

Министерство образования Республики Беларусь  
Учреждение образования  
«Белорусский государственный университет  
информатики и радиоэлектроники»

Факультет компьютерного проектирования

Кафедра инженерной психологии и эргономики

## **ЭРГОНОМИЧЕСКИЕ ВЕБ-ТЕХНОЛОГИИ**

*Рекомендовано УМО по образованию в области информатики  
и радиоэлектроники в качестве пособия для специальности  
1-58 01 01 «Инженерно-психологическое обеспечение  
информационных технологий» и направления специальности  
1-40 05 01-09 «Информационные системы и технологии  
(в обеспечении промышленной безопасности)»*

Минск БГУИР 2017

УДК 004.777:331.10.1(076)  
ББК (32.973.26-018.2+30.17)я73  
Э74

Авторы:

А. С. Панфилёнок, А. В. Телого, М. М. Меженная, К. Д. Яшин

Рецензенты:

кафедра интеллектуальных систем  
Белорусского национального технического университета  
(протокол №6 от 08.02.2016);

главный инженер общества с ограниченной ответственностью  
«Современные технологии связи» С. А. Олейников

**Эргономические веб-технологии : пособие / А. С. Панфилёнок [и др.]. –**  
Э74 Минск : БГУИР, 2017. – 64 с. : ил.  
ISBN 978-985-543-265-5.

Пособие посвящено вопросам разработки и сопровождения веб-приложений с учетом особенностей взаимодействия человека с информационными системами. Является методическим обеспечением выполнения лабораторных работ для студентов специальности 1-58 01 01 «Инженерно-психологическое обеспечение информационных технологий», направления специальности 1-40 05 01-9 «Информационные системы и технологии (в обеспечении промышленной безопасности)».

**УДК 004.777:331.10.1(076)**  
**ББК (32.973.26-018.2+30.17)я73**

**ISBN 978-985-543-265-5**

© УО «Белорусский государственный  
университет информатики  
и радиоэлектроники», 2017

## Содержание

<b>Введение</b> .....	4
<b>Тема 1.</b> Определение целей пользователей программных продуктов .....	5
<b>Тема 2.</b> Выработка стратегии решения проблем пользователей. Сторибоардинг .....	8
<b>Тема 3.</b> Варфрейминг и прототипирование интерфейса пользователя .....	11
<b>Тема 4.</b> Юзабилити-тестирование .....	14
<b>Тема 5.</b> Знакомство с программой Adobe Photoshop .....	17
<b>Тема 6.</b> Основы композиции элементов веб-приложения .....	20
<b>Тема 7.</b> Создание графических элементов веб-приложения.....	24
<b>Тема 8.</b> Работа с текстом в Adobe Photoshop .....	27
<b>Тема 9.</b> Создание графического макета веб-приложения. ....	34
<b>Тема 10.</b> Подготовка макета веб-приложения к верстке.....	38
<b>Тема 11.</b> Верстка макета веб-страниц .....	43
<b>Тема 12.</b> Работа с изображениями, ссылками, таблицами и списками в HTML ....	47
<b>Тема 13.</b> Каскадные таблицы стилей.....	50
<b>Тема 14.</b> Интерактивность в веб-приложении.....	53
<b>Тема 15.</b> Клиент-серверное взаимодействие .....	56
<b>Тема 16.</b> Поисковая оптимизация .....	60
<b>Литература</b> .....	64

## **Введение**

Настоящее пособие предназначено для проведения лабораторных работ по дисциплине «Эргономические веб-технологии» для студентов специальности 1-58 01 01 «Инженерно-психологическое обеспечение информационных технологий», направления специальности 1-40 05 01-9 «Информационные системы и технологии (в обеспечении промышленной безопасности)», а также для студентов других специальностей, направленных на подготовку инженеров-программистов со знаниями основ веб-технологий.

Пособие посвящено вопросам разработки и сопровождения веб-приложений с учетом особенностей взаимодействия человека с информационными системами: освещаются психологические особенности и проблемы взаимодействия пользователей с веб-приложениями, принципы проектирования и юзабилити-тестирования пользовательских интерфейсов веб-приложений. Подробно раскрываются темы по основам веб-дизайна и работе с веб-графикой, необходимые для разработки графического макета веб-приложения. Изучаются основы верстки веб-приложений (язык гипертекстовой разметки HTML) с подключением каскадных таблиц стилей CSS и использованием возможностей языка JavaScript, программирования серверной бизнес-логики на основе языка PHP, а также продвижения веб-приложений в поисковых системах на основе SEO-технологий.

# Тема 1

## Определение целей пользователей программных продуктов

*Цель:* формирование у студентов знаний и навыков проведения наблюдений и интервью для выявления целей пользователей программных продуктов.

### *План занятия*

1. Изучить теоретические сведения.
2. Выполнить практическое задание по лабораторной работе.
3. Оформить отчет и ответить на контрольные вопросы.

### *Теоретические сведения*

Создание программных продуктов требует исследований пользовательской аудитории, результаты которых преобразуются в набор требований к продукту, а затем реализуются в инфраструктуре интерфейса. Такой процесс разработки гарантирует, что проектирование происходит соответственно запросам пользователей.

Эффективным методом сбора качественных данных о пожеланиях пользователей является сочетание интервьюирования и наблюдения. Это дает проектировщику возможность задавать пользователю уточняющие вопросы и получать пояснения к тем ситуациям и действиям, которые он наблюдает.

Американские разработчики Хью Бейер и Карен Хольцблат создали технику исследования пользовательской аудитории, получившую название контекстного исследования (Contextual Inquiry). Техника контекстного исследования базируется на ремесленной (мастер и ученик) модели обучения: необходимо наблюдать за пользователем и задавать ему вопросы так, как если бы он был высококлассным мастером, а задающий вопросы – его новым учеником.

Рассмотрим четыре базовых принципа организации интервью.

1. *Контекст.* Следует взаимодействовать с пользователями и наблюдать за ними в естественной рабочей среде. Наблюдение за пользователями, когда они заняты своей деятельностью, и расспросы, происходящие в привычном для них окружении, помогают выявить самые важные особенности их поведения.

2. *Сотрудничество.* Интервью и наблюдение должны иметь характер совместного с пользователем исследования, в котором наблюдение деятельности сочетается с обсуждением ее особенностей.

3. *Интерпретация.* Работа проектировщика в большой степени сводится к выявлению того, что стоит за словами и поведением пользователей.

4. *Направленность*. Проектировщик должен направлять ход интервью в поисках данных, имеющих отношение к вопросам проектирования.

Алан Купер пересмотрел методику Бейера и Хольцблата с точки зрения оптимизации затрат временных и человеческих ресурсов и предложил следующие пути улучшения методики контекстных исследований.

1. *Сокращение продолжительности интервью*. Купер выяснил, что для сбора необходимых данных достаточно часовой беседы при условии, что запланировано достаточное количество интервью.

2. *Использование компактной команды проектировщиков*. Контекстное исследование подразумевает наличие большой команды проектировщиков, проводящей параллельно сразу несколько интервью, за которыми следует специальное совещание (дебрифинг) с участием всей команды. Причем более эффективно проводить сессии интервью последовательно с участием одних и тех же проектировщиков. Тем самым размер команды остается небольшим (два или три человека), вся команда получает возможность напрямую взаимодействовать с каждым из интервьюируемых пользователей, что позволяет максимально эффективно анализировать и синтезировать собранные данные.

3. *Определение целей в начале исследования*. Контекстное исследование рассчитано на процесс проектирования, принципиально ориентированный на задачи. При проведении интервью Купер предлагает прежде всего выявлять цели пользователей и назначать им приоритеты, а только затем приступать к выяснению того, какие задачи соответствуют этим целям.

Для фиксации того, что говорят и делают пользователи, многие юзабилити-специалисты применяют технические средства, такие как аудио- и видеозапись. Использование указанных технических средств не влияет на искренность обмена информацией, а также позволяет убедить заинтересованных лиц в реальности спорных или неожиданных результатов исследований.

### ***Практическое задание***

1. Выбрать задачу для исследования из следующих вариантов:

- разработать сайт пиццерии;
- разработать сайт по продаже билетов в кинотеатры;
- разработать сайт по продаже косметики;
- разработать сайт по продаже авиабилетов;
- разработать сайт по организации праздников;
- разработать сайт приюта для домашних животных;
- разработать мобильное приложение по поиску попутчиков для путешествий;
- разработать мобильное приложение для прокладки маршрутов по городу с использованием общественного транспорта;

- разработать мобильное приложение для отслеживания активности домашних животных во время отсутствия хозяина дома;

- предложить и выполнить свой вариант.

2. Выбрать трех человек для наблюдений.

3. Использовать фотоаппарат, чтобы задокументировать действия наблюдаемых.

4. Провести интервью (продолжительностью 10–15 мин) с каждым участником исследований относительно задачи, за решением которой вы наблюдали.

5. По результатам наблюдений выявить минимум 15 конкретных пользовательских потребностей.

6. Проанализировать сформулированные пользовательские потребности и выявить среди них базовые для решения исследуемой задачи и направленные на совершенствование пользовательского интерфейса для того, чтобы лучше справиться с задачей, за которой вы наблюдали.

#### *Содержание отчета*

1. Цель работы.

2. Задача для исследования.

3. Результаты наблюдений и интервьюирования трех человек, сопровождающиеся фотографиями.

4. Выявление 15 и более пользовательских потребностей.

5. Выводы по работе.

#### *Контрольные вопросы*

1. Какие способы используются для выявления пользовательских потребностей?

2. Назовите основные принципы контекстного исследования.

3. Поясните, каким образом технические средства (аудио- и видеозапись) помогают при наблюдении за пользователями.

## Тема 2

### Выработка стратегии решения проблем пользователей. Сторибоардинг

*Цель:* формирование у студентов знаний и навыков анализа результатов наблюдений, выработки идей создания новых пользовательских интерфейсов, построения сторибоарда.

#### *План занятия*

1. Изучить теоретические сведения.
2. Выполнить практическое задание по лабораторной работе.
3. Оформить отчет и ответить на контрольные вопросы.

#### *Теоретические сведения*

После получения результатов исследований пользовательской аудитории требуется сформировать точку зрения на решение проблем пользователей.

В данном случае *точка зрения* – это взгляд на стратегию решения проблемы до того, как начнется процесс разработки. Различные подходы к решению проблемы способствуют созданию хорошего дизайна.

Выявленные в результате наблюдения и интервьюирования пользовательские потребности составляют основу для выработки точки зрения. При этом умение взглянуть на проблему с разных сторон способствует разработке более качественных интерфейсов, поэтому рекомендуется найти две или более точки зрения на решение пользовательской проблемы.

Точка зрения иллюстрируется с помощью *сторибоарда*. Аналогом сторибоарда является раскадровка фильма. Сториборды сохраняют название и форму, как и в киноиндустрии, однако при проектировании интерфейсов выполняют другие задачи: формирование общего видения для всех членов рабочей группы и презентацию заказчику своих решений. Периодически сториборды выступают еще и как этап проектирования между формированием персонажей и созданием первого прототипа системы.

Сториборд должен отвечать на вопросы: кто ваш персонаж; какую потребность удовлетворяет система; какая задача должна быть выполнена; что приводит пользователя к использованию вашей системы; в каких условиях она используется; какова последовательность действий.

Спроектируем сториборд для программы под условным названием DarkSide, содержащей рецепты печенья. Сформулируем ответы на вопросы.

1. Персонаж: женщина среднего возраста.



2. Потребность: приготовление печенья.

3. Задача: подбор необходимых продуктов и изучение последовательности действий для готовки.

4. К использованию приложения пользователя приводит незнание рецепта.

5. Первая часть задачи выполняется либо в домашних условиях, либо, при нехватке продуктов, в магазине; вторая часть выполняется на кухне.

6. Последовательность: выбор конкретного рецепта, изучение списка необходимых продуктов, при необходимости приобретение недостающих продуктов, следование инструкции по приготовлению.

Все ответы зависят от формулировки исходной задачи. Если исходная потребность «приготовить печенье из того, что есть дома», то все последующие ответы будут другими.

Получившийся сторибоард представлен на рисунке 2.1.

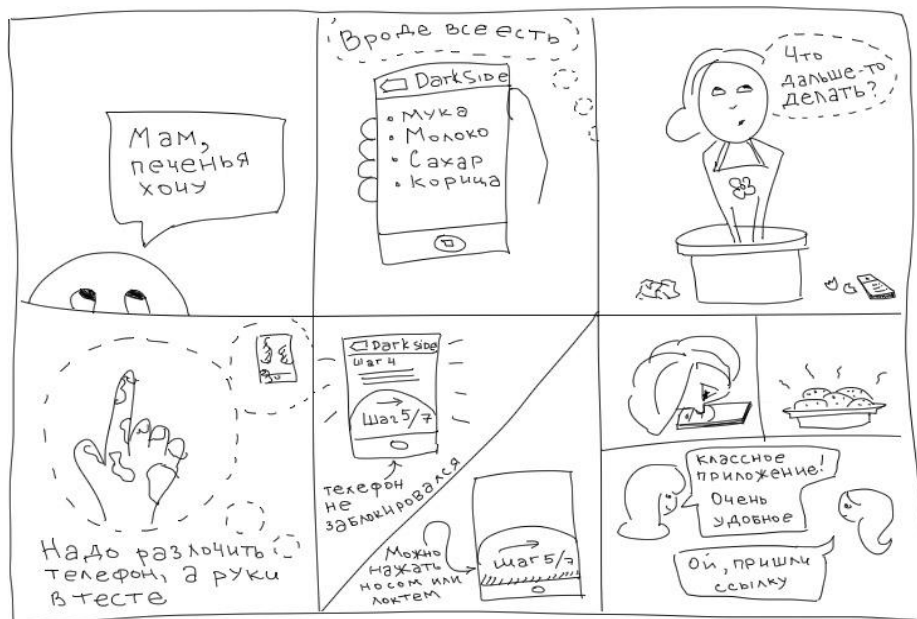


Рисунок 2.1 – Сторибоард приложения DarkSide

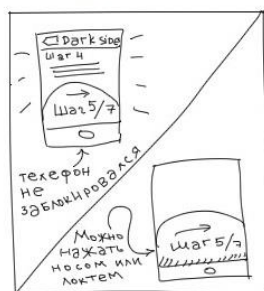
Одно приложение может включать несколько сценариев и, соответственно, может иллюстрироваться несколькими сторибоардами. Например, нарисуем сторибоард для DarkSide, чтобы показать его заказчику перед презентацией прототипа интерфейса. Для этого последний кадр обязательно должен показывать решение бизнес-задачи клиента, а не решение задачи пользователя, как на предыдущем кадре (рисунок 2.2).

Сторибоарды должны быть таким же быстрым инструментом, как и проектирование на бумаге, в связи с чем на технику создания сторибоарда не налагаются какие-либо требования.

## 1 постановка проблемы



## 2 решение



## 3 выгода



Рисунок 2.2 – Сюжетные этапы приложения DarkSide

### **Практическое задание**

1. Выработать точку зрения на решение проблемы пользователя.
2. Составить сторифоард, иллюстрирующий точку зрения.
3. Изобразить в сторифоарде пользователя приложения.
4. Отразить в сторифоарде, что приводит пользователя к использованию приложения.
5. Отразить в сторифоарде, какого результата достигает пользователь при использовании приложения.
6. Выработать альтернативную точку зрения.
7. Составить сторифоард, иллюстрирующий альтернативную точку зрения.
8. Провести сравнительный анализ сформулированных точек зрения и соответствующих им сторифоардов.
9. Выбрать один из двух вариантов точек зрения для дальнейшей реализации. Обосновать свой выбор.

### **Содержание отчета**

1. Цель работы.
2. Две точки зрения на решение проблемы пользователя.
3. Сторифоарды, иллюстрирующие точки зрения.
4. Выводы по работе.

### *Контрольные вопросы*

1. Что такое точка зрения в контексте процесса проектирования пользовательских интерфейсов?
2. Что такое сторибоард? Для каких целей он используется?
3. На какие вопросы должен отвечать сторибоард?

## **Тема 3**

### **Варфрейминг и прототипирование интерфейса пользователя**

*Цель:* формирование у студентов знаний и навыков построения прототипов пользовательского интерфейса.

#### *План занятия*

1. Изучить теоретические сведения.
2. Выполнить практическое задание по лабораторной работе.
3. Оформить отчет и ответить на контрольные вопросы.

#### *Теоретические сведения*

*Функциональные и информационные элементы пользовательского интерфейса* – это зримые представления функций и данных, доступные пользователю посредством интерфейса. Они являются проявлениями функциональных и информационных потребностей, выявленных на стадии выработки требований. Каждый элемент должен создаваться на основе конкретного требования, чтобы каждый аспект проектируемого продукта имел определенное назначение.

