

УТВЕРЖДАЮ
Генеральный директор
ООО «Экварта Групп»

Горелов Сергей Владиславович

«12» мая 2026 г.



РАБОЧАЯ ПРОГРАММА - ОНЛАЙН-ЛЕКЦИЯ (ВЕБИНАР)
по Дополнительной профессиональной программе –
программе повышения квалификации
«ВЕБ-ДИЗАЙН И РАЗРАБОТКА»

Тема 5: «Веб-дизайн и программирование» (1 вебинар)

Тип: Онлайн-лекция (вебинар)

Объем часов: 4 ак.ч.

П Л А Н

1. ВВЕДЕНИЕ В WEB-КОНСТРУИРОВАНИЕ
2. ПРОГРАММИРОВАНИЕ НА PHP.
3. СТИЛЕВОЕ ОФОРМЛЕНИЕ HTML-ДОКУМЕНТОВ

1. ВВЕДЕНИЕ В WEB-КОНСТРУИРОВАНИЕ

При реализации проектных творческих работ и коллективного общения (в том числе удаленного) является создание собственных web-сайтов. При этом важно не забывать, что web-технология позволяет не только представить результаты своей работы на всеобщее обозрение в сети Интернет, но и создать такой сайт в рамках локальной сети или подготовить свои медиа материалы в виде локальной копии сайта, распространяемой на любом носителе и просматриваемой на компьютере без подключения к Интернету. В том числе в форме такой «локальной копии сайта» может быть разработана интерактивная интерфейсная (пользовательская) оболочка, например, существенно повышающая удобство работы с компакт-диском, на котором учитель или учащиеся размещают разработанные ими учебные, демонстрационные или дополнительные материалы.

Для создания сайта или «интерфейсной оболочки» могут использоваться различные инструментальные средства web-редактирования, прежде всего — специальные редактирующие программы и программные пакеты, реализующие принципы «визуального редактирования» web-страниц и сайтов. Кроме того, современные версии офисного пакета Microsoft Office и ряд других прикладных программ позволяют сохранять обрабатываемые в них документы в web-совместимом формате, позволяющем размещать их на web-сайте и просматривать через сети Интернет. Однако для понимания

принципов web-конструирования по-прежнему необходимо знание языка гипертекстовой разметки HTML — как основы основ web-представления текстовых и медиа материалов.

Данная информация призвана предназначена для обучения основным приемам создания web-сайтов с использованием языка HTML, а также сопутствующих web-технологий (стилевые таблицы, слои, фреймы, скриптовое программирование и др.).

Предполагается, что слушатели обладают навыками работы в среде операционной системы Windows, работы в многоуровневой файловой системе, а также ввода и редактирования текста хотя бы в простейших текстовых редакторах (Блокнот).

Основные определения

Язык HTML (HyperText Markup Language — «язык гипертекстовой разметки») — набор команд (тегов), вставляемых в текст web-страницы и определяющих форматирование абзацев, вид шрифта, ссылки на внешние файлы, другие web-страницы или части той же web-страницы.

Гипертекст — «многомерный» текстовый документ (либо объединение нескольких текстовых документов), построенный по принципу «нелинейного» структурирования материала за счет применения гипертекстовых ссылок (как внутренних, в пределах данного документа, так и перекрестных, указывающих на другие документы, в том числе размещенные на других компьютерах сети, или на их фрагменты), позволяющих одним щелчком мыши перемещаться из одной «смысловой точки» гипертекстового документа в другую.

Контейнер — конструкция из парных «открывающего» и «закрывающего» тегов. При этом действие открывающего тега и его параметров распространяется на весь текстовый фрагмент, заключенный внутри контейнера.

Метатеги (теги, начинающиеся со слова META) — размещаются внутри контейнера HEAD и служат для указания различных дополнительных сведений о содержимом страницы. На экран браузера текст метатегов не выводится.

Среди метатегов можно выделить по крайней мере три группы:

- команды управления отображением страницы — прежде всего указание кодовой таблицы кириллицы;
- команды, предназначенные для автономных поисковых программ, рассылаемых по сети поисковыми серверами для сбора сведений о новых web-сайтах;
- теги, предоставляющие дополнительную информацию о содержании страницы

Типовая структура метатега обычно включает в себя два параметра: NAME (или HTTP-EQUIV) и CONTENT, первый из которых определяет название (тип) метатега, определяющее его назначение, а второй — собственно значение метатега.

Web-страница (интернет-страница, WWW-страница), web-документ — обособленный документ, хранящийся в отдельном файле на диске и включающий в себя текст, отображаемый на экране во время просмотра в браузере, а также теги языка HTML, дополненный хранящимися в отдельных файлах и подгружаемыми дополнительно по размещенным в тексте страницы ссылкам мультимедиа-иллюстрациями (рисунками, видео, аудио- и пр.).

Сайт, web-сайт — набор web-страниц, составляющих единую подборку и связанных между собой перекрестными ссылками. Одна из этих страниц является основной (головной, индексной, стартовой) и автоматически выдается на просмотр пользователю, указавшему в браузере только адрес сайта, тогда как все остальные страницы, как правило, вызываются из основной с помощью гиперссылок.

Сервер — обособленный компьютер, подключенный к сети Интернет (чаще всего круглосуточно) и имеющий собственный адрес (URL), на диске которого расположены один или несколько сайтов. Серверы могут быть служебными (пример — поисковые системы), принадлежащими организации или частному лицу (обычно такой сервер содержит только один сайт, и тогда эти два понятия можно считать идентичными) или общедоступными для размещения сайтов всех желающих (например, сервер www.narod.ru).

Еще одно значение понятия сервер предполагает кроме самого компьютера как узла сети также установленное на нем специализированное программное обеспечение, поддерживающее информационный обмен с пользователями (web-сервер, FTP-сервер, почтовый сервер и т. д.).

