

МДК 02.02. Web-программирование. Язык PHP

Управляющие конструкции

Выражения

Выражение представляет собой сочетание значений, переменных, операторов и функций, в результате вычисления которого выдается новое значение.

Пример

$$y = 3 * (\text{abs}(2 * x) + 4);$$

Конструкция if

Синтаксис конструкции if

```
<?php
```

```
if (логическое выражение) оператор;
```

```
?>
```

Оператор, следующий за конструкцией if, будет исполнен, если логическое выражение истинно (true)

Пример

```
<?php
    $a=5;
    $b=3;
    if ($a > $b) echo "значение а больше, чем b";
?>
```

Логическое выражение может включать несколько условий.

Пример

```
<?php  
$a=1;  
$b=3;  
if (isset ($a) and $a < $b and $a>0)  
echo "значение a меньше, чем b";  
?>
```

Если необходимо выполнить блок операторов, то их нужно поместить в фигурные скобки {...}

```
<?php
```

```
    if (логическое выражение) {
```

```
        оператор1;
```

```
        оператор2;
```

```
    }
```

```
?>
```

Пример

```
<?php
$a=5;
$b=3;
if (isset($a, $b) and $a > $b)
{
    echo "a больше b";
    $b = $a;
}
?>
```

Конструкция if-else

Синтаксис конструкции if-else

if (логическое_выражение)

инструкция_1;

else

инструкция_2;

Если логическое_выражение истинно, то выполняется инструкция_1, а иначе – инструкция_2.

Если инструкция_1 или инструкция_2 состоят из нескольких команд, то они заключаются в фигурные скобки.

Пример

```
<?php
$a=5;
$b=3;
if (isset($a, $b) and $a > $b)
{
    echo "a больше b";
    $b = $a;
}
else echo "укажите значение переменной";
?>
```

Конструкция **elseif**

elseif – это комбинация конструкций **if** и **else**, расширяет условную конструкцию **if-else**.

Синтаксис конструкции **elseif**:

if (логическое_выражение_1)

оператор_1;

elseif (логическое_выражение_2)

оператор_2;

else

оператор_3;

Пример

```
<?php
$a=1;
$b="1";
if ($a > $b) {
    echo "a больше, чем b";
} elseif ($a === $b) {
    echo "a равен b";
} else {
    echo "a меньше, чем b";
}
?>
```

Альтернативный синтаксис:

if (логическое_выражение):

команды;

elseif (другое_логическое_выражение):

команды;

else:

команды;

endif

Пример

```
<?php
$a=5;
$b=2;
$c=3;
if ($a<$b):
    echo $b=$a;
elseif($b<$c):
    echo $b=$c;
else:
    echo $c=$a;
endif
?>
```

Оператор ?

Оператору ? передается выражение, которое он должен вычислить, и два выполняемых оператора: один для выполнения, когда результат вычисления выражения TRUE, а другой – когда FALSE.

Пример

```
<?php
```

```
echo $a <= 5 ? "Требуется дозаправка" : "Топлива еще  
достаточно";
```

```
?>
```

Циклы в РНР

Цикл — разновидность управляющей конструкции, предназначенная для организации многократного исполнения набора инструкций.

Итерация — это выполнение операторов внутри тела цикла.

Виды циклов

1. Цикл с предусловием (while);
2. Цикл с постусловием (do-while);
3. Цикл со счетчиком (for);
4. Специальный цикл перебора массивов (foreach).

При использовании циклов используются операторы **break** и **continue**.

break прерывает работу всего цикла,
continue прерывает текущую итерацию.

Цикл с предусловием `while`

Принцип работы:

1. Вычисляется значение логического выражения.
2. Если значение истинно, выполняется тело цикла, в противном случае - переход на следующий за циклом оператор.

