

Игорь Борисов

PHP. Уровень 1.

Основы создания сайтов

<http://igor-borisov.ru>

Темы курса

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

Модуль 1

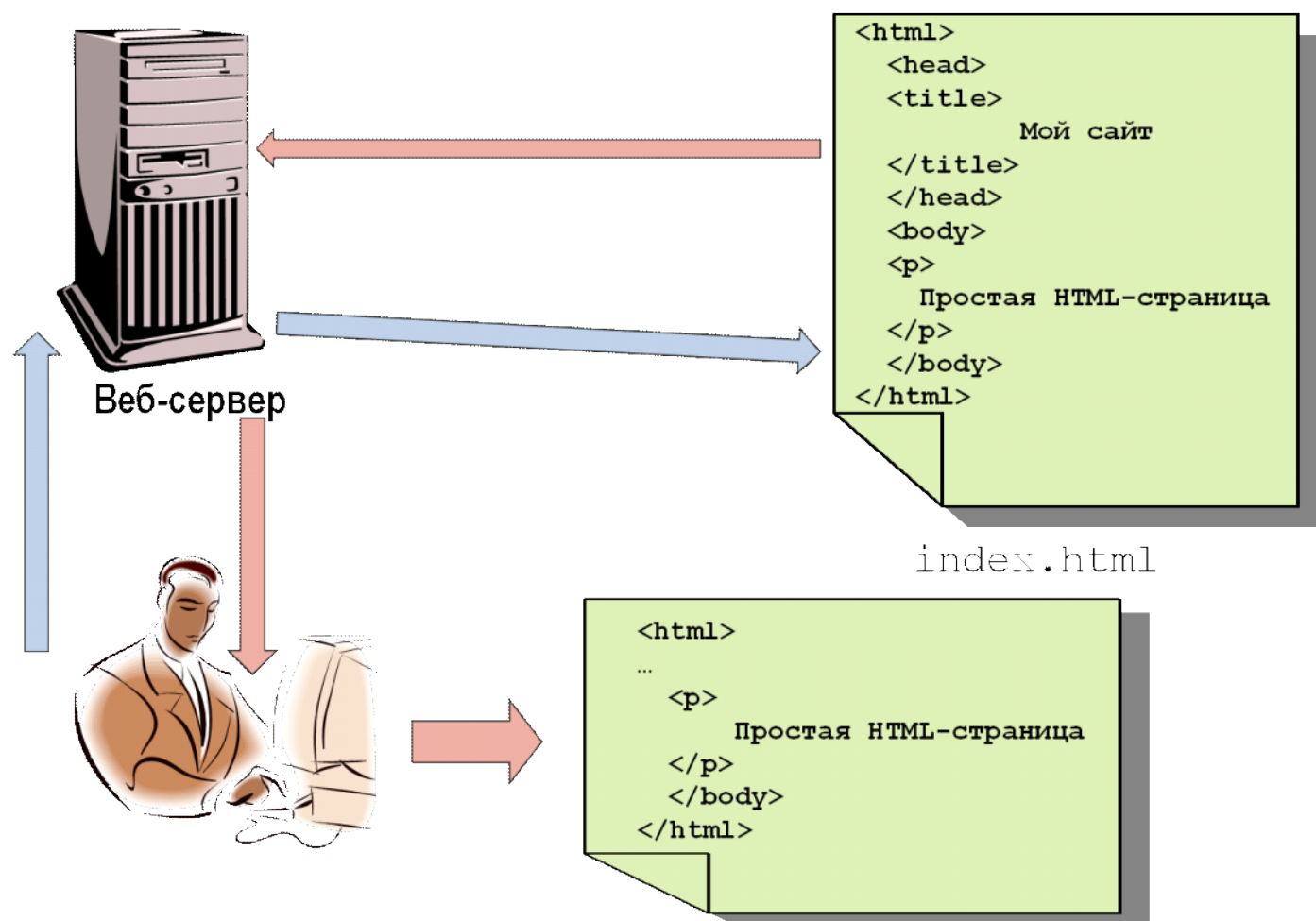
РНР. Уровень 1

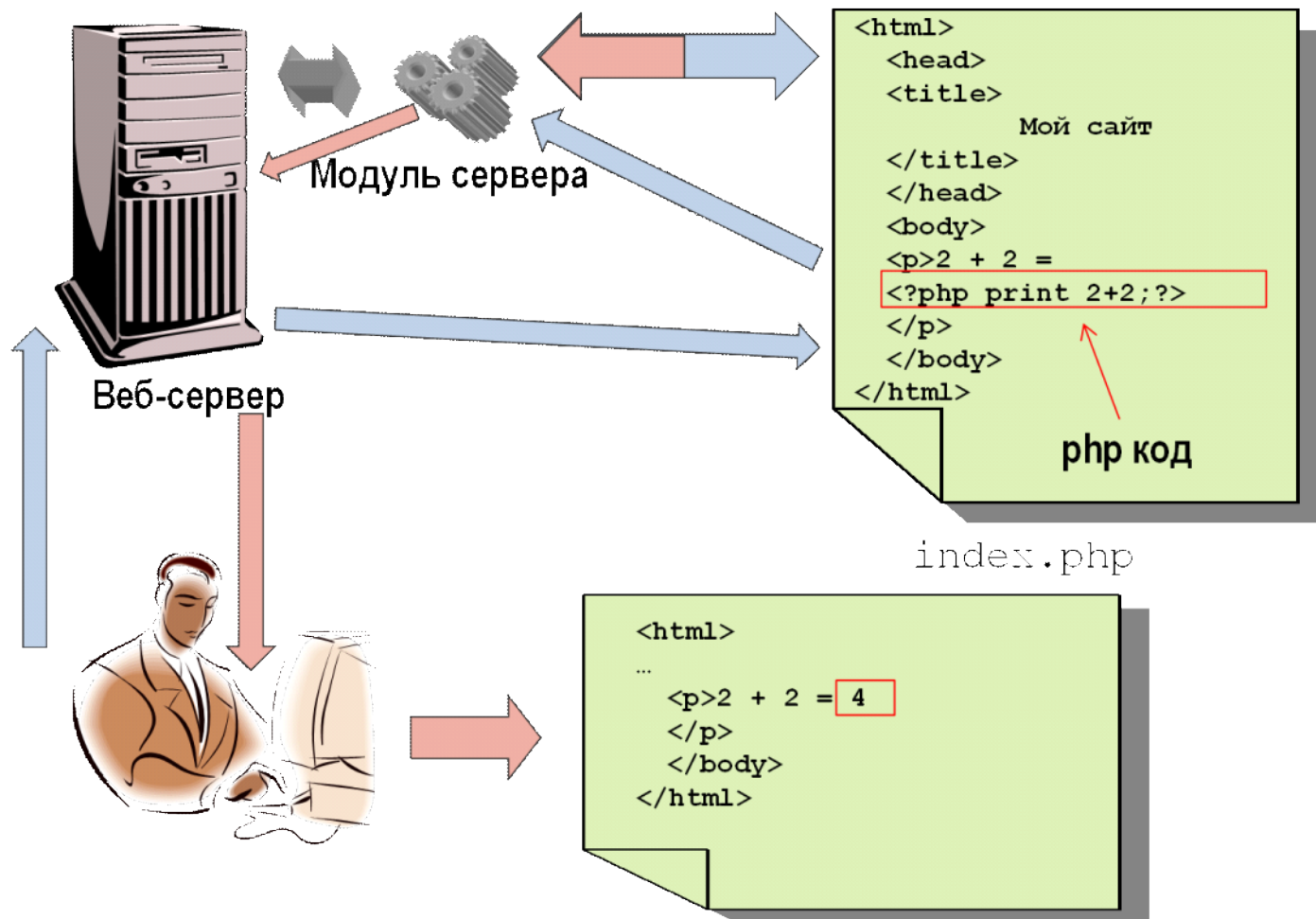
**Установка и настройка РНР
и веб-сервера**

Темы модуля

- Как это работает?
- Установка веб-сервера
- Настройка веб-сервера
- Установка PHP
- Настройка PHP
- Первый скрипт на PHP

Как это работает





Какой сервер использовать?

- Сервер Apache
 - <http://httpd.apache.org/>
- Сервер Microsoft IIS
 - <http://www.iis.net/>
- Сборка Денвер
 - <http://www.denwer.ru/>
- Сборка XAMPP
 - <http://www.apachefriends.org/ru/xampp.html>
- Сборка Wamp Server
 - <http://www.wampserver.com>
- Сборка EasyPHP
 - <http://www.easyphp.org/>

Сборка Open Server

<http://open-server.ru>



- ✓ Простое и понятное управление
- ✓ Уникальные возможности по настройке
- ✓ Open Server скачали уже **498 400** раз!

ГлавнаяНовостиСкачатьХостингМануалФорумСпасибо

Что это такое?



Open Server — это портативная серверная платформа и программная среда, созданная специально для веб-разработчиков с учётом их рекомендаций и пожеланий.

Программный комплекс имеет богатый набор серверного программного обеспечения, удобный, многофункциональный продуманный интерфейс, обладает мощными возможностями по администрированию и настройке компонентов. Платформа широко используется с целью разработки, отладки и тестирования веб-проектов, а так же для предоставления веб-сервисов в локальных сетях.

Хотя изначально программные продукты, входящие в состав комплекса, не разрабатывались специально для работы друг с другом, такая связка стала весьма популярной среди пользователей Windows, в первую очередь из-за того, что они получали бесплатный комплекс программ с надёжностью на уровне Linux серверов.

