Оглавление

[1. Введение 2](#_Toc420653387)

[2 Общая часть 3](#_Toc420653388)

[2.1 Цель разработки 3](#_Toc420653389)

[2.2 Анализ использования разработки 3](#_Toc420653390)

[2.3 Анализ методов решения проблемы 3](#_Toc420653391)

[2.4 Описание среды разработки 6](#_Toc420653392)

[2.4.1 Описание языка HTML 6](#_Toc420653393)

[2.4.1.1Понятие HTML 6](#_Toc420653394)

[2.4.1.2 История 6](#_Toc420653395)

[2.4.1.3 Браузеры 8](#_Toc420653396)

[2.4.1.4 Общее представление 9](#_Toc420653397)

[2.4.1.5 Новые возможности HTML 9](#_Toc420653398)

[2.4.1.6 Тэги 10](#_Toc420653399)

[2.4.1.7 Структура HTML-документа 12](#_Toc420653400)

[2.4.1.8 Основные тэги. 14](#_Toc420653401)

[2.4.2.9 О возможных ошибках 18](#_Toc420653402)

[2.4.2. Описание языка PHP 19](#_Toc420653403)

[2.4.2.1 Понятие языка PHP 19](#_Toc420653404)

[2.4.2.2 История языка PHP 20](#_Toc420653405)

[2.4.2.3 Область применения PHP 21](#_Toc420653406)

[2.4.2.4 Возможности языка PHP 22](#_Toc420653407)

[2.4.2.5 Типы данных в PHP 23](#_Toc420653408)

[2.4.2.6 Общие правила построения PHP-программы 23](#_Toc420653409)

[2.4.2.7 Управляющие конструкции языка PHP 24](#_Toc420653410)

[2.4.2.8 Синтаксис 24](#_Toc420653411)

[2.4.2.9 Описание типов данных PHP 25](#_Toc420653412)

[2.4.2.10 PHP и HTML 27](#_Toc420653413)

[3 Специальная часть 28](#_Toc420653414)

[3.1 Постановка задачи 28](#_Toc420653415)

[3.2 Назначение задачи 28](#_Toc420653416)

[3.3 Требования к программе 28](#_Toc420653417)

[3.4 Описание алгоритма 29](#_Toc420653418)

[3.5 Оценка результатов выполнения задачи 38](#_Toc420653419)

[4 Организация производства 38](#_Toc420653420)

[4.1 Организация работ по внедрению программного продукта заказчика 38](#_Toc420653421)

[5 Заключение 39](#_Toc420653422)

[5.1 Перечень использованной литературы 39](#_Toc420653423)

# 1. Введение

Учреждения среднего профессионального образования позволяют приобрести студенту определенный круг ключевых профессиональных компетенций, которые обеспечат ему эффективное поведение и творческую профессиональную деятельность в постоянно развивающихся социально-экономических условиях. Одной из технологий, способствующей объективной оценке сформированных компетенций обучающихся, является технология портфолио. Федеральные государственные образовательные стандарты нового поколения предполагают, что в результате изучения профессионального модуля обучающийся должен иметь практический опыт оформления портфолио педагогических достижений. **Электронное портфолио** – это совокупность студенческих работ и документов, собранных с применением электронных средств и носителей, представленных в виде либо компакт дисков в форме CD – ROM / DVD, либо в виде Web – сайта (онлайн – портфолио).

При разработке элементов электронного портфолио студенты должны придерживаться следующих принципов:

1.Самооценка результатов (промежуточных, итоговых) овладения определёнными видами самостоятельной познавательной, исследовательской, рефлексивной, коммуникативной и творческой деятельности.

2.Систематичность и регулярность самомониторинга (студент систематично отслеживает результаты своей деятельности по избранной специальности, отбирает наиболее интересные работы в своё портфолио, организует их в предусмотренную структуру на компетентностной основе).

3.Структуризация и логичность материалов, представленных в портфолио.

4.Целостность, тематическая завершенность материалов.

5.Наглядность и обоснованность презентации портфолио студента.

6. Адекватность структуры портфолио формируемым компетенциям.

7. Доступность электронного обеспечения.

8.Технологичность электронного обеспечения.

Создание портфолио – творческий процесс, позволяющий учитывать результаты, достигнутые студентом в разнообразных видах деятельности (учебной, творческой, социальной, коммуникативной) за время обучения в техникуме.

Портфолио создаётся в течение всего периода обучения. Завершается его формирование публичной защитой или презентацией.

В широком смысле слова портфолио – это метод фиксирования, накопления и оценки индивидуальных достижений студента за период его обучения. Портфолио дополняет традиционные контрольно-оценочные средства, метод направлен на оценивание комплексной подготовки студента к профессиональной деятельности. Портфолио студента является важным элементом практико-ориентированного, деятельностного, компетентностного подхода к профессиональному образованию.

Наиболее удобным и предпочтительным вариантом реализации электронного портфолио представляется формат собственного небольшого веб-сайта.

2 Общая часть

**2.1 Цель разработки**

Цель дипломного проекта: разработка макета электронного портфолио для студентов.

**2.2 Анализ использования разработки**

Данный макет электронного портфолио позволит студентам быстро и эффективно заполнить информацию, демонстрирующую поведение и творческую профессиональную деятельность в постоянно развивающихся социально-экономических условиях. Для образовательного учреждения СПО введение портфолио может повысить образовательную активность студентов, уровень осознания ими своих целей и возможностей.

Портфолио ставит своей целью демонстрацию процесса студента, его учебные достижения относительно ФГОС СПО нового поколения, умение размышлять над своими успехами неудачами. Это своеобразный отчет по различным видам деятельности студента: учебная, научно-исследовательская, творческая, практическая, общественная и т.д.

## 2.3 Анализ методов решения проблемы

Для решения поставленной задачи можно использовать разные программы и языки программирования, такие как:

* HTML
* PHP
* Microsoft Access

HTML

HTML ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) HyperText Markup Language  — «язык [гипертекстовой](https://ru.wikipedia.org/wiki/%D0%93%D0%B8%D0%BF%D0%B5%D1%80%D1%82%D0%B5%D0%BA%D1%81%D1%82) разметки») — стандартный [язык разметки](https://ru.wikipedia.org/wiki/%D0%AF%D0%B7%D1%8B%D0%BA_%D1%80%D0%B0%D0%B7%D0%BC%D0%B5%D1%82%D0%BA%D0%B8) документов во [Всемирной паутине](https://ru.wikipedia.org/wiki/%D0%92%D1%81%D0%B5%D0%BC%D0%B8%D1%80%D0%BD%D0%B0%D1%8F_%D0%BF%D0%B0%D1%83%D1%82%D0%B8%D0%BD%D0%B0). Большинство[веб-страниц](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D1%82%D1%80%D0%B0%D0%BD%D0%B8%D1%86%D0%B0) содержат описание разметки на языке HTML (или [XHTML](https://ru.wikipedia.org/wiki/XHTML)). Язык HTML интерпретируется [браузерами](https://ru.wikipedia.org/wiki/%D0%91%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80), полученный в результате интерпретации форматированный текст отображается на экране монитора компьютера или мобильного устройства.

Язык HTML является приложением [SGML](https://ru.wikipedia.org/wiki/SGML) (стандартного обобщённого языка разметки) и соответствует международному стандарту [ISO](https://ru.wikipedia.org/wiki/ISO) 8879.

Язык [XHTML](https://ru.wikipedia.org/wiki/XHTML) является более строгим вариантом HTML, он следует всем ограничениям [XML](https://ru.wikipedia.org/wiki/XML) и, фактически, XHTML можно воспринимать как приложение языка XML к области разметки гипертекста.

Во всемирной паутине HTML-страницы, как правило, передаются браузерам от сервера по протоколам [HTTP](https://ru.wikipedia.org/wiki/HTTP) или [HTTPS](https://ru.wikipedia.org/wiki/HTTPS), в виде простого текста или с использованием [шифрования](https://ru.wikipedia.org/wiki/%D0%A8%D0%B8%D1%84%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5).

PHP

PHP (англ. PHP: HypertextPreprocessor — «PHP: препроцессор гипертекста»; первоначально Personal Home PageTools — «Инструменты для создания персональных веб-страниц»; произносится пи-эйч-пи) — скриптовый язык программирования общего назначения, интенсивно применяемый для разработки веб-приложений. В настоящее время поддерживается подавляющим большинством хостинг-провайдеров и является одним из лидеров среди языков программирования, применяющихся для создания динамических веб-сайтов.

Язык и его интерпретатор разрабатываются группой энтузиастов в рамках проекта с открытым кодом. Проект распространяется под собственной лицензией, несовместимой с GNU GPL.

Microsoft Access

Microsoft Access — программа управляющая базами данных, которые обладают определенными свойствами и позволяют проводить обработку внесенных данных. Удобство использования программы обусловлено ее практическим применением в сфере бизнеса и группирования информации.

Гибкая система, интегрированная в мощную платформу, дает возможность использования расширенного инструментария для обработки данных. Microsoft Access разработана для ОС Microsoft Windows, как функциональная СУБД с гибкой средой. В программе реализуется принцип внесения информации в различные типы полей, которые предназначены для выполнения определенных действий.

Самое впечатляющее нововведение Access 2007 - новый пользовательский интерфейс программы. Но новый внешний вид - не единственное усовершенствование. Когда корпорация Microsoft решила переработать программу Access 2007. она сформировала команду разработчиков в семь раз больше той, которая разрабатывала Access 2003. Этот дополнительный человеческий ресурс помог внести ряд долгожданных изменений.

Далее перечислены основные из них:

1. Новый усовершенствованный процессор баз данных (database engine). Многолетние профессиональные пользователи программы Access знают, что программа использует за кадром процессор Jet для управления операциями (такими как добавление и обновление данных). У Access 2007 теперь есть собственная специализированная версия Jet, позволившая создателям Access добавить новые функциональные возможности и применить тонкую настройку рабочих характеристик программы. Не беспокойтесь, программа Access 2007 на 100% обратно совместима с предыдущими версиями программы. Это означает, что вы можете продолжать использовать базы данных старого стиля,созданные в Access 2003, но при этом не сможете применять новые возможности (например, вложения или сложные данные, описанные далее) до тех пор, пока не создадите файл базы данных нового формата.
2. Вложения (Attachments). Одна из самых замечательных функциональных возможностей Access - тип данных Вложение (Attachment), позволяющий хранить целые файлы в вашей базе данных, включая изображения, документы и электронные таблицы. (Не стоит использовать видео, аудио и другие объемистые медиафайлы, т. к. размер любой базы данных Access ограничен 2 Гбайт.)
3. Легко конструируемые формы и отчеты. Создание привлекательной формы для ввода данных или отчета для вывода на принтер всегда требовало множества рутинных операций. В Access 2007 сформировать их гораздо легче благодаря автоматической верстке - функциональной возможности, группирующей связанные порции данных в единое целое в виде подходящих столбцов или таблиц. В программе есть и новый Режим макета (Layout), позволяющий вставить форматирование и тут же увидеть результат.
4. Сложные данные (Complex data). Это необязательное новое свойство, позволяющее хранить в одном поле несколько значений (или «сегмент» данных). Сложные данные облегчают процесс связывания таблиц. Например, с помощью сложных данных вы можете связать несколько авторов с одной книгой. Главная причина введения сложных данных - поддержка функции объединения сервисов SharePoint.

В данном дипломном проекте для создания макета электронного портфолио для студентов были использованы следующие средства для создания сайта: язык программирования PHP и язык гипертекстовой разметки HTML**.**Эти языки относятся к категории лицензионного программного обеспечения, просты и удобны в использовании.

## 2.4 Описание среды разработки

#### 2.4.1 Описание языка HTML

#### 2.4.1.1Понятие HTML

*HTML* (язык гипертекстовой разметки) — это основной язык, с помощью которого создаются веб-страницы. Страница, которую пользователь видит в своем браузере, может состоять из множества разных файлов — например, изображений, анимационных роликов, сценариев JavaScript, апплетов и т. д. - но основой страницы практически всегда является документ *HTML*. Другие языки разметки (прежде всего многочисленные приложения XML) пока еще слабо поддерживаются браузерами и поэтому не вытесняют *HTML* на компьютерах обычных пользователей.



Рис. 1. Логотип языка HTML

**2.4.1.2 История**

Начало истории языка **HTML** было положено в 1986 году, когда Международной организацией по стандартизации (ISO) был принят стандарт ISO-8879. Он носил названиеStandard Generalized Markup Language(Стандартный Обобщённый Язык Разметки). **SGML** представлял из себя обобщенный метаязык, позволяющий строить системы логической, структурной разметки любых разновидностей текстов. При этом управляющие коды или теги не несли никакой информации о внешнем виде документа, а только задавали его логическую структуру.

Благодаря таким ограничениям, размеченный текст без труда интерпретировался любой компьютерной программой и любым устройством вывода. При этом вид и размер текста задавался исключительно настройками последних. Все это обеспечило небывалую совместимость и универсальность SGML. SGML получил широкое признание и стал активно использоваться в больших проектах. Но в целом этот язык был громоздок и труден для изучения. Главный побочный эффект SGML заключался в том, что он пытался сделать слишком много и быть чересчур гибким.

В 1989 году сотрудник Европейского института физики частиц (CERN) ***Тим Бернерс-Ли***(Tim Berners-Lee) выдвинул предложение о создании Системы гипертекстовых документов. В 1990 году он назвал ее ***World Wide Web***(Всемирная паутина). Одной из составляющих системы был язык гипертекстовой разметки. Его созданием Бернерс-Ли занялся в 1990 году, когда разрабатывал первый браузер — программу, позволяющую просматривать гипертекстовые документы.

  В 1991 году группой Бернерс-Ли, работавшей над созданием системы передачи гипертекстовой информации через Интернет, было принято решение, выбрать SGML в качестве основы для нового языка разметки гипертекстовых документов. Этот язык был назван ***HTML*** (Hyper Text Markup Language, "язык разметки гипертекста").

