

PHR. Уровень 1.

Основы создания сайтов



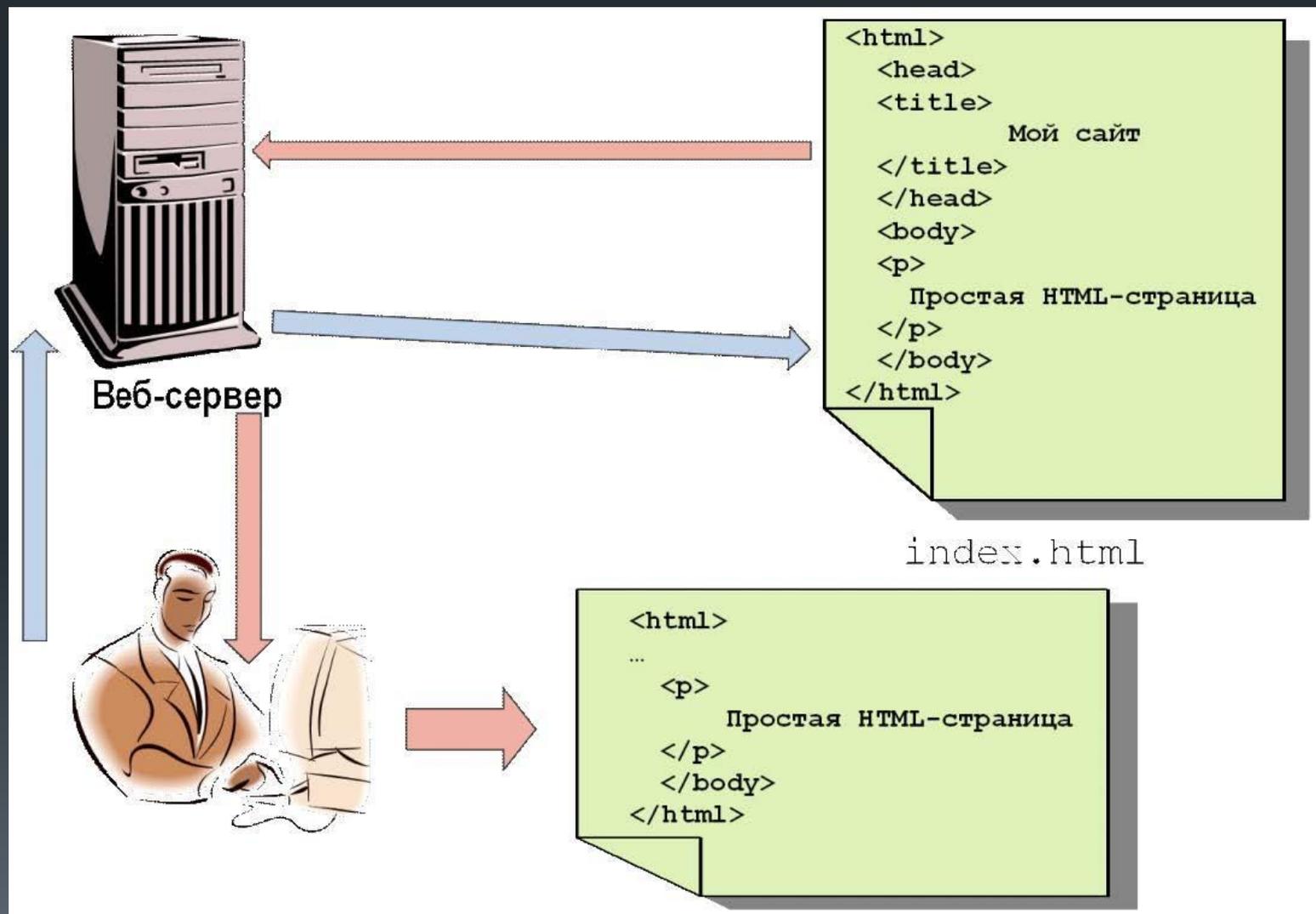
Темы для изучения

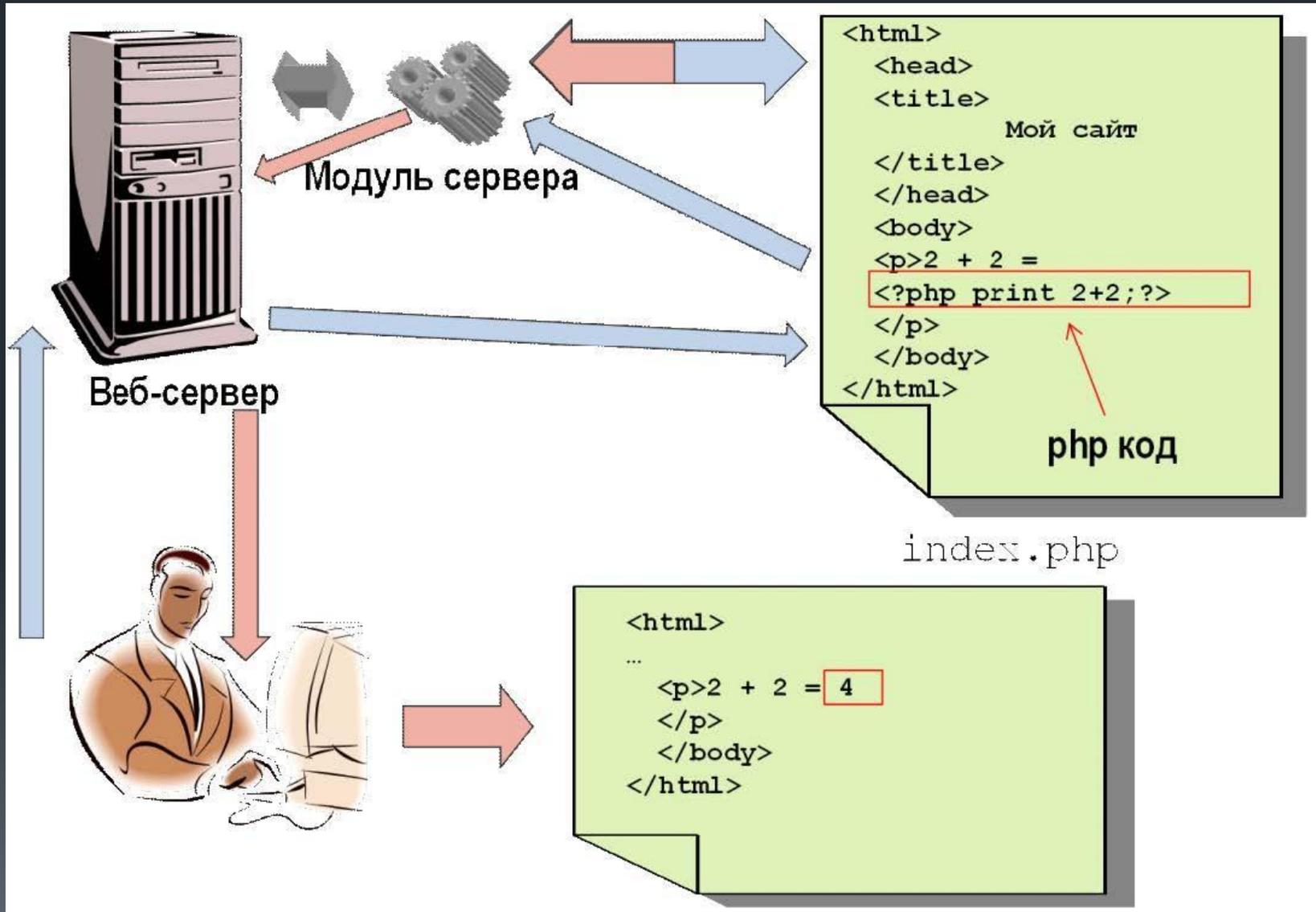
- Установка/настройка веб-сервера и PHP
- Основы PHP
- Циклы
- Пользовательские функции
- Изучаем HTTP: формы



Установка и настройка РНР и веб-сервера

Как это работает



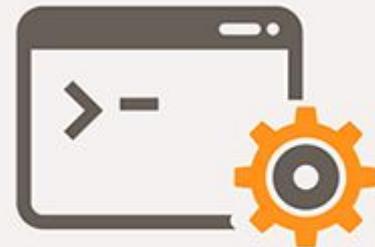


Сборка Open Server



> OPEN SERVER
ПЛАТФОРМА ДЛЯ ВЕБ-РАЗРАБОТЧИКОВ

<http://open-server.ru>



OPEN SERVER

Подготовка рабочего места

- Перейдите в директорию **C:/openserver/domains/**
- В этой директории создайте папку **wiki_your_name.local**
- Запустите сервер (программа находится в директории **C:/webserver/OSPanel/Open Server x64.exe**)
- В правом нижнем углу (рядом с часами) кликните правой кнопкой мыши по красному флажку и в появившемся меню выберите пункт “Запустить”
- Убедитесь, что ваш сайт работает, перейдя на страницу **wiki_your_name.local** в браузере

