МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Тульский государственный университет»

Политехнический институт
Кафедра «Промышленная автоматика и робототехника»

Утверждено на заседании кафедры
«Промышленная автоматика
и робототехника»
«17» января 2023 г., протокол № 2

И.о. заведующего кафедрой

_____О.А. Ерзин

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ КУРСОВОЙ РАБОТЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ) «Web-дизайн»

основной профессиональной образовательной программы высшего образования – программы магистратуры

по направлению подготовки
29.03.03 Технология полиграфического и упаковочного производства

с направленностью (профилем)
Технология полиграфического производства

Формы обучения: заочная

Идентификационный номер образовательной программы: 290303-01-23

Тула 2023 год

ЛИСТ СОГЛАСОВАНИЯ
методических указаний по выполнению курсовой работы дисциплины (модуля)

Разработчик:

Яковлев Б.С., доцент, канд. техн. наук _____
(ФИО, должность, ученая степень, ученое звание) (подпись)

СОДЕРЖАНИЕ

1. ЦЕЛЬ И ЗАДАЧИ ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ

Курсовая работа по дисциплине "Web-дизайн" является самостоятельной работой студента, цель которой - закрепление и углубление знаний по дисциплине и получение практических навыков по разработки концепции дизайна и созданию веб-сайта.

1.1. Цель курсовой работы

Целью выполнения курсовой работы является развитие у студента способности самостоятельно решать задачи, связанные с созданием электронных публикаций, построением единой концепции дизайна макета и стратегии создания веб-сайта.

1.2. Задачи курсовой работы

Основными задачами выполнения курсовой работы являются:

- закрепление, углубление и обобщение знаний, полученных студентом в процессе изучения дисциплины " Web-дизайн ";
- изучение дополнительных материалов, связанных с основными правилами технического дизайна;
- освоение программ, используемых при создании и эксплуатации web-сайтов;
- овладение методами подготовки текстов и графики, предназначенных для публикации;
- формирование практических навыков по расчету модульной сетки, цветовой схемы, построению системы навигации и созданию веб-сайтов с использованием языка разметки гипертекста и каскадных списков стилей.
- изучение правил оформления работы в виде отчета в соответствии с требованиями.

2. ОСНОВНЫЕ ТРЕБОВАНИЯ К КУРСОВОЙ РАБОТЕ

2.1.Тематика курсовой работы

Курсовая работа посвящена разработке дизайна и проектированию веб-сайта, включая его текстовый и изобразительный контент, html-код и каскадные списки стиля заглавной и внутренней страниц.

2.2. Исходные данные к курсовой работе

Студент самостоятельно выбирает тематику проектируемого веб-сайта.

Размещаемые на сайте материалы не должны:

- нарушать действующее законодательство, честь и достоинство, права и охраняемые законом интересы третьих лиц, способствовать разжиганию религиозной, расовой или межнациональной розни, содержать сцены насилия, либо бесчеловечного обращения с животными, и т. д.;
- носить непристойный или оскорбительный характер;
- содержать рекламу наркотических средств;
- нарушать права несовершеннолетних лиц;
- нарушать авторские и смежные права третьих лиц;
- носить порнографический характер;

2.3. Задание на курсовую работу

В рамках выполнения курсовой работы требуется разработать следующие элементы сайта:

1. Эскизные материалы:

- техническое задание на проектирование веб-сайта;
- эскиз модульной сетки со всеми функциональными размерами;
- цветовую схему сайта в словесном описании и в 16-ричных кодах цвета;
- логическую схему системы навигации.

2. Окончательный проект сайта:

- все фалы изображений, используемых в дизайне сайта;
- файл каскадного списка стилей;
- файл, содержащий html-код заглавной страницы.
- файлы, содержащий html-код внутренних страниц.

Задание на курсовую работу оформляется бланком задания (см. приложение 1.) и включает полное наименование учебного курса, индекс группы, фамилию и инициалы студента, получающего задание, название темы курсовой работы, перечень вопросов, требующих проработки, дату выдачи задания и срок защиты курсовой работы. Задание подписывается студентом, принявшим задание, и преподавателем, выдавшим задание.

2.4. Объем курсовой работы

Законченная курсовая работа оформляется в виде отчета, включающего:

- расчетно-пояснительную записку объемом 20 - 40 страниц текста на листах формата А4;
- оффлайновую версию сайта на любом оптическом дисковом носителе.

2.5. Работа над курсовой работой

На начальном этапе студент при помощи руководителя, ведущего курсовую работу, осуществляет выбор тематики проектируемого веб-сайта, производит подбор литературы. Он знакомится с возможными методами решения поставленной задачи, производит сравнительный анализ возможных вариантов решения поставленной задачи, цели и задачи, решаемых в работе. Результаты этой работы излагаются во введении.

На следующем этапе работы рассматривается, производится и описывается процесс разработки сначала эскизных материалов, а затем и окончательного варианта веб-сайта.

2.6. Защита курсовой работы

Защита законченной курсовой работы проводится с демонстрацией презентации, включающей в себя основные ключевые моменты разработки дизайна и скриншоты страниц веб-майта.

При оценке работы учитываются:

- техническая грамотность, соответствие результатов заданию, последовательность решения, лаконичность и эффективность предлагаемых решений;
- эрудиция студента (общая, техническая);
- качество выполненных работ (оформление, грамотность, аккуратность);
- соблюдение сроков, предусмотренных графиком выполнения работы.

Учет всех параметров определяет объективность **оценки защиты работы**:

- "отлично" - безукоризненная по всем пунктам защита работы;
- "хорошо" - недостаточная эффективность решений и некоторые погрешности оформления;
- "удовлетворительно" - нечеткость изложения и некоторое несоблюдение требований к элементам, отсутствие единой концепции, несоблюдение графика выполнения работ;
- "не удовлетворительно" – отсутствие одного или нескольких элементов.

3. МЕТОДИЧЕСКИЕ УКАЗАНИЯ К РАБОТЕ НАД КУРСОВОЙ РАБОТОЙ

3.1. План построения и содержание разделов пояснительной записки к курсовой работе

Примерное распределение материала расчетно-пояснительной записки по разделам в процентах от полного объема приведено в таблице 1.

Таблица 1.

Разделы курсовой работы	Примерный объем
Введение	5 %
Разработка технического задания на проектирование веб-сайта	5 %
Построение эскиза модульной сетки	10 %
Построение цветовой схемы	10 %
Построение системы навигации	5 %
Построение предварительного макета заглавной страницы	10 %
Построение предварительного макета внутренней страницы	10 %
Построение каскадного списка стилей и адаптация макетов заглавной и внутренней страниц	25 %
Создание остальных внутренних страниц	5%
Заполнение страниц контентом	10%
Отладка и тестирование	5%

3.2. Методические указания по выполнению разделов курсовой работы

3.2.1. Основные принципы дизайна.