Удобство и простота управления безусловно не оставят вас равнодушными, за время своего существования Open Server зарекомендовал себя как **первоклассный и надёжный инструмент** необходимый каждому веб-мастеру.

Встречайте: Open Server!

Подготовительная работа

Подготовка рабочего места

Содержание подготовительной работы

Подготовка рабочего места

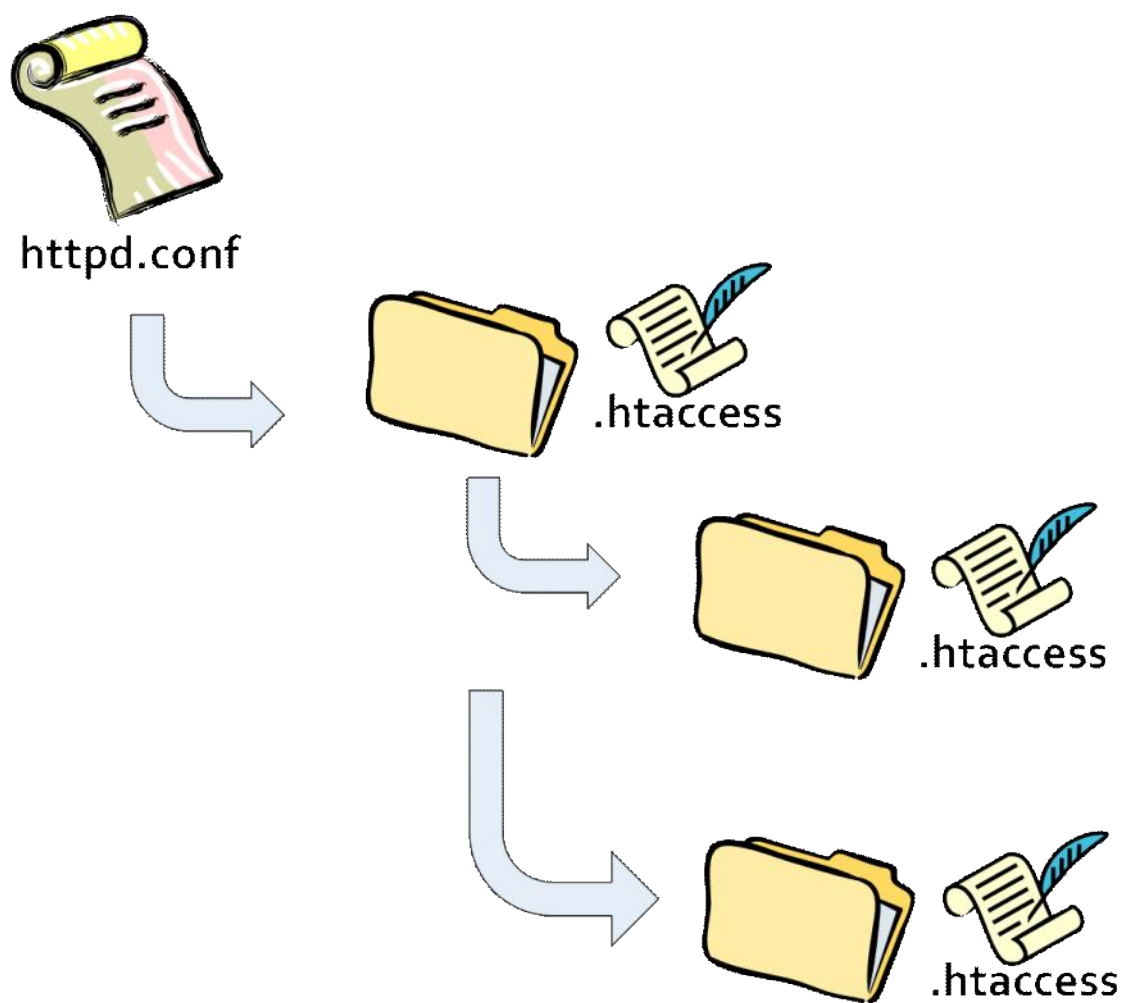
Задание 1: Создание виртуального хоста и запуск сервера

- Откройте проводник Windows
- Перейдите в директорию **C:\Пользователи\Общие\OpenServer\domains**
(Внимание! В некоторых ситуациях русскоязычному пути C:\Пользователи\Общие\ соответствует англоязычный путь C:\Users\Public\. Это одно и то же.)
- В этой директории создайте папку **mysite.local**
- Запустите сервер. Для этого нажмите **[Пуск -> Open Server]**
(На всякий случай, сама программа находится по пути C:\Пользователи\Общие\OpenServer\Open Server.exe)
- В правом нижнем углу (рядом с часами) кликните по иконке с красным флажком
- В открывшемся меню выберите первый пункт **Запустить**
- Дождитесь пока цвет иконки с флажком изменится с желтого на зеленый
- Если запуск закончился неудачей - флажок опять стал красным, то кликните по иконке, выберите последний пункт **Выход** и повторите последние 4 пункта

Задание 2: Копирование необходимых файлов

- Получите у преподавателя архив с файлами для работы на курсе
- Скопируйте файл в созданную в предыдущем упражнении директорию **C:\Пользователи\Общие\OpenServer\domains\mysite.local**
- Распакуйте файл в **текущую** директорию
- Запустите браузер и в адресной строке наберите: <http://mysite.local/>
- Убедитесь, что сайт работает

Файл .htaccess



Лабораторная работа 1.1

Создание файла .htaccess

Содержание лабораторной работы 1.1

Создание файла .htaccess

Упражнение 1: Создание файла конфигурации директории

- Откройте текстовый редактор
- В текстовом редакторе создайте новый файл и напишите следующий текст:

```
# Настройки сервера Apache
Options Indexes FollowSymLinks
DirectoryIndex index.php
```

- Сохраните файл в папке **С:\Пользователи\Общие\OpenServer\domains\mysite.local** под именем **.htaccess**
- Запустите браузер и в адресной строке наберите: <http://mysite.local/>
- Убедитесь, что нет ошибок и сайт работает

Где живёт PHP?

<http://php.net>

The screenshot shows the PHP website homepage. At the top is a navigation bar with links for 'php', 'Downloads', 'Documentation', 'Get Involved', and 'Help', along with a search box. The main content area features a dark header with the text: 'PHP is a popular general-purpose scripting language that is especially suited to web development. Fast, flexible and pragmatic, PHP powers everything from your blog to the most popular websites in the world.' Below this, there are two news items. The first is 'PHP 5.4.37 Released »' dated '22 Jan 2015', with a summary of security fixes and links to download pages and changelog. The second is 'PHP 5.6.5 is available »' also dated '22 Jan 2015'. On the right side, there are sections for 'Download' (listing 5.6.5, 5.5.21, and 5.4.37 with release notes), '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).

php Downloads Documentation Get Involved Help Search

PHP is a popular general-purpose scripting language that is especially suited to web development. Fast, flexible and pragmatic, PHP powers everything from your blog to the most popular websites in the world.

PHP 5.4.37 Released » 22 Jan 2015

The PHP development team announces the immediate availability of PHP 5.4.37. Six security-related bugs were fixed in this release, including CVE-2015-0231, CVE-2014-9427 and CVE-2015-0232. All PHP 5.4 users are encouraged to upgrade to this version.

For source downloads of PHP 5.4.37 please visit our [downloads page](#), Windows binaries can be found on windows.php.net/download/. The list of changes is recorded in the [ChangeLog](#).

PHP 5.6.5 is available » 22 Jan 2015

Download

- 5.6.5 · Release Notes
- 5.5.21 · Release Notes
- 5.4.37 · Release Notes

Upgrading to PHP 5.6

Conferences calling for papers

- Dutch PHP Conference 2015
- Ski PHP 2015

Upcoming conferences

- Midwest PHP 2015
- ConFoo 2015 - Become a Master
- Bulgaria Web Summit 2015
- International PHP Conference Spring 2015

Что внутри PHP?

- php5ts.dll
- ext
 - php_gd2.dll
 - php_mysqli.dll
 - php_mbstring.dll
 - ...
- php.exe
- php-cgi.exe
- php5apache2_4.dll
- php.ini
- ...

Лабораторная работа 1.2

Первый скрипт. Вывод системной информации

Содержание лабораторной работы 1.2

Первый скрипт. Вывод системной информации

Упражнение 1: Вывод системной информации

- Откройте текстовый редактор
- В текстовом редакторе создайте новый файл и напишите следующий текст:

```
<?php  
    phpinfo();  
?>
```

- Сохраните файл в папке **С:\Пользователи\Общие\OpenServer\domains\mysite.local** под именем **info.php**
- Запустите браузер и в адресной строке наберите:
<http://mysite.local/info.php>
- Вы должны увидеть специальную страницу с логотипом PHP

Что мы изучили?