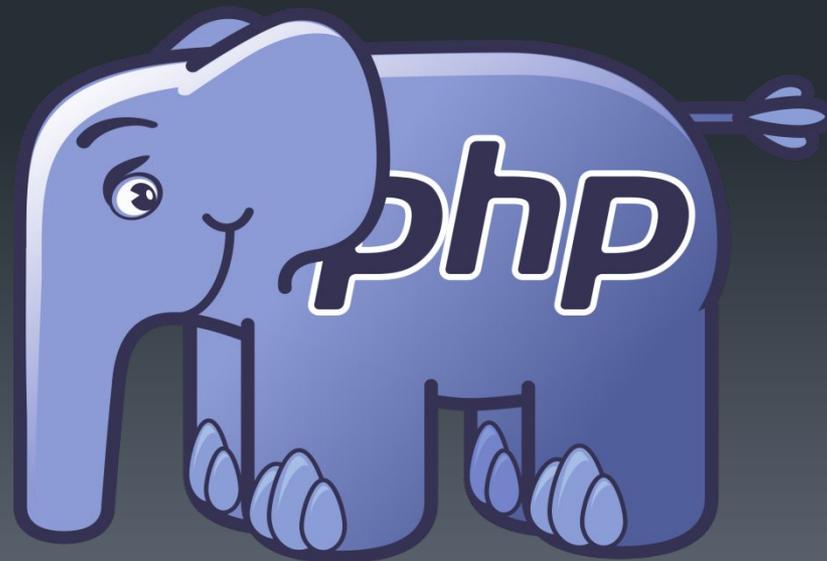
ОСНОВЫ PHP



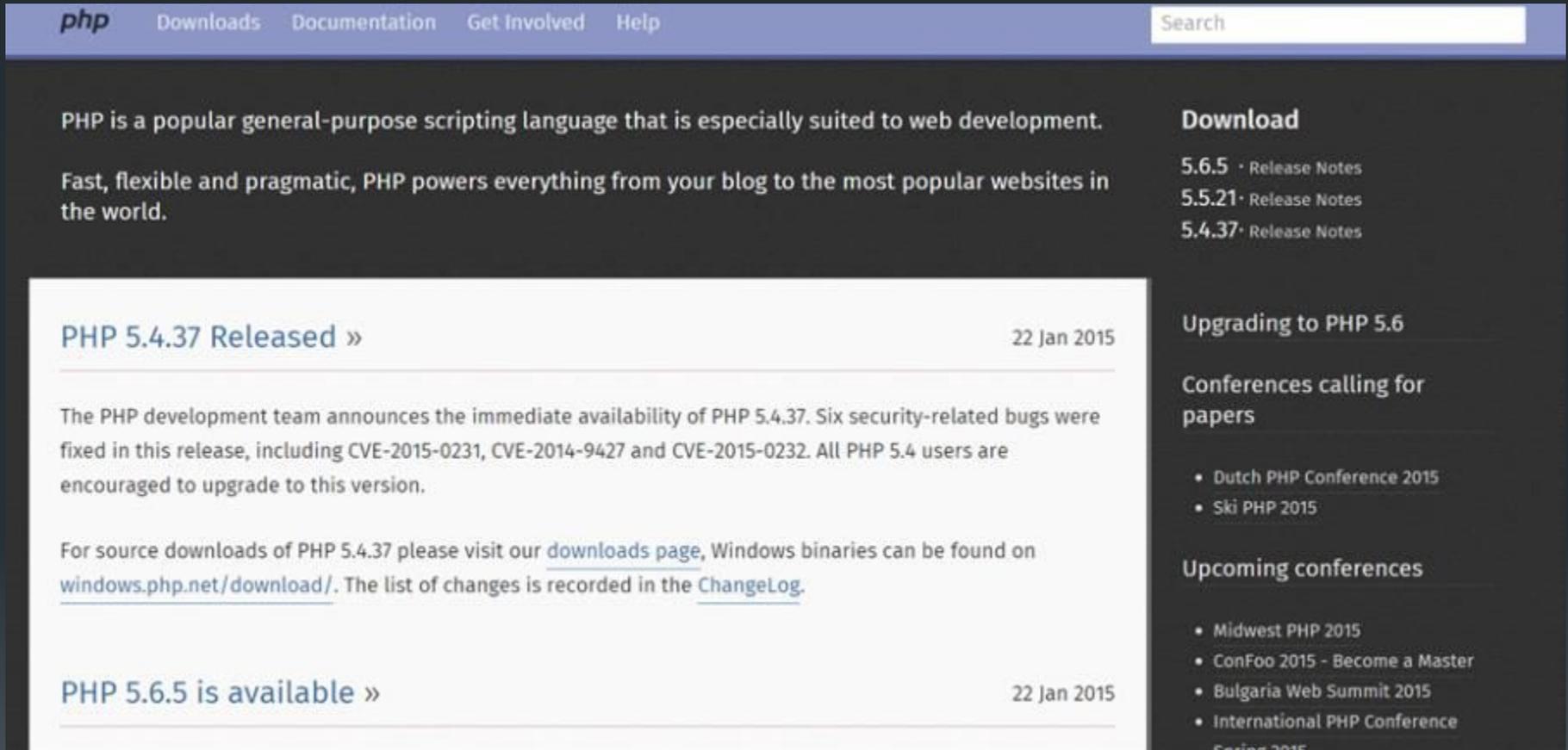
Что такое PHP?