Основной принцип дизайна заключается в том, что элементы никогда не бывают сами по себе, они всегда связаны в некую композицию, подчиненную некой идее. Перечислим основные принципы графического дизайна:

1) Целесообразность: любой авторский замысел и весь строй произведения должны быть подчинены какой-то цели, идее, художественной задаче. Применительно к веб-дизайну принцип целесообразности также довольно прост - не допускать избыточного количества декоративных элементов, не несущих функциональной нагрузки.

2) Единство: все элементы в композиции должны быть взаимосвязаны, они должны так взаимодействовать друг с другом, чтобы композиция смотрелась единым гармоничным целым. В веб-дизайне принцип единства проявляется на том, что два или более объекта в составе всей композиции имеют нечто общее, что их сближает на фоне других. Приемы, которые следует использовать для достижения единства в композиции:

наложение объектов друг на друга; единство цвета; единство стиля; единство текстуры; единство формы; выравнивание элементов, уменьшение количества вертикалей и горизонталей.

3) Равновесие: условие устойчивости зрительной композиции, это состояние, когда все элементы сбалансированы между собой.

4) Доминанта: наличие в композиции смыслового центра, где завязывается основное действие, и возникают основные связи. Кроме смыслового центра существует еще и визуальный центр, с него начинается восприятие работы зрителем, именно он первым притягивает взгляд. Иногда смысловой и визуальный центры могут совпадать.

5) Соподчинение: наличие менее и более важных элементов композиции таким образом, чтобы части композиции воспринимались зрителем в определенной последовательности, определенным образом.

6) Динамизм: это управление взглядом и вниманием зрителя, при котором так или иначе, создается движение.

7) Гармония: наличие между всеми элементами работы связи, которая примиряет форму и содержание, предмет и пространство, сводя все воедино.

По типу дизайна сайты классифицируются следующим образом.

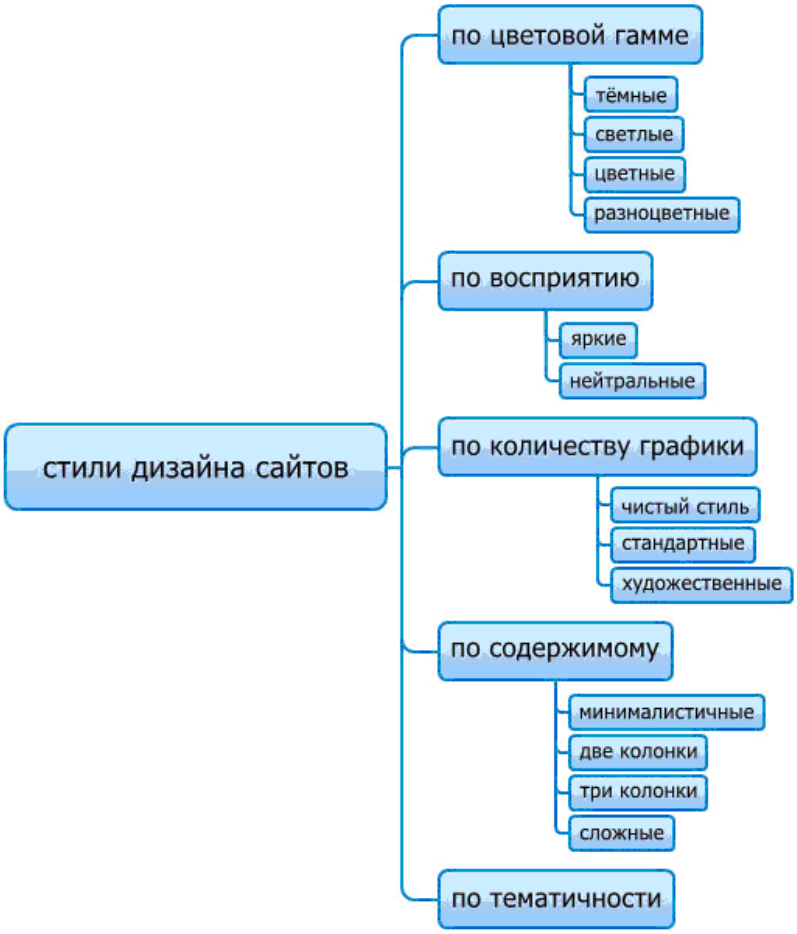


Рисунок 1. Визуальная классификация стилей дизайна сайтов.

Далее приведено краткое описание обозначенных на схеме стилей.

Темный стиль дизайна сайтов – к данному типу относятся сайты, на которых доминирующие цвета - темные.

Светлый стиль дизайна сайтов – к данному типу относятся сайты, на которых доминирующие цвета - темные.

Цветной стиль дизайна сайта - к данному типу относятся сайты, в дизайне которых преобладает какой-либо хроматический (отличный от белого, черного и серого) цвет.

Разноцветный стиль дизайна сайтов - к данному типу относятся сайты, в дизайне которых используется два или более цветов, имеющих большой визуальный вес.

Яркий стиль - к данному типу относятся сайты, в дизайне которых использованы чистые основные и бинарные цвета.

Нейтральный стиль - к данному типу относятся сайты, в дизайне которых использованы триадные цвета с ахроматической компонентой.

Чистый стиль - к данному типу относятся сайты, в дизайне которых количество графики, не являющейся содержимым сайта, минимально.

Стандартный стиль - в этом случае неинформативная графика используется в только в некоторых элементах дизайна, как правило - в шапке сайта.

Художественный стиль - графика является основным, доминирующим элементом дизайна.

Минималистичная структура содержимого сайта - акцент делается на каком-либо одном, главном объекте.

Двухколоночная и трехколоночная структура дизайна сайта - это две наиболее распространенные структуры.

Сложная структура – это все то, что не вписывается в предыдущие стандартные модели.

Исходя из основных целей и задач, а так же из того, каким типам дизайна будет относиться проектируемый сайт, заполняется бланк технического задания (см. приложение 4).

3.2.2. Построение логической структуры сайта.

У сайтов с ***линейной структурой*** страницы идут одна за другой, и пользователь должен просматривать их как слайд-шоу.

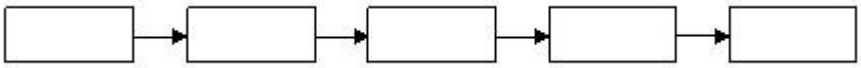


Рисунок 2. Линейная структура.

На сайтах, построенных по принципу ***линейной структуры с альтернативами и вариантами***, посетители могут проявить некоторую инициативу.