Синтаксис цикла с предусловием:

`while` (логическое_выражение)

инструкция;

Пример

```
<?php  
$x=0;  
while ($x++<10) echo $x;  
?>
```

Группирование операторов внутри тела цикла **while**

1 способ

while (логическое_выражение):

{

инструкция1;

инструкция2;

}

2 способ

while (логическое_выражение):

инструкция;

...

endwhile;

Пример

```
<?php
    $x=0;
    while ($x<10)
    {
        $x++;
        echo $x.'<br>';
    }
?>
```

Цикл с постусловием do while

Цикл проверяет условие после каждой итерации. Тело цикла выполняется хотя бы один раз.

Синтаксис цикла с постусловием:

do

{

тело_цикла;

}

while (логическое_выражение);

Пример

```
<?php  
$x = 1;  
do {  
    echo $x;  
} while ($x++<10);  
?>
```

Цикл со счетчиком **for**

Используется для выполнения тела цикла определенное число раз.

Синтаксис цикла **for**:

```
for (инициализирующие_команды; условие_цикла;  
команды_после_итерации)  
{  
    тело цикла;  
}
```

Инициализирующие команды, условие цикла и итерационные команды могут состоять из нескольких выражений, разделенных запятыми.

Тело цикла, состоящее из нескольких выражений должно быть заключено в фигурные скобки.

Пример 1

```
<?php
    for ($x=0; $x<10; $x++) echo $x;
?>
```

Пример 2

```
<?php
    for ($x=0; $x++<10;) echo $x;
?>
```

Пример 3

```
<?php
    for ($x=0, $y=0; $x<10; $x++, $y++) echo $x;
?>
```

Альтернативный синтаксис цикла for

for (инициализирующие_команды; условие_цикла;
команды_после_итерации):

операторы;

endfor;

Конструкция **break**

Для управления работой цикла существует конструкция **break**, которая осуществляет выход из цикла.

Пример

```
<?php
$x=0;
while ($x++<10) {
if ($x==3) break;
echo "Итерация $x<br>";
}
// Когда $x равен 3, цикл прерывается
?>
```

Конструкция continue

Завершает текущую итерацию цикла и переходит к новой (если выполняется условие цикла для цикла с предусловием).

Пример

```
<?php
$x=0;
while ($x++<5) {
if ($x==3) continue;
echo "Итерация $x<br>";
}
// Цикл прервется только на третьей итерации
?>
```

Конструкции выбора

Предназначена для выбора действий в зависимости от значения указанного выражения.

Используется, если предполагаемых вариантов много, и для каждого варианта нужно выполнить определенные действия.

Синтаксис конструкции switch-case

```
switch (выражение) {  
    case значение1: команды1; [break;]  
    case значение2: команды2; [break;]  
    ...  
    case значениеN: командыN; [break;]  
    [default: команды_по_умолчанию; [break]]  
}
```

Принцип работы конструкции switch-case

1. Вычисляется значение выражения.
2. Конструкция switch выполняется поэтапно. Когда конструкция case находит значение, которое соответствует значению выражения switch, РНР исполняет определенные конструкции.
3. РНР продолжает исполнять конструкции до конца блока switch, пока не встречается оператор break.
4. Если ни одно значение из набора не совпало со значением выражения, тогда выполняется блок default, если он указан.

Пример 1

```
<?php
    $x=1;
    switch ($x) {
    case 0:
        echo "x=0<br>";
    case 1:
        echo "x=1<br>";
    case 2:
        echo "x=2<br>";
    }
?>
```

Пример 2

```
<?php
    $x=0;
    switch ($x) {
    case 0:
        echo "x=0<br>"; break;
    case 1:
        echo "x=1<br>"; break;
    case 2:
        echo "x=2<br>"; break;
    }
?>
```

Пример 3

```
<?php
$х="Тыква";
switch ($х) {
case "Яблоко":
    echo "Это Яблоко"; break;
case "Груша":
    echo "Это Груша"; break;
case "Арбуз":
    echo "Это Арбуз"; break;
default: echo "Значение не найдено";
}
?>
```