- Как работает связка веб-сервера с PHP
- Как удобно развернуть удобную среду на локальной машине
- Получили представление о настройках веб-сервера
- Получили представление о том, что из себя представляет PHP
- Написали первый скрипт

Модуль 2

PHP. Уровень 1

Основы PHP

Темы модуля

- 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
```

PHP тэги

```
<?php
    // ...код
?>
```

```
<?
    // ...код
    // short_open_tag (php.ini)
?>
```

```
<script language="php">
    // ...код
</script>
```

```
<%
    // ...код
    // asp_tags (php.ini)
%>
```

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
?>
```

Особенности PHP

```
<?php
    инструкция 1;
    инструкция 2;
?>
```

```
<?php инструкция 1; инструкция 2; ?>
```

```
<?php
    инс
    трук
    ция 1;
?>
```


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 2.1

Вывод данных

Содержание лабораторной работы 2.1

Вывод данных

Упражнение 1: Вывод текущей даты

- Откройте в текстовом редакторе файл **index.php**
- В области основного контента **перед** строкой `<h3>Зачем мы ходим в школу?</h3>` напишите следующий текст:

```
<blockquote>  
    <?php echo strftime('Сегодня %d-%m-%Y'); ?>  
</blockquote>
```
- Сохраните файл **index.php**
- Запустите браузер и в адресной строке наберите: <http://mysite.local/>
- Убедитесь, что нет ошибок и результат выводится на страницу

Переменные

- 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

Использование переменных

```
// 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>

// Преобразование кодировок
echo iconv("windows-1251", "UTF-8", $name);
```

Константы

```
define("USER_NAME", "Вася");  
echo USER_NAME; // выводит "Вася"  
  
define("ONE_HUNDRED", 100);  
define("ONE_HUNDRED_TEN", ONE_HUNDRED_TEN + 10);  
  
const TWO_HUNDRED = 200; // PHP 5.3  
const TWO_HUNDRED_TEN = TWO_HUNDRED + 10; // PHP 5.6
```

Лабораторная работа 2.2

Использование переменных

Содержание лабораторной работы 2.2

Использование переменных

Упражнение 1: Вывод текущей даты используя переменные

- Откройте в текстовом редакторе файл **index.php**
- В самом начале файла введите следующий текст:

```
<?php
// Установка локали и выбор значений даты
setlocale(LC_ALL, "russian");
$day = strftime('%d');
$mon = strftime('%B');
$year = strftime('%Y');
?>
```
- Внутри тэгов **<blockquote></blockquote>** вместо текста:

```
echo strftime('Сегодня %d-%m-%Y');
```

введите следующий текст:

```
echo 'Сегодня ', $day, ' число, ', $mon, ' месяц, ',
$year, ' год.';
```
- Сохраните файл **index.php** и посмотрите результат в браузере
- Внизу файла в блоке **<!-- Нижняя часть страницы -->** вместо **2015** выведете значение необходимой переменной
- Сохраните файл **index.php**
- Посмотрите результат в браузере

Ошибки - наше всё

- Уровни ошибок
 - E_PARSE
 - E_ERROR
 - E_WARNING
 - E_NOTICE
 - E_DEPRECATED
- Директивы PHP.INI
 - display_errors = on
 - error_reporting = E_ALL & ~E_NOTICE

```
// Включаем вывод всех ошибок  
error_reporting(E_ALL);
```

```
// Отключаем вывод всех ошибок  
error_reporting(0);
```

```
// Включаем определённые уровни ошибок  
error_reporting(E_ERROR | E_WARNING);  
error_reporting(E_ALL & ~E_DEPRECATED);
```

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
```

Типы данных: числа

// Целые числа (integer)

```
$int = 1234; // десятичное число
$int = -123; // отрицательное число
$int = 0123; // восьмеричное число (эквивалентно 83 в
десятичной системе)
$int = 0x1A; // шестнадцатеричное число (эквивалентно 26 в
десятичной системе)
$int = 0b11111111; // двоичное число (эквивалентно 255 в
десятичной системе)
```

// Числа с плавающей точкой (float, также известны как double
и real)

```
$float = 1.234;
$float = 1.2e3;
$float = 7E-10;
```

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';
```

```
echo <<<HEREDOC
```

```
Значения переменных $name и $age
```

```
подставятся в строку, а сама строка развернётся
```

```
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.";
```

// Доступ к символу в строке

```
$str = 'This is a test.';
```

// Получение первого символа в строке

```
$first = $str{0};
```

// Получение третьего символа в строке

```
$third = $str{2};
```

// Получение последнего символа в строке

```
$last = $str{ strlen($str)-1 };
```

// Изменение последнего символа в строке

```
$str{ strlen($str)-1 } = '!';
```


Лабораторная работа 2.3

Использование строк в двойных кавычках

Содержание лабораторной работы 2.3

Использование строк в двойных кавычках

Упражнение 1: Вывод текущей даты используя подстановку значений переменных в двойных кавычках

- Откройте в текстовом редакторе файл **index.php**
- Переделайте строку:
`echo 'Сегодня ', $day, ' число, ', $mon, ' месяц, ',
$year, ' год.';`
в строку:
`echo "Сегодня $day число, $mon месяц, $year год.";`
- Сохраните файл **index.php**
- Посмотрите результат в браузере

Арифметические операторы

Пример	Название	Результат
$-$a$	Отрицание	Смена знака $$a$
$$a + b	Сложение	Сумма $$a$ и $$b$
$$a - b	Вычитание	Разность $$a$ и $$b$
$$a * b	Умножение	Произведение $$a$ и $$b$
$$a / b	Деление	Частное от деления $$a$ на $$b$
$$a \% b	Деление по модулю	Целочисленный остаток от деления $$a$ на $$b$

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";
```

Типы данных: boolean и NULL

```
$x = true; // присвоить $x значение TRUE
$y = false; // присвоить $y значение TRUE

$z = null; // присвоить $z значение NULL

unset($x);
echo gettype($x); // NULL
```

Манипуляции с типами

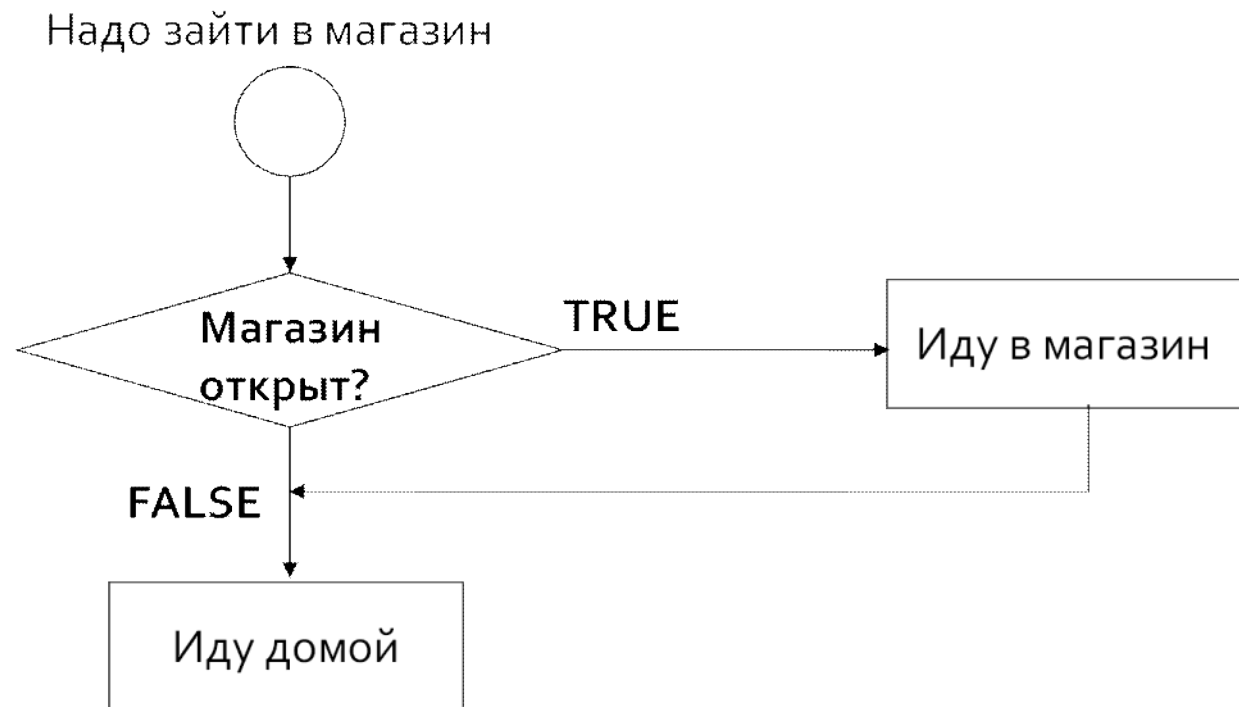
```
// Проверка типов
$str = "John";
$int = 10;
$bool = true;

echo gettype($str); // "string"
echo gettype($int); // "integer"
echo gettype($bool); // "boolean"
echo gettype($x); // "NULL"

echo is_string($str); // "1"
echo is_integer($int); // "1"
echo is_boolean($bool); // "1"
echo is_null($x); // ""

// Приведение типов
$int_from_str = (int)$str;
$str_from_int = (string)$int;
$bool_from_int = (bool)$int;
```

Управление кодом: if



