variant i unazës while, i cili vlerëson gjendjen në fund të çdo përsëritje të unazës.
- Në një unazë do...while blloku i kodit ekzekutohet fillimisht dhe pastaj vlerësohet kushti, nëse kushti është i vërtetë, unaza përsëritet për sa kohë që kushti është i vërtetë.

```
do{  
    // Code to be executed  
}  
while(condition);
```

- Unaza while ndryshon nga ajo do-while në një mënyrë të rëndësishme – në një unazë while, kushti që do të vlerësohet testohet në fillim të çdo përsëritje të unazës, kështu që nëse shprehja e kushtëzuar vlerësohet e pavërtetë, unaza nuk do të ekzekutohet kurrë .
- Nga ana tjetër, me një unazë do-while, ajo do të ekzekutohet gjithmonë së paku një herë, edhe nëse shprehja e kushtëzuar është false, sepse gjendja vlerësohet në fund të përsëritjes së unazës, jo në fillim.

Unaza for

- Unaza for përsërit një bllok kodi derisa të plotësohet një kusht i caktuar.
- Zakonisht përdoret për të ekzekutuar një bllok kodi për një numër të caktuar herësh

```
for(initialization; condition; increment){  
    // Code to be executed  
}
```

- Parametrat e lakut kanë këto kuptime:
 - inicializimi - përdoret për inicimin e variablave të numëruesit dhe vlerësohet një herë pa kushte para ekzekutimit të parë të trupit të lakut.
 - kushti - në fillim të çdo përsëritje, kushti vlerësohet. Nëse vlerësohet si e vërtetë, cikli vazhdon dhe deklaratat e vendosura ekzekutohen. Nëse vlerësohet si false, ekzekutimi i lakut mbaron.
 - rritja - azhurnon numëruesin e lakut me një vlerë të re. Vlerësohet në fund të çdo përsëritjeje.

Unaza foreach (1)

- Unaza foreach përdoret për të kaluar mbi elemente të vargut.

```
foreach($array as $value){  
    // Code to be executed  
}
```

- Shembulli i mëposhtëm demonstroi një unazë që do të paraqesë vlerat e vargut të dhënë

```
<?php  
$colors = array("Red", "Green", "Blue");  
  
// Loop through colors array  
foreach($colors as $value){  
    echo $value . "<br>";  
}  
?>
```

Unaza foreach (2)

- Ekziston edhe një sintaksë e unazës foreach, e cila është zgjerim i së parës.

```
foreach($array as $key => $value){  
    // Code to be executed  
}
```

- Shembull:

```
<?php  
$superhero = array(  
    "name" => "Peter Parker",  
    "email" => "peterparker@mail.com",  
    "age" => 18  
);  
  
// Loop through superhero array  
foreach($superhero as $key => $value){  
    echo $key . " : " . $value . "<br>";  
}  
?>
```



Pyetje???