Локальный компьютер (терминал) — компьютер пользователя, работающего в Интернете. При посещении несколькими пользователями одного и того же сервера размещенные на его диске web-страницы и другие необходимые файлы данных (графика, звук, цифровое видео и пр.) пересылаются (копируются) по сети на диски локальных компьютеров (локальные диски) и отображаются на экранах, запущенных на локальных компьютерах браузеров.

«Домашняя страница» (HomePage) — своего рода «наследие» прежних времен, когда предполагалось создание частными пользователями лишь простейших одностраничных web-страниц — своего рода «сетевых визитных карточек» их владельцев. В настоящее время этот термин используется для обозначения указанной в настройках браузера некоторой (назначенной по умолчанию или самим пользователем) «изначальной» web-страницы, автоматически загружаемой при запуске браузера (это может быть любая, чаще всего головная страница какого-либо сайта в сети Интернет, произвольный HTML-документ на жестком диске локального компьютера либо «пустая страница» — web-документ с белым фоном, не содержащий никакого текста).

Браузер (web-браузер) — программа, запускаемая на локальном компьютере для просмотра web-страниц, их сохраненных на локальном диске копий, а также любых документов, созданных с использованием языка HTML. Сегодня наиболее популярными являются браузеры Microsoft Edge, Opera, Mozilla, Firefox и др.

Фрейм — «вложенное» окно («подокно», поле), создаваемое в окне браузера при помощи специальной конструкции HTML-тегов, содержимое которого (отдельная web-страница, рисунок и пр.) полностью независимо от содержимого остальных (соседних) фреймов.

Каскадная стилевая разметка (CSS, Cascading StyleSheet — «каскадная стилевая таблица») — технология, позволяющая разделить содержание и оформление (форматирование) web-страницы путем создания «шаблонных» (типовых) наборов параметров оформления тех или иных фрагментов текста (заголовков, ссылок, таблиц и пр.) — стилей. При указании для любого абзаца того или иного стиля этот абзац автоматически приобретает все указанное в данном стиле оформление, а любое изменение стиля немедленно отражается на оформлении всех абзацев, которым назначен этот стиль. Набор стилей называется стилевой таблицей. В HTML можно создавать разные уровни («каскады») стилевых таблиц — как общие, так и «частные», содержащие общие стили для всего сайта, стили для отдельных его разделов, особые стили для отдельных страниц. При этом значения однотипных параметров более «частных» стилей перекрывают (замещают) значения более общих, а локальное форматирование при помощи обычных тегов — перекрывает однотипное стилевое оформление.

Слой — составной элемент web-страницы с «прозрачным» фоном, содержащий какие-либо объекты (абзацы текста, таблицы, списки, иллюстрации и пр., — все, что может содержаться на обычной web-странице), который может быть «наложен» поверх «основного» содержимого web-страницы и/или поверх других слоев, аналогично наложению текста и рисунков на прозрачной пленке поверх изображения на листе бумаги. При этом можно управлять как взаимным расположением по высоте наложенных друг на друга слоев, так и расположением каждого слоя (понимаемого как некая прямоугольная область с заданным содержимым) по горизонтали и вертикали относительно верхнего левого угла окна браузера, создавая тем самым различные визуальные и даже (при помощи скриптового программирования) анимационные эффекты.

Скриптовое программирование — возможность создавать динамические и/или интерактивные web-страницы (технология Dynamic HTML, или DHTML), содержимое которых может меняться согласно заданному алгоритму и/или в зависимости от действий пользователя, просматривающего web-страницу. Эта технология реализуется путем

написания скрипта (текста программы на скриптовом языке программирования), внедряемого непосредственно в HTML-код web-документа: при просмотре web-страницы браузер выделяет этот скрипт и выполняет записанную в нем программу. Среди скриптовых языков программирования наиболее часто используются языки JavaScript и VBScript, представляющие собой упрощенные разновидности языков программирования Java и Visual Basic соответственно.

«Активное содержимое» web-страницы — условный термин, используемый, в частности, в браузерах Microsoft Edge последних версий и обозначающий любые элементы web-страницы, реализующие динамические, визуальные эффекты, интерактивность и пр., например, скрипты, фильтры, «всплывающие» окна, содержащие рекламу, и т. д. Как правило, настройки безопасности в современных браузерах предписывают автоматически блокировать «активное содержимое» просматриваемых web-страниц, поэтому при выполнении ряда соответствующих заданий, предлагаемых в данной книге, необходимо заблаговременно изменить настройки безопасности в браузере, либо каждый раз вручную разрешать отображение активного содержимого web-страницы, щелкая мышью на соответствующей ссылке в верхней части окна браузера

Структура HTML-документа

Документ HTML представляет собой обычный текстовый файл, который содержит конструкции языка HTML. Поэтому этот документ можно создавать в обычных текстовых редакторах, например в программе Блокнот, а затем сохранять созданные файлы с расширением .htm или .html.

Суть языка HTML - в разметке текста с помощью управляющих символов - тегов, которые располагаются в угловых скобках.

Большинство тегов парные, т.е. имеют открывающий элемент `<>` и закрывающий элемент `</>`.

Между ними и находятся коды, которые распознает браузер.

HTML-документ всегда должен начинаться открывающим тегом `<html>` и заканчиваться закрывающим `</html>`.

Внутри расположены: блок заголовка `<head> </head>` и тело `<body> </body>`, в котором размещаются тексты, рисунки, аудио и видеофрагменты.

В блоке заголовка размещается тег `<title>` текст `</title>`. Текст, указанный в этом теге, отображается в заголовке окна браузера.

В HTML-документе можно разместить комментарии, которые браузером не отображаются: `<!-- Комментарии-->`.

Пример 1. Создать простой HTML-документ в Блокноте. Сохранить созданный файл `prim1.htm`. Открыть созданный документ в браузере.

