Osnove PHP-a

PHP kontrolne strukture

Kontrolne strukture su poput znakova za preusmjeravanje prometa.

Usmjeravaju tijek izvršenja koda ovisno o nekim unaprijed definiranim uvjetima.

Postoje različite kontrolne strukture, ali možemo ih kategorizirati u uvjetne i petlje.

Uvjetne nam omogućuju da odaberemo hoćemo li izvršiti izjavu ili ne.

Petlja izvršava izjavu onoliko puta koliko nam je potrebno.

Pogledajmo svaku od njih!

PHP kontrolne strukture – uvjetovane

Provjeravaju uvjet koji je uvijek logički izraz.

Ako je izraz istinit, izvršit će sve što se nalazi unutar njegovog bloka koda.

Blok koda je grupa izjava unutar **{}** zagrada.

Neke od uvjetovanih kontrolnih struktura su ***if***, ***if-else***, ***if-elseif*** i ***switch-case***.

PHP kontrolne strukture – uvjetovane (*if*)

U ovom primjeru koristimo dvije uvjetovane kontrolne strukture ***if***.

Definira se ključnom riječi ***if*** iza koje slijedi logički izraz u zagradama te blok koda unutar **{}** zagrada.

Ako je izraz istinit, izvršit će blok, inače će ga preskočiti.

```php
<?php
echo "Before the conditional.";
if (4 > 3)
{
    echo "Inside the conditional.";
}
if (3 > 4)
{
    echo "This will not be printed.";
}
echo "After the conditional.";
?>
```

PHPOsnove/Primjeri/Kontrolne_strukture/Uvjetovane/if.php

PHP kontrolne strukture – uvjetovane (*if-else*)

Uvjet možete proširiti dodavanjem ključne riječi ***else***. Ona govori PHP-u da izvrši neki blok koda ako prethodni uvjeti nisu bili zadovoljeni.

Unutar bloka ***else***, kod će se izvršiti samo ako uvjet u bloku ***if*** nije zadovoljen.

```php
<?php
if (2 > 3)
{
    echo "Inside the conditional.";
}
else
{
    echo "Inside the else.";
}
?>
```

PHPOsnove/Primjeri/Kontrolne_strukture/Uvjetovane/if_else.php

PHP kontrolne strukture – uvjetovane (*if-elseif*)

Na kraju, možete dodati i ključnu riječ *elseif* nakon čega slijedi drugi uvjet i blok koda.

Nakon ***if***-a možete dodati ***elseif*** uvjete koliko god puta želite. Ako dodate ***else***, on mora biti posljednji u lancu uvjeta.

Također imajte na umu da će, čim PHP nađe uvjet koji je istinit (*true*), prestati provjeravati ostale uvjete.

```php
<?php
if (4 > 5) {
    echo "Not printed";
} elseif (4 > 4) {
    echo "Not printed";
} elseif (4 == 4) {
    echo "Printed.";
} elseif (4 > 2) {
    echo "Not evaluated.";
} else {
    echo "Not evaluated.";
}
if (4 == 4) {
    echo "Printed";
}
```

PHPOsnove/Primjeri/Kontrolne_strukture/Uvjetovane/if_elseif.php

PHP kontrolne strukture – uvjetovane (*switch-case*)

Još jedna kontrolna struktura slična ***if-else*** je ***switch-case***.

Ova struktura provjerava samo jedan izraz i izvršava blok ovisno o njegovoj vrijednosti.

Switch uzima izraz, u većini slučaja varijablu, a zatim definira niz slučajeva.
Kada slučaj odgovara trenutnoj vrijednosti izraza, izvršava se kod unutar njega.
Čim PHP pronađe izjavu o prekidu (***break***), izlazi iz kontrolne strukture.
U slučaju da nijedan slučaj nije jednak izrazu, PHP izvršava zadanu postavku (***default***), ako ona postoji, ali to nije obavezno definirati.

```php
<?php
$title = 'Twilight';
switch ($title) {
    case 'Harry Potter':
        echo "Nice story, a bit too long.";
        break;
    case 'Lord of the Rings':
        echo "A classic!";
        break;
    default:
        echo "Dunno that one.";
        break;
}
?>
```

PHPOsnove/Primjeri/Kontrolne_strukture/Uvjetovane/switch_case.php

PHP kontrolne strukture – vježba 1

- Definirajte varijable **a**, **b** i **c**, te im istim redoslijedom dodijelite vrijednosti 5,10 i 15.
- Koristeći uvjetovani tip kontrolne strukture provjerite je li vrijednost **b** između **a** i **c**.
- Ako je uvjet istinit, ispišite da je **b** između **a** i **c**, a ako je uvjet lažan ispišite da nije.
- Kod mora raditi i ako zamijenimo vrijednosti u varijablama **a** i **c**.

