

**УТВЕРЖДАЮ**  
Директор  
ООО «Крымская компьютерная академия «ШАГ»  
\_\_\_\_\_ А. Н. Глухова  
« \_\_\_\_ » \_\_\_\_\_ 2017 года

**Дополнительная образовательная программа  
«Веб-разработка»**

Симферополь  
2017

## Общие положения

- категория слушателей – от 15 до 55 лет (включительно).
- форма обучения – очная, возможны индивидуальные и групповые занятия
- срок обучения – 140 (астрономических)
- режим занятий – обучение проходит по рабочим дням 3 раза в неделю по 2 астрономических часа
- выдаваемый итоговый документ – по завершении обучения выдается Документ установленного образца

## 1. Пояснительная записка

### 1.1. Актуальность

Несмотря на то, что интернет давно и прочно вошел в нашу жизнь, многие предприниматели и даже крупные фирмы не понимают, что им даст создание собственного сайта, ведь есть другие хорошо зарекомендовавшие себя проверенные способы саморекламы: телевидение, радио, СМИ, баннеры, флайеры и тому подобное.

У любой современной компании существует сайт. Это один из элементов престижа, ведь именно в Интернете потенциальные клиенты будут в первую очередь искать информацию о фирме. И если у нее нет хотя бы одностраничника с прайсом, это покажется подозрительным – насколько же это неуспешная фирма, если не может даже небольшой веб-ресурс создать?

Актуальность создания сайта состоит также в том, что если вы хотите донести информацию максимально быстро до огромного количества людей, то лучше, чем с помощью собственного сайта сделать это не получится никак. Веб-ресурс позволяет представить информацию о компании и ее товарах или услугах сжато и одновременно полноценно. Также сайт может сообщать о новостях фирмы, об изменениях в прайсе или режиме работы, содержать отзывы благодарных клиентов.

### 1.2. Цель курса:

Обучить слушателя созданию и верстке статических web-страниц с использованием технологий XHTML1.0, HTML4/5, CSS2/3, JavaScript, Php/MySQL. Сложить для слушателя целостное представление о технологической цепочке создания web-сайтов и сформировать понимание актуальных тенденций развития web-технологий. Научить слушателя выбирать наиболее подходящий способ для создания web-страниц. Научить тестировать и проверять код web-страниц.

### 1.3. Задачи курса:

- Развить алгоритмический и логический стили мышления;
- научить применять теоретические знания при создании компьютерных программ;
- обучить основам HTML;
- обучить основам разработки клиентских сценариев с использованием JavaScript;
- обучить разработке веб-приложений на платформе PHP с использованием СУБД MySQL;
- научить самостоятельно практически решать поставленные задачи и работать с электронными справочниками.

## 1.4. Ожидаемые результаты:

В результате освоения курса слушатель должен знать:

- Знать и уметь применять основы HTML — теги, атрибуты и способы структурирования содержимого web-страниц для создания форматированных документов;
- Знать и уметь применять основы CSS — значения, списки, цвета, шрифты и другие метрики форматирования;
- Знать основы HTML5 и CSS3.
- Знать ООП и его основные понятия;
- Отличия BOM и DOM;
- Особенности применения HTML5 по отношению к JavaScript;
- Основы взаимодействия с AngularJS;
- Особенности реализации механизмов ООП в PHP;

В результате освоения курса слушатель должен владеть:

- Навыками проверки и отладки кода web-документов;
- Навыками формирования содержимого web-документов для различных экранов — от стандартных браузеров до мобильных устройств;
- Навыками быстрого и качественного форматирования сложных web-документов.
- Базовыми конструкциями языка JavaScript такими как переменные, условия, циклы, строки, массивы функции, и т.д.;
- Навыками формирования содержимого web-документов для различных экранов — от стандартных браузеров до мобильных устройств;
- Навыками быстрого и качественного форматирования сложных web-документов;
- Принципами создания форм и анализа данных пользователя с использованием регулярных выражений;
- Принципами создания асинхронных запросов при помощи Ajax;
- базовыми конструкциями библиотеки jQuery;
- Способами внедрения анимации с использованием jQuery.

В результате освоения курса слушатель должен уметь:

- Обрабатывать возникающие ошибки;
- Разбираться в понятиях событие, обработчик события;
- Создавать функции-обработчики различных событий;
- Взаимодействовать с объектами из BOM и DOM;
- Разбираться в тонкостях реализации клиентских сценариев под разные браузеры;
- Сохранять пользовательские данные с помощью механизма cookie;
- Сериализовать и парсить данные используя JSON;
- Применять механизмы jQuery для работы с Ajax;
- Подключать и использовать jQuery плагины;
- Разбираться в тонкостях построения веб-приложений с использованием PHP;
- Владеть стандартными функциями PHP;
- Применять регулярные выражения;
- Сохранять данные пользователя в файлах cookies;
- Внедрять Ajax в веб-приложения.

