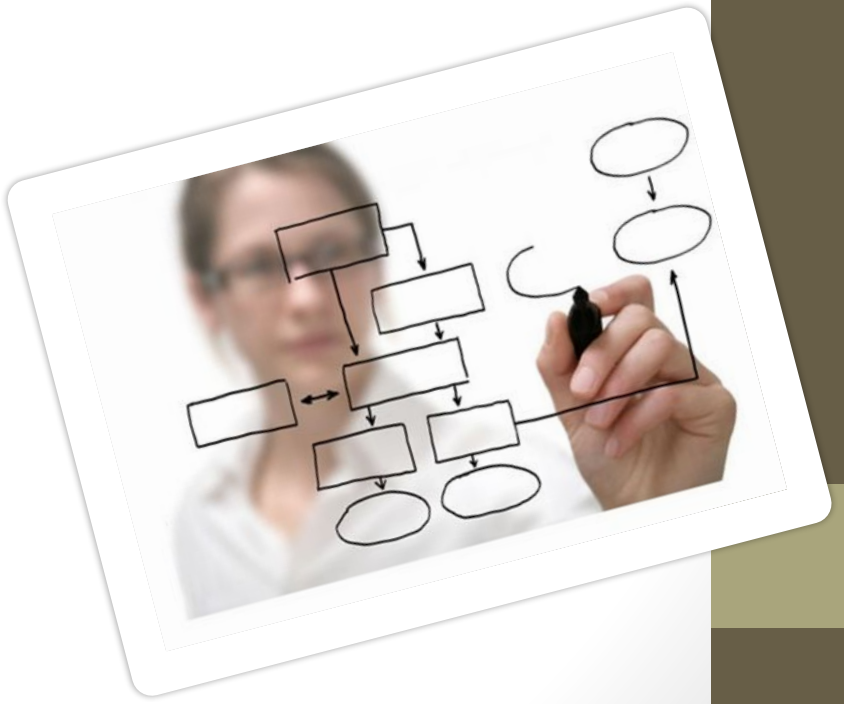
Управляющие структуры



Управляющие конструкции языка

Это наборы служебных слов, позволяющие изменить ход выполнения скрипта.

Условно делятся на следующие виды:

- ✔ Конструкции бинарного выбора;
- ✔ Конструкции множественного выбора;
- ✔ Конструкции повторения;
- ✔ Конструкции включения.

Бинарный выбор

При создании кода часто требуется выполнять различные действия на основе некоторого выбора. В PHP это можно делать с помощью условных операторов – оператора if, оператора if … else и оператора elseif.

if – этот оператор используется для выполнения блока кода, когда выполняется условие (true).

if...else – этот оператор используется для выполнения блока кода, когда условие выполняется (true), или для выполнения другого блока кода, когда условие не выполняется (false).

elseif – комбинация if и else. Оператор расширяет оператор if, чтобы выполнялся другой оператор в случае, если исходное выражение if оценивается как FALSE. В отличие от else он будет выполнять альтернативное выражение, только если условное выражение elseif оценивается как TRUE.

Пример использования конструкции if:

```
<?php
$number = 5;
if ($number <= 10) {
        echo "Число меньше или равно 10";
        }
?>
```

Пример использования конструкции If – else:

```
<?php
$number = 15;
if ($number <= 10) {
        echo "Число меньше или равно 10.";
        } else {
        echo "Число больше 10";
    } ?>
```

elseif

Часто возникает необходимость выполнять каскадную последовательность проверок с вложенными операциями if:

```
if ($day == 1)
    print ("Понедельник");
    else
      if ($day == 2)
      print ("Вотрник");
      else
            if ($day == 3)
            print ("Среда");
            ...
```

Для оптимизации этой конструкции создана конструкция elseif:

```
If ($day == 1)
    print ("Понедельник");
elseif ($day == 2)
    print ("Вотрник");
elseif ($day == 3)
    print («Среда");
…
```

Множественный выбор (switch)

Оператор switch похож или является альтернативой для команд if...else if...else.

