

МИНОБРНАУКИ РОССИИ  
ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ОБНОВЛЕНО  
для набора 2022  
Зам. директора по УР  
О.А. Улитина  
24 04 2023 г.

**РАБОЧАЯ ПРОГРАММА  
УЧЕБНОЙ ДИСЦИПЛИНЫ**

ОП.11 WEB-ТЕХНОЛОГИИ В ДИЗАЙНЕ  
программы подготовки квалифицированных рабочих и служащих  
54.01.20 Графический дизайнер

Форма обучения: очная

Уссурийск 2022



## Содержание

|   |    |
|---|----|
| <u>1. ОБЩАЯ ХАРАКТЕРИСТИКА ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ</u> .....                 | 4  |
| <u>1.1. Место дисциплины в структуре основной образовательной программы</u> ..... | 4  |
| <u>1.2. Цель и планируемые результаты освоения дисциплины</u> .....               | 4  |
| <u>2 СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ</u> .....                          | 6  |
| <u>3. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ ДИСЦИПЛИНЫ</u> .....                           | 10 |
| <u>4 КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ДИСЦИПЛИНЫ</u> .....                  | 13 |

# 1. ОБЩАЯ ХАРАКТЕРИСТИКА ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ

## 1.1. Место дисциплины в структуре основной образовательной программы

Учебная дисциплина **ОП.11 Web-технологии в дизайне** является частью общепрофессионального цикла основной образовательной программы (далее ООП) в соответствии с ФГОС СПО по профессии 54.01.20 Графический дизайнер.

## 1.2. Цель и планируемые результаты освоения дисциплины

По итогам освоения дисциплины, обучающиеся должны продемонстрировать результаты обучения, соотнесённые с результатами освоения ООП СПО, приведенные в таблице.

| Код ПК, ОК   | Умения  | Знания   |
|--|---|--|
| ОК 01<br>Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.                   | распознавать задачу и/или проблему в профессиональном и/или социальном контексте; анализировать задачу и/или проблему и выделять ее составные части; определять этапы решения задачи; выявлять и эффективно искать информацию, необходимую для решения задачи и/или проблемы; составлять план действия; определять необходимые ресурсы; применять актуальные методы работы в профессиональной и смежных сферах; реализовывать составленный план; оценивать результат и последствия своих действий (самостоятельно или с помощью наставника) | основные источники информации и ресурсы для решения задач и проблем в профессиональном и/или социальном контексте; алгоритмы выполнения работ в профессиональной и смежных областях; методы работы в профессиональной и смежных сферах; структуру плана для решения задач; порядок оценки результатов решения задач профессиональной деятельности. |
| ОК 02<br>Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности | определять задачи для поиска информации, необходимые источники информации; планировать процесс поиска; структурировать получаемую информацию; выделять наиболее значимое в перечне информации; оценивать практическую значимость результатов поиска; оформлять результаты поиска,   | перечень информационных источников, применяемых в профессиональной деятельности; приемы структурирования информации; формат оформления результатов поиска информации,  |
| ОК 03<br>Планировать и реализовывать собственное профессиональное и личностное развитие  | определять актуальность нормативно-правовой документации в профессиональной деятельности; применять современную научную   | содержание актуальной нормативно-правовой документации; современную научную и профессиональную терминологию; возможные   |

|   |   |  |
|---|---|--|
|   | <p>профессиональную терминологию; определять и выстраивать траектории профессионального развития и самообразования;</p>   | <p>траектории профессионального развития и самообразования;</p>  |
| <p>ОК 09 Использовать информационные технологии профессиональной деятельности В</p>   | <p>применять средства информационных технологий для решения профессиональных задач; использовать современное программное обеспечение; использовать различные цифровые средства для решения профессиональных задач.</p>  | <p>современные средства и устройства информатизации; порядок их применения и программное обеспечение в профессиональной деятельности в том числе с использованием цифровых средств.</p>  |
| <p>ОК 11 Использовать знания по финансовой грамотности, планировать предпринимательскую деятельность В профессиональной сфере</p> | <p>выявлять достоинства и недостатки коммерческой идеи; презентовать идеи открытия собственного дела в профессиональной деятельности; оформлять бизнес-план; рассчитывать размеры выплат по процентным ставкам кредитования; определять инвестиционную привлекательность коммерческих идей в рамках профессиональной деятельности; презентовать бизнес-идею; определять источники финансирования.</p> | <p>основы предпринимательской деятельности; основы финансовой грамотности; правила разработки бизнес-планов; порядок выстраивания презентации; кредитные банковские продукты.</p>  |
| <p>ПК 1.1 Осуществлять сбор, систематизацию и анализ данных необходимых для разработки технического задания дизайн-продукта.</p>  | <p>проводить проектный анализ; производить расчеты основных технико-экономических показателей проектирования; собирать, обобщать и структурировать информацию</p>   | <p>: методики исследования рынка, сбора информации, ее анализа и структурирования; теоретических основ композиционного построения в графическом и в объемно-пространственном дизайне; законов формообразования; систематизирующих методов формообразования (модульность и комбинаторика); преобразующих методов формообразования (стилизиция и трансформация); законов создания цветовой гармонии; программных приложений работы с данными</p> |
| <p>ПК 1.4. Выполнять</p>  | <p>презентовать разработанное ТЗ</p>  | <p>основ менеджмента и</p>   |

|   |   |   |
|---|---|---|
| процедуру согласования (утверждения) заказчиком   | согласно требованиям к структуре и содержанию   | коммуникации, договорных отношений; стандартов производства; программных приложений работы с данными для презентации  |
| ПК 2.1. Планировать выполнение работ по разработке дизайн-макета на основе технического задания   | разрабатывать планы выполнения работ; распределять время на выполнение поставленных задач; определять место хранения и обработки разрабатываемых макетов; разрабатывать технологическую карту изготовления авторского проекта | структуры ТЗ, его реализации; основ менеджмента времени и выполнения работ; программных приложений работы с данными   |
| ПК 2.2. Определять потребности программных продуктах, материалах и оборудовании при разработке дизайн-макета на основе технического задания | выбирать материалы и программное обеспечение с учетом их наглядных и формообразующих свойств; понимать сочетание в дизайн-проекте собственного художественного вкуса и требований заказчика                                   | технологических, эксплуатационных и гигиенических требований, предъявляемых к материалам, программным средствам и оборудованию; программных приложений работы с данными для разработки дизайн-макетов |
| ПК 4.2. Проводить мастер-классы, семинары и консультации по современным технологиям в области графического дизайна                          | организовывать и проводить мероприятия профориентационного и мотивационного характера   | системы управления трудовыми ресурсами в организации; основ менеджмента и коммуникации, договорных отношений  |

## 2 СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

### 2.1 Объем учебной дисциплины и виды учебной работы

| Вид учебной работы  | Объем часов |
|---|-------------|
| <b>Объем образовательной программы учебной дисциплины</b> | <b>36</b>   |
| <b>Обязательная аудиторная учебная нагрузка (всего)</b>   | 36          |
| В том числе:  | -           |
| – теоретическое обучение                                  | -           |
| – практические занятия                                    | 36          |
| – лабораторные занятия                                    | -           |

|   |   |
|---|---|
| – курсовая работа (проект)                            | - |
| – самостоятельная работа                              | - |
| – консультации  | - |
| – промежуточная аттестация - Дифференцированный зачет | - |

## 2.2 Тематический план и содержание учебной дисциплины

| Наименование разделов и тем   | Содержание учебного материала, лабораторные и практические занятия, самостоятельная работа обучающихся. | Объем часов      | Коды компетенций, формированию которых способствует элемент программы |
|---|---|------------------|---|
| <b>Раздел 1. Веб-дизайн и программирование</b>                                  |   | 28               |   |
| <b>Тема 1.1. Введение в Web-конструирование.</b>                                | Содержание учебного материала   | не предусмотрено | ОК 01,02. 03, 09<br>ПК 1.1, 2.1,2.2,<br>4.2                           |
|   | 1   |                  |   |
|   | Лабораторные занятия  | не предусмотрено |   |
|   | Практические занятия  |                  |   |
|   | 1   № 1 Создание web-страниц в HTML.  | 2                |   |
|   | 2   № 2 Создание web-страниц в HTML.  | 2                |   |
|   | Контрольные работы  | не предусмотрено |   |
|   | Самостоятельная работа обучающихся  | не предусмотрено |   |
| <b>Тема 1.2. Программирование на PHP. Стилиевое оформление HTML-документов.</b> | Содержание учебного материала   | не предусмотрено |   |
|   | 1   |                  |   |
|   | Лабораторные занятия  | не предусмотрено |   |
|   | Практические занятия  |                  |   |
|   | 1   № 3 Оформление текста в HTML  | 2                |   |
|   | 2   № 4 Списки, гиперссылки и якоря в HTML  | 2                |   |
|   | 3   № 5 Таблицы и фреймы в HTML   | 2                |   |
|   | 4   № 6 Формы в HTML  | 2                |   |
|   | 5   № 7 Каскадные таблицы стилей CSS.   | 2                |   |
|   | 6   № 8 Публикация сайта в интернете.   | 2                |   |
|   | Контрольные занятия   | не предусмотрено |   |
|   | Самостоятельная работа обучающихся  | не предусмотрено |   |



|   |   |  |                  |   |
|---|---|--|------------------|---|
| <b>Тема 1.3 Сценарии JavaScript и DHTML.</b>                              | Содержание учебного материала           |  |                  | ОК 01,02. 03, 09<br>ПК 1.1, 2.1,2.2,<br>4.2 |
|   | 1                                       |  | не предусмотрено |   |
|   | Лабораторные занятия                    |  | не предусмотрено |   |
|   | Практические занятия                    |  |                  |   |
|   | 1                                       | № 9 Разработка веб-сайтов с использованием сценариев JavaScript и DHTML              | 2                |   |
|   | 2                                       | № 10 Совместное использование HTML, CSS и JavaScript                                 | 2                |   |
|   | Контрольные работы                      |  | не предусмотрено |   |
| Самостоятельная работа обучающихся  |   | не предусмотрено   |                  |   |
| <b>Тема 1.4 Webсерверы. Безопасность и шифрование.</b>                    | Содержание учебного материала           |  | не предусмотрено | ОК 01,02. 03, 09<br>ПК 1.1, 2.1,2.2,<br>4.2 |
|   | 1                                       |  |                  |   |
|   | Лабораторные занятия                    |  | не предусмотрено |   |
|   | Практические занятия                    |  |                  |   |
|   | 1                                       | № 11 Создание Web-страницы «Моя маленькая родина», используя контейнерный Web-Дизайн | 2                |   |
|   | 2                                       | № 12 Динамический HTML   | 2                |   |
|   | 3                                       | № 13 Разработка веб-сайтов с использованием PHP.                                     | 2                |   |
|   | 4                                       | № 14 Разработка веб-сайтов с использованием PHP.                                     | 2                |   |
|   | 5                                       | № 15 Создание интернет приложений на стороне сервера.                                | 2                |   |
|   | 6                                       | № 16 Создание интернет приложений на стороне сервера.                                | 2                |   |
|   | 7                                       | № 17 Технология SSI. Регистрационная форма.  | 2                |   |
| 8   | № 18 Интернет-приложения и базы данных. | 2  |                  |   |
| Контрольные работы  |   | не предусмотрено   |                  |   |
| Самостоятельная работа обучающихся  |   | не предусмотрено   |                  |   |
| <b>Примерная тематика курсовой работы</b>                                 |   |  | не предусмотрено |   |
| <b>Самостоятельная работа обучающихся над курсовой работой (проектом)</b> |   |  | не предусмотрено |   |
| <b>Промежуточная аттестация Дифференцированный зачет</b>                  |   |  | -                |   |
| <b>Всего:</b>   |   |  | 36               |   |

### 3. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ ДИСЦИПЛИНЫ

#### 3.1 Требования к минимальному материально-техническому обеспечению

Учебная аудитория для проведения учебных занятий (урок, практическое занятие, лабораторное занятие, лекция, семинар), групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации:

Кабинет - компьютерных (информационных) технологий.

количество посадочных мест - 20 шт., стол для преподавателя 1 шт., стул для преподавателя 1 шт., мультимедийное оборудование 1 шт., доска маркерная

ПО: Microsoft Windows 7 Professional Russian, ООО "Битроникс Владивосток" Контракт № 0320100030814000018-45081 от 09.09.14, лицензия №64099496, бессрочно

#### 3.2. Информационное обеспечение реализации программы

Для реализации программы учебной дисциплины библиотечный фонд ВГУЭС укомплектован печатными и электронными изданиями.

Обучающиеся из числа инвалидов и лиц с ограниченными возможностями здоровья обеспечены печатными и (или) электронными образовательными ресурсами в формах, адаптированных к ограничениям их здоровья.

#### Основная литература

1 Черпаков, И. В. Основы программирования : учебник и практикум для среднего профессионального образования / И. В. Черпаков. — Москва : Издательство Юрайт, 2023. — 219 с. — (Профессиональное образование). — ISBN 978-5-9916-9984-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/513113> (дата обращения: 19.05.2023).

2 Полуэктова, Н. Р. Разработка веб-приложений : учебное пособие для вузов / Н. Р. Полуэктова. — Москва : Издательство Юрайт, 2023. — 204 с. — (Высшее образование). — ISBN 978-5-534-13715-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/519714> (дата обращения: 19.05.2023).

3 Трофимов, В. В. Основы алгоритмизации и программирования : учебник для среднего профессионального образования / В. В. Трофимов, Т. А. Павловская ; под редакцией В. В. Трофимова. — Москва : Издательство Юрайт, 2023. — 137 с. — (Профессиональное образование). — ISBN 978-5-534-07321-8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/515434> (дата обращения: 19.05.2023).

4 Боресков, А. В. Компьютерная графика : учебник и практикум для среднего профессионального образования / А. В. Боресков, Е. В. Шикин. — Москва : Издательство Юрайт, 2023. — 219 с. — (Профессиональное образование). — ISBN 978-5-534-11630-4.

— Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/518504> (дата обращения: 19.05.2023).

### **Дополнительная литература**

1. Практика разработки Web-сайтов. Гурова, Т. Ф. Экология и рациональное природопользование : учебник и практикум для среднего профессионального образования / Т. Ф. Гурова, Л. В. Назаренко. — 3-е изд., испр. и доп. — Москва : Издательство Юрайт, 2020. — 188 с. — (Профессиональное образование). — ISBN 978-5-534-09485-5. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/452780>

2. Чернышев, С. А. Основы программирования на Python : учебное пособие для вузов / С. А. Чернышев. — Москва : Издательство Юрайт, 2023. — 286 с. — (Высшее образование). — ISBN 978-5-534-14350-8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/519949> (дата обращения: 19.05.2023).

