VENTSPILS DIGITĀLAIS CENTRS

# KONTROLES STRUKTŪRAS

PHP

- Tās tiek izmantotas, lai **kontrolētu loģisko darbību kārtību** visā PHP skriptā.
- PHP kontroles struktūras pieļauj divus veidus, kā tās pierakstīt:
  1. ar figūriekavām atdalot komandu blokus.
  2. lietojot nobeiguma komandas bloku beigās.

- Pirmais stils ir ieteicams, kad kontroles struktūra **ietilpst tikai un vienīgi PHP koda blokā**.
- Savukārt otrais pieraksta stils **varētu būt derīgs**, kad tiek veidoti **lieli bloki** ar **jauktu PHP un HTML kodu**.
- Abi divi veidi ir savstarpēji aizvietojami jebkurā situācijā, tāpēc katra atsevišķā stila lietošana vienā vai otrā gadījumā ir tikai un vienīgi gaumes jautājums.

# PIERAKSTA STILI (PIEMĒRS)

```
if(expr) {                  if(expr):
    statements                  statements
} elseif(expr) {            elseif(expr):
    statements                  statements
} else {                    else:
    statements                  statements
}                           endif;
```

Expr = izteiksme, nosacījums.

Statements = vērtība, kas izdrukāsies.

# IF

If komanda ir standarta **nosacījuma kontroles struktūra**, kas sastopama lielākajā daļā programmēšanas valodu.

# IF SINTAKSE

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

```
if (nosacījums) {
    kods, kas parādīsies, ja nosacījums izpildās;
}
```

# IF…ELSE

VENTSPILS DIGITĀLAIS CENTRS

```
if (condition) {
    code to be executed if condition is true;}
else {
    code to be executed if condition is false;
}
```

```
if (nosacījums) {
    kods, kas parādīsies, ja nosacījums izpildās;}
else {
    kods, kas parādīsies, ja nosacījums neizpildās;
}
```

# MAINĪGĀ VĒRTĪBAS PIEŠĶIRŠANA IZMANTOJOT IF

- Ja ir nepieciešamība lietot if kontroles struktūru, kurai atkarībā no izteiksmes vērtības (true vai false) būs jāpiešķir vienam mainīgajam kāda vērtība, tad ir iespējams lietot if īso pierakstu:

```
$variable = (expr ? statement1 : statement2)
```

- Šis koda fragments nozīmē to, ka $variable vērtība būs statement1, ja expr izteiksmes vērtība būs True, un statement2, ja expr vērtība būs False.

# PIEMĒRS I

Pieņemsim, ka mums ir jāizveido PHP skripts, kas noteiks, vai skaitlis ir pāra, vai nepāra un ir jāizvada atbilstošs paziņojums.

```php
<?php
$skaitlis=10;
if ($skaitlis % 2 == 0) {
    echo "Skaitlis $skaitlis dalās ar divi, līdz ar to tas ir pāra skaitlis";
} else {
    echo "Skailis $skaitlis nedalās ar divi, līdz ar to tas ir nepāra skaitlis";
}
?>
```

- Šajā gadījumā, kā redzams, **expr** izteiksme satur vērtību $**skaitlis** % 2 == 0. % operācija nozīmē, ka skaitlis tiek dalīts un rezultātā atgriezts atlikums.
- Ja skaitlis dalās ar attiecīgo skaitli, tad atlikums, protams, ir 0.
- Savukārt == operators nodrošina abu izteiksmes pušu salīdzināšanu.