```
$shop = true;
```

```
// Одна инструкция
```

```
if ($shop)
    echo "Иду в магазин";
echo "Иду домой";
```

```
// Группа инструкций
```

```
if ($shop) {
    echo "Иду в магазин";
    echo "Покупаю хлеб";
}
echo "Иду домой";
```

Таблица сравнения типов

Выражение	gettype()	empty()	isset()	boolean : <i>if(\$x)</i>
<code>\$x = "";</code>	string	TRUE	TRUE	FALSE
<code>\$x = null;</code>	NULL	TRUE	FALSE	FALSE
<code>\$x</code> неопределена	NULL	TRUE	FALSE	FALSE
<code>\$x = false;</code>	boolean	TRUE	TRUE	FALSE
<code>\$x = true;</code>	boolean	FALSE	TRUE	TRUE
<code>\$x = 1;</code>	integer	FALSE	TRUE	TRUE
<code>\$x = 42;</code>	integer	FALSE	TRUE	TRUE
<code>\$x = 0;</code>	integer	TRUE	TRUE	FALSE
<code>\$x = -1;</code>	integer	FALSE	TRUE	TRUE
<code>\$x = "1";</code>	string	FALSE	TRUE	TRUE
<code>\$x = "0";</code>	string	TRUE	TRUE	FALSE
<code>\$x = "-1";</code>	string	FALSE	TRUE	TRUE
<code>\$x = "php";</code>	string	FALSE	TRUE	TRUE
<code>\$x = "true";</code>	string	FALSE	TRUE	TRUE
<code>\$x = "false";</code>	string	FALSE	TRUE	TRUE

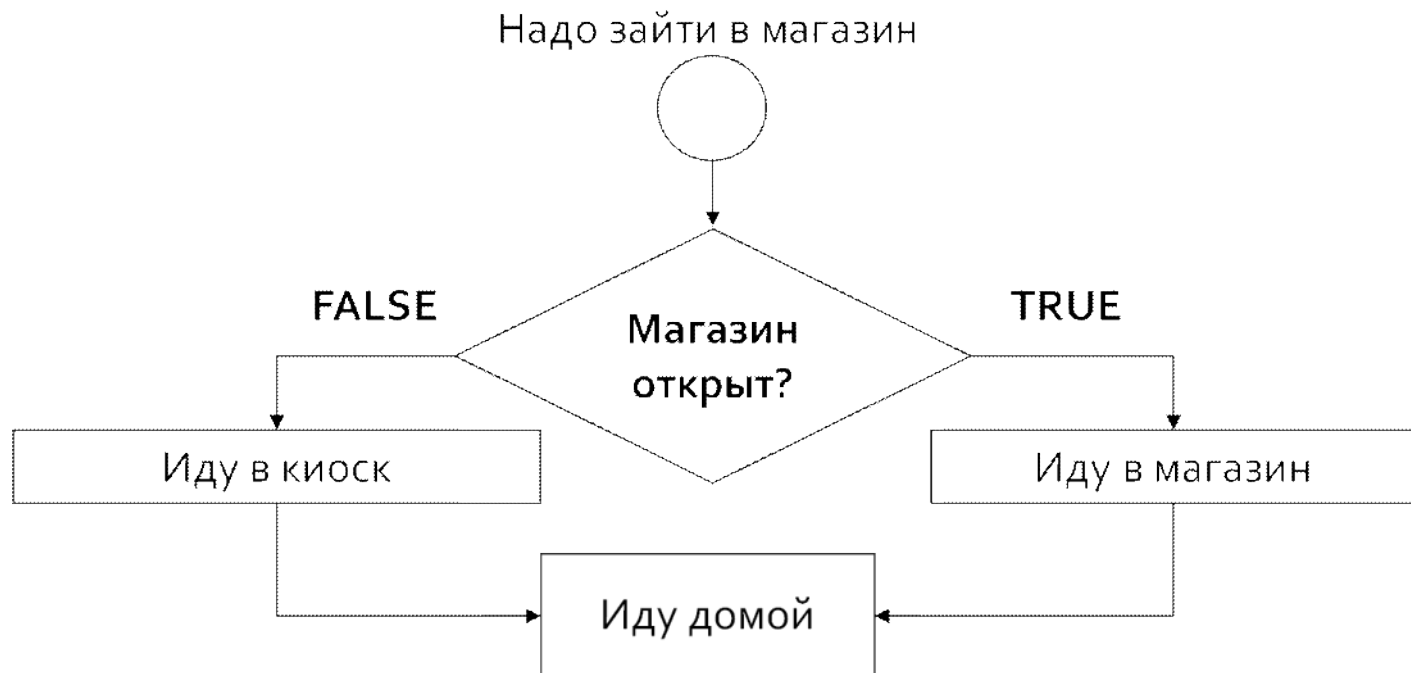
Операторы сравнения

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

Логические операторы

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

Управление кодом: else



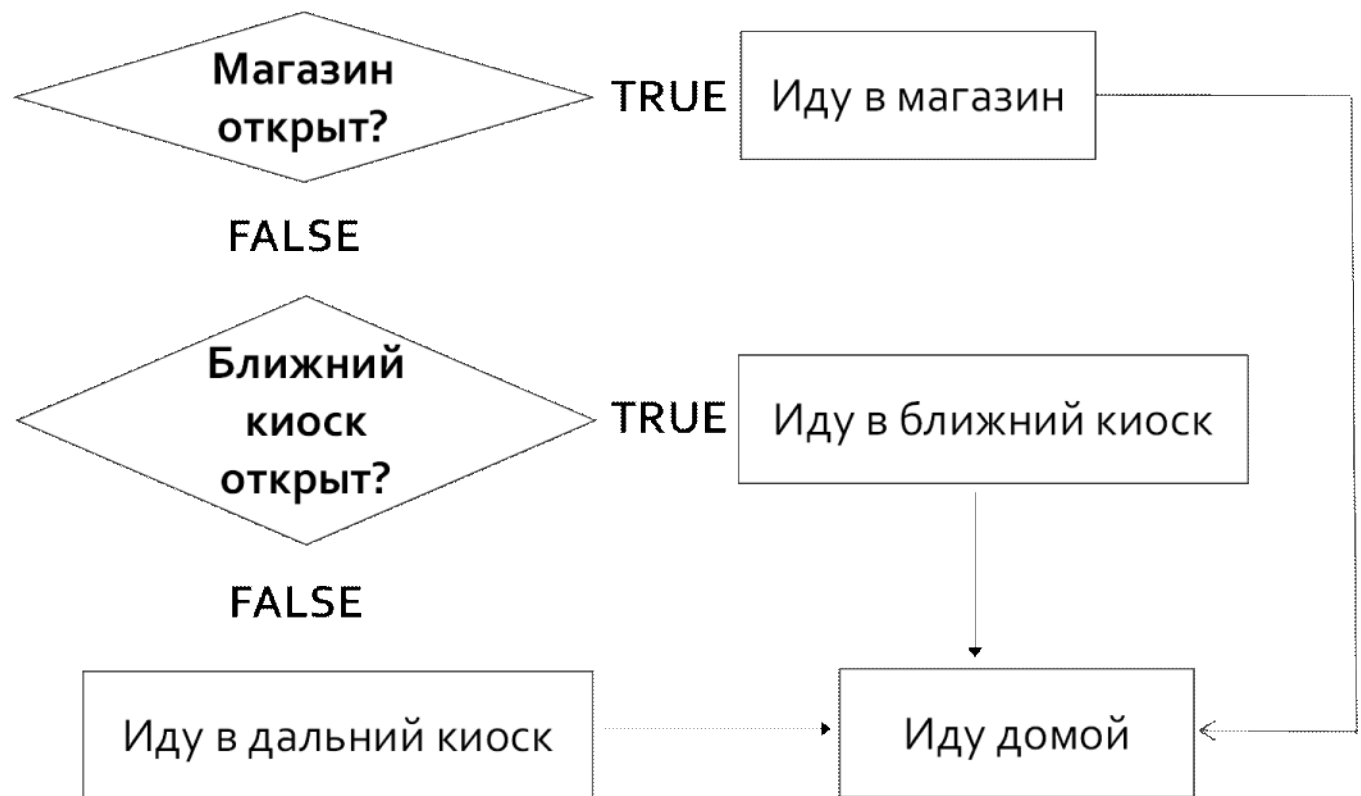
```
$shop = "open";  
// Одна инструкция  
if ($shop == "open")  
    echo "Иду в магазин";  
else  
    echo "Иду в киоск";  
echo "Иду домой";
```

```
$shop = true; $money = true;  
// Группа инструкций  
if ($shop and $money) {  
    echo "Иду в магазин";  
    echo "Покупаю хлеб";  
}else{  
    echo "Иду домой";  
    echo "Туплю в телевизор";  
}
```