## 2. Учебно-тематический план

№ темы	Наименование темы	Кол-во занятий	Кол-во астрон.
--------	-------------------	----------------	----------------

			часов
1.	Разработка веб-страниц на языке разметки HTML5 с использованием каскадных таблиц стилей CSS3	20	40
2.	Разработка клиентских сценариев с использованием JavaScript	16	32
3.	Создание web-приложений, исполняемых на стороне сервера при помощи языка программирования PHP, СУБД MySQL и технологии AJAX	34	68
	<b>Итого</b>	70	140

### 3. Содержание программы

#### Тема №1. Разработка веб-страниц на языке разметки HTML5 с использованием каскадных таблиц стилей CSS3.

##### 1. Введение в Web-технологии. Структура HTML.

- Введение в предмет.
- Введение в языки разметки. Язык разметки гипертекста HTML.
  - Развитие HTML, версии. Текущие используемые версии: HTML и XHTML.
  - Вопросы межбраузерной совместимости. Война браузеров.
  - W3C.
- Теги – основной элемент структуры HTML. Правила записи тегов и их атрибутов в стандарте XHTML на примере тегов <b>, <i>, <u>, <font>, <sup>, <sub>, <br />. Синтаксические отличия HTML4 от XHTML.
- Основные ошибки в записях тегов.
  - Спецификации <!DOCTYPE HTML>.
  - Валидация документа при помощи FireFox – дополнение HTML Validator.
  - Понятие well-formed.
  - Прародители HTML4/5 и XHTML: SGML и XML.
- Основная структура XHTML документа. Основные элементы и их назначение.
- Кодировки страницы и теги <meta>:
  - Применение тега <meta> – задание информации о странице (expires, refresh, autor, copyright, keywords, description).
  - Задание кодировки страницы при помощи тега <meta>.
  - Символьные подстановки и кодировки.

##### 2. Форматирование текста при помощи HTML.

- Классификация тегов: линейные и блочные.
  - Линейные: <b>, <i>, <u>, <font>.
  - Блочные: <p>, <h1>..<h6>.
- Модель форматирования текста: заголовки и абзацы. Элементы <p>, <h1>..<h6>. Выравнивание текста в блочных элементах: атрибут align.
- Классификация тегов: логическое и физическое форматирования.
  - Теги физического форматирования: <b>, <i>.
  - Теги логического форматирования: <strong>, <em>. Их отличие.
  - Краткий обзор основных тегов логического форматирования: <abbr>, <acronym>, <cite>, <code>, <del>, <dfn>, <ins>.
- Цвета в Web.
  - Web-палитра.

- Photoshop/GIMP – дополнительные инструменты верстальщика. Рассмотрение основных возможностей: открытие изображений, подбор цветов при помощи палитры Photoshop|GIMP, инструмент пипетка.
- Использование выбранного цвета в атрибуте color тега <font>.
- Практика: создание простейшей web-страницы.

### 3. Форматирование при помощи CSS.

- CSS – каскадные таблицы стилей.
  - Введение. Обзор версий. Назначение: HTML служит для задания структуры, CSS – для форматирования.
  - Встраивание CSS в HTML при помощи атрибута style. Правила записи CSS свойств.
- Теги без форматирования <div> – блочный, <span> – линейный.
- Аналогия HTML и CSS на примере линейных и блочных тегов:
  - Тег <font> – свойства color, font-size, font-family.
  - Тег <b> – свойства font-weight.
  - Тег <i> – свойства font-style.
  - Тег <u> – свойства text-decoration.
  - Тег <sup>, <sub> – свойства vertical-align.
  - Атрибут align – свойства text-align.
  - Сокращенная запись свойства font.
  - Дополнительные свойства CSS для форматирования текста: letter-spacing, line-height, text-indent, text-transform, white-space, word-spacing.
- Использование атрибутов class и id для задания стилей.
  - Создание стилей для тегов, классов, идентификаторов внутри тега <style>. Понятие селекторов. Правило записи селекторов: селектор тегов, селектор классов, селектор идентификаторов, универсальный селектор \*.
  - Приоритет использования стилей (tag / class / id / style). Повышение приоритета правилом !important.
  - Наследуемость стилей. Стандартные значения свойств.
  - Отслеживание стилей при помощи средства разработки firebug (дополнение для Firefox)
- Использование внешних CSS файлов стилей.
  - Подключение CSS файлов при помощи тега <link> и инструкции @import.
  - CSS файлы и кэш браузера.
- Практика: форматирование текста при помощи CSS.