PHP (Hypertext Preprocessor) - это распространенный **язык программирования** с открытым исходным кодом.

PHP специально сконструирован **для веб-разработок** и его код может **внедряться** непосредственно **в HTML**.



Где живет PHP?

A screenshot of the PHP website homepage. The top navigation bar includes links for 'php', 'Downloads', 'Documentation', 'Get Involved', and 'Help', along with a search box. The main content area features a introductory paragraph about PHP, followed by a news article titled 'PHP 5.4.37 Released' dated 22 Jan 2015. The article text describes the release, mentioning six security-related bugs fixed and providing links to the downloads page and changelog. A second news item 'PHP 5.6.5 is available' is partially visible at the bottom. On the right sidebar, there are sections for 'Download' (listing versions 5.6.5, 5.5.21, and 5.4.37), 'Upgrading to PHP 5.6', 'Conferences calling for papers' (listing Dutch PHP Conference 2015 and Ski PHP 2015), and 'Upcoming conferences' (listing Midwest PHP 2015, ConFoo 2015, Bulgaria Web Summit 2015, and International PHP Conference Spring 2015).

<http://php.net/>

Что входит в основы?

- Синтаксис
- Операторы
- Переменные
- Константы
- Типы данных
- Управляющие конструкции

Как PHP обрабатывает файл

<p>Это будет проигнорировано PHP и отображено браузером.</p>

```
<?php echo 'А это будет обработано.'; ?>
```

<p>Это тоже будет проигнорировано PHP и отображено браузером.</p>

```
<?php
```

```
echo "Hello world";
```

```
//... Еще код
```

```
echo "Последнее выражение";
```

```
//Скрипт заканчивается тут без закрывающего тега PHP
```

Комментарии

```
<?php
echo "Это тест"; // Это однострочный комментарий в стиле c++

/* Это многострочный комментарий
Еще одна строка комментария*/

echo "Это еще один тест";
echo "Последний тест"; # Это комментарий в стиле оболочки Unix
?>
```

Вывод данных

```
<?php
```

```
echo "Привет мир!";
```

```
print "Привет мир!";
```

```
//Некоторые предпочитают передачу нескольких аргументов
```

```
echo 'Эта', 'строка', 'была', 'создана', 'несколькими параметрами.';
```

```
//Вывод результата функции
```

```
echo strftime('%d-%B-%Y, %A');
```

Лабораторная работа 1. Вывод данных

Задача: Вывести “Hello, world!” и “Some text” на разных строчках.

Ход работы:

- Создайте и откройте в текстовом редакторе Notepad++ файл `index.php` в вашей папке на сервере.
- Используя языковую конструкцию `echo` выведите на экран сначала строку “Hello, world!”, а затем на следующей строчке выведите строку “Some text”
- Сохраните и проверьте работу по адресу **`wiki_your_name.local`**

У вас должно получиться так:

```
Hello, world!  
Some text
```

Переменные

Переменная — поименованная область памяти, адрес которой можно использовать для доступа к данным и изменять значение в ходе выполнения программы.