HTML разделял все особенности идеологии SGML, т.е. подразумевалась только логическая разметка текста. Например, в ***HTML версии 1.2*** (июнь 1993 г.) присутствовало около 40 тегов. И только три из них отвечали за физические параметры отображения документа (но эти теги не рекомендовались к использованию). А первым графическим браузером в то время была программа***Mosaic***, разработанная в Национальном центре суперкомпьютерных приложений.

  В апреле 1994 г. началась подготовка спецификации следующей версии языка - 2.0. Этим уже занимался ***Консорциум W3***(W3C), унаследовавший от CERN власть и авторитет в мире ***WWW***. В том же году было решено разбить язык HTML, на уровни. Это было сделано, чтобы обеспечить обратную совместимость версий: каждый новый уровень непременно включал в себя предыдущие. Было выделено четыре уровня:

**•  Уровень 0** — обязательный для поддержки всеми браузерами. Включает в себя заголовки, ссылки, списки.

**• Уровень 1** — добавляются рисунки и элементы выделения текста (полужирным, курсивом, подчеркнутым).

**• Уровень 2** — добавляются формы, позволяющие пользователю вводить информацию.

**• Уровень 3** — добавляются таблицы, позволяющие размещать на страницах информацию табличного характера и, кроме этого, точно задавать положение объектов на странице.

Язык ***HTML 2.0*** (1995 г.) является, например, языком второго уровня, с его помощью нельзя верстать страницы со сложным дизайном, так как это невозможно без таблиц.

Самым большим прорывом явился ***HTML 3.2***. Он включил в себя такие нововведения, как теги для создания таблиц, вставки рисунков, примечаний, разметку математических формул и др. Наиболее важным нововведением явились ***CSS*** (таблицы стилей), которые позволяли задавать единый вид всему сайту. Они имели свою структуру, свой синтаксис и применялись вместе с HTML.

Примечательна ***история HTML 3.2***. Это скорее история первых браузеров. С созданием WWW практически сразу началось его коммерческое освоение. В начале 1994 года группа разработчиков браузера Mosaic под предводительством Джеймса Кларка основала корпорацию Netscape Communications и вскоре выпустила первую версию коммерческого браузера ***Netscape***. Спрос на него, при отсутствии альтернативы, превысил все ожидания и сделал Netscape к концу 1995 года самой быстрорастущей компанией в мировой истории.

Чтобы закрепить лидерство Netscape Communications вводила в HTML все новые и новые усовершенствования. И эти усовершенствования поддерживались только браузером Netscape. Практически все новые теги, предложенные Netscape, были направлены на улучшение внешнего вида документа и расширение возможностей его форматирования. Такая политика компании принесла ей впечатляющий успех. Девять из десяти используемых в то время браузеров были версии **Netscape Navigator**. И продавались они миллионами штук.

Компания ***Microsoft*** изначально не предавала серьезного значения коммерческим перспективам WWW. Однако невероятный взлет Netscape заставил Microsoft изменить свое мнение. Летом 1996 года на свет появился браузер Internet Explorer 3.0, который поддерживал почти все расширения Netscape. Одновременно с разработкой конкурентного браузера Microsoft навела порядок в мире HTML, взяв под свою опеку Консорциум W3. В итоге был создан стандарт HTML 3.2, который, по сути, всего лишь описывал большинство расширений Netscape.

К концу 1996 года практически все браузеры поддерживали HTML 3.2, и благодаря этому Web-дизайн испытал небывалый взлет. Появилась возможность проектировать и отображать на экране сложные композиции графических элементов, ничем не уступающие печатным изданиям. Это положило начало эре Web-дизайна.

В 1997 году появилась спецификация языка ***HTML 4.0***. Она включала поддержку фреймов, унифицированную процедуру вставки различных объектов, поддержку каскадных таблиц стилей (CSS). Кроме того, были усовершенствованы формы и таблицы.

Версия ***HTML 4.01*** стала стандартом в 1999 году и господствует в мире WWW уже более 10 лет. Но ее очень скоро потеснит ***HTML 5.0*** – новая версия языка гипертекстовой разметки. Над разработкой HTML 5.0 трудится рабочая группаW3C. В эту группу входят представители таких компаний, как Microsoft, Opera, Mozilla, Google, IBM, Apple и другие. Новый стандарт позволит более эффективно управлять мультимедийным содержимым. Также будет улучшена совместимость с новыми языками web-программирования. И наконец, появятся новые теги, расширяющие возможности web-дизайна.

**2.4.1.3 Браузеры**

Текстовые документы, содержащие разметку на языке HTML (такие документы традиционно имеют [расширение](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D1%81%D1%88%D0%B8%D1%80%D0%B5%D0%BD%D0%B8%D0%B5_%D0%B8%D0%BC%D0%B5%D0%BD%D0%B8_%D1%84%D0%B0%D0%B9%D0%BB%D0%B0) .html или .htm), обрабатываются специальными [приложениями](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B8%D0%BA%D0%BB%D0%B0%D0%B4%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5), которые отображают документ в его форматированном виде. Такие приложения, называемые «браузерами» или «интернет-обозревателями», обычно предоставляют пользователю удобный [интерфейс](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D0%BA%D0%B8%D0%B9_%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81) для запроса [веб-страниц](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D1%82%D1%80%D0%B0%D0%BD%D0%B8%D1%86%D0%B0), их просмотра (и вывода на иные внешние устройства) и, при необходимости, отправки введённых пользователем данных на сервер. Наиболее популярными на сегодняшний день браузерами являются [Google Chrome](https://ru.wikipedia.org/wiki/Google_Chrome), [Mozilla Firefox](https://ru.wikipedia.org/wiki/Mozilla_Firefox), [Opera](https://ru.wikipedia.org/wiki/Opera), [Internet Explorer](https://ru.wikipedia.org/wiki/Internet_Explorer) и [Safari](https://ru.wikipedia.org/wiki/Safari) (см.: [Браузер#Рыночные доли](https://ru.wikipedia.org/wiki/%D0%91%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80#.D0.A0.D1.8B.D0.BD.D0.BE.D1.87.D0.BD.D1.8B.D0.B5_.D0.B4.D0.BE.D0.BB.D0.B8)).

**2.4.1.4 Общее представление**

Язык HTML был разработан [британским](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%BB%D0%B8%D0%BA%D0%BE%D0%B1%D1%80%D0%B8%D1%82%D0%B0%D0%BD%D0%B8%D1%8F) учёным [Тимом Бернерсом-Ли](https://ru.wikipedia.org/wiki/%D0%91%D0%B5%D1%80%D0%BD%D0%B5%D1%80%D1%81-%D0%9B%D0%B8,_%D0%A2%D0%B8%D0%BC) приблизительно в [1986](https://ru.wikipedia.org/wiki/1986_%D0%B3%D0%BE%D0%B4)—[1991 годах](https://ru.wikipedia.org/wiki/1991_%D0%B3%D0%BE%D0%B4) в стенах [ЦЕРНа](https://ru.wikipedia.org/wiki/%D0%A6%D0%95%D0%A0%D0%9D) в [Женеве](https://ru.wikipedia.org/wiki/%D0%96%D0%B5%D0%BD%D0%B5%D0%B2%D0%B0) в [Швейцарии](https://ru.wikipedia.org/wiki/%D0%A8%D0%B2%D0%B5%D0%B9%D1%86%D0%B0%D1%80%D0%B8%D1%8F). HTML создавался как язык для обмена научной и технической документацией, пригодный для использования людьми, не являющимися специалистами в области [вёрстки](https://ru.wikipedia.org/wiki/%D0%92%D1%91%D1%80%D1%81%D1%82%D0%BA%D0%B0). HTML успешно справлялся с проблемой сложности SGML путём определения небольшого набора структурных и [семантических](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D0%BC%D0%B0%D0%BD%D1%82%D0%B8%D0%BA%D0%B0) элементов — дескрипторов. Дескрипторы также часто называют «[тегами](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%B3_(%D1%8F%D0%B7%D1%8B%D0%BA%D0%B8_%D1%80%D0%B0%D0%B7%D0%BC%D0%B5%D1%82%D0%BA%D0%B8))». С помощью HTML можно легко создать относительно простой, но красиво оформленный документ. Помимо упрощения структуры документа, в HTML внесена поддержка [гипертекста](https://ru.wikipedia.org/wiki/%D0%93%D0%B8%D0%BF%D0%B5%D1%80%D1%82%D0%B5%D0%BA%D1%81%D1%82). [Мультимедийные](https://ru.wikipedia.org/wiki/%D0%9C%D1%83%D0%BB%D1%8C%D1%82%D0%B8%D0%BC%D0%B5%D0%B4%D0%B8%D0%B0) возможности были добавлены позже.

Изначально язык HTML был задуман и создан как средство структурирования и форматирования документов без их привязки к средствам воспроизведения (отображения). В идеале, текст с разметкой HTML должен был без стилистических и структурных искажений воспроизводиться на оборудовании с различной технической оснащённостью (цветной экран современного компьютера, монохромный экран органайзера, ограниченный по размерам экран мобильного телефона или устройства и программы голосового воспроизведения текстов). Однако современное применение HTML очень далеко от его изначальной задачи. Например, тег <TABLE> предназначен для создания в документах таблиц, но часто используется и для оформления размещения элементов на странице. С течением времени основная идея платформонезависимости языка HTML была принесена в жертву современным потребностям в мультимедийном и графическом оформлении.

**2.4.1.5 Новые возможности HTML**

Как уже упоминалось выше на сегодняшний день существует два законченных стандарта языка HTML: HTML 1.0 и HTML 2.0. Для дальнейшего развития этого языка было предложено несколько путей. Первый из них - это HTML+, который является базой для следующей версии HTML - HTML 3.0. Именно в этом варианте языка были определены таблицы, стили и математические формулы. Концептуальным интерфейсом для HTML 3.0 является программа Arena, главное назначение которой - продемонстрировать возможности HTML 3.0. Другим направлением развития HTML стали, так называемы Netscape Extensions. Первоначально эта спецификация развивала синтаксис HTML 2.0 в сторону более выразительных средств представления документов (возможность смены фона, выравнивание текста, обтекание графики текстом и т.п.). Netscape Extensions были предложены компанией Netscape Communication и реализованы в программе Netscape Navigator версии 1.1 и старше. Фактически, если HTML+ был направлен на расширение HTML в сторону реализации публикаций научно технического характера, то Netscape Extensions развивали HTML в сторону рекламно-издательской деятельности. Это направление Netscape Communication продолжила и в следующей версии Navigator. В Navigator 2.0 и Navigator Gold 2.0 были добавлены две принципиально новые возможности разработки Web-страниц: фреймы и JavaScript. Заявлена также и поддержка Java, но об этом языке как о концептуальном развитии технологии лучше говорить в контексте другой программы - HotJava компании Sun Microsystems. Не отстает на поприще развития стандартов Web и Microsoft. В настоящее время для пользователей Internet доступны свободнораспростроняемые сервер и интерфейс Internet Explorer. Важным здесь является тот факт, что Microsoft не просто путается поддерживать то: что предлагают другие: но и сама предлагает новые расширения языка. Они касаются главным образом возможностей разработки мультимедийных страниц и являются в некотором смысле альтернативой тому, что предлагают разработчики Java и JavaScript. Таким образом в настоящее время следует выделить следующие основные направления развития технологии Web: HTML 3.0 (HTML+), HTML 2.0 Netscape Extensions, JavaScript, Java, Microsoft Extensions.

На самом деле, чуть в стороне от основных направлений развития Web-технологии стоит еще одна ветвь исследований - создание трехмерных страниц, ярким представителем которой является язык моделирования виртуальной реальности VRML, который предложен компанией Silicon, но во-первых ни один из наиболее популярных с Сети интерфейсов его не поддерживает (загружается специальная программа), а во-вторых работать на нем широкие массы пользователей Internet вряд ли будут - все-таки это скорее средство для профессионалов, чем для простых пользователей.

**2.4.1.6 Тэги**

Тэгом считается определенное выражение, помещенное в скобки вида <>. Весьма условно тэги можно разделить на граничные и унарные.Унарные тэги представляют собой очевидные операции, предписывающие браузеру совершить определенное действие по постижению данного тэга при обра ботке html - документа.Пример - наиболее часто употребляемый тэг <BR> - переход на другую строку, весь следующий за ним гипертекст браузер будет представлять, начиная с новой строки.Унарные тэги не относятся к определенным участкам текста,а ,скорее всего, к всей последующей части документа.С граничными тегами сложнее, но надо учитывать, что их много больше , нежели унарных.Граничные тэги определяют, как будет выглядеть на экране пользователя часть текста,определеная этими элементами, причем в начальном тэге можно указывать различные параметры,опреде ленные данному тэгу, в конечном тэге эти параметры присутствовать не должны(браузером они игнорируются).  
  
Пример граничного тэга: гиперссылка <a href="http://uic.tula.ru/">Сервер ТулГУ</a>

Здесь

* *a* - управляющее слово, показывающие, что текст вплоть до тега </а> ,будет являться гипертекстовой ссылкой (то есть указнием браузеру просмотреть определенный html - документ при совершении управляющего действия ( шелчка мыши) над "ограниченным" текстом)
* *href="..."* - параметр, определяющий ,какой документ будет просмотрен (это может быть как файл на локальной или удаленной машинах,так и просто IP-адрес удаленной(с условием,что эта машина предоставляет http - сервис))

Некоторые параметры определены по умолчанию браузером, некоторые- пользователем браузера,но есть и такие,определять которые необходимо при создании страницы(пример - тотже параметр href тэга <a>).   
Из всего вышесказанного следует, что ,зная наиболее употребляемые тэги и их параметры, любой пользователь легко может редактировать и создавать гипертекст.При этом необходимо учитывать, что тэги могут включать в себя другие тэги, а так же необходимо иметь представление о структуре html - документов.