Структура такого документа представляется следующим образом:

```
<html> <!--Открытие HTML документа>
<head>
<title> Заголовок </title>
</head>
<body> <!--Содержание (тело) документа> Текст первой странички
</body>
</html>
```

Оформление HTML-документа

Теги языка HTML могут содержать атрибуты, которые являются параметрами или свойствами элементов разметки документа.

Правило записи атрибутов в теге следующее:

```
<тег атрибут1=значение атрибут2=значение ...>
```

Тег `<body>` определяет внешний вид всей веб-страницы, в то время как отдельные ее элементы, например, заголовки, таблицы могут иметь свое особенное оформление.

Выбор цвета страницы, фонового рисунка и цвета текста на ней является весьма важным, так как от этого зависит визуальное восприятие всего сайта.

Цвет страницы задается атрибутом bgcolor, а цвет текста - text. Значением атрибутов является цвет, который задается своим названием на английском языке или его шестнадцатеричным кодом.

Работа с редактором визуального конструирования FrontPage. Основные элементы интерфейса FrontPage.

Редактор Microsoft FrontPage относится к редакторам визуального веб-конструирования. При работе с веб-редактором FrontPage можно обойтись без знания языка разметки гипертекстовых документов HTML. Веб-страница просто конструируется на экране и сохраняется в формате .html. Выполняемые при этом действия напоминают работу в текстовом редакторе Word.

Интерфейс редактора FrontPage достаточно прост, чтобы пользователь смог быстро освоить основные приемы работы. После запуска FrontPage открывается окно. Панели Стандартная, Форматирование и Рисование по своим возможностям напоминают аналогичные панели текстового редактора Word.

Вид Рабочей области зависит от выбранного режима работы. В режиме Конструктор веб-страница конструируется из текстовых блоков и графических объектов. При этом автоматически генерируется ее HTML-код, который можно просматривать и редактировать в режиме Код. Режим с разделением является комбинацией этих двух режимов. Режим просмотр позволяет просматривать созданные страницы.

С помощью меню вид можно выбрать и другие полезные при конструировании сайта режимы. Режим страница предназначен для создания и редактирования веб-страниц. В режиме Папки просматривается структура папок сайта, выполняются файловые операции. Заметим, что любые изменения файлов и папок сайта необходимо производить из среды разработки, так как это позволяет отследить изменения в структуре взаимосвязанных веб-страниц сайта. Режим Переходы отображает структуру связей между страницами и позволяет ее изменять. Режим Гиперссылки наглядно отображает систему ссылок, обеспечивает их проверку и редактирование.

Создание веб-сайта в редакторе FrontPage

Рассмотрим этапы и приемы работы в редакторе FrontPage на примере разработки веб-сайта «Беларускія пісьменнікі».

Для создания этого веб-сайта необходимо уяснить его структуру, и предполагаемое содержание, т.е. подготовить проект сайта. Фрагмент структуры сайта для наглядности изобразим в виде двухуровневой схемы.

На первой странице (верхний уровень) размещается заголовок сайта «Беларускія пісьменнікі», текстовые гиперссылки и гиперссылки с изображением писателей и поэтов, позволяющие открыть страницы второго уровня.

На втором уровне разместим три страницы одинаковой структуры.

После просмотра любой страницы второго уровня внизу расположим гиперссылки для возврата на главную (первую) страницу сайта и две гиперссылки для движения вперед и назад по сайту.

При проектировании сайта «Беларускія пісьменнікі» мы будем использовать методику построения сверху вниз, то есть конструировать от верхнего уровня к нижнему уровню. При создании веб-страниц сайта с помощью редактора FrontPage мы будем для удобства создавать страницы снизу-вверх, то есть начнем создавать страницы нижнего уровня.

Файлы всех веб-страниц сайта размещаются в отдельной папке. Это в дальнейшем поможет упростить размещение сайта в сети Интернет. Файлу главной веб-страницы принято давать имя index или main с расширением .htm, хотя это требование не является обязательным.

В рамках каждого уровня страницы сайта имеют одинаковую структуру. Все страницы одного уровня обычно оформляются в едином стиле. Стиль оформления веб-страницы определяется стилем текста и графических элементов.

Под стилем текста понимается тип шрифта, размер, начертания; отступы, выравнивание, межстрочный интервал и др. Стиль графических элементов задается формой, размером, цветом, фактурой материала и т. д.

В редакторе FrontPage для оформления документов разработаны шаблоны – Темы.

Темой называется специально разработанный компьютерным дизайнером набор элементов оформления и цветовых схем документа.

С помощью меню Файл → Создать в появившемся окне Шаблоны веб-узлов выберем Одностраничный веб-узел.

Для облегчения подбора элементов оформления и цветовых схем используем готовый шаблон оформления – тему.

С помощью меню Формат → Тема в появившемся диалоговом окне выберем, например, тему Горизонт и укажем применить как тему по умолчанию.

Тему можно применять к отдельным страницам, а также ко всему сайту. В последнем случае при создании каждой новой страницы тема устанавливается автоматически.

Форматирование текста и вставка изображений в редакторе FrontPage

Форматирования текста в редакторе FrontPage осуществляется с помощью панели инструментов Форматирование, а также окон Формат и Абзац.

Выделив текст в редакторе FrontPage (выполняется аналогично Word) в окне Шрифт можно выбрать его имя, начертание и указать размер, а в окне Абзац указать выравнивание, отступы и межстрочный интервал. Важно помнить, что отображение веб-страницы на экране зависит от разрешения монитора и настройки браузера. Поэтому размер шрифта принято задавать в условных единицах от 1 до 7. Если размер шрифта не указан, то по умолчанию он принимается равным 3.