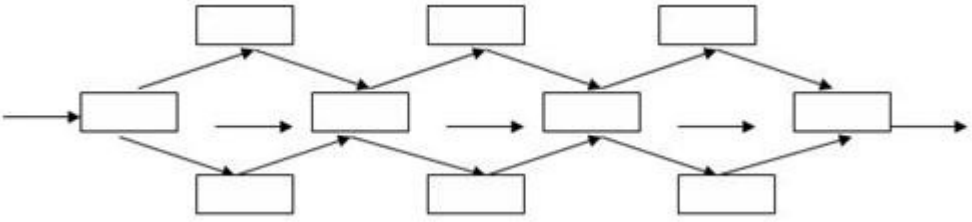


Рисунок 3. Линейная структура с альтернативами.

Линейная структура с ответвлениями - это контролируемая структура, которая напоминает дорогу с ответвляющимися от нее время от времени тупиковыми тропинками .

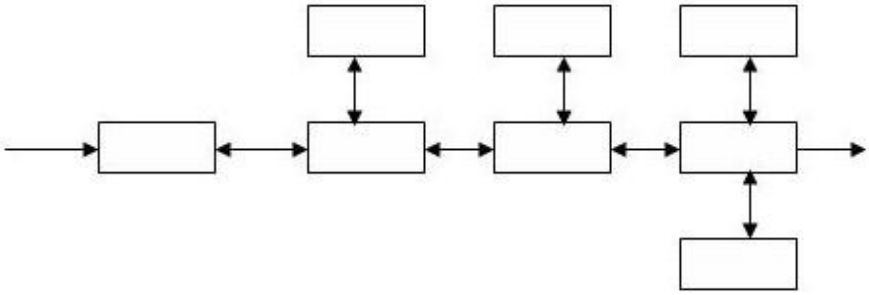


Рисунок 4. Линейная структура с ответвлениями.

Древовидная структура - самый универсальный способ размещения веб-страниц. Она подходит для создания практически любых типов сайтов. В древовидной структуре очень сложно соблюдать баланс между глубиной и шириной.

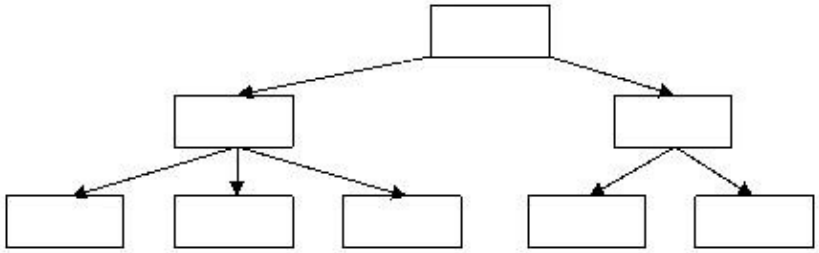


Рисунок 5. Древовидная структура.

В разветвленной структуре все страницы также размещаются в различных ветках, но у пользователя есть возможность перемещаться по ним не только вертикально (вверх-вниз) но и горизонтально (то есть между ветками на разных уровнях).

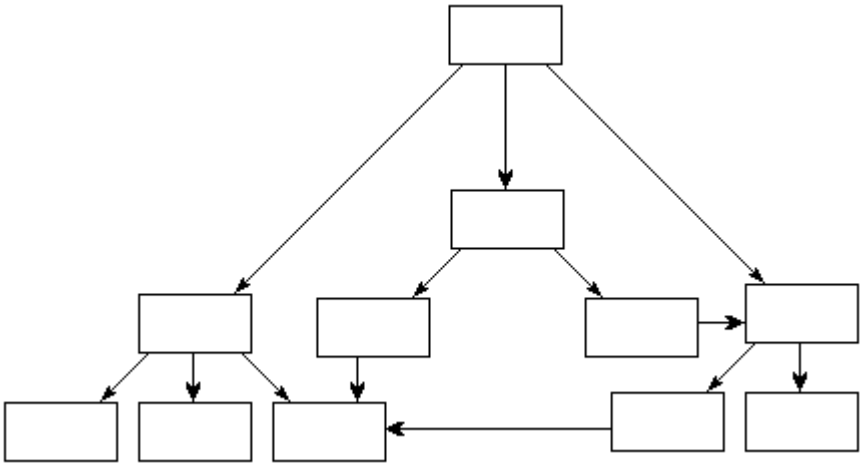


Рисунок 6. Разветвленная структура.

Правило двух кликов - каждый пользователь должен иметь возможность попасть с одной страницы ресурса на любую другую, совершив всего два клика. При этом неважно насколько глубоко вглубь каталога он ушёл. Соблюдение этого, простого на первый взгляд , правила является одной из первоочередных задач стоящих перед дизайнером и разработчиком при изготовлении сайта. Его соблюдение позволяет любому пользователю не плутать по сайту, а попасть в нужный ему раздел практически моментально, что позволяет сохранить внимание клиента и направить его в нужный раздел.

С некоторой долей условности ***логически обусловленные переходы*** можно разделить на следующие группы:

1) Переходы к содержательно соотносимым страницам - пары страниц могут содержать содержательно соотносимую информацию, когда развитие, раскрытие какого-то аспекта темы страницы представлено на другой.

2) Переходы к информационно значимым страницам - группа страниц может быть информационно значимой и содержать информацию, необходимость в которой может возникнуть при просмотре прочих страниц сайта. Гиперссылки, направленные на информационно значимые страницы, обеспечивают возможность прервать просмотр сайта и обратиться к этим страницам или к исходной странице навигации (например, к главной странице сайта).

3) Переходы между страницами, связанными логическими отношениями следования и/или иерархичности - между страницами могут иметь место отношениям следования или уровней иерархии. Гиперссылки между такими страницами обеспечивают переходы между страницами разных иерархических уровней (от верхней к нижней и/или наоборот) и между последовательными страницами (к следующей и/или к предыдущей).

3.2.3. Построение модульной сетки сайта.

Веб-страница фактически рассматривается как набор прямоугольных блоков, которые выкладываются в определенном порядке. При этом, как правило, данные располагаются по колонкам, поэтому при верстке применяют термин одно-, двух-, трехколонный макет и т.д.

При этом обычно вместо текста и рисунков применяют схематические значки. Например, текст обозначается несколькими горизонтальными линиями, а рисунки изображаются затемненными блоками или перечеркнутыми прямоугольниками .

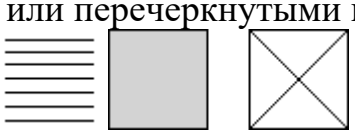


Рисунок 7. Обозначение текста и изображений в макетах.

Одноколонная сетка чаще всего встречается в академическом дизайне, при фиксированном макете и публикации большого текста.

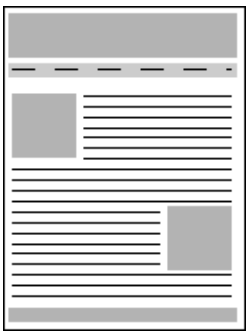


Рисунок 8. Одноколонная модульная сетка

Двухколонная сетка **-** это один из самых распространенных вариантов при использовании на сайтах. При такой модульной сетке используется две колонки - одна отводится под основной текст, а вторая используется для навигации и другой полезной информации. Принципиального значения не имеет, слева или справа располагается колонка с навигацией, встречается и тот и другой вариант.