Оператор switch проверяет условие. Результат этой проверки определяет, какой case выполняется. switch используется обычно, когда ищут точный (равенство) результат, вместо условия больше или меньше. При проверке диапазона значений должен применяться оператор if.

Синтаксис оператора switch:

```
<?php
switch (выражение) {
    case "значение1":
    // код, который будет выполнен, если выражение = значение1;
    break;
    case "значение2":
    // код, который будет выполнен, если выражение = значение2;
    break;
    default:
    // код, который будет выполнен, если выражение не равно ни значение1, ни значение2;
    }
?>
```

```php
<?php
$number = 25;
 switch ($number) {
      case 40:
      echo "Значение \$number равно 40";
      break;
      case 25:
      echo "Значение \$number равно 25";
      break;
      default:
      echo "Значение \$number отлично от 25 и 40";
      }
?>
```

Конструкции повторения (циклы)

В программировании часто необходимо повторить один и тот же блок кода несколько раз. Это можно реализовать с помощью операторов цикла. Язык PHP содержит несколько типов операторов цикла.

Цикл while

Оператор while циклически повторяет блок кода, пока указанное условие имеет значение true. Базовый синтаксис цикла while:

```
while (условие)
    { выполняемый код; }
```

Код в цикле while будет повторно выполняться, пока условие в начале цикла имеет значение true. Блок кода, связанный с оператором while, всегда заключается в фигурные скобки.

Пример использования цикла while:

```
<?php
$number = 1;
while ($number <= 10)
    {
      print $number;
      $number += $number;
      }
?>
```

Циклы do…while…

Оператор do...while повторяет циклически блок кода, пока определенное условие принимает значение true. Т.о. оператор do…while будет выполнять блок кода, если и пока условие будет выполняться (т.е. оцениваться как true).

Цикл do...while аналогичен по своей природе циклу while; ключевое различие состоит в том, что тело цикла do...while будет обязательно выполнено как минимум один раз. Это связано с тем, что оператор условия оценивается в конце оператора цикла после выполнения тела цикла.

Базовый синтаксис цикла do…while показан ниже.

```
do {
выполняемый код; }
while (условие);
```

Выполнение кода внутри цикла do…while будет повторяться, пока условие в конце цикла будет оцениваться как true.

Цикл for

Оператор цикла for используется, когда известно, сколько раз необходимо выполнить оператор или последовательность операторов.

Базовый синтаксис цикла for:

```
for (инициализация; условие; шаг цикла) {
    выполняемый код;
    }
```

Оператор цикла for имеет три параметра: первый параметр используется для инициализации переменных, второй содержит условие, а третий включает в себя приращения, требуемые для реализации цикла.

Пример использования цикла for:

```
<?php
for ($counter=1; $counter < 5; $counter++) {
    echo "Добро пожаловать в мир PHP!<br/>";
    }
?>
```

Ключевые слова управления циклами

- **Break** – применяется внутри циклов и служит для немедленного прекращения итераций цикла. При этом управление передается следующему после цикла выражению;
- **Continue** – предназначено для немедленного перехода к следующей итерации;
- **Return** – применяется для немедленного выхода из функции и возврата значения.

Конструкции включения

Предназначены для включения в текст скрипта каких-либо данных и кода, находящихся в другом файле:

include() и **include_once()** – пытаются продолжить выполнение скрипта даже в случае отсутствия включаемого файла;

require() и **require_once()** – прекращают выполнение скрипта в случае отсутствия включаемого файла.

Префикс **_once** предотвращает включение одного и того же файла более одного раза на странице.

Завершение выполнения

Для прекращения работы программы предусмотрены следующие конструкции:

exit() – принимает в качестве фактического параметра строку или число, выводит этот параметр, а затем прекращает выполнение сценария.

die() - псевдоним для конструкции exit() и действует таким же образом.

Применяются для прекращения формирования web-страницы без дополнительных усилий по оформлению дополнительных ветвей конструкций.