Для привлечения внимания к информации создается бегущая строка с помощью цепочки действий: Вставка → Веб-компонент → Бегущая строка. В окне свойства бегущей строки вводится текст, задается направление и скорость движения, время задержки и другие параметры.

Размещение изображений на веб-странице производится аналогично тому, как это выполнялось в редакторе Word (команда меню Вставка→Рисунок.)

Задавать параметры изображения на странице можно в окне положение, которое активизируется с помощью меню Формат → Положение. Можно задать размер изображения (в пикселях или процентах), выравнивание, толщину границы, обтекание текстом и другие параметры.

Важно подчеркнуть, что все изображения, которые мы видим на веб-страницах, хранятся в отдельных файлах, а на самой странице имеются лишь ссылки на соответствующие файлы.

В зависимости от обтекания текстом вставляемые изображения могут располагаться различными способами. Регулирование позиции изображения относительно других элементов веб-страницы, например текста, осуществляется в диалоговом окне, которое активизируется с помощью меню Формат → Положение.

В редакторе FrontPage предусмотрена возможность простейшей обработки изображений (коррекция яркости и контрастности, повороты, обрезка и т.д.). Для этого используется панель Рисунки аналогичная текстовому редактору Word.

На веб-странице в редакторе FrontPage вставляются таблицы, которые часто используются для структурирования информации.

С помощью таблиц можно выполнять верстку в несколько колонок, применять эффекты состыковки картинки и фона, размещать тонкие линии на всю ширину или высоту странички и т.д.

Для вставки таблицы можно воспользоваться стандартной панелью инструментов, однако наиболее полно команды работы с таблицами представлены в меню таблица. В диалоговом окне вставка таблицы можно указать не только количество строк и столбцов таблицы, но и способ выравнивания, цвет фона ячеек, цвет и толщину границ, интервал между ячейками и другие параметры.

Создание гиперссылок в редакторе FrontPage

Для создания гиперссылки требуется выделить ее в документе (определить элемент привязки) и задать адрес перехода по данной ссылке. В качестве элементов привязки могут выступать слова, группы слов, изображения.

Текстовые ссылки обычно отмечаются цветом и подчеркиваются, а графические - иногда обводятся рамкой. Благодаря этому они отличаются от обычного текста и других элементов страницы.

Ссылки могут обеспечивать переход к веб-страницам или иным документам, например, рисункам, презентациям, видеофрагментам, расположенным как на данном сайте (внутренние ссылки), так и на других сайтах (внешние ссылки). Переходы внутри документа выполняют внутристраничные гиперссылки. Они облегчают навигацию по длинной странице, например, быстрый переход из конца страницы в начало.

Перед созданием внутристраничных ссылок нужно сначала расставить закладки. Для этого курсор устанавливается на нужное место веб-страницы. С помощью меню Вставка→Закладка вызывается диалоговое окно Закладка, в котором вводится имя закладки, например, вниз.

Для организации гиперссылки необходимо выделить элемент привязки (текст или рисунок), с помощью кнопки на Стандартной панели инструментов или меню Вставка→Гиперссылка вызвать диалоговое окно, в котором указать адрес перехода (адрес веб-страницы или электронной почты, имя файла или документа, а в случае внутренней ссылки имя закладки)

По умолчанию новый документ открывается в текущем окне браузера, однако из окна Добавление гиперссылки можно щелкнуть по кнопке Выбор рамки и в открывшемся окне Конечная рамка задать открытие нового документа в новом окне.

Заметим, что в редакторе FrontPage создание гиперссылок выполняется в режиме конструктор, а проверить работу гиперссылок можно в режиме просмотр.

Публикация сайта

Публикацией сайта называют его размещение на сервере или локальном компьютере с возможностью вызова из сети (глобальной или локальной).

Некоторые серверы бесплатно предоставляют дисковое пространство под сайт, например, www.narod.ru. Процесс публикации сайта заключается в переносе файлов вашего сайта на этот сервер. Адрес перенесенного сайта может иметь вид, например, www.belpisateli.narod.ru.

Для публикации сайта, подготовленного в редакторе FrontPage, необходимо с помощью меню Файл → Опубликовать узел вызвать диалоговое окно Свойства удаленного веб-узла.

Для размещения созданного сайта на сервере narod.ru в строке Расположение удаленного веб-узла следует указать адрес ftp-узла сервиса narod.ru, который для всех пользователей является одинаковым: <ftp://ftp.narod.ru>.

В дальнейшем пользователю необходимо ввести имя и пароль в окне требуется имя и пароль.

В открывшемся новом окне слева будут размещены файлы и папки созданного локального веб-узла, а справа файлы и папки удаленного веб-узла, например, странички сайта «Беларускія пісьменнікі».

После указания режима все файлы локального веб-узла копировать на сервер необходимо нажать кнопку опубликовать веб-узел.

Когда копирование всех файлов завершится, пользователь может просмотреть размещенный в сети сайт. Для этого в строке браузера вводится адрес, например, www.belpisateli.narod.ru.

2. ПРОГРАММИРОВАНИЕ НА PHP

История PHP

PHP, как всем известно, на самом деле преемник продукта под названием PHP/FI. Созданное в 1994 году Расмусом Лерддорфом, самое первое воплощение PHP было простым набором CGI-скриптов, написанных на языке программирования Си. Изначально используя

их для отслеживания посещений своего веб-резюме, он назвал этот набор скриптов "Personal Homepages Tools" ("Инструменты для персональных домашних страниц"), но более часто упоминалось название "RHP Tools". Со временем требовалось все больше улучшений функциональности, и Расмус переписал RHP Tools, создав более крупную и богатую реализацию. Эта новая реализация была способна взаимодействовать с базами данных и многое другое, что создавало фреймворк, с помощью которого пользователи могли создавать простые динамические веб-приложения, такие как гостевые книги. В июне 1995 года Расмус » открыл исходный код RHP Tools общественности, что позволило разработчикам использовать его по своему усмотрению. Это также дало возможность пользователям исправлять ошибки в коде и улучшать его.