Рисунок 9. Двухколонная модульная сетка.

Трехколонная сетка часто применяется на если одновременно требуется показать пользователю множество возможностей, которые он обнаружит на сайте. Одна из колонок отдается под навигацию, вторая, самая широкая - под основной текст, а в третью колонку добавляют рекламу, ссылки, текст и т.д. Трехколонная сетка обеспечивает больше простора для дизайна, ведь в некоторых местах можно объединять колонки, разбивать материал на отдельные фрагменты и визуально отделять один блок от другого.

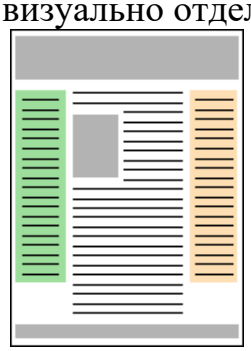


Рисунок 10. Трехколонная модульная сетка.

Золотое сечение - деление отрезка на части в таком соотношении, при котором большая часть относится к меньшей, как сумма к большей.

На рисунке демонстрируется принцип Золотого сечения для двухколоночного макета.

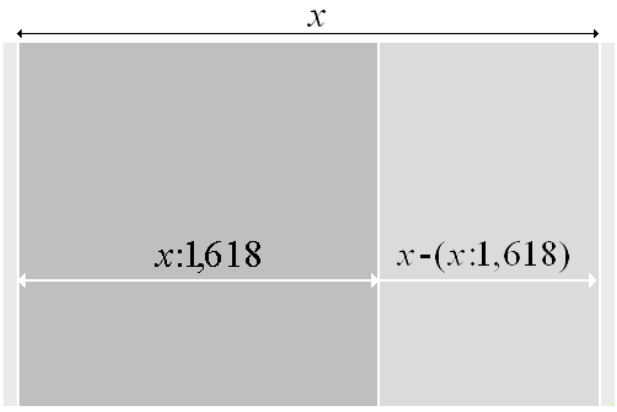


Рисунок 11. Принцип золотого сечения.

Правило третей -это принцип построения композиции, основанный на упрощенном правиле золотого сечения.

При определении зрительных центров комопзиция, как правило, делится линиями, параллельными его сторонам, в пропорциях 3:5, 2:3 или 1:2 (берутся последовательно идущие числа Фибоначчи). Последний вариант дает деление изображения на три равные части (трети) вдоль каждой из сторон. На пересечениях выделяется самая важная информация.

Несмотря на заметное отличие положения центров внимания, полученных по правилу третей, от золотого сечения, технологическая простота и наглядность сделали эту схему композиции популярной. Чаще всего, невозможно, да и просто бесполезно использовать все четыре точки, для выделения на первый план самых важных деталей макета сайта. Но, не менее успешно использовать одну или две точки, как кстати обычно и делают. ***В веб-дизайне левый верхний угол обычно самый «сильный»***, так как пользователи просматривают сайт по принципу формы буквы «F». Именно поэтому в данной позиции чаще всего размещают логотип.

Можно говорить о существовании ***3-х видов html-верстки****:* фиксированной, резиновой и гибридной. Иногда выделяют и четвертый вид–- эластичная верстка сайта (относительные размеры элементов веб-страницы по отношению к размеру шрифта).

Жесткая (фиксированная) верстка отличается четко определенными размерами элементов - элементы веб-страницы соответственно имеют фиксированные размеры, в пикселях.

Мягкая (резиновая) верстка отличается от фиксированной верстки относительными размерами элементов веб-страницы (процентное отношение), все элементы соответственно имеют относительные размеры, в зависимости от размеров своих родительских элементов. В отличие от фиксированной верстки, резиновая верстка позволяет адаптировать шаблон под любой тип и разрешение монитора, выводя элементы страницы в процентном отношении к размеру монитора.

Гибридная верстка **-** это наиболее распространенный вид верстки который включает в себя как фиксированные, так и относительные размеры элементов, некий компромисс между фиксированной и резиновой верстской.

Таблицы с невидимой границей долгое время использовались для верстки веб-страниц, позволяя разделять документ на модульные блоки. Подобный способ применения таблиц нашел воплощение на многих сайтах, пока ему на смену не пришел более современный способ верстки с помощью слоев.

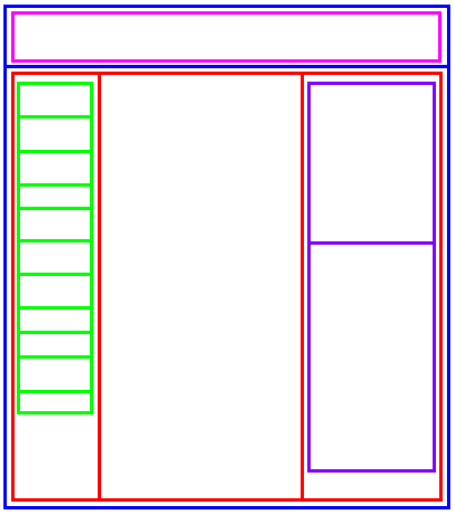


Рисунок 12. Избыточное использование таблиц – верстка «матрешками».

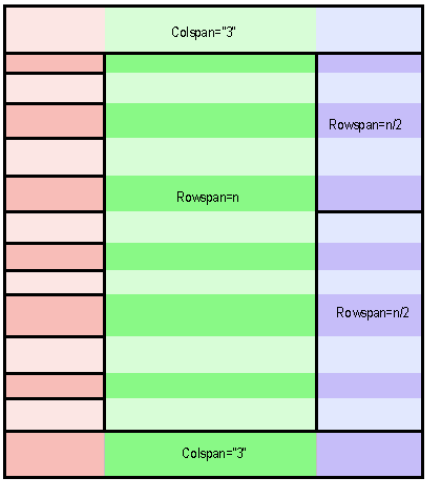


Рисунок 13. Табличная верстка с объединением ячеек.

3.2.4. Построение цветовой схемы сайта.

Цветовая гармония в дизайне представляет собой ***определенное сочетание цветов с учетом всех их основных характеристик***, таких как

- цветового тона;
- светлоты;
- насыщенности;
- формы;
- размеров занимаемых этими цветами на плоскости, их взаимного расположения в пространстве, которое приводит к цветовому единству и наиболее благоприятно эстетически воздействует на человека.

Для построения гармоний используется ***треугольник цвета***.