### **Нормативные документы**

1. Федеральный закон от 27 июля 2006 г. N 149-ФЗ "Об информации, информационных технологиях и о защите информации" (с изменениями и дополнениями).- Электронный ресурс. Режим доступа: <https://base.garant.ru/12148555/>

### **Электронные ресурсы**

1 Руководство по изучению php с нуля с примерами“.- Электронный ресурс. Режим доступа: <https://wiki.merionet.ru/servevnye-resheniya/88/rukovodstvo-po-izucheniyu-php-s-nulya-s-primerami/?amp&amp>

2 Современный учебник JavaScript“.- Электронный ресурс. Режим доступа: <https://learn.javascript.ru/>

3 Единная система актуальных требований <https://esat.worldskills.ru>

4 Портал дизайна Электронный ресурс. Режим доступа: <http://ru.vectorboom.com>

5 Крупнейший обучающий портал для дизайнеров и творческих людей Электронный ресурс. Режим доступа: <http://skillsup.ru> - «Skillsup

6 Блог о Веб-дизайне Электронный ресурс. Режим доступа: <http://design-mania.ru>

7 Веб-дизайн Электронный ресурс. Режим доступа: <http://egraphic.ru>

8 «Видеоуроки» Электронный ресурс. Режим доступа: <http://compteacher.ru/>

9 «Викиучебник» Электронный ресурс. Режим доступа: <http://ru.wikibooks.org/>

10 «Все о графике и дизайне» Электронный ресурс. Режим доступа: <http://balbesof.net/>

11 «Журнал по дизайну» Электронный ресурс. Режим доступа: <http://www.dejurka.ru>

12 Web-технологии: HTML, DHTML, JavaScript, PHP, MySQL, XML+XLST, Ajax Электронный ресурс. Режим доступа: [www.htmlweb.ru](http://www.htmlweb.ru)

13 HTML, DHTML, пример создания сайта, пример раскрутки сайта, обзор CMS, пример работы с CSS Электронный ресурс. Режим доступа: [www.on-line-teaching.com](http://www.on-line-teaching.com) азы

#### 4 КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ДИСЦИПЛИНЫ

| Результаты обучения   | Критерии оценки  | Методы оценки  |
|---|--|--|
| <p>У1 распознавать задачу и/или проблему в профессиональном и/или социальном контексте; анализировать задачу и/или проблему и выделять ее составные части; определять этапы решения задачи; выявлять и эффективно искать информацию, необходимую для решения задачи и/или проблемы; составлять план действия; определять необходимые ресурсы; применять актуальные методы работы в профессиональной и смежных сферах; реализовывать составленный план; оценивать результат и последствия своих действий (самостоятельно или с помощью наставника)</p> | <p>умеет распознавать задачу и/или проблему в профессиональном и/или социальном контексте; анализировать задачу и/или проблему и выделять ее составные части; определять этапы решения задачи; выявлять и эффективно искать информацию, необходимую для решения задачи и/или проблемы; составлять план действия; определять необходимые ресурсы; применять актуальные методы работы в профессиональной и смежных сферах; реализовывать составленный план; оценивать результат и последствия своих действий (самостоятельно или с помощью наставника)</p> | <p>Текущий контроль<br/>Дифференцированный зачет</p> |
| <p>У2 определять задачи для поиска информации, необходимые источники информации; планировать процесс поиска; структурировать получаемую информацию; выделять наиболее значимое в перечне информации; оценивать практическую значимость результатов поиска; оформлять результаты поиска,</p>   | <p>умеет определять задачи для поиска информации, необходимые источники информации; планировать процесс поиска; структурировать получаемую информацию; выделять наиболее значимое в перечне информации; оценивать практическую значимость результатов поиска; оформлять результаты поиска,</p>   | <p>Текущий контроль<br/>Дифференцированный зачет</p> |
| <p>У3 определять актуальность нормативно-правовой документации в профессиональной деятельности; применять современную научную профессиональную терминологию; определять и выстраивать траектории</p>  | <p>умеет определять актуальность нормативно-правовой документации в профессиональной деятельности; применять современную научную профессиональную терминологию; определять и</p>   | <p>Текущий контроль<br/>Дифференцированный зачет</p> |

|  |   |  |
|--|---|--|
| <p>профессионального развития и самообразования;</p>   | <p>выстраивать траектории профессионального развития и самообразования;</p>   |  |
| <p>У4 применять средства информационных технологий для решения профессиональных задач; использовать современное программное обеспечение; использовать различные цифровые средства для решения профессиональных задач.</p>  | <p>умеет применять средства информационных технологий для решения профессиональных задач; использовать современное программное обеспечение; использовать различные цифровые средства для решения профессиональных задач.</p>  | <p>Текущий контроль<br/>Дифференцированный зачет</p> |
| <p>У5 выявлять достоинства и недостатки коммерческой идеи; презентовать идеи открытия собственного дела в профессиональной деятельности; оформлять бизнес-план; рассчитывать размеры выплат по процентным ставкам кредитования; определять инвестиционную привлекательность коммерческих идей в рамках профессиональной деятельности; презентовать бизнес-идею; определять источники финансирования.</p> | <p>умеет выявлять достоинства и недостатки коммерческой идеи; презентовать идеи открытия собственного дела в профессиональной деятельности; оформлять бизнес-план; рассчитывать размеры выплат по процентным ставкам кредитования; определять инвестиционную привлекательность коммерческих идей в рамках профессиональной деятельности; презентовать бизнес-идею; определять источники финансирования.</p> | <p>Текущий контроль<br/>Дифференцированный зачет</p> |
| <p>31 основные источники информации и ресурсы для решения задач и проблем в профессиональном и/или социальном контексте; алгоритмы выполнения работ в профессиональной и смежных областях; методы работы в профессиональной и смежных сферах; структуру плана для решения задач; порядок оценки результатов решения задач профессиональной</p>   | <p>демонстрирует знание основных источников информации и ресурсы для решения задач и проблем в профессиональном и/или социальном контексте; алгоритмы выполнения работ в профессиональной и смежных областях; методы работы в профессиональной и смежных сферах; структуру плана для решения задач; порядок оценки результатов решения</p>  | <p>Текущий контроль<br/>Дифференцированный зачет</p> |

|   |  |  |
|---|--|--|
| деятельности.   | задач профессиональной деятельности.   |  |
| 32 перечень информационных источников, применяемых в профессиональной деятельности; приемы структурирования информации; формат оформления результатов поиска информации,            | демонстрирует знание перечня информационных источников, применяемых в профессиональной деятельности; приемы структурирования информации; формат оформления результатов поиска информации,            | Текущий контроль<br>Дифференцированный зачет |
| 33 содержание актуальной нормативно-правовой документации; современную научную и профессиональную терминологию; возможные траектории профессионального развития и самообразования;  | демонстрирует знание содержания актуальной нормативно-правовой документации; современной научной и профессиональной терминологии; возможные траектории профессионального развития и самообразования; | Текущий контроль<br>Дифференцированный зачет |
| 34 современные средства и устройства информатизации; порядок их применения и программное обеспечение в профессиональной деятельности в том числе с использованием цифровых средств. | демонстрирует знание современных средств и устройств информатизации; порядок их применения и программное обеспечение в профессиональной деятельности в том числе с использованием цифровых средств.  | Текущий контроль<br>Дифференцированный зачет |
| 35 основы предпринимательской деятельности; основы финансовой грамотности; правила разработки бизнес-планов; порядок выстраивания презентации; кредитные банковские продукты.       | демонстрирует знание основ предпринимательской деятельности; основ финансовой грамотности; правила разработки бизнес-планов; порядок выстраивания презентации; кредитные банковские продукты.        | Текущий контроль<br>Дифференцированный зачет |

МИНОБРНАУКИ РОССИИ  
ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

**КОНТРОЛЬНО-ОЦЕНОЧНЫЕ СРЕДСТВА**  
для проведения текущего контроля и промежуточной аттестации  
по учебной дисциплине  
ОП.11 WEB-ТЕХНОЛОГИИ В ДИЗАЙНЕ  
программы подготовки квалифицированных рабочих и служащих  
54.01.20 Графический дизайнер  
Форма обучения: *очная*

Уссурийск 2022





## 1 Общие сведения

Контрольно-оценочные средства (далее – КОС) предназначены для контроля и оценки образовательных достижений обучающихся, освоивших программу учебной дисциплины ОП.11 Web-технологии в дизайне .

КОС включают в себя контрольные материалы для проведения текущего контроля успеваемости и промежуточной аттестации по дисциплине, которая проводится в форме дифференцированного зачёта (с использованием оценочного средства - выполнение заданий, тестирование)

## 2 Планируемые результаты обучения по дисциплине, обеспечивающие результаты освоения образовательной программы

| Код ОК, ПК | Код результата обучения | Наименование результата обучения  |
|------------|-------------------------|---|
| ОК01       | 31                      | основные источники информации и ресурсы для решения задач и проблем в профессиональном и/или социальном контексте; алгоритмы выполнения работ в профессиональной и смежных областях; методы работы в профессиональной и смежных сферах; структуру плана для решения задач; порядок оценки результатов решения задач профессиональной деятельности.  |
|            | У1                      | распознавать задачу и/или проблему в профессиональном и/или социальном контексте; анализировать задачу и/или проблему и выделять ее составные части; определять этапы решения задачи; выявлять и эффективно искать информацию, необходимую для решения задачи и/или проблемы; составлять план действия; определять необходимые ресурсы; применять актуальные методы работы в профессиональной и смежных сферах; реализовывать составленный план; оценивать результат и последствия своих действий (самостоятельно или с помощью наставника) |
| ОК02       | 32                      | перечень информационных источников, применяемых в профессиональной деятельности; приемы структурирования информации; формат оформления результатов поиска информации,   |
|            | У2                      | определять задачи для поиска информации, необходимые источники информации; планировать процесс поиска; структурировать получаемую информацию; выделять наиболее значимое в перечне информации; оценивать практическую значимость результатов поиска; оформлять результаты поиска,   |
| ОК03       | 33                      | содержание актуальной нормативно-правовой документации; современную научную и профессиональную терминологию; возможные траектории профессионального развития и самообразования;   |
|            | У3                      | определять актуальность нормативно-правовой документации в профессиональной деятельности; применять современную научную профессиональную терминологию; определять и выстраивать траектории профессионального развития и самообразования;  |
|            | 34                      | современные средства и устройства информатизации; порядок их применения и программное обеспечение в профессиональной  |

| Код ОК, ПК | Код результата обучения | Наименование результата обучения  |
|------------|-------------------------|---|
| ОК09       |                         | деятельности в том числе с использованием цифровых средств  |
|            | У4                      | применять средства информационных технологий для решения профессиональных задач; использовать современное программное обеспечение; использовать различные цифровые средства для решения профессиональных задач.   |
| ОК 11      | 35                      | основы предпринимательской деятельности; основы финансовой грамотности; правила разработки бизнес-планов; порядок выстраивания презентации; кредитные банковские продукты.  |
|            | У6                      | выявлять достоинства и недостатки коммерческой идеи; презентовать идеи открытия собственного дела в профессиональной деятельности; оформлять бизнес-план; рассчитывать размеры выплат по процентным ставкам кредитования; определять инвестиционную привлекательность коммерческих идей в рамках профессиональной деятельности; презентовать бизнес-идею; определять источники финансирования.  |
| ПК 1.1     | 311                     | методики исследования рынка, сбора информации, ее анализа и структурирования;<br>теоретических основ композиционного построения в графическом и в объемно-пространственном дизайне;<br>законов формообразования;<br>систематизирующих методов формообразования (модульность и комбинаторика);<br>преобразующих методов формообразования (стилизация и трансформация);<br>законов создания цветовой гармонии;<br>программных приложений работы с данными |
|            | У11                     | проводить проектный анализ;<br>производить расчеты основных технико-экономических показателей проектирования;<br>собирать, обобщать и структурировать информацию  |
| ПК 1.4.    | 312                     | основ менеджмента и коммуникации, договорных отношений; стандартов производства;<br>программных приложений работы с данными для презентации   |
|            | У12                     | презентовать разработанное ТЗ согласно требованиям к структуре и содержанию   |
| ПК 2.1.    | 313                     | структуры ТЗ, его реализации;<br>основ менеджмента времени и выполнения работ;<br>программных приложений работы с данными   |
|            | У13                     | разрабатывать планы выполнения работ;<br>распределять время на выполнение поставленных задач;<br>определять место хранения и обработки разрабатываемых макетов;<br>разрабатывать технологическую карту изготовления авторского проекта  |
| ПК 2.2     | 314                     | технологических, эксплуатационных и гигиенических   |

| Код ОК, ПК | Код результата обучения | Наименование результата обучения  |
|------------|-------------------------|---|
|            |                         | требований, предъявляемых к материалам, программным средствам и оборудованию; программных приложений работы с данными для разработки дизайн-макетов   |
|            | У14                     | выбирать материалы и программное обеспечение с учетом их наглядных и формообразующих свойств; понимать сочетание в дизайн-проекте собственного художественного вкуса и требований заказчика |
| ПК 4.2     | 315                     | системы управления трудовыми ресурсами в организации; основ менеджмента и коммуникации, договорных отношений  |
|            | 315                     | организовывать и проводить мероприятия профориентационного и мотивационного характера   |

### 3 Соответствие оценочных средств контролируемым результатам обучения

#### 3.1 Средства, применяемые для оценки уровня теоретической подготовки

| Краткое наименование раздела (модуля) / темы дисциплины                         | Наименование оценочного средства и представление его в КОС <sup>3</sup> |                                       |
|---|---|---------------------------------------|
|   | Текущий контроль <sup>4</sup>   | Промежуточная аттестация <sup>4</sup> |
| <b>Раздел 1. Веб-дизайн и программирование</b>                                  |   |                                       |
| <b>Тема 1.1. Введение в Web-конструирование.</b>                                | Устный опрос (п. 5.1)   | Вопросы (п. 5.1)                      |
| <b>Тема 1.2. Программирование на PHP. Стилиевое оформление HTML-документов.</b> | Устный опрос (п. 5.1)   |                                       |
| <b>Тема 1.3 Сценарии JavaScript и DHTML.</b>                                    | Устный опрос (п. 5.1)   |                                       |
| <b>Тема 1.4 Webсерверы. Безопасность и шифрование.</b>                          | Устный опрос (п. 5.1)   |                                       |

#### 3.2 Средства, применяемые для оценки уровня практической подготовки

| Краткое наименование раздела (модуля) / темы дисциплины | Наименование оценочного средства и представление его в КОС |
|---|--|
|---|--|

|  | Текущий контроль                            | Промежуточная аттестация             |
|--|---|--------------------------------------|
| <b>Тема 1.1. Введение в Web-конструирование.</b>                                 | <i>Практическое задание № 1, 2 (п.5.2)</i>  | <i>Практическое задание к зачету</i> |
| <b>Тема 1.2. Программирование на PHP. Стилизовое оформление HTML-документов.</b> | <i>Практическое задание № 3-8 (п.5.2)</i>   |                                      |
| <b>Тема 1.3 Сценарии JavaScript и DHTML.</b>                                     | <i>Практическое задание № 9-10 (п.5.2)</i>  |                                      |
| <b>Тема 1.4 Webсерверы. Безопасность и шифрование.</b>                           | <i>Практическое задание № 11-18 (п.5.2)</i> |                                      |

#### **4 Описание процедуры оценивания**

Результаты обучения по дисциплине, уровень сформированности компетенций оцениваются по четырём бальной шкале оценками: «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

Текущая аттестация по дисциплине проводится с целью систематической проверки достижений обучающихся. Объектами оценивания являются: степень усвоения теоретических знаний, уровень овладения практическими умениями и навыками по всем видам учебной работы, качество выполнения самостоятельной работы, учебная дисциплина (активность на занятиях, своевременность выполнения различных видов заданий, посещаемость всех видов занятий по аттестуемой дисциплине).

При проведении промежуточной аттестации оценивается достижение студентом запланированных по дисциплине результатов обучения, обеспечивающих результаты освоения образовательной программы в целом.

#### **Критерии оценивания устного ответа**

(оценочные средства: *устный опрос, собеседование, дискуссия,*)

**5 баллов** - ответ показывает прочные знания основных процессов изучаемой предметной области, отличается глубиной и полнотой раскрытия темы; владение терминологическим аппаратом; умение объяснять сущность, явлений, процессов, событий, делать выводы и обобщения, давать аргументированные ответы, приводить примеры; свободное владение монологической речью, логичность и последовательность ответа; умение приводить примеры современных проблем изучаемой области.

**4 балла** - ответ, обнаруживающий прочные знания основных процессов изучаемой предметной области, отличается глубиной и полнотой раскрытия темы; владение

терминологическим аппаратом; умение объяснять сущность, явлений, процессов, событий, делать выводы и обобщения, давать аргументированные ответы, приводить примеры; свободное владение монологической речью, логичность и последовательность ответа. Однако допускается одна - две неточности в ответе.

**3 балла** – ответ, свидетельствующий в основном о знании процессов изучаемой предметной области, отличающийся недостаточной глубиной и полнотой раскрытия темы; знанием основных вопросов теории; слабо сформированными навыками анализа явлений, процессов, недостаточным умением давать аргументированные ответы и приводить примеры; недостаточно свободным владением монологической речью, логичностью и последовательностью ответа. Допускается несколько ошибок в содержании ответа; неумение привести пример развития ситуации, провести связь с другими аспектами изучаемой области.