В сентябре того же года, Расмус расширил RHP и на короткое время убрал из названия RHP. Теперь в виде инструмента FI (сокращение от "Интерпретатор Форм"), новая реализация включала в себя некоторые основные функциональные возможности того RHP, который мы знаем сегодня. Она имела Perl-подобные переменные, автоматическую интерпретацию форм и встраиваемый в HTML синтаксис. Синтаксис языка был похож на Perl, хотя и был гораздо более ограниченным, простым, и в некоторой степени противоречивым. Для того, чтобы вставлять код в HTML-файл, разработчикам пришлось использовать HTML комментарии. Хотя этот метод был не совсем хорошо принят, FI по-прежнему набирал популярность в качестве CGI-инструмента, но все-таки не в качестве языка. Однако, перемены начались в следующем месяце, когда в октябре 1995 года Расмус выпустил полностью переписанный код. С вернувшимся именем RHP, но уже сокращенным от "Personal Home Page Construction Kit", это был первый релиз, который мог похвастаться расширенным интерфейсом сценариев. Язык намеренно напоминал Си по структуре, что делал его легким для восприятия разработчиками, знакомыми с Си, Perl и подобными языками. Будучи все еще ограниченными UNIX и POSIX-совместимыми системами, был изучен вопрос для реализации языка в Windows NT.

Код получил еще одно существенное преобразование в апреле 1996 года. Объединив названия предыдущих версий, Расмус представил RHP/FI. Реализации второго поколения начали по-настоящему развивать RHP из набора инструментов в самостоятельный язык программирования. RHP включал в себя встроенную поддержку для DBM, mSQL и Postgres95 баз данных, cookies, поддержку определяемых пользователем функций и многое другое. В июне RHP/FI была присвоена версия 2.0. Интересно, однако, что существовала только одна версия RHP 2.0. Когда она, наконец, в ноябре 1997 года преодолела статус бета-версии, парсер языка был уже полностью переписан.

Несмотря на короткую историю разработки, RHP/FI продолжал набирать популярность в молодом мире веб-разработки. В 1997 и 1998, RHP/FI стал культом для нескольких тысяч пользователей по всему миру. Исследования Netcraft в мае 1998 года показали, что почти 60 тысяч доменов передавали заголовки, содержащие "RHP". Это число равнялось примерно 1% от всех доменов в Интернете в то время. Несмотря на эти впечатляющие цифры, развитие RHP/FI было ограничено: несмотря на нескольких второстепенных участников, в целом он по-прежнему разрабатывался одним человеком.

RHP 3.0 был первой версией, напоминающей RHP, каким мы знаем его сегодня. Посчитав RHP/FI 2.0 все еще неэффективным и недостаточно функциональным для использования в коммерческих приложениях, разрабатываемых для их университетского проекта, Энди Гутманс и Зив Сураски из Тель-Авива начали еще раз заново переписывать парсер в 1997 году. Связавшись с Расмусом, они обсудили различные аспекты текущей реализации и их новой разработки RHP. Для улучшения движка и использования уже существующей базы пользователей RHP/FI, Энди, Расмус и Зив решили сотрудничать в развитии нового, независимого языка программирования. Этот совершенно новый язык был выпущен под новым именем, без упоминания о персональном использовании, как в RHP/FI 2.0. Он был назван просто "RHP" - аббревиатура, означающая рекурсивный акроним - RHP: Hypertext Preprocessor.

Одной из сильнейших сторон RHP 3.0 была возможность расширения ядра. Кроме обеспечения пользователей надежной инфраструктурой из множества различных баз данных,

протоколов и API, расширяемость PHP 3.0 привлекла к нему множество сторонних разработчиков, желающих добавить к языку свои модули. Возможно, это и был главный ключ к успеху, но стоит добавить, что немаловажным шагом оказалась поддержка ООП синтаксиса и намного более мощного и последовательного синтаксиса самого языка.

В июне 1998 года, со множеством новых разработчиков со всего мира присоединившихся к проекту, PHP 3.0 был представлен новой командой разработчиков, как официальный преемник PHP/FI. Активное развитие PHP/FI 2.0, фактически прекратившееся в ноябре прошлого года, теперь официально закончилось. Примерно через девять месяцев открытого публичного тестирования, при объявлении официального выпуска PHP 3.0, он уже был установлен на более чем 70000 доменах по всему миру, и уже не ограничивается POSIX-совместимыми операционными системами. Относительно небольшая доля доменов с установленным PHP была размещена на серверах под управлением Windows 95, 98 и NT, а также Macintosh. На пике своего развития, PHP 3.0 был установлен приблизительно на 10% веб-серверов в Интернете.

К зиме 1998 года, практически сразу после официального выхода PHP 3.0, Энди Гутманс и Зив Сураски начали переработку ядра PHP. В задачи входило увеличение производительности сложных приложений и улучшение модульности кодовой базы PHP. PHP 3.0 дал возможность подобным приложениям успешно работать с набором баз данных и поддерживать большое количество различных API и протоколов, но PHP 3.0 не имел качественной поддержки модулей и приложения работали неэффективно.

Новый движок, названный 'Zend Engine' (от имен создателей: Zeev и Andi), успешно справлялся с поставленными задачами и впервые был представлен в середине 1999 года. PHP 4.0, основанный на этом движке и принесящий с собой набор дополнительных функций, официально вышел в мае 2000 года, почти через два года после выхода своего предшественника. Кроме значительного улучшения производительности, PHP 4.0 имел еще несколько ключевых нововведений, таких как поддержка намного большего числа веб-серверов, поддержка HTTP сессий, буферизация вывода, более безопасные способы обработки вводимой пользователем информации и несколько новых языковых конструкций.