### 4. Списки. CSS отступы и поля.

- Создание списков
  - Неупорядоченные списки: элементы <ul>, <li>.
  - Упорядоченные списки: элементы <ol>, <li>.
  - Атрибуты type, value, start.
- Создание вложенных списков.
- Форматирование списков при помощи CSS.
  - Свойства list-style-type, list-style-image, list-style-position.
  - Сокращенная запись свойства list-style.
  - Оформление многоуровневых списков. Вложенные селекторы.
- Списки определений: элементы <dl>, <dd>, <dt>.
- Управление отступами и полями.
  - Свойство margin и его потомки margin-left, margin-top, margin-right, margin-bottom.
  - Свойство padding и его потомки padding-left, padding-top, padding-right, padding-bottom.
  - Отличие padding от margin и их назначения.

- Отмена отступов по умолчанию у некоторых тегов: <body>, <h1>..<h6>, <p>.
- Практика: создание списков.

## 5. Графика в web-дизайне. Оптимизация графики.

- Форматы графических файлов в Web.
- Рассмотрение инструментов Photoshop/GIMP для работы с изображениями.
  - Слои в Photoshop/GIMP.
  - Прямоугольное выделение. Линейки. Направляющие.
  - Оптимизация изображений в Photoshop.
- Тег <img /> и его атрибуты (src, alt, width, height, border).
  - Свойство border – аналог атрибута border.
  - Задание свойств margin, padding, border для изображения.
  - Выравнивание изображений на странице при помощи атрибута align. Аналог атрибута align – свойство float.
- Фон страницы – свойство background.
  - Задание фона в виде цвета: background-color. Обязательное задание фона для элемента <body>.
  - Задание фона в виде изображения: background-image, background-repeat, background-position, background-attachment.
  - Изображения и кэш браузера.
- Спрайты: меньше картинок – больше скорость. Создание спрайтов при помощи онлайн сервисов (spritegen.website-performance.org, csssprites.com, printf.ru/spritr).
- Практика: проектирование web-страниц с использованием графики.

## 6. Гиперссылки. Принципы навигации web-сайта.

- Общие сведения о гиперссылках.
  - Тег <a> и его атрибуты (href, target).
  - Эргономика, удобство навигации.
- Абсолютная и относительная адресация.
  - Организация внешних ссылок.
  - Организация внутренних ссылок с помощью элемента <a>. Атрибуты id и name.
  - Организация "смешанного" перехода (на указанный элемент во внешнем HTML-документе).
  - Графические ссылки. Отмена границ у ссылок.
- Создание меню при помощи структуры списков (<ul>, <li>), его форматирование. Свойство display. Преобразование ссылки в блочный элемент.
- Псевдоклассы.
  - Псевдоклассы ссылок: active, hover, link, visited.
  - Псевдоклассы для обычных элементов: first-child, first-line, first-letter.
- CSS свойство cursor.
- Практика: работа по разработке галереи изображений.

## 7. Таблицы.

- Создание простейшей таблицы. Теги <table>, <tr> и <td>.
  - Атрибуты border, cellspacing cellpadding. Их возможные аналоги CSS: border, padding.
  - Указание ширины и высоты ячейки: атрибуты width, height. Правила задания ширины и высоты. Аналоги CSS: свойства width, height.
  - Выравнивание данных в таблице: атрибуты align и valign. Аналоги CSS: свойства text-align, vertical-align.
  - Управление цветом фона и цветом рамок таблицы (отдельной строки, отдельной ячейки).

- Использование изображений в качестве фона таблицы (отдельной строки, отдельной ячейки).
- Объединение ячеек: атрибуты `colspan`, `rowspan`.
- Теги логического структурирования таблиц: `<thead>`, `<tbody>`, `<tfoot>`. Теги логического группирования столбцов: `<colgroup>`, `<col>`.
- Управление рамками таблицы: атрибуты `frame`, `rules`.
- Практика: создание сложных таблиц.
- Основы табличной верстки. Пример табличной верстки: ее минусы.

## 8. Позиционирование. Верстка web-страниц блоками.

- Свойство `position`.
  - Рассмотрение позиционирования: `relative` и `absolute`.
  - Свойства `top`, `left`, `bottom`, `right`.
- Свойства `visibility`, `overflow`.
- Практика.
- Основы верстки блоками. Правила верстки.
  - Вложение блоков.
  - Задание ширины и высоты блокам при помощи свойства `width` и `height`.
  - Обтекание блоков. Отмена обтекания блоков. Свойства `float` и `clear`.
  - Правила задания отступов и полей.
  - Задание минимальной высоты и ширины блока: свойства `min-height`, `min-width`.
  - Выравнивание внутри блоков (`margin`, `text-align`, `line-height`, `position`).  
Кроссбраузерность выравниваний.
- Рассмотрение простейших структур страниц.
  - Структура фиксированного размера.
- Резиновая структура. Блоки с отрицательными `margin`.