Atcerieties, ka šādās vietās vienmēr ir jālieto == nevis =

# PIEMĒRS II

VENTSPILS DIGITĀLAIS CENTRS

```
<?php
$t = date("H");
if ($t <  "20") { //The example below will output "Have a good day!" if the current time (HOUR) is less than 20
    echo "Have a good day!";
} else {
    echo "Have a good night!";
}
?>
```

# IF...ELSEIF....ELSE SINTAKSE

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

elseif (condition) {
    code to be executed if this condition is true;}

else {
    code to be executed if all conditions are
false;
}
```

```
if (nosacījums) {
    kods, kas parādīsies, ja nosacījums izpildās;}
elseif (nosacījums) {
    kods, kas parādīsies, ja šis nosacījums izpildās;}
else {
    kods, kas parādīsies, ja visi nosacījumi neizpildās;
}
```

# PIEMĒRS III

```
<?php
$t = date("H");
if ($t < "10") {
    echo "Have a good
morning!";
}
elseif ($t < "20") {
    echo "Have a good
day!";
}
else {
    echo "Have a good
night!";
}
?>
```

# SWITCH

- **switch** komanda var tikt lietota gadījumos, kad nepieciešams aprakstīt **viena mainīgā daudzas** vērtības.
- Līdzīgi kā **if** gadījumā ar divām vērtībām, šeit tādas var būt **neierobežotā daudzumā**.

Taču nevajag arī pārāk aizrauties.

# SWITCH SINTAKSE

```
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;
}
```

- *Tas notiek šādi:* sākumā ir viena izteiksme "n" (kas visbiežāk ir mainīgais). Izteiksmes vērtību salīdzina ar katra gadījuma vērtību struktūrā.
- Ja sakrīt, tiek izpildīts ar gadījumu saistītais koda bloks. Lai kods automātiski nepārietu nākamajā gadījumā, izmantojiet **atstarpi**.
- Ja nesakrīt, tiek izmantota **noklusējuma** izteiksme.

# PIEMĒRS IV

```
<?php
$milaka_krasa = "violeta";
switch ($milaka_krasa) {
    case "violeta":
        echo "Tava mīļākā krāsa ir violeta!";
        break;
    case "zila":
        echo "Tava mīļākā krāsa ir zila!";
        break;
    case "zaļa":
        echo "Tava mīļākā krāsa ir zaļa!";
        break;
    default:
        echo "Tava mīļākā krāsa nav ne violeta, ne zila, nedz arī zaļa....!";
}
?>
```

- Katra **case** vērtība tiek salīdzināta ar **switch** iekavās norādīto izteiksmes vērtību, un, ja tās sakrīt, tad tiek izpildīts koda bloks aiz attiecīgās **case** iezīmes.

- **break** atslēgas vārds signalizē, ka attiecīgais koda bloks ir **jāpārtrauc izpildīt** un **jāiet ārā** no **switch** kontroles struktūras.

- Ja nelietosim break atslēgas vārdu, tiks izpildīti arī visi pārējie koda bloki, kas seko **aiz** šīs **case** iezīmes.

- Ja neviena no **case** izteiksmēm nav vienāda ar **switch** izteiksmi, **tiek izsaukts koda bloks**, kas atrodas aiz **default** iezīmes.

# PHP CIKLI (LOOPS)

VENTSPILS DIGITĀLAIS CENTRS

PHP ir sekojoši **looping** (cikliskie) paziņojumi:

- **while**- iziet cauri koda blokam , kamēr norādītais nosacījums ir patiess;
- **do...while**- iziet cauri koda blokam vismaz vienu reizi, un pēc tam atkārto cilpu, kamēr norādītais nosacījums ir patiess;
- **for**- iziet cauri koda blokam noteiktu reižu skaitu;
- **foreach** - iziet cauri koda blokam katram masīva elementam .

# WHILE

- WHILE komanda izpilda tajā iekļauto koda bloku tik ilgi, kamēr norādītā izteiksme satur vērtību **true**.
- WHILE izteiksme tiek pārbaudīta pirms katras iterācijas (koda bloka izpildīšanas reizes) sākuma. Ja izteiksme ir vienāda ar **true**, koda bloks tiek izpildīts. Ja izteiksme ir vienāda ar **false**, PHP turpina izpildīt skriptu aiz **while** kontroles struktūras.

- Tāpat, kā if gadījumā, ja ir nepieciešams izpildīt tikai vienu rindiņu koda, var nerakstīt figūriekavas, taču labais stils prasa rakstīt tās jebkurā gadījumā.

- Gluži kā switch gadījumā ir iespējams jebkurā brīdī lekt ārā no kontroles struktūras, izmantojot break komandu. Ja ir virāki cikli (viens iekš otra), tad ar break var tikt ārā tikai no paša iekšējā cikla, nevis no visiem uzreiz.

Ja tomēr ir nepieciešamība lekt ārā uzreiz no diviem cikliem (vai vairāk), iespējams **break** komandai norādīt parametru - **ciklu skaitu (break n**). Tāpat ir pieejama arī komanda **continiue**, kas pārtrauks koda bloka izpildi un sāks nākamo iterāciju no sākuma. Ar **continiue n** var pārlekt pašreizējām iterācijām **n** iekšējajos ciklos.

# SINTAKSE

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