**2 балла** – ответ, обнаруживающий незнание процессов изучаемой предметной области, отличающийся неглубоким раскрытием темы; незнанием основных вопросов теории, несформированными навыками анализа явлений, процессов; неумением давать аргументированные ответы, слабым владением монологической речью, отсутствием логичности и последовательности. Допускаются серьезные ошибки в содержании ответа; незнание современной проблематики изучаемой области.

### **Критерии оценивания письменной работы**

(оценочные средства: *письменный отчет по практической работе*).

**5 баллов** - студент выразил своё мнение по сформулированной проблеме, аргументировал его, точно определив ее содержание и составляющие. Проблема раскрыта полностью, выводы обоснованы. Приведены данные отечественной и зарубежной литературы, статистические сведения, информация нормативно-правового характера. Студент владеет навыком самостоятельной работы по заданной теме; методами и приемами анализа теоретических и/или практических аспектов изучаемой области. Фактических ошибок, связанных с пониманием проблемы, нет; графически работа оформлена правильно.

**4 балла** - работа характеризуется смысловой цельностью, связностью и последовательностью изложения; допущено не более 1 ошибки при объяснении смысла или содержания проблемы. Проблема раскрыта. Не все выводы сделаны и/или обоснованы. Для аргументации приводятся данные отечественных и зарубежных авторов. Продемонстрированы исследовательские умения и навыки. Фактических ошибок, связанных с пониманием проблемы, нет. Допущены одна-две ошибки в оформлении работы.

**3 балла** – студент проводит достаточно самостоятельный анализ основных этапов и смысловых составляющих проблемы; понимает базовые основы и теоретическое обоснование выбранной темы. Проблема раскрыта не полностью. Выводы не сделаны и/или выводы не обоснованы. Проведен анализ проблемы без привлечения дополнительной литературы. Допущено не более 2 ошибок в смысле или содержании проблемы, оформлении работы.

**2 балла** - работа представляет собой пересказанный или полностью переписанный исходный текст без каких бы то ни было комментариев, анализа. Не раскрыта структура и теоретическая составляющая темы. Проблема не раскрыта. Выводы отсутствуют. Допущено три или более трех ошибок в смысловом содержании раскрываемой проблемы, в оформлении работы.

### **Критерии оценивания тестового задания на зачете**

| Оценка                        | <i>Отлично</i> | <i>Хорошо</i>    | <i>Удовлетворительно</i> | <i>Неудовлетворительно</i> |
|-------------------------------|----------------|------------------|--------------------------|----------------------------|
| Количество правильных ответов | 91 % и ≥       | от 81% до 90,9 % | не менее 70%             | менее 70%                  |

**Критерии выставления оценки студенту на зачете**  
(оценочные средства: *выполнение заданий*)

| Оценка по промежуточной аттестации   | Характеристика качества сформированности компетенций   |
|--------------------------------------|--|
| «зачтено» / «отлично»                | Студент демонстрирует сформированность дисциплинарных компетенций на продвинутом уровне: обнаруживает всестороннее, систематическое и глубокое знание учебного материала, умеет тесно увязывать теорию с практикой, свободно справляется с задачами, вопросами и другими видами применения знаний, причем не затрудняется с ответом при видоизменении заданий, правильно обосновывает принятое решение, владеет разносторонними навыками и приемами выполнения практических задач. |
| «зачтено» / «хорошо»                 | Студент демонстрирует сформированность дисциплинарных компетенций на базовом уровне: основные знания, умения освоены, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.  |
| «зачтено» / «удовлетворительно»      | Студент демонстрирует сформированность дисциплинарных компетенций на пороговом уровне: имеет знания только основного материала, но не усвоил его деталей, в ходе контрольных мероприятий допускаются значительные ошибки, недостаточно правильные формулировки, нарушения логической последовательности в изложении программного материала, испытывает затруднения при выполнении практических работ, при оперировании знаниями и умениями при их переносе на новые ситуации.      |
| «не зачтено» / «неудовлетворительно» | Студент демонстрирует сформированность дисциплинарных компетенций на уровне ниже порогового: выявляется полное или практически полное отсутствие знаний значительной части программного материала, студент допускает существенные ошибки, неуверенно, с большими затруднениями выполняет практические работы, умения и навыки не сформированы.   |

## **5. Примеры оценочных средств для проведения текущей аттестации**

### **5.1 Оценочные материалы для текущего промежуточного контроля в форме устного опроса:**

1. Определение, классификация и характеристика WEB-сайтов по различным признакам
2. Распространенные форматы данных в Интернете. Архитектура клиентсервер. Основы работы web-сервера.
3. Составные элементы HTML-документа. Типы данных HTML. Структура HTML-документа.
4. Общие атрибуты элементов HTML. Теги заголовка документа. Теги тела документа. Блочные и строчные элементы разметки. Работа с текстом. Заголовки и абзацы. Списки: нумерованные, маркированные.
5. Объекты HTML-документов. Типы файлов иллюстраций. Управление размещением иллюстрации и обтеканием текста. Вставка объектов. Карты ссылок. Создание гиперссылок.
6. Создание таблиц. Основные атрибуты таблиц, строк, ячеек. Особенности использования таблиц для верстки web-документов.
7. Понятие объекта в HTML-документах. Карта ссылок. Вставка апплетов, элементов ActiveX, объектов, обрабатываемых с помощью расширений обозревателя, и др. Элементы форм.
8. Типы управляющих элементов. Правила работы с формами. Понятие фреймовой структуры web-страницы.
9. Стилиевое оформление HTML-документов. Каскадные таблицы стилей (CSS).
10. Основные понятия компьютерной графики
11. Сценарии JavaScript и DHTML
12. Web-серверы. Пассивные и активные web-серверы. Основы клиентсерверного взаимодействия.
13. Обзор технологий серверного интернет-программирования (CGI/Perl, PHP
14. ASP, SSI и др.), их поддержка различными операционными системами и web-серверами.
15. Основы работы с базами данных в интернет-приложениях. Обзор типичных интернет-технологий баз данных.
16. XML: стандарты, области применения, связанные технологии и возможности.
17. DTD — определение типа документа. Основные структурные элементы DTD. Внешние и внутренние DTD



## 5.2 Примеры практических заданий

### Практическое занятие № 1 - 2

**Цель:** научиться создавать простейшие HTML-документы, разбивать текст на абзацы, выполнять выравнивание различными способами, просматривать Web-страницы с помощью браузера.

**Теги** — это определённые последовательности символов, заключённые между знаками < (меньше) и > (больше). Символ < обозначает начало тега, символ > обозначает конец тега. **HTML-документ** – это всё, что заключено между тегами <HTML> и </HTML>.

#### Структура HTML-документа

```
<HTML>
  <head >
    Заголовок
  <TITLE>
    Название докумен-
  </TITLE>
  </head>
  <BODY>
    Тело документа
  </BODY>
</HTML>
```

**Заголовок** находится между тегами <HEAD> и </HEAD>. В нем содержится информация о документе, которая не выводится на экран (является необязательным элементом).

Название странички располагается между тегами <TITLE> и </TITLE> и появляется в верхней рамке окна программы просмотра.

Прописные и строчные буквы в написании тегов значения не имеют.

**Тело** выводится на экран программой просмотра (**браузером**) – текст, картинки, видеофрагменты.

Оно заключается между тегами <BODY> и </BODY>.

#### Задание 1. Создание простейшего HTML-документа.

1. Откройте текстовый редактор **Блокнот**.
2. Создайте в нем следующий документ:

```
<HTML>
<HEAD>
<TITLE> Моя страничка </TITLE>
</HEAD> <BODY>
Привет! Это моя первая страничка!
</BODY>
</HTML>
```

3. Сохраните этот файл в своей папке, выполнив команду **Файл / Сохранить как**, в поле **Имя файла** введите **Пример.html**. НЕ ЗАКРЫВАЯ **Блокнот**, откройте этот файл в окне программы просмотра **Internet Explorer** с помощью команды **Файл / Открыть**.

#### Разбиение на абзацы, различные способы выравнивания

|                     |                                |                 |
|---------------------|--------------------------------|-----------------|
| Разбиение на абзацы | <P> текст </P>                 |                 |
| Абзац с выравниваем | <P ALIGN="Left"> текст </P>    | По левому краю  |
|                     | <P ALIGN="Center"> текст </P>  | По центру       |
|                     | <P ALIGN="Right"> текст </P>   | По правому краю |
|                     | <P ALIGN="Justify"> текст </P> | По ширине       |


|                                  |   |  |
|----------------------------------|---|--|
| Перевод на новую строку в абзаце | <BR>  |  |
| Заголовок (уровни от 1 до 6)     | <H1> текст1 </H1><br>...<br><H6> текст6 </H6> |  |
| Заголовок с выравниваем          | <H1 ALIGN="Left"> текст </H1>                 |  |

**Замечание:** пробелы и клавишу *Enter* внутри абзаца можно использовать произвольно (в браузере это не отображается).

## Задание 2. Усовершенствование странички.

1. Добавьте в свою страничку следующий текст:

```
<html> <title> первые стихи </title> <body>
<p align=center> Привет! Это моя первая страничка!
<p align=left> В лесу родилась елочка.
В лесу она росла.
<p align=right> Зимой и летом стройная, зеленая была.
<p align=justify> Метель ей пела песенку:
«Спи, елочка, бай-бай».
Мороз снежком укутывал:
«Смотри, не замерзай!»
</body>
</html>
```

2. Сохраните изменения, выполнив команду **Файл/ Сохранить**. НЕ ЗАКРЫВАЯ **Блокнот**, откройте этот файл в окне программы просмотра *Internet Explorer* с помощью команды **Файл/ Открыть**, чтобы увидеть изменения нажмите кнопку  **Обновить**.

3. Приведите свою страничку к предложенному виду, добавив, необходимые теги.

Привет! Это моя первая страничка!

В лесу родилась елочка.


В лесу она росла.

Зимой и летом стройная, Зеленая была. Метель ей пела песенку:

«Спи, елочка, бай-бай».

Мороз снежком укутывал:

«Смотри, не замерзай!»

4. Сохраните изменения, НЕ ЗАКРЫВАЯ **Блокнот**, просмотрите с помощью браузера *Internet Explorer* (чтобы увидеть изменения нажмите кнопку  **Обновить**).

5. Вставьте после фразы «Привет! Это моя первая страничка!» два заголовка:

Заголовок первого уровня: Меня зовут (впишите свое имя).

Заголовок второго уровня: Я учусь в классе (впишите 11 класс).

Заголовок третьего уровня: Это моя любимая песенка.

6. Сохраните изменения и просмотрите их с помощью браузера *Internet Explorer*.

7. Выровняйте заголовки следующим образом:

Заголовок первого уровня по левому краю,

Заголовок второго уровня по центру,

Заголовок третьего уровня по правому краю.

8. Сохраните изменения и просмотрите их с помощью браузера *Internet Explorer* .

### Подведение итогов

- Сделайте в тетради конспект теоретического материала.
- Ответьте на следующие вопросы:
  1. Что такое HTML, для чего используется?
  2. Какими способами можно создать Web-страницу?
  3. Назовите обязательные и необязательные элементы HTML-документа. Каково их назначение?
  4. Каким образом можно сохранить HTML-документ, созданный в Блокноте? Как его просмотреть?
  5. Что делать, если закрыли Блокнот, а вам необходимо внести изменения в HTML-документ?

### Домашнее задание:

1. Выучите конспект.
2. Сформулируйте название вашего будущего сайта, подберите к нему материал.

## Практическая работа № 3 Оформление текста в HTML

Цель: научиться оформлять и форматировать текст Web-страницы.

### Форматирование шрифта

| Назначение   | Формат                                | Значение аргументов (вместо звездочки *)  |
|--|---------------------------------------|---|
| Жирный   | <B> <i>текст</i> </B>                 |   |
| Подчеркнутый   | <U> <i>текст</i> </U>                 |   |
| Курсив   | <I> <i>текст</i> </I>                 |   |
| Верхний индекс                                       | <SUB> <i>текст</i> </SUB>             |   |
| Нижний индекс  | <SUP> <i>текст</i> </SUP>             |   |
| Размер шрифта(от 0 до 7)                             | <FONT SIZE=*> <i>текст</i> </FONT>    | 0, 1, 2 , 3, 4, 5 ,6 ,7   |
| Цвет шрифта(задается названием цвета или его кодом ) | <FONT COLOR="*"> <i>текст</i> </FONT> | См. ниже значения   |
| Гарнитура шрифта (задается название шрифта)          | <FONT FACE="*"> <i>текст</i> </FONT>  | Arial, Arial Black, Comic Sans MS, Monotype Corsiva, Courier New, Times New Roman и др. |

### Таблица значений цветов

| Название              | Код     | Имя    | Название          | Код     | Имя     |
|-----------------------|---------|--------|-------------------|---------|---------|
| <i>черный</i>         | #000000 | black  | <i>серебряный</i> | #C0C0C0 | silver  |
| <i>темно-бордовый</i> | #800000 | maroon | <i>красный</i>    | #FF0000 | red     |
| <i>зеленый</i>        | #008000 | green  | <i>известь</i>    | #00FF00 | lime    |
| <i>оливковый</i>      | #808000 | olive  | <i>желтый</i>     | #FFFF00 | yellow  |
| <i>темно-синий</i>    | #000080 | navy   | <i>голубой</i>    | #0000FF | blue    |
| <i>фиолетовый</i>     | #800080 | purple | <i>фуксия</i>     | #FF00FF | fuchsia |

|              |         |      |              |         |       |
|--------------|---------|------|--------------|---------|-------|
| <i>чирок</i> | #008080 | teal | <i>аква</i>  | #00FFFF | aqua  |
| <i>серый</i> | #808080 | gray | <i>белый</i> | #FFFFFF | white |

### Бегущая строка

`<MARQUEE> текст </MARQUEE>` — направление движения – справа-налево.

Scroll – стандартное движение от правого края к левому — бесконечный цикл. Число циклов можно ограничить:

`<MARQUEE LOOP=n BENAUTION=scroll>текст</MARQUEE>`

slide – надпись один раз пробегает от правого края к левому и там остаётся.

alternate – движение от правого края страницы к левому и обратно, бесконечный цикл. Определение ширины участка, занимаемого бегущей строкой:

`<MARQUEE WIDTH=n>текст</MARQUEE>`, где n – ширина той части страницы, на которой расположена бегущая строка.

#### Задание 1. Оформление шрифта HTML-документа.

1. Откройте созданный вами файл **Пример.html** и преобразуйте его следующим образом:

```
<html> <title> первые стихи </title>
<body text="#0000ff">
<p align=center>
<font face="Arial Black"> <font color="#ff0000"> <big> Привет!
Это моя первая страничка! </big> </font>
<font color="#ffff00">
<h1 align=left>Меня зовут </h1> <h2
align=center>Я учусь в классе </h2>
<h3 align=right>Это моя любимая песенка</h3> </font>
<p align=left> <font face="Comic Sans MS">
В лесу родилась <i>елочка. </i><br>
В лесу она росла.</font><br> <font
face="Courier New"> Зимой и летом
стройная, <br>
<font color="green"> Зеленая</font> была.<br> </font>
<b> Метель </b> ей пела песенку:<br>
<font face="Monotype Corsiva"> «Спи, елочка, бай-бай»</font> <br>
<b> <i>Мороз </b> </i>снежком укутывал:<br>
<font face="Monotype Corsiva"> «Смотри, не
замерзай!»</font>
</body>
</html>
```

2. Сохраните изменения, НЕ ЗАКРЫВАЯ **Блокнот**, просмотрите с помощью браузера

**Internet Explorer** (чтобы увидеть изменения нажмите кнопку  **Обновить**).

3. Подчеркните фразу «Зимой и летом стройная».
4. Выделите все «елочки» полужирным курсивом.
5. Измените, размер шрифта на 5 единиц для фразы «Смотри, не замерзай!».
6. Сделайте бегущей строкой фразу «Привет! Это моя первая страничка!».

7. Сохраните изменения и просмотрите их с помощью браузера *Internet Explorer* .
8. Для третьего заголовка внесите следующие изменения:  
**<h3 align=right>**  
**<P>**  
**<MARQUEE scrollDelay=28 behavior=alternate bgColor="gold" height=66>**  
**<font color="#ff0000"> Это моя любимая песенка </font>**  
**</MARQUEE></P></B></FONT><I><FONT face=Arial size=4> </h3>**  
**</font>**
9. Сохраните изменения и просмотрите их с помощью браузера *Internet Explorer* .