## 9. Формы. Фреймы.

- Введение в формы.
- Управляющие элементы форм.
  - Кнопки (отправки, сброса, пр.).
  - Флажки.
  - Кнопки с зависимой фиксацией (радиокнопки).
  - Всплывающие списки.
  - Текстовый ввод.
  - Выбор файлов.
  - Скрытые управляющие элементы.
- Создание форм при помощи HTML:
  - Элемент `<form>`.
  - Элемент `<input>`.
  - Элемент `<button>`.
  - Элементы `<select>`, `<optgroup>` и `<option>`.
  - Элемент `<textarea>`.
  - Метки `<label>`.
  - Структура форм: `<fieldset>` и `<legend>`.
- Форматирование элементов форм при помощи CSS.
- Фреймы и их структура (теоретические сведения).
  - Тег `<iframe>`.
  - Использование Спецификации `<!DOCTYPE HTML>` для фреймов.
  - Вредность использования фреймов.

## 10. Первичная оптимизация сайта. Размещение сайта в сети Internet.

- Оптимизация сайта под поисковые системы.
  - Задание ключевых слов и описания при помощи тега `<meta>`.
  - Важность использования заголовков `<h1>..<h6>` и логических тегов `<strong>` и `<em>`.
  - Важность использования уникального содержимого.
- Хостинг, регистрация доменного имени.
- Размещение сайта в Internet. Работа с FTP-клиентами.
- Регистрация в поисковых системах и каталогах. Размещение счетчика на web-странице.

## 11. Создание страниц посредством HTML5 и CSS3.

- Структура HTML5 документа.
  - Новые теги задания структуры: `<header>`, `<nav>`, `<section>`, `<article>`, `<aside>`, `<footer>`. Доступность новых тегов в современных браузерах. Отображение новых тегов в устаревших браузерах.
- HTML5 – конкурент Flash.
  - Вставка видео на странице посредством тега `<video>`.
  - Вставка аудио на странице посредством тега `<audio>`.
  - Создание изображений и анимации посредством тега `<canvas>`.
  - Использование SVG формата.
- Новые элементы форм.
- Новые свойства: CSS3.
  - Работа с фоном: создание градиентов, изменение размеров фона – свойства `background` и `background-size`.
  - Работа с границами: скругленные края у блоков – свойства `border-radius`.
  - Задание полупрозрачности элементам страниц – свойство `opacity`.
  - Полная поддержка селекторов CSS 2.1.

## Тема №2. Разработка клиентских сценариев с использованием JavaScript

### 1. Введение в JavaScript

- Сценарии, выполняемые на стороне клиента
- Что такое JavaScript?
- История создания JavaScript
- Различия между JavaScript и Java, JScript, ECMAScript
- Версии JavaScript
- Понятие Document Object Model
- Понятие Browser Object Model
- Внедрение в HTML документы. Редакторы кода JavaScript
- Тег `<noscript>`
- Основы синтаксиса
  - Регистрозависимость
  - Комментарии
  - Ключевые и зарезервированные слова
- Переменные. Правила именования переменных
- Типы данных
- Операторы
  - Арифметические операторы
  - Операторы отношений
  - Логические операторы
  - Оператор присваивания
  - Битовые операторы
  - Приоритет операторов
  - Оператор `typeof`
- Ввод/вывод данных. Диалоговые окна



- Условия
  - Что такое условие?
  - if
  - if else
  - Тернарный оператор ?:
  - switch
- Циклы
  - Что такое цикл?
  - while
  - do while
  - for
  - break
  - continue
  - Понятие метки
- Что такое функция?
  - Синтаксис объявления функции
  - Параметры функции
  - Возвращаемое значение функции. Ключевое слово return
- Объект arguments
  - Цель и задачи объекта
  - Свойство length
- Область видимости переменной. Ключевое this
- Рекурсия

## 2. Объект. Массивы. Введение в объектно-ориентированное программирование

- Объекты
  - Что такое объект?
  - Введение в объектный тип данных
  - Объект Object
  - Ключевое слово new
  - Понятие свойства
  - Добавление свойств. Синтаксис добавления свойств.
  - Синтаксис обращения к свойствам
- Массивы
  - Что такое массив?
  - Объект Array
  - Создание массива
  - Обращение к элементам массива
  - Свойства и методы Array
- Строки
  - Объект String
  - Свойства и методы String
- Задержки и интервалы. Периодический вызов функций
- Объект Date. Обработка даты и времени
- Объект Math. Свойства и методы. Случайные числа
- Что такое ООП?
- Три фундаментальных принципа ООП
  - Инкапсуляция
  - Наследование
  - Полиморфизм
- Понятие класса и объекта в терминах JavaScript
- Свойства

- Методы
- Свойства-аксессоры
  - get – свойства (геттеры)
  - set – свойства (сеттеры)
- Конструктор
- Понятие prototype
  - Что такое prototype
  - Цели и задачи prototype
- Наследование