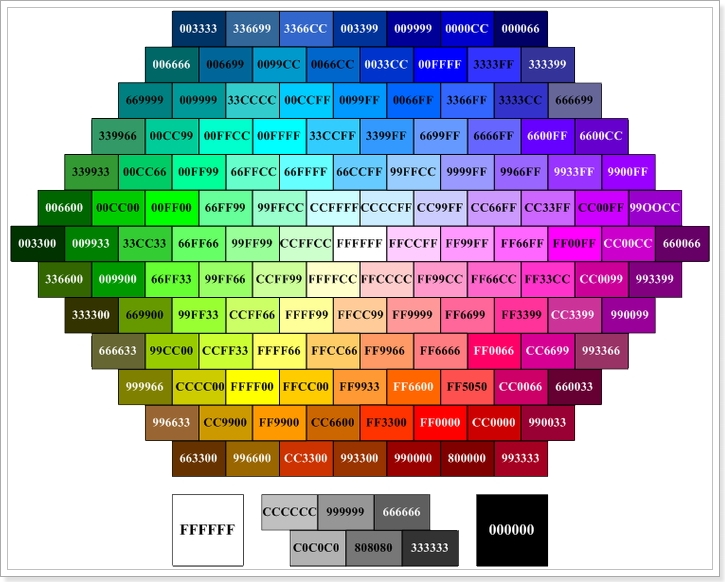


Рис. 2. Таблица цветов в HTML

**2.4.1.7 Структура HTML-документа**

HTML — теговый язык разметки [документов](https://ru.wikipedia.org/wiki/HTML-%D0%B4%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82). Любой документ на языке HTML представляет собой набор [элементов](https://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%B5%D0%BC%D0%B5%D0%BD%D1%82%D1%8B_HTML), причём начало и конец каждого элемента обозначается специальными пометками —[*тегами*](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%B3_(%D1%8F%D0%B7%D1%8B%D0%BA%D0%B8_%D1%80%D0%B0%D0%B7%D0%BC%D0%B5%D1%82%D0%BA%D0%B8)). Элементы могут быть *пустыми*, то есть не содержащими никакого текста и других данных (например, тег перевода строки <br>). В этом случае обычно не указывается закрывающий тег. Кроме того, элементы могут иметь *атрибуты*, определяющие какие-либо их свойства (например, размер шрифта для элемента font). Атрибуты указываются в открывающем теге. Вот примеры фрагментов HTML-документа:

* <strong>Текст между двумя тегами — открывающим и закрывающим.</strong>
* <a href="http://www.example.com">Здесь элемент содержит атрибут href, то есть гиперссылку.</a>
* А вот пример пустого элемента: <br>

Регистр, в котором набрано имя элемента и имена атрибутов, в HTML значения не имеет (в отличие от XHTML). Элементы могут быть вложенными. Например, следующий код:

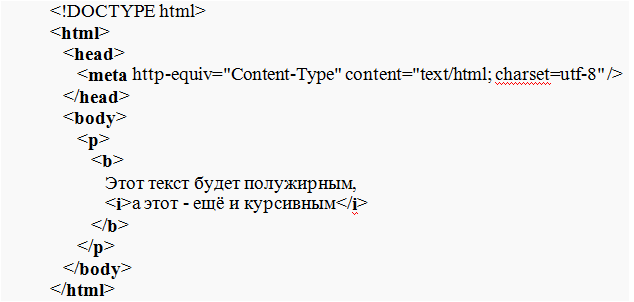


Рис. 2. Пример кода HTML

даст такой результат:

**Этот текст будет полужирным, *а этот — ещё и курсивным***

Кроме элементов, в HTML-документах есть и *сущности* ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) *entities)* — [«специальные символы»](https://ru.wikipedia.org/wiki/%D0%9C%D0%BD%D0%B5%D0%BC%D0%BE%D0%BD%D0%B8%D0%BA%D0%B8_%D0%B2_HTML). Сущности начинаются с символа [амперсанда](https://ru.wikipedia.org/wiki/%D0%90%D0%BC%D0%BF%D0%B5%D1%80%D1%81%D0%B0%D0%BD%D0%B4) и имеют вид &*имя*; или &#*NNNN*;, где *NNNN* — код символа в [Юникоде](https://ru.wikipedia.org/wiki/%D0%AE%D0%BD%D0%B8%D0%BA%D0%BE%D0%B4) в десятичной системе счисления.

Например, &copy; — знак [авторского права](https://ru.wikipedia.org/wiki/%D0%90%D0%B2%D1%82%D0%BE%D1%80%D1%81%D0%BA%D0%BE%D0%B5_%D0%BF%D1%80%D0%B0%D0%B2%D0%BE) (©). Как правило, сущности используются для представления символов, отсутствующих в кодировке документа, или же для представления «специальных» символов: &amp; — амперсанда (&), &lt; — символа «меньше» (<) и &gt; — символа «больше» (>), которые некорректно записывать «обычным» образом, из-за их особого значения в HTML.

Каждый HTML-документ, отвечающий спецификации HTML какой-либо версии, должен начинаться со строки объявления версии HTML <!DOCTYPE…>, которая обычно выглядит примерно так:

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"

"http://www.w3.org/TR/html4/strict.dtd">

Если эта строка не указана, то добиться корректного отображения документа в браузере становится труднее.

Далее обозначается начало и конец документа тегами <html> и </html> соответственно. Внутри этих тегов должны находиться теги заголовка (<head></head>) и тела (<body></body>) документа.



Рис.3. Возможность взаимодействия языка HTML с другими языками

**2.4.1.8 Основные тэги.**

**Основные теги**

<html></html> Указывает программе просмотра страниц что это HTML документ.

<head></head> Определяет место, где помещается различная информация не отображаемая в теле документа. Здесь располагается тег названия документа и теги для поисковых машин.

<body></body> Определяет видимую часть документа.

**Теги оглавления**

<title></title> Помещает название документа в оглавление программы просмотра страниц

**Атрибуты тела документа**

<body bgcolor=?> Устанавливает цвет фона документа, используя значение цвета в виде RRGGBB - пример: FF0000 - красный цвет.

<body text=?> Устанавливает цвет текста документа, используя значение цвета в виде RRGGBB - пример: 000000 - черный цвет.

<body link=?> Устанавливает цвет гиперссылок, используя значение цвета в виде RRGGBB - пример: 00FF00 - зеленый цвет.

<body vlink=?> Устанавливает цвет гиперссылок на котох вы уже побывали, используя значение цвета в виде RRGGBB - пример: 333333 - серый цвет.

<body alink=?> Устанавливает цвет гиперссылок при нажатии.

**Теги для форматирования текста**

<pre></pre> Обрамляет предварительно отформатированный текст.

<h1></h1> Создает САМЫЙ БОЛЬШОЙ заголовок

<h6></h6> Создает самый маленький заголовок

<b></b> Создает жирый текст

<i></i> Создает наклонный текст

<tt></tt> Создает текст - имитирующий стиль печатной машинки.

<cite></cite> Используется для цитат, обычно наклонный текст.

<em></em> Используется для выделения из текста слова (наклонный или жирный текст)

<strong></strong> Используется для выделения наиболее выжных частей текста (наклонный или жирный текст)

<font size=?></font> Устанавливает размер текста в пределах от 1 до 7.

<font color=?></font> Устанавливает цвет текста, используя значение цвета в виде RRGGBB.

**Гиперссылки**

<a href="URL"></a> Создает гиперссылку на другие документы или часть текущего документа.

<a href="mailto:EMAIL">

</a> Создает гиперссылку вызова почтовой программы для написания письма автору документа.

<a name="NAME"></a> Отмечает часть текста как цель для гипперссылок в документе.

<a href="#NAME"></a> Создает гиперссылку на часть текущего документа.

**Форматирование**

<p> Создает новый параграф

<p align=?> Выравнивает параграф относительно одной из сторон документа, значения: left, right, или center

<br> Вставляет перевод строки.

<blockquote>

</blockquote> Создает отступы с обеих сторон текста.

<dl></dl> Создает список определений.

<dt> Определяет каждый из терминов списка

<dd> Описывает каждое определение

<ol></ol> Создает нумерованный список

<li> Определяет каждый элемент списка и присваивает номер

<ul></ul> Создает ненумерованный список

<li> Предваряет каждый элемент списка и добавляет кружок или квадратик.

<div align=?> Важный тег используемый для форматирования больших блоков текста HTML документа, также используется в таблицах стилей

**Графические элементы**

<img src="name"> Добавляет изображение в HTML документ

<img src="name" align=?> Выравнивает изображение к одной из сторон документа, принимает значения: left, right, center; bottom, top, middle

<img src="name" border=?> Устанавливает толщину рамки вокруг изображения

<hr> Добавляет в HTML документ горизонтальную линию.

<hr size=?> Устанавливает высоту(толщину) линии

<hr width=?> Устанавливает ширину линии, можно указать ширину в пикселах или процентах.

<hr noshade> Создает линию без тени.

<hr color=?> Задает линии определенный цвет. Значение RRGGBB.

Таблицы

<table></table> Создает таблицу.

<tr></tr> Определяет строку в таблице.

<td></td> Определяет отдельную ячейку в таблице.

<th></th> Определяет заголовок таблицы (нормальная ячейка с отцентрованным жирным текстом)

Атрибуты таблицы

<table border=#> Задает толщину рамки таблицы.

<table cellspacing=#> Задает расстояние между ячейками таблицы.

<table cellpadding=#> Задает расстояние между содержимым ячейки и ее рамкой.

<table width=#> Устанавливает ширину таблицы в пикселах или процентах от ширины документа.

<tr align=?> или <td align=?> Устанавливает выравнивание ячеек в таблице, принимает значения: left, center, или right.

<tr valign=?> или <td valign=?> Устанавливает вертикальное выравнивание для ячеек таблицы, принимает значения : top, middle, или bottom.

<td colspan=#> Указывает кол-во столбцев которое объединено в одной ячейке. (по умолчанию=1)

<td rowspan=#> Указывает кол-во строк которое объединено в одной ячейке. (по умолчанию=1)

<td nowrap> Не позволяет программе просмотра делать перевод строки в ячейке таблицы.

**Кадры**

<frameset></frameset> Предваряет тег <body> в документе, содержащем кадры;

<frameset rows="value,value"> Определяет строки в таблице кадров, высота которых определена кол-вом пикселов или в процентном соотношении к высоте таблицы кадров.

<frameset cols="value,value"> Определяет столбцы в таблице кадров, ширина которых определена кол-вом пикселов или в процентном соотношении к ширине таблицы кадров.

<frame> Определяет единичный кадр или область в таблице кадров.

<noframes></noframes> Определяет, что будет показано в окне браузера если он не поддерживает кадры.

**Атрибуты кадров**

<frame src="URL"> Определяет какой из HTML документов будет показан в кадре.

<frame name="name"> Указывает Имя кадра или области, что позволяет перенаправлять информацию в этот кадр или область из других кадров.

<frame marginwidth=#> Определяет величину отступов по левому и правому краям кадра; должно быть равно или больше 1.

<frame marginheight=#> Определяет величину отступов по верхнему и нижнему краям кадра; должно быть равно или больше 1.

<frame scrolling=VALUE> Указывает будет-ли выводится линейка прокрутки в кадре; значение value может быть "yes," "no," или "auto". Значение по умолчанию для обычных документов - auto.

<frame noresize> Препятствует изменению размеров кадра.

**Формы**

Для форм, выполняющих какие-то функции должны быть запущены соответствующие CGI скрипты на сервере. HTML только создает внешний интерфейс формы.

<form></form> Создает формы

<select multiple name="NAME" size=?></select> Создает скролируемое меню. Size устанавливает кол-во пунктов меню, которое будет показано на экране, остальные будут доступны при использовании прокрутки.

<option> Указывает каждый отдельный элемент меню

<select name="NAME"></select> Создает ниспадающее меню

<option> Указывает каждый отдельный элемент меню

<textarea name="NAME" cols=40 rows=8></textarea> Создает окно для ввода текста. Columns указывает ширину окна; rows указывает его высоту.

<input type="checkbox" name="NAME"> Создает checkbox. За тегом следует текст.

<input type="radio" name="NAME" value="x"> Создает radio кнопку. За тегом следует текст.

<input type=text name="foo" size=20> Создает строку для ввода текста. Параметром Size указывается длина в символах.

<input type="submit" value="NAME"> Создает кнопку "Принять"

<input type="image" border=0 name="NAME" src="name.gif"> Создает кнопку "Принять" - для этого используется изображение

<input type="reset"> Создает кнопку "Отмена"

#### 2.4.2.9 О возможных ошибках

Непонятные конструкции языка броузером просто игнорируются. Так сделано умышленно: ведь основное применение броузера - сеть Internet, а работающий в сети, просматривая HTML-документы на чужой машине, не должен получать от своего броузера многочисленные сообщения из-за чужих ошибок.

Так что практически все ошибки не могут привести ни к чему фатальному - страница будет видна. Но результат, скорее всего, будет существенно отличаться от ожидаемого.

При изготовлении собственных страниц Вы наверняка будете ошибаться, и анализ причин ошибки может стать довольно непростым делом. Поэтому попытаемся указать хотя бы некоторые типичные случаи.

Отсутствие закрывающей скобки приводит к тому, что выделенный элемент текста продолжается дальше, чем это предусмотрено автором. Например, текст, выделяемый как ссылка скобками <a href> - </a> при отсутствии </a> остается выделенным до конца страницы или до следующей ссылки. В последнем случае срабатывает закрывающая скобка второй ссылки, но зато сама эта ссылка отсутствует.

Ссылка не срабатывает в случае неверного указания файла, на который мы ссылаемся. В этом случае броузер выдает сообщение об ошибке. Интересный эффект иногда получается при полном отсутствии имени илиуказании вместо имени знака вопроса или звездочки - выдается одно из оглавлений диска.