```
// Тернарный оператор  
if ($shop == "open")  
    echo "Иду в магазин";  
else  
    echo "Иду в киоск";  
  
echo ($shop) ? "Иду в магазин" : "Иду в киоск";
```


Управление кодом: elseif

Надо зайти в магазин



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

Лабораторная работа 2.4

Использование управляющих конструкций if-elseif-else

Содержание лабораторной работы 2.4

Использование управляющих конструкций if-elseif-else

Упражнение 1: Вывод приветствия в зависимости от времени суток

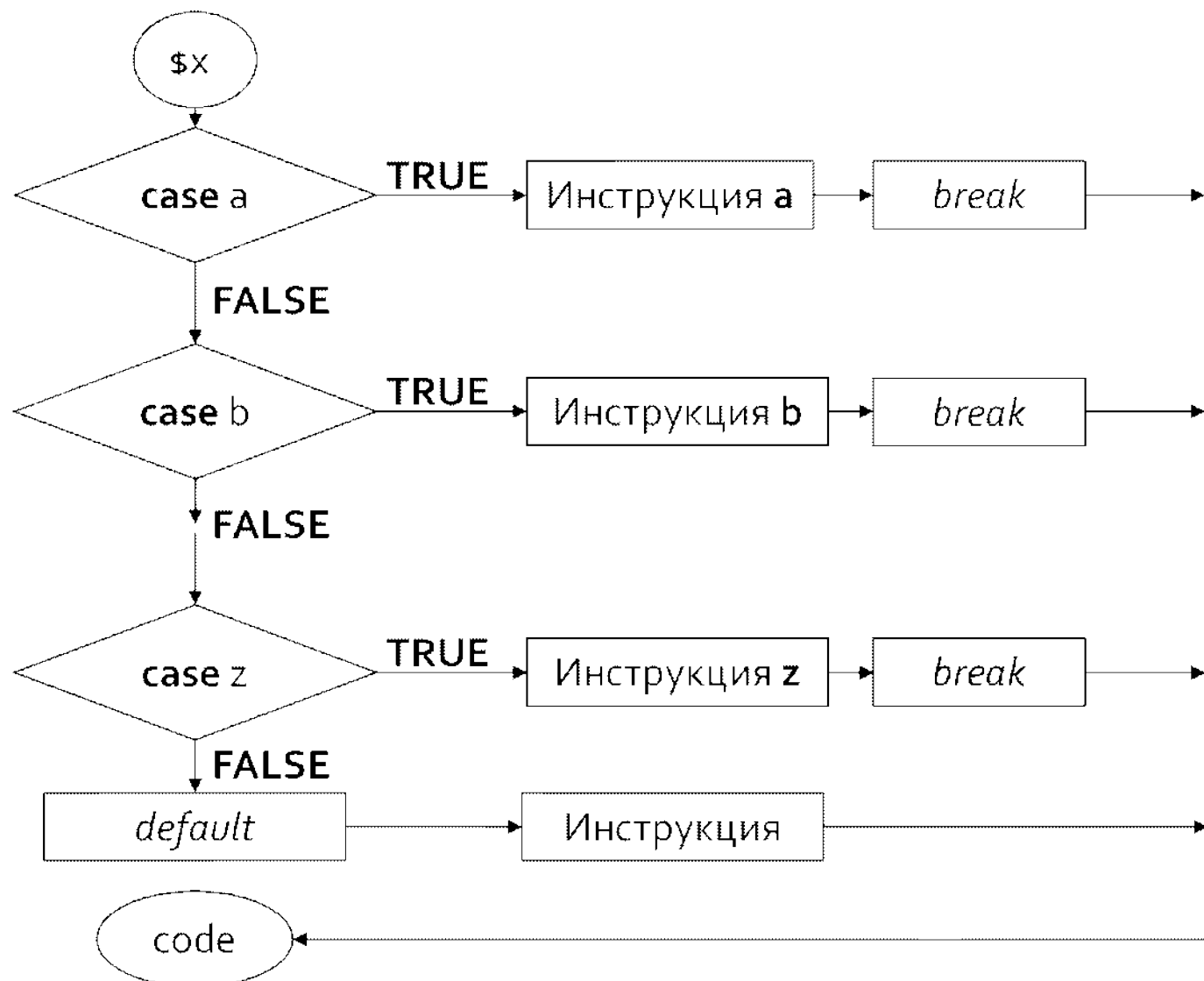
- Откройте в текстовом редакторе файл **index.php**
- В начале файла в php-блоке напишите:

```
/*  
 * Получаем текущий час в виде строки от 00 до 23  
 * и приводим строку к целому числу от 0 до 23  
 */  
$hour = (int) strftime('%H');  
$welcome = ''; // Инициализируем переменную для  
приветствия
```
- Используя управляющую конструкцию if – elseif – else присвойте переменной **\$welcome** значение, исходя из следующих условий если число в переменной **\$hour** попадает в диапазон:
 - от **0** до **6** - **Доброй ночи**
 - от **6** (включительно) до **12** - **Доброе утро**
 - от **12** (включительно) до **18** - **Добрый день**
 - от **18** (включительно) до **23** - **Добрый вечер**
- Если число в переменной **\$hour** не попадает ни в один из вышеперечисленных диапазонов, то присвойте переменной **\$welcome** значение **Доброй ночи**
- Между тэгами **<h1></h1>** вместо строки **Добро пожаловать на наш сайт!** напишите: **<?= \$welcome ?>**, **Гость!**
- Сохраните файл **index.php**
- Посмотрите результат в браузере

Управление кодом: 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";
```



```
$i = 1;
switch ($i) {
    case 0:
        echo "Результат: 0";
    case 1:
        echo "Результат: 1";
    case 2:
        echo "Результат: 2";
    case 3:
        echo "Результат: 3";
    case 4:
        echo "Результат: 4";
}

switch ($i) {
    case 0:
        echo "Результат: 0"; break;
    case 1:
        echo "Результат: 1"; break;
    case 2:
        echo "Результат: 2"; break;
    case 3:
        echo "Результат: 3"; break;
    case 4:
        echo "Результат: 4"; break;
}

$i = 20;
switch ($i) {
    case 0:
        echo "Результат: 0"; break;
    case 1:
        echo "Результат: 1"; break;
    case 2:
        echo "Результат: 2"; break;
    case 3:
        echo "Результат: 3"; break;
    case 4:
        echo "Результат: 4"; break;
    default:
        echo "Результат: много";
}
```

Лабораторная работа 2.5

Использование управляющей конструкции switch

Содержание лабораторной работы 2.5

Использование управляющей конструкции switch

Упражнение 1: Вывод значения директивы PHP `post_max_size`

- **Задача:** вывести значение директивы `php.ini post_max_size` в байтах
- Откройте в текстовом редакторе файл **`contact.php`**
- В самом начале блока **`<!-- Область основного контента -->`** напишите:
`<?php`
`?>`
- Задайте два вопроса, необходимые для решения задачи
- Создайте переменную **`$size`**, которая будет содержать текущее значение директивы **`post_max_size`**
- Получите данные, о величине в которой представлено значение (т.е. килобайты, мегабайты...)
- Используя управляющую конструкцию **`switch`** вычислите результат и сохраните его в переменную **`$size`**
- После закрывающего тэга веб-формы **`</form>`** напишите:
`<p>Максимальный размер отправляемых данных <?= $size ?>`
`байт.</p>`
- Сохраните файл **`contact.php`**
- Посмотрите результат в браузере

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;
$arr[] = true;

// Сколько элементов в массиве?
echo count($arr); // 5

// Выводим удобочитаемую информацию о переменной
print_r($arr);
// Выводим полную информацию о переменной
var_dump($arr);

// Ещё раз об индексации элементов массива
// Если массив не существует, он будет создан
$arr[] = 1; // Массив с одним элементом с ключом 0
$arr[5] = 2; // Добавили второй элемент с ключом 5
$arr[] = 3; // Добавили третий элемент с ключом 6
unset($arr); // Удалили массив
$arr = [12=>1, 5=>2]; // Создали массив из двух элементов с ключами 12 и 5
$arr[] = 3; // Добавили третий элемент с ключом 13
unset($arr[12]); // Удалили элемент массива с ключом 12

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

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



```
        "login" => "mike",  
        "pass" => "5678"  
    ];  
    echo $users[1]["login"]; // mike
```

Лабораторная работа 2.6

Использование многомерного массива

Содержание лабораторной работы 2.6

Использование многомерного массива

Упражнение 1: Создание динамического меню

- Откройте в текстовом редакторе файл **index.php**
- В начале блока **<!-- Меню -->** напишите:

```
<?php
$leftMenu = [
    ['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']
];
?>
```
- В значениях атрибута **href** тэгов **<a>** вместо текущих значений выведите значения элементов массива **\$leftMenu** по следующему образцу:

```
<li><a href='<?> $leftMenu[0]['href']?'>'><?> $leftMenu[0]['link']?'></a></li>
<li><a href='<?> $leftMenu[1]['href']?'>'><?> $leftMenu[1]['link']?'></a></li>
...
```
- Сохраните файл **index.php**
- Посмотрите результат в браузере

Что мы изучили?

- Познакомились с особенностями синтаксиса PHP
- Уяснили понятие переменных и констант
- Познакомились с типами данных: integer, float, string, boolean, null, array
- Научились манипулировать типами данных с помощью различных операторов
- Изучили несколько полезных встроенных функций
- Научились управлять кодом с помощью конструкций: if-elseif-else и switch

Модуль 3

РНР. Уровень 1

Циклы

Темы модуля

- Операторы инкремента и декремента
- Цикл for
- Цикл while
- Цикл do-while
- Управление циклами
- Цикл foreach

Операторы инкремента и декремента

// Постфиксный инкремент

```
$x = 1;
```

```
echo "Должно быть 1: " . $x++;
```

```
echo "Должно быть 2: " . $x;
```

// Что и

```
echo "Должно быть 1: $x"; $x += 1;
```

// Префиксный инкремент

```
$x = 1;
```

```
echo "Должно быть 2: " . ++$x;
```

```
echo "Должно быть 2: " . $x;
```

// Что и

```
$x += 1; echo "Должно быть 2: $x";
```

Цикл for

```
for (Часть А; Часть В; Часть С) {  
    // Тело цикла  
}
```

```
for ($i = 1; $i <= 10; $i++) {  
    echo $i;  
}
```

```
for ($i = 1; $i <= 10; print $i++);
```


Лабораторная работа 3.1

Использование цикла for

Содержание лабораторной работы 3.1

Использование цикла `for`

Упражнение 1: Вывод нечетных чисел из заданного диапазона

- Откройте проводник Windows
- Перейдите в папку **C:\Users\Public\OpenServer\domains\mysite.local\demo**
- Создайте в текущей папке файл **for.php**
- Откройте созданный файл **for.php** в текстовом редакторе
- Используя цикл **for** выведите в столбик **нечётные** числа от **1** до **50**
- Сохраните файл **for.php**
- Посмотрите результат в браузере