### **3. Обработка событий, Browser Object Model, Document Object Model.**

- Что такое событие?
- Что такое обработчик события?
- Обработка событий в сценариях
- Управление стилями элементов web-страницы
- Объект event и его свойства
- Обработчики событий по умолчанию (стандартные обработчики), запрет вызова стандартного обработчика
- Объект Image. Управление рисунками и ролловерами
- Что такое Browser Object Model?
- Объекты Browser Object Model
  - Объект Window. Открытие, перемещение и изменение размера окон
  - Объект Navigator. Управление браузером
  - Объект Screen. Свойства экрана
  - Объекты Location и History. Перемещение по страницам.
  - Коллекция Frames. Управление фреймами.
- Что такое Document Object Model?
- Отличия DOM от BOM
- Представление HTML-документа в виде дерева
- Объекты модели DOM. Иерархия узлов
- Свойства и методы модели DOM. Модель событий DOM
- Изменение дерева DOM
- Знакомство с объектами Document и Link
- Управление выделением и текстовым диапазоном: объекты Selection и TextRange
- Особенности DOM в HTML5

### **4. Формы, Проверка достоверности форм. Использование Cookie**

- Применение форм. Размещение элементов формы в HTML.
- Коллекция Forms. Создание и программирование элементов формы:
  - Кнопки: элементы Button, Submit, Reset
  - Текстовые поля: элементы Text, Password, File Upload, Textarea
  - Скрытое поле формы: общее понятие об элементе Hidden
  - Флажок: элемент Checkbox
  - Переключатель: элемент Radio
  - Список: элементы Select, Option
- Объект RegExp. Правила записи регулярных выражений
- Методы объектов String и RegExp для работы с регулярными выражениями
- Проверка достоверности данных формы
- Что такое cookie?
- Преимущества и недостатки cookie

- Создание, использование и удаление cookie

## 5. Рисование с помощью canvas, HTML5 и JavaScript

- Что такое canvas?
- Базовые возможности
  - Заливка
  - Операции с графическими примитивами. Рисование точек, линий, прямоугольников, кругов, кривых Безье и т.д.
  - Вывод текста
  - Вывод изображений
  - Работа с тенями и градиентом
- Cross-document messaging или XDM
  - Цели и задачи XDM
  - Отправка сообщений. Метод postMessage
  - Получение сообщений
- Drag and Drop
  - Поддержка drag and drop в различных браузерах
  - События, возникающие при drag and drop
  - Объект dataTransfer.
    - Методы объекта dataTransfer
    - Свойства dropEffect и effectAllowed
  - Свойство draggable
- Поддержка медиа возможностей
  - Использование тега <video>
  - Использование тега <audio>

## 6. JSON, Ajax

- Что такое JSON?
- Цели и задачи JSON
- Синтаксис JSON
  - Переменные
  - Объекты
  - Массивы
- Объект JSON
  - Что такое сериализация?
  - Что такое парсинг?
  - Методы stringify и parse
- Настройка пользовательской сериализации в JSON. Метод toJSON
- Синхронные и асинхронные запросы
- Что такое Ajax?
- Объект XMLHttpRequest
  - Создание через ActiveX объект
  - Создание через объект XMLHttpRequest
- Методы и свойства XMLHttpRequest
- Понятие HTTP заголовка
- Использование метода GET. URL кодирование
- Использование метода POST

## 7. Введение в jQuery, События и jQuery

- Что такое jQuery?
- Цели и задачи jQuery
- История создания jQuery

- Версии jQuery
- Подключение jQuery
- Доступ к элементам страницы при помощи функции \$
- Понятие селектора
- Типы селекторов
  - CSS селекторы
  - jQuery селекторы
- Traversing. Методы обхода DOM. Метод filter, next, nextAll, prev, prevAll, siblings и др.
- Создание обработчиков событий с использованием jQuery
- Удаление обработчиков событий
- Объект Event и jQuery
- Воздействие на обработку события
- Запуск обработки события

## **8. Стили и анимация, взаимодействие с DOM, AJAX и jQuery, использование jQuery плагинов**

- Метод css
- Отображение и скрытие элементов. Методы show и hide
- Создание эффектов
- Анимация
- Создание новых элементов DOM
- Вставка элементов DOM
- Передвижение элементов DOM
- Копирование элементов DOM
- Взаимодействие с атрибутами
- JSON
- Механизмы Ajax внутри библиотеки jQuery
- Использование метода GET
- Использование метода POST
- События и Ajax в рамках jQuery
- Обработка ошибок
- Понятие плагина jQuery
- Подключение плагина
- Примеры плагинов
  - Cycle
  - jQuery UI

## **9. AngularJS**

- Что такое AngularJS?
- Цели и задачи AngularJS
- История появления AngularJS
- Как добавить AngularJS в веб-страницу
- Концепция MVC (Model-View-Controller)
- Контроллер
- Что такое контроллер?
- Что такое модуль?
- Что такое директива?
- Что такое вид?
- Что такое фильтр?