#### Цвет фона и текста

| Назначение  | Формат                | Значение аргументов (вместо звездочки *)  |
|---|-----------------------|---|
| Цвет фона (задается названием цвета или его кодом )   | <BODY BGCOLOR="*">    | См. цвет шрифта   |
| Цвет текста (задается названием цвета или его кодом ) | <BODY TEXT="*">       |   |
| Фоновое изображение                                   | <BODY BACKGROUND="*"> | Вместо * прописывается название файла в формате <b>bmp</b> Например:<br><BODY BACKGROUND="цветы.bmp"> |

#### Вставка изображений

| Назначение                            | Формат   | Значение аргументов (вместо звездочки *)                                   |
|---------------------------------------|--|--|
| Вставка изображений                   | <IMG SRC="*">  | Вместо * прописывается название файла в формате <b>bmp, jpg</b>            |
| Выравнивание текста около изображения | <IMG SRC="*" ALIGN=" top"><br><IMG SRC="*" ALIGN=" bottom"><br><IMG SRC="*" ALIGN=" middle"><br><IMG SRC="*" ALIGN=" left"><br><IMG SRC="*" ALIGN=" right "> |  |
| Изображение как ссылка                | <A HREF=адрес> <IMG SRC="*"> </A>  | В качестве <i>адреса</i> может быть: адрес документа, адрес сайта, e-mail. |

#### Задание 2. Оформление фона HTML-документа, вставка картинки.

1. Откройте созданный вами файл **Пример.html**.
2. Сделайте цвет фона голубой.
3. После фразы «В лесу родилась» вставьте изображение елочки (*елка.bmp*).
4. Сохраните изменения и просмотрите их с помощью браузера *Internet Explorer* .
5. Попробуйте все способы выравнивания текста около изображения, не меняя место вставки картинки. Запишите в тетрадь назначение всех аргументов выравнивания текста около изображения.
6. Вставьте в качестве фона файл *шары.jpg*.
7. Сохраните изменения и просмотрите их с помощью браузера *Internet Explorer* .

#### Подведение итогов

□ Сделайте в тетради конспект теоретического материала. □

Ответьте на следующие вопросы:

1. Какие существуют параметры форматирования шрифта в HTML-документе?
2. Какие существуют способы выравнивания текста около изображения?
3. Можно ли в качестве фона для Web-страницы использовать картинку?

**Домашнее задание:**

1. Выучите конспект.
2. Разработайте структуру вашего будущего сайта, распределите материал, продумайте и подберите оформление.

## Практическая работа № 4 Списки, гиперссылки и якоря в HTML

Цель: научиться создавать различные виды списков.

### Виды списков

| Назначение  | Формат  | Значения атрибутов  |
|---|---|---|
| Маркированный   | <pre>&lt;UL *&gt; &lt;LI&gt; элемент списка ... &lt;LI&gt; элемент списка &lt;/UL&gt;</pre> | устанавливает тип маркера. type="disc"<br>type="square"<br>type="circle"  |
| Нумерованный<br>start="" устанавливает<br>начальный номер в текущем<br>списке,<br>type="" устанавливает тип<br>маркера. | <pre>&lt;OL *&gt; &lt;LI&gt; элемент списка ... &lt;LI&gt; элемент списка &lt;/OL&gt;</pre> | A — маркер в виде прописных букв, a<br>— маркер в виде строчных букв,<br>I — маркер в виде больших римских<br>цифр,<br>i — маркер в виде маленьких римских<br>цифр,<br>1 — маркер в виде арабских цифр. |
| Многоуровневый  |   |   |
| (используют теги<br>нумерованного и<br>маркированного списков)  |   |   |

#### Задание 1. Создание списков.

- Откройте текстовый редактор **Блокнот**.
- Создайте в нем следующий документ:
 

```
<HTML>
<TITLE> Списки</TITLE>
<BODY>
<UL>
<LI>Сливы
<LI>Яблоки
<LI>Груши
</UL>
</BODY>
</HTML>
```
- Сохраните этот файл в своей папке, с именем **Списки.html**, просмотрите его в браузере **Internet Explorer**.
- Добавьте в этот документ предложенный нумерованный список:
  - Информатика,
  - Английский язык, 3. Математика,
  - История.
- Сохраните изменения и просмотрите их в браузере **Internet Explorer**.
- Добавьте в этот документ предложенный многоуровневый список. Измените вид маркеров, как в образце.

Компания «Аэрофлот» предлагает следующие услуги:

Продажа и бронирование авиабилетов.

Туристические поездки:

- I. Испания,
- II. Греция,
- III. Таиланд,
- IV. Коста-Рика,
- V. Мексика и другие страны мира.

Иммиграция и визы.

Обучение за рубежом.

7. Сохраните изменения и просмотрите их в браузере *Internet Explorer* .

**Задание 2. Самостоятельная творческая работа над своим сайтом.**

**Подведение итогов**

- Сделайте в тетради конспект теоретического материала.
- Ответьте на следующие вопросы:
  1. Что такое список?
  2. Какие виды списков вы знаете?
  3. Как можно создать нумерованный список?
  4. Как можно создать многоуровневый список?

**Домашнее задание:**

1. Выучите конспект.

### **Практическая работа № 5 Таблицы и фреймы в HTML**

**Цель:** научиться создавать различные виды таблиц, оформлять их; научиться разделять экран на несколько областей согласно сценарию (с помощью фреймов).

#### **HTML-таблицы**

| <b>Назначение</b> | <b>Формат</b> | <b>Значение аргументов (вместо звездочки *)</b> |
|-------------------|---------------|---|
|-------------------|---------------|---|



|   |                            |   |
|---|----------------------------|---|
| Задание таблицы   | <TABLE *<br>>текст</TABLE> | <b>width="" height=""</b> ширина и высота таблицы в пикселах или в % от ширины и высоты окна, соответственно; <b>border=""</b> ширина рамки. Если значение - ноль, то рамка не требуется; <b>align=""</b> задает положение данных в ячейках таблицы по горизонтали: <i>left</i> — данные прижать к левому краю ячеек, <i>center</i> — данные расположить по центру ячеек, <i>right</i> — данные прижать к правому краю ячеек.             |
| Заголовок таблицы   | <CAPTION * > .             | <b>align=""</b> задает место заголовка: <i>top</i> — заголовок над таблицей, <i>bottom</i> — заголовок под таблицей.  |
| Тег строки таблицы. Если тег пуст - строка пустая. Заданные атрибуты строки подавляют атрибуты таблицы.                       | <TR * >текст</TR>          | <b>bgcolor=""</b> задает цвет фона ячеек строки; <b>bordercolor=""</b> задает цвет рамки к ячейкам строки; <b>valign=""</b> задает положение данных в ячейке по вертикали. <i>baseline</i> — данные располагать в ячейках по базовой линии строки, <i>bottom</i> — данные прижимать к нижнему краю ячеек строки, <i>middle</i> — данные располагать по центру ячеек в строке, <i>top</i> — данные прижимать к верхнему краю ячеек строки. |
| Тег ячейки «Заголовок»: заголовок столбца или строки. По умолчанию текст в этой ячейке показывается жирным шрифтом по центру. | <TH * >текст</TH>          | <b>align=""</b> задает положение данных в ячейке "Заголовок" по горизонтали. <i>left</i> — заголовок прижать к левому краю ячейки, <i>center</i> — заголовок расположить по центру ячейки, <i>right</i> — заголовок прижать к правому краю ячейки; <b>valign=""</b> задает положение данных в ячейке "Заголовок" по вертикали. <i>bottom</i> — заголовок прижать к нижнему  |

|  |   |  |
|--|---|--|
|  |   | <p>краю ячейки, <i>middle</i> — заголовок расположить по центру ячейки,<br/> <i>top</i> — заголовок прижать к верхнему краю ячейки;<br/> <b>colspan=""</b> задает число ячеек, объединяемых в одну на строке;<br/> <b>rowspan=""</b> задает число ячеек, объединяемых в одну в столбце.</p>  |
| <p>Тег столбца таблицы. Если тег пуст или содержит значение <b>&amp;nbsp;</b> — ячейка пустая.</p> | <p><b>&lt;TD * &gt; текст &lt;/TD&gt;</b></p> | <p><b>width="" height=""</b> ширина и высота ячейки в пикселах или в % от ширины и высоты окна, соответственно;<br/> <b>align=""</b> задает положение данных в ячейке по горизонтали.<br/> <i>left</i> - данные прижать к левому краю ячейки,<br/> <i>center</i> - данные располагать по центру ячейки, <i>right</i>- данные прижать к правому краю ячейки; <b>valign=""</b> задает положение данных в ячейке по вертикали. <i>baseline</i> — данные расположить в ячейке по базовой линии,<br/> <i>bottom</i> — данные прижать к нижнему краю ячейки,<br/> <i>middle</i> — данные расположить по центру ячейки, <i>top</i> — данные прижать к верхнему краю ячейки;<br/> <b>bgcolor=""</b> задает цвет фона ячейки;<br/> <b>bordercolor=""</b> задает цвет рамки к ячейке;<br/> <b>colspan=""</b> задает число ячеек, объединяемых в одну на строке; <b>rowspan=""</b> задает число ячеек, объединяемых в одну в столбце;<br/> <b>nowrap</b> отмена переноса слов на другую строку внутри ячейки.</p> |

Горизонтальная табличная строка задается тегами **<TR> текст </TR>**, и является основой построения таблицы. Внутри строк задаются столбцы таблицы тегами **<TD> текст </TD>**, число пар этих тегов в строке таблицы соответствует числу столбцов.

### Задание 2. Создание таблиц.

1. Откройте текстовый редактор **Блокнот**
2. Создайте в нем следующий документ:

```

<HTML>
<BODY>
<table border=2>
<tr>
<td>Первая колонка</td>
<td>Вторая колонка</td>
<td>Третья колонка</td>
</tr>
<tr>
<td>Первая колонка</td>
<td>Вторая колонка</td>
<td>Третья колонка</td>
</tr>
</table>
</BODY>
</HTML>

```

3. Сохраните изменения и просмотрите их с помощью браузера *Internet Explorer*

4. Преобразуйте документ следующим образом:

```

<HTML>
<BODY>
<table border=2 &amp bordercolor="#800000" &amp bgcolor="#008080 ">
(заданы три атрибута: двойная линия рамки и ее цвет, цвет фона таблицы)
<tr>
<td bgcolor="yellow">
Первая колонка</td>
<td>Вторая колонка</td>
<td bgcolor="yellow">Третья колонка</td>
</tr>
<tr>
<td>Первая колонка</td>
<td bgcolor="yellow">Вторая колонка</td>
<td>Третья колонка</td>
</tr>
</table>
</BODY>
</HTML>

```

5. Сохраните изменения и просмотрите их с помощью браузера *Internet Explorer*.

### Задание 3. Создание сложных таблиц.

1. Создайте предложенные таблицы на одной Web-странице, разделив их пустыми абзацами:

**colspan=n** задает число ячеек, объединяемых в одну на строке.

**rowspan=n** задает число ячеек, объединяемых в одну в столбце.

|                 |                |
|-----------------|----------------|
| Пгода на завтра |                |
| днем            | отличный денек |

|       |                     |
|-------|---------------------|
| ночью | тихая звездная ночь |
|-------|---------------------|

|                  |                             |
|------------------|-----------------------------|
| Погода на завтра | днем — отличный денек       |
|                  | ночью — тихая звездная ночь |

2. Сохраните изменения и просмотрите их с помощью браузера *Internet Explorer*.

#### Задание 4. Самостоятельная творческая работа над своим сайтом.

##### Подведение итогов

- Сделайте в тетради конспект теоретического материала.
- Ответьте на следующие вопросы:
  1. Как создать таблицу размером 3 строки на 2 столбца?
  2. Как объединить или разбить ячейки?
  3. Как залить таблицу цветом?
  4. Как сделать «невидимой» таблицу?
  5. С помощью каких символов можно записать несколько атрибутов?

С помощью фреймов экран разделяется на несколько областей, в каждой из которых отображается содержимое отдельной страницы и даже Web-узла.

Второстепенные документы вставляются в базовый с помощью тегов:

```
<FRAMESET COLS="размер разделяемых колонок">
```

```
<FRAME SRC="имя файла">
```

```
</FRAMESET>
```

##### Примеры простейших фреймов:

Горизонтальное деление экрана производится при помощи атрибута `rows`, а вертикальное при помощи атрибута `cols`. Значения атрибутов могут быть выражены в пикселах, процентах или \* для обозначения оставшейся части экрана.

`cols = 50%, 50%`;

Деление области просмотра на равные правую и левую части.

`cols = 20%, 80%`;

Деление области просмотра на неравные правую и левую части.

`rows = 100, 20%, *`

Деление области просмотра на три части: первой отведено 100 пикселей, второй — 20% доступного пространства, а третьей — все остальное

`cols = *, 3*`

Деление области просмотра на неравные правую и левую части. Правая часть в три раза шире левой.

`cols=25%,75%`

Деление области просмотра аналогично предыдущему примеру.

#### Задание 1. Создание макета страницы с фреймами.

1. Создайте HTML-документ следующего содержания:

```
<HTML>
<HEAD>
<TITLE>FRAMES</TITLE>
</HEAD>
<FRAMESET FRAMEBORDER=1 BORDER=2 ROWS=150,*>
<FRAME SCROLLING="YES" NORESIZE SRC="page1.html" NAME="1">
<FRAMESET FRAMEBORDER=1 BORDER=2 COLS=150,*>
<FRAME SCROLLING="YES" NORESIZE SRC="page2.html" NAME="2">
<FRAME SCROLLING="YES" NORESIZE SRC="page3.html" NAME="3">
</FRAMESET>
</FRAMESET>
</HTML>
```

2. Сохраните изменения и просмотрите их с помощью браузера *Internet Explorer* .

**Задание 2. Самостоятельная творческая работа над своим сайтом.**

### **Подведение итогов**

- Сделайте в тетради конспект теоретического материала.
- Ответьте на следующие вопросы:
  1. Что такое фреймы? Для чего они используются?

## **Практическая работа № 6 Формы в HTML**

**Цель:** научиться создавать различные элементы форм.

Формы представляют собой интерактивные элементы HTML, позволяющие разработчикам страниц взаимодействовать с посетителями. С их помощью пользователь может возвращать комментарии по поводу посещения определенного узла, пересылать запросы или регистрироваться. Разработчик задает вопросы, создавая форму, а пользователь отвечает на них, заполняя её.

Форма создается при помощи различных тэгов и атрибутов, заключенных в пару **<FORM> текст </FORM>**.

- атрибут **Method** определяет способ пересылки данных сценарию **CGI**. Здесь протокол **GET** выбран по умолчанию, но в большинстве случаев разработчики пользуются протоколом **POST**, который позволяет передавать большие объемы данных.
- атрибут **Action** определяет путь к сценарию **CGI** или адрес электронной почты.
- атрибут **Enctype** определяет способ кодирования содержимого формы. Другими словами, он сообщает браузеру о способе кодирования информации перед отсылкой серверу. По умолчанию используется значение **x-www-form-encoded**.  
для сценария: `<FORM method="get" или "post" action="URL сценария" ></FORM>`

## Формы

Формы для почты: `<FORM method="get" или "post" action="mailto:адрес" ></FORM>` При помощи элемента `<TEXTAREA>` создается область для ввода и просмотра текста. Может использоваться и не в составе формы, а как самостоятельные детали страницы. Область ввода помогает сэкономить место благодаря полосам прокрутки. Может иметь атрибуты:

- **name** — задаёт ключевое слово, по которому сценарий может обращаться к его содержимому.
- **Rows** — задаёт высоту области в строках.
- **Cols** — задаёт ширину области в символах.

