

**Что такое PHP?**

PHP – это широко используемый язык сценариев общего назначения с открытым исходным кодом. Говоря проще, PHP это язык программирования, специально разработанный для написания web-приложений (сценариев), исполняющихся на Web-сервере. PHP позволяет создавать качественные Web-приложения за очень короткие сроки, получая продукты, легко модифицируемые и поддерживаемые в будущем.

## **Что означает аббревиатура PHP?**

Аббревиатура PHP означает "Hypertext Preprocessor (Препроцессор Гипертекста)". Первоначально аббревиатура PHP означала "Personal Home Pages".

**Каковы основные отличия PHP от других языков, используемых для web-разработки?**

Синтаксис языка берет начало из C, Java и Perl. Важным преимуществом языка PHP перед такими языками, как языков Perl и C заключается в возможности создания HTML документов с внедренными командами PHP.

# *Каковы основные преимущества языка PHP?*

Практический характер PHP обусловлен пятью важными характеристиками:

- традиционностью;
- простотой;
- эффективностью;
- безопасностью;
- гибкостью.

## ***Для каких операционных систем (платформ) существует PHP?***

PHP доступен для большинства операционных систем, включая Linux, многие модификации Unix (такие, как HP-UX, Solaris и OpenBSD), Microsoft Windows, Mac OS X, RISC OS, и многих других. Также в PHP включена поддержка большинства современных вебсерверов, таких, как Apache, Microsoft Internet Information Server, Personal Web Server, серверов Netscape и iPlanet, сервера Oreilly Website Pro, Caudium, Xitami, OmniHTTPd и многих других. Для большинства серверов PHP поставляется в качестве модуля, для других, поддерживающих стандарт CGI, PHP может функционировать в качестве процессора CGI.

## Какую версию PHP выбирать, чем они отличаются?

Как правило, сейчас активно используются версии PHP4 и PHP5.

Версия PHP3 уже практически не используется, поскольку значительно уступает по функциональности и производительности старшим версиям PHP.

Принципиальным отличием PHP5 от PHP4 является новое ядро Zend Engine 2. В PHP5 объектная модель была значительно переработана, добавлено много новых возможностей, благодаря которым PHP5 получил некоторые черты таких объектно-ориентированных языков, как C++ и Java. Производительность встроенных стандартных функций PHP увеличилась на 40% и более.

# *Синтаксис PHP*

```
<html>
  <head>
    <title>Пример</title>
  </head>
  <body>

    <?
    echo "Привет, я - скрипт PHP!";
    ?>

  </body>
</html>
```

Код сценария начинается после открывающего тэга `<?>` и заканчивается закрывающим `?>`

Закрывающий тег (`?>`) также подразумевает конец инструкции, поэтому два следующих фрагмента кода эквиваленты:

```
<?php
    echo "Это тест";
?>
```

```
<?php echo "Это тест" ?>
```



# Комментарии в PHP скриптах

```
<?php
```

```
echo "Это тест"; // Это однострочный коммента  
рий в стиле c++
```

```
/* Это многострочный комментарий  
еще одна строка комментария */  
echo "Это еще один тест";
```

```
echo "Последний тест"; # Это комментарий в ст  
иле оболочки Unix  
?>
```

```
<h1>Это <?php # echo "простой";?> пример.</h1>  
<p>Заголовок вверху выведет 'Это пример'.
```

```
<?php  
/*
```

```
echo "Это тест"; /* Этот комментарий вызовет п  
роблему */  
*/  
?>
```

Однострочные комментарии идут только до конца строки или текущего блока PHP-кода, в зависимости от того, что идет перед ними. Это означает, что HTML-код после // ?> БУДЕТ напечатан: ?> выводит из режима PHP и возвращает в режим HTML, но // не позволяет этого сделать.

# Переменные в PHP

Имена переменных обозначаются знаком \$.

```
<?php  
$message = "Привет, я - скрипт PHP!";  
echo $message;  
?>
```

## Типы данных в PHP

PHP поддерживает восемь простых типов данных:

Четыре скалярных типа:

- boolean (двоичные данные)
- integer (целые числа)
- float (числа с плавающей точкой или 'double')
- string (строки)

Два смешанных типа:

- array (массивы)
- object (объекты)

И два специальных типа:

resource (ресурсы)

NULL ("пустые")

Существуют также несколько псевдотипов:

- mixed (смешанные)
- number (числа)
- callback (обратного вызова)

## Выражения в PHP

Основными формами выражений являются константы и переменные. Например, если вы записываете "\$a = 100", вы присваиваете '100' переменной \$a:

```
$a = 100;
```

В приведенном примере \$a - это переменная, = - это оператор присваивания, а 100 - это и есть выражения. Его значение 100.

Выражением может быть и переменная, если ей сопоставлено определенное значение:

```
$x = 7;
```

```
$y = $x;
```

В первой строке рассмотренного примера выражением является константа 7, а во второй строке - переменная \$x, т.к. ранее ей было присвоено значение 7. \$y = \$x также является выражением.

## Операторы PHP

Операторы присвоения:

```
<?php
```

`$a = ($b = 4) + 5;` // результат: `$a` установлена значением 9, переменной `$b` присвоено 4.

```
?>
```

Комбинированные операторы:

```
<?php
```

```
$a = 3;  
$a += 5; // устанавливает $a значением 8, аналогично записи:  
$a = $a + 5;  
$b = "Hello ";  
$b .= "There!"; // устанавливает $b строкой "Hello There!", как  
и $b = $b . "There!";
```

```
?>
```

## Строковые операторы:

```
<?php
```

```
$a = "Hello ";
```

```
$b = $a . "World!"; // $b содержит строку "Hello World!"
```

```
$a = "Hello ";
```

```
$a .= "World!"; // $a содержит строку "Hello World!"
```

```
?>
```

## Управляющие конструкции языка PHP

Основными конструкциями языка PHP являются:

- Условные операторы (if, else);
- Циклы (while, do-while, for, foreach, break, continue);
- Конструкции выбора (switch);
- Конструкции объявления (declare);
- Конструкции возврата значений (return);
- Конструкции включений (require, include).



## Примеры конструкций языка PHP:

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

```
<?php  
if ($a) echo "значение a истинно (true) ";  
?>
```

## Пример цикла while:

```
<?php
    $x=0;
    while ($x++<10) echo $x;
    // Выводит 12345678910
?>
```

## Пример пользовательской функции на PHP:

```
<?php
```

```
function funct() {  
    $a = 100;  
    echo "<h4>$a</h4>";  
}  
funct();
```

```
?>
```

Сценарий выводит 100:

**100**