Вставляемый в документ графический файл может не отображаться и заменяться каким-нибудь условным значком вроде разорванной картинки по тем же причинам, что и несрабатывающая ссылка. А может быть, используемый Вами формат не поддерживается броузером. Выглядит это, например, так:

1. Если не отображаются все картинки на всех страницах, это скорее всего связано с настройками броузера - ему просто запрещено выдавать картинки (обычно просто отмечены места, где они должны быть). Что ж, придется менять настройки.
2. Довольно сложен для диагностики случай непарных кавычек, например, при указании имени файла. Внешние проявления могут быть самыми разнообразными, поскольку следующие открывающие кавычки воспринимаются закрывающими к первой паре, и далее вся структура документа полностью нарушается. При этом выпадают (т.е. не отображаются) большие куски документа. Нужно искать самое начало таких искажений, а затем внимательно смотреть исходный текст.
3. Русские буквы отображаются закорючками в случае указания отсутствующего на машине шрифта либо при несовпадении кодировки для редактора и броузера. Для тех, кто не знает: о кодировке русских букв можно написать целую книгу, и даже не одну. В сетях шли такие споры, что от мордобоя участников удерживали только разделяющие их тысячи километров. Если же говорить коротко, имеется около десятка базовых способов хранения русских букв (кириллицы) в компьютере, и их неисчислимые вариации. Часто даже в одной машине используются разные способы одновременно! Например, если Вы напечатали свой HTML-документ с использованием встроенного в Norton Commander редактора, в броузере под Windows Вы вряд ли увидите кириллицу. Тем не менее я рекомендую пользоваться этим редактором, а затем обрабатывать файл прилагаемой к Руководству программой. Так что давайте и дальше чтить Кирилла и Мефодия...
4. Закорючки не только вместо кириллицы, а вообще вместо всего, по крайней мере их много в начале документа. Вероятно, Вы готовили HTML-документ редактором Word. Не делайте этого! Вам нужен чистый текст, а Word пишет в файл огромное количество собственной служебной информации. Чтобы спасти положение, коль файл уже готов, можно попробовать превратить его в HTML встроенными в Word средствами, а потом поправить в нормальном текстовом редакторе. О редакторах, делающих чистый текст без всяких вставок [будет дальше](file:///C:\Users\nikita\Downloads\ch5.htm).

**2.4.2. Описание языка PHP**

**2.4.2.1 Понятие языка PHP**

PHP - язык создания сценариев, который давно перерос свое название. Дело в том, что PHP - это аббревиатура от слов Personal Home Page. Первая версия PHP была создана Расмусом Лердорфом в 1994 г. и представляла собой набор инструментов для отслеживания посетителей Web-страницы. Со временем PHP из набора инструментов превратился в полноценный язык программирования, а его название было изменено как рекурсивное образование PHP HyperText Preprocessor (препроцессор гипертекста PHP).

PHP - это серверный язык создания сценариев. Конструкции PHP, вставленные в HTML-текст, выполняются сервером при каждом посещении страницы. Результат их обработки вместе с обычным HTML-текстом передается браузеру.



Рис. 4. Логотип языка PHP

**2.4.2.2 История языка PHP**

В [1994 году](https://ru.wikipedia.org/wiki/1994_%D0%B3%D0%BE%D0%B4) [датский](https://ru.wikipedia.org/wiki/%D0%94%D0%B0%D0%BD%D0%B8%D1%8F) [программист](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%81%D1%82) [Расмус Лердорф](https://ru.wikipedia.org/wiki/%D0%9B%D0%B5%D1%80%D0%B4%D0%BE%D1%80%D1%84,_%D0%A0%D0%B0%D1%81%D0%BC%D1%83%D1%81) создал набор [скриптов](https://ru.wikipedia.org/wiki/%D0%A1%D0%BA%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D0%B2%D1%8B%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) на [Perl/CGI](https://ru.wikipedia.org/wiki/Perl) для вывода и учёта посетителей его [онлайн](https://ru.wikipedia.org/w/index.php?title=%D0%9E%D0%BD%D0%BB%D0%B0%D0%B9%D0%BD&action=edit&redlink=1)-[резюме](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%B7%D1%8E%D0%BC%D0%B5), обрабатывающий шаблоны [HTML](https://ru.wikipedia.org/wiki/HTML)-документов. Лердорф назвал набор Personal Home Page (Личная Домашняя Страница). Вскоре функциональности и быстроты Perl — интерпретатора скриптов — перестало хватать, и Лердорф разработал с использованием языка [C](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8_(%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)) новый интерпретатор шаблонов PHP/FI ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) Personal Home Page / Forms Interpreter — «персональная домашняя страница / интерпретатор форм»).

В [1997 году](https://ru.wikipedia.org/wiki/1997_%D0%B3%D0%BE%D0%B4) после длительного [бета-тестирования](https://ru.wikipedia.org/wiki/%D0%91%D0%B5%D1%82%D0%B0-%D1%82%D0%B5%D1%81%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5) вышла вторая версия обработчика, написанного на [C](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8_(%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)) — PHP/FI 2.0. Её использовали около 1 % (приблизительно 50 тысяч) всех [интернет-доменов](https://ru.wikipedia.org/wiki/%D0%94%D0%BE%D0%BC%D0%B5%D0%BD%D0%BD%D0%BE%D0%B5_%D0%B8%D0%BC%D1%8F)мира.

Версия PHP 3.0 подверглась значительной переработке, определившей современный облик и стиль языка программирования. В [1997 году](https://ru.wikipedia.org/wiki/1997_%D0%B3%D0%BE%D0%B4) два [израильских](https://ru.wikipedia.org/wiki/%D0%98%D0%B7%D1%80%D0%B0%D0%B8%D0%BB%D1%8C) [программиста](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%81%D1%82), [Энди Гутманс](https://ru.wikipedia.org/wiki/%D0%93%D1%83%D1%82%D0%BC%D0%B0%D0%BD%D1%81,_%D0%AD%D0%BD%D0%B4%D0%B8) и [Зеев Сураски](https://ru.wikipedia.org/wiki/%D0%A1%D1%83%D1%80%D0%B0%D1%81%D0%BA%D0%B8,_%D0%97%D0%B5%D0%B5%D0%B2), полностью переписали код интерпретатора. PHP 3.0 был официально выпущен в июне [1998 года](https://ru.wikipedia.org/wiki/1998_%D0%B3%D0%BE%D0%B4).

Одной из сильнейших сторон PHP 3.0 была возможность расширения ядра дополнительными модулями. Впоследствии интерфейс написания расширений привлёк к PHP множество сторонних разработчиков, работающих над своими модулями, что дало PHP возможность работать с огромным количеством [баз данных](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85), [протоколов](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%82%D0%B5%D0%B2%D0%BE%D0%B9_%D0%BF%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB), поддерживать большое число [API](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F_%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B9). Большое количество разработчиков привело к быстрому развитию языка и стремительному росту его популярности. С этой версии акроним php расшифровывается как «PHP: hypertext Preprocessor», вместо устаревшего «Personal Home Page».

К зиме [1998 года](https://ru.wikipedia.org/wiki/1998_%D0%B3%D0%BE%D0%B4), практически сразу после официального выхода PHP 3.0, Энди Гутманс и Зеев Сураски начали переработку ядра PHP. В задачи входило увеличение производительности сложных приложений и улучшение модульности базиса кода PHP. Новый движок, названный [Zend Engine](https://ru.wikipedia.org/wiki/Zend_Engine), успешно справлялся с поставленными задачами и впервые был представлен в середине [1999 года](https://ru.wikipedia.org/wiki/1999_%D0%B3%D0%BE%D0%B4). PHP 4.0, основанный на этом движке и принёсший с собой набор дополнительных функций, официально вышел в мае [2000 года](https://ru.wikipedia.org/wiki/2000_%D0%B3%D0%BE%D0%B4). В дополнение к улучшению производительности, PHP 4.0 имел ещё несколько ключевых нововведений, таких как поддержка сессий, [буферизация](https://ru.wikipedia.org/wiki/%D0%91%D1%83%D1%84%D0%B5%D1%80%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F_(%D0%B8%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)) вывода, более безопасные способы обработки вводимой пользователем информации и несколько новых языковых конструкций.

Пятая версия PHP была выпущена разработчиками [13 июля](https://ru.wikipedia.org/wiki/13_%D0%B8%D1%8E%D0%BB%D1%8F) [2004 года](https://ru.wikipedia.org/wiki/2004_%D0%B3%D0%BE%D0%B4). Изменения включают обновление ядра Zend (Zend Engine 2), что существенно увеличило эффективность [интерпретатора](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BF%D1%80%D0%B5%D1%82%D0%B0%D1%82%D0%BE%D1%80). Введена поддержка языка разметки [XML](https://ru.wikipedia.org/wiki/XML). Полностью переработаны функции [ООП](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5), которые стали во многом схожи с моделью, используемой в [Java](https://ru.wikipedia.org/wiki/Java). В частности, введён [деструктор](https://ru.wikipedia.org/wiki/%D0%94%D0%B5%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%82%D0%BE%D1%80), [открытые, закрытые и защищённые члены и методы](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D0%BA%D0%B0%D0%BF%D1%81%D1%83%D0%BB%D1%8F%D1%86%D0%B8%D1%8F_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)), окончательные члены и методы, [интерфейсы](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%82%D0%B5%D1%80%D1%84%D0%B5%D0%B9%D1%81_(%D0%BE%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) и клонирование объектов. В последующих версиях также были введены [пространства имён](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%81%D1%82%D1%80%D0%B0%D0%BD%D1%81%D1%82%D0%B2%D0%BE_%D0%B8%D0%BC%D1%91%D0%BD_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)), [замыкания](https://ru.wikipedia.org/wiki/%D0%97%D0%B0%D0%BC%D1%8B%D0%BA%D0%B0%D0%BD%D0%B8%D0%B5_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) и целый ряд достаточно серьёзных изменений, количественно и качественно сравнимых с теми, которые появились при переходе на PHP 5.0.

Шестая версия PHP разрабатывалась с [октября 2006](https://ru.wikipedia.org/wiki/%D0%9E%D0%BA%D1%82%D1%8F%D0%B1%D1%80%D1%8C_2006_%D0%B3%D0%BE%D0%B4%D0%B0) года. Было сделано множество нововведений, как, например, исключение из ядра регулярных выражений [POSIX](https://ru.wikipedia.org/wiki/POSIX) и «длинных» суперглобальных массивов, удаление директив safe\_mode, magic\_quotes\_gpc и register\_globals из конфигурационного файла php.ini. Одним из основных новшеств должна была стать поддержка [Юникода](https://ru.wikipedia.org/wiki/%D0%AE%D0%BD%D0%B8%D0%BA%D0%BE%D0%B4). Однако в [марте 2010](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%80%D1%82_2010_%D0%B3%D0%BE%D0%B4%D0%B0) года разработка PHP6 была признана бесперспективной из-за сложностей с поддержкой [Юникода](https://ru.wikipedia.org/wiki/%D0%AE%D0%BD%D0%B8%D0%BA%D0%BE%D0%B4). Исходный код PHP6 перемещён на [ветвь](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D1%82%D0%B2%D1%8C_(%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5_%D0%B2%D0%B5%D1%80%D1%81%D0%B8%D1%8F%D0%BC%D0%B8)), а основной линией разработки стала версия 5.4.

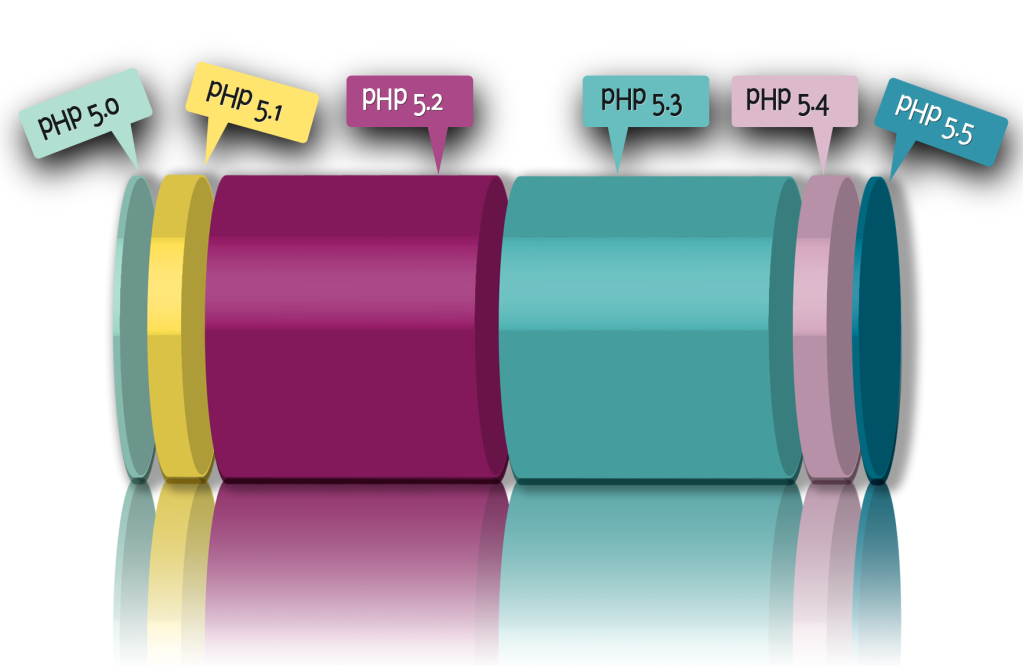


Рис. 5. График использования различных версий языка PHP

**2.4.2.3 Область применения PHP**

* Создание скриптов для выполнения на стороне сервера. PHP традиционно и наиболее широко используется именно таким образом. Для этого вам будут необходимы три вещи. Интерпретатор PHP (в виде программы CGI или серверного модуля), веб-сервер и браузер. Для того чтобы можно было просматривать результаты выполнения PHP-скриптов в браузере, нужен работающий веб-сервер и установленный PHP. Просмотреть вывод PHP-программы можно в браузере, получив PHP-страницу, сгенерированную сервером. В случае, если вы просто экспериментируете, вы вполне можете использовать свой домашний компьютер вместо сервера. За более подробными сведениями обратитесь к главе Советы по установке.
* Создание скриптов для выполнения в командной строке. Вы можете создать PHP-скрипт, способный запускаться без сервера или браузера. Все, что вам потребуется - парсер PHP. Такой способ использования PHP идеально подходит для скриптов, которые должны выполняться регулярно, например, с помощью cron (на платформах \*nix или Linux) или с помощью планировщика задач (Task Scheduler) на платформах Windows. Эти скрипты также могут быть использованы в задачах простой обработки текстов. За дополнительной информацией обращайтесь к главе Использование PHP в среде командной строки.
* Создание оконных приложений, выполняющихся на стороне клиента. Возможно, PHP является не самым лучшим языком для создания подобных приложений, но, если вы очень хорошо знаете PHP и хотели бы использовать некоторые его возможности в своих клиентских приложениях, вы можете использовать PHP-GTK для создания таких приложений. Подобным образом вы можете создавать и кросс-платформенные приложения. PHP-GTK является расширением PHP и не поставляется вместе с основным дистрибутивом PHP. Если вы заинтересованы, посетите » сайт PHP-GTK.