Цикл while

```
while (Условие) {  
    // Тело цикла  
}
```

```
$i = 1;  
while ($i <= 10) {  
    echo $i++;  
}
```

```
// Бесконечный цикл  
while (true);
```

Лабораторная работа 3.2

Использование цикла `while`

Содержание лабораторной работы 3.2

Использование цикла `while`

Упражнение 1: Вывод строки посимвольно

- Откройте проводник Windows
- Перейдите в папку **C:\Users\Public\OpenServer\domains\mysite.local\demo**
- Создайте в текущей папке файл **while.php**
- Откройте созданный файл **while.php** в текстовом редакторе
- Создайте переменную **\$var** и присвойте ей строковое значение **HELLO**
- Используя цикл **while** выведите значение переменной **\$var** в столбик так, чтобы на выходе в браузере получилось:
Н
Е
Л
Л
О
- Сохраните файл **while.php**
- Посмотрите результат в браузере

Цикл do-while

```
$i = 100;  
  
do {  
    echo $i++;  
} while ($i <= 10);
```

Управление циклами

// Прерывание цикла

```
$i = 1;
while ($i <= 10) {
    echo $i++;
    if($i == 5)
        break;
}
```

// Продолжение цикла

```
$i = 0;
while ($i < 9) {
    $i++;
    if($i == 5)
        continue;
    echo $i;
}
```

// Управление вложенными циклами

```
$i = 1; $j = 1;
while ($j <= 10) {
    while ($i <= 10) {
        echo $i++;
        if($i == 5)
            break 2;
    }
    $j++;
}
```

Лабораторная работа 3.3

Создание динамической таблицы умножения

Содержание лабораторной работы 3.3

Создание динамической таблицы умножения

Упражнение 1: Создание HTML-таблицы

- В текстовом редакторе откройте файл **table.php**
- В начале файла создайте php-блоке, в котором создайте две целочисленные переменные **\$cols** и **\$rows**
- Присвойте созданным переменным произвольные значения в диапазоне от **1** до **10**
- В блоке **<!-- Таблица -->** удалите весь html-код и напишите:
**<?php
?>**
- В текущем php-блоке используя циклы **for** отрисуйте таблицу умножения в виде HTML-таблицы на следующих условиях:
 - Число **столбцов** должно быть равно значению переменной **\$cols**
 - Число **строк** должно быть равно значению переменной **\$rows**
 - Ячейки на пересечении столбцов и строк должны содержать значения, являющиеся **произведением** порядковых номеров столбца и строки
- Сохраните файл **table.php**
- Посмотрите результат в браузере

Упражнение 2: Приводим таблицу к товарному виду

- Отрисуйте значения в ячейках первой строки и первого столбца полужирным шрифтом и выровняйте их по центру ячейки
- Сделайте фоновый цвет ячеек первой строки и первого столбца отличным от фонового цвета таблицы
- Сохраните файл **table.php**
- Посмотрите результат в браузере

Цикл foreach

```
$arr = ['a'=>'one', 'b'=>'two', 'c'=>'three'];
```

```
foreach ($arr as $val) {  
    echo "$val\n";  
}  
/* one  
   two  
   three  
*/
```

```
foreach ($arr as $key => $val) {  
    echo "$key : $val\n";  
}  
/* a : one  
   b : two  
   c : three  
*/
```

```
$nums = [1, 2, 3, 4, 5];  
foreach ($arr as $val) {  
    $val *= 10;  
}  
// Массив не изменился
```

```
foreach ($arr as &$amp;val) {  
    $val *= 10;  
}  
// Теперь массив: [10, 20, 30, 40, 50]
```

Лабораторная работа 3.4

Создание динамического навигационного меню

Содержание лабораторной работы 3.4

Создание динамического навигационного меню

Упражнение 1: Вывод меню с использованием цикла

- В текстовом редакторе откройте файл **index.php**
- Перенесите в php-блок в начале файла код инициализации массива из блока **<!-- Меню -->**, то есть:

```
$leftMenu = [  
    [ 'link'=>'Домой', 'href'=>'index.php' ];  
    ...  
];
```


добавив комментарий **// Инициализация массива**
- В блоке **<!-- Меню -->** удалите всё **html-содержимое** (от **** до **** включительно)
- В php-блоке блока **<!-- Меню -->** отрисуйте вертикальное меню с помощью цикла **foreach**, передав ему в качестве аргумента массив **\$leftMenu**.
Обратите **внимание**, что массив - многомерный.
- Сохраните файл **index.php**
- Посмотрите результат в браузере

Что мы изучили?

- Познакомились с операторами инкремента и декремента
- Научились использовать основные циклы for и while
- Научились управлять циклами
- Научились итерированию массивов

Модуль 4

РНР. Уровень 1

Пользовательские функции

Темы модуля

- Декларация функции
- Вызов функции
- Варианты объявления функции
- Аргументы функции
- Области видимости переменных
- Статические переменные
- Возвращаемые значения
- Функция с переменным числом аргументов
- Уточнение типа аргумента функции

Декларация и вызов функции

// Декларация функции

```
<?php
function sayHello()
{
    echo "<h1>Hello, world!</h1>";
}
```

// Вызов функции

```
sayHello();
?>
```

// Функции, зависящие от условий

```
<?php
```

```
$makefoo = true;
```

```
/* Мы не можем вызвать функцию foo() в этом месте,
   поскольку она еще не определена, но мы можем
   обратиться к bar() */
```

```
bar();
```

```
if ($makefoo) {
    function foo()
    {
        echo "Я не существую до тех пор, пока выполнение программы
меня не достигнет.\n";
    }
}
```

```
/* Теперь мы благополучно можем вызывать foo(),
   поскольку $makefoo была интерпретирована как true */
```

```
if ($makefoo) foo();
```

```
function bar()
{
    echo "Я существую сразу с начала старта программы.\n";
}
```

```
?>
```

// Вложенные функции

```
<?php
```

```
function foo()
```



```
{
  function bar()
  {
    echo "Я не существую пока не будет вызвана foo().\n";
  }
}

/* Мы пока не можем обратиться к bar(),
   поскольку она еще не определена. */

foo();

/* Теперь мы можем вызвать функцию bar(),
   обработка foo() сделала ее доступной. */

bar();
```

Аргументы функции

```
function sayHello($name)
{
    echo "<h1>Hello, $name!</h1>";
}

// Передаём литерал
sayHello("John"); // Hello, John!

// Передаём значение переменной
$n = "Mike";
sayHello($n); // Hello, Mike!

// Обращение к функции через переменную
$func = "sayHello";
$func("Guest"); // Hello, Guest!

// Аргументы по-умолчанию
function sayHello($name="Guest")
{
    echo "<h1>Hello, $name!</h1>";
}
sayHello("John"); // Hello, John!
sayHello(); // Hello, Guest!
```

Лабораторная работа 4.1

Создание функции отрисовки таблицы умножения

Содержание лабораторной работы 4.1

Создание функции отрисовки таблицы умножения

Упражнение 1: Отрисовка таблицы с помощью функции

- В текстовом редакторе откройте файл **table.php**
- В самом начале файла создайте php-блок
- В текущем php-блоке создайте функцию **drawTable()**
- Задайте для функции три аргумента: **\$cols**, **\$rows**, **\$color**
- Перенесите код (**[Ctrl] + [X]**), который отрисовывает таблицу умножения из блока **<!-- Таблица -->** в тело функции
- В блоке **<!-- Таблица -->** (там, где ранее отрисовывалась таблица) отрисуйте таблицу умножения вызывая функцию **drawTable()** с произвольными параметрами
- Сохраните файл **table.php**
- Посмотрите результат в браузере

Области видимости переменных

```
function sayHello($name)
{
    echo "<h1>Hello, $name!</h1>";
    $name = "Вася";
}
```

```
sayHello("John");
```

```
$name = "Mike";
sayHello($name);
echo $name; // ???
```

// Обращение к глобальным переменным: вариант 1

```
function sayHello($name)
{
    echo "<h1>Hello, $name!</h1>";
    global $name;
    $name = "Вася";
}
```

```
$name = "Mike";
sayHello($name);
echo $name; // Вася
```

// Обращение к глобальным переменным: вариант 2

```
function sayHello($name)
{
    echo "<h1>Hello, $name!</h1>";
    $GLOBALS["name"] = "Вася";
}
```

```
$name = "Mike";
sayHello($name);
echo $name; // Вася
```

// Обращение к глобальным переменным: вариант 3

```
function sayHello(&$name)
{
    echo "<h1>Hello, $name!</h1>";
    $name = "Вася";
}
```

```
$name = "Mike";
sayHello($name);
echo $name; // Вася
```

```
sayHello("John"); // ERROR!!!
```

Статические переменные

```
function test(){  
    $a = 0;  
    echo $a++;  
}
```

```
test(); // 0  
test(); // 0  
test(); // 0
```

```
function test(){  
    static $a = 0;  
    echo $a++;  
}
```

```
test(); // 0  
test(); // 1  
test(); // 2
```

