Wprowadzenie do instrukcji sterujących i funkcji

Wprowadzenie do instrukcji sterujących i funkcji

Instrukcje sterujące i funkcje są kluczowymi elementami PHP, które pozwalają na dynamiczne wykonywanie kodu w zależności od warunków i na organizowanie powtarzalnych operacji w moduły.

Instrukcje sterujące umożliwiają podejmowanie decyzji (if, switch) oraz powtarzanie operacji (for, while, foreach).
Funkcje pozwalają na podział kodu na fragmenty wielokrotnego użytku, zwiększając czytelność i zmniejszając powtarzalność kodu.

Instrukcje warunkowe

Instrukcje warunkowe pozwalają na wykonanie różnych bloków kodu w zależności od spełnienia określonych warunków.

Instrukcja if

Instrukcja if sprawdza warunek. Jeśli jest on prawdziwy (true), kod wewnątrz {} zostanie wykonany.

```php
<?php
$wiek = 20;

if ($wiek >= 18) {
    echo "Jesteś pełnoletni.";
}
?>
```

Działanie: Jeśli $wiek jest większe lub równe 18, skrypt wyświetli „Jesteś pełnoletni.”.

Instrukcja if-else

Dodanie else pozwala na określenie alternatywnego kodu, gdy warunek nie zostanie spełniony.

```php
<?php
$liczba = 7;

if ($liczba % 2 == 0) {
    echo "Liczba jest parzysta.";
} else {
    echo "Liczba jest nieparzysta.";
```

```
}
?>
```

Działanie: Jeśli liczba jest podzielna przez 2, zostanie wyświetlony komunikat o jej parzystości, w przeciwnym razie informacja, że jest nieparzysta.

Instrukcja if-elseif-else

Umożliwia sprawdzenie wielu warunków.

```php
<?php
$punkty = 85;

if ($punkty >= 90) {
    echo "Ocena: 5";
} elseif ($punkty >= 75) {
    echo "Ocena: 4";
} elseif ($punkty >= 50) {
    echo "Ocena: 3";
} else {
    echo "Ocena: 2";
}
?>
```

Działanie: Skrypt przypisuje ocenę w zależności od liczby punktów.

Instrukcja switch

switch jest alternatywą dla if-elseif, gdy sprawdzamy wartość jednej zmiennej.

```php
<?php
$dzien = "środa";

switch ($dzien) {
    case "poniedziałek":
        echo "Dzisiaj jest poniedziałek.";
        break;
    case "środa":
        echo "Dzisiaj jest środa.";
        break;
    default:
        echo "To nie jest poniedziałek ani środa.";
}
?>
```

Działanie: Wykonuje kod dla odpowiedniego przypadku, a default obsługuje sytuację, gdy żadna wartość nie pasuje.

Pętle w PHP

Pętle pozwalają powtarzać kod wielokrotnie.

Pętla for

Używana, gdy znana jest liczba iteracji.

```php
<?php
for ($i = 1; $i <= 5; $i++) {
    echo "Iteracja: " . $i . "<br>";
}
?>
```

Działanie: Wykona się 5 razy, zwiększając $i o 1 w każdej iteracji.

Pętla while

Wykonuje kod dopóki warunek jest spełniony.

```php
<?php
$i = 1;
while ($i <= 5) {
    echo $i . "<br>";
    $i++;
}
?>
```

Działanie: Wyświetli liczby 1-5, ponieważ warunek $i <= 5 jest spełniony.

Pętla do-while

Gwarantuje wykonanie kodu przynajmniej raz.

```php
<?php
$i = 1;
do {
    echo $i . "<br>";
    $i++;
} while ($i <= 5);
?>
```

Działanie: Nawet jeśli warunek nie jest spełniony, kod wykona się przynajmniej raz.

Pętla foreach

Najczęściej używana do iteracji po tablicach.

```php
<?php
$owoce = ["Jabłko", "Banan", "Gruszka"];

foreach ($owoce as $owoc) {
    echo $owoc . "<br>";
}
?>
```

Działanie: Dla każdego elementu tablicy owoce wykona echo.

Funkcje w PHP

Funkcje pozwalają grupować kod i używać go wielokrotnie.

Tworzenie funkcji

```php
<?php
function powitanie() {
    echo "Witaj w PHP!";
}

powitanie();
?>
```

Działanie: Po wywołaniu powitanie() funkcja wypisze tekst.

Funkcja z parametrami

```php
<?php
function przywitaj($imie) {
    echo "Witaj, " . $imie . "!";
}

przywitaj("Jan");
?>
```

Działanie: przywitaj("Jan") wypisze „Witaj, Jan!”.

Funkcja zwracająca wartość

```php
<?php
function suma($a, $b) {
    return $a + $b;
```

```php
}

$wynik = suma(5, 10);
echo "Wynik: " . $wynik;
?>
```

Działanie: Funkcja zwraca sumę argumentów.

Funkcja z wartością domyślną

```php
<?php
function witaj($imie = "Gość") {
    echo "Witaj, " . $imie . "!";
}

witaj();
witaj("Anna");
?>
```

Działanie: Gdy funkcja jest wywołana bez argumentu, użyje wartości domyślnej "Gość".

Zadania do wykonania

1. Napisz warunek if, który sprawdzi, czy liczba 30 jest większa od 20.
2. Stwórz instrukcję switch, która wyświetli dzień tygodnia na podstawie liczby 1-7.
3. Napisz pętlę for, która wyświetli liczby od 1 do 20.
4. Stwórz pętlę while, która wypisze liczby od 10 do 0.
5. Napisz pętlę foreach, która wyświetli elementy tablicy ["pies", "kot", "ryba"].
6. Stwórz pętlę for, która zatrzyma się na liczbie 7 używając break.
7. Napisz pętlę for, która pominie liczbę 4 używając continue.
8. Napisz funkcję poleProstokata($a, $b), która obliczy pole prostokąta.
9. Napisz funkcję srednia($tablica), która obliczy średnią wartości w tablicy.
10. Stwórz funkcję czyParzysta($liczba), która zwróci true, jeśli liczba jest parzysta.
11. Napisz funkcję potega($a, $b), która podnosi liczbę $a do potęgi $b.
12. Stwórz funkcję czyPelnoletni($wiek), zwracającą true, jeśli wiek ≥ 18.
13. Napisz funkcję zwracającą większą z dwóch liczb.
14. Stwórz funkcję obliczającą silnię (n!).
15. Napisz funkcję zwracającą liczbę znaków w napisie.