**2.4.2.4 Возможности языка PHP**

PHP может практически все, по крайне мере так заявляют его создатели ;). Главной областью применения PHP является написание сценариев, которые работают на стороне сервера. Помимо различных операций с содержимым ваших html-документов, с помощью PHP можно форматировать HTTP-заголовки, устанавливать cookie, управлять аутентификацией, а также перенаправлять пользователя на другие страницы. PHP предоставляет великолепные возможности по доступу к базам данных, позволяет интеграцию с различными внешними библиотеками. Это позволит вам делать практически все, что вы захотите - от генерации PDF-документов до подробного парсинга в XML.

Рассмотрим некоторые достоинства PHP:

* Весь код обрабатывается и исполняется на стороне сервера
* Поддерживает работу с множеством СУБД (MySQL, Oracle, PostgreSQL и т.д.)
* Является программным обеспечением с открытым исходным кодом
* Работает на разных платформах (Windows, Linux, Unix подобных)
* PHP очень прост для освоения
* Много возможностей по расширению возможностей языка
* Поддержка различных веб-серверов

Таким образом, выбирая в качестве языка программирования PHP, вы получаете огромные возможности по использованию языка, свободу выбора понравившегося веб-сервера и операционной системы. Более того, у вас появляется выбор между использованием объектно-ориентированного программирования (ООП) или процедурного или же их сочетания.

**2.4.2.5 Типы данных в PHP**

PHP поддерживает восемь простых типов данных:

1. Четыре скалярных типа:

- boolean (двоичные данные)

- integer (целые числа)

- float (числа с плавающей точкой или 'double')

- string (строки)

1. Два смешанных типа:

- array (массивы)

- object (объекты)

1. И два специальных типа:

resource (ресурсы)

NULL ("пустые")

1. Существуют также несколько псевдотипов:

- mixed (смешанные)

- number (числа)

- callback (обратного вызова)

**2.4.2.6 Общие правила построения PHP-программы**

PHP-программы состоят из простого текста, поэтому набирать их можно в любом текстовом редакторе. Популярные HTML-редакторы имеют встроенную поддержку для редактирования PHP-программ.

Расширение файлов PHP-программ по умолчанию в PHP4 - **.php**. На основании этого расширения сервер распознает файл как PHP-программу и запускает интерпретатор.

PHP-программа должна быть отделена от обычного HTML-текста. Существует четыре стиля обрамления PHP-кода:

Стили обрамления PHP кода Таблица 1

|  |  |  |
| --- | --- | --- |
| **Стиль** | **Открывающий тег** | **Закрывающий тег** |
| Сокращенный | <? | ?> |
| XML (стандартный) | <?php | ?> |
| ASP | <% | %> |
| SCRIPT (программный) | <SCRIPT LANGUAGE="php"> | </SCRIPT> |

#### 2.4.2.7 Управляющие конструкции языка PHP

Основными конструкциями языка PHP являются:

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

**2.4.2.8 Синтаксис**

Синтаксис PHP подобен синтаксису языка [Си](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8_(%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)). Некоторые элементы, такие как [ассоциативные массивы](https://ru.wikipedia.org/wiki/%D0%90%D1%81%D1%81%D0%BE%D1%86%D0%B8%D0%B0%D1%82%D0%B8%D0%B2%D0%BD%D1%8B%D0%B9_%D0%BC%D0%B0%D1%81%D1%81%D0%B8%D0%B2) и цикл foreach, заимствованы из [Perl](https://ru.wikipedia.org/wiki/Perl).

Для работы программы не требуется описывать какие-либо переменные, используемые модули и т. п. Любая программа может начинаться непосредственно с оператора PHP.

Простейшая [программа Hello world](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0_Hello_world) на PHP выглядит следующим образом:

<?php

echo 'Hello, world!';

?>

Также возможен более короткий вариант вывода строки:

<?= 'Hello, world!' ?>

Открывающий тег вида <?= используется для сокращённой записи конструкций используемых для вывода строки.

PHP исполняет код, находящийся внутри ограничителей, таких как <?php ?>. Всё, что находится вне ограничителей, выводится без изменений. В основном это используется для вставки PHP-кода в HTML-документ, например, так:



Рис. 6. Пример простейшей программы на языке PHP и результат выполнения программы

Помимо ограничителей <?php ?>, допускается использование дополнительных вариантов, таких как <? ?> и <script language="php"> </script>. Кроме того, до версии 6.0 допускается использование ограничителей языка программирования ASP <% %> (конструкции <? ?> и <% %> могут быть выключены в конфигурационном файле php.ini).

Имена переменных начинаются с символа $, тип переменной объявлять не нужно. Имена переменных и констант чувствительны к регистру символов. Имена классов, методов классов и функций к регистру символов не чувствительны. Переменные обрабатываются в строках, заключённых в двойные кавычки, и heredoc-строках (строках, созданных при помощи оператора <<<). Переменные в строках, заключенных в одинарные кавычки, не обрабатываются.

PHP рассматривает переход на новую строку как пробел, так же как HTML и другие языки со свободным форматом. Инструкции разделяются с помощью точки с запятой (;), за исключением некоторых случаев, после объявления конструкции if/else и циклов.

Переменные в функцию можно передавать как по значению, так и по ссылке (используется знак &).

PHP поддерживает три типа комментариев: в стиле языка Си (ограниченные /\* \*/), C++ (начинающиеся с // и идущие до конца строки) и оболочки UNIX (с # до конца строки).

#### 2.4.2.9 Описание типов данных PHP

PHP является языком программирования с [динамической типизацией](https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%BD%D0%B0%D0%BC%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D1%82%D0%B8%D0%BF%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F), не требующим указания типа при объявлении переменных, равно как и самого объявления переменных. Преобразования между скалярными типами зачастую осуществляются неявно без дополнительных усилий (впрочем, PHP предоставляет широкие возможности и для явного преобразования типов).

Диапазон целых чисел (integer) в PHP зависит от [платформы](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%BF%D0%BB%D0%B0%D1%82%D1%84%D0%BE%D1%80%D0%BC%D0%B0) (обычно, это диапазон 32-битных знаковых целых чисел, то есть, от −2 147 483 648 до 2 147 483 647). Числа можно задавать в десятичной,[восьмеричной](https://ru.wikipedia.org/wiki/%D0%92%D0%BE%D1%81%D1%8C%D0%BC%D0%B5%D1%80%D0%B8%D1%87%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%81%D1%87%D0%B8%D1%81%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F) и [шестнадцатеричной](https://ru.wikipedia.org/wiki/%D0%A8%D0%B5%D1%81%D1%82%D0%BD%D0%B0%D0%B4%D1%86%D0%B0%D1%82%D0%B5%D1%80%D0%B8%D1%87%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%81%D1%87%D0%B8%D1%81%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F) системах счисления. Диапазон вещественных чисел (double) также зависит от платформы (для 32-битной архитектуры диапазон позволяет оперировать числами от ±1.7×10−308 до ±1.7×10+308).

PHP предоставляет разработчикам [логический тип](https://ru.wikipedia.org/wiki/%D0%9B%D0%BE%D0%B3%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D1%82%D0%B8%D0%BF) (boolean), способный принимать только два значения TRUE («истина») и FALSE («ложь»). При преобразовании в логический тип число 0, пустая строка, ноль в строке «0», NULL и пустой массив считаются равными FALSE. Все остальные значения автоматически преобразуются в TRUE.

Специальный тип NULL предназначен для переменных без определённого значения. Единственным значением данного типа является константа NULL. Тип NULL принимают неинициализированные переменные, переменные инициализированные константой NULL, а также переменные, удалённые при помощи конструкции unset().

Ссылки на внешние ресурсы имеют тип «ресурс» (resource). Переменные данного типа, как правило, представляют собой [дескриптор](https://ru.wikipedia.org/wiki/%D0%94%D0%B5%D1%81%D0%BA%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D1%80_(%D1%8F%D0%B7%D1%8B%D0%BA%D0%B8_%D1%80%D0%B0%D0%B7%D0%BC%D0%B5%D1%82%D0%BA%D0%B8)), позволяющий управлять внешними объектами, такими как файлы, динамические изображения, результирующие таблицы базы данных и т. п.

[Массивы](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D0%B4%D0%B5%D0%BA%D1%81%D0%BD%D1%8B%D0%B9_%D0%BC%D0%B0%D1%81%D1%81%D0%B8%D0%B2) (array) поддерживают числовые и строковые ключи и являются [гетерогенными](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D0%B4%D0%B5%D0%BA%D1%81%D0%BD%D1%8B%D0%B9_%D0%BC%D0%B0%D1%81%D1%81%D0%B8%D0%B2#.D0.A1.D0.BF.D0.B5.D1.86.D0.B8.D1.84.D0.B8.D1.87.D0.B5.D1.81.D0.BA.D0.B8.D0.B5_.D1.82.D0.B8.D0.BF.D1.8B_.D0.BC.D0.B0.D1.81.D1.81.D0.B8.D0.B2.D0.BE.D0.B2). Массивы могут содержать значения любых типов, включая другие массивы. Порядок элементов и их ключей сохраняется. Не совсем корректно называть php-массивы массивами, на самом деле это, скорее всего, упорядоченный [хеш](https://ru.wikipedia.org/wiki/%D0%90%D1%81%D1%81%D0%BE%D1%86%D0%B8%D0%B0%D1%82%D0%B8%D0%B2%D0%BD%D1%8B%D0%B9_%D0%BC%D0%B0%D1%81%D1%81%D0%B8%D0%B2#PHP). Возможно неожиданное поведение при использовании цикла for со счетчиком вместо foreach. Так, например, при сортировке массива с численными индексами функциями из стандартной библиотеки, сортируются и ключи тоже.

Указатель на функцию в PHP может быть представлен замыканием или псевдотипом [callback](https://ru.wikipedia.org/wiki/Callback_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)). Замыкание доступно с версии 5.3 и в коде выглядит как простое определение функции, в которую явно можно утянуть значения из контекста, например:

**function**($args..$argsN) **use**($ctxVar,$ctxVar1) { definition ; }

callback тип может быть представлен:

* строкой (интерпретируется как название функции);
* массивом где нулевой и первый элемент — строки (интерпретируется как название статической функции класса);
* массивом где нулевой элемент — объект, а первый — строка (интерпретируется как метод у объекта).

Для проверки является ли значение вызываемым следует использовать is\_callable($var).

#### 2.4.2.10 PHP и HTML

Синтаксис любого языка программирования гораздо легче "почувствовать" на примерах, нежели используя какие-то диаграммы и схемы. Поэтому приведем пример простейшего скрипта на PHP:

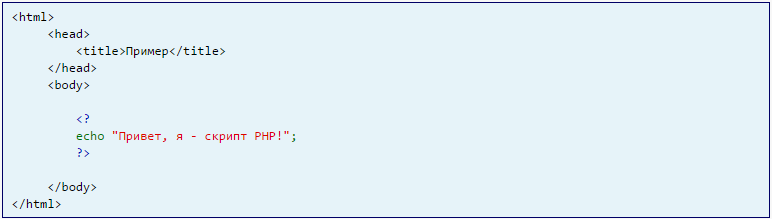


Рис. 7. Пример синтаксиса языков PHP и HTML

Это классический скрипт, с которого начинают изучение языка программирования. HTML-код корректно обрабатывается интерпретатором PHP.

Начало сценария начинается тегов HTML: <html> и <body>. Это главная особенность языка PHP: PHP-скрипт может вообще не отличаться от обычного HTML-документа.

Сам код сценария начинается после открывающего тэга <? и заканчивается закрывающим ?>. Между этими двумя тэгами текст интерпретируется как программа, и в HTML-документ не попадает. Если же программе нужно что-то вывести, она должна воспользоваться оператором echo.

PHP устроен так, что любой текст, который расположен вне программных блоков, ограниченных <? и ?>, выводится в браузер непосредственно. В этом и заключается главная особенность PHP, в отличие от Perl и C, где вывод осуществляется только с помощью стандартных операторов.

**3 Специальная часть**

## 3.1 Постановка задачи

Разрабатываемый макет электронного портфолио должен быть создан в виде сайта, который будет содержать следующие разделы:

1. Титульный лист
2. Содержание портфолио
3. Визитная карточка
4. Характеристика профессиональной деятельности
5. Сводная таблица проверяемых ОК, ПК
6. Достижения в освоении образовательной программы
7. Учебная деятельность (аудиторная, внеаудиторная)
8. Результаты прохождения практики
9. Участие в УИРС, НИРС
10. Дополнительные личные достижения (участие в общественной жизни)
11. Военная подготовка (для юношей)
12. Оценка достижений
13. Перечень документов и материалов «Достигнутые результаты освоения образовательной программы»
14. Оценка портфолио (ОК, ПК)

## 3.2 Назначение задачи

Разработанный макет электронного портфолио предназначен для автоматизации процесса создания портфолио, экономии времени студента при создании и заполнении портфолио, улучшения оформления и восприятия портфолио.