После долгой разработки и нескольких пре-релизов в июле 2004 был выпущен PHP 5. В основном он управляется ядром Zend Engine 2.0 с новой объектной моделью и множеством различных других нововведений. Команда разработчиков PHP включает в себя десятки разработчиков, а также десятки других организаций, работающих над связанными с PHP и его поддержкой проектами, такими как PEAR, PECL и документацией, а также базовую инфраструктуру сети более чем из ста серверов на шести из семи континентах мира. Основываясь на статистике прошлых лет, можно с уверенностью предположить, что PHP теперь установлен на десятки или даже, возможно, сотни миллионов доменов по всему миру.

Константы

Константы определяются в PHP-программе с помощью функции `define()`. Например: `define("PI", 3.1415927);`

После определения константа не может быть изменена. В имени константы обычно используются только заглавные буквы.

PHP имеет ряд предопределенных констант. Например,:

- `__FILE__` содержит имя файла, включая полный путь, который в данный момент читает PHP.

- `__LINE__` содержит номер строки этого файла.
- `__DIR__` представляет только путь к файлу.
- `__CLASS__` представляет имя текущего класса.
- `__FUNCTION__` представляет имя текущей функции.
- `__METHOD__` представляет имя текущего метода.
- `__NAMESPACE__` представляет имя текущего пространства имен.

Переменные

Имя любой переменной в PHP начинается со знака `$`. Имена переменных чувствительны к регистру символов.

Тип переменной не требуется задавать специально. Конкретный тип переменной устанавливается и меняется в ходе выполнения программы.

RНР поддерживает восемь типов данных:

- логический (принимает значения true или false);
- целое число;
- вещественное число с плавающей точкой;
- строка;
- объект;
- массив;
- ресурс (специальный тип);
- null (специальный тип).

Тип переменной можно проверить с помощью функции `gettype()`.

Приведение типа переменной осуществляется с помощью операторов:

- (bool) - к логическому типу;
- (int) - к целому числу;
- (double) - к вещественному числу;
- (string) - к строке;
- (array) - к массиву;
- (object) - к объекту.

Отличие от изменения типа с помощью функции `settype()` состоит в том, что оператор приведения создает временную копию нового типа, оставляя саму переменную без изменений.

Иногда для упрощения логики программы удобнее использовать переменные имена переменных. RНР предоставляет такую возможность в виде динамических переменных. Динамической называют переменную, имя которой хранится в ней самой.

В RНР возможно обращение к одной и той же переменной с использованием различных имен. Для реализации этого используются ссылки. Ссылки позволяют двум или большему количеству переменных ссылаться на одну и ту же область памяти.

Операторы

Операторы RНР напоминают общеизвестные операторы языка Си.

Унарные операторы

- Изменение знака на противоположный

! Дополнение. Используется для реверсирования значения логических переменных

++ Увеличение значения переменной. Может применяться и как префикс, и как суффикс

-- Уменьшение значения переменной. Может применяться и как префикс, и как суффикс

Арифметические операторы

- Вычитание

+ Сложение

* Умножение

/ Деление

% Остаток от деления

Оператор конкатенации

Оператор конкатенации присоединяет правую строку к левой.

Операторы присваивания

= Присваивание

+= Сложение (\$n += 777; аналогично \$n = \$n + 777;)

-= Вычитание (\$n -= 777; аналогично \$n = \$n - 777;)

*= Умножение

/= Деление

%= Остаток от деления

.= Конкатенация (\$n .= "777"; аналогично \$n = \$n."777";)

Битовые операторы

Битовые операторы позволяют изменять отдельные биты целых чисел.

& И

| ИЛИ

^ Исключающее ИЛИ

~ Инверсия

>> Сдвиг вправо

<< Сдвиг влево

Операторы сравнения

> Больше (Больше ли первое значение, чем второе?)

>= Больше или равно (Верно ли, что первое значение не меньше второго?)

< Меньше (Меньше ли первое значение, чем второе?)

<= Меньше или равно (Верно ли, что первое значение не больше второго?)

== Равно (Равнозначны ли значения двух переменных?)

=== Идентично (Одинаковы ли как значения, так и типы двух переменных?)

!= , <> Не равно (Не равны ли значения двух переменных?)

!== Не идентично (Не одинаковы ли значения или типы данных двух переменных?)

Логические операторы

Логические операторы отличаются от битовых тем, что работают не с числами, а с логическими значениями: TRUE и FALSE.

and И

or ИЛИ

xor Исключающее ИЛИ

! Инверсия

>> Сдвиг вправо

<< Сдвиг влево

&& И

|| ИЛИ

Логические операторы "И" и "ИЛИ" имеют два формата. Это не синонимы. Дело в том, что оператор `or` имеет приоритет ниже, чем `||`, а `and` - ниже, чем `&&`. Таким образом, при построении сложных условных выражений можно обойтись без скобок. Однако, в таком способе указания порядка проще и запутаться.

Проверка содержимого переменной

Иногда необходимо проверить, существует ли переменная или какое она имеет значение. Ниже приведены функции, позволяющие выполнить такие действия. `isset($имя_переменной)` #Истина, если переменная объявлена даже без присваивания значения.

`empty($имя_переменной)` #Истина, если значение переменной равно нулю или пустой строке, либо переменная не объявлена.

PHP также позволяет проверить тип переменной. Например, для того чтобы проверить, является ли переменная целочисленной, следует воспользоваться функцией `is_int($number)`.

Результатом выполнения этой функции является TRUE, если переменная `$number` имеет тип `integer`. Рассмотрим подобные функции.

`is_array ($var2)` проверяет, является ли переменная `$var2` массивом.

`is_float ($number)` проверяет, является ли переменная `$number` числом с плавающей точкой.