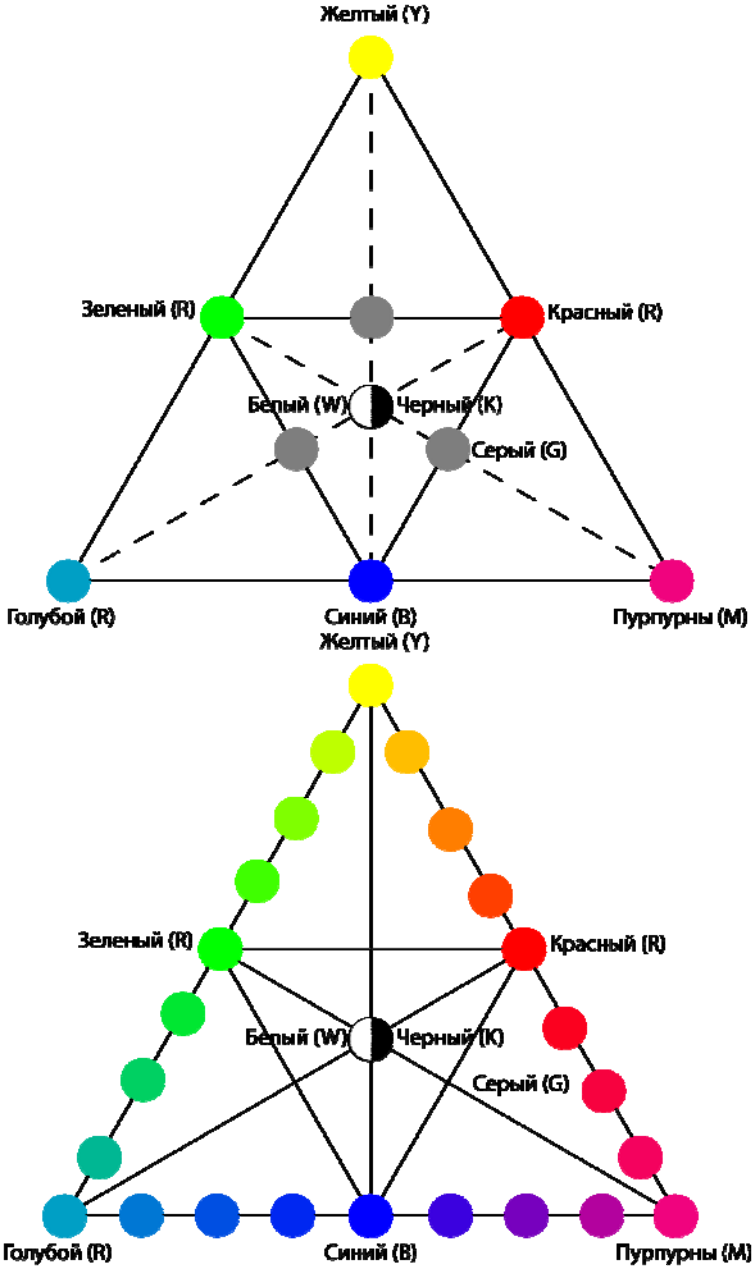


Рисунок 14. Треугольник цвета

Монохромная гармония строится на одном хроматическом цвете и всевозможных его оттенках, которые будут светлее основного, если разбавлять его с белым в разных пропорциях, или темнее, если к основному цвету добавлять черный.

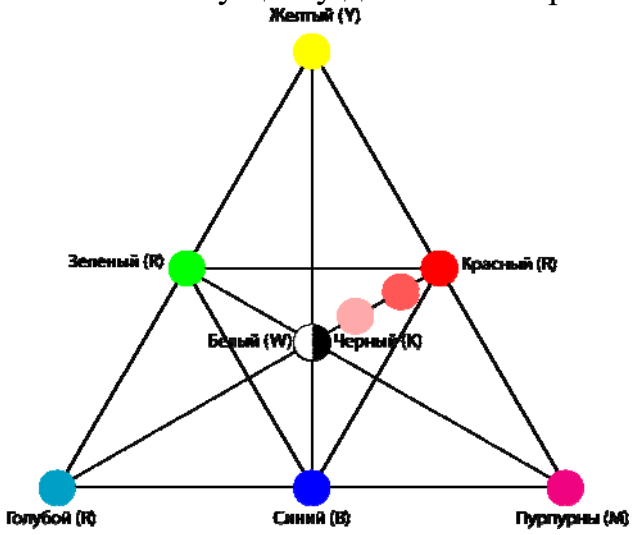


Рисунок 15. Монохромная гармония.

Аналоговая гармония - это три оттенка, на цветовом круге идущих друг за другом. Если трех разных оттенков мало, можно использовать в дополнение к ним ахроматические цвета.

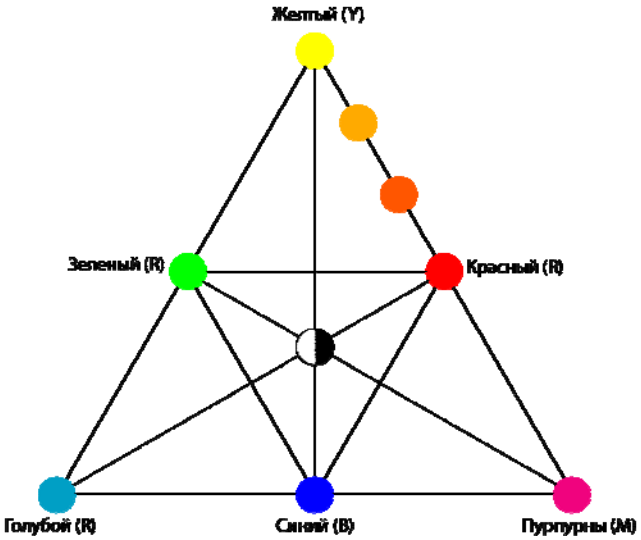


Рисунок 16. Аналоговая гармония.

В ***комплементарной гармонии*** находятся любые два цвета, расположенные друг напротив друга на цветовом круге или треугольнике. Эти цвета находятся в максимальном контрасте по оттенку, и если при этом они равны по яркости и в дизайне используются наравне, есть риск разрушить композицию. Лучше, если один из цветов более приглушенный, менее яркий или менее насыщенный. Комплементарные пары, как правило, используются для создания яркого, динамичного дизайна, особенно это касается сайтов спортивной тематики. Разумеется, два цвета - недостаточно. Следует дополнять набор более светлыми или более темными оттенками основных цветов.

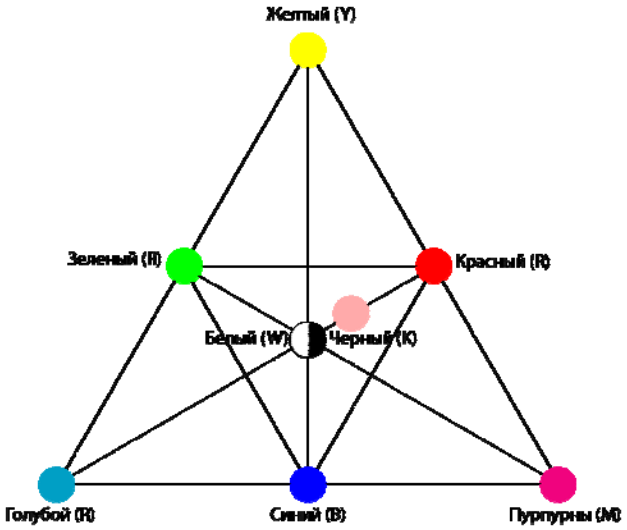


Рисунок 17. Комплементарная гармония.