### Задание 1. Создание области для ввода текста.

1. Откройте текстовый редактор **Блокнот**.
2. Создайте в нем следующий документ:

```
<HTML>
<BODY>
<FORM><H3>Введи текст
<TEXTAREA name="ключевое слово" rows=5 cols=30>Область для ввода текста
</TEXTAREA></H3>
<INPUT type="reset" value="очистка"></FORM>
</BODY>
</HTML>
```

3. Сохраните изменения и просмотрите их с помощью браузера **Internet Explorer**

Элемент `<SELECT>` может принимать форму раскрывающегося списка или меню элементов. Имеет вложенный тэг `<OPTION>` и атрибуты:

- **Name** — задаёт имя.
- **Size** — задаёт максимальное количество элементов списка, одновременно отображаемых на экране.
- **Multiple** — задаёт возможность одновременного выбора нескольких значений.

Элемент `<OPTIONS>` задает возможные варианты выбора меню `<SELECT>`  
`<OPTION value="n" selected>` значение имеет атрибуты:

- **selected** — задаёт изначально выбранное слово.
- **value** — задаёт значение выбранного слова для сценария.

### Задание 2. Создание раскрывающегося списка.

1. Добавьте в созданный вами HTML-документ следующий фрагмент: `<P>Выберите:`  
`<SELECT size=1>`  
`<OPTION selected value=1> Первый </OPTION>`  
`<OPTION value=2> Второй </OPTION>`  
`<OPTION value=3> Третий </OPTION>`  
`<OPTION value=4> Четвертый </OPTION>`  
`</SELECT>`

<P> 2. Сохраните изменения и просмотрите их с помощью браузера Internet Explorer . Элемент <OPTGROUP> применяется для **логической** группировки элементов <OPTION> внутри тэга <SELECT> имеет атрибут **label**. Элемент <INPUT> является базовым для всех элементов формы. Он используется для внедрения в форму кнопок, графических изображений, флажков, переключателей, паролей и текстовых полей. Несмотря на внешние отличия форм все они пересылают сценарию CGI данные в виде пар **имя:значение**. Элемент может иметь восемь атрибутов обозначаемые как **type**:

- **TEXT** текстовое поле, используется для ввода информации. Сюда вводятся имена, адреса, должности, телефоны, хобби, и данные практически любого типа. Элемент может иметь атрибуты:
  - **maxlength**. Задаёт максимально допустимую длину вписываемого значения в символах.
  - **size**. Задаёт максимально допустимую длину поля в символах.
  - **value**. Задаёт значение по умолчанию, которое можно менять.
- **PASSWORD** поле, в котором вместо вводимых символов отображаются звездочки. Элемент может иметь атрибуты:
  - **maxlength**. Задаёт максимально допустимую длину вписываемого значения в символах.
  - **size**. Задаёт максимально допустимую длину поля в символах.
  - **value**. Задаёт значение по умолчанию, которое можно менять.

```
<INPUT type="PASSWORD" name="PASSWORD_BOX" maxlength="35" size="20">
```

- **CHECKBOX** Флажки используются для предоставления возможности пользователю ответить односложно: **да/нет истина/ложь больше/меньше** и т.д. Выглядит обычно в виде крестика или птички. Элемент может иметь атрибуты:
  - **checked**. Задаёт начальный статус флажка по умолчанию.
  - **value**. Задаёт значение по умолчанию, которое можно менять.

```
<INPUT type="checkbox" name="send_mail" value="yes" checked>
```

- **RADIO** Переключатели во многом напоминают флажки, отличаясь лишь более широкими функциональными возможностями выбора. В группе переключателей может быть выбран лишь один. Для каждого переключателя указывается отдельный элемент **INPUT**

```
Visa <INPUT type="radio" name="payment_type" value="visa">
```

```
Mastercard <INPUT type="radio" name="payment_type" value="mastercard">
```

```
American Express <INPUT type="radio" name="payment_type" value="AmEx" checked>
```

- **SUBMIT** Щелчок на этой кнопке приводит к пересылке содержимого формы сценарию, который был задан атрибутом **action** в элементе <FORM>. С помощью кнопок можно вычислять сумму, загружать страницы, пересылать данные, сбрасывать значения.

```
<FORM method="get" или "post" action="mailto:name@domen.ru" >
```

```
<INPUT type="submit" value="послать"></FORM>
```

- **RESET** Кнопка используется для восстановления значений, заданных по умолчанию. Если значение по умолчанию не предусмотрено, то оно просто обнулится. Ширина кнопки может меняться в зависимости от других элементов. Имеет так же атрибут **value**.

```
<INPUT type="reset" value="очистка">
```

□ **IMAGE** Во многом похож на кнопку **SUBMIT**, только в качестве кнопки используется изображение. Одним из преимуществ является возможность передачи координат щелчка пользователя, что позволяет организовать карту изображений. Элемент может иметь атрибуты:

- **src**. Задаёт URL файла с изображением.
- **align**. Задаёт выравнивание изображения относительно текста при помощи значений **TOP**, **MIDDLE** и **BOTTOM**.
- **name**. Задаёт имя карты, которое так же пересылается сценарию вместе с координатами.

```
<INPUT type="image" src="кнопка.gif">
```

Элемент **<ISINDEX>** Это самый простой элемент, позволяющий создать подобие формы и ввода строки, содержащей текст и генерации запроса.

Пример: **<ISINDEX prompt=" строка для ввода критерия поиска">** Допустим что на текущей странице задан базовый URL при помощи элемента

**<BASE href="URL поискового средства в Internet">** тогда, если пользователь введет в поле ключевые слова для поиска **слово1,слово2,слово3**, то браузер сформирует и отошлет запрос для поисковой машины сервера в виде: **http://www.название.домен/?слово1+слово2+слово3** Если поисковая программа сервера поддерживает стандартный синтаксис запроса с использованием знаков **?** и **+**, поиск будет осуществлен.

Элемент **<LABEL>** применяется для альтернативного задания информации для управляющих полей формы. Поддерживает атрибут **for**, который связывает элемент **<LABEL>** с другим элементом формы, а значение атрибута **for** должно совпадать со значением атрибута **id** связанного управляющего элемента.

### Задание 3. Создание управляющих полей.

1. Добавьте в созданный вами HTML-документ следующий фрагмент:

```
<P>
<FORM action="URL" method="post">
<LABEL for="firstname">Имя: </LABEL>
<INPUT type="text" id="firstname"><BR>
<LABEL for="lastname">Фамилия: </LABEL>
<INPUT type="text" id="lastname"><BR>
<LABEL for="email">e-mail: </LABEL>
<INPUT type="text" id="email"><BR>
<INPUT type="radio" name="sex" value="Мужской">Мужской<BR>
<INPUT type="radio" name="sex" value="Женский">Женский<BR>
<INPUT type="submit" value="Отправить"> <INPUT type="reset">
</FORM>
```

2. Сохраните изменения и просмотрите их с помощью браузера Internet Explorer .

### Задание 4. Самостоятельная творческая работа над своим сайтом.

#### Подведение итогов

- Сделайте в тетради конспект теоретического материала.
- Ответьте на следующие вопросы:
  1. Что такое формы? Для чего они используются?
  2. Какие элементы форм вы знаете? Как их создать?
  3. Какие элементы форм чаще всего встречаются на сайтах (и на каких)?



## Практическая работа №7 Каскадные таблицы стилей CSS.

Цель: научиться создавать связанную таблицу стилей CSS и модифицируете главную страницу учебного сайта «Компьютер».

**Задание 1.** Создайте в папке *comp* текстовый документ *styles.css*, который будет в дальнейшем использоваться как связанная таблица каскадных стилей.

Откройте *Блокнот*.

1. Введите тэги:

`<STYLE type="text/css">` начало содержимого связанной таблицы стилей

`<!--` В данном случае комментарии языка HTML используются для сокрытия CSS-кода от старых браузеров, не поддерживающих язык CSS.

Здесь будет находиться основное содержимое таблицы стилей

`--!>`

`</STYLE>` конец содержимого связанной таблицы стилей

2. Сохраните документ в папке *comp* под именем *styles.css*

**Задание 2.** Осуществите в связанной таблице макет веб-страницы с «плавающим» вложенным блоком (рис.3).

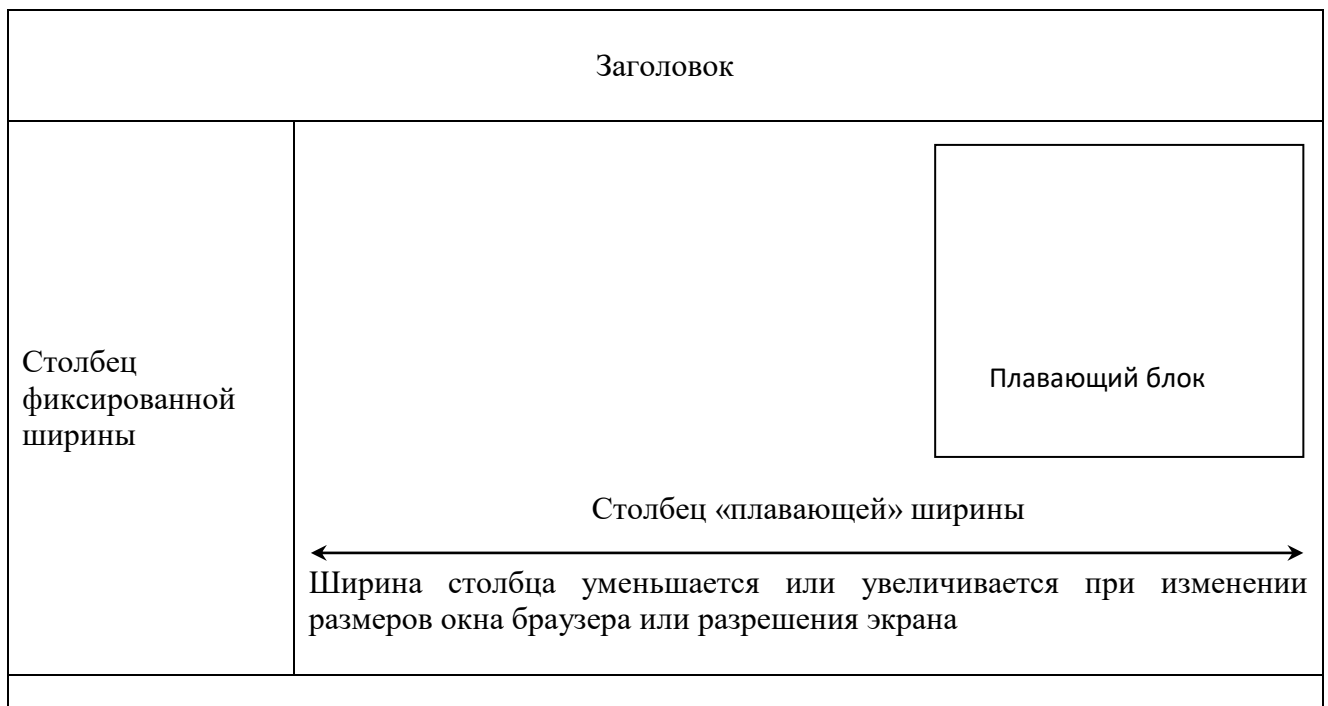


Рис. 3

*Справочный материал.* Структуру этого макета веб-страницы в связанной таблице стилей *styles.css* будут осуществлять следующие элементы (селекторы):

- `#masthead` — заголовок;
- `#nav` — левый столбец фиксированной ширины, предназначенный для навигации по сайту с помощью вертикальной панели гиперссылок;
- `#content` — правый столбец «плавающей» ширины, предназначенный для основного содержимого;
- `#content #box` — «плавающий» блок (в нашем случае он будет содержать рисунок);
- `#footer` — нижний колонтитул.

Перечисленные селекторы могут иметь различные значения следующих свойств:

- `float` — позволяет элементу с плавающим размещением выравниваться по левому или правому краю;
- `width` — задает ширину содержимого элемента в процентах или в пикселях (px);
- `margin-top` — задает значение для верхнего поля элемента;
- `margin-left` — задает значение для левого поля элемента;
- `margin-right` — задает значение для правого поля элемента;
- `clear` — используется для запрещения обтекания элементов с плавающим размещением.

Для выполнения задания:

1. Откройте документ *styles.css* в папке *comp*.
2. Добавьте между существующими тэгами код, осуществляющий структуру выбранного макета веб-страницы:

```
<STYLE type="text/css">
```

```
<!-- #masthead {
    width: 100%;
    margin-top: 10 px;
    margin-left: 10 px;
    margin-right: 5 px;
}
#nav {
    float: left;
    width: 25%;
    margin-top: 5 px;
    margin-left: 5 px;
```

```

        margin-right: 5 px;
    }
    #content {
        float: left;
        width: 72%;
        margin-top: 5 px;
        margin-right: 5 px;
    }
    #content #box{
        float: right;
        width: 300 px;
        margin-top: 5 px;
        margin-right: 5 px;
    }
    #footer {
        width: 100%;
        clear: both;
        margin-top: 5 px;
        margin-left: 10 px;
    }
--!>
</STYLE>

```

3. Сохраните изменения в документе *styles.css*.

**Задание 3.** Задайте в связанной таблице стилей стили для форматирования текста.

В данном задании вы определите стили для следующих селекторов:

- `body` — задает стилизацию текста основного содержимого веб-страницы;
- `h1` — определяет стили для заголовков веб-страницы;
- `p` — осуществляет форматирование абзацев;
- `.p1` и `.p2` — элементы класса `p` с новым форматированием абзацев.

Перечисленные селекторы имеют следующие свойства:

- font-family — название шрифта;
- font-size — размер шрифта;
- color — цвет шрифта;
- text-align — выравнивание текста.

1. Откройте документ *styles.css*, если он закрыт.
2. Добавьте в конец кода (до тэгов `--!> </STYLE>`) код, осуществляющий форматирование текста:

```

body {

    font-family: Times,Arial, sans-serif;

    font-size: 14 pt;

}

h1 {

    font-family: Times,Arial, sans-serif;

    font-size: 24 pt;

    color: blue;

}

p {

    text-align: center;

}

.p1 {

    text-align: left;

}

.p2{

    text-align: right;

}

```

3. Сохраните изменения в документе *styles.css*.

**Задание 4.** Добавьте в связанную таблицу стилей код, осуществляющий список без маркеров.

Список без маркеров будет использоваться как основа для вертикальной панели навигации по сайту с помощью гиперссылок. Значение none свойства list-style-type селектора ul позволяет избавиться от маркеров при организации списка. Для выполнения этого задания:

1. Добавьте в документ *styles.css* следующий код:  

```
ul {  
  
    list-style-type:none;  
  
}
```
2. Сохраните изменения.

**Задание 5.** Добавьте в связанную таблицу стилей стили для состояний ссылок.

*Справочный материал.* Для описания состояния ссылок используются следующие селекторы:

- `link` — состояние ссылки до ее активации;
- `visited` — состояние, в которое переходит ссылка после ее активации;
- `hover` — состояние, когда указатель мыши наведен на ссылку;
- `active` — состояние, возникающее при щелчке на ссылке для ее активации.

В нашем случае описаны свойства селекторов ссылок:

- `color` — цвет ссылки;
- `text-decoration` — декоративное оформление текста ( значение `none` используется для удаления стандартного подчеркивания у ссылок, значение `underline` подчеркивает выбранный текст);
- `background-color` — цвет фона.

Для выполнения задания:

1. Добавьте в документ *styles.css* следующий код:

```
a {  
  
    color: orange;  
  
    text-decoration: none;  
  
}  
  
a:link {  
  
    color: orange;  
  
}  
  
a:visited {  
  
    color: yellow;  
  
}  
  
a:hover {  
  
    color: fuchsia;  
  
    text-decoration: underline;
```

```

        background-color: aqua;
    }

    a:active {
        color: red;
    }

```

2. Закройте документ *styles.css*, сохранив изменения.

**Задание 6.** Создайте главную страницу учебного сайта «Компьютер», используя макет с «плавающим» блоком (см. рис.1). В столбце фиксированной ширины осуществите вертикальную панель навигации по сайту с гиперссылками на страницы *software.htm*, *glossary.htm* и *anketa.htm*. В «плавающем» блоке разместите изображение компьютера. В нижнем колонтитуле разместите гиперссылку на электронный адрес автора.