*Информационные элементы* представлены фундаментальными объектами интерактивных продуктов: фотографиями, сообщениями электронной почты, учетными карточками клиентов или заказов и др. Пользователи продукта могут работать с информационными элементами, реагировать и ссылаться на них. Иногда одни информационные объекты включают в себя другие (фотография в альбоме, песня в списке для проигрывания, отдельный счет внутри клиентской записи и т.д.).

*Функциональные элементы* – это операции, которые могут выполняться над информационными объектами и представляющими эти объекты элементами интерфейса. Функциональные элементы представляют собой инструменты, работающие с информационными элементами, а также контейнеры, содержащие информационные элементы.

Каждое выявленное требование имеет обычно несколько вариантов реализации посредством информационных и функциональных элементов. Важно определить, какое из возможных решений с наибольшей вероятностью позволит пользователям эффективно достигать целей, окажется в рамках бюджета и технологических возможностей. Для этого выполняется *варфрейминг (макетирование)* общей инфраструктуры взаимодействия.

Макетирование общей инфраструктуры взаимодействия предполагает разделение каждого представления на прямоугольные области, соответствующие панелям, элементам управления и другим высокоуровневым контейнерам. Рекомендуется делать несколько макетов, отражающих различные варианты расположения высокоуровневых контейнеров внутри интерфейса. Высокий уровень абстрактности, применяемый при макетировании, позволяет исследовать различные способы представления информации и функциональности, а при необходимости вносить радикальные изменения. Часто оказывается полезным попробовать несколько вариантов расположения областей, выполнить проверочные сценарии, а затем остановиться на лучшем решении.

Создание макета рекомендуется выполнять группой из графического дизайнера, промдизайнера и одного-двух проектировщиков взаимодействия. Простая белая доска – лучший инструмент для создания первоначальных набросков, который способствует совместной работе, обсуждениям и позволяет быстро стереть и нарисовать заново информацию.

Детализация пользовательского интерфейса выполняется посредством *прототипирования*. Прототипирование служит для визуального представления будущего пользовательского интерфейса и выполняется в специализированных программах моделирования (Axure, balsamiq) или на бумаге. Последний способ позволяет изложить свое видение интерфейса тем, кто не владеет или не имеет времени для освоения программных средств. Кроме того, бумажное прототипирование хорошо подходит для редактирования.

Для каждой ссылки и кнопки создается свой рисунок, чтобы подставить его в нужный момент на общий макет прототипа. Пример окна интерфейса с кнопками и возможностью демонстрации действия по клику на выпадающий список приведен на рисунке 3.1.

Чтобы «обновить» страницу «бумажного сайта», необходимо заменить элемент модели на другой. Пример сообщения об ошибке при попытке входа в систему по неверному логину или паролю приведен на рисунке 3.2.

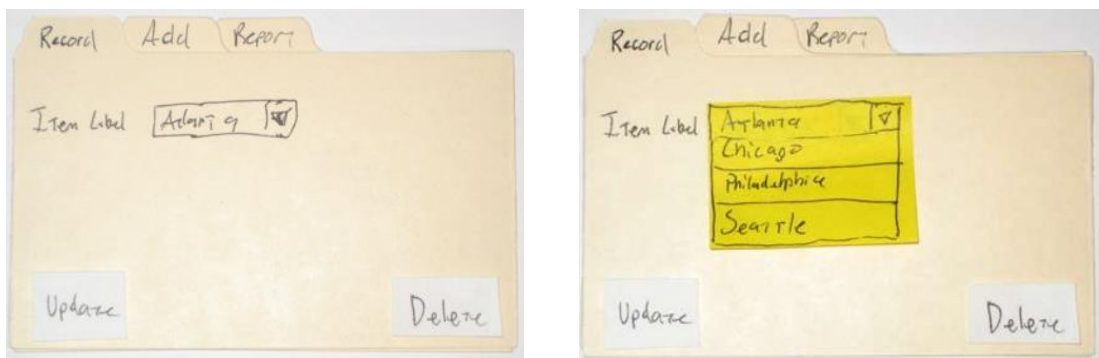


Рисунок 3.1 – Пример бумажных прототипов

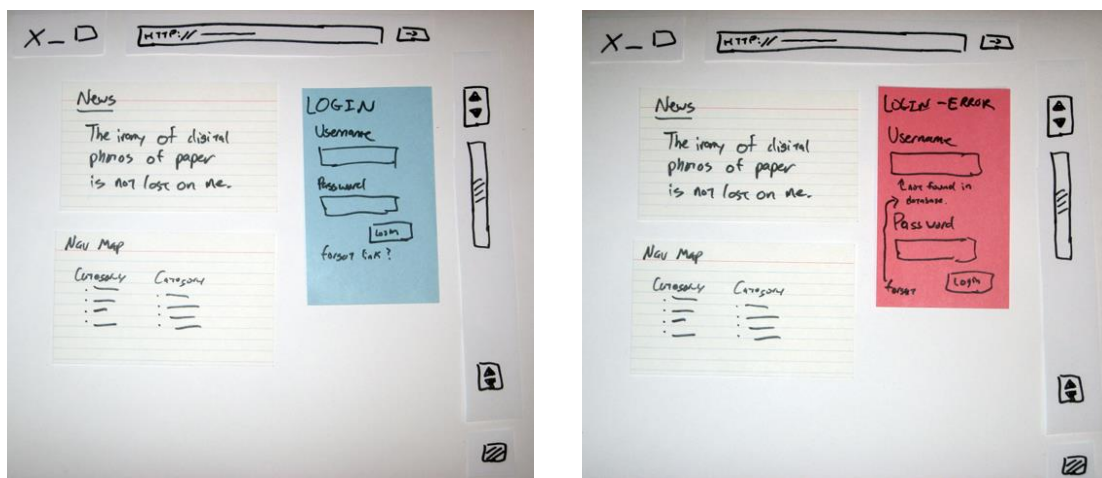


Рисунок 3.2 – Пример «обновления» страницы на бумажных прототипах

Однако на бумажных прототипах сложно воспроизвести процесс загрузки страницы, горизонтальную и вертикальную прокрутку, многообразие цветов, изображений, шрифтовых гарнитур и др. В этих случаях вместе с рисунками от руки можно использовать напечатанные на принтере изображения.

### ***Практическое задание***

1. Выполнить макетирование (варфрейминг) пользовательского интерфейса, отображающего выбранную точку зрения на решение проблемы.
2. Разработать три варианта расположения высокоуровневых функциональных и информационных элементов интерфейса на макете.
3. Выбрать и обосновать один вариант расположения высокоуровневых функциональных и информационных элементов интерфейса на макете.

4. В соответствии с макетом разработать бумажные прототипы пользовательского интерфейса, отображающие выбранную точку зрения на решение проблемы.

5. Предусмотреть «обновление» страниц бумажных прототипов.

6. Проанализировать полноту бумажных прототипов пользовательского интерфейса на предмет выполнения всех сформулированных по результатам наблюдений пользовательских потребностей.

#### *Содержание отчета*

1. Цель работы.
2. Разработанные макеты пользовательского интерфейса.
3. Бумажные прототипы пользовательского интерфейса.
4. Выводы по работе.

#### *Контрольные вопросы*

1. Что такое функциональные и информационные элементы интерфейса?
2. Зачем и как необходимо выполнять макетирование общей инфраструктуры взаимодействия пользователя с информационным продуктом?
3. Что такое прототипирование?
4. Каковы преимущества бумажных прототипов?

## **Тема 4**

### **Юзабилити-тестирование**

*Цель:* формирование у студентов знаний и навыков проведения юзабилити-тестирования.

#### *План занятия*

1. Изучить теоретические сведения.
2. Выполнить практическое задание по лабораторной работе.
3. Оформить отчет и ответить на контрольные вопросы.

#### *Теоретические сведения*

Юзабилити-тестирование применяется, чтобы узнать, насколько просто и удобно пользоваться разработанным программным обеспечением. Проводится данный вид тестирования циклически и включает в себя наблюдение за пользователями, установление проблем взаимодействия с программным продуктом и

устранение этих проблем. Тестирование продолжается до устранения всех значимых для пользователя проблем взаимодействия. Важно провести больше циклов тестирования, чем стремиться собрать всю возможную информацию из одного цикла. В каждом цикле тестирования должно участвовать не больше четырех пользователей.

Рекомендуется привлекать пользователей из целевой аудитории. Однако, если поиск идеального пользователя обусловит проведение меньшего количества тестов, следует отдать предпочтение исследованию среднестатистического пользователя как представителя всей аудитории (например, обычного пользователя Интернета).

Если предполагается, что сайт будет ориентирован на конкретную группу людей, то следует по возможности тестировать только людей из этой группы. Если целевая аудитория состоит из нескольких четко определенных групп с разными интересами и потребностями, необходимо протестировать пользователей из каждой группы. Например, если разрабатывается сайт университета, то по крайней мере в одном цикле тестирования должны участвовать два студента, два профессора, два аспиранта, два администратора. Но в следующих циклах уже могут участвовать любые представители целевой аудитории. Если для использования вашего сайта нужно обладать знаниями из какой-то специальной области (например, для сайта по валютным операциям, предназначенного для специалистов по финансам), то хотя бы для одного цикла тестирования следует привлечь людей из этой области.

Во время тестирования рекомендуется вести видеозапись исследования.

Можно использовать различные подходы в юзабилити-тестировании.

*Тестирование «Что здесь?».* Пользователям демонстрируют сайт и выясняют, понимают ли они, что видят: понятно ли назначение сайта, его «ценное предложение», структура, функционирование и т. д.

*Выполнение заданий.* Пользователя просят выполнить то или иное действие и затем наблюдают, насколько легко ему удастся это сделать.

После окончания наблюдений за пользователями следует: рассмотреть проблемы, с которыми они сталкивались; решить, какие из этих проблем должны быть исправлены; определить возможные решения для отобранных проблем. Особенность юзабилити-тестирования заключается в том, что данный вид тестирования позволяет посмотреть на свою работу чужими глазами и найти новые решения или применения старых идей. Следует отметить, что тестирование – это неоднократный процесс, поэтому можно применять различные решения в каждом цикле тестирования.

В рамках проведения юзабилити-тестирования следует соблюдать 10 основных *принципов юзабилити*, сформулированных Якобом Нильсеном.

1. *Информированность о состоянии системы.* Пользователь всегда должен понимать, что происходит в системе. Взаимодействие между пользователем и системой должно быть как можно более логичным и быстрым. Для этого целесообразно реализовать обратную связь в виде сообщений подтверждения успешности выполнения действий, запросов на подтверждение удаления, сообщений об ошибках и др.

2. *Схожесть системы с реальным миром.* Система должна общаться с пользователем на понятном ему языке. Использование слов, фраз и понятий, знакомых пользователю в реальном мире, намного предпочтительнее, чем использование специализированных терминов.

3. *Свобода действий.* Необходимо предоставить пользователям возможность отмены действий, а также возврата к ранее отмененным действиям.

4. *Единообразие и стандарты.* Не следует вводить в заблуждение пользователя, описывая одни и те же понятия разными терминами.

5. *Предотвращение ошибок.* Важно свести к минимуму количество условий, в которых могут быть допущены ошибки. Например, можно давать пользователям подсказки, поясняющие, какую информацию надо вводить в текстовые поля.

6. *На виду, а не в памяти.* Не следует вынуждать пользователя запоминать большое количество объектов, действий и опций. Посетитель не должен держать в голове информацию, перемещаясь из одной части системы в другую.

7. *Гибкость и эффективность.* Не следует нагружать пользователей лишней информацией, необходимо предоставить им возможность совершать часто повторяющиеся действия как можно быстрее и проще.

8. *Эстетичный и минималистичный дизайн.* Тексты не должны содержать бесполезной или устаревшей информации. Каждое лишнее слово мешает восприятию текста и лишает посетителя возможности достичь цели.

9. *Понимание проблем и их решение.* Сообщения об ошибках должны быть выражены на понятном пользователю языке, как можно более точно описывать проблему и предоставлять возможные варианты ее решения.

10. *Справочные материалы и документация.* Даже если система может использоваться без документации, в процессе работы с ней все же может потребоваться справочная информация. Подобные документы должны составляться таким образом, чтобы в них легко было найти необходимую информацию.



### ***Практическое задание***

1. Провести юзабилити-тестирование разработанных бумажных прототипов пользовательского интерфейса с привлечением двух человек.
2. Провести юзабилити-тестирование разработанных бумажных прототипов пользовательского интерфейса на предмет соответствия десяти принципам юзабилити Якоба Нильсена.
3. Выявить основные проблемы разработанного интерфейса по результатам выполнения пунктов 1, 2.
4. Устранить выявленные проблемы разработанного интерфейса.
5. Повторять пункты 1–4 до тех пор, пока не будут устранены все юзабилити-дефекты пользовательского интерфейса.

### ***Содержание отчета***

1. Цель работы.
2. Результаты юзабилити-тестирования бумажного прототипа пользовательского интерфейса.
3. Выводы по работе.

### ***Контрольные вопросы***

1. С какой целью проводится юзабилити-тестирование?
2. Каким образом проводится юзабилити-тестирование?
3. Кто участвует в юзабилити-тестировании?

## **Тема 5**

### **Знакомство с программой Adobe Photoshop**

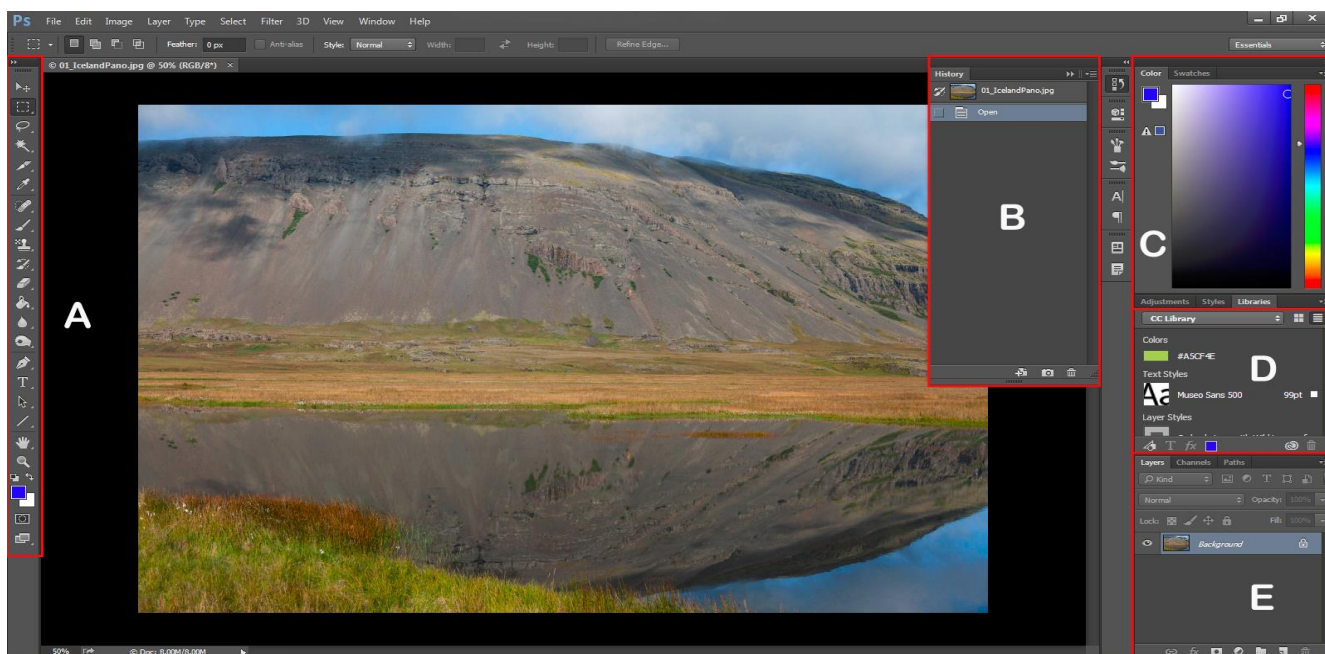
*Цель:* формирование у студентов знаний и практических навыков использования инструментов программы Adobe Photoshop.

### ***План занятия***

1. Изучить теоретические сведения.
2. Выполнить практическое задание по лабораторной работе.
3. Оформить отчет и ответить на контрольные вопросы.

### ***Теоретические сведения***

Внешний вид рабочей среды Adobe Photoshop приведен на рисунке 5.1.