В ***сплит-комплементарной гармонии*** при выборе основного цвета в дополнение к нему выбирается не противоположный, а два соседних противоположному. Такой выбор делает гармонию менее агрессивной и не такой бескомпромиссной.

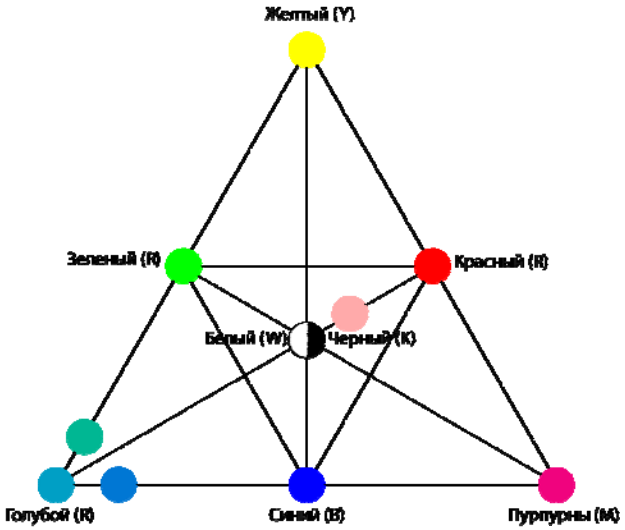


Рисунок 18. Сплит-комплементарная гармония.

Аналого-комплементарная гармония вобрала в себя аналоговую и комплементарную гармонии. Суть ее в том, что к набору цветов из аналоговой гармонии присоединяется цвет, противоположный среднему из аналогового набора.

Цвета ***триадной гармонии*** лежат на вершинах равнобедренного треугольника, помещенного внутрь цветового круга.

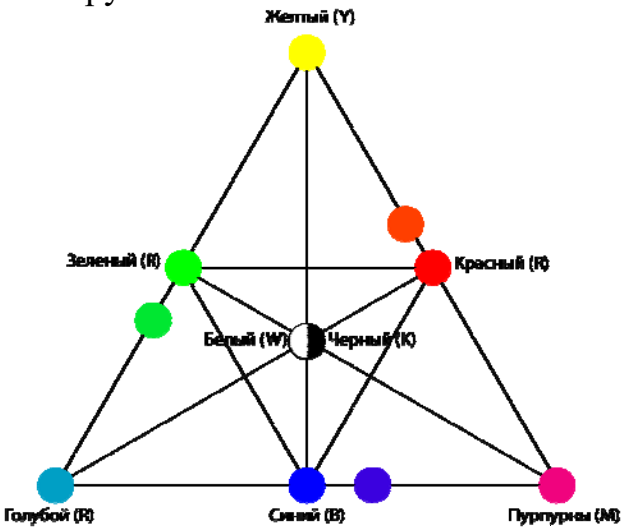


Рисунок 19. Триадная гармония.

При ***тетрадной гармонии*** выбор цветов в этом случае зависит от того, какую фигура будет вписана в цветовой круг - квадрат или прямоугольник.

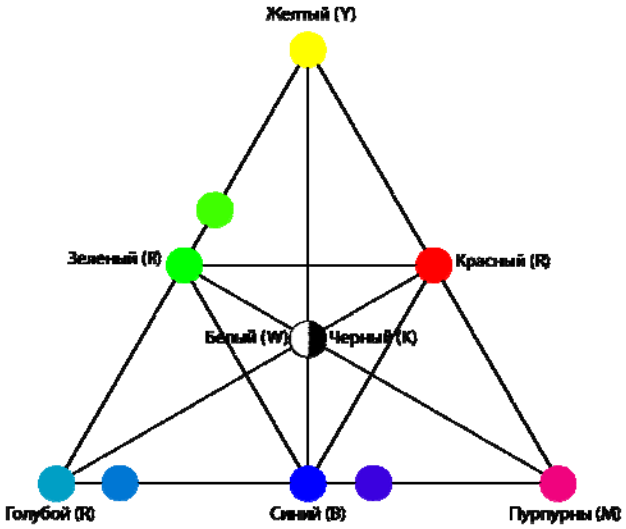


Рисунок 20. Тетрадная гармония.

Сейчас в Интернете существует целый ряд сервисов подбора цвета, которые помогаю на основе определенного набора базовых цветов построить тот или иной вид гармонии для сайта. В частности – это **Color Scheme Designer** (http://colorschemedesigner.com/**).**

Очень часто этапе разработки цветовой схемы сайта веб-сайта возникает следующая проблема: трактовка словесных названий цветов неоднозначна и строго индивидуальна для каждого человека. В веб-дизайне, как правило, из всех методов обозначения цветов используется ***шестнадцатеричный код цвета*** в терминах формата RGB.

3.2.5. Базовая структура html-документа.

Как правило, заглавная страница сайта носит название ***index.html***. HTML-документ можно создавать и редактировать в любом текстовом редакторе. Для просмотра документа подойдет любой браузер.

Чтобы браузер при отображении документа понимал, что имеет дело не с простым текстом, а с элементом форматирования и применяются теги. ***Общий синтаксис написания тегов*** следующий.

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

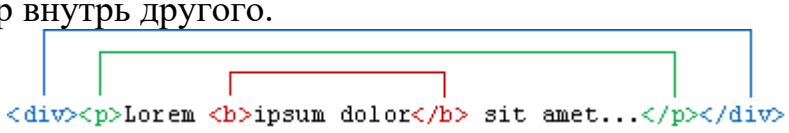
Одиночный тег используется самостоятельно, а ***парный*** может включать внутри себя другие теги или текст.

```
<meta http-equiv="content-type">
<p>Lorem ipsum dolor sit amet consectetuer</p>
```

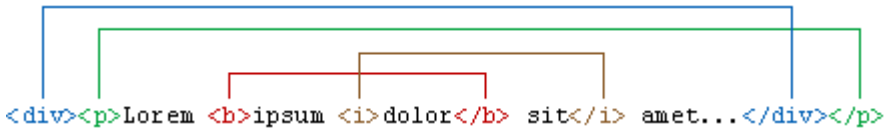
У тегов допустимы различные атрибуты, которые разделяются между собой пробелом. Есть теги без всяких дополнительных атрибутов.

```
<u>Меню</u>
<td colspan="2" bgcolor="#D3EDF6" align="center">
Ячейка 1</td>
```

Парные теги, называемые по-другому ***контейнеры***, состоят из двух частей - открывающий и закрывающий тег. На рисунке демонстрируется, как можно и нельзя добавлять один контейнер внутрь другого.