- Основы взаимодействия с контроллером
- Использование модулей
- Использование
  - ng-model
  - ng-click
  - ng-repeat
  - ng-show
  - ng-hide
  - ng-include
- Что такое сервис?
- Зачем нужно использовать сервисы?
- Примеры использования сервисов
- Роутинг
  - Что такое роутинг?
  - Настройка роутинга
  - Примеры использования

### **Тема №3. Создание web-приложений, исполняемых на стороне сервера при помощи языка программирования PHP, СУБД MySQL и технологии AJAX**

#### **Введение в web-программирование на PHP**

1. Введение в web-программирование и принципы работы web-приложений
  - 1.1 Обоснование и истоки возникновения. Отличия серверного web-программирования от клиентского. Цели, задачи, направление развития, краткая история.
  - 1.2 Архитектура «клиент-сервер». Выполнение серверных сценариев. CGI. Взаимодействие с СУБД.
    - 1.3 Принципы и этапы загрузки web-страницы.
2. Обзор и сравнительный анализ существующих серверных web-технологий:
  - 2.1 ASP.NET (ASP)
  - 2.2 JSP (sun)
  - 2.3 Ruby on Rails (RoR)
  - 2.4 ColdFusion (www.cffaq.com)
  - 2.5 Python
3. Введение в PHP.
  - 3.1 Что такое PHP и область применения.
  - 3.2 История возникновения PHP. Обзор релизов PHP. Отличия PHP5 и PHP6.
  - 3.3 Информационные ресурсы, документация.
4. Описание и обзор инструментария для разработчика: web-серверы, СУБД.
5. Подготовка инструментария.
  - 5.1 Установка и настройка web-сервера Apache под Windows.
  - 5.2 Обзор директорий и файлов установленного сервера (модули и т.д.).
  - 5.3 Конфигурационный файл httpd.conf.
  - 5.4 Создание виртуальных хостов.
  - 5.5 Установка PHP для web-серверов Apache, IIS и первичная настройка.
  - 5.6 Обзор директорий и файлов. Конфигурационный файл php.ini.
- 5.7 Краткий обзор отличий установки под Windows и под UNIX.
- 5.8 Обзор и сравнение редакторов кода PHP: Eclipse PDT, Zend Studio, PHP Expert Editor, DreamWeaver, PHPStorm
6. Формы включения PHP кода внутрь страницы.
  - 6.1 Полные теги
  - 6.2 Короткие теги
  - 6.3 Теги ASP
  - 6.4 Теги script.

7. Разделение выражений. Комментарии.
8. Правила именования переменных. Константы.
9. Типы данных.
  - 9.1 Целочисленный, вещественный, логический.
  - 9.2 Строковый, массивы, объектный, ресурсный.
  - 9.3 Преобразование типов данных.
  - 9.4 Семейство функций is..() (is\_int, is\_float,...).
10. Операторы и операнды.
  - 10.1 Арифметические и логические операторы.
  - 10.2 Операторы присваивания.
11. Условные конструкции
  - 11.1 Конструкция if
  - 11.2 Конструкция switch
12. Циклические конструкции
  - 12.1 Цикл while
  - 12.2 Цикл do...while
  - 12.3 Цикл for
  - 12.4 Досрочное прерывание итераций: break, continue
13. Массивы.
  - 13.1 Массивы в PHP. Особенности строения массивов.
  - 13.2 Ассоциативные и индексные (списки) массивы. Способы инициализации. Конструкция array.
  - 13.3 Одномерные и многомерные массивы. Отличия многомерных массивов от одномерных. Способы инициализации.
  - 13.4 Способы перебора массивов. Конструкции list, each. Цикл foreach.

## **Работа с формами, функции**

1. Работа с формами. Способы связывания XHTML-формы и PHP-скрипта.
  - 1.1 Работа форм. Описание протокола http.
  - 1.2 Методы GET и POST. Структура и назначение. Совмещение методов.
  - 1.3 Глобальные переменные. Настройка файла php.ini. Суперглобальные массивы \$\_POST, \$\_GET, \$\_REQUEST.
    - 1.4 Передача различных элементов форм в скрипт и их обработка.
    - 1.5 Форма и её обработчик в одном сценарии.
2. Функции.
  - 2.1 Синтаксис функций и примеры объявления.
  - 2.2 Передача параметров по значению и по ссылке.
  - 2.3 Области видимости переменных. Локальные и глобальные переменные.
  - 2.4 Статические переменные.
  - 2.5 Условные функции.
  - 2.6 Объявление функции внутри функции.
  - 2.7 Функции с параметрами по умолчанию.
  - 2.8 Функции с бесконечным количеством параметров.
  - 2.9 Использование функций func\_num\_args(), func\_get\_arg(), func\_get\_args().
3. Создание библиотек.
  - 3.1 Инstrukция require.
  - 3.2 Инstrukция include.
  - 3.3 Инstrukция require\_once.
  - 3.4 Инstrukция include\_once.
4. Функции для работы с массивами.
  - 4.1 Функции сортировки массива (sort, asort, ksort, krsort, ...).
  - 4.2 Поиск в массиве.
  - 4.3 Слияние и разделение массивов.
- 4.4 Другие полезные функции для массивов (array\_sum, array\_fill, array\_walk, array\_unique, ...).