## 3.3 Требования к программе

Разрабатываемый макет электронного портфолио должен быть представлен в форме сайта, который:

* имеет простой и удобный интерфейс;
* представляет информацию в удобном для восприятия виде;

Требования к программному обеспечению ПК:

* выход в Интернет;
* браузер Mozilla FireFox;
* текстовый редактор Ms Word;

## 3.4 Описание алгоритма

В результате работы над проектом был создан макет электронного портфолио для студента в форме сайта.

После запуска браузера переходим по адресу сайта http://portfoliostud.esy.es/.

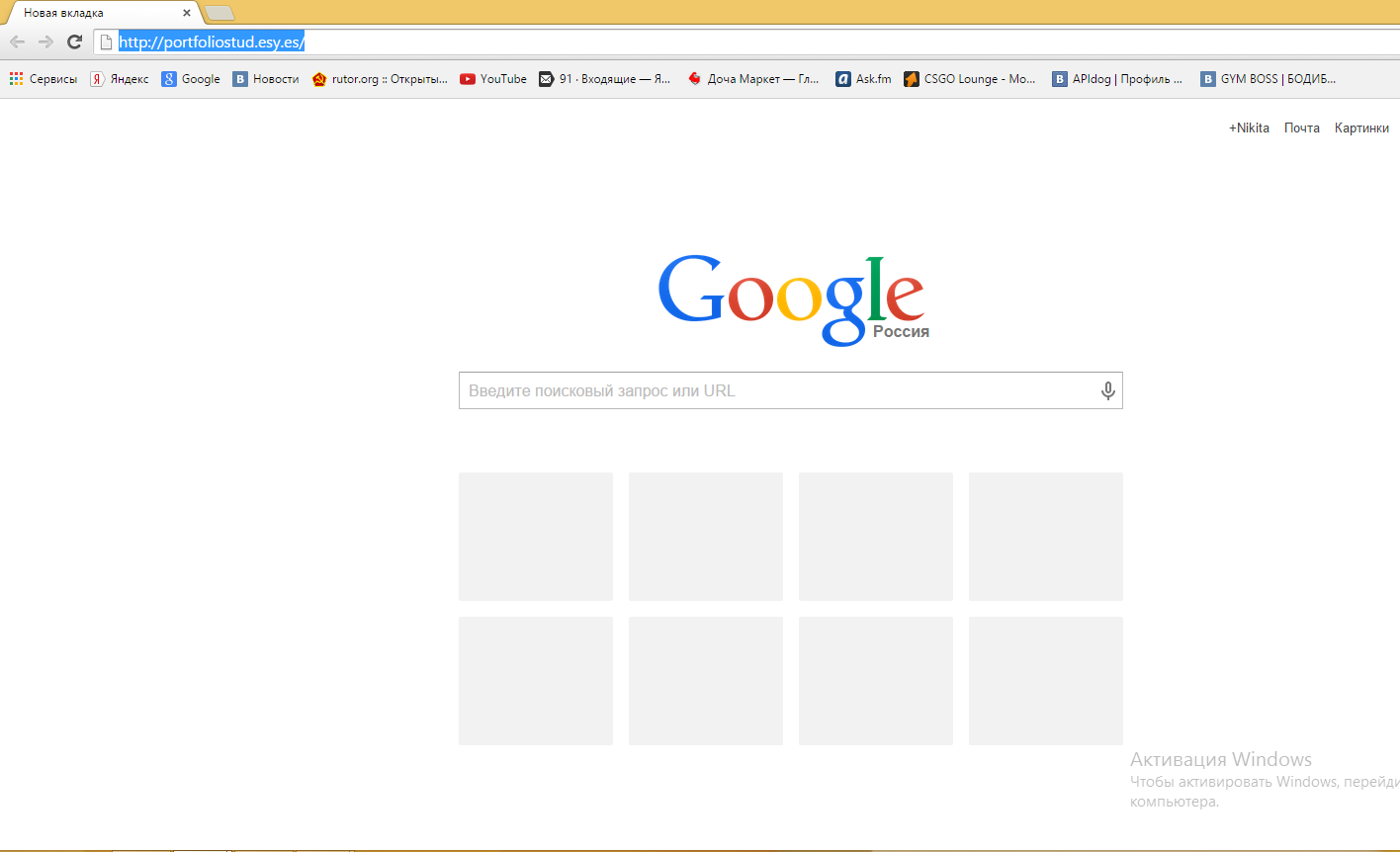


Рис.8. Запуск браузера и написание в адресной строке адрес сайта

Переходим на сайт http://portfoliostud.esy.es/.

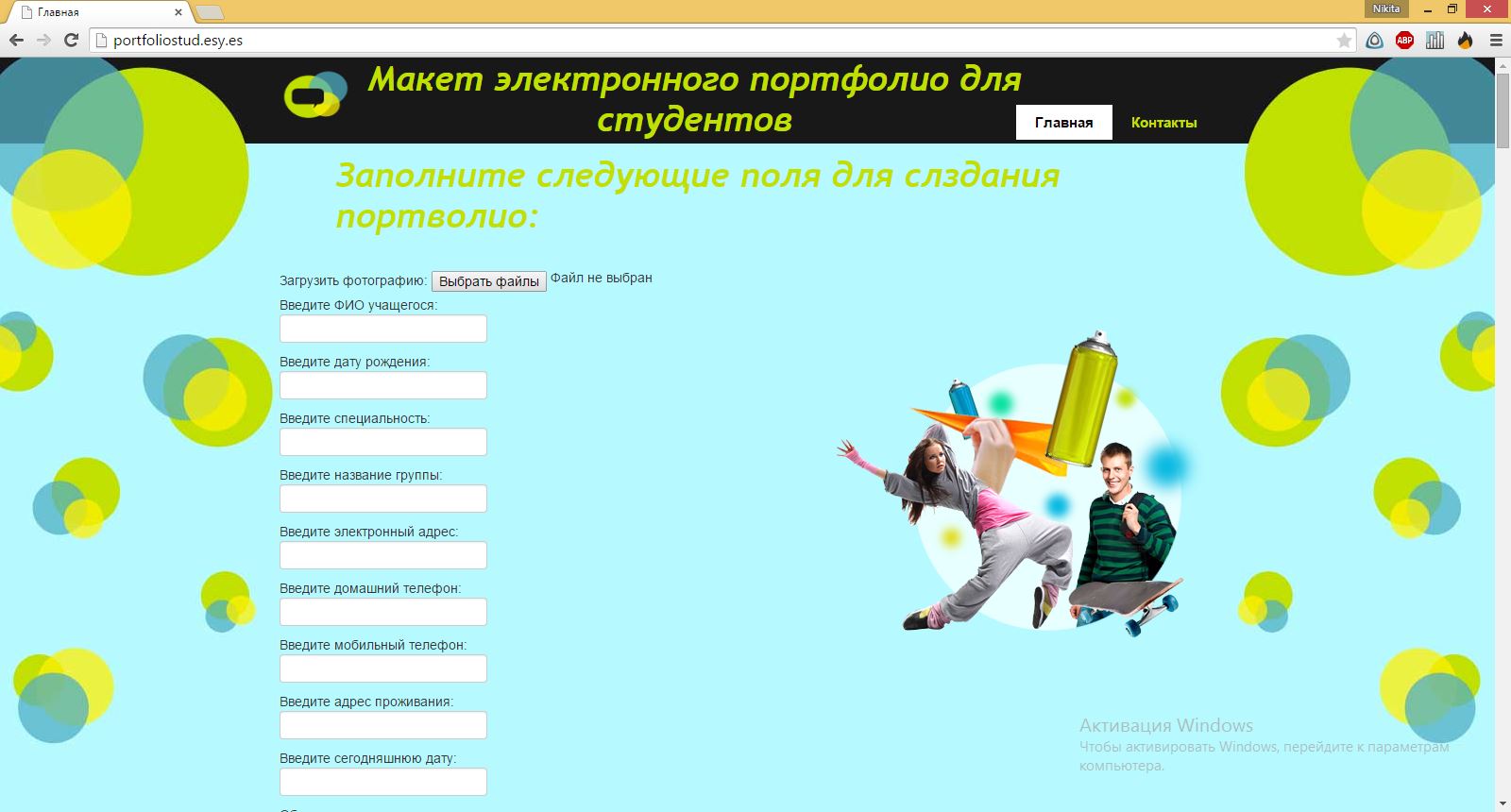


Рис.9. Открытие сайта ://portfoliostud.esy.es/.

Для создания портфолио необходимо заполнить все поля своими данными.

Для начала необходимо выбрать и загрузить свою фотографию:

Для этого нажимаем на кнопку "Выбрать файл".



Рис.10. Кнопка "Выбрать файл".

Открывается окно для выбора фотографии. Выбираем фотографию и нажимаем на кнопку "Открыть".



Рис.11. Кнопка "Открыть".

После выбора, справа от кнопки будет написано имя загруженной фотографии.



Рис.12. Имя загруженной фотографии.

Далее необходимо заполнить все поля своей информацией

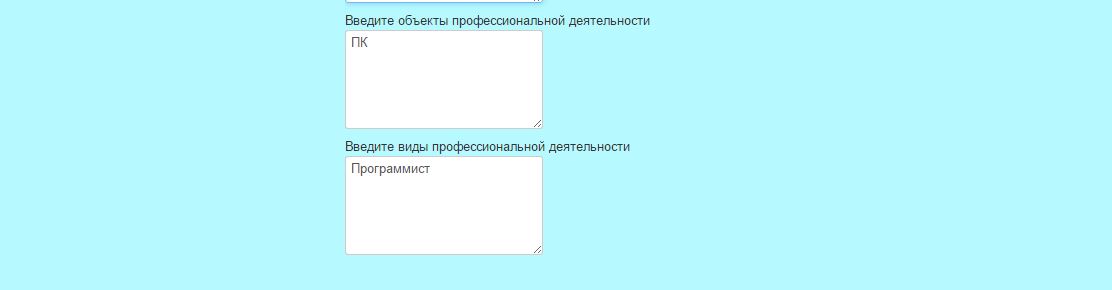
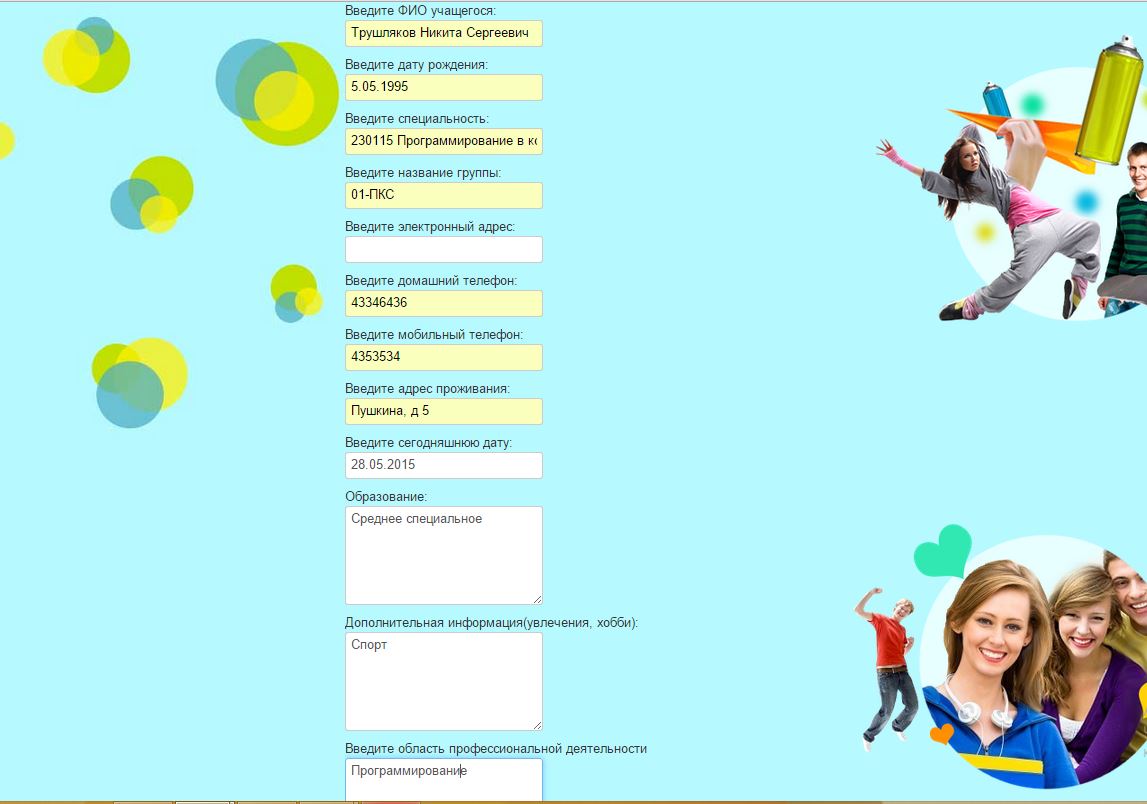


Рис. 14. Заполнение полей информацией.

