

Основы разработки на стороне сервера

СОЗДАНИЕ WEB-ПРИЛОЖЕНИЙ, ИСПОЛНЯЕМЫХ НА СТОРОНЕ
СЕРВЕРА ПРИ ПОМОЩИ ЯЗЫКА ПРОГРАММИРОВАНИЯ PHP, СУБД
MYSQL И ТЕХНОЛОГИИ AJAX

МОДУЛЬ 01. ОСНОВЫ ЯЗЫКА ПРОГРАММИРОВАНИЯ PHP

Содержание

1. Что такое веб-сервер?	6
2. Как работает Apache	7
3. Настройка окружения	11
4. Введение в PHP	14
5. Типы данных	18
6. Подключение файлов	21
Дополнительная литература	25

1. Что такое веб-сервер?

ОСНОВНЫЕ ОПРЕДЕЛЕНИЯ, ВИДЫ
И ПРИНЦИП РАБОТЫ

Основные определения

- *С точки зрения аппаратной части, **веб-сервер** — это компьютер, который хранит файлы сайта (стили, скрипты, картинки и другие) и доставляет их на клиентское устройство.*
- *С точки зрения программной части, **веб-сервер** — это множество программ, которые контролируют доступ веб-пользователей к файлам на удалённом компьютере.*

Виды веб-сервера

- **Статический веб-сервер** состоит из аппаратной части с HTTP-сервером. Такой сервер отправляет размещённые файлы «как есть».
- **Динамический веб-сервер** состоит из статического веб-сервера и дополнительного ПО (чаще виртуального сервера и базы данных). Такой веб-сервер может выполнять сценарии.

Принцип работы

- Часть сайтов в Интернете обеспечивается работой благодаря веб-серверу Apache. При запросе клиента на адрес отправляется запрос информации на сервер, а Apache возвращает ответ с необходимыми данными (страницей, текстом, мультимедиа и так далее).

2. Как работает Apache

РАЗМЕЩЕНИЕ ФАЙЛОВ НА СЕРВЕРЕ,
КОНФИГУРАЦИЯ И КОНФИГУРАЦИОННЫЕ ФАЙЛЫ

Размещение файлов на сервере

- **Корневая папка сервера (public_html)** — каталог, к которому получает доступ пользователь, не должен включать важные скрипты, файлы конфигурации и так далее.
- **Индексный файл** — файл главной страницы каталога на сервере. Этот файл загружается, когда пользователь обращается к каталогу напрямую.

Конфигурация веб-сервера

Самые распространённые способы настройки — это изменение **главного файла конфигурации сервера Apache** (`httpd.conf`) или **файла уровня каталога** (`.htaccess`). Оба файла можно изменять в любом текстовом редакторе.

Различия файлов конфигурации

- **Главный конфигурационный файл** содержит основное техническое описание работы фоновой функции. Управлять настройками файла могут только администраторы, а изменения вступают в силу только после перезагрузки сервера.
- **Файл конфигурации каталога** отвечает за переопределение настроек главного конфигурационного файла для отдельного каталога и позволяет вносить изменения в работу веб-сервера без прав администратора.

3. Настройка окружения

ВЕБ-СЕРВЕР ХАМРР И ТЕКСТОВЫЕ РЕДАКТОРЫ

Веб-сервер ХАМРР

Для работы с PHP потребуется установить сборку веб-сервера ХАМРР, которая включает в себя поддержку выполнения кода PHP, а также работу с базой данных MySQL.

Загрузить установочный пакет можно с официального сайта: <https://www.apachefriends.org>. Для установки сборки могут потребоваться права администратора.

Текстовые редакторы

В разработке скриптов PHP можно использовать различные текстовые редакторы. В некоторых редакторах предусмотрена расширенная поддержка работы с PHP, включая выделение ключевых терминов и интеграцию с инструментарием.