*A* – панель «Инструменты»; *B* – панель «История»; *C* – панель «Цвет»;  
*D* – панель «Библиотеки Creative Cloud»; *E* – панель «Слои»  
 Рисунок 5.1 – Внешний вид рабочей среды Adobe Photoshop

Находящаяся в верхней части окна *панель приложения* содержит переключатель рабочих пространств/сред, меню и другие элементы управления приложением.

*Панель «Инструменты»* содержит инструменты для создания и редактирования изображений, графических объектов, элементов страниц и т. д. Связанные инструменты располагаются в группах.

На *панели «Управление»* в строке «Параметры» выводятся настройки выбранного в данный момент инструмента.

*Окно документа* отображает находящийся в работе файл. В окна документа можно поместить закладки, в некоторых случаях их можно группировать и закреплять.

*Палитры* упрощают процесс отслеживания и изменения обрабатываемого объекта. Палитры можно группировать, собирать в подборки или закреплять.

*Фрейм приложения* группирует все элементы рабочей среды в интегрированное окно, которое позволяет работать с приложением как с единым целым. При перемещении или изменении размера фрейма приложения или его элементов все элементы внутри него взаимодействуют друг с другом и не перекрываются. Па-

литры не исчезают при переключении приложений или при случайном щелчке за пределами приложения. При работе с несколькими приложениями можно расположить их рядом на экране или на нескольких мониторах.

### *Практическое задание*

1. В программной среде Adobe Photoshop создать кнопку в стиле скевоморфизм.

2. Разработать четыре состояния кнопки в стиле скевоморфизм: «Активная», «При наведении мышкой», «При клике мышкой», «Неактивная».

3. В программной среде Adobe Photoshop создать кнопку в стиле материал-дизайн.

4. Разработать четыре состояния кнопки в стиле материал-дизайн: «Активная», «При наведении мышкой», «При клике мышкой», «Неактивная».

5. В программной среде Adobe Photoshop создать кнопку в стиле минимализм.

6. Разработать четыре состояния кнопки в стиле минимализм: «Активная», «При наведении мышкой», «При клике мышкой», «Неактивная».

7. Провести сравнительный анализ кнопок в различных стилях.

8. Выявить преимущества и недостатки каждого стиля.

### *Содержание отчета*

1. Цель работы.

2. Результаты выполнения задания.

3. Выводы по работе.

### *Контрольные вопросы*

1. Опишите компоненты программы Adobe Photoshop.

2. Какие элементы содержит панель «Инструменты»?

3. Какие элементы содержит панель «Управление»?

4. Какие элементы содержит панель «Палитры»?

5. Охарактеризуйте современный стиль скевоморфизм.

6. Охарактеризуйте современный стиль материал-дизайн.

7. Охарактеризуйте современный стиль минимализм.

## Тема 6

### Основы композиции элементов веб-приложения

*Цель:* формирование у студентов знаний и практических навыков работы с композицией элементов веб-приложения.

#### *План занятия*

1. Изучить теоретические сведения.
2. Выполнить практическое задание по лабораторной работе.
3. Оформить отчет и ответить на контрольные вопросы.

#### *Теоретические сведения*

*Точка, линия и прямоугольник* – примитивы верстки, которые составляют геометрию веб-страницы.

Идеальному *точечному объекту* соответствует круг, который имеет собственный центр и мгновенно привлекает внимание. Однако его стоит использовать ограниченно (пример злоупотребления кругами приведен на рисунке 6.1).

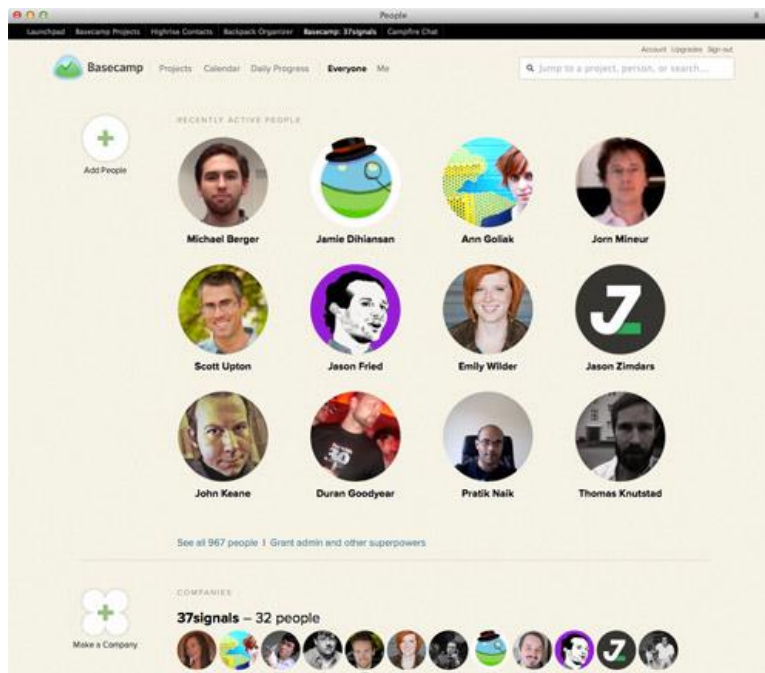


Рисунок 6.1 – Пример злоупотребления кругами как геометрическими примитивами

В верстке применяются различные точечные объекты: отдельные буквы и цифры, знаки, логотипы, пиктограммы. Они имеют разнообразную форму, но, в отличие от линий и прямоугольников, «центростремительны». Точечный объект имеет один фокус внимания и создает на странице акцент.

Вырезанные из фона изображения не имеют четкой прямоугольной формы и также привлекают внимание. На рисунке 6.2 роль точек играют крупные номера страниц в левой колонке, кружок-фактоид справа, фотографии на белом фоне. Но даже на прямоугольных фотографиях лица и другие важные объекты выступают в качестве самостоятельных силовых точек. Обратите внимание на лицо продавца, курицу и знак на фотографии рынка – по сути, они полноценно участвуют в верстке.

## В номере



Пашающиеся живые курыца на рынке в Марокко

54

ТАЖИКИ

### Морок и дым

ТАЖИКИ, ХАРИРА И НОС-НОС  
У СЕБЯ НА РОДИНЕ

8

УТНИКИ

### Утние истории

ЧТО ДЕЛАЮТ С ПТИЦЕЙ  
НА КРУПНЕЙШЕЙ УТНИНОЙ  
ФАБРИКЕ В СТРАНЕ

38

ПЕТЕРБУРЖ

### Вдаль, по-питерски

БУМ НЕПРОФЕССИОНАЛЬНЫХ  
ВАРОВ И КАФЕ В ПЕТЕРБУРГЕ

#### Письмо редактора 6

#### Дом

- ИЗДАТЕЛЬСТВО Утние истории 8
- ИЗДАТЕЛЬСТВО Слайсер 18
- ИЗДАТЕЛЬСТВО Баранина семья Сиротинных 20
- ИЗДАТЕЛЬСТВО Пасха 22
- ИЗДАТЕЛЬСТВО Холодильник: Николай Борисова 24
- ИЗДАТЕЛЬСТВО Пьемонт Анатолия Корнеева 26
- ИЗДАТЕЛЬСТВО Лби 28
- ИЗДАТЕЛЬСТВО Кухня Аллы Белкиной 30
- ИЗДАТЕЛЬСТВО Не от жира сего 32
- ИЗДАТЕЛЬСТВО Паско-сауэр 36

#### Рестораны

- ИЗДАТЕЛЬСТВО Вдаль, по-питерски 38
- ИЗДАТЕЛЬСТВО Китайская грамота 44
- ИЗДАТЕЛЬСТВО Ольга Захарова 46
- ИЗДАТЕЛЬСТВО Jerome & Patrice 50
- ИЗДАТЕЛЬСТВО Чужой вкусок 52

#### Путешествия

- ИЗДАТЕЛЬСТВО Морок и дым 54
- ИЗДАТЕЛЬСТВО Marchal 66
- ИЗДАТЕЛЬСТВО Рай или больше 68
- ИЗДАТЕЛЬСТВО Париж 70

#### Кухня

- Весенние обеды 80
- Масляная рыба Пако Моралеса 90
- Мультинарки 94
- Азиатско-русские закуски 99
- Как нарезать овощи 106
- Пельби 110
- Соевые соусы 114
- Пасхальный обед 120
- Адреса 126

Фокус: Как сделать баншио Татина из шоколада 125



Эта машина только для того, чтобы нарезать тонкие ветчины



Все великие и красивые Пасхи, выложенные в одном эскизе



Лам, дополнивший ежу, с чем вступит в контакт

Рисунок 6.2 – Использование точечных объектов разнообразной формы

Заметность объекта-точки зависит от его размера, плотности, контраста к фону, окружающего свободного пространства, а также измеряется относительно соседних элементов.

В геометрии точки составляют прямую линию. Линия в типографике – это строка, второй важнейший примитив.

*Строка (линия)* – самый удобный, наглядный, лаконичный элемент верстки (рисунок 6.3), один из способов вместить и срочное объявление, и временный блок на сайте.

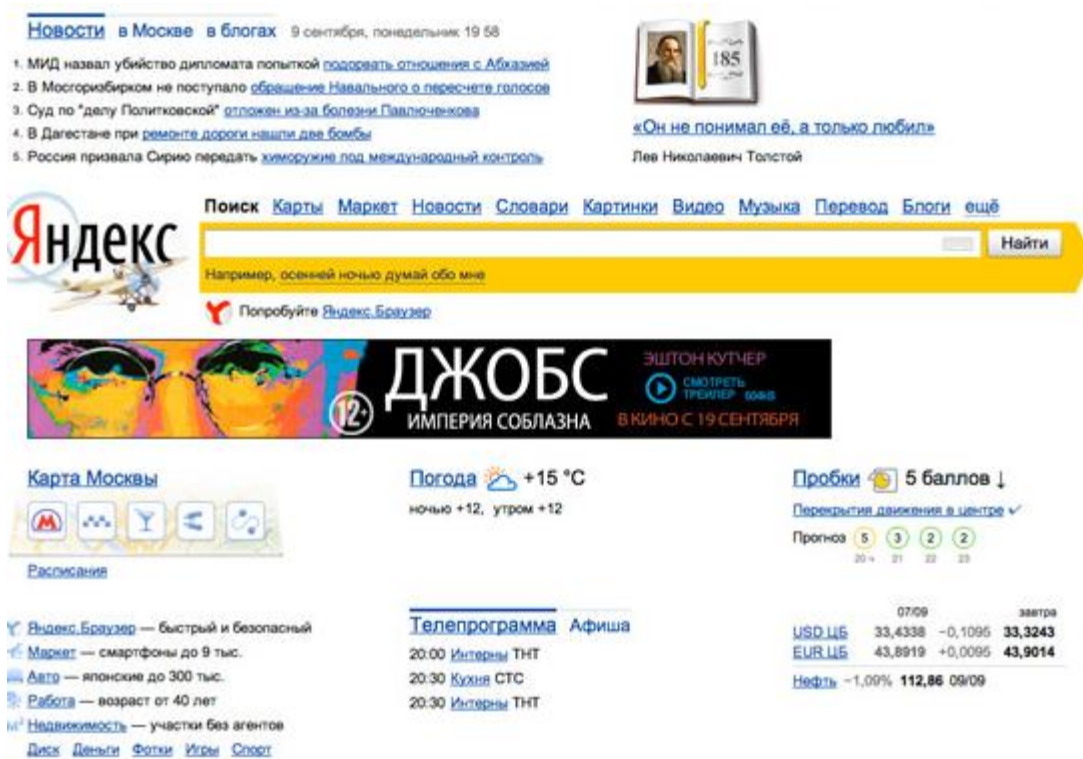


Рисунок 6.3 – Строчное представление информации на странице

Лучше всего размещать строки в привязке к верхней или нижней частям модуля (примеры – «шапка» и «подвал» сайта в форме строк). Строка отлично подходит для организации меню и панелей инструментов (рисунок 6.4).

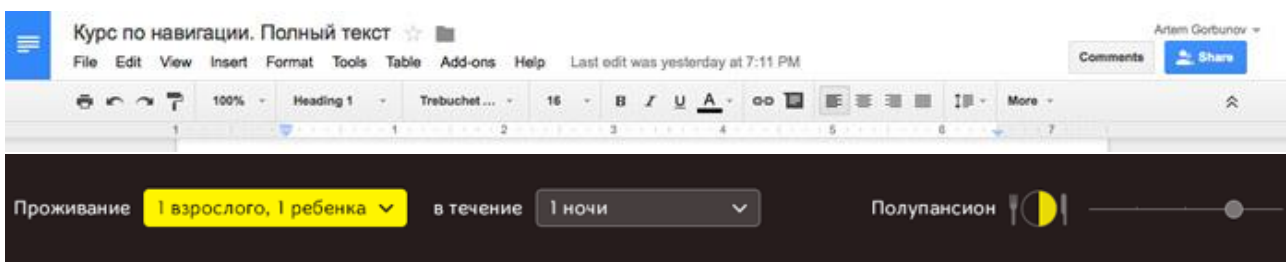


Рисунок 6.4 – Организация меню и панелей инструментов в виде строк

*Прямоугольник* – третий примитив верстки, самая технологичная фигура, способная делиться на другие прямоугольники. Информационные носители – экран, бумага, рекламный щит – имеют прямоугольную форму.

Все модули макета должны иметь прямоугольную форму и составлять вместе общий прямоугольник страницы или экрана, которым они ограничены. Простейшие примеры прямоугольного модуля: абзац текста, иллюстрация или элемент интерфейса. Но прямоугольник – это универсальный контейнер для любых объектов (точечных, строчных и прямоугольных). Модульная верстка – это поиск наиболее удачной компоновки прямоугольника.

### ***Практическое задание***

1. Создать рекламный баннер для сайта размером 500×400 пикселей.
2. На баннере разместить изображение, выданное преподавателем.
3. Удалить фон с полученного изображения векторной маской.
4. На баннере разместить текст двух уровней важности: заголовок и описание.
5. Добавить на баннер действие.
6. Создать яркий фон баннера без использования изображений.

### ***Содержание отчета***

1. Цель работы.
2. Рекламный баннер.
3. Выводы по работе.

### ***Контрольные вопросы***

1. Перечислите основные примитивы верстки.
2. Охарактеризуйте примитив *точка*.
3. Охарактеризуйте примитив *линия*.
4. Охарактеризуйте примитив *прямоугольник*.
5. В чем сущность модульной верстки?

## Тема 7

### Создание графических элементов веб-приложения

*Цель:* формирование у студентов знаний и практических навыков подготовки графики для веб-приложений посредством Adobe Photoshop.

#### *План занятия*

1. Изучить теоретические сведения.
2. Выполнить практическое задание по лабораторной работе.
3. Оформить отчет и ответить на контрольные вопросы.

#### *Теоретические сведения*

*Стили слоев* позволяют быстро применять эффекты ко всему слою. На панели «Эффекты» находятся разнообразные стили слоя, которые можно применить одним щелчком мыши. Границы эффекта автоматически обновляются при редактировании слоя. Например, при применении тени к текстовому слою тень автоматически меняется при редактировании текста.

Стили слоя имеют накопительный характер, что означает возможность создания комплексного эффекта при использовании нескольких стилей к слою.

Когда к слою применяется стиль, справа от имени слоя на панели «Слои» появляется значок стиля. Эффекты слоя привязываются к его содержимому. При перемещении или редактировании содержимого слоя эффекты также меняются соответствующим образом. После выполнения команды «Слой» → «Стиль слоя» → «Настройки стиля» можно отредактировать настройки стиля слоя, применить другой стиль или изменить настройки стиля или атрибутов, доступных в диалоговом окне.

*Угол освещения* позволяет задать угол, под которым освещение падает на слой. *Тень* позволяет задать расстояние тени от объекта слоя. Непрозрачность и толщину тени можно менять с помощью ползунков. *Внешнее свечение* позволяет указать толщину свечения, появляющегося от внешних краев объекта слоя. *Внутреннее свечение* позволяет указать толщину свечения, появляющегося из внутренних краев объекта слоя. Непрозрачность как внешнего, так и внутреннего свечения можно менять с помощью ползунка. Имеется также возможность указать толщину фаски внутренних краев объекта слоя, направление фаски (вверх или вниз), толщину обводки, непрозрачность обводки.

Чтобы применить стиль слоя, необходимо выполнить следующие действия (рисунок 7.1).