5. Математические функции.
  - 5.1 Округление.
  - 5.2 Вычисление чисел с произвольной точностью.
  - 5.3 Генерация случайных чисел.
6. Функции для работы со временем.
  - 6.1 Получение и конвертация дат.
  - 6.2 Сравнение дат.
7. Функции для работы со строками.
  - 7.1 Вывод строк в браузер. Сериализация.
  - 7.2 Поиск, замена, урезание и разбор строк.
  - 7.3 Функции для работы с XHTML. Кодировки.
  - 7.4 Другие функции. Шифрование.

## **ООП, регулярные выражения**

1. Объектно-ориентированное программирование в PHP.
  - 1.1 Основные концепции ООП (инкапсуляция, полиморфизм, наследование) и их реализация в PHP. Анализ отличий аппарата ООП в PHP4 и PHP5.
    - 1.2 Синтаксис объявления классов.
  - 1.3 Поля и методы класса. Свойства класса. Спецификаторы доступа. Константы.
    - 1.4 Конструкторы и деструкторы.
    - 1.5 Клонирование объектов.
    - 1.6 Статические свойства и методы класса.
    - 1.7 Метод `__toString()`.
    - 1.8 Наследование и перегрузка методов.
    - 1.9 «Волшебные» (magic) методы `__sleep()`, `__wakeup()` и др.
    - 1.10 Оператор `instanceof`.
    - 1.11 Финальные классы и методы.
    - 1.12 Абстрактные классы и методы.
    - 1.13 Интерфейсы.
  - 1.14 Функции для работы с классами: `class_exists()`, `get_class_methods()`, `get_class_vars()`, `get_declared_classes()`, `get_declared_interfaces()`, `get_object_vars()`, `get_parent_class()`, `is_a()`, `is_subclass_of()`, `method_exists()`
2. Поддержка регулярных выражений в PHP.
  - 2.1 Системы регулярных выражений PERL и POSIX.
  - 2.2 Регулярные выражения PERL. Синтаксис PERL совместимых выражений. Примеры регулярных выражений.
    - 2.3 Функции для использования PERL совместимых регулярных выражений.

## **Работа с файлами, ошибки и исключения, cookies и сессии**

1. Принципы взаимодействия с файлами и директориями средствами PHP.
  - 1.1 Понятие текстового и двоичного файла.
  - 1.2 Открытие и закрытие файла. Прямая работа с файлами.
  - 1.3 Чтение данных из файла. Запись данных в файл.
  - 1.4 Позиционирование по файлу.
  - 1.5 Дополнительные функции для работы с файлами: `filemtime()`, `filesize()`, `filetype()`, `ftruncate()`, `is_file()`, `is_writable()`, `basename()`.
    - 1.6 Копирование, удаление и переименование файлов.
    - 1.7 Функции для работы с директориями.
  - 1.8 Взаимодействие с операционной системой и использование PHP в командной строке.
    - 1.9 Конфигурация PHP для работы с загрузкой файлов на сервер.
    - 1.10 Простая загрузка. Использование массива `$_FILES`
    - 1.11 Множественная загрузка.
2. Ошибки и исключения.

- 2.1 Регулирование вывода сообщений об ошибках и настройка конфигурационного файла.
- 2.2 Изменение стандартного обработчика ошибок. Оператор отключения сообщений об ошибках. Отладка скриптов.
  - 2.3 Каскадная обработка исключений.
3. Cookies
  - 3.1 Сравнение подходов к хранению пользовательской информации. Анализ клиентского подхода (cookies) и серверного подхода (сессии).
  - 3.2 Использование "cookies". Установка "cookies". Функция setcookie. Использование массива \$\_COOKIE.
  - 3.3 Удаление "cookies". Проверка поддержки "cookies".
4. Сессии
  - 4.1 Сессии. Два подхода к использованию сессий. Использование cookies, URL для хранения id сессии.
  - 4.2 Настройка файла php.ini для сессий. Функции для использования сессий. Глобальный массив \$\_SESSION.