а)



б)

Рисунок 21. Вложение тегов, а -правильное, б -неверное

Все значения атрибутов тегов следует указывать в двойных или одинарных кавычках.

Базовая структура html-документа выглядит следующим образом:

```
<!DOCTYPE html PUBLIC
"-//W3C//DTD XHTML 1.0 Transitional//EN"
http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
  <head>
    ...
  </head>
  <body>
    ...
  </body>
</html>
```

Тег <!DOCTYPE> предназначен для указания типа текущего документа -DTD (document type definition, описание типа документа).

Синтаксис:

`<!DOCTYPE [Элемент верхнего уровня] [Публичность] "[Регистрация]//[Организация]//[Тип] [Имя]//[Язык]" "[URL]">`

Атрибуты:

`Элемент верхнего уровня = Html` -указывает элемент верхнего уровня в документе, для HTML это атрибут `html`.

`Публичность = Public | System` -объект является публичным или системным, например, таким как локальный файл. Для HTML/XHTML указывается значение `Public`.

`Регистрация = + | –` -сообщает, что разработчик DTD зарегистрирован в международной организации по стандартизации (International Organization for Standardization, ISO). Принимает одно из двух значений: разработчик зарегистрирован в ISO или разработчик не зарегистрирован.

`Организация = W3C` -уникальное название организации, разработавшей DTD. Официально HTML/XHTML публикует `W3C`, это название и пишется в `<!DOCTYPE>`.

`Тип = DTD` - тип описываемого документа. Для HTML/XHTML значение указывается `DTD`.

`Имя = …` -уникальное имя документа для описания DTD.

`Язык = EN` язык, на котором написан текст для описания объекта. Содержит две буквы, пишется в верхнем регистре. Для документа HTML/XHTML указывается английский язык (`EN`).

`URL = http://.....` – адрес документа с DTD.

Тег <head> предназначен для хранения других элементов, цель которых -помочь браузеру в работе с данными.

Внутри контейнера <head> допускается размещать следующие элементы: <base>, <basefont>, <link>, <meta>, <script>, <style>, <title>.

Синтаксис:

```
<head>
  ...
</head>
```

Тег <base> определеяетя внутри контейнера <head> и инструктирует браузер относительно полного базового адреса текущего документа. Тег <base> предназначен для документов, в которых используется относительный адрес и эти документы могут переноситься в другую папку или даже на другой компьютер без потери связи. Браузер ищет тег <base>, определяет полный адрес документа и корректно загружает его. Второе применение тега <base> -задание целевого окна для всех ссылок на текущей странице.

Синтаксис:

```
<base>
```

Атрибуты:

href = "http://...." - адрес, который должен использоваться для указания полного пути к файлам.

target = "_blank | _self | _parent | _top" - имя окна или фрейма, куда будет загружаться документ, открываемый по ссылке.

Тег <basefont> предназначен для задания шрифта, размера и цвета текста по умолчанию. Указанные значения будут использоваться во всем документе за исключением тега <font>, в котором можно переопределить параметры оформления текста. Допускается использование тега в контейнере <head> или <body>, причем несколько раз. Это позволяет изменять вид шрифта для части документа.

Синтаксис:

```
<basefont>
```

Атрибуты:

color = "название цвета | 16-ричный код цвета" –устанавливает цвет текста.

face = "название гарнитуры шрифта" - определяет гарнитуру шрифта.

size="1 | 2 | 3 | 4 | 5 | 6 | 7 | +1| –1" - задает размер шрифта в условных единицах.

Тег <link> устанавливает связь с внешним документом вроде файла со стилями или со шрифтами. Тег <link> размещается всегда внутри контейнера <head> и не создает ссылку.

Синтаксис:

```
<link>
```

Атрибуты:

charset = "кодировка" - кодировка связываемого документа.

href = "http://..." - путь к связываемому файлу.

media = "all | braille | handheld | print | screen | speech | projection | tty | tv " - определяет устройство, для которого следует применять стилевое оформление.

rel = " stylesheet | alternate " - определяет отношения между текущим документом и файлом, на который делается ссылка.

sizes = "ширинаXвысота" - указывает размер иконок для визуального отображения.

type = "MIME-тип" - MIME-тип данных подключаемого файла.

Тег <style> применяется для определения стилей элементов веб-страницы. Тег <style> необходимо использовать внутри контейнера <head>. Можно задавать более чем один тег <style>.

Синтаксис:

```
<head>
  <style type="text/css">
   ...
  </style>
</head>
```

Атрибуты:

media - определяет устройство вывода, для работы с которым предназначена таблица стилей.

type="text/css" - сообщает браузеру, какой синтаксис использовать, чтобы правильно интерпретировать стили.

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>.

Синтаксис:

```
<head>
  <title>Заголовок</title>
</head>
```

Метаинформация - информация о способах и методах переработки информации или о том, где найти информацию.

Тег <meta> определяет метатеги, которые используются для хранения информации предназначенной для браузеров и поисковых систем. Например, механизмы поисковых систем обращаются к метатегам для получения описания сайта, ключевых слов и других данных. Разрешается использовать более чем один метатег, все они размещаются в контейнере <head>. Как правило, атрибуты любого метатега сводятся к парам «имя=значение».

Синтаксис:

```
HTML <head>
  <meta content="...">
</head>
```

Атрибуты:

charset - задает кодировку документа.

content - устанавливает значение атрибута, заданного с помощью name или http-equiv.

http-equiv - предназначен для конвертирования метатега в заголовок HTTP.

name - имя метатега, также косвенно устанавливает его предназначение.

С помощью метатегов веб-мастера могут предоставлять поисковым системам информацию о своих сайтах. Метатеги можно использовать для предоставления сведений

самым разным клиентам, и каждая система обрабатывает метатеги только заданного формата, игнорируя остальные.

Тег <body> предназначен для хранения содержания веб-страницы (контента), отображаемого в окне браузера, то есть того, которое будет визуализировано для пользователя. Информацию, которую следует выводить в документе, следует располагать именно внутри контейнера <body>. К такой информации относится текст, изображения, теги, скрипты JavaScript и т.д.

Тег <body> также применяется для определения цветов ссылок и текста на веб-странице. Подобная практика в HTML 4 осуждается и взамен для указания цветовой схемы рекомендуется использовать стили, применяя их к селектору BODY. Тем не менее, большинство атрибутов до сих пор поддерживается разными браузерами.

Часто тег <body> используется для размещения обработчика событий, например, onload, которое выполняется после того, как документ завершил загрузку в текущее окно или фрейм.