Далее следует заполнить поля всех необходимых таблиц:



Рис.15. Заполнение полей таблицы



Рис.16. Заполнение полей таблицы

Необходимо так же выбрать фотографии грамот (при их наличие), нажав на кнопку выбрать файл:

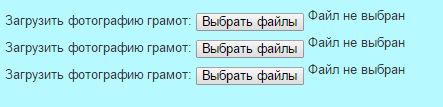


Рис.17. Кнопки для выбора фотографий грамот

И выбираем необходимую фотографию выбрав и нажав на кнопку открыть:

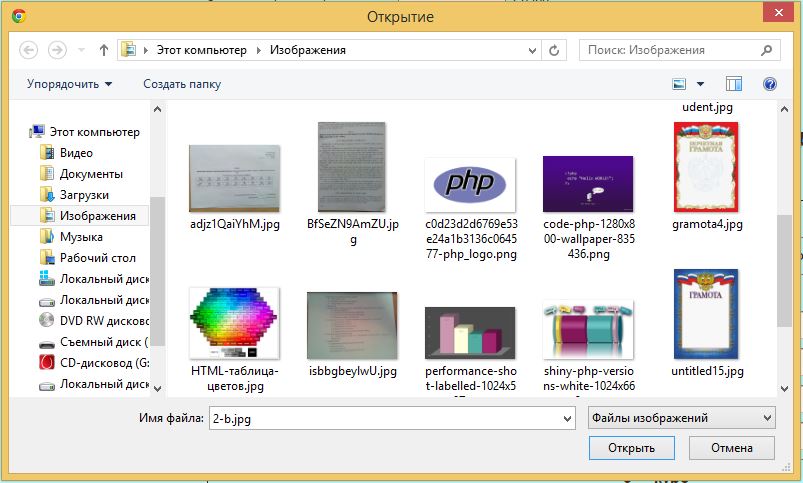


Рис.18. Выбор фотографии грамоты

Далее продолжаем заполнять поля в таблицах:

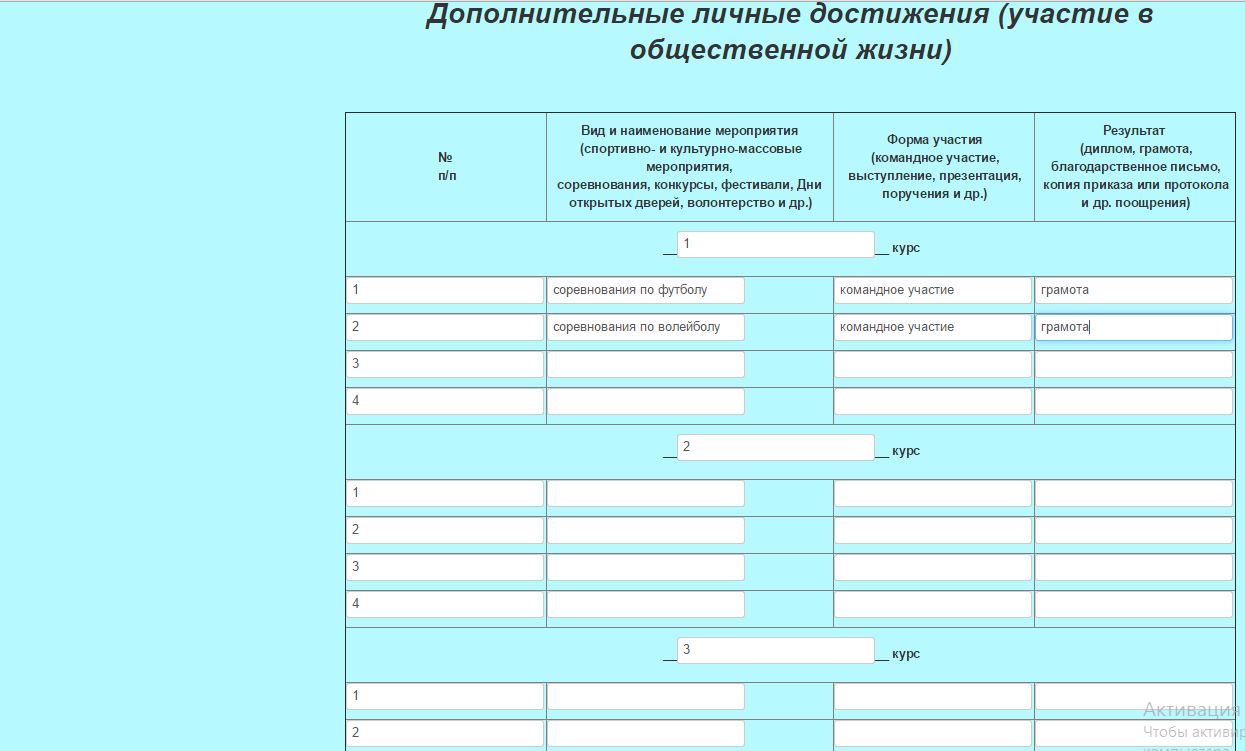


Рис.19. Заполнение полей таблицы



Рис.20. Заполнение полей таблицы

Рис.15. Заполнение полей таблицы



После того, как заполнены все поля таблицы, необходимо пройти анкетирование, ответив на 5 вопросов, вписывая ответы в поля для ввода:

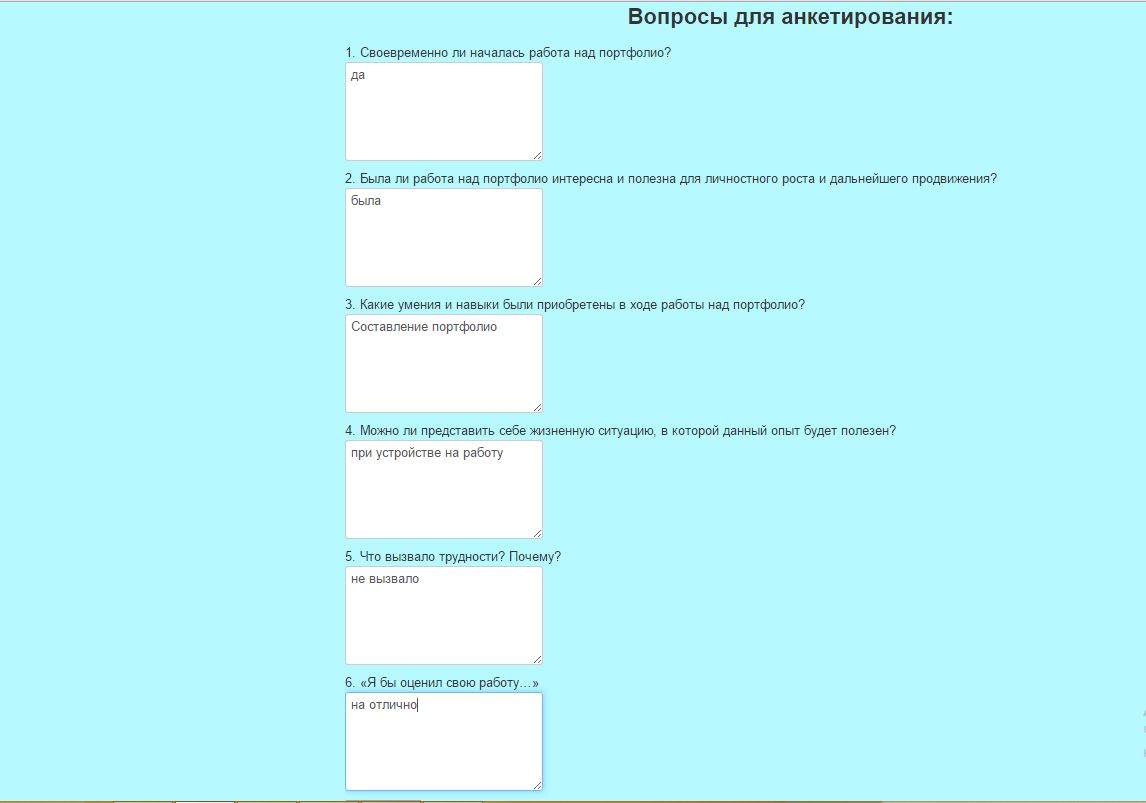


Рис.21. Заполнение полей анкетирования

После того, как все необходимые данные были внесены, нажимаем на кнопку "Отправить".



Рис.22. Кнопка отправить

Произойдет переход на другую страницу, где будут отображаться все внесенные данные.