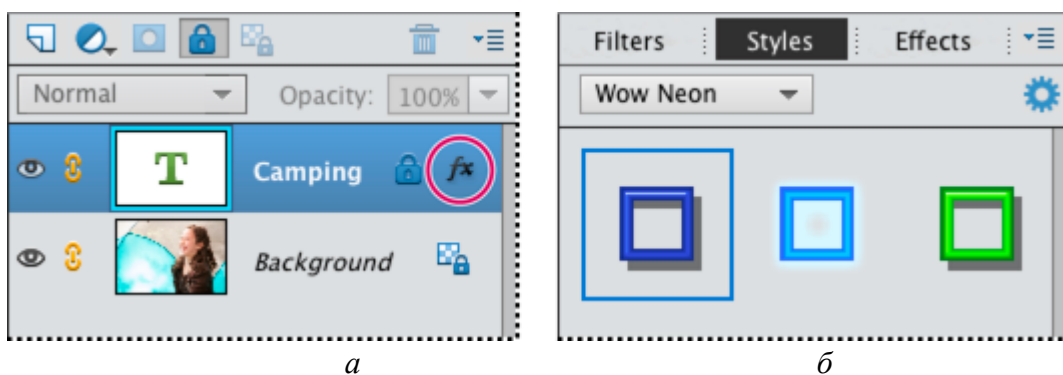
1. Выберите слой на панели «Слои».



2. На панели «Эффекты» выберите команду «Стили слоя» в меню категорий.
3. Выполните одно из следующих действий:
  - выберите стиль и нажмите кнопку «Применить»;
  - выполните двойной щелчок мышью на стиле;
  - перетащите стиль на слой.

Значок стиля указывает на то, что к слою применен стиль.

Чтобы отменить полученный результат, воспользуйтесь комбинацией клавиш Ctrl+Z или выполните команду «Редактирование» → «Отменить».



*a* – выбираем слой на панели «Слои»;  
*б* – выбираем стиль на вкладке «Стили слоя»  
Рисунок 7.1 – Применение стиля к слою

Настройки стиля можно редактировать, копировать с одного слоя на другой, а также можно удалять стиль со слоя. Чтобы отредактировать стиль слоя, необходимо выполнить следующие действия.

1. Выполните команду «Слой» → «Стиль слоя» → «Настройки стиля».
2. Для оперативного просмотра изменений изображения установите флажок «Предварительный просмотр».
3. Отрегулируйте настройки стиля одного или нескольких слоев или добавьте новые и нажмите «ОК».

Изменить масштаб стиля слоя можно, выполнив следующие действия.

1. На панели «Слои» выделите слой, содержащий эффект, который требуется масштабировать.
2. Выполните команду «Слой» → «Стиль слоя» → «Эффект масштабирования».
3. Для оперативного просмотра изменений изображения установите флажок «Предварительный просмотр».

4. Укажите нужный масштаб эффекта. Например, при увеличении размера внешнего свечения 100 % – текущий масштаб, в то время как масштаб 200 % увеличивает размер свечения вдвое. Затем нажмите кнопку «ОК».

Копирование настроек стиля одного слоя в другой осуществляется следующим образом.

1. На панели «Слои» выделите слой с настройками стиля, которые требуется скопировать.

2. Выполните команду «Слой» → «Стиль слоя» → «Скопировать стиль слоя».

3. Выберите нужный слой на панели «Слои» и выполните команду «Слой» → «Стиль слоя» → «Вставить стиль слоя».

Удаление стиля слоя осуществляется следующим образом.

1. На панели «Слои» выделите слой, содержащий удаляемые стили.

2. Выполните команду «Слой» → «Стиль слоя» → «Очистить стиль слоя».

### ***Практическое задание***

1. Создать в программной среде Adobe Photoshop макет формы регистрации.

2. На макете формы регистрации разместить форму ввода информации.

3. Предусмотреть сообщения об ошибках в рамках валидации полей формы ввода.

4. Предусмотреть хинты (всплывающие подсказки для пользователя) для полей формы ввода.

5. На макете формы регистрации разместить кнопки действия.

6. На макете формы регистрации разместить иконки.

7. На макете формы регистрации разместить текст.

### ***Содержание отчета***

1. Цель работы.

2. Макет формы регистрации.

3. Выводы по работе.

### ***Контрольные вопросы***

1. Как применить стиль слоя?

2. Как изменить масштаб стиля слоя?

3. Как выполнить копирование настроек стиля одного слоя в другой?

4. Как удалить стиль слоя?

## Тема 8

### Работа с текстом в Adobe Photoshop

*Цель:* формирование у студентов знаний и практических навыков работы с текстом в программе Adobe Photoshop.

#### *План занятия*

1. Изучить теоретические сведения.
2. Выполнить практическое задание по лабораторной работе.
3. Оформить отчет и ответить на контрольные вопросы.

#### *Теоретические сведения*

##### **Редактирование текста.**

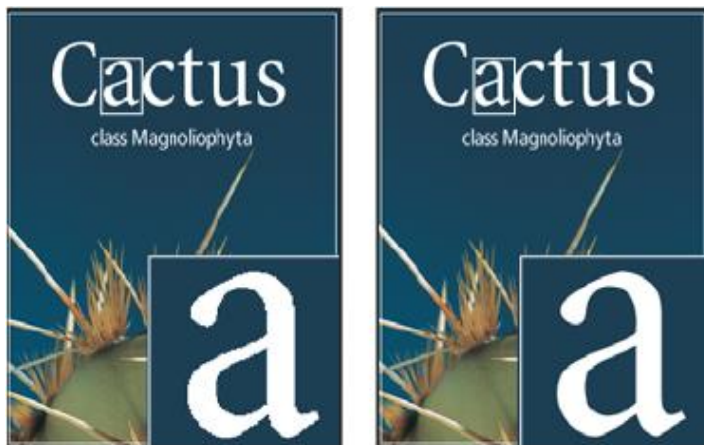
1. Выберите инструмент «Горизонтальный текст» (Т) или «Вертикальный текст» (↓Т).
2. Выберите текстовый слой на панели «Слои» или щелкните текст на изображении для автоматического выбора текстового слоя.
3. Поместите курсор вставки в текст и выполните одно из следующих действий:
  - щелкните, чтобы установить курсор вставки;
  - выделите один или несколько символов, которые необходимо редактировать.
4. Введите нужный текст.
5. На панели параметров выполните одно из следующих действий:
  - нажмите кнопку «Сохранить» (✓), чтобы применить изменения к слою текста;
  - нажмите кнопку «Отмена» (⊖) или клавишу «Esc».

##### **Применение сглаживания к текстовому слою.**

Сглаживание позволяет получить символы текста с гладкой кромкой, реализуя частичное заполнение краевых пикселей. В результате этого края текста смешиваются с фоном (рисунок 8.1).

При подготовке текста, который будет отображаться на веб-странице, следует учесть, что сглаживание значительно увеличивает количество цветов в исходном изображении. Это ограничивает возможности по снижению размера файла изображения. Сглаживание может также вызвать появление на краях символов текста случайных цветов. Когда важно снизить размер файла и уменьшить количество цветов, предпочтительнее отказаться от сглаживания, несмотря на то, что в изображении останутся зубчатые края. Кроме того, следует рассмотреть возможность использования шрифта большего размера, чем при выводе на печать, по-

сколькx такой шрифт легче читается и по нему проще решить, следует ли применить сглаживание.




*a* – сглаживание установлено в значение «Нет»;

*б* – сглаживание установлено в значение «Насыщенное»

Рисунок 8.1 – Применение сглаживания к текстовому слою

Если использовать сглаживание при малых размерах и низкой разрешающей способности (например, в веб-изображениях), текст может показаться неразборчивым. Чтобы повысить разборчивость, сбросьте флажок «Дробная ширина» в меню панели «Символ».

1. Выберите текстовый слой на панели «Слои».
2. Выполните одно из следующих действий:
  - установите параметр в меню сглаживания  на панели параметров или на панели «Символ»;
  - выберите меню «Слой» → «Текст», а затем выберите параметр во вложенном меню: «Без сглаживания» (сглаживание не применяется); «Жесткое» (текст выводится с максимальной резкостью); «Четкое» (текст выводится с большой резкостью); «Насыщенное» (текст получается более отчетливым); «Плавное» (текст получается более плавным).

### **Проверка и исправление орфографии.**

Во время проверки орфографии в документе Adobe Photoshop запрашивает пользователя обо всех словах, отсутствующих в словаре программы. Если слово написано правильно, можно подтвердить правильность его написания, добавив это слово в пользовательский словарь. Если слово написано неверно, можно исправить его.

При необходимости проверки и исправления орфографии выполните следующие действия.

1. Выберите язык из ниспадающего меню в нижней части панели «Символ» – это словарь, которым Photoshop пользуется для проверки орфографии.

2. Отобразите или разблокируйте текстовые слои. Проверка орфографии не выполняется в скрытых и заблокированных слоях.

3. Выполните одно из следующих действий:

- выберите текстовый слой;
- для проверки фрагмента текста выделите его;
- для проверки слова поместите курсор вставки внутрь этого слова.

4. Выберите меню «Редактирование» → «Проверка орфографии».

5. Если необходимо проверить орфографию только в выбранном текстовом слое, отмените выделение параметра «Проверить все слои».

6. Если Adobe Photoshop нашел незнакомые слова или другие возможные ошибки, нажмите одну из следующих кнопок:

- «Пропустить» (проверка продолжается, не изменяя текст);
- «Пропустить все» (это слово будет пропущено при дальнейшей проверке);
- «Заменить» (ошибка исправляется; убедитесь, что в поле «Заменить на» слово написано правильно, а затем нажмите кнопку «Заменить»; если предложено не то слово, выберите другое в поле «Варианты» или введите правильное слово в поле «Заменить на»);
- «Заменить все» (все случаи неправильного написания этого слова в документе будут исправлены; убедитесь, что в поле «Заменить на» слово написано правильно);
- «Добавить» (неопознанное слово сохраняется в словаре, чтобы в дальнейшем появление этого слова не воспринималось как ошибка).

### **Масштабирование и вращение текста.**

Можно задать пропорцию между высотой и шириной текста относительно исходной ширины и высоты символов. Для немасштабированных символов эта величина имеет значение 100 %. Некоторые гарнитурки включают в себя настоящий широкий шрифт, символы которого специально разработаны с большей шириной, чем в обычном шрифте. Масштабирование приводит к визуальному искажению символов текста, поэтому обычно предпочтительнее использовать шрифты, специально разработанные как узкие или широкие, если они доступны.

Для масштабирования текста выполните следующие действия.

1. Выделите символы или текстовые объекты, которые необходимо изменить. Если текст не выделен, масштабирование применяется к вновь создаваемому тексту.

2. В палитре «Символ» задайте значения для параметра «Масштаб по вертикали» (**I**) или «Масштаб по горизонтали» (**T**).

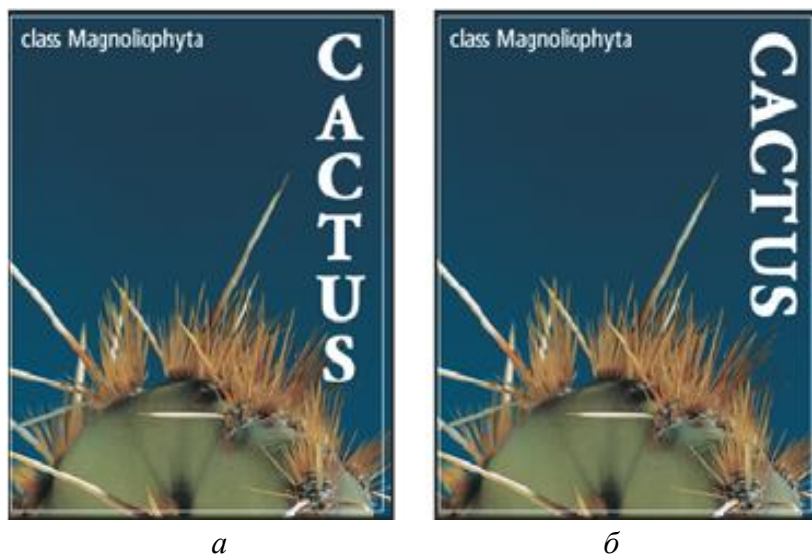
Для поворота текста выберите текстовый слой и воспользуйтесь любой командой вращения или командой «Свободное трансформирование». Для типа абзаца можно также выделить ограничительную рамку и воспользоваться маркером для вращения текста вручную.

При работе с вертикальным текстом символы можно поворачивать с шагом 90°. Повернутые символы отображаются вертикально, а неповернутые – перпендикулярно линии текста (рисунок 8.2).

Поворот символов вертикального текста осуществляется следующим образом.

1. Выберите вертикальный текст, который необходимо поворачивать (или отменить поворот).

2. Выберите в меню панели «Символ» пункт «Стандартная вертикальная ориентация». Флажок показывает, что параметр выбран.




*a* – исходный текст; *б* – текст с поворотом

Рисунок 8.2 – Поворот символов вертикального текста

Ориентация текстового слоя определяет направление линий текста относительно окна документа (для начала набора) или ограничительной рамки (для типа абзаца). Если текстовый слой ориентирован вертикально, текст выводится сверху вниз, если горизонтально – слева направо. Не следует путать ориентацию текстового слоя с направлением символов в строке текста.



Изменение ориентации текстового слоя выполняется следующим образом.

1. Выберите текстовый слой на панели «Слои».
2. Выполните одно из следующих действий:
  - выберите инструмент «Текст» и нажмите кнопку «Ориентация текста» () на панели параметров;
  - выберите меню «Слой» → «Текст» → «Горизонтальный» или «Слой» → «Текст» → «Вертикальный»;
  - выберите пункт «Изменить ориентацию текста» в меню панели «Символ».

### Форматирование абзацев.

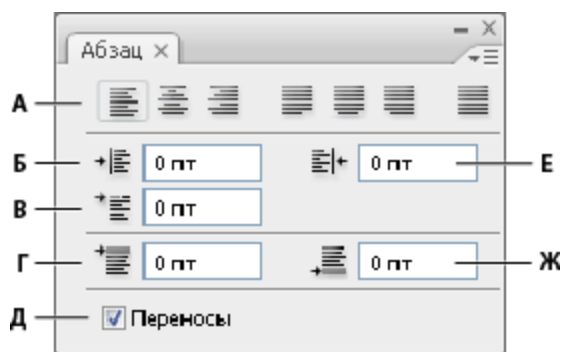
В начале набора каждая строка представляет собой отдельный абзац. В типе абзаца каждый абзац может состоять из нескольких строк в зависимости от размеров ограничительной рамки.

Для настройки параметров форматирования одного или нескольких абзацев (или для всех абзацев слоя) можно выделить их и воспользоваться панелью «Абзац».

1. Выберите инструмент «Горизонтальный текст» () или «Вертикальный текст» ()


2. Для форматирования одного абзаца щелкните его. Чтобы форматировать несколько абзацев, выделите диапазон абзацев. Форматировать все абзацы слоя можно, выделив текстовый слой на панели «Слой».

Панель «Абзац» используется для форматирования абзацев и колонок (рисунок 8.3).



- А* – выравнивание и выключка; *Б* – добавить отступ слева;  
*В* – добавить отступ слева в первой строке; *Г* – добавить отступ перед абзацем;  
*Д* – перенос; *Е* – добавить отступ справа; *Ж* – добавить отступ после абзаца

Рисунок 8.3 – Панель «Абзац»

Для отображения панели выберите меню «Окно» → «Абзац» или щелкните ярлык панели «Абзац», если она видна, но не активна. Можно также выбрать инструмент «Текст» и нажать кнопку «Панель» () на панели параметров.

Задать числовые значения параметров на панели «Абзац» можно стрелками *вверх* и *вниз* или введением значения непосредственно в текстовом поле. После изменения значения выполните одно из следующих действий:

- нажмите клавишу «Ввод» («Return»), чтобы сохранить значение;
- используйте комбинацию клавиш «Shift» + «Ввод» («Shift» + «Return»), чтобы сохранить и выделить его;
- нажмите клавишу «Tab», чтобы сохранить значение и перейти к следующему полю панели.

В меню панели «Абзац» можно получить доступ к дополнительным командам и параметрам. Чтобы воспользоваться этим меню, щелкните треугольник в правом верхнем углу панели.

### **Выравнивание текста.**

Производится выравнивание текста к одному из краев абзаца (для горизонтального текста – влево, по центру или вправо, для вертикального – вверх, по центру или вниз). Параметры выравнивания доступны только для типа абзаца.

Выполните одно из следующих действий.

1. Выделите текстовый слой, если надо изменить все абзацы в этом слое, или выделите только те абзацы, которые необходимо изменить.