Основные редакторы с поддержкой PHP

- [Visual Studio Code](#)
- [Brackets](#)
- [Atom](#)
- [WebStorm](#) \$
- [Sublime Text](#) \$

4. Введение в PHP

ОСНОВНЫЕ ОПРЕДЕЛЕНИЯ, СОЗДАНИЯ СКРИПТОВ,
АЛЬТЕРНАТИВНЫЙ СИНТАКСИС

Основное определение

- **PHP (Hypertext Preprocessor)** — это язык программирования, который создан для веб-разработки.
- Основная задача PHP — «оживление» HTML-страниц. PHP умеет исполнять сценарии (код).
- В ходе исполнения PHP может изменить или создать HTML-код, который результатом сценария.

Создание скриптов

- Скрипты PHP могут работать только в файле с расширением .php и начинаются с команд `<?php` или `<?`.
- Переменные в PHP обозначаются знаком доллара (`$variable`); их имена чувствительны к регистру.
- Благодаря динамической типизации возможно не указывать тип данных перед объявлением переменной.

Альтернативный синтаксис

- В PHP имеется альтернативный синтаксис, при использовании которого не ставятся фигурные скобки.
- Альтернативный синтаксис можно использовать при интеграции кода между PHP и HTML. Языки будут разделены, что упрощает разработку и восприятие страницы.
- Смешивание синтаксиса в одном и том же блоке кода не поддерживается.

5. Типы данных

ОСНОВНЫЕ ТИПЫ ДАННЫХ И ИХ ПРЕОБРАЗОВАНИЕ

Основные типы данных

PHP поддерживает восемь основных простых типов данных:

- String
- Integer
- Double
- Boolean
- Array
- Object
- Resource
- NULL

Преобразование типов

В PHP все типы данных определяются автоматически после создания переменных, а преобразование может происходить неявным образом.

В отличие от JavaScript, невозможно сложить строки и цифры, для этого используется специальный символ для объединения — точка.

6. Подключение файлов

ОПРЕДЕЛЕНИЕ И СПОСОБЫ ПОДКЛЮЧЕНИЯ

Основное определение

В RHP можно вызвать сценарий из отдельного файла или целый файл. Такая способность называется **подключением файлов**.

Файлом может являться как RHP-сценарий, так и любой другой текстовый файл (или HTML-страница).

Способы подключения файлов

Есть несколько способов подключения файлов:

- **До выполнения сценария** (require) — код целиком заменяется на содержание до обработки запроса. Так подключаются библиотеки, HTML-страницы и так далее.
- **Во время выполнения сценария** (include) — файл включается во время выполнения кода по необходимости время выполнения программного кода.

Однократное включение

Используя конструкции однократного включения `require_once` и `include_once`, можно быть уверенным, что один файл не будет включён дважды. Работают конструкции однократного включения так же, как и `require` и `include`.

Дополнительная литература

1. Документация по PHP: <https://www.php.net>
2. Список примеров кода: <https://www.php.net/manual/ru/indexes.examples.php>
3. Информация о версии PHP: <https://www.php.net/manual/en/function.phpversion.php>
4. Подсчёт количества символов в строке: <https://www.php.net/manual/ru/function.iconv-strlen.php>

© Луцишин Михаїл, 2023

© Компьютерная Академия «Шаг», www.itstep.org

Операторы сравнения, ветвление и формы

СОЗДАНИЕ WEB-ПРИЛОЖЕНИЙ, ИСПОЛНЯЕМЫХ НА СТОРОНЕ
СЕРВЕРА ПРИ ПОМОЩИ ЯЗЫКА ПРОГРАММИРОВАНИЯ PHP, СУБД
MYSQL И ТЕХНОЛОГИИ AJAX

МОДУЛЬ 01. ЯЗЫК ПРОГРАММИРОВАНИЯ PHP

Содержание

1. Формы	3
2. Суперглобальные массивы	7
3. Операторы сравнения	11
4. Ветвление	16
Дополнительная литература	32

1. Формы

ВВОД ДАННЫХ

Основные определения

- **Веб-формы** позволяют ввести данные, которые отправляются на сервер для обработки и хранения или используются на клиента для обновления интерфейса (например, добавление элемента в список).
- **HTML-формы** — состоят из нескольких элементов управления, которыми могут быть однострочные или многострочные текстовые поля, выпадающие списки, кнопки, и так далее.