*Справочный материал.* Селектор `<div id=содержимое</div>` позволяет поместить содержимое в нужный контейнер макета веб-страницы. Свойство `id` указывает название нужного контейнера, описанного в связанной таблице стилей.

Вновь создаваемая главная страница веб-сайта «Компьютер» будет связана с таблицей каскадных стилей *styles.css* с помощью элемента `link`. Для элемента `link` используется несколько атрибутов, перечисленных ниже:

- `rel` — атрибут связи, который описывает связь ссылки. В данном случае связь осуществляется с основной таблицей стилей, поэтому используется значение `stylesheet`;
- `type` — определяет тип языка и используемый формат, в данном случае, `text/css`;
- `href` — указывает имя связанной таблицы стилей. В данном случае указано только имя файла *styles.css*, поскольку оба документа (главная страница веб-сайта «Компьютер» и таблица стилей) находятся в одном каталоге.

Для выполнения задания:

1. Откройте программу Блокнот.
2. Введите тэги, выделенные полужирным шрифтом:

**<html>**

**<head>**

**<title> Компьютер</title>**

**</head>**

**<link rel="stylesheet" type="text/css" href="styles.css">**Обращение к связанной таблице стилей *styles.css*.

**<body>**

**<div id="masthead">** Начало контейнера заголовка макета веб-страницы.

```

<h1><p>Все о компьютере</p></h1>Заголовок веб-страницы.
</div>Конец контейнера заголовка макета веб-страницы.
<div id="nav"> Начало контейнера левого столбца макета веб-страницы.
<ul>Начало списка вертикальной панели навигации.
<li><a href="software.htm"> Программы </a></li>
<li>< a href="glossary.htm">Словарь</a></li>
<li>< a href="anketa.htm">Анкета </a></li>
</ul>Конец списка вертикальной панели навигации
</div>Конец контейнера левого столбца макета веб-страницы.
<div id="content">Начало столбца с «плавающей» шириной.
<div id="box">Начало контейнера «плавающего» блока.

</div>Конец контейнера «плавающего» блока.
<p class="p1">На этом сайте вы сможете получить различную информацию о
компьютере и его программном обеспечении.
</p>
<p class="p2">Терминологический словарь познакомит вас с компьютерными
терминами, а также вы сможете заполнить анкету.
</p2>
</div>Конец столбца с «плавающей» шириной.
<div id="footer">Начало нижнего колонтитула.
<p><a href="mailto:olsor2007@yandex.ru">Написать автору</a></p>
</div>Конец нижнего колонтитула.
</body>
</html>

```

3. Сохраните документ с именем *index1.htm* в каталоге *comr* в своей папке.
4. Просмотрите документ *index1.htm* в браузере.
5. Попробуйте изменить границы окна и проследите за изменениями, происходящими с веб-страницей.
6. Перейдите по ссылкам на страницы веб-сайта *Программы, Словарь, Анкета*.
7. Закройте веб-страницу.

## Практическая работа № 8 Публикация сайта в интернете.

Познакомиться с сервисами размещения сайта в сети интернет. Научиться размещать сайт на бесплатном сервисе narod.ru

### Ход выполнения работы:

#### 1. Введение

Знакомство с Интернетом часто начинают с поисковых систем. Не забывает новый Интернет-пользователь совершать экскурсии в мир WWW, где он встречается со множеством сайтов. Проходит какое-то время, и этот посетитель хочет принять участие в наполнении Интернета. Чаще всего это выражается в его желании разместить на просторах Интернета свой сайт, то есть найти место в Интернете для его размещения. Это место – земля, -- на которой будет стоять ваш дом(сайт), называется **хостинг**.

Хостинг бывает двух типов: **бесплатный** и **платный**. Чтобы выяснить, какой лучше, попытаемся сравнить оба типа.

| Платный хостинг                                     | Бесплатный хостинг   |
|---|--|
| Выдаётся своё доменное имя, например: ProstoSite.ru | Сайт будет называться так: ProstoSite.NAROD.ru               |
| Надёжный сервер                                     | Разные «неполадки» с сервером                                |
| Невозможно удаление сайта клиента из базы данных    | Удаление сайта клиента из базы данных без объяснения причины |
| Оперативная служба поддержки                        | Медленная служба поддержки                                   |
| Поддержка технологий PHP, MySQL и т.д.              | Данные технологии поддерживаются не всегда                   |

Преимущество платного хостинга перед бесплатным очевидно, но нет смысла тратить деньги на платный хостинг, когда сайт только в начале пути, поэтому поговорим о бесплатном хостинге.

Подобных хостингов существует множество. Немного позже мы рассмотрим все достоинства и недостатки основных из них. Но сначала о мышеловке, которая скрыта за этой «бесплатностью». Подвох заключается в следующем: при размещении на бесплатном хостинге сайт будет автоматически использован как рекламная площадка для этого хостинга.

Выражаться эта реклама может разными способами. Например, при заходе на сайт появляется всплывающее окно с рекламой, либо на каждой странице сайта будет добавлен сверху большой баннер и т.д.

А бывает, что так называемый бесплатный хостинг оказывается бесплатным только на определённый срок, по прошествии которого придётся либо заплатить, либо ваш сайт удалят из базы данных.



Но, несмотря на всё это, бесплатный хостинг – великая вещь, потому что есть возможность размещения своего сайта совершенно бесплатно. А это значит, что сайт будет доступен для просмотра любому человеку, выходящему в Интернет.

Итак, рассмотрим несколько популярных и, по словам пользователей, надёжных бесплатных хостингов.

Самый популярный бесплатный хостинг – [www.narod.ru](http://www.narod.ru). Этот хостинг предоставляет под сайт 100 Мбайт. Есть чат, гостевая книга и анкета. Имеется много готовых шаблонов. К сожалению, не поддерживает никаких «наворотов» типа PHP или SSI. Из рекламы – только небольшое окошко в правом верхнем углу, да и то с крестиком. Имя для сайта **Название\_вашего\_сайта.narod.ru**. Хостинг неплохой, поэтому и популярный. Настолько популярный, что имя для сайта придётся придумывать довольно долго, так как практически все имена уже заняты.

Следующий хостинг -- [www.boom.ru](http://www.boom.ru). Место под сайт – 50 Мбайт. Есть готовые шаблоны. Нет ни PHP, ни SSI, вообще ничего нет. Имя для сайта **Название\_вашего\_сайта.boom.ru**. Главный недостаток – это навязчивая реклама в виде большого баннера в нижней части окна, который при прокрутке остаётся неподвижным.

Хостинг [www.by.ru](http://www.by.ru). Под сайт даётся неограниченное место. Есть гостевая книга. Поддерживает SSI. Не имеет службы поддержки, частые сбои на сервере. Доменное имя **Название\_вашего\_сайта.by.ru**.

Хостинг [www.newmail.ru](http://www.newmail.ru). Под сайт даёт 16 Мбайт. Есть форум. Предоставляет несколько доменных имён:

**Название\_вашего\_сайта.newmail.ru.**

**Название\_вашего\_сайта.orc.ru.**

**Название\_вашего\_сайта.hotmail.ru.**

**Название\_вашего\_сайта.nm.ru.**

Для более-менее серьёзных проектов не подходит, так как не поддерживает ни одну из технологий. Рекламы нет. Конечно, это не все бесплатные хостинги, существуют и другие.

## **Практическая часть выполнения работы.**

### **Часть 1.**

1. Используя программу WebSiteX5 создайте сайт, посвященный красотам родного края (о городе Улан-Удэ или Республике Бурятия или об озере Байкал или Тункинской долине и т.д.)

**Внимание! При создании сайта указывайте доменное имя, полученное при регистрации на хостинге Yandex.Диск.**

**Например вы зарегистрировались на Yandex.Диске под логином name\_2012@yandex.ru. Тогда доменное имя Вашего сайта будет**

следующим [name 2012.narod.ru](http://name.2012.narod.ru) или [name 2012.narod2.ru](http://name.2012.narod2.ru). Вот это доменное имя и используйте при создании сайта в программе WebSiteX5Free.

## Часть 2.

Размещение сайта на хостинге [www.narod.ru](http://www.narod.ru).

**Внимание!** Данный пункт используйте, если Вы не регистрировались в сервисе Яндекс.Диск или Яндекс-почта. Если у вас есть аккаунт на Яндекс, то просто войдите в него.

### 2.1. . Регистрация на хостинге:

Зайдём на сайт хостинга [www.narod.ru](http://www.narod.ru).

Если же вы не зарегистрированы на Яндекс почте по прейдите по адресу Narod.ru. На главной странице в левом верхнем углу есть ссылка **Регистрация**, после щёлчка по которой, заполнить предлагаемую форму. В форме необходимо указать свой логин, пароль для доступа, фамилию, имя, e-mail, а также имя для сайта., которое нужно будет ввести в соответствующее поле латинскими буквами. **2. Переход по ссылке:**

После завершения регистрации обычно высылается письмо на электронный почтовый ящик, который был указан. В письме повторно указывается логин и пароль и даётся ссылка на панель управления вашего сайта. Зайти по ссылке.

### 3. Загрузка файлов на хостинг:

На открывшейся странице необходимо найти ссылку **Создать сайт**. Перейти по данной ссылке на следующую страницу. На этой странице выберите стиль вашего сайта (выбирайте любой, т.к в дальнейшем мы установим свой сайт) и нажимаете **Далее**. Во вновь открывшейся странице выбираем структуру сайта и нажимаем **Далее**. На следующем шаге вводим любой текст и нажимает кнопку **Опубликовать** (в левом верхнем углу окна).

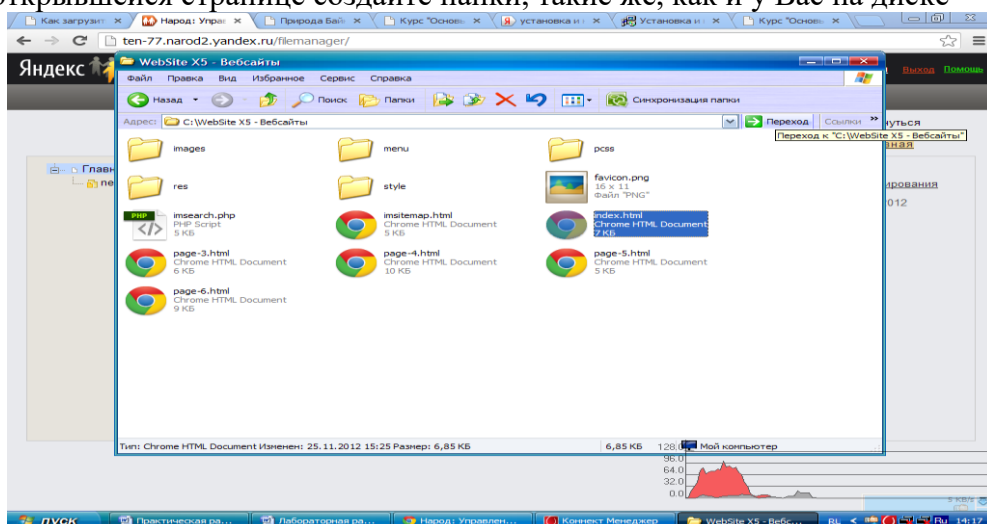
Вот теперь мы переходим к самому главному- публикации Вашего сайта (созданного ранее в программе WebSiteX5Free).

Вверху окна (в режиме конструктора) раскройте список



Выберите пункт **Управление файлами**.

В открывшейся странице создайте папки, такие же, как и у Вас на диске



С.:

Для этого используйте кнопку **Новая папка**.

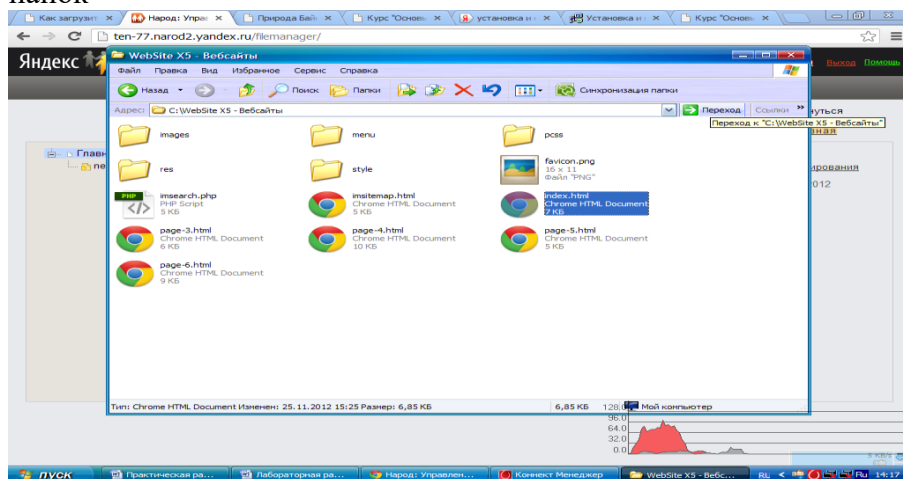
Для загрузки файлов выберите пункт **Загрузить файл**. В появившемся окне нажимаете кнопку **Выбрать файл** и в открывшемся окне выбираете необходимый файл и нажимаете **ОК**.

**Внимание! Загружайте файлы в те папки на хостинге, что и папки в которых они находились на диске C:!!!**

Файлы находящиеся вне



папок



загружаете непосредственно на страницу **Главная**.

## 5. Отладка работы сайта

Просмотрите результаты своей работы нажав кнопку **Опубликовать**

**Практическая работа № 9 Разработка веб-сайтов с использованием сценариев JavaScript и DHTML**

**Цель:** Научиться динамически изменять информацию на веб-странице с помощью выборки элементов DOM дерева.

**Необходимое оборудование и программное обеспечение:** персональный компьютер с операционной системой windows v.7-10, редактор кода (visual studio code или аналог), браузер (Google Chrome или аналог).

#### **Вопросы по самоподготовке**

1. Понятие DOM дерева
2. Назначение узлов (нод, node)
3. Свойство childNodes
4. Поиск элементов в DOM дереве
5. На основе DOM-дерева, представленного на рисунке приложения 1, создайте HTML-код.

**Теоретическая часть:** При разработке веб-страницы необходимо управлять DOM элементами. Для этого их нужно вначале получить. Наиболее часто используемые методы – это `document.getElementById`, `querySelectorAll` и `querySelector`.

**querySelector** - это метод, который вернет первое вхождение элемента (Element) документа, который соответствует указанному тегу, или селектору, или группе селекторов. Если совпадений не найдено, вернет null.

Основное предназначение метода `querySelector` обратиться к тегу и после этого, мы можем либо получить данные из тега, либо послать данные в тег в любой атрибут, который только существует в этом теге.

`querySelector` отличается от `getElementById` (выбирает только по id) - тем, что с помощью `querySelector`, можно обратиться к любому элементу DOM :

по классу (class), по тегу, по id, по атрибуту.

`querySelectorAll` находит все элементы и возвращает их список.

## Порядок выполнения работы:

Сверстать страницу согласно рисунку 1. Для верстки можно использовать фреймворк bootstrap или flex верстку. Т.е. на странице должно быть 3-4 карточки товаров, согласно варианту. Изначально текстовый блок и кнопка очистить корзину должны быть скрыты.