```
while (nosacījums ir true) {

    kods, kas parādīsies;

}
```

# PIEMĒRS V

VENTSPILS DIGITĀLAIS CENTRS

## Kods

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

## Rezultāts

The number is: 1
The number is: 2
The number is: 3
The number is: 4
The number is: 5

# DO/WHILE

DO/WHILE kontroles struktūra ir gandrīz vienāda ar **while**, vienīgā atšķirība ir tā, ka nosacījuma izteiksmes vērtība tiek pārbaudīta **cikla beigās**, tāpēc koda bloks tiks izpildīts **kā minimums vienu reizi**, neatkarīgi no izteiksmes vērtības.

# SINTAKSE

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

```
do {

kods, kas parādīsies;

} while (nosacījums ir true);
```

# PIEMĒRS VI

## Kods

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

## Rezultāts

The number is: 1
The number is: 2
The number is: 3
The number is: 4
The number is: 5

Piemērā definēts mainīgais x. DO WHILE cikls izvadīs attiecīgo rezultātu, tad palielinās mainīgo par 1. Tālāk pārbaudīs nosacījumu, attiecīgi vai x ir mazāks vai vienāds par 5, un tad cikls turpināsies tik ilgi, kamēr x būs vienāds ar 5.

# FOR

VENTSPILS DIGITĀLAIS CENTRS

For ciklu parasti lieto, ja darbība jāatkārto zināmu skaitu reižu.

FOR kontroles struktūra izmanto trīs izteiksmes.

1. sākuma izteiksme, tā tiek izpildīta līdz ar cikla sākšanos. Tā parasti tiek lietota, lai uzdotu sākuma vērtību cikla skaitītājam.

2. nosacījuma izteiksme, kura kontrolē ciklu. Tā tiek pārbaudīta pirms katra cikla izpildīšanas.
3. cikla izteiksme, tā tiek izpildīta katru reizi pēc cikla. Parasti tā tiek izmantota, lai palielinātu cikla skaitītāju.

# SINTAKSE

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

# PIEMĒRS VII

## Kods

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

## Rezultāts

The number is: 0
The number is: 1
The number is: 2
The number is: 3
The number is: 4
The number is: 5
The number is: 6
The number is: 7
The number is: 8
The number is: 9
The number is: 10

BREAK un CONTINIUE komandas darbojas tāpat, kā WHILE gadījumā, vienīgā atšķirība ir ar CONTINIUE komandu, kura liek izpildīt cikla izteiksmi pirms nosacījuma izteiksmes.

# FOREACH

FOREACH kontroles struktūra tiek izmantota, lai apstaigātu visus masīva elementus.

```
foreach(array_expression as $value) {
    statements
}
```

```
foreach(array_expression as $value):
    statements
endforeach;
```

Šis koda piemērs iet cauri visiem array_expression elementiem un piešķir katru masīva vērtību $value mainīgajam.

# SINTAKSE

VENTSPILS DIGITĀLAIS CENTRS

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

# PIEMĒRS VIII

VENTSPILS DIGITĀLAIS CENTRS

## Kods

```
<?php
$colors = array("red", "green",
"blue", "yellow");

foreach ($colors as $value) {
   echo "$value <br>";
}
?>
```

M.Kalneja

## Rezultāts

red
green
blue
yellow

VENTSPILS DIGITĀLAIS CENTRS

# Jautājumi ????