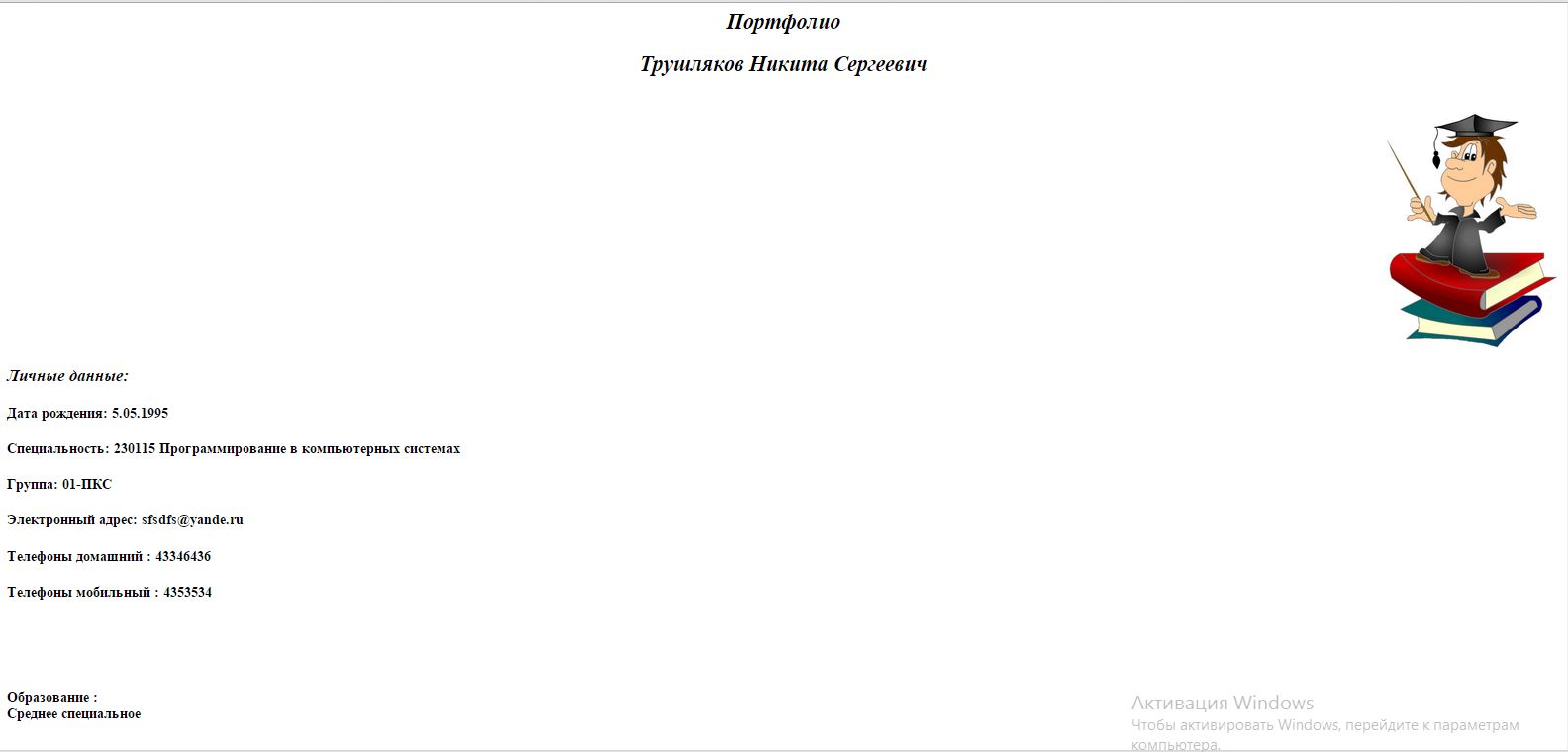


Рис.23. Переход на новую страницу с отображением данных

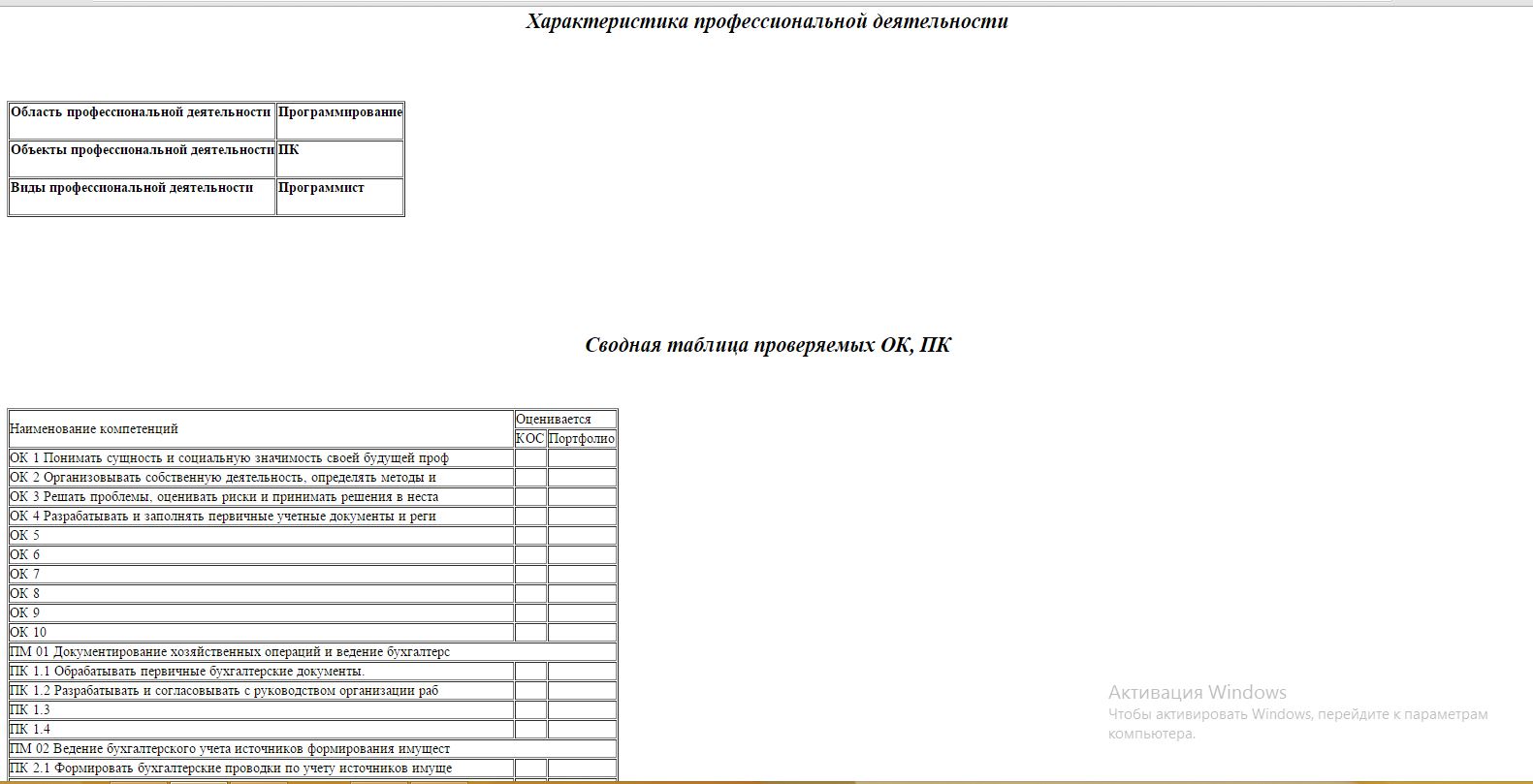


Рис.24. Вывод заполненных таблиц



Рис.25. Вывод загруженных грамот

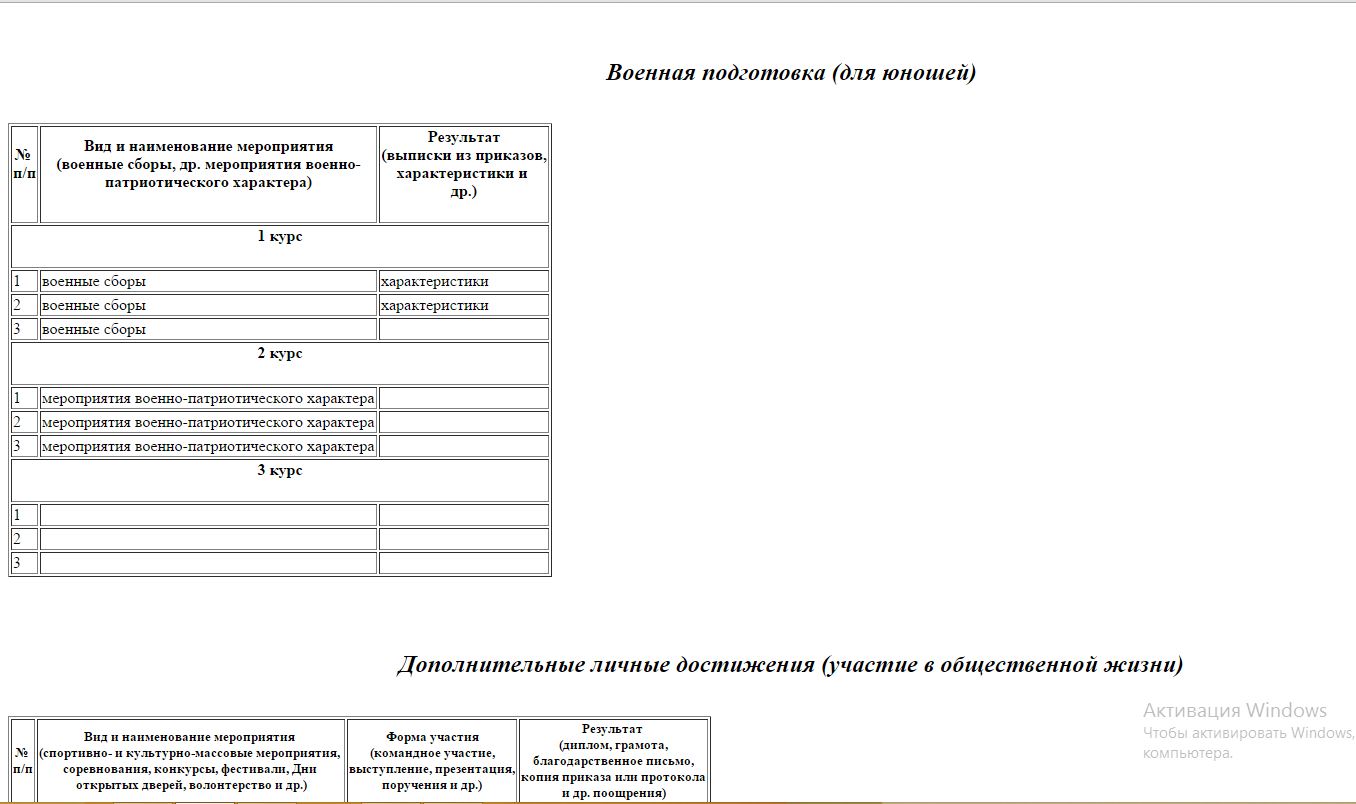


Рис.26.Вывод заполненных таблиц

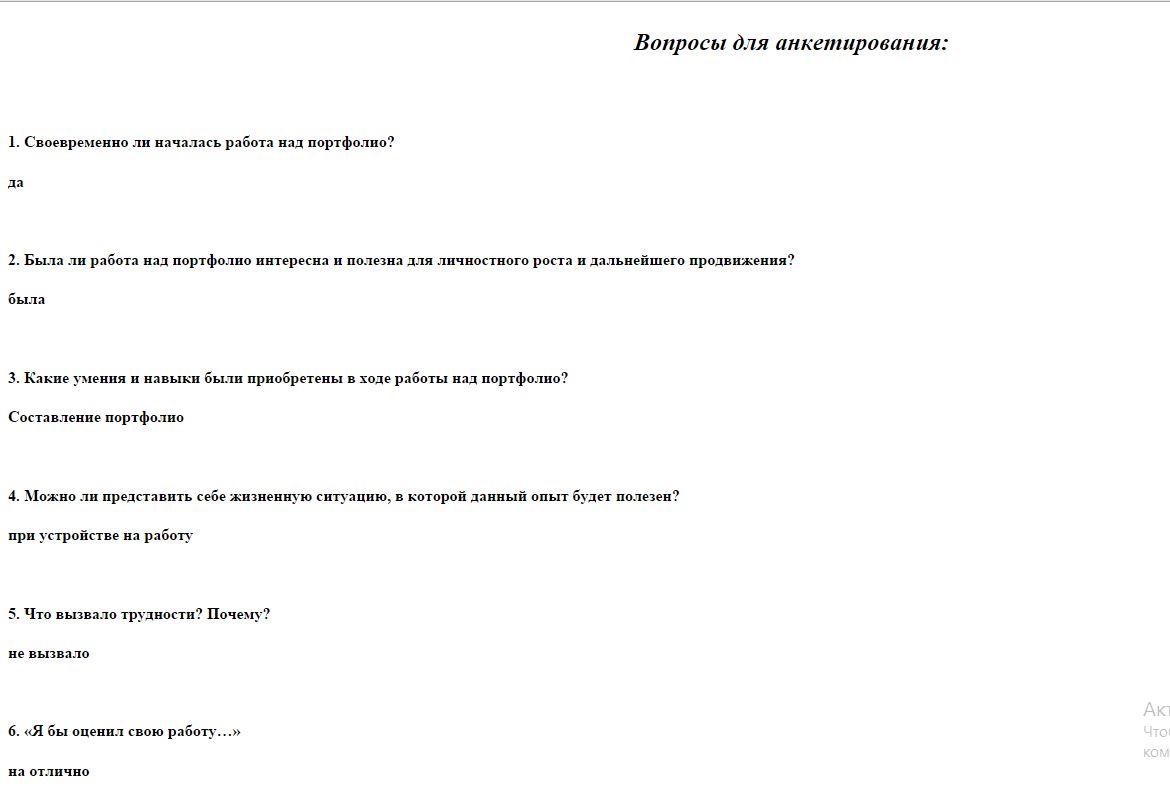


Рис.27. Вывод ответов на вопросы по анкетированию

После чего, пользователю необходимо выделить всю информацию, скопировать ее, зайти в программу Microsoft Office Word, создать новый документ и вставить все скопированные данные, после чего немного их отформатировать.

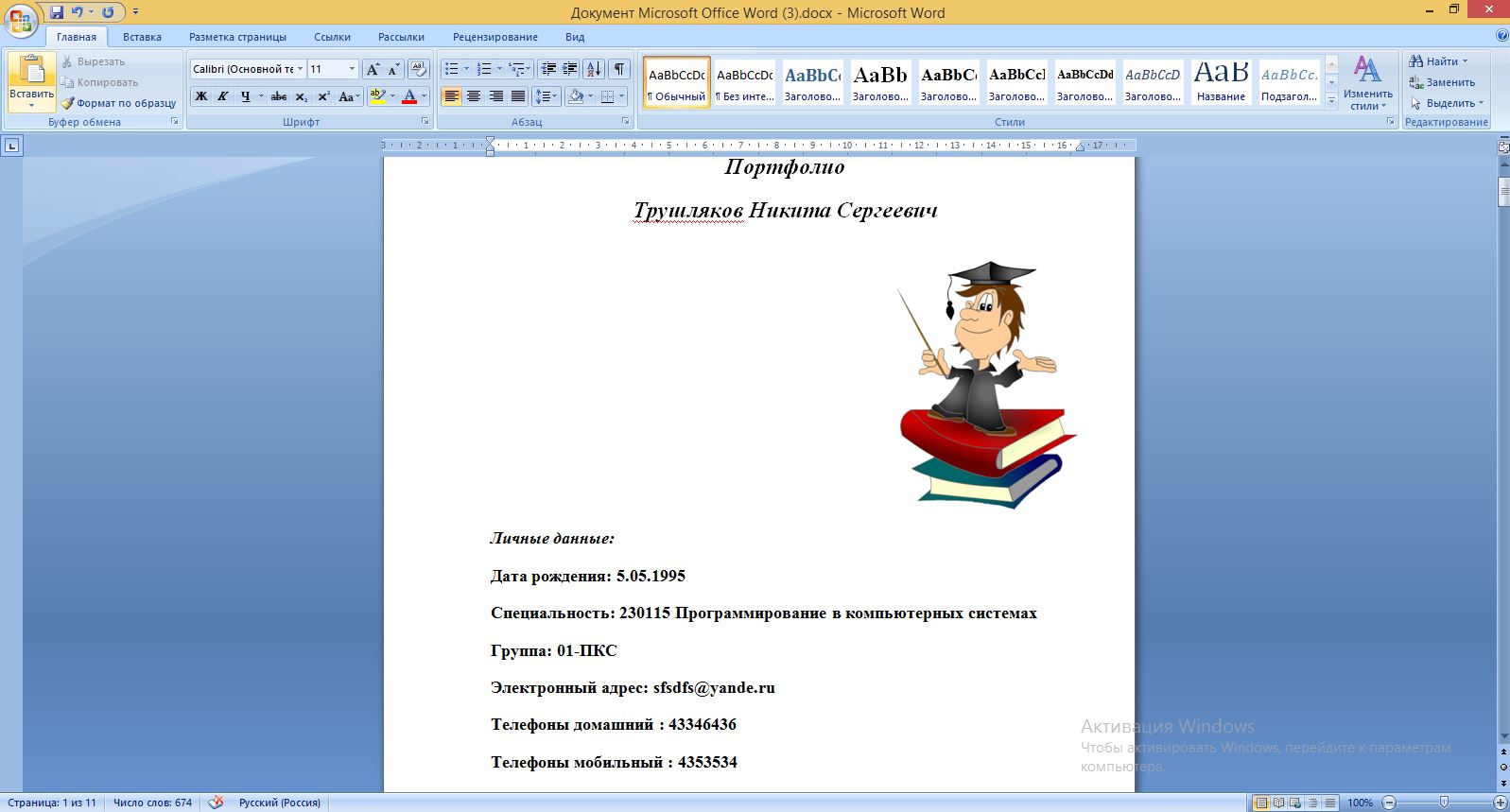


Рис.27. Перенос данных с сайта в программу Microsoft Office Word

3.5 Оценка результатов выполнения задачи

В результате работы над проектом был создан макет электронного портфолио для студента в форме сайта. Использование электронного портфолио позволит продемонстрировать процесс обучения студента, его достижения относительно ФГОС СПО нового поколения, умение размышлять над своими успехами неудачами. Это своеобразный отчет по различным видам деятельности студента: учебная, научно-исследовательская, творческая, практическая, общественная и т.д.

.

4 Организация производс**тва**

4.1 Организация работ по внедрению программного продукта заказчика

Программный продукт составлен в соответствии с заданием Волгоградского Колледжа Управлений и Новых технологий на дипломный проект по специальности 09.02.03 (230115) «Программирование в компьютерных системах». Разработанный макет электронного портфолио отвечает всем требованиям, поставленным перед его разработкой и может успешно применяться в учебном процессе. Разработанный макет портфолио был успешно опробован в группах, обучающихся по специальности 09.02.03 (230115) Программирование в компьютерных системах.

5 Заключение

В результате работы над проектом был создан макет электронного портфолио для студента в форме сайта. Сайт был разработан на языке HTML и PHP. Он позволяет автоматизации процесса создания портфолио, экономии времени студента при создании и заполнении портфолио, улучшения оформления и восприятия портфолио

Электронное портфолио (е-портфолио) является современной педагогической технологией, которая провоцирует студента на самостоятельную деятельность, как в образовательном процессе, так и в выстраивании будущей профессиональной карьеры.

Электронное портфолио студента позволяет формировать образовательный запрос через:

* визуализацию результатов образовательной деятельности
* пробно-поисковые действия
* расширение границ за пределы пространства образовательного учреждения
* создание системы оценивания, значимой для самого студента.

Каждое портфолио уникально, так как является продуктом конкретного обучающегося. Сама технология е-портфолио является мощным катализатором для формирования информационной ключевой компетентности студента.

5.1 Перечень использованной литературы