Основные теги

1. `<form>` — контейнер формы
2. `<label>` — подпись темы
3. `<input>` — поле для ввода
4. `<textarea>` — многострочное поле для ввода
5. `<button>` — кнопка

1. ФОРМЫ

The diagram illustrates a contact form structure with five numbered components:

- 1**: The entire form container, represented by a large rectangle.
- 2**: The label "Полное имя:" (Full name:).
- 3**: The input field for the full name.
- 4**: The text area for the message, labeled "Текст сообщения:" (Message text:).
- 5**: The submit button labeled "Отправить сообщение" (Send message).

Обработка данных формы

- Форма отправляется на сервер в виде запроса, который может передаваться в виде двух методов запросов — GET и POST.
- Информация из запроса преобразуется в суперглобальный массив.
- Существует заблуждение, что GET-запросы к серверу «менее безопасны», чем POST, так как передаются в открытом виде и могут быть перехвачены, но **это не так**. Без использования HTTPS любые данные будут переданы в открытом виде.

2. Суперглобальные массивы

ДЛЯ ОБРАБОТКИ ДАННЫХ ФОРМ

Определение

- **Суперглобальный массив** или **суперглобальная переменная** — это встроенные переменные, которые всегда доступны во всех областях видимости.
- То есть, в любом месте скрипта и без использования синтаксиса `global $variable`; для доступа в других функциях и методах, среди них:

Виды суперглобальных массивов

- **\$_GET** — массив переменных, переданных скрипту через URL (строку запроса);
- **\$_POST** — массив переменных, переданных в заголовке запроса HTTP.
- **\$_COOKIE** — ассоциативный массив значений, переданных через файлы Cookies.
- **\$_REQUEST** — массив, который содержит данные переменных **\$_GET**, **\$_POST** и **\$_COOKIE**.

Проверка на существование

- Массивы `$_GET` и `$_POST` будут существовать только при наличии запроса к серверу. Для проверки, существует ли дан-ная переменная в программном коде, используется функция `isset()` и принимает искомую переменную в качестве аргумента.
- GET-запрос при обработке формы подразумевает, что информация будет передана в адресной строке, как на примере:

```
https://example.kz/index.php?key=value
```

3. Операторы сравнения

ОСНОВНЫЕ ОПРЕДЕЛЕНИЯ, ВИДЫ
И ПРИНЦИП РАБОТЫ

Основные определения

- **Операторы сравнения** позволяют сравнивать два значения, такие операции всегда возвращают либо `true`, либо `false`.
- В PHP разрешается сравнивать только скалярные переменные. Массивы и объекты при сравнении преобразуются в слово `Array`.

Операторы сравнения в PHP

`$a == $b`

`$a === $b`

`$a != $b`

`$a <> $b`

`$a !== $b`

`$a < $b`

`$a > $b`

`$a <= $b`

`$a >= $b`

`$a <=> $b`

Космический корабль

Новый оператор возвращает следующие значения:

- -1 , если a меньше, чем b
- 0 , если a равно b
- 1 , если a больше, чем b

Преобразование

- В случае сравнения числа со строкой или строки, содержащие числа, строка будет преобразована в число, и сравниваться как числа.
- Преобразование типов не происходит при использовании операторов `===` и `!==`, так как в этом случае сравниваются так же типы данных.

4. Ветвление

СУЩЕСТВУЮЩИЕ ВИДЫ

Основное определение

- **Ветвление** — это операция, применяющаяся, когда выполнение кода зависит от условия. Ветвление является одной из базовых конструкций структурного программирования.
- В языке PHP существует четыре основных формы реализации ветвления — *условный оператор, тернарный оператор и оператор множественного выбора, оператор сравнения.*

Оператор `if...else`

Условный оператор начинается с ключевого слова `if` и реализует выполнение команд, когда логическое выражение принимает значение `true`.

