**Лабораторна робота №2**

**Умовні оператори у PHP**

Умовні оператори php представлені трьома основними конструкціями:

* оператор умови **if,**
* оператор перемикання **switch**
* і **тернарний оператор.**

**PHP оператор if**

if (условие)

 {

 *// якщо умова істинна*

 operator1;

 operator2;

 }

 elseif(умова)

 {

 operator1;

 ...

 }

 else

 {

 *// якщо умова хибна*

 operator1;

 operator2;

 }

**Скорочений варіант:**

<?php

$a=1;

if ($a==1)

{echo"а дорівнює 1";

$a++;}

**Оператор IF ELSE:**

<?php

$a=1;

if ($a==1)

{echo"а дорівнює 1";

$a++;}

else

{echo"а не дорівнює 1";

 $a--;}

**Оператор IF ELSEIF:**

<?php

$a=2;

if ($a==1)

{echo"а дорівнює 1";

$a++;}

elseif($a==0)

{echo"а дорівнює 0";

}

else

{echo"а не дорівнює ні 1, ні 0 ";

$a--;}

* Скорочений синтаксис може не містити частини конструкції з else і не містити додаткової умови elseif
* Замість службового слова elseif можна писати else if (окремо)
* В одній конструкції if може бути кілька elseif. Буде виконано перше зустрілося вираз elseif рівне TRUE.
* У разі наявності альтернативного умови elseif конструкція else повинна йти останньої в синтаксисі.

! В умовному операторі може використовуватися двокрапка **:** замість фігурних дужок. В такому випадку закінчується оператор службовим словом **endif:**

<?php

$x=1;

$y=2;

if($x > $y):

 echo $x." більше, нуж ".$y;

elseif($x == $y): // при використанні":" неможна писати окремо else if

 echo $x." дорівнює ".$y;

else:

 echo $x." не > и не = ".$y;

endif;

**! При використанні в конструкції двокрапки замість фігурних дужок elseif не можна писати в два слова!**

**ЛОГІЧНИХ ОПЕРАЦІЇ В УМОВАХ**

**В умові if в дужках можуть бути присутніми наступні операції:**

== дорівнює

!= не дорівнює

> більше

< меньше

>= більше, дорівнює

<= менше, дорівнює

**Завдання 1**

Виводити переклад кольорів з англійської мови на українську, перевіряючи значення змінної (в якій присвоєно колір: $ a = 'blue')

**Завдання 2**

Знайти максимальне з трьох чисел

**ОПЕРАЦІЇ ПОРІВНЯННЯ І ПРАВИЛО БРЕХНІ**

У конструкції **if** в дужках має бути логічний вираз або змінна, які розглядаються з точки зору алгебри логіки, яка повертає значення або **true або false**

Тобто в якості умови може виступати єдина змінна. Розглянемо приклад:

<?php

$a=1;

 if ($a) {

 echo $a;

 }

У прикладі транслятор мови php розглядатиме змінну в дужках на правило брехні:

**Правило БРЕХНІ або що вважається помилковим**:

логічне False

цілий нуль (0)

дійсний нуль (0.0)

порожній рядок і рядок «0»

масив без елементів

об'єкт без змінних

спеціальний тип NULL

Таким чином, в розглянутому прикладі змінна $ a дорівнює одиниці, відповідно умова буде істинним і оператор echo $ a; виведе на екран значення змінної.

**Завдання 3**

Дана змінна а зі строковим значенням. Якщо а дорівнює імені, то виводити «Привіт, ім'я!», Якщо а дорівнює пустому значенням, то виводити «Привіт, незнайомцю!»

**ЛОГІЧНІ КОНСТРУКЦІЇ І АБО І НЕ В умовний оператор**

1. Іноді необхідно передбачити виконання одночасно декількох умов. Тоді умови з'єднуються логічним оператором І - &&:

$a=1;

if ($a>0 && $a<5) {

 echo "a > 0 и a < 5";

}

1. При необхідності виконання однієї з умов застосовується логічний оператор АБО - ||:

$ A = 1;

if ($ a> 0 || $ a> 1) {

echo "a> 0 або a> 1";

}

1. Для вказівки хибності умови використовується логічний оператор НЕ -!:

$ A = 1;

if (! ($ a <0)) {

echo "a не" 0 ";

}

**Оператор Switch PHP**

Оператор **switch** або «перемикач» замінює кілька поспіль конструкцій **if**. При цьому він порівнює одну змінну з безліччю значень. Таким чином, це найбільш зручний засіб для організації мультіветвленія.

switch($змінна){

case "значення1": оператор1;

 break;

case "значення2": оператор2;

 break;

case "значення3": оператор3;

 break;

[default: оператор4;

 break;]

}

Оператор може перевіряти як строкові значення (тоді вони вказуються в лапках), так і числові (без лапок).

Оператор **break** в конструкції обов'язковий. Він здійснює вихід з конструкції при істинній умові і виконання відповідної умови оператору. Без **break** будуть виконуватися оператори всіх **case** незалежно від їх істинності.

<?php

$a=5;

switch($a)

{

 case 1: echo "$a=1";

 break;

 case 2: echo "$a=2";

 break;

 case 3: echo "$a=3";

 break;

 default: echo "не дорівнює 1,2,3";

 break;

}

Приклад: заданий масив з повними чоловічими іменами. Перевіряти перший елемент масиву і в залежності від імені, видавати вітання з коротким ім'ям.

<?php

$names=array("Иван","Петро","Семен");

switch($names[0]){

case "Петр":

 echo "Привіт, Петро!";

 break;

case "Иван":

 echo "Привіт, Иване!";

 break;

case "Семен":

 echo "Привет, Семене!";

 break;

default:

 echo "Привет, $names[0]!";

 break;

}

**Завдання 4:**

Створіть змінну $ day і надайте їй довільне числове значення

За допомогою конструкції switch виведіть фразу «Це робочий день», якщо значення змінної $ day потрапляє в діапазон чисел від 1 до 5 (включно)

Виведіть фразу «Це вихідний день», якщо значення змінної $ day одно числах 6 або 7

Виведіть фразу «Невідомий день», якщо значення змінної $ day не влучає у діапазон чисел від 1 до 7 (включно)

**Тернарний оператор PHP**

Тернарний оператор, тобто з трьома операндами, має досить простий синтаксис, в якому зліва від знака? записується умова, а справа - два оператора розділені знаком:, зліва від знака виконується оператор при істинності умови, а праворуч від знака: виконується оператор за помилкової умови.

умова ? оператор1 : оператор2;

Приклад: якщо значення змінної speed менше 60, то виводити повідомлення «Показання швидкості в нормі», інакше виводити повідомлення «Швидкість перевищена!»

<?php

$speed = 45;

echo ($speed < 60) ? "Показання швидкості у нормі" : "Швидкість перевищено!";

**Завдання 5**

Виконати попереднє завдання 4, використовуючи тернарний оператор і логічні оператори