Синтаксис:

```
<body>
  ...
</body>
```

Атрибуты:

alink = "название цвета | 16-ричный код цвета" - устанавливает цвет активной ссылки.

background =" http:// " - задает адрес фонового рисунка веб-страницы

bgcolor = "название цвета | 16-ричный код цвета" - цвет фона веб-страницы.

bgproperties ="fixed" - определяет, прокручивать фон совместно с текстом или нет. Если требуется, чтобы фон не фиксировался, следует удалить атрибут.

link = "название цвета | 16-ричный код цвета" - цвет ссылок на веб-странице.

scroll ="yes | no" - устанавливает, отображать полосы прокрутки или нет.

text = "название цвета | 16-ричный код цвета" - цвет текста в документе.

vlink = "название цвета | 16-ричный код цвета" - цвет посещенных ссылок.

3.2.6. Таблицы.

Для добавления таблицы на веб-страницу используется тег <table>. Этот элемент служит контейнером для элементов, определяющих содержимое таблицы. Любая таблица состоит из строк и ячеек. Таблица должна содержать хотя бы одну ячейку.

Тег <table> служит контейнером для элементов, определяющих содержимое таблицы. Любая таблица состоит из строк и ячеек, которые задаются с помощью тегов <tr> и <td>. Внутри <table> допустимо использовать следующие элементы: <caption>, <col>, <colgroup>, <tbody>, <td>, <tfoot>, <th>, <thead> и <tr>.

Атрибуты:

align="left | center | right" - определяет выравнивание таблицы.

background = "http://...." - задает фоновый рисунок в таблице.

bgcolor = "название цвета | 16-ричный код цвета" - цвет фона таблицы.

border ="целое положительное число в пикселях"- толщина рамки в пикселах.

bordercolor = "название цвета | 16-ричный код цвета" - цвет рамки.

cellpadding = "целое значение в пикселях | целое число в процентах от доступного пространства" - отступ от рамки до содержимого ячейки.

cellspacing ="целое положительное число" - расстояние между ячейками.

cols ="целое положительное число" - число колонок в таблице.

frame=" void |border | above |below | hsides | vsides | rhs | lhs " - сообщает браузеру, как отображать границы вокруг таблицы.

height = "целое значение в пикселях | целое число в процентах от доступного пространства" - высота таблицы.

rules =" all | groups | cols | none | rows" - сообщает браузеру, где отображать границы между ячейками.

summary ="текст" - краткое описание таблицы.

width = "целое значение в пикселях | целое число в процентах от доступного пространства" - ширина таблицы.

Тег <tr> служит контейнером для создания строки таблицы. Каждая ячейка в пределах такой строки может задаваться с помощью тега < th> или <td>.

Атрибуты:

align = "left | center | right | justify | char" - определяет выравнивание содержимого ячеек по горизонтали.

bgcolor = "название цвета | 16-ричный код цвета" - цвет фона ячеек.

bordercolor = "название цвета | 16-ричный код цвета" - цвет рамки.

char = "текстовый символ" - выравнивание содержимого ячеек относительно заданного символа.

charoff = "любое целое число" - смещение содержимого ячеек относительно указанного символа.

valign = "top | middle | bottom | baseline" - выравнивание содержимого ячеек по вертикали.

Тег <td> Предназначен для создания одной ячейки таблицы. Тег <td> должен размещаться внутри контейнера <tr>, который в свою очередь располагается внутри тега <table>.

Атрибуты:

abbr ="текст" - краткое описание содержимого ячейки.

align = "left | center | right | justify | char" - определяет выравнивание содержимого ячейки по горизонтали.

axis ="текст" - группирует ячейки связанные между собой похожей информацией. Значение атрибута - любой текст, представляющий собой произвольное имя ячейки. Для связывания ячеек между собой значения атрибута axis должны совпадать.

background = "http://...." - задает фоновый рисунок в ячейке.

bgcolor = "название цвета | 16-ричный код цвета" - цвет фона ячейки.

bordercolor = "название цвета | 16-ричный код цвета" - цвет рамки.

char = "текстовый символ" - выравнивание содержимого ячеек относительно заданного символа.

charoff = "любое целое число" - смещение содержимого ячеек относительно указанного символа.

colspan = "целое положительное число больше 1" - объединяет горизонтальные ячейки.

height = "целое значение в пикселях | целое число в процентах от доступного пространства" - высота ячейки.

nowrap - запрещает перенос строк.

rowspan = "целое положительное число больше 1" - объединяет вертикальные ячейки.

valign = "top | middle | bottom | baseline" - выравнивание содержимого ячейки по вертикали.

width = "целое значение в пикселях | целое число в процентах от доступного пространства" - ширина ячейки.

Тег <th> предназначен для создания одной ячейки таблицы, которая обозначается как заголовочная. Текст в такой ячейке отображается браузером обычно жирным шрифтом и выравнивается по центру. Тег <th> должен размещаться внутри контейнера <tr>, который в свою очередь располагается внутри тега <table>.

Атрибуты: аналогично тегу <td>.

Синтаксис:

```
<table>
  <tr>
    <td>...</td>
  </tr>
</table>
```

Для объединения двух и более ячеек в одну используются атрибуты colspan и rowspan тега <td>. Атрибут colspan устанавливает число ячеек объединяемых по горизонтали. Аналогично работает и атрибут rowspan, с тем лишь отличием, что объединяет ячейки по вертикали. Перед добавлением ***следует проверять число ячеек в каждой строке***, чтобы не возникло ошибок. Если число ячеек в каждой строке не будет совпадать, появятся пустые **фантомные ячей**ки.

После того, как таблица макета сформирована, следует заверстать в нее основные элементы сайта – текст и изображения.

Для изменения вида текста существует достаточно большое количество различных тегов. Это и немудрено, ведь текст самый популярный вид информации.

Прежде чем редактировать код веб-страницы, следует принять во внимание некоторые особенности, которые присущи HTML при работе с текстом.

1) Любое количество пробелов идущих подряд, в браузере отображается как один. Сколько бы ни было пробелов между словами, это никак не повлияет на конечный вид текста. Это же правило относится к символам табуляции и переносу текста. Исключением из этого правила является тег <pre>, внутри которого любое число пробелов отображается именно так, как оно указано в коде.

2) HTML не поддерживает расстановку переносов в словах, как это делают текстовые редакторы, иначе говоря, все слова пишутся целиком без их разбиения. Это условие несущественно, пока не используется выравнивание текста по ширине. В этом случае блок текста выравнивается по левому и правому краю. Короткие строки при этом растягиваются за счет автоматического добавления пробелов между словами. Иногда пустые блоки между словами настолько велики, что портят внешний вид страницы и ухудшают читабельность текста.

3) Текст занимает ширину окна браузера или отведенного под него элемента макета. Если написать одну длинную строку в коде HTML, то в браузере она будет отформатирована, чтобы текст поместился по ширине в окно. Переносы текста будут добавлены автоматически в местах пробела или дефиса