- Переменные в PHP начинаются со знака доллара (\$)
- Имя переменной должно начинаться с буквы или символа подчеркивания
- Последующие символы в имени переменной могут быть буквами, цифрами или символом подчеркивания в любом количестве
- Имя переменной чувствительно к регистру

Манипуляции с переменными

// Копирование переменных

```
$x = 10;
```

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

```
$y = 20;
```

```
echo $x; // 10
```

```
echo $y; // 20
```

// Ссылки

```
$x = 10;
```

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

```
$y = 20;
```

```
echo $x; // 20
```

```
echo $y; // 20
```

// Переменные переменных

```
$x = "name";
```

```
$$x = "Вася";
```

```
$y = 20;
```

```
echo "Привет, ", $name; // Привет, Вася
```

Использование

```
// Присвоим переменной $x значение 10
```

```
$x = 10;
```

```
// Добавим к значению в переменной $x значение 15
```

```
$x = $x + 15;
```

```
// Выведем значение переменной $x
```

```
echo $x;
```

```
// Удалим переменную $x
```

```
unset($x);
```

```
// Вывод значения переменной
```

```
$name = "Вася";
```

```
<h1>Привет <?php echo $name ?></h1>
```

```
<!-- Что и -->
```

```
<h1>Привет <?= $name ?></h1>
```

Упражнение

Чему будет равен X по окончании работы программы?

```
$x = 10;
```

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

```
$x = $y + 15;
```

```
$y = $x + 15;
```

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

```
echo $x;
```

Лабораторная работа 2.

Использование переменных

Задача: найти расстояние между двумя заданными точками.

Ход работы:

- Откройте в текстовом редакторе Notepad++ файл index.php в вашей папке на сервере.
- Создайте 4 переменные, которые будут отвечать за координаты точек.
- На основе этих переменных вычислите и сохраните в новую переменную расстояние между двумя точками с помощью следующей формулы:

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

- Выведите значение на экран.

Типы данных: строки

// Строка(string) -это набор символов, где символ - это то же самое, что и байт

```
echo 'это простая строка в апострофах';
```

```
echo "это простая строка в двойных кавычках";
```

```
echo "Это будет развернуто: \n в две строки";
```

```
echo 'А это не будет развернуто: \n в две строки';
```

```
$name = "John"; $age = 25;
```

```
echo "Значения переменных $name и $age подставятся в строку";
```

```
echo 'А здесь значения переменных $name и $age не разворачиваются';
```

HEREDOC и 'NOWDOC'

```
echo<<<HEREDOC
```

Значения переменных **\$name** и **\$age**
подставляются в строку, а сама строка развернется

```
HEREDOC;
```

```
echo<<<'NOWDOC'
```

Эквивалент строки в одинарных кавычках.
Переменные **\$name** и **\$age** не развернутся.
И строки тоже.

```
NOWDOC;
```

Экранирование переменных

```
$juice = "apple";
```

```
// Проблема
```

```
echo "He drank some $juice juice.";
```

```
echo "He drank some juice made of $juices.";
```

```
// Решение
```

```
echo "He drank some juice made of {$juice}s.";
```

```
echo "He drank some juice made of ${juice}s.";
```

Конкатенация строк

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

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

```
// Кстати, $d тоже содержит строку "Hello World!"
```

```
$d = "$a $b";
```

Лабораторная работа 3.

Работа со строками

Задача: преобразуйте данную строку в более короткую, используя при этом экранирование и конкатенацию

Ход работы:

- Дана строка: `echo 'Сегодня', $day, 'число, ', $mon, ' месяц, ', $year, ' год. ', "А я пью juice made of $juice", 's.';`
- Преобразуйте данную строку так, чтобы она была короче и хоть один раз используйте конкатенацию строк.

Пример вывода на экран:

```
Сегодня 22 число, декабрь месяц, 2018 год. А я пью juice made of apples.
```



Ветвление кода

Операторы сравнения

Пример	Название	Результат
$\$a == \b	Равно	TRUE если $\$a$ равно $\$b$ после преобразования типов.
$\$a === \b	Тождественно равно	TRUE если $\$a$ равно $\$b$ и имеет тот же тип.
$\$a != \b	Не равно	TRUE если $\$a$ не равно $\$b$ после преобразования типов.
$\$a !== \b	Тождественно не равно	TRUE если $\$a$ не равно $\$b$ или в случае, если они разных типов
$\$a < \b	Меньше	TRUE если $\$a$ строго меньше $\$b$.
$\$a > \b	Больше	TRUE если $\$a$ строго больше $\$b$.
$\$a <= \b	Меньше или равно	TRUE если $\$a$ меньше или равно $\$b$.
$\$a >= \b	Больше или равно	TRUE если $\$a$ больше или равно $\$b$.