2. На панели «Абзац» или на панели параметров щелкните параметр выравнивания. Для горизонтального текста это следующие параметры:

- *выключка текста влево*. Выравнивает текст влево, оставляя правый край абзаца неровным;
- *выключка текста по центру*. Выравнивает текст по центру, оставляя оба края абзаца неровными;
- *выключка текста вправо*. Выравнивает текст вправо, оставляя левый край абзаца неровным.

Для вертикального текста параметры выравнивания следующие:

- *выключка текста вверх*. Выравнивает текст по верхнему краю, оставляя нижний край абзаца неровным;
- *выключка текста по центру*. Выравнивает текст по центру, оставляя низ и верх абзаца неровными;
- *выключка текста по нижнему краю*. Выравнивает текст по нижнему краю, оставляя верхний край абзаца неровным.

### **Отступы абзацев.**

Отступ задает величину интервала между текстом и ограничительной рамкой или линией, на которой размещается текст. Отступы действуют только для



выделенных абзацев, поэтому для разных абзацев могут быть заданы разные отступы.

1. Выделите текстовый слой, если надо изменить все абзацы в этом слое, или выделите только те абзацы, которые необходимо изменить.

2. На панели «Абзац» введите значения параметров отступа.

- *Отступ от левого края.* Величина отступа от левого края абзаца. Для вертикального текста этот параметр управляет отступом от верхнего края абзаца.

- *Отступ от правого края.* Величина отступа от правого края абзаца. Для вертикального текста этот параметр управляет отступом от нижнего края абзаца.

- *Отступ первой строки.* Величина отступа первой строки текста абзаца. Для горизонтального текста отступ первой строки определяется относительно отступа от левого края, для вертикального текста – относительно отступа от верхнего края. Чтобы создать висячий отступ (выступ), введите отрицательное значение.

### **Настройка интервалов между абзацами.**

1. Выделите абзацы, которые необходимо изменить, или весь текстовый слой при необходимости изменить все абзацы этого слоя. Если в абзац не установлен курсор и не выбран текстовый слой, настройка будет применяться к новому создаваемому тексту.

2. На панели «Абзац» настройте значения параметров «Добавить отступ перед абзацем» (☰) и «Добавить отступ после абзаца» (☷).

### ***Практическое задание***

1. Сверстать текстовую статью, содержащую заголовки пяти уровней (h1–h5) и изображения.

2. Выполнить проверку орфографии в статье.

3. Применить масштабирование и вращение текста.

4. Выполнить изменение ориентации текстового слоя.

5. Выполнить форматирование абзацев.

6. Выполнить выравнивание текста.

7. Задать отступы и интервалы между абзацами.

### ***Содержание отчета***

1. Цель работы.

2. Сверстанная текстовая статья.

3. Выводы по работе.

### ***Контрольные вопросы***

1. Как выполняется редактирование текста?

2. Как выполняется сглаживание к текстовому слою?

3. Как выполняется масштабирование и вращение текста?

4. Как выполняется поворот текста?
5. Как выполняется изменение ориентации текстового слоя?
6. Как выполняется форматирование абзацев?
7. Как выполняется выравнивание текста?
8. Как указываются отступы абзацев?
9. Как выполняется настройка интервалов между абзацами?

## Тема 9

### Создание графического макета веб-приложения

*Цель:* формирование у студентов знаний и навыков создания графических макетов страниц веб-приложения.

#### *План занятия*

1. Изучить теоретические сведения.
2. Выполнить практическое задание по лабораторной работе.
3. Оформить отчет и ответить на контрольные вопросы.

#### *Теоретические сведения*

Рассмотрим принципы работы с вертикальными отступами.

Шапка страницы (положение заголовка, подзаголовка и отступы вокруг них) приведена на рисунке 9.1.

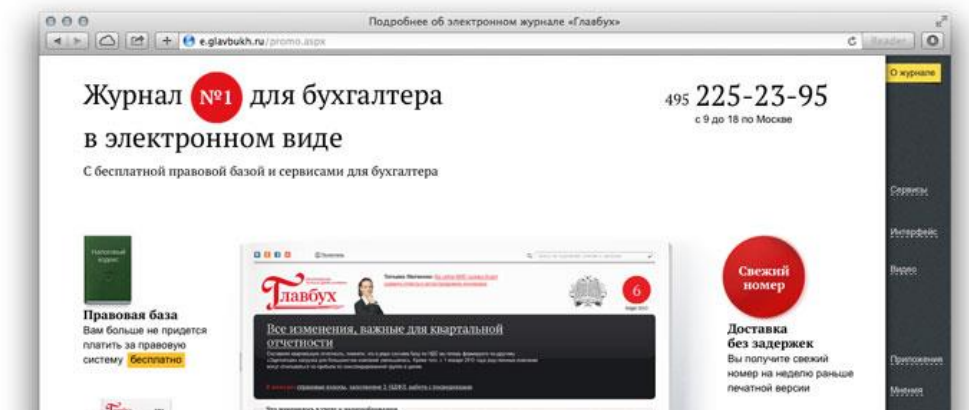


Рисунок 9.1 – Пример некорректной установки вертикальных отступов заголовка веб-страницы

Макет на первый взгляд выглядит аккуратно. Однако с профессиональной точки зрения требуют корректировки верхнее поле и расстояния между строками в соответствии с правилами соотношения внутренних и внешних отступов.

Типографический объект любого уровня состоит из других объектов: буквы из штрихов, слова из букв, строки из слов, абзацы из строк. Для того чтобы объект имел самостоятельное значение, то есть буква, слово, строка и абзац отделялись от соседних, его внутренние расстояния должны быть меньше внешних.

Рассмотрим однострочный заголовок (рисунок 9.2): верхнее поле («внешнее» для заголовка) даже с учетом круга значительно шире, чем пробелы между словами («внутреннее» для заголовка).

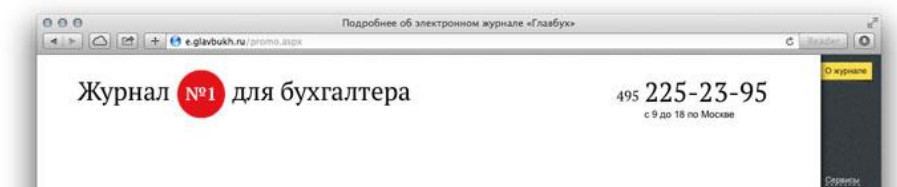


Рисунок 9.2 – Однострочный заголовок

Добавим вторую строку (рисунок 9.3). Верхнее поле («внешнее» для заголовка) по ширине стало визуально сравнимо с межстрочным просветом («внутреннее» для заголовка), и заголовок утратил значимость.

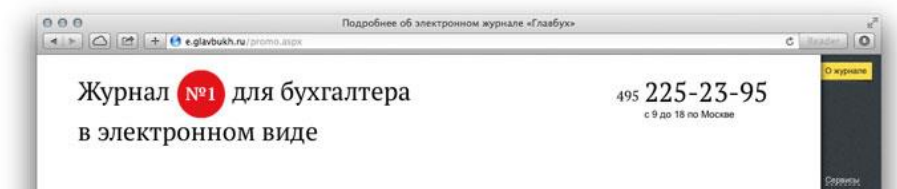


Рисунок 9.3 – Двустрочный заголовок

Увеличим внешнее поле (рисунок 9.4), при этом достаточно, чтобы внешний отступ был больше внутреннего. Таким образом типографический объект получит больше самостоятельности и значимости.

Добавим подзаголовок (рисунок 9.5), он кажется более близким ко второй строке заголовка, чем собственные строки заголовка друг к другу. В результате заголовок вновь теряет самостоятельность и независимость.

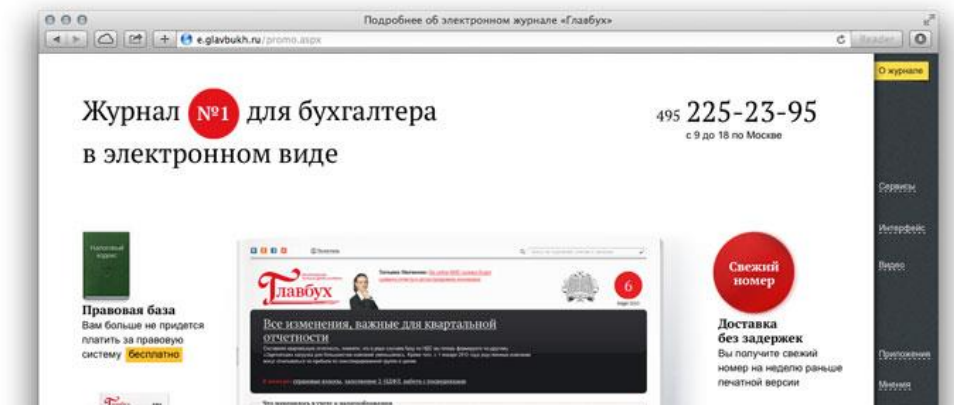


Рисунок 9.4 – Двустрочный заголовок с увеличенным внешним отступом



Рисунок 9.5 – Двустрочный заголовок и подзаголовок

Аналогичная ситуация может сложиться при добавлении иллюстрации с неправильным вертикальным отступом по отношению к заголовку (рисунок 9.6).

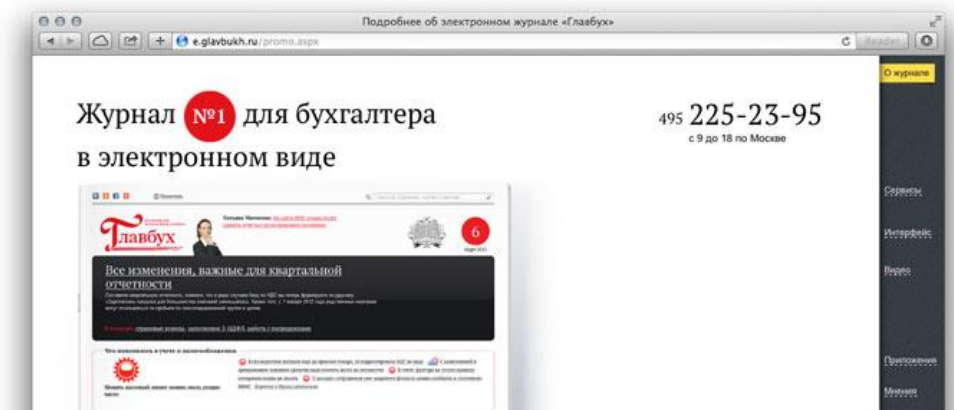


Рисунок 9.6 – Двустрочный заголовок и иллюстрация с некорректным внешним отступом

Отодвинем подзаголовок на минимальное расстояние, равное межстрочному просвету заголовка. Этого достаточно, чтобы заголовок и подзаголовок работали как один объект (рисунок 9.7).

Получился общий модуль заголовка с подзаголовком, у которого собственные отношения внутреннего и внешнего отступов. По правилу «внутреннего» и «внешнего» расстояние до иллюстрации должно быть больше, чем расстояние между строками блока (рисунок 9.8).

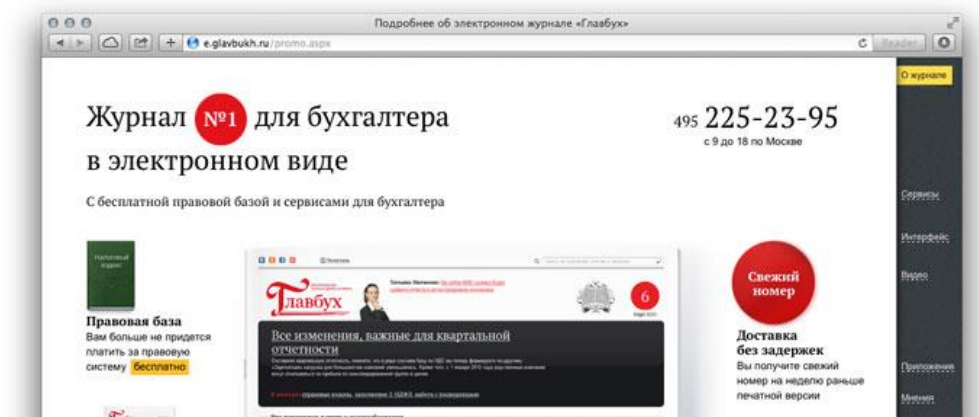


Рисунок 9.7 – Двустрочный заголовок, подзаголовок и иллюстрация с корректными отступами



Рисунок 9.8 – Двустрочный заголовок, подзаголовок и иллюстрация с увеличенным отступом

### ***Практическое задание***

1. Создать дизайн-макет страницы сайта.
2. Разместить на дизайн-макете страницы сайта заголовки, изображения, текст различных уровней.
3. Задать величину верхнего поля и расстояния между строками в соответствии с правилами соотношения внутренних и внешних отступов.
4. Проанализировать полученный результат.
5. При необходимости выполнить корректировку верхнего поля и расстояния между строками дизайн-макета.

### ***Содержание отчета***

1. Цель работы.
2. Дизайн-макет страницы сайта.
3. Выводы по работе.

### ***Контрольные вопросы***

1. Какие параметры предпочтительно использовать при создании макета?
2. Как выполняется выравнивание элементов на странице?
3. Как выполняется группировка слоев?
4. Как правила композиции применяются к дизайну страниц сайта?

## **Тема 10**

### **Подготовка макета веб-приложения к верстке**

*Цель:* формирование у студентов знаний структуры дизайн-макета и навыков подготовки макета к верстке.

### ***План занятия***

1. Изучить теоретические сведения.
2. Выполнить практическое задание по лабораторной работе.
3. Оформить отчет и ответить на контрольные вопросы.

### ***Теоретические сведения***

Разработаны 11 правил подготовки макета к верстке.

1. Группируйте слои по папкам и придерживайтесь иерархии в макете. Логические блоки сайта объединяются в группы с соответствующими названиями. Каждая группа слоев располагается в соответствии с ее положением на макете (рисунок 10.1).

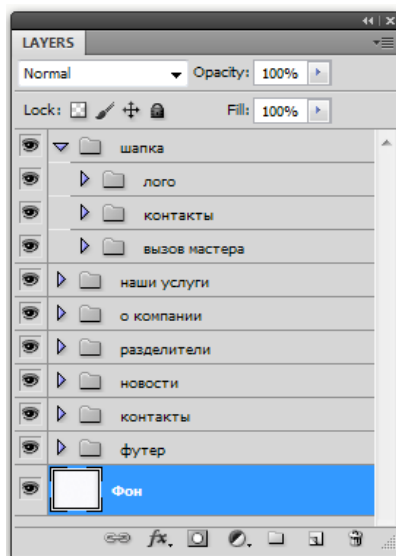
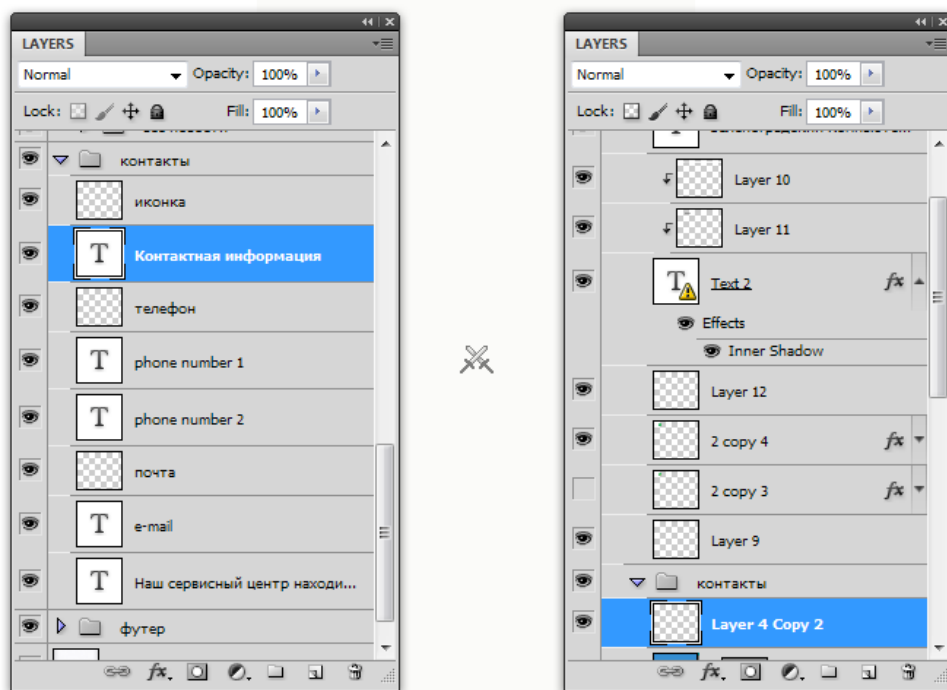


Рисунок 10.1 – Пример правильно организованных слоев

2. Давайте слоям осмысленные названия (рисунок 10.2).