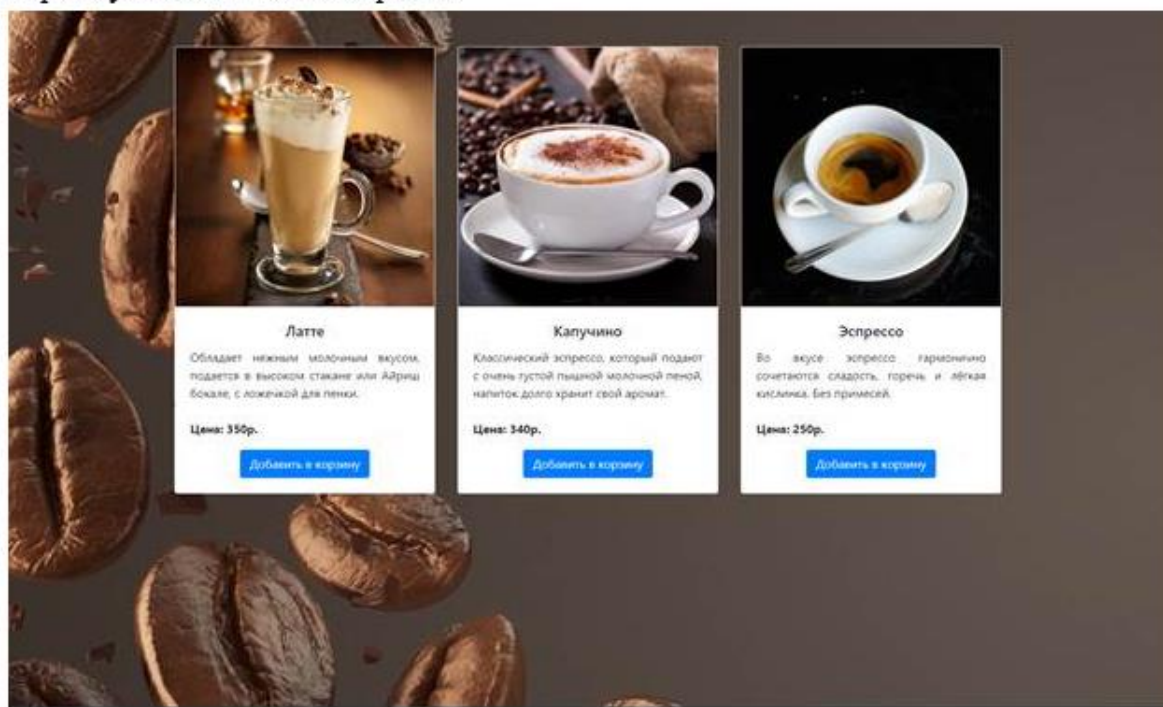


Рисунок 1

При нажатии на кнопку «Добавить в корзину», должна появляться кнопка «Очистить корзину» и текстовый блок, с содержанием информации о товаре согласно рисунку 2.

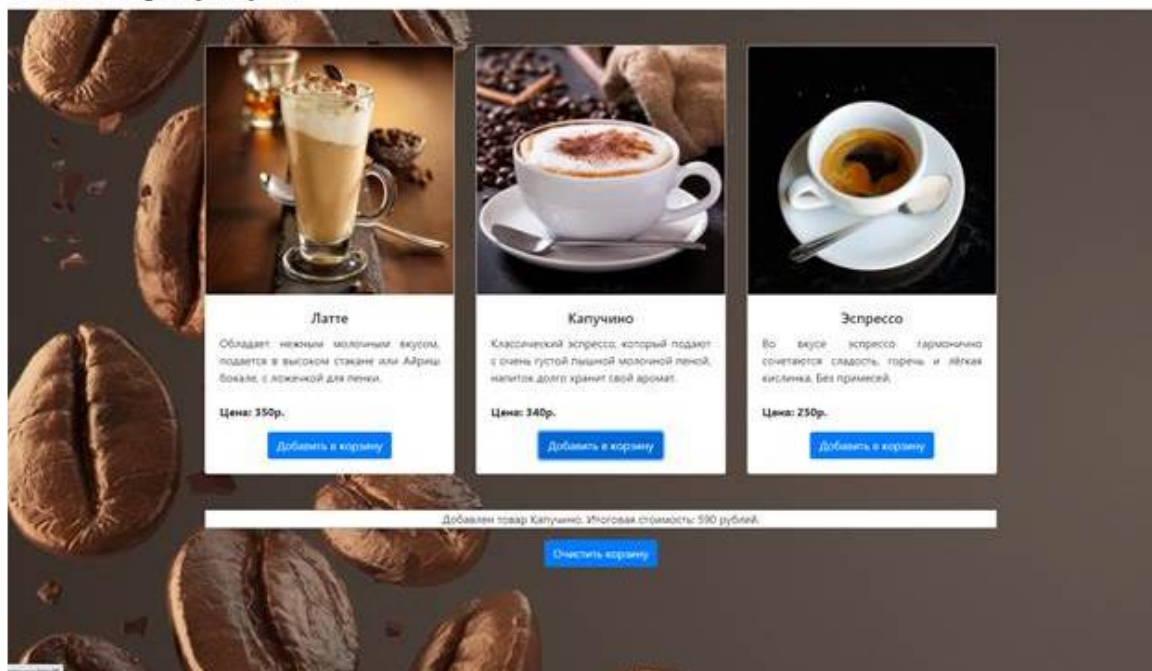


Рисунок 2

Примерный код скрипта файл script.js размещен в приложении 2

### Этапы выполнения практической работы:

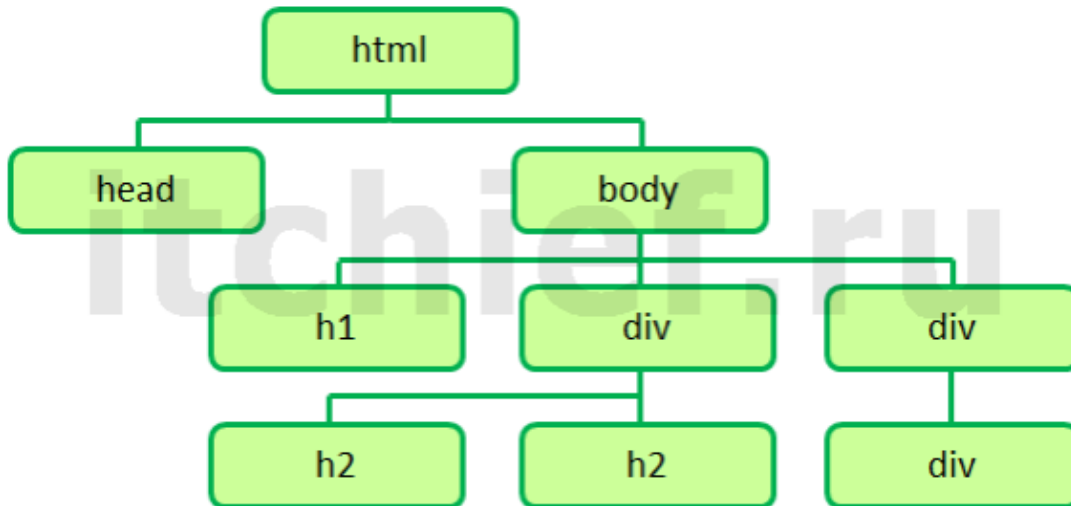
1. Разработать файл index.html, взаимодействующий с данным скриптом
2. Усовершенствовать приложение, добавив выборку по QuerySelectorAll
3. Создать страничку по отдельному товару, передавая в качестве параметра id товара или услуги
4. Оформить отчет

### Контрольные вопросы

1. Назначение метода querySelector
2. Метод innerHTML
3. Различия между методами document.getElementById, querySelectorAll и querySelector
4. Метод addEventListener
5. Есть `<p id="elem1">text1</p>` и получить абзац #elem1 и поменять ему текст с помощью innerHTML.
6. Получить абзац с классом elem и поменять ему текст:
7. `<p class="elem">text</p>`



## Приложение 1



## Приложение 2

```
let lattey = document.querySelector("#Латте");
let cappuccino = document.querySelector("#Капучино");
let espresso = document.querySelector("#Эспрессо");

let textBox = document.querySelector("#textBox");

let clearBtn = document.querySelector("#clearBtn");
clearBtn.hidden = true;
textBox.hidden = true;

let totalPrice = 0;

lattey.addEventListener('click', function() {
    totalPrice += 350;
    textBox.hidden = false;
    textBox.innerHTML = 'Добавлен товар Латте. Итоговая стоимость: '+totalPrice+' рублей.';
    clearBtn.hidden = false;
});
cappuccino.addEventListener('click', function() {
    totalPrice += 340;
    textBox.hidden = false;
```

```

        textBox.innerHTML = 'Добавлен товар Капучино. Итоговая стоимость: '+totalPrice+' рублей.';
        clearBtn.hidden = false;
    });
    espresso.addEventListener('click', function() {
        totalPrice += 250;
        textBox.hidden = false;
        textBox.innerHTML = 'Добавлен товар Эспрессо. Итоговая стоимость: '+totalPrice+' рублей.';
        clearBtn.hidden = false;
    });
    clearBtn.addEventListener('click', function() {
        totalPrice = 0;
        textBox.hidden = false;
        textBox.innerHTML = 'Корзина была очищена. Итоговая стоимость: '+totalPrice+' рублей.';
    });

```

## № 10 Совместное использование HTML, CSS и JavaScript

1. Разработать тематический сайт содержащий не менее 3-х страниц, оформленных в едином стиле.
2. С использованием HTML5/CSS
3. Используя CSS+JS добавить на свой сайт **форму расчёта** (в соответствии с вариантом вашей работы) - например:

Вариант: Сайт компании ЖКХ - расчёт стоимости энергоресурсов по введённым показателям счетчиков. Вводятся на форме Начальные и конечные показатели электроэнергии, холодной и горячей воды, и по фиксированному тарифу считается сумма платежа. (Расчет в JavaScript)

4. **Анимация меню** (выпадающее, всплывающее), анимация формы расчёта, анимация кнопки расчёт - переход из одного цвета в другой.
5. Форма расчёта должна отображаться **в блоке по центру экрана с затемнением основной области**.

Примеры можно посмотреть [здесь](#), [здесь](#), и [здесь](#)

Варианты:

- 1 - Сайт компании продавца одежды
- 2 - Сайт компании продавца it услугу
- 3 - Сайт компании продавца услуг по бухгалтерскому сопровождению
- 4 - Сайт налоговой службы
- 5 - Сайт системы GPS мониторинга
- 6 - Сайт компании "Марсианские курорты"
- 7 - Сайт компании производителя Электро-автомобилей
- 8 - Сайт корпорации оборонного комплекса
- 9 - Сайт президента Зимбабве
- 10 - Сайт компании разработки лунный горных месторождений
- 11 - Сайт компании выращивающей водоросли для производства косметики
- 12 - Сайт государственной службы безопасности
- 13 - Сайт ТСЖ (собственников жилья - большого жилого комплекса)
- 14 - Сайт компании ритуальных услуг
- 15 - Сайт производителя Андроидов
- 16 - Сайт компании разработчика искусственного интеллекта



- 17 - Сайт производителя пончиков
- 18 - Сайт финансовой корпорации ДеньгиДаром
- 19 - Сайт банды мексиканских преступников
- 20 - Сайт по продаже скрепок
- 21 - Сайт автоматизированного приема жалоб на жизнь
- 22 - Сайт компании Транспортный Портал
- 23 - Сайт компании Стар-Трек
- 24 - Сайт анимационной студии
- 25 - Сайт музыкальной группы
- 26 - Сайт школы иностранных языков (Албанский, Суахили)
- 27 - Сайт кафедры университета (направление: клонирование, биотехнологии, генная инженерия)

## **№ 11 Создание Web-страницы «Моя маленькая родина», используя контейнерный Web-Дизайн**

### **№ 12 Динамический HTML**

#### *Цель работы:*

Изучить основы клиентского скриптового языка JavaScript, работу с объектной моделью документа DOM (Document Object Model), работу с cookie, познакомиться с возможностями, предоставляемые фреймворком jQuery.

#### *Задание:*

Создать клиентский скрипт на языке JavaScript, выполняющий действия в соответствии с вариантом. Использовать возможности, предоставляемые объектной моделью документа DOM, использовать фреймворк jQuery (или аналог).

#### *Вариант 1:*

Сборка мозаики. Элементы мозаики перетаскиваются указателем мыши. Предусмотреть возможность автоматической сборки. Положение элементов в собранной мозаике фиксировано. Среди прочего использовать возможности, предоставляемые фреймворком jQuery.

#### *Вариант 2:*

Калькулятор цвета. Отобразить таблицу, фоны ячеек которой окрашены в web-гарантированные цвета. По щелчку левой кнопки мыши на образце цвета изменяется цвет текста документа, по щелчку правой кнопки мыши — цвет фона документа, также появляется окно с шестнадцатеричным кодом цвета. Предусмотреть три поля для задания цветовых составляющих и отображения цвета, в отдельном, например, окне. Среди прочего использовать возможности, предоставляемые фреймворком jQuery.

### *Вариант 3:*

Игра «Жизнь».

Игра моделирует жизнь поколений гипотетической колонии живых клеток на прямоугольном игровом поле, которые выживают, размножаются или погибают в соответствии со следующими правилами.

Для каждого поколения (шага игры) применяются следующие правила: каждая живая клетка, количество соседей которой меньше двух или больше трёх, погибает; каждая живая клетка, у которой от двух до трёх соседей, живёт до следующего хода; каждая мёртвая клетка, у которой есть ровно три соседа, оживает. Соседи клетки – это все соседние с ней клетки по горизонтали, вертикали и диагонали, всего восемь соседей.

Правила применяются ко всему игровому полю одновременно, а не к каждой из клеток по очереди. То есть подсчёт количества соседей происходит в один момент перед следующим шагом, и изменения, происходящие в соседних клетках, не влияют на новое состояние клетки.

Среди прочего использовать возможности, предоставляемые фреймворком jQuery.

### *Вариант 4:*

Создание эффекта анимированного текста. В тексте символ за символом изменяется цвет и размер очередного символа. Предыдущий символ становится прежним. Предусмотреть возможность выбора основного и дополнительного цвета и размера символов. Среди прочего использовать возможности, предоставляемые фреймворком jQuery.

### *Вариант 5:*

За указателем мыши перемещаются часы и дата (предусмотреть возможность установки часов и даты). Среди прочего использовать возможности, предоставляемые фреймворком jQuery.

### *Вариант 6:*

Тест на скорость реакции. После щелчка по кнопке в тестовом поле случайным образом, через случайные промежутки времени появляются изображения, по которым нужно успеть щелкнуть. Попадание обозначается каким-либо образом (например, «взрывом» изображения). Тестирование можно прекратить щелчком по кнопке, но не ранее, чем через некоторый отрезок времени. Выводится результат — процент удачных щелчков. Среди прочего использовать возможности, предоставляемые фреймворком jQuery.

### *Вариант 7:*

Три линейки с бегунками для каждой цветовой составляющей. Изменение положения каждого из бегунков влечет за собой изменение

цвета фона документа. Среди прочего использовать возможности, предоставляемые фреймворком jQuery.

*Вариант 8:*

Калькулятор на четыре действия (с нажимающимися кнопками). Среди прочего использовать возможности, предоставляемые фреймворком jQuery.

*Вариант 9:*

Игра «Падающие мячи». По игровому полю сверху вниз в случайном порядке падают мячи, которые нужно ловить корзиной, передвигаемой с помощью клавиатуры горизонтально вдоль нижней границы игрового поля. Игру можно начать и прекратить щелчком по соответствующей кнопке. Со временем скорость падения мячей увеличивается. После остановки игры выводится результат — процент пойманных мячей. Среди прочего использовать возможности, предоставляемые фреймворком jQuery.

*Вариант 10:*

Просмотр набора изображений со сменой подписей к изображениям с помощью кнопок «Назад» и «Далее». При просмотре первого изображения блокируется кнопка «Назад», при просмотре последнего — кнопка «Далее». Среди прочего использовать возможности, предоставляемые фреймворком jQuery.



*Фото 1. Фудзияма.*

*Порядок выполнения лабораторной работы:*

1. Создать html-документ.
  2. Написать скрипт в соответствии с заданным вариантом. Для справки по языку Javascript можно использовать источники, расположенные по адресам <http://learn.javascript.ru> и <http://javascript.ru>. Для справки по фреймворку jQuery можно использовать источники, расположенные по адресам <http://jquery.com> и <http://jquery-docs.ru>.
  3. Протестировать созданный документ.
- Содержание отчета (отчет в электронном виде):
  - отчет сохранить в файле с именем АВТ-000 Иванов (лр4).doc;
  - титульный лист;

- цель работы;
- задание;
- порядок выполнения лабораторной работы
- разметка html-документа с исходным кодом скрипта; □ скриншот html-документа; □ выводы по работе.

#### *Теоретические сведения*

Теоретические материалы доступны по адресу <http://learn.javascript.ru>.