Лабораторная работа 4.2

Создание функции отрисовки меню навигации по сайту

Содержание лабораторной работы 4.2

Создание функции отрисовки меню навигации по сайту

Упражнение 1: Отрисовка меню с помощью функции

- В текстовом редакторе откройте файл **index.php**
- В начале файла в php-блоке создайте функцию **drawMenu()**
- Задайте для функции первый аргумент **\$menu** - в него будет передаваться массив, содержащий структуру меню
- Задайте для функции второй аргумент **\$vertical** со значением по умолчанию равным **true**. Данный параметр указывает, каким образом будет отрисовано меню - вертикально или горизонтально
- Перенесите код (**[Ctrl] + [X]**), который отрисовывает меню навигации из блока **<!-- Меню -->** в тело функции
- Измените код таким образом, чтобы меню отрисовывалось в зависимости от входящего параметра **\$vertical** - либо вертикально, либо горизонтально
- В блоке **<!-- Меню -->** (там, где ранее отрисовывалось меню) отрисуйте меню навигации вызвав функцию **drawMenu()** с необходимыми параметрами
- Сохраните файл **index.php**
- Посмотрите результат в браузере

Возвращаемые значения

```
function square($num) {  
    return $num * $num;  
    // Этот код никогда не исполнится  
    echo "Мертвый код";  
}  
  
echo square(4);    // 16  
$result = square(4);
```

```
// Возвращение массива  
function numbers() {  
    return [1, 2, 3];  
}  
list($one, $two, $three) = numbers();
```

```
// Разыменовывание массива (PHP 5.4)  
$two = numbers()[1];
```

Использование аргументов переменной длины

```
foo(1, 2, 3);
```

```
function foo() {  
    $numargs = func_num_args();  
    echo "Всего аргументов: $numargs\n";  
  
    echo "Второй  
        аргумент: " . func_get_arg(1) . "\n";  
  
    $args = func_get_args();  
    foreach ($args as $key => $value) {  
        echo "Аргумент $key : $value\n";  
    }  
}
```

Уточнение типа

```
function foo(array $var) {  
    // Ожидается только массив!  
}
```

```
function mult($num) {  
    return $num * $num;  
}  
// PHP 5.4  
function bar(callable $var, $arg) {  
    return $var($arg);  
}  
bar("mult", 4);
```

Что мы изучили?

- Научились создавать и вызывать функции
- Функции могут принимать аргументы
- Аргументы могут иметь значения по-умолчанию
- Функция может не иметь объявленных аргументов, но мы имеем доступ к переданным параметрам
- Переменные внутри функции локальны, но мы имеем несколько вариантов доступа к глобальным переменным
- Функция может возвращать значение

Модуль 5

PHP. Уровень 1

Что внутри PHP?

Темы модуля

- Учимся читать документацию
- Как читать описание функций
- Обзор встроенных функций
- Встроенные константы и псевдоконстанты
- Суперглобальные переменные
- Функции эмуляции SSI

Документация РНР

- <http://php.net/docs.php>
 - <http://php.net/download-docs.php>
 - <http://php.net/manual/ru>
 - [Установка и настройка](#)
 - [Справочник языка](#)
 - [Appendices](#)
 - [Справочник функций](#)

Обзор встроенных функций

- Функции для работы с переменными
- Математические функции
- Функции обработки строк
- Функции для работы с массивами
- Функции даты и времени

Языковые конструкции

- die и exit
- echo и print
- isset и unset
- include и include_once
- require и require_once
- empty
- eval
- list
- return

Константы и псевдоконстанты

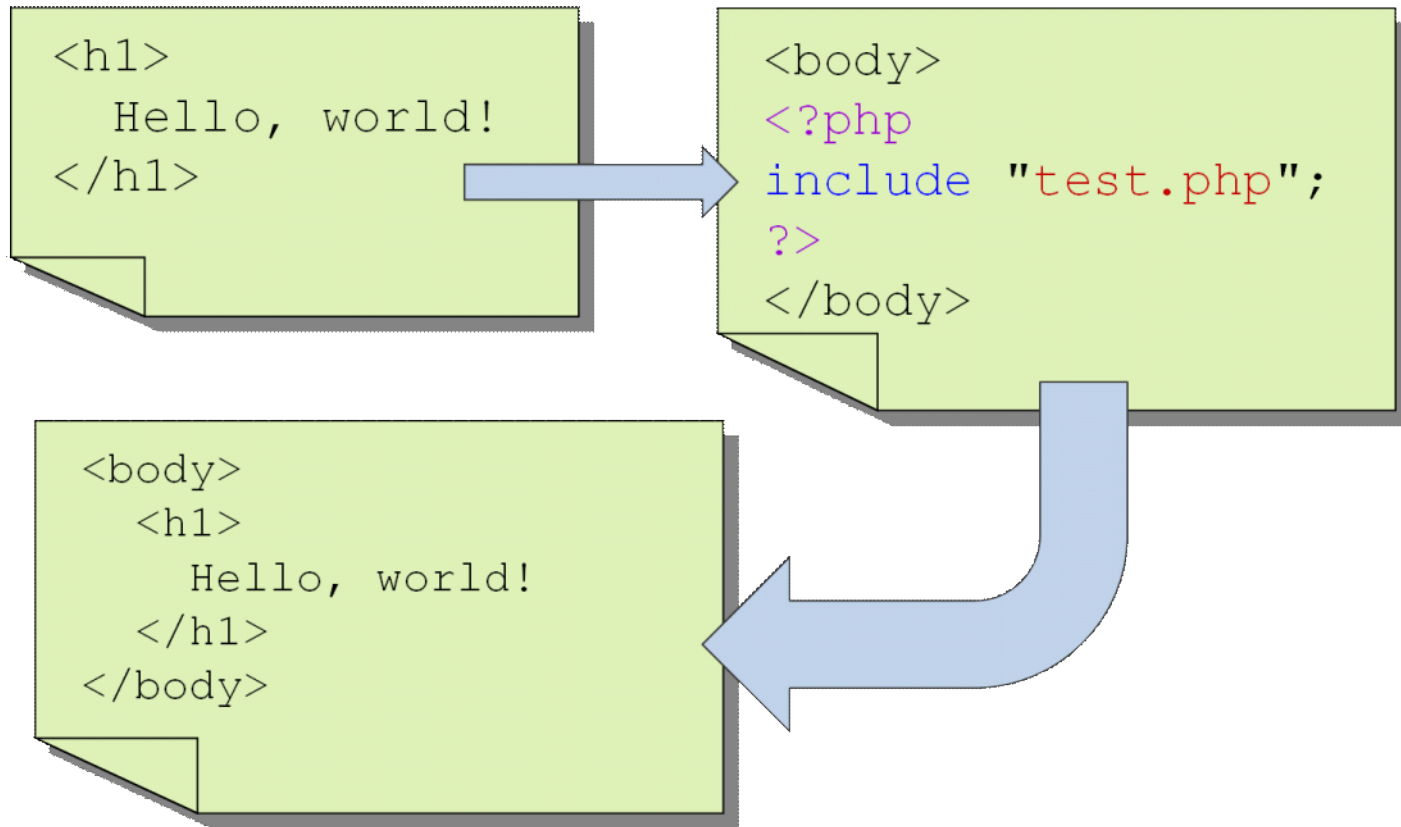
- Псевдоконстанты
 - `__LINE__`
 - `__FILE__`
 - `__FUNCTION__`
 - `__DIR__`

- `get_defined_constants(true)`
 - `E_ALL`
 - `M_PI`
 - `PHP_VERSION`
 - `PHP_OS`
 - ...

Суперглобальные переменные

- `$GLOBALS`
- `$_ENV`
- `$_SERVER`
- `$_COOKIE`
- `$_SESSION`
- `$_FILES`
- `$_GET`
- `$_POST`
- `$_REQUEST`

Принцип подключения файлов



// Файл vars.php

```
<?php
  $color = 'green';
  $fruit = 'apple';
?>
```

// Файл test.php

```
<?php
  echo "A $color $fruit"; // A
  include 'vars.php';
  echo "A $color $fruit"; // A green apple
?>
```

```
include 'vars.php';
require 'vars.php';
include_once 'vars.php';
require_once 'vars.php';
```

Лабораторная работа 5

Конструирование сайта

Содержание лабораторной работы 5

Конструирование сайта

Упражнение 1: Создание подключаемых файлов

- В папке сайта создайте папку под названием **inc**
- В текстовом редакторе создайте новый файл
- Перенесите (**[Ctrl] + [X]**) в файл php-блок с функцией **drawTable()** из файла **table.php**
- Перенесите (**[Ctrl] + [X]**) в файл функцию **drawMenu()** из файла **index.php**
- Сохраните файл под именем **lib.inc.php** в папке **inc** и сохраните файлы **index.php** и **table.php**
- В текстовом редакторе создайте новый файл и создайте в нем php-блок
- Перенесите (**[Ctrl] + [X]**) в файл весь php-код из самого верхнего php-блока файла **index.php**
- Сохраните файл под именем **data.inc.php** в папке **inc** и сохраните файл **index.php**
- В текстовом редакторе создайте новый файл
- Перенесите (**[Ctrl] + [X]**) в файл всё, что находится внутри блока **<!-- Верхняя часть страницы -->** из файла **index.php**
- Сохраните файл под именем **top.inc.php** в папке **inc** и сохраните файл **index.php**
- В текстовом редакторе создайте новый файл
- Перенесите (**[Ctrl] + [X]**) в файл всё, что находится внутри блока **<!-- Навигация -->** из файла **index.php**
- Сохраните файл под именем **menu.inc.php** в папке **inc** и сохраните файл **index.php**
- В текстовом редакторе создайте новый файл
- Перенесите (**[Ctrl] + [X]**) в файл всё, что находится внутри блока **<!-- Нижняя часть страницы -->** из файла **index.php**