*а*

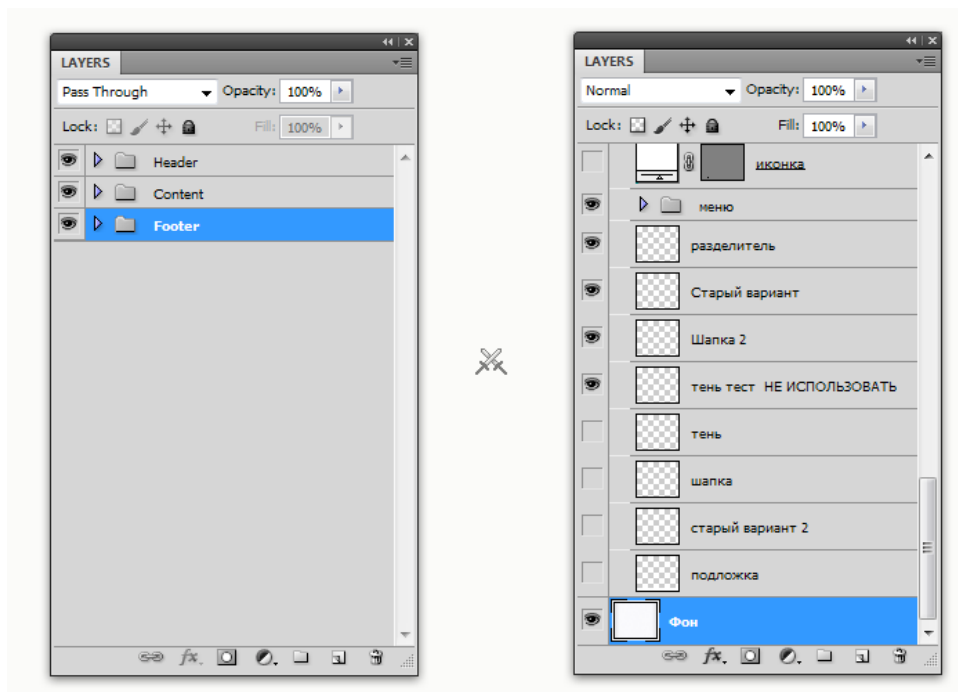
*б*

*а* – правильно; *б* – неправильно

Рисунок 10.2 – Примеры названий слоев

### 3. Удаляйте все ненужные слои.

Верстальщик сверстает все, что будет в макете, независимо от того, скрыт слой или нет. Поэтому целесообразно удалить ненужные и неиспользуемые слои из макета (рисунок 10.3).



*а*

*б*

*а* – правильно; *б* – неправильно

Рисунок 10.3 – Пример работы со слоями

### 4. Используйте сетку и направляющие (рисунок 10.4).

Рекомендуется использовать генератор сеток и отдельный слой с сеткой в макете или направляющие в Adobe Photoshop, чтобы показать верстальщику, какие параметры отступов есть у сетки.

### 5. Давайте пояснения к макету, используя инструмент «Notes».

Рекомендуется подробно комментировать все элементы, которые могут вызвать вопросы, имеют специфические или альтернативные значения. Например, радиус скругления углов, цвет ссылки при наведении или после ее посещения.

6. Для элементов, у которых может быть несколько состояний, отрисовывайте все варианты этих состояний.

Пример: у каждой кнопки, ссылки или поля для ввода в макете будет по три состояния, которым соответствуют группы или слои с осмысленными названиями.

7. Не присылайте верстальщику макет, где какой-либо эффект достигается с помощью режимов наложения.



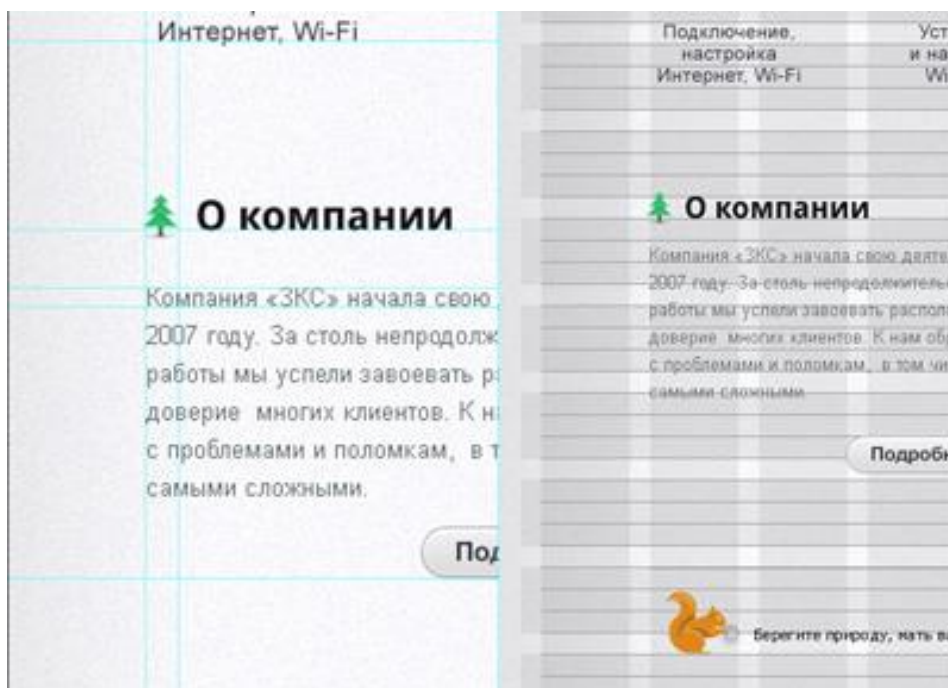


Рисунок 10.4 – Пример использования направляющих и сетки

**8.** Фигуры должны быть отрисованы в векторе.

Все нестандартные элементы в макете сохраняют в векторном формате, т. к. в любой момент может понадобиться изменить их размер.

**9.** Давайте информативные названия файлам.

Пример: последний вариант макета называется final, а предыдущие версии пронумерованы, чтобы избежать путаницы.

**10.** Прикрепляйте к файлу PSD также дополнительные материалы и изображения, использованные в макете.

**11.** Давайте пояснения к макету, используя инструмент «Notes».

Рекомендуется подробно комментировать все элементы, которые могут вызвать вопросы, имеют специфические или альтернативные значения. Например, радиус скругления углов, цвет ссылки при наведении или после ее посещения.

**12.** Для элементов, у которых может быть несколько состояний, отрисовывайте все варианты этих состояний.

Пример: у каждой кнопки, ссылки или поля для ввода в макете будет по три состояния, которым соответствуют группы или слои с осмысленными названиями.

**13.** Не присылайте верстальщику макет, где какой-либо эффект достигается с помощью режимов наложения.

**14.** Фигуры должны быть отрисованы в векторе.

Все нестандартные элементы в макете сохраняют в векторном формате, т. к. в любой момент может понадобиться изменить их размер.

**15.** Давайте информативные названия файлам.

Пример: последний вариант макета называется final, а предыдущие версии пронумерованы, чтобы избежать путаницы.

**16.** Прикрепляйте к файлу PSD также дополнительные материалы и изображения, использованные в макете.

Пример: все исходные изображения и дополнительные материалы прикрепляются к проекту в папку source или assets. Так даже при объединенных слоях всегда есть возможность получить отдельное изображение.

**17.** Прикрепляйте нестандартные шрифты, использованные в макете.

Пример: все нестандартные шрифты скопированы в папку с дополнительными материалами для проекта на тот случай, если у верстальщика нет этого шрифта.

### *Практическое задание*

1. В созданном дизайн-макете сгруппировать слои по папкам, придерживаясь иерархии слоев.

2. Отредактировать названия слоев на предмет соответствия названия назначению слоя.

3. Удалить из дизайн-макета все неиспользуемые слои.

4. Применить к дизайн-макету сетку и направляющие.

5. Оформить пояснения к дизайн-макету, используя инструмент «Notes».

6. Для элементов с несколькими состояниями предусмотреть все варианты этих состояний.

7. Проверить дизайн-макет на предмет отсутствия эффектов, достигаемых с помощью режимов наложения.

8. Проверить дизайн-макет на предмет реализации всех фигур средствами векторной графики.

9. Все файлы дизайн-макета назвать в соответствии с их версией и назначением.

10. К дизайн-макету прикрепить все используемые изображения и шрифты.

### *Содержание отчета*

1. Цель работы.

2. Отредактированный дизайн-макет страницы сайта.

3. Выводы по работе.

### *Контрольные вопросы*

1. Как подготовить к верстке иконки, используемые в макете?
2. Как выполняется объединение слоев по группам?
3. Как выполняется сохранение изображений, требуемых для верстки?
4. Какие названия для файлов предпочтительнее?

## **Тема 11**

### **Верстка макета веб-страниц**

*Цель:* формирование у студентов знаний о структуре HTML-документа и навыков верстки веб-страниц.

### *План занятия*

1. Изучить теоретические сведения.
2. Выполнить практическое задание по лабораторной работе.
3. Оформить отчет и ответить на контрольные вопросы.

### *Теоретические сведения*

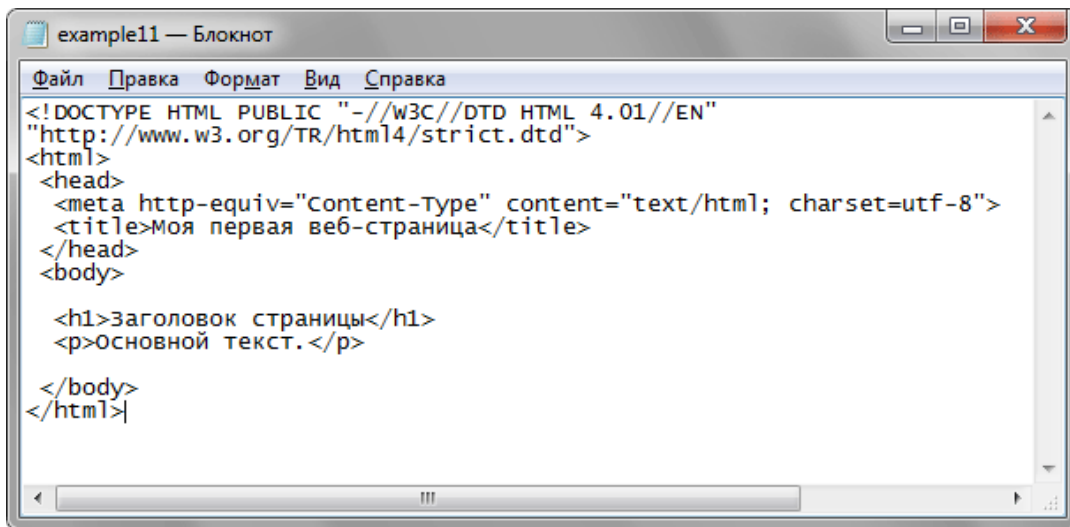
*HTML (HyperText Markup Language, язык разметки гипертекста)* – это система верстки, которая определяет, какие элементы и каким образом должны располагаться на веб-странице. Ниже приведен пример HTML-документа.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Моя первая веб-страница</title>
</head>
<body>
<h1>Заголовок страницы</h1>
<p>Основной текст.</p>
</body>
</html>
```

Посмотреть результат примера в действии можно, проделав следующие шаги.

1. В Windows откройте программу «Блокнот» («Пуск» → «Выполнить» → набрать «notepad» или «Пуск» → «Программы» → «Стандартные» → «Блокнот»).

2. Наберите код в «Блокноте» (рисунок 11.1).



```
example11 — Блокнот
Файл  Правка  Формат  Вид  Справка
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Моя первая веб-страница</title>
</head>
<body>

<h1>Заголовок страницы</h1>
<p>Основной текст.</p>

</body>
</html>
```

Рисунок 11.1 – Пример HTML-кода в «Блокноте»

3. Сохраните готовый документ («Файл» → «Сохранить как...») под именем `c:\www\example1.html`, при этом обязательно обозначьте в диалоговом окне сохранения тип файла «Все файлы» и кодировку UTF-8. Обратите внимание, что расширение файла должно быть именно `html` (рисунок 11.2).

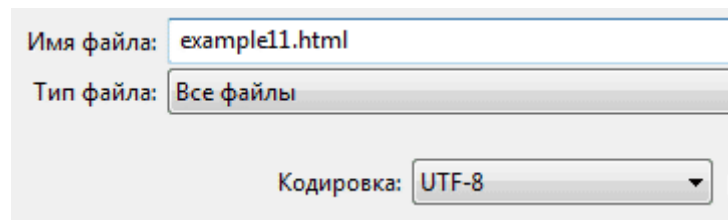


Рисунок 11.2 – Сохранение HTML-документа

4. Запустите браузер Internet Explorer («Пуск» → «Выполнить» → набрать «`iexplore`» или «Пуск» → «Программы» → Internet Explorer).

5. В браузере выберите пункт меню «Файл» → «Открыть» и укажите путь к вашему файлу.

6. Если все сделано правильно, то в браузере вы увидите результат, представленный на рисунке 11.3.

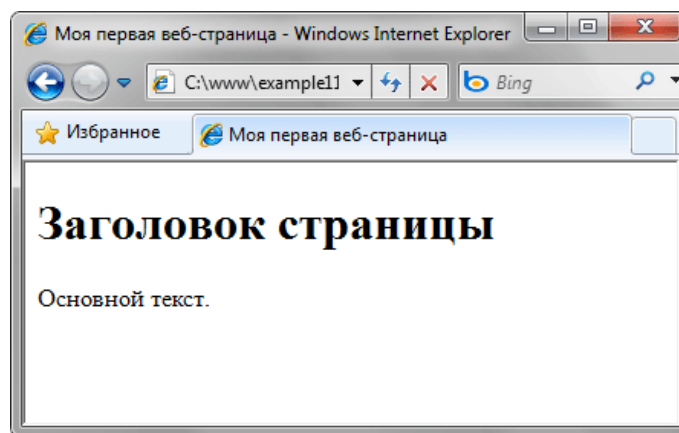


Рисунок 11.3 – Результат выполнения HTML-кода в браузере

Элемент `<!DOCTYPE>` предназначен для указания типа текущего документа – DTD (document type definition, описание типа документа). Это необходимо, чтобы браузер смог интерпретировать текущую веб-страницу, поскольку HTML существует в нескольких версиях, кроме того, имеется XHTML (EXtensible HyperText Markup Language, расширенный язык разметки гипертекста), похожий на HTML, но различающийся с ним по синтаксису.

Существует несколько видов `<!DOCTYPE>`, они различаются в зависимости от версии языка, на который ориентированы.

Тег `<html>` является контейнером, который заключает в себе все содержимое веб-страницы, включая теги `<head>` и `<body>`. Открывающий и закрывающий теги `<html>` в документе необязательны, но хороший стиль диктует непременно их использование. Как правило, тег `<html>` идет в документе вторым после определения типа документа, устанавливаемого через элемент `<!DOCTYPE>`. Закрывающий тег `<html>` должен всегда стоять в документе последним.

Тег `<head>` предназначен для хранения других элементов, цель которых – помочь браузеру в работе с данными. Содержимое тега `<head>` не отображается напрямую на веб-странице, за исключением тега `<title>`, устанавливающего заголовок окна веб-страницы. Внутри контейнера `<head>` допускается размещать следующие элементы: `<base>`, `<basefont>`, `<bgsound>`, `<link>`, `<meta>`, `<script>`, `<style>`, `<title>`.

Метатеги `<meta>` используются для хранения информации, предназначенной для браузеров и поисковых систем. Механизмы поисковых систем обращаются к метатегам для получения описания сайта, ключевых слов и других данных. Разрешается использовать более чем один метатег. Как правило, атрибуты любого

метатега сводятся к парам «имя = значение», которые определяются ключевыми словами content, name или http-equiv.

Тег <title> определяет заголовок документа. Элемент <title> не является частью документа и не показывается напрямую на веб-странице. В операционной системе Windows текст заголовка отображается в левом верхнем углу окна браузера. Допускается использовать только один тег <title> на документ и размещать его в контейнере <head>.

### ***Практическое задание***

1. Зарегистрироваться на онлайн-платформе <https://htmlacademy.ru>.
2. Пройти тему «Знакомство».
3. Пройти тему «Структура HTML-документа».
4. Пройти тему «Разметка текста с помощью HTML».
5. Проанализировать полученные результаты, обратить внимание на допущенные ошибки.
6. Для успешной защиты лабораторной работы набрать в сумме 45 баллов за все темы.
7. Повторить пункты 1–6 в случае, если итоговый результат окажется менее 45 баллов.

### ***Содержание отчета***

1. Цель работы.
2. Результаты работы по темам «Знакомство», «Структура HTML-документа», «Разметка текста с помощью HTML».
3. Выводы по работе.

### ***Контрольные вопросы***

1. Для чего предназначен элемент <!DOCTYPE> в HTML-документе?
2. Как используется тег <html> в HTML-документе?
3. Как используется тег <head> в HTML-документе?
4. Как используются метатеги?
5. Как используется тег <title>?

## Тема 12

### Работа с изображениями, ссылками, таблицами и списками в HTML

*Цель:* формирование у студентов знаний и навыков работы с изображениями, ссылками, таблицами и списками в HTML-документе.

#### *План занятия*

1. Изучить теоретические сведения.
2. Выполнить практическое задание по лабораторной работе.
3. Оформить отчет и ответить на контрольные вопросы.

#### *Теоретические сведения*

**Изображения.** Добавление изображения в HTML-документ происходит в два этапа: вначале готовится графический файл желаемого размера, затем он добавляется на страницу через тег `<img>`. HTML предназначен для того, чтобы отобразить требуемую картинку без изменений. Поскольку веб-страница загружается по сети, существенным фактором скорости является объем графического файла, встроенного в документ. Чем он меньше, тем быстрее отобразится изображение. Размер картинки необходимо ограничить по ширине, например, установить не более 800 пикселей, иначе изображение целиком не поместится в окне браузера, и появятся полосы прокрутки.

Тег `<img>` предназначен для отображения на веб-странице изображений в графическом формате GIF, JPEG или PNG. Адрес файла с картинкой задается через атрибут `src`. При необходимости рисунок можно сделать ссылкой на другой файл, поместив тег `<img>` в контейнер `<a>`. При этом вокруг изображения отображается рамка, которую можно убрать, добавив атрибут `border="0"` в тег `<img>`.

Для добавления изображения на веб-страницу используется тег `<img>`, атрибут `src` которого определяет адрес графического файла. Общий синтаксис добавления изображения будет следующий:

```

```

*URL (Universal Resource Locator, универсальный указатель ресурсов)* представляет собой путь к графическому файлу. Для его указания можно использовать как абсолютный, так и относительный адрес.

**Ссылки.** Ссылки являются основой гипертекстовых документов и позволяют переходить с одной веб-страницы на другую. Особенность их состоит в том,

что сама ссылка может вести не только на HTML-файлы, но и на файл любого типа, причем этот файл может размещаться на другом сайте. Главное, чтобы к документу, на который делается ссылка, был доступ. Иными словами, если путь к файлу можно указать в адресной строке браузера, и файл при этом будет открыт, то на него можно сделать ссылку.

Для создания ссылки необходимо сообщить браузеру, что является ссылкой, а также указать адрес документа, на который следует сделать ссылку. Оба действия выполняются с помощью тега <a>. Общий синтаксис создания ссылок следующий:

```
<a href="URL">текст ссылки</a>
```

Атрибут href определяет URL, то есть адрес документа, на который следует перейти, а содержимое контейнера <a> является ссылкой. Текст, расположенный между тегами <a> и </a>, по умолчанию становится синего цвета и подчеркивается.

**Таблицы.** Благодаря универсальности таблиц, большому числу параметров, управляющих их видом, таблицы надолго стали определенным стандартом для верстки веб-страниц. Таблица с невидимой границей представляет собой модульную сетку, в блоках которой удобно размещать элементы веб-страницы. Тем не менее это не совсем правильный подход, ведь каждый объект HTML определен для собственных целей, и если он используется не по назначению повсеместно, это значит, что альтернатив нет. Однако на смену таблицам при верстке сайтов пришли слои, и наметилась следующая тенденция: таблицы применяются для размещения табличных данных, а слои – для верстки и оформления.

Таблица состоит из строк и столбцов ячеек, которые могут содержать текст и рисунки. Обычно таблицы используются для упорядочения и представления данных, однако возможности таблиц этим не ограничиваются. С их помощью также удобно верстать макеты страниц, расположив нужным образом фрагменты текста и изображений.

Для добавления таблицы на веб-страницу используется тег <table>. Этот элемент служит контейнером для элементов, определяющих содержимое таблицы. Строки и ячейки задаются соответственно с помощью тегов <tr> и <td>. Таблица должна содержать хотя бы одну ячейку. Допускается вместо тега <td> использовать тег <th>. Текст в ячейке, оформленной с помощью тега <th>, отображается браузером шрифтом жирного начертания и выравнивается по центру ячейки. В остальном разницы между ячейками, созданными через теги <td> и <th>, нет. Далее приведен пример создания таблицы:



```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>Тер TABLE</title>
</head>
<body>
<table border="1" width="100%" cellpadding="5">
  <tr>
    <th>Ячейка 1</th>
    <th>Ячейка 2</th>
  </tr>
  <tr>
    <td>Ячейка 3</td>
    <td>Ячейка 4</td>
  </tr>
</table>
</body>
</html>

```

**Списки.** Списком называется взаимосвязанный набор отдельных фраз или предложений, которые начинаются с маркера или цифры. Списки предоставляют возможность упорядочить и систематизировать разные данные и представить их в наглядном и удобном для пользователя виде.

Любой список представляет собой контейнер `<ul>`, который устанавливает маркированный список, или контейнер `<ol>`, определяющий список нумерованный. Каждый элемент списка должен начинаться с тега `<li>`.

Маркированный список определяется тем, что перед каждым элементом списка добавляется небольшой маркер, как правило, в виде закрашенного кружка.

```

<ul>
  <li>Первый пункт</li>
  <li>Второй пункт</li>
  <li>Третий пункт</li>
</ul>

```

Нумерованные списки представляют собой набор элементов с их порядковыми номерами. Вид и тип нумерации зависит от атрибутов тега `<ol>`.

```
<ol>
  <li>Первый пункт</li>
  <li>Второй пункт</li>
  <li>Третий пункт</li>
</ol>
```

### ***Практическое задание***

1. Зарегистрироваться на онлайн-платформе <https://htmlacademy.ru>.
2. Пройти темы «Ссылки и изображения», «Знакомство с таблицами».
3. Для успешной защиты лабораторной работы необходимо набрать в сумме 64 балла за эти темы.
4. Повторить пункты 1–3 в случае, если итоговый результат окажется менее 64 баллов.

### ***Содержание отчета***

1. Цель работы.
2. Результаты работы по темам «Ссылки и изображения», «Знакомство с таблицами».
3. Выводы по работе.

### ***Контрольные вопросы***

1. Как добавить изображение в HTML-документ?
2. Как создать ссылку в HTML-документе?
3. Как добавить таблицу в HTML-документ?
4. Как создать список в HTML-документе?

## **Тема 13**

### **Каскадные таблицы стилей**

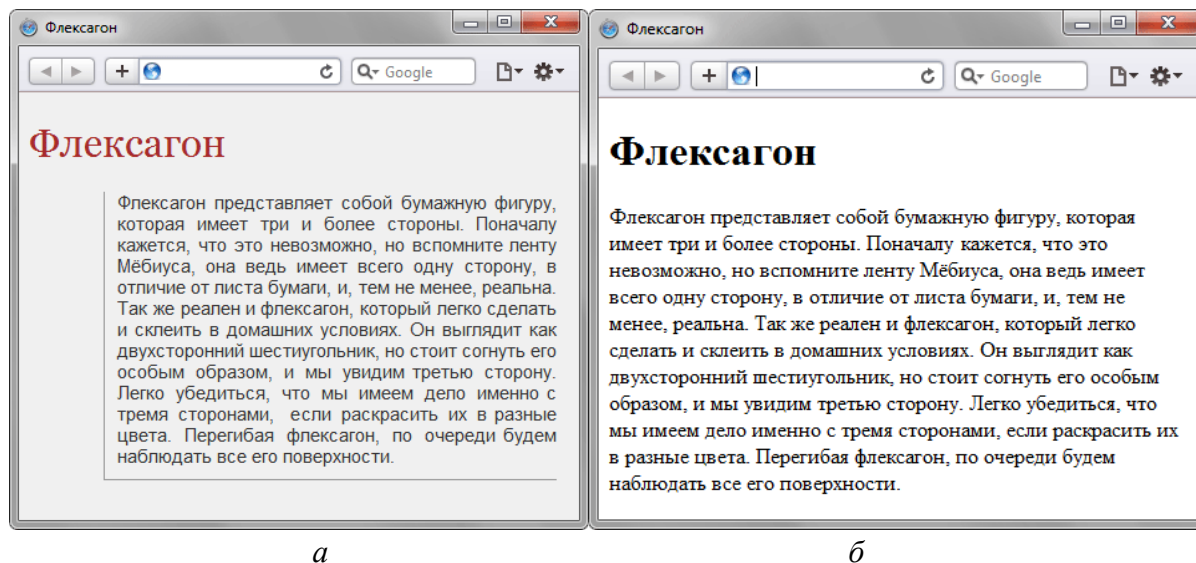
*Цель:* формирование у студентов знаний и навыков работы с каскадными таблицами стилей.

### ***План занятия***

1. Изучить теоретические сведения.
2. Выполнить практическое задание по лабораторной работе.
3. Оформить отчет и ответить на контрольные вопросы.

## Теоретические сведения

*Стилем*, или *CSS (Cascading Style Sheets, каскадные таблицы стилей)*, называется набор параметров форматирования, который применяется к элементам документа, чтобы изменить их внешний вид (рисунок 13.1).



*а* – веб-страница без CSS; *б* – с применением CSS

Рисунок 13.1 – Применение CSS

Код страницы с CSS приведен ниже.

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Флексагон</title>
    <meta charset="utf-8">
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <h1>Флексагон</h1>
    <p>Флексагон представляет собой бумажную фигуру, которая имеет три и более сторон. Поначалу кажется, что это невозможно, но вспомните ленту Мебиуса, она ведь имеет всего одну сторону в отличие от листа бумаги и, тем не менее, реальна. Так же реален и флексагон, который легко сделать и склеить в домашних условиях. Он выглядит как двухсторонний шестиугольник, но стоит согнуть его особым образом, и мы увидим третью сторону. Легко убедиться, что мы имеем дело именно с тремя сторонами, если раскрасить их в разные цвета. Перегибая флексагон, по очереди будем наблюдать все его поверхности.</p>
  </body>
</html>
```

разные цвета. Перегибая флексагон, по очереди будем наблюдать все его поверхности.</p>  
</body>  
</html>

Сам HTML-код изменений не претерпел, и единственное добавление – это строка <link rel="stylesheet" href="style.css">. Она ссылается на внешний файл с описанием стилей под именем style.css. Содержимое этого файла показано далее.

```
body {
  font-family: Arial, Verdana, sans-serif; /* Семейство шрифтов */
  font-size: 11pt; /* Размер основного шрифта в пунктах */
  background-color: #f0f0f0; /* Цвет фона веб-страницы */
  color: #333; /* Цвет основного текста */
}
h1 {
  color: #a52a2a; /* Цвет заголовка */
  font-size: 24pt; /* Размер шрифта в пунктах */
  font-family: Georgia, Times, serif; /* Семейство шрифтов */
  font-weight: normal; /* Нормальное начертание текста */
}
p {
  text-align: justify; /* Выравнивание по ширине */
  margin-left: 60px; /* Отступ слева в пикселах */
  margin-right: 10px; /* Отступ справа в пикселах */
  border-left: 1px solid #999; /* Параметры линии слева */
  border-bottom: 1px solid #999; /* Параметры линии снизу */
  padding-left: 10px; /* Отступ от линии слева до текста */
  padding-bottom: 10px; /* Отступ от линии снизу до текста */
}
```

В файле style.css описаны все параметры оформления таких тегов, как <body>, <h1> и <p>. Следует отметить, что сами теги в коде HTML пишутся обычным способом.

Поскольку на файл со стилем можно ссылаться из любого веб-документа, это приводит в итоге к сокращению объема повторяющихся данных. А благодаря разделению кода и оформления повышается гибкость управления видом документа и скорость работы над сайтом.

### ***Практическое задание***

1. Зарегистрироваться на онлайн-платформе <https://htmlacademy.ru>.
2. Пройти тему «Знакомство с CSS».
3. Пройти тему «Селекторы».
4. Пройти тему «Наследование и каскадирование».
5. Проанализировать полученные результаты, обратить внимание на допущенные ошибки.
6. Для успешной защиты лабораторной работы необходимо набрать в сумме 51 балл за эти темы.
7. Повторить пункты 1–6 в случае, если итоговый результат окажется менее 51 балла.

### ***Содержание отчета***

1. Цель работы.
2. Результаты работы по темам «Ссылки и изображения», «Знакомство с таблицами».
3. Выводы по работе.

### ***Контрольные вопросы***

1. Что такое CSS?
2. Как CSS интегрируется в HTML-документ?
3. Что содержит файл .css?
4. Каковы преимущества использования CSS?

## **Тема 14**

### **Интерактивность в веб-приложении**

*Цель:* формирование у студентов знаний и навыков программирования на языке JavaScript.

### ***План занятия***

1. Изучить теоретические сведения.
2. Выполнить практическое задание по лабораторной работе.
3. Оформить отчет и ответить на контрольные вопросы.

### ***Теоретические сведения***

Цель *JavaScript* – сделать веб-страницы интерактивными. В браузере программы на JavaScript подключаются напрямую к HTML и выполняются, как толь-

ко загружается страница. Для этого во все основные браузеры встроен интерпретатор JavaScript. Программы на JavaScript также можно запускать на сервере.

Современный JavaScript – это «безопасный» язык программирования общего назначения. Он не предоставляет низкоуровневых средств работы с памятью, процессором. Дополнительные возможности зависят от окружения, в котором запущен JavaScript. В браузере JavaScript позволяет: создавать новые HTML-теги, удалять существующие, менять стили элементов, прятать или показывать элементы и т. п.; реагировать на действия посетителя, обрабатывать щелчки мыши, перемещения курсора, нажатия на клавиатуру и т. п.; посылать запросы на сервер и загружать данные без перезагрузки страницы (технология AJAX); получать и устанавливать cookie, запрашивать данные, выводить сообщения. JavaScript не может читать и записывать произвольные файлы на жесткий диск, копировать их или вызывать программы. Также он не имеет прямого доступа к операционной системе. JavaScript, работающий в одной вкладке, не может общаться с другими вкладками и окнами, за исключением случая, когда он сам открыл это окно или несколько вкладок из одного источника (одинаковый домен, порт, протокол).

Из JavaScript можно легко посылать запросы на сервер, с которого пришла страница. Запрос на другой домен тоже возможен, но менее удобен, т. к. имеет ограничения безопасности.

Программы на языке JavaScript можно вставить в HTML-код при помощи тега `<script>`.

```
<!DOCTYPE HTML>
<html>
  <head>
    <!-- Тег meta для указания кодировки -->
    <meta charset="utf-8">
  </head>
  <body>
    <p>Начало документа...</p>
    <script>
      alert('Привет, Мир!');
    </script>
    <p>...Конец документа</p>
  </body>
</html>
```

Этот пример использует следующие элементы:

```
<script> ... </script>
```

Тег `<script>` содержит исполняемый код. Предыдущие стандарты HTML требовали обязательного указания атрибута `type`. Теперь необходимость в нем отсутствует, достаточно использовать тег `<script>`. Браузер, когда видит этот тег, начинает отображать страницу и показывает часть документа до `<script>`. Встретив тег `<script>`, переключается в JavaScript-режим и не показывает, а исполняет его содержимое. Закончив выполнение, браузер возвращается обратно в HTML-режим, после чего отображает оставшуюся часть документа.

Если JavaScript-кода много – его выносят в отдельный файл, который подключается в HTML:

```
<script src="/path/to/script.js"></script>
```

Здесь `/path/to/script.js` – это абсолютный путь к файлу, содержащему скрипт (из корня сайта). Браузер сам скачивает и выполняет скрипт.

Можно указать и полный URL, например:

```
<script  
src="https://cdnjs.cloudflare.com/ajax/libs/lodash.js/4.3.0/lodash.js">  
</script>
```

Можно использовать путь относительно текущей страницы. Например, `src="lodash.js"` обозначает файл из текущей директории.

Для подключения нескольких скриптов используйте несколько тегов:

```
<script src="/js/script1.js"></script>  
<script src="/js/script2.js"></script>
```

### ***Практическое задание***

1. Сверстать контейнер с тремя закладками, используя JavaScript. Все закладки должны иметь разные цвета. Каждая закладка должна открывать содержимое контейнера соответствующего цвета (рисунок 14.1).
2. Протестировать корректность работы разработанного скрипта.
3. Проанализировать разработанную программу с точки зрения простоты и эффективности использования в ней JavaScript.