Логические операторы

Пример	Название	Результат
$\$a$ and $\$b$	И	TRUE если и $\$a$, и $\$b$ TRUE .
$\$a$ or $\$b$	Или	TRUE если или $\$a$, или $\$b$ TRUE .
$\$a$ xor $\$b$	Исключающее или	TRUE если $\$a$, или $\$b$ TRUE , но не оба.
! $\$a$	Отрицание	TRUE если $\$a$ не TRUE .
$\$a$ && $\$b$	И	TRUE если и $\$a$, и $\$b$ TRUE .
$\$a$ $\$b$	Или	TRUE если или $\$a$, или $\$b$ TRUE .

Конструкция if-elseif-else

```
if ($a > $b) {  
    echo "a больше, чем b";  
}  
elseif ($a == $b) {  
    echo "a равно b";  
}  
else {  
    echo "a меньше, чем b";  
}
```

Конструкция switch

```
// Абсолютно правильный код
$day = 2;
if ($day == 1)
    echo "Понедельник";
elseif ($day == 2)
    echo "Вторник";
elseif ($day == 3)
    echo "Среда";
elseif ($day == 4)
    echo "Четверг";
elseif ($day == 5)
    echo "Пятница";
elseif ($day == 6)
    echo "Суббота";
elseif ($day == 7)
    echo "Воскресенье";
else
    echo "Неизвестный день";
```

```
$day = 2;
switch ($day) {
    case 1: echo "Понедельник"; break;
    case 2: echo "Вторник"; break;
    case 3: echo "Среда"; break;
    case 4: echo "Четверг"; break;
    case 5: echo "Пятница"; break;
    case 6: echo "Суббота"; break;
    case 7: echo "Воскресенье"; break;
    default: echo "Неизвестный день";
}
```

Лабораторная работа 4.

Ветвление кода

Задача: в зависимости от переменной h , значение которой равно от 0 до 23, выводить на экран “Доброе утро”, “Добрый день”, “Доброй ночи”

Ход работы:

- Присвойте переменной h любое значение от 0 до 23
- С помощью конструкции `if-elseif-else`, определите, какую фразу необходимо вывести: “Доброе утро”, “Добрый день”, “Доброй ночи”
- Протестируйте с разными значениями h
- Затем повторите решение с помощью конструкции `switch`

Типы данных: массив

// Создание пустого массива

```
$arr = []; // До PHP 5.4 $arr = array();
```

// Создание массива с элементами

```
$arr = ["John", "root", "1234"]; // До PHP 5.4 $arr = array("John", "root", "1234");
```

// Обращение к элементу массива

```
echo $arr[1]; // root
```

// Добавление элементов в массив

```
$arr[] = 25;
```

```
unset($arr[3]); // Удалили элемент массива с ключом 3
```

```
$arr[] = true;
```

// Сколько элементов в массиве?

```
echo count($arr); // 4
```

// Выводим удобочитаемую информацию о переменной

```
print_r($arr);
```

// Выводим полную информацию о переменной

```
var_dump($arr);
```

Ассоциативный массив

```
// Ассоциативный массив
$user = [
    "name"=>"John",
    "login"=>"root",
    "password"=>"1234",
    "age"=>25,
    true
];
echo $user["name"]; // John
echo $user[0]; // 1
```

Многомерный массив

```
// Многомерный массив
$users[0] = [
    "login" => "john",
    "pass" => "1234"
];
$users[1] = [
    "login" => "mike",
    "pass" => "5678"
];
echo $users[1]["login"]; // mike
echo $users[0]["login"]; //john
```

Лабораторная работа 4.

Использование многомерного массива

Задача: создать динамическое меню посредством использования многомерного массива.

Ход работы:

- Создайте многомерный массив, элементом которого является массив со следующими ключами: title, href, где title – название элемента меню, а href – ссылка на страницу.
- Вывести меню в виде столбца.

```
$leftMenu = [ //пример многомерного массива
    ['link' => 'Домой', 'href' => 'index.php'],
    ['link' => 'О нас', 'href' => 'about.php'],
    ['link' => 'Контакты', 'href' => 'contact.php'],
    ['link' => 'Таблица умножения', 'href' => 'table.php'],
    ['link' => 'Калькулятор', 'href' => 'calc.php']
];
```