- Сохраните файл под именем **bottom.inc.php** в папке **inc** и сохраните файл **index.php**
- В текстовом редакторе создайте новый файл
- Перенесите (**[Ctrl] + [X]**) в файл всё, что находится внутри блока **<!-- Область основного контента -->** из файла **index.php**
- Сохраните файл под именем **index.inc.php** в папке **inc** и сохраните файл **index.php**

Упражнение 2: Подключение файлов

- В текстовом редакторе откройте (если не открыт) файл **index.php**
- В самом верху файла подключите файлы **lib.inc.php** и **data.inc.php** из папки **inc**
- В блоке **<!-- Верхняя часть страницы -->** подключите файл **top.inc.php** из папки **inc**
- В блоке **<!-- Навигация -->** подключите файл **menu.inc.php** из папки **inc**
- В блоке **<!-- Область основного контента -->** подключите файл **index.inc.php** из папки **inc**
- В блоке **<!-- Нижняя часть страницы -->** подключите файл **bottom.inc.php** из папки **inc**
- Сохраните файл **index.php**
- Посмотрите результат в браузере

Обработка ошибок

```
// Функция перехвата ошибок
function myError($errno, $errmsg, $errfile, $errline){

    // Логгируем пользовательские ошибки
    switch ($errno) {
        case E_USER_ERROR:
        case E_USER_WARNING:
        case E_USER_NOTICE:
            error_log("$errmsg\n", 3, "error.log");
        }
    }

// Установка перехватчика ошибок
set_error_handler("myError");

// Отлавливаем ошибки
if ($error)
    trigger_error("Что-то случилось", E_USER_ERROR);
```

Что мы изучили?

- Познакомились с документацией по PHP
- Научились читать прототипы функций
- Совершили обзор основных групп встроенных функций
- Познакомились с предопределёнными константами, псевдоконстантами и суперглобальными переменными
- Научились подключать файлы друг к другу
- Узнали, как обрабатывать ошибки

Модуль 6

РНР. Уровень 1

Работа с HTTP: формы

Темы модуля

- HTTP 1/1
 - Заголовки запроса и ответа
 - Статусы сервера
- Переменные окружения сервера
- Работа с веб-формами:
 - Передача и приём данных
 - HTTP методы GET и POST
- Проверка передаваемых значений

HTTP: HyperText Transfer Protocol

GET /folder/index.html HTTP/1.1

Host: www.example.com

Accept: */*

Accept-Language: ru

Referer: <http://google.com/?q=sometext>

User-Agent: Mozilla 4.0 (compatible; MSIE 6.1,...)

HTTP/1.1 200 OK

Server: Microsoft IIS 7.5

Content-Type: text/html

Content-Length: 16345

Last-Modified: Sun, 03 Jul 2005 18:00:00 GMT

<html>

...

</html>

Использование веб-форм

```
<form action="..." method="...">
```

Логин:

```
<input name="login" type="text">
```

Пароль:

```
<input name="pwd" type="password">
```

```
<input type="submit">
```

```
</form>
```

GET /action.php?login=John&pwd=1234 HTTP/1.1 ↵
... ↵
↵

POST /action.php HTTP/1.1 ↵
... ↵
Referer: <http://example.com/page-with-form.html> ↵
Content-Type: application/x-www-form-urlencoded ↵
Content-Length: 19 ↵
↵
login=John&pwd=1234
↵

Приём и обработка данных

```
// Приём параметров QueryString
echo $_GET['name'];
echo $_GET['age'];

// Приём параметров из тела запроса и их фильтрация
$name = trim( strip_tags( $_POST['name'] ) );
$age = (int) $_POST['age'];

echo "<p>Ваше имя $name</p>";
echo "<p>Вам $age лет</p>";

// Каким методом был осуществлён запрос?
if( $_SERVER['REQUEST_METHOD'] == 'POST' )
    echo "Данные переданы методом POST";
```

Лабораторная работа 6

Передача параметров на сервер

Содержание лабораторной работы 6

Передача параметров на сервер

Упражнение 1: Создание подключаемых файлов

- В текстовом редакторе откройте файл **about.php**
- Удалите всё, кроме содержимого блока **<!-- Область основного контента -->**
- Сохраните файл **about.php**
- В текстовом редакторе откройте файл **contact.php**
- Удалите всё, кроме содержимого блока **<!-- Область основного контента -->**
- Сохраните файл **contact.php**
- В текстовом редакторе откройте файл **calc.php**
- Удалите всё, кроме содержимого блока **<!-- Область основного контента -->**
- Сохраните файл **calc.php**

Упражнение 2: Изменение меню

- В текстовом редакторе откройте файл **data.inc.php**
- Необходимо изменить значения во всех элементах **href** массива **\$leftMenu** на **index.php** и добавить параметр (например **id**), который будет передавать методом **GET** уникальные значения
- Измените значения, которые в результате могут выглядеть так:
index.php
index.php?id=about
index.php?id=contact
index.php?id=table
index.php?id=calc
- Сохраните файл **data.inc.php**

Упражнение 3: Приём данных от пользователя

- В текстовом редакторе откройте файл **index.php**
- В верхней части файла перед закрывающим тэгом **?>** напишите:

```
// Инициализация заголовков страницы
$title = 'Сайт нашей школы';
$header = "$welcome, Гость!";
$id = strtolower(strip_tags(trim($_GET['id'])));
switch($id){
    case 'about':
        $title = 'О сайте';
        $header = 'О нашем сайте';
        break;
    case 'contact':
        $title = 'Контакты';
        $header = 'Обратная связь';
        break;
    case 'table':
        $title = 'Таблица умножения';
        $header = 'Таблица умножения';
        break;
    case 'calc':
        $title = 'Он-лайн калькулятор';
        $header = 'Калькулятор';
        break;
}
```

- Между тэгами <title></title> напишите:
<?php echo \$title?>
- В блоке <!-- Заголовок --> между тэгами <h1></h1> напишите:
<?php echo \$header?>
- В блоке <!-- Область основного контента --> удалите всё содержимое и напишите:

```
<?php
switch($id){
    case 'about':
        include 'about.php';
        break;
    case 'contact':
        include 'contact.php';
        break;
    case 'table':
        include 'table.php';
        break;
    case 'calc':
        include 'calc.php';
        break;
    default:
        include 'index.inc.php';
}
?>
```

- Сохраните файл **index.php**
- Посмотрите результат в браузере. Обратите внимание, что файл **table.php** пока не готов к использованию

Упражнение 4: Допиливаем таблицу умножения

- В текстовом редакторе откройте файл **table.php**
- Удалите всё, кроме блока **<!-- Область основного контента -->**
- В верхней части файла напишите:


```
<?php
    if($_SERVER['REQUEST_METHOD'] == 'POST'){
        $cols = abs((int) $_POST['cols']);
        $rows = abs((int) $_POST['rows']);
        $color = trim(strip_tags($_POST['color']));
    }
    $cols = ($cols) ? $cols : 10;
    $rows = ($rows) ? $rows : 10;
    $color = ($color) ? $color : 'yellow';
?>
```
- В блоке **<!-- Таблица -->** исправьте вызов функции на:


```
drawTable($cols, $rows, $color);
```
- В значении атрибута **action** тэга **<form>** напишите:


```
<?= $_SERVER['REQUEST_URI']?>
```
- Добавьте в тэг **<form>** атрибут **method** со значением **POST**
- Сделайте так, чтобы введенные значения оставались в текстовых полях формы после перезагрузки страницы
- Сохраните файл **table.php**
- Посмотрите результат в браузере

Что мы изучили?

- Узнали, как взаимодействуют браузер и сервер
- Получили понимание HTTP-заголовков запроса и ответа
- Научились отправлять и принимать данные из веб-формы
- Поняли, что входящие данные необходимо фильтровать

Итоговая работа

Создание простого калькулятора

Содержание итоговой работы

Создание простого калькулятора

Уяснение задачи

- В текстовом редакторе откройте файл **calc.php**
- Калькулятор принимает два числа и производит над ними математические действия
- Математические действия зависят от передаваемого оператора. При том что:
 - Калькулятор оперирует только целыми числами (тип integer)
 - Калькулятор понимает 4 действия: сложение, вычитание, умножение и деление
- Результат выводится в виде строки. Например, при переданных данных **2, 3** и **+**, можно вывести: **Результат 2 + 3 = 5**

Упражнение 1: Прием данных из формы

- Убедитесь, что передача данных передаваемых веб-формой осуществлена методом **POST**
- Примите данные переданные веб-формой
- Убедитесь, что все данные пришли со значениями
- Отфильтруйте пришедшие данные

Упражнение 2: Получение необходимого результата

- Используя управляющую конструкцию **switch**, производите различные математические действия в зависимости от оператора
- В случае деления, проверьте делитель на равенство с нулем (на ноль делить нельзя)
- Помните, что калькулятор выполняет только 4 действия, а пользователь может передать неверный оператор - это надо отследить и уведомить пользователя об ошибке

Упражнение 3: Вывод результата и проверка работы

- Выведите результат вычислений перед html-кодом отрисовки веб-

формы

- Сохраните файл **calc.php**
- Запустите калькулятор в браузере и проверьте его работу

Что почитать?

- Документация РНР

Что дальше?

- PHP. Уровень 2. Разработка web-сайтов и взаимодействие с MySQL