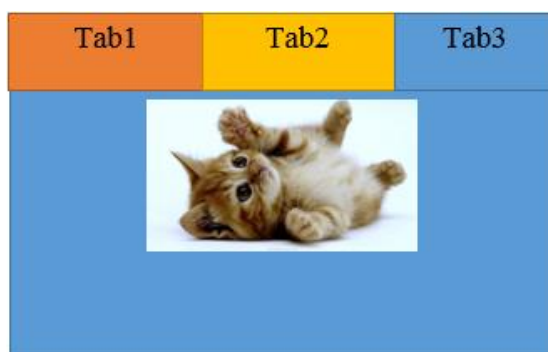


Рисунок 14.1 – Пример выполнения практического задания

#### *Содержание отчета*

1. Цель работы.
2. Результаты верстки контейнера с тремя закладками разных цветов.
3. Выводы по работе.

#### *Контрольные вопросы*

1. Для чего предназначен язык программирования JavaScript?
2. Каковы ограничения исполнения JavaScript в браузере?
3. Как интегрировать JavaScript в HTML-код?
4. Как подключить несколько скриптов?

## **Тема 15**

### **Клиент-серверное взаимодействие**

*Цель:* формирование у студентов знаний о клиент-серверном взаимодействии и навыков разработки HTML-контента на сервере.

#### *План занятия*

1. Изучить теоретические сведения.
2. Выполнить практическое задание по лабораторной работе.
3. Оформить отчет и ответить на контрольные вопросы.

#### *Теоретические сведения*

Веб-приложения используют клиент-серверную модель взаимодействия. Клиент-серверное взаимодействие обеспечивает необходимый обмен информацией между клиентским устройством и сервером и работает с целью доставки ин-



формации как от сервера клиенту (например, доставка информации о доступных рейсах в аэропорту), так и от клиента серверу (например, доставка информации о конкретном рейсе, билет на который хочет купить пользователь).

Для веб-приложений, предоставляющих данные в формате HTML с применением таблиц стилей, скриптов на языке JavaScript, изображений, видео, звука, клиент-серверное взаимодействие реализуется с использованием HTTP-протокола (протокола передачи гипертекстовых документов).

*HTTP* – это текстовый протокол прикладного уровня, в котором единичный факт передачи информации представляет собой специальный запрос, содержащий шапку и тело запроса. Пример такого запроса приведен ниже.

```
GET http://www.msn.com/ HTTP/1.1
Host: www.msn.com
Accept: text/html
Accept-Language: en-US
User-Agent: Mozilla/5.0
```

Запрашиваем по адресу данные, расположенные на узле [www.msn.com](http://www.msn.com), просим прислать ответ в формате text/html на английском языке, передаем имя пользовательской программы Mozilla/5.0.

В этом случае можно получить такой ответ:

```
HTTP/1.1 200 OK
Content-Length: 309260
Content-Type: text/html; charset=utf-8
Server: Microsoft-IIS/8.5
Date: Wed, 09 Sep 2015 13:50:28 GMT
```

Получаем подтверждение от сервера об успешном формировании данных длиной 309260 байт в формате text/html и кодировке utf-8. Обслуживающий веб-сервер Microsoft-IIS/8.5, дата 9 сентября 2015, время 13:50. Далее следуют сами данные.

```
<!DOCTYPE html><html><head . .
```

Можно также передать данные на сервер. Пример такого запроса приведен ниже.

```
POST
https://profile.tut.by/authorize/
HTTP/1.1
Host: profile.tut.by
Content-Length: 52
Content-Type: application/x-www-form-urlencoded
```

Передача данных с использованием защищенного канала (https) по адресу profile.tut.by/authorize/ расположенному на узле profile.tut.by.

Данные длиной 52 байта в формате application/x-www-form-urlencoded.

Сами данные представляют со-

log-  
in\_action=login&login=test&password=tes  
t&memory=1

бой пары «ключ – значение»:  
login\_actio=login  
login=test  
password=test  
memory=1

В качестве ответа можно получить следующее.

HTTP/1.1 302 Found	Ответ-перенаправление, сервер
Server: nginx/1.7.6	nginx/1.7.6, дата 9 сентября 2015, время
Date: Wed, 09 Sep 2015 14:01:52 GMT	14:01, данные в формате text/html в ко- дировке windows-1251 длиной 0 байт.
Content-Type: text/html; charset=windows-1251	Адрес перенаправления
Content-Length: 0	http://www.tut.by/?error=login_error.
Location: http://www.tut.by/?error=login_	
Error	

Стандарт HTTP описывает структуру запросов, важной частью которых является Method, или Verb (глагол), описывающий намерения пользователя. Основными методами, или глаголами, являются GET (используется для выражения намерения запроса информации с сервера, как в первом примере) и POST (используется для выражения намерения передачи информации на сервер, как во втором примере).

Если пользователь взаимодействует со статическим сайтом (набором сверстаных HTML-страниц на веб-сервере), все варианты взаимодействия можно описать методом GET. Клиент-серверное взаимодействие в этом случае выглядит следующим образом.

1. Клиент отправляет HTTP-запрос на сервер с указанием идентификатора ресурса.
2. Сервер принимает запрос от клиента.
3. Сервер находит на диске документ, соответствующий идентификатору ресурса.
4. Сервер формирует HTTP-ответ клиенту и отдает ему HTML-документ.

Как правило, вместе с HTML-версткой используется язык программирования (например, PHP). Это позволяет при наличии одного HTML-документа описать изменения во внутренней структуре этого документа как реакцию приложения на запросы пользователя.

*PHP* – это интерпретируемый язык программирования. Это означает, что программа-интерпретатор выполняет инструкции, написанные на языке PHP, строка за строкой, без каких-либо промежуточных преобразований. Файлы, содержащие инструкции на языке PHP, имеют расширение .php, однако результатом выполнения этих файлов обычно является HTML-документ. Клиент-серверное взаимодействие в этом случае выглядит следующим образом.

1. Клиент отправляет HTTP-запрос на сервер с указанием идентификатора ресурса.
  2. Сервер принимает запрос от клиента.
  3. Сервер определяет, что запрашиваемый ресурс ассоциирован с интерпретатором PHP.
  4. Сервер передает управление интерпретатору, сообщая ему детали запроса.
  5. Интерпретатор выполняет инструкции, указанные в файле, на который ссылается запрос пользователя, в результате получается HTML-документ.
  6. Интерпретатор возвращает HTML-документ серверу.
  7. Сервер формирует HTTP-ответ клиенту и отдает ему HTML-документ.
- Для выполнения инструкций на языке PHP нужен PHP-интерпретатор и веб-сервер. Установить их можно, используя инсталлятор AppServ.

### ***Практическое задание***

1. Установить AppServ <http://htmlbook.ru/webserver/appserv>.
2. Написать программу, динамически генерирующую HTML-страницу по POST-запросу пользователя.
3. Протестировать выполнение программы.
4. Проанализировать полученные результаты. Выделить клиентскую и серверную часть.

### ***Содержание отчета***

1. Цель работы.
2. Программа, динамически генерирующая HTML-страницу по POST-запросу пользователя.
3. Выводы по работе.

### ***Контрольные вопросы***

1. Какую модель взаимодействия использует веб-приложение?
2. Что такое и для чего необходим HTTP-протокол?
3. Что представляет собой клиент-серверное взаимодействие в случае статического сайта?

4. Для чего предназначен язык программирования PHP?
5. Что представляет собой клиент-серверное взаимодействие при программировании серверной логики на PHP?

## **Тема 16**

### **Поисковая оптимизация**

*Цель:* формирование у студентов знаний и навыков использования инструментов анализа качества интернет-контента с точки зрения поисковой оптимизации.

#### *План занятия*

1. Изучить теоретические сведения.
2. Выполнить практическое задание по лабораторной работе.
3. Оформить отчет и ответить на контрольные вопросы.

#### *Теоретические сведения*

*Поисковой оптимизацией (SEO, Search Engine Optimization)* называется процесс разработки и сопровождения веб-ресурсов с целью достижения и сохранения высоких позиций в результатах поисковых систем по конкретным ключевым фразам.

Методы SEO направлены на влияние и совершенствование в трех основных областях:

- 1) критические аспекты, на которые может повлиять профессиональный UX-проектировщик: инфраструктура сайта, используемые технологии и принципы организации сайта;
- 2) контент, ключевые слова и все, что относится к оптимизации по критериям поисковых систем;
- 3) ссылки и популярность по ссылкам – количество и качество ссылок, ведущих на сайт с других сайтов, а также структура организации ссылок внутри сайта.

Основные принципы работы поисковых систем почти не изменились с 1993 года, когда в Интернете впервые была запущена программа World Wide Web Wanderer для обработки страниц и построения первой поисковой системы. В каждой поисковой системе имеется специальное приложение, которое называется пауком (spider), ботом (bot) или обходчиком (crawler); оно находит ссылки и переходит по ним, отправляя в базу данных копию обнаруженного ресурса. Далее содер-

жимое базы данных анализируется по закрытому алгоритму, разработанному создателями поисковой системы. Ресурс индексируется по правилам этого алгоритма и затем ранжируется в зависимости от показателей, вычисленных по критериям поисковой системы. Процесс может показаться довольно бесхитростным, однако в нем скрыты бесчисленные ловушки для UX-проектировщиков.

Понимание основных принципов индексирования позволит взглянуть на сайт с точки зрения поисковых систем. Хорошо оптимизированные сайты облегчают задачу поисковых систем с помощью грамотно спроектированной структуры и правильно выбранной технологической базы. Кроме того, на ранжирование результатов поисковыми системами влияют решения в области организации контента. Таким образом, очень многое зависит от исходных решений, принятых при прототипировании страниц сайта и в дискуссиях по поводу стилового оформления и управления контентом.

**Динамические URL-адреса.** Поисковые системы воспринимают не «страницы» контента, а пути к этому контенту. Изменение пути (или URL-адреса) к контенту приведет к тому, что поисковая система клонирует контент и будет хранить несколько экземпляров. Такая ситуация существенно ухудшает поисковые перспективы сайта.

**Непреднамеренное клонирование.** Если исходить из того, что любая уникальная страница должна быть доступна только по одному URL-пути, то будет нетрудно просчитать другие ситуации, приводящие к непреднамеренному клонированию контента. Архитектура системы должна связывать один блок содержимого только с одним URL-путем.

**Циклические ссылки.** Непреднамеренное клонирование часто приводит к появлению циклических ссылок. Проследите за тем, чтобы поисковые системы не попали на теоретически бесконечный маршрут отслеживания ссылок «далее», например в календаре: если поисковая система перейдет по ссылке на следующий день календаря, он обнаружит там еще одну ссылку, перейдет по ней и т. д. Подобные ситуации предотвращаются при помощи JavaScript-ссылок, по которым поисковые системы переходить не могут; тем самым вы направите усилия поисковых систем на тот контент, который хотите видеть проиндексированным.

**Название страницы.** Тег <title> содержит название страницы. Каждая страница сайта должна иметь уникальное название. Заполните тег <title> самыми важными для данной страницы словами.

**Заголовки.** Поисковые системы распознают заголовки <h1>, <h2> и так далее и считают их влиятельным фактором индексирования при условии, что ими не злоупотребляют. Старайтесь снабжать разделы такими заголовками, которые были

бы одновременно информативными и содержали актуальные для этой страницы ключевые слова.

**Текст якорных ссылок.** Текст якорных ссылок серьезно влияет на то, как поисковая система интерпретирует страницу на другом конце ссылки. Именно этот фактор создал «эффект GoogleBomb». Если на какую-то страницу ведет достаточно большое количество ссылок с одним и тем же текстом внутри, Google считает, что страница релевантна ключевым словам в тексте ссылки. Например, если провести поиск в Google по словам «click here» (щелкните здесь), в верхних позициях результата окажется сайт Adobe. На него ведут сотни тысяч ссылок с текстом «click here to download Adobe Reader» (щелкните здесь, чтобы загрузить Adobe Reader) и т. п. Можно обратить это обстоятельство себе на пользу: якорная ссылка должна содержать не обобщенный текст вроде «Далее» или «Щелкните здесь» – вместо этого следует использовать в ней ключевые слова, имеющие прямое отношение к той странице, на которую она ведет.

**Индекс цитирования.** Это мера количества и качества ссылок, ведущих на ресурс с других веб-страниц. Google называет это термином PageRank. Каждая ссылка рассматривается как голос, поданный в пользу ресурса. В основе этой идеи лежит предположение, что если информация интересна или полезна для других, то на нее будут вести ссылки с других доверенных ресурсов. Такая концепция в наши дни стала фундаментальным принципом получения качественных результатов поиска. UX-проектировщик должен хорошо разбираться в этом вопросе и понимать, как индекс цитирования распределяется по страницам в структуре веб-сайта.

### ***Практическое задание***

1. Сформировать динамические URL-адреса в рамках поисковой оптимизации веб-сайта.
2. Проверить веб-сайт на предмет отсутствия циклических ссылок.
3. Отредактировать веб-сайт при необходимости по результатам п. 2.
4. Названия страниц и заголовки разделов сформулировать таким образом, чтобы они содержали актуальные для этой страницы ключевые слова.
5. Проверить веб-сайт на предмет уникальности названий страниц.
6. Отредактировать веб-сайт при необходимости по результатам п. 5.
7. В названиях якорных ссылок применить ключевые для веб-сайта слова.

### ***Содержание отчета***

1. Цель работы.
2. Инструменты и результаты поисковой оптимизации веб-сайта.
3. Выводы по работе.

### *Контрольные вопросы*

1. Что такое поисковая оптимизация?
2. Какие направления охватывают методы SEO-оптимизации?
3. Каковы основные принципы работы поисковых систем?
4. Каковы особенности обработки динамических адресов поисковыми системами?
5. Каковы особенности обработки циклических ссылок поисковыми системами?
6. Каковы особенности обработки названия страницы, заголовков поисковыми системами?
7. Каковы особенности обработки якорных ссылок поисковыми системами?
8. Что такое индекс цитирования?

## Литература

1. Купер, А. Алан Купер об интерфейсе. Основы проектирования взаимодействия /А. Купер, Р. Рейман, Д. Кронин ; пер. с англ. – СПб. : Символ-Плюс, 2009. – 688 с.
2. Проектирование с помощью сторибордов [Электронный ресурс]. – 2012. – Режим доступа : <https://special.habrahabr.ru/kyocera/p/152375/>.
3. Прототипирование на бумаге [Электронный ресурс]. – 2010. – Режим доступа : <https://watcherr.wordpress.com/2010/08/04/прототипирование-на-бумаге>.
4. Инструменты быстрого прототипирования [Электронный ресурс]. – 2009. – Режим доступа : <http://habrahabr.ru/post/70001>.
5. Круг, С. Веб-дизайн: книга Стива Круга или «не заставляйте меня думать!» / С. Круг ; пер. с англ. – СПб. : Символ-Плюс, 2008. – 224 с.
6. Bradley, J. Wizard of Oz Experiments for Companions / J. Bradley, O. Mival, D. Benyon // HCI 2009 People and Computers XXIII. Celebrating people and technology. – Edinburgh HCI, 2009. – P. 313–317.
7. Example Usability Test with a Paper Prototype [Электронный ресурс]. – 2010. – Режим доступа : [https://www.youtube.com/watch?v=9wQkLthhHKA&ab\\_channel=BlueDuckLabs](https://www.youtube.com/watch?v=9wQkLthhHKA&ab_channel=BlueDuckLabs).
8. Adobe Support [Электронный ресурс]. – 2016. – Режим доступа : <https://helpx.adobe.com>.
9. Дизайн-бюро Артёма Горбунова. Как собрать страницу из модулей: зачем нужна сетка? [Электронный ресурс]. – 2014. – Режим доступа : <http://artgorbunov.ru/bb/soviet/20141013/>.
10. Дизайн-бюро Артема Горбунова. Якорные объекты [Электронный ресурс]. – 2014. – Режим доступа : <http://artgorbunov.ru/bb/soviet/20140324/>.
11. Проектирование, дизайн и внедрение эффективных интерфейсов [Электронный ресурс]. – 2016. – Режим доступа : <http://popel.agency/ru/>.
12. Самоучитель HTML4 [Электронный ресурс]. – 2010. – Режим доступа : <http://htmlbook.ru/samhtml>.
13. Современный учебник JavaScript [Электронный ресурс]. – 2010. – Режим доступа : <http://learn.javascript.ru/>.
14. Унгер, Р. UX-дизайн. Практическое руководство по проектированию опыта взаимодействия/ Р. Унгер, К. Чендлер ; пер. с англ. – СПб. : Символ-Плюс, 2011. – 336 с.