PHPOsnove/Vježbe/Vježba6/

PHP kontrolne strukture – vježba 2

- Koristeći uvjetovani tip kontrolne strukture **switch** ispišite koji je trenutno dan u tjednu.
- Za ispravno izvršenu vježbu koristite PHP funkciju **date()**. Nazivi dana moraju biti na hrvatskom jeziku.

PHPOsnove/Vježbe/Vježba7/

PHP kontrolne strukture – petlje

Petlje su kontrolne strukture koje omogućuju izvršavanje određene izjave nekoliko puta, tj. onoliko puta koliko vam je potrebno.

Možete ih koristiti u nekoliko različitih scenarija, ali najčešći je u interakciji s nizovima.

Na primjer, zamislite da imate niz s elementima, ali ne znate što je u njemu. Želite ispisati sve njegove elemente pa pomoću petlje iterirate kroz niz.

Postoje četiri vrste petlje. Svaka od njih ima svoje načine upotrebe, ali općenito, jednu vrstu petlje možete transformirati u drugu.

PHP kontrolne strukture – petlje (*while*)

While je najjednostavnija petlja. Izvodi blok koda sve dok izraz u uvjetu ne poprimi vrijednost ***false***.

U ovom primjeru prvo definiramo varijablu **$i** te joj dodijelimo vrijednost 1.

Zatim imamo uvjet u petlji **$i < 4**. Petlja će izvršavati blok koda dok je uvjet istinit(***true***).

Kao što vidite, unutar petlje svaki put povećavamo vrijednost **$i** za 1, tako da petlja završava nakon 4 ponavljanja.

Kada vrijednost **$i** dosegne 4, uvjet postaje lažan (***false***), te se time petlja završava.

```php
<?php
$i = 1;
while ($i < 4) {
    echo $i . " ";
    $i++;
}
?>
```

PHPOsnove/Primjeri/Kontrolne_strukture/Petlje/while.php

PHP kontrolne strukture – petlje (*do-while*)

Petlja ***do-while*** je vrlo slična petlji ***while***.

Jedina je razlika što će se blok koda barem jednom izvršiti kada je uvjet lažan (**false**).

U primjeru su definirane dvije petlje s istim izrazom i blokom koda, ali ako ih izvršite, vidjet ćete da je samo kod unutar ***do-while*** izvršen.

U oba slučaja izraz je od početka lažan (**false**), pa ***while*** ni ne ulazi u petlju, dok ***do-while*** ulazi u petlju jednom.

```php
<?php
echo "with while: ";
$i = 1;
while ($i < 0) {
    echo $i . " ";
    $i++;
}

echo "with do-while: ";
$i = 1;
do {
    echo $i . " ";
    $i++;
} while ($i < 0);
?>
```

PHPOsnove/Primjeri/Kontrolne_strukture/Petlje/do_while.php

PHP kontrolne strukture – petlje (*for*)

Petlja ***for*** je najkompleksnija od sve četiri petlje.

Definira inicijalni izraz, stanje izlaza i kraj iteracijskog izraza.

Kad PHP prvi put dolazi do petlje, izvršava ono što je definirano kao inicijalni izraz. Zatim ispituje uvjet i, ako je rezultat istinit (**true**), ulazi u petlju.

```php
<?php
for ($i = 1; $i < 10; $i++) {
    echo $i . " ";
}
?>
```

PHPOsnove/Primjeri/Kontrolne_strukture/Petlje/for.php

PHP kontrolne strukture – petlje (*foreach*)

Posljednja, ali ne najmanje bitna, vrsta petlje je ***foreach***. Ova petlja se koristi za nizove i omogućuje iteriranje niza u cijelosti , čak i ako ne znate njegove ključeve. Postoje dvije mogućnosti za sintaksu, kao što možete vidjeti u sljedećim primjerima.

Petlja ***foreach*** prihvaća niz - u ovom slučaju **$names** - i određuje varijablu koja će sadržavati vrijednost unosa niza. Možete vidjeti da ne trebamo specificirati nijedan krajnji uvjet, jer će PHP po broju elemenata u nizu znati koliko puta treba iterirati. Po želji možete odrediti varijablu koja sadrži ključ svake iteracije, kao u drugoj petlji.

```php
<?php
$names = ['Harry', 'Ron', 'Hermione'];
foreach ($names as $name) {
    echo $name . " ";
}
foreach ($names as $key => $name) {
    echo $key . " -> " . $name . " ";
}
?>
```

PHPOsnove/Primjeri/Kontrolne_strukture/Petlje/foreach.php

PHP petlje – vježba 1

Koristeći petlju **while**, ispišite prvih deset brojeva.

Koristeći petlju **for**, ispišite sve parne brojeve do 100.

PHPOsnove/Vježbe/Vježba8/

PHP petlje – vježba 2

- Definirajte varijablu **names** i dodijelite joj niz koji sadrži pet imena.
- Koristeći petlju **foreach**, iz niza ispišite ključeve i pripadajuće im vrijednosti.

PHPOsnove/Vježbe/Vježba9/

Hvala na pažnji!