## Взаимодействие с СУБД MySQL

1. Обзор возможностей PHP по поддержке работы с разными СУБД.
2. Административные возможности СУБД MySQL.
  - 2.1 История развития и инсталляция СУБД MySQL.
  - 2.2 Основы взаимодействия с СУБД MySQL. Отличия различных веток MySQL. Особенности диалекта SQL.
  - 2.3 Приложение phpMyAdmin
- 2.4 Поддержка расширенных конструкций SQL. Представления. Хранимые процедуры.
  - 2.5 Функции. Триггеры. Встроенные функции MySQL.
  - 2.6 Управление пользовательскими учетными записями в СУБД MySQL.
  - 2.7 Файлы журналов.
- 2.8 Поддержка и восстановление баз данных. Проверка таблиц на наличие ошибок.
- 2.9 Резервирование и копирование баз данных. Методы резервирования. Восстановление таблиц.
3. Взаимодействие PHP и MySQL.
  - 3.1 Библиотека mysql. Функции: mysql\_connect(), mysql\_pconnect(), mysql\_select\_db(), mysql\_close(), mysql\_query(), mysql\_result(), mysql\_db\_name(), mysql\_errno(), mysql\_error(), mysql\_fetch\_array(), mysql\_fetch\_assoc(), mysql\_fetch\_row(), mysql\_field\_flags(), mysql\_field\_len(), mysql\_field\_type(), mysql\_free\_result(), mysql\_list\_dbs(), mysql\_list\_fields(), mysql\_list\_tables(), mysql\_num\_fields(), mysql\_num\_rows()
  - 3.2 Библиотека mysqli.
  - 3.3 Процедурный стиль работы с mysqli.
  - 3.4 Объектный стиль работы с mysqli.
4. Библиотека SQLite.
5. Расширение PHP Data Objects.
6. Работа с графикой.
  - 6.1 Графическая библиотека GD. Обзор её возможностей и недостатков.
  - 6.2 Функции для создания, удаления и модификации изображений.
  - 6.3 Функции для получения информации о изображениях.
  - 6.4 Функции для рисования геометрических фигур.
  - 6.5 Функции для работы с текстом и шрифтами.
  - 6.6 Функции для работы с цветом.
  - 6.7 Функции для работы с пикселями.

## Ajax и PHP, Использование frameworks, CMS и PHP

1. Что такое Ajax?
2. Цели и задачи Ajax
3. Объект XMLHttpRequest



4. Пример использования XMLHttpRequest
5. Библиотеки для работы с Ajax
  - 5.1 Ajax
  - 5.2 AjaxAC
  - 5.3 Ajax
  - 5.4 Другие библиотеки
6. Практические примеры использования Ajax
7. PHP frameworks. Обзор существующих решений
  - 7.1 Zend Framework
  - 7.2 CodeIgniter
  - 7.3 CakePHP
  - 7.4 Symfony2
    - a. Обзор
    - b. Анализ архитектуры
    - c. Примеры использования
8. Web-мастеринг на основе CMS и обзор готовых проектов.
  - 8.1 Обзор рынка основных CMS, основанных на связке PHP и MySQL
  - 8.2 Обзор устройства и принципов работы CMS WordPress.
  - 8.3 Процесс написания плагинов для WordPress.
  - 8.4 Обзор CMS Joomla!
  - 8.5 Технология написания компонентов и плагинов для CMS Joomla!
  - 8.6 WYSIWYG редакторы для сайтов: FCKEditor, TinyMCE

#### 4. Программное обеспечение

Программное обеспечение	Пример
Операционная система	Windows 7, Windows 8, Windows 10
Текстовый редактор	Блокнот, Sublime Text, Notepad++
Браузеры	Internet Explorer, Google Chrome, Mozilla FireFox
Локальный веб-сервер	MAMP, OpenServer, Denwer

#### 5. Литература и информационное обеспечение

1. Роббинс Д.Н. HTML5, CSS3 и Javascript. Исчерпывающее руководство. - 4-е издание. — М.: Эксмо, 2014. — 516 с. — (Мировой компьютерный бестселлер). — ISBN: 978-5-699-67603-3.
2. Гаевский А.Ю. 100% самоучитель. Создание Web-страниц и Web-сайтов. HTML и JavaScript / А.Ю. Гаевский, В.А. Романовский. - М.: Триумф, 2014. - 464 с.
3. Дакетт Джон. HTML и CSS. Разработка и дизайн веб-сайтов (+ CD-ROM) / Джон Дакетт. - М.: Эксмо, 2013. - 480 с.
4. Квинт Игорь. Создаем сайты с помощью HTML, XHTML и CSS / Игорь Квинт. - М.: Питер, 2014. - 448 с.
5. Лазаро Исси Коэн. Полный справочник по HTML, CSS и JavaScript / Лазаро Исси Коэн, Джозеф Исси Коэн. - М.: ЭКОМ Паблшерз, 2014. - 938 с.

6. Фримен Элизабет. Изучаем HTML, XHTML и CSS / Элизабет Фримен, Эрик Фримен. - М.: Питер, 2016. - 720 с.
7. Конверс PHP 5 и MySQL. Библия пользователя / Конверс, др. Т. и. - М.: Вильямс, 2006. - 907 с.
8. Кузнецов PHP. Практика создания Web-сайтов / Кузнецов, М.В. и. - М.: БХВ-Петербург, 2008. - 619 с.
9. Стеймец PHP. 75 готовых решений для вашего сайта +CD / Стеймец, Ульям. - М.: СПб: Наука и Техника, 2009. - 256 с.