### **№ 13 - 14 Разработка веб-сайтов с использованием PHP.**

**Цель работы:** формирование умений использования PHP

Литература

- **Учебник PHP 4.2 (RUS)**

Поясняющая информация

PHP - это действующий на стороне сервера встраиваемый в HTML язык, имеющий синтаксис, близкий к языку Си. Язык PHP дает возможность вставлять в файлы HTML инструкции языка PHP для создания динамического содержания. Эти инструкции обрабатывает препроцессор-интерпретатор PHP и заменяет их тем содержимым, которое производит этот код. Стандартное расширение файла с PHP-программой - php.

Принцип такой же, как и с SSI.

#### **Синтаксис PHP**

PHP-код включаются в html-код в следующем виде:

```
<?PHP текст_кода ?>
```

или

```
<?PHP  
текст_кода;  
?>
```

Например, вставка содержимого файла будет выглядеть так:

```
<?PHP include "ssi/top.php"; ?>
```

#### **Коментарии**

PHP поддерживает комментарии 'C', 'C++' и оболочки Unix. Например:

```
<?php
echo "This is a test"; // Это однострочный комментарий в стиле c++
/* Это многострочный комментарий,
это ещё одна его строка */
echo "This is yet another test";
echo "One Final Test"; # Это комментарий в shell-стиле
?>
```

## echo

```
<?php echo "Эта информация будет выведена в HTML";?>
```

## Присвоение значений переменным

Переменные в программах на PHP, отделяются символами \$.

```
$city = "Kazan";
```

city - переменная

Kazan - значение

## Некоторые операции

инкремента/декремента;

++\$a Pre-increment Увеличивает \$a на 1, затем возвращает \$a.

\$a++ Post-increment Возвращает \$a, затем увеличивает \$a на 1.

--\$a Pre-decrement Уменьшает \$a на 1, затем возвращает \$a.

\$a-- Post-decrement Возвращает \$a, затем уменьшает \$a на 1.

арифметические:

\$a + \$b Сложение Сумма \$a и \$b.

\$a - \$b Вычитание Разность \$a и \$b.

\$a \* \$b Умножение Произведение \$a и \$b.

\$a / \$b Деление Частное от деления \$a на \$b.

\$a % \$b Modulus Целочисленный остаток от деления \$a на \$b.

Выражения сравнения вычисляются в 0 или 1, означая FALSE или TRUE (соответственно).

PHP поддерживает > (больше), >= (больше или равно), == (равно), != (не равно), < (меньше)

и <= (меньше или равно). Эти выражения чаще всего используются внутри условных операторов, таких как if.

сравнения:

\$a == \$b равно TRUE, если \$a равно \$b.

\$a != \$b не равно TRUE, если \$a не равно \$b.

\$a <> \$b не равно TRUE, если \$a не равно \$b.

\$a < \$b меньше TRUE, если \$a строго меньше \$b.

\$a > \$b больше TRUE, если \$a строго больше \$b.

\$a <= \$b меньше или равно TRUE, если \$a меньше или равно \$b.

\$a >= \$b больше или равно TRUE, если \$a больше или равно \$b.

строковые:

Имеются две строковые операции. Первая - операция ('.'), которая возвращает объединение из правого и левого аргументов. Вторая - операция присвоения ('.='), которая присоединяет правый аргумент в левому аргументу.

```
$a = "Hello ";  
$b = $a . "World!"; // теперь $b содержит "Hello World!"
```

```
$a = "Hello ";  
$a .= "World!"; // теперь $a содержит "Hello World!"
```

## Некоторые операторы

### if/else

```
if ($a > $b) {  
    print "a больше чем b";  
} else {  
    print "a не больше чем b";  
}
```

### while

```
$i = 1;  
while ($i <= 10) {  
    print $i++; /* будет печататься значение $i++ до тех пор пока $i не больше 10 */  
}
```

### for

```
for ($i = 1; $i <= 10; $i++) {  
    print $i; /* будет выполняться цикл от $i=1 $i <= 10 с шагом $i++ */  
}
```

## Установка cookie с помощью PHP.

setcookie - отправляет куки.

*Вызовы этой функции должен быть перед тэгами <html> или <head>!!!*

Синтаксис:

```
int setcookie (string name [, string value [, int expire [, string path [, string domain [, int secure]]]])
```

Пример:

```
setcookie ("student", "pupkin");
```

## Получение и использование cookie сервером с помощью PHP.

После того как куки установлены, доступ к ним может быть получен при загрузке страницы через массив `$_COOKIE`

Cookie автоматически декодируется и присваивается переменной с тем же именем, что и имя куки. Для просмотра содержимого нашей тестовой куки в скрипте просто используйте один из следующих примеров:

```
echo $student;
echo $_COOKIE["student"];
```

## Получение и использование переменных посланных методом GET с помощью PHP.

К переменным GET доступ можно получить через массив `$_GET`

```
$stud=$_GET['student'];
```

## Необходимые компоненты для создания счетчика каждой страницы

Счетчик должен вестись для каждой страницы, значит нужно хранить значения для каждой страницы отдельно, будем хранить их в отдельном файле.

Так как файлов будет много, создадим для них отдельный каталог на локальном диске `/www/students/group/login/counter`.

Чтобы его использовать в программе присвоим переменной `$counter_path` путь к этому каталогу.

### Пример:

```
$counter_path = "путь к вашему каталогу";
```

Для автоматического формирования имени файлов для каждой страницы будем использовать URI этой страницы. Но в URI нам надо изменить все `/` на `_`, потому что имя файла не может содержать `/`.

`str_replace("что меняем", "на что меняем", строка в которой производится замена)` - функция замены

`getenv("переменной окружения")` - функция получает значение переменной окружения.

`REQUEST_URI` - переменная окружения URI

Присвоим исправленный URI переменной `$count_page`, т.е. имя создаваемого файла.

### Пример:

```
$count_page = str_replace("что меняем", "на что меняем", URI этой страницы);
```

```
getenv("REQUEST_URI") - URI этой страницы
```

Если файл (\$count\_page) уже существует, то есть страница посещалась ранее, то нужно считать значение счётчика из файла и увеличить его на единицу.

file("путь к каталогу/имя файла") - функция считывает содержимое текстового файла в массив, каждый элемент массива соответствует строке файла с символом '\n'

Откроем файл, и присвоим значения переменной \$contents

**Пример:**

```
$contents = @file("путь к каталогу/имя файла")
```

```
$count = $contents[0] + 1;
```

Перед вызовом функции file(...) стоит знак '@' - это сделано для того, чтобы PHP не выдавал предупреждения, если файл отсутствует.

Если файл не существует, то есть мы на этой странице впервые, то присвоить счётчику единичное значение.

Обработка этого условия должна выглядеть так:

```
if (присвоение содержимого файла переменной $contents) {  
увеличение счетчика на 1  
} else {  
присвоить счётчику единичное значение  
}
```

Теперь откроем файл для записи нового значения счётчика

fopen("/путь к каталогу/имя файла", "w") - открывает файл или URL.

Параметр 'w' означает - Открыть только для записи; поместить указатель в начало файла и усечь файл до нулевой длины. Если файл не существует, делается попытка создать его.

Создадим указатель файла

**Пример:**

```
$file = fopen("/путь к каталогу/имя файла", "w");
```

запишем новое значение счётчика

**Пример:**

```
fwrite (указатель файла, строка для записи в файл) - функция записи в файл
```

Закроем файл

```
fclose (указатель файла) - закрывает открытый указатель файла
```

После этого нужно вывести значение счетчика, применив echo.



## Практические задания

### Задание 1

1. Переделайте первую страницу (можно создать новую), применив PHP вместо SSI.
2. Сделайте, чтобы cookie передавались не с помощью META - данных, а с помощью setcookie.
3. Проверьте работоспособность на сервере.
4. Укажите, в виде комментариев, для чего предназначена каждая PHP команда.

### Задание 2

1. Создайте счетчик посещаемости страниц (файл count.inc). Для записи файлов со значением счетчика, используйте каталог counter (он у вас уже создан!!!).
2. Вставьте его с помощью include в botom.php.
3. Проверьте работоспособность на сервере.
4. Укажите, в виде комментариев, для чего предназначена каждая PHP команда.

*К сдаче лабораторной предоставляются: работающие страницы на сервере.*

## **№ 15 – 16 Создание интернет приложений на стороне сервера.**

**Цель работы:**

### **№ 17 Технология SSI. Регистрационная форма.**

**Цель работы:** формирование умений использования SSI.

Поясняющая информация

### **Server Side Include**

SSI расшифровывается как Server Side Include. SSI - это набор команд, включенных в страницу HTML, команды выполняются на сервере перед отсылкой запрошенной страницы клиенту.

При получении запроса на документ, содержащий SSI-директивы, HTTP-сервер обрабатывает документ, выполняет все SSI-директивы, а получившийся в результате документ возвращает клиенту. HTTP-сервер не проверяет автоматически все документы на наличие SSI-директив, а только те, которые относятся к типу, указанному в настройках сервера ( в нашем случае все файлы с расширением .shtml).

SSI-директивы включаются в html-код в виде комментариев определенного формата:

```
<!--#SSI_директива="параметры" -->
```

**echo var**

Используется для вывода значения переменной окружения.

```
<!--#echo var="Переменная"-->
```

например:

```
<!--#echo var="REMOTE_ADDR"-->
```

Некоторые переменные:

REMOTE\_ADDR - ваш IP адрес

REMOTE\_PORT - номер вашего порта

HTTP\_USER\_AGENT - ваш браузер

SERVER\_NAME - имя сервера

SERVER\_PORT - номер порта сервера

SERVER\_SOFTWARE - какой HTTP сервер установлен

SERVER\_PROTOCOL - протокол который поддерживает сервер

SERVER\_ADMIN - администратор сервера

HTTP\_REFERER - с какой страницы вы пришли

DOCUMENT\_URI - URI документа

DOCUMENT\_NAME - название файла текущего документа

QUERY\_STRING - строка запроса методом GET

LAST\_MODIFIED - время последней модификации файла

### **include file**

### **include virtual**

Включает содержимое указанного файла в html-файл в месте, где идет вызов данной команды. Отличие include file от include virtual :

include file - путь к файлу, относительно текущей директории. (То есть не может быть абсолютным путем к файлу (начинаться с / или содержать ../)).

include virtual - виртуальный путь к документу на сервере. Он может начинаться с /, но должен быть на том же сервере.

Пример:

```
<!--#include file="docext.html"-->
```

```
<!--#include virtual="/counters/spylog.txt"-->
```

```
<!--#include virtual="/cgi-bin/counter.pl" -->
```

Если с помощью include file или include virtual включается cgi-скрипт, то в содержимое страницы попадет вывод cgi-скрипта, а не его содержание.

С помощью этих директив вы можете сделать автоматическую сборку страниц. Если ваша страница состоит из блоков, и некоторые блоки одинаковые на всех страницах, то эти блоки можно вынести в отдельные файлы, а на страницах вставить эти директивы. Благодаря этому вы сможете внести изменения в одном файле, и информация на всех страницах изменится.

### **fsize file**

Вставляет размер указанного файла.

Например:

```
<!--#fsize file="karta.shtml"-->
```

### **flastmod file**

Вставляет время последней модификации указанного файла.

Например:

```
<!--#flastmod file="karta.shtml" -->
```

### **printenv**

Выводит все переменные окружения. Параметров не имеет. Вы можете вывести весь список переменных, которые вам доступны.

Список переменных окружения:

```
<!--#printenv -->
```

### **set var**

Устанавливает значение переменной окружения. Формат такой set

```
var="ИМЯ_ПЕРЕМЕННОЙ" value="ЗНАЧЕНИЕ".
```

Значение переменной TITLE:

```
<!--#set var="TITLE" value="Лабораторная работа 2" -->
```

Значение переменной SERVER\_NAME:

```
<!--#echo var="TITLE" -->
```

Последняя директива выведет "Лабораторная работа 2".

Если у вас "шапочка" (<head></head>) вынесена в отдельный файл, и в нем задан основной <title> сайта, то с помощью этой директивы вы сможете дополнить <title>.

Пример:

В отдельном файле содержащим <head></head> (например, top.shtml), пишем:

```
<title>Сервер Васи Пупкина: <!--#echo var="TITLE" --> </title>
```

В файле, куда предыдущий файл вставляется (до директивы вставки), пишем:

```
<!--#set var="TITLE" value="Лабораторная работа 2" -->
```

В результате получаем:

```
<title>Сервер Васи Пупкина: Лабораторная работа 2 </title>
```

### **if/else**

Применяется для управления выводом страницы по условию. Синтаксис такой:

```
<!--#if expr="УСЛОВИЕ1" -->
```

HTML-код, который будет выводиться, если УСЛОВИЕ1 истинно

```
<!--#elif expr="УСЛОВИЕ2" -->
```

HTML-код, который будет выводиться, если УСЛОВИЕ1 ложно, а УСЛОВИЕ2 истинно

```
<!--#else -->
```

HTML-код, который будет выводиться, если все условия ложны

```
<!--#endif -->
```

Условие - это либо строка, которая является истинной, если непустая, или набор операторов сравнения строк. Операторы могут быть =, !=, <, <=, > и >. Если вторая строка заключена в ""(слэши), то условие истинно, если в первой строке встречается хоть одно вхождение

второй строки. Можно объединять несколько операторов сравнения с помощью операторов &&(И) и ||(ИЛИ). Для группирования условий используются "("(скобки).

Пример:

```
string1 = string2
string1 != string2
string1 < string2
string1 <= string2
string1 > string2
string1 >= string2
```

Если string2 имеет форму /string/, то тогда она интерпретируется, как регулярное выражение.

Пример:

```
<!--#if expr="$HTTP_COOKIE=/user\=old/" -->
```

Данным примером проверяем, содержится ли в cookies user=old.

## goto

Команда goto оператор перехода на метку label.

Формат команды: <!--#goto = "<label>" -->

где <label> имя метки, определенное командой label.

Пример.

```
<!--#goto = "testlabel" -->
Эта строка не будет видна
<!--#label = "testlabel" -->
Эта строка будет видна
```

*Замечание: Между символами <!--#goto и знаком = обязательно должен стоять пробел.*

## label

Команда label устанавливает метку в документе, которая используется командами goto или if..goto

Формат команды: <!--#label = "<label>" -->

где <label> строка из не более чем 51 символа без пробелов, идентифицирующая место в документе. Задание метки не влияет на форматирование документа.

*Замечание: Между символами <!--#label и знаком = обязательно должен стоять пробел.*

## break

Команда break останавливает вывод документа, как только встречается.

Пример:

Эта строка будет видна

```
<!--#break -->
```

Эта строка не будет видна

Практические задания

### Задание 1

1. Создайте каталог /ssi/, для хранения включаемых файлов.
2. Создайте в нем пустые файлы (например: top.shtml, botom.shtml, left.htm, right.htm).
3. Переделайте первую страницу, вынеся верхнюю часть (включая верхнюю таблицу и "шапку" (<head> </head>)) и нижнюю (включая нижнюю таблицу), а также левую и правую часть соответственно в файлы, которые вы создали. Не забудьте изменить расширение на .shtml.
4. Проверьте работоспособность на сервере.
5. Сделайте, чтобы <title> формировался динамически.
6. На первой странице, выведите следующую информацию:  
размер файла  
дата последней модификации файла  
ваш IP адрес  
номер вашего порта  
ваш браузер  
имя сервера  
номер порта сервера  
какой HTTP сервер установлен  
протокол, который поддерживает сервер  
с какой страницы вы пришли  
URI документа  
название файла текущего документа
7. В исходниках первой страницы и включаемых файлов укажите, в виде комментариев, для чего предназначена каждая SSI команда.

### Задание 2

1. Создайте новую страницу. Выведите все переменные окружения.
2. Проверьте работоспособность на сервере.

### Задание 3

1. Приведите все остальные страницы к общему виду с помощью SSI, т.е. верхняя и нижняя часть должны быть одинаковы для всех страниц (кроме фреймовых страниц).
2. Пропишите на каждой странице дополнение к <title>, соответствующее данному документу.
3. Проверьте работоспособность на сервере.

*К сдаче лабораторной предоставляются: работающие страницы на сервере.*

**№ 18 Интернет-приложения и базы данных.**