`is_null ($var1)` проверяет, равно ли значение переменной `$var1` нулю

`is_numeric($string)` проверяет, является ли переменная `$string` числовой строкой.

`is_string ($string)` проверяет, является ли переменная `$string` строкой.

Для проверки обратных условий следует воспользоваться символом восклицания (!). Например, при обработке следующего выражения будет получено значение TRUE, если переменная не объявлена: `! isset($имя_переменной)`

3. СТИЛЕВОЕ ОФОРМЛЕНИЕ HTML-ДОКУМЕНТОВ

Сегодня мы будем знакомиться со стилевыми таблицами. HTML код задает базовую структуру документа, а с помощью стилей можно задавать различные варианты оформления веб-страницы. Разберемся в чем разница между внутренними, внешними и глобальными стилями. Подробно рассмотрим, как они применяются и узнаем об иерархии каскадных приоритетов. Также рассмотрим правила написания стилей и познакомимся с определением классов. В следующем шаге мы научимся форматировать текст с помощью стилей и узнаем основные стилевые свойства. А также уделим внимание способам оформления ссылок, меняющим их внешний вид.

Стилевые

таблицы

Как вы уже знаете, HTML разрабатывался как язык разметки страниц, задающий базовую структуру документа (абзацы, заголовки, таблицы итд). При этом в HTML уже содержались такие оформительские элементы, как `bcolor`, `font size` или `align`. В далеком 1996 году консорциум W3C впервые сформулировал идею использования каскадных стилевых таблиц (Cascading Style Sheets — CSS) для форматирования HTML документов. Данная рекомендация, более детализированная в 1998 году, позволяла разработчикам разделить коды структуры веб-страниц и коды оформления документа.

Стилевые таблицы — это набор правил, определяющих способ применения стилей к тегам HTML документа. В данных рекомендациях описывались перечисленные ниже три типа стилевых таблиц.

Внутренние

стили

Содержатся в теле веб-страницы в качестве атрибутов HTML элементов. Каждому тегу назначаются свои собственные стили по мере их определения.

Глобальные

стили

Свойства стиля описываются (внутри тега `<style>`) в верхней части HTML документа. Стил, назначенный какому-то тегу, применяется ко всем прочим тегам этого же типа в пределах одной веб-страницы.

Внешние

стили

Информация о свойствах стилей сохраняется в отдельном файле. Этот файл может быть подключен к любому HTML документу с помощью тега `<link>`, помещаемого внутри тега `<head>`.

Даже если не учитывать преимущества, предоставляемые стилевыми таблицами при форматировании документов, их использование, вне всяких сомнений, позволит вам, как веб-дизайнеру, сэкономить массу рабочего времени и усилий. Поскольку уже созданные стилевые таблицы можно применять к произвольному количеству HTML документов, последующее внесение корректирующих изменений будет выполняться в течение минут, а не часов или целых дней.

Если не применять стилевые таблицы, то для изменения внешнего вида какого-то отдельного тега веб-сайта пришлось бы открывать каждый документ, находить в нем этот тег, вносить в него изменения, сохранять документ, переходить к следующему документу и т.д. С другой стороны, можно изменить этот тег только в одном документе, где описаны стилевые таблицы, и эти изменения сразу же вступят в силу для всех страниц, к которым подключен документ.

Определение

правил

Правила стилевых таблиц состоят из селекторов (тегов HTML, для которых создается стиль) и описаний (свойств стилевых таблиц и их значений). В приведенном далее примере селектором является тег `<body>`, а описание состоит из стилевого свойства (`background`) и его значения (`black`). В данном случае для всего документа в качестве фонового устанавливается черный цвет. `body {background:black}`

Как видите, в стилевой таблице тег HTML не заключается в угловые скобки (как это происходит обычно при добавлении его в HTML документ), а описание окружают фигурные скобки. В описании может содержаться более одного свойства. Так, в следующем примере

цвет текста для данной страницы устанавливается белым. Обратите внимание, что в описании одно свойство от другого отделяется точкой с запятой. `body {background:black; color:white}`

Если необходимо применить одни и те же правила сразу к нескольким тегам HTML, можете сделать это одновременно, как показано ниже.

```
body, td, hi {  
background:black;  
color:white }
```

Определение

классов

Как известно, правила существуют для того, чтобы их нарушать. Что, если вы не хотите, чтобы все добавляемые в документе заголовки первого уровня, задаваемого с помощью тега `<h1>`, отображались белым цветом на черном фоне? Может необходимо, чтобы каждый второй заголовок этого уровня отображался синим цветом на белом фоне? Что ж, справиться с данной задачей можно с помощью атрибута `class`. Этот атрибут можно задавать практически для любого HTML тега, и результат такого действия сравним с созданием собственных тегов.

Например, нужно сделать, чтобы одна таблица была с голубым фоном и выравниванием по левому краю, а другая таблица — с желтым фоном и выравниванием по центру. Тогда в теге `<style>` в шапке документа прописываются сразу два стиля. За именем HTML тега `table` следует точка (.) и далее имя класса (`nav` или `rest`).

```
<style type=text/css>  
table.nav {background:aqua}  
table.rest {background:yellow; text-align:center; color:black}  
</style>
```

При добавлении таблицы в теле документа необходимо задать значение атрибута `class`, чтобы сообщить браузеру, какие стилевые свойства должны быть к этой таблице применены. Коды добавления на веб-страницу обеих таблиц приведены ниже. Обратите внимание, что названия классов указываются в кавычках, подобно значениям любых других атрибутов (например, как значения используемого здесь же атрибута `width`).

```
<table class="nav" width="100%">  
<table class="rest" width="50%">
```

Применение

стилей

Прежде чем двигаться далее, вкратце рассмотрим процедуру применения стилевых свойств к элементам HTML документов. Как вы уже знаете, существуют три метода добавления стилевых таблиц: внутренние, глобальные и внешние стили. Остановимся на каждом из них в отдельности.