ЕДИНИЧНЫЙ ВЫБОР

Выполняется если при выполнении условие возвращает true,,
иначе выполнение кода продолжается со следующей за условным
оператором команды.

Классический синтаксис

```
if (cond) {  
    // Code  
}
```

Альтернативный синтаксис

```
<? if (cond): ?>  
    // Alt code
```

Множественный выбор

При возвращении `false` первого блока выбор, может быть выполнено альтернативное условие — `elseif`.

Классический синтаксис

```
elseif (cond) {  
    // Code  
}
```

Альтернативный синтаксис

```
<? elseif (cond): ?>  
    // Alt code
```

Двойной выбор

Используется в качестве последнего возможного альтернативного значения, который находится в самом конце. Обозначается как `else`.

Классический синтаксис

```
else {  
    // Code  
}
```

Альтернативный синтаксис

```
<? else: ?>  
    // Alt code
```

Конец оператора

Используется только в альтернативном синтаксисе в качестве завершающего элемента

Классический синтаксис

—

Альтернативный синтаксис

`<? endif; ?>`

Тернарный оператор

- Тернарный оператор работает почти как условный оператор, но вместо ключевых слов требуется писать ? в случае значения `true` и : для `false`.
- Тернарный оператор имеет следующий синтаксис:
`cond ? true : false;`

Switch...case

- Конструкцию выбора можно использовать, если предполагаемых вариантов много и для каждого варианта нужно выполнить определённые действия.
- Использование условного оператора с множественным выбором становится неудобным и менее производительным.
- В случае использования альтернативного синтаксиса **нельзя разрывать условие и первое выражение.**

Объявление конструкции

Классический синтаксис

```
switch (var) {  
    // Cases  
}
```

Альтернативный синтаксис

```
<? switch (var): ?>  
    // Alt cases
```

Первое выражение

Классический синтаксис

```
case value:  
    // Case code  
break;
```

Альтернативный синтаксис

```
case value: ?>  
    // Alt case code  
<? break; ?>
```

Последующие выражения

Классический синтаксис

```
case value:  
    // Case code  
break;
```

Альтернативный синтаксис

```
<? case value: ?>  
    // Alt case code  
<? break; ?>
```

Дополнительное выражение

Классический синтаксис

```
default:  
    // Case code  
    break;
```

Альтернативный синтаксис

```
<? default: ?>  
    // Alt case code  
<? break; ?>
```

Конец конструкции

Классический синтаксис

—

Альтернативный синтаксис

`<? endswitch; ?>`

Match

Новое выражение `match` напоминает работе оператора `switch...case` и возвращает какой-либо результат, который можно сохранить в переменной.

Имеется поддержка конструкции `default`, что позволяет задать значение по умолчанию, если совпадение не найдено (в ином случае будет получено исключение).

Сравнение происходит с учётом типа данных.

Синтаксис

Использование switch...case

```
$var = 'foo';  
switch ($var) {  
    case 'foo':  
        echo 'bar';  
        break;  
    default:  
        echo 'default';  
        break;  
}
```

Использование match

```
$var = 'foo';  
echo match ($var) {  
    'foo', 'baz' => 'bar',  
    default => 'default'  
};
```


Дополнительная литература

1. Документация по PHP: <https://www.php.net>
2. Операторы сравнения: <https://www.php.net/manual/ru/language.operators.comparison.php>
3. Условный оператор: <https://www.php.net/manual/ru/control-structures.elseif.php>
4. Конструкция выбора: <https://www.php.net/manual/ru/control-structures.switch.php>
5. Подсчёт количества символов в строке: <https://www.php.net/manual/ru/function.iconv-strlen.php>
6. Работа с формами: <https://www.php.net/manual/ru/tutorial.forms.php>

© Луцишин Михайл, 2023

© Компьютерная Академия «Шаг», www.itstep.org