Глобальные

стили

Все стили данного типа определяются в верхней части HTML документа внутри тега `<head>`, поскольку они содержат описания, применяемые ко всей странице. Действие определяемых таким способом стилей ограничивается пределами текущего документа. Если необходимо использовать эти же стили в каком-то другом HTML документе, вам придется непосредственно добавить их определения в этот документ. Тер `<style>` почти всегда включает в себя атрибут `text=и text/css`. Запомните это и не забывайте его добавлять.

```
<head>  
<style type="text/css">  
table.nav {background:aqua}  
table.rest {background:yellow; text-align:center; color:black}  
a:link {color:red; text-decoration:none}  
</style>  
</head>
```

Внешние

стили

При использовании внешних стилевых таблиц все описания стилей содержатся в отдельном файле. Этот файл подключается ко всем HTML документам, где заданные в нем стилевые свойства должны использоваться.

```
<head>  
<link rel="stylesheet" href="mystyles.css" type="text/css">
```

</head>

В данном примере был создан отдельный файл для каскадных стилевых таблиц, содержащий описания всех стилевых свойств. Для него было задано имя `mystyles.css`. Обратите внимание, что здесь также присутствует атрибут `type="text/css"`. Далее приведено содержание файла `mystyles.css`. Как видите, здесь описаны те же самые стили, что и в предыдущем примере с глобальными стилевыми таблицами, однако в данном случае они вынесены в отдельный текстовый файл.

```
table.nav {background:aqua}
table.rest {background:yellow; text-align:center; color:black}
a:link {color:red; text-decoration:none}
```

Внутренние

стили

При использовании данного метода стилевые свойства применяются к тем тегам HTML, внутри которых они определяются. Например, если вы хотите с помощью этого метода применить один и тот же стиль оформления ко всем заголовкам первого уровня, необходимо набрать их описание внутри всех тегов `<h1>` данного документа. Рассмотрим это на следующем примере. В данном случае используются те же стилевые свойства, что и в предыдущих примерах, однако теперь каждый стиль описывается внутри тега своей таблицы.

```
<table style="background:aqua" width="100%">
<table style="background:yellow; text-align:center; color:black" width=50%>
```

При использовании метода внутритекстового определения стилей тег `<style>` превращается в атрибут `style`. Разные стилевые свойства по-прежнему отделяются друг от друга точкой с запятой, однако вся группа свойств, относящихся к какому-то конкретному тегу, описывается внутри него. Этот метод удобен в тех случаях, когда требуется применить какие-то специальные стили оформления к одному-двум элементам веб-страницы, однако вы вряд ли захотите его использовать для добавления множества стилей по всему документу.

Каскадный

приоритет

Прежде чем двигаться дальше, необходимо прояснить еще один момент. Для браузера стили этих трех типов не эквивалентны друг другу. Браузер устанавливает более высокий приоритет для стиля, который является ближайшим к конкретному тегу. Таким образом, внутренний стиль (определяемый как атрибут внутри самого тега) всегда имеет наивысший приоритет. Следующими применяются глобальные стили (определяемые в верхней части HTML документа); и, наконец, внешние стили (вообще задаваемые в отдельном файле) применяются в последнюю очередь.

Форматирование текста с помощью стилей

Текст является наиболее важным компонентом любой веб-страницы. Если на веб-странице вообще не будет никакой текстовой информации, посетители вряд ли смогут извлечь из нее какую-либо пользу. Текст в HTML документы добавляется с помощью тегов `<body>`, `<p>`, `<td>`, `<tr>`, `<th>`, `<h1>...<h6>`, `<i>` и др. Для текста каждого из этих тегов можно задавать собственные параметры форматирования, используя стилевые свойства, перечисленные ниже.

Если вы сами или посетители веб-страницы не измените заданных по умолчанию настроек браузера, обычный текст HTML документа будет воспроизводиться шрифтом Times New Roman размером 12 пунктов.

`background` — Задает фоновый цвет текста;

`color` — Задает цвет самого текста;

`font-family` — Задает шрифт;

`font-size` — Указывает размер шрифта;

`font-style` — Определяет стиль начертания текста; может принимать значения `normal` (по умолчанию) или `italic`;

`font-weight` — Принимает значения от `extra-light` до `extra-bold`;

`text-align` — Определяет способ выравнивания текста; может принимать значения `left`, `right`, `center` или `justify`;

`text-indent` — Задает отступ текста; величина отступа может быть задана как фиксированное

значение или в процентном выражении;

text-decoration — Принимает значения underline, overline, strikethrough и none.

Стили

оформления

ссылок

Наверное, вы уже привыкли к тому, что на веб-страницах ссылки обычно оформляются как подчеркнутый текст синего цвета. Далее перечислены предусмотренные для стилевых таблиц селекторы, позволяющие изменить внешний вид ссылок.

a:link — Задаёт стиль оформления ещё не использованных ссылок;

a:visited — Задаёт стиль оформления уже использованных ссылок;

a:active — Задаёт стиль оформления активных в данный момент ссылок;

a:hover — Задаёт стиль оформления ссылки, на которую наведён указатель мыши.

Для ссылок, добавляемых на веб-страницы можно использовать следующие стилевые свойства: background-color (цвет фона для ссылки), color, font-family, text-decoration. В последнее время один из наиболее популярных эффектов стилевых таблиц состоит в удалении для ссылок линии подчёркивания. Чтобы сделать это, просто добавьте декларацию text-decoration:none к стилям a, как показано на примере следующего кода.

```
a:link {color:red; text-decoration:none}
```

Если вы хотите видеть свои ссылки подчеркнутыми, никаких специальных действий выполнять не нужно. Подчёркивание и так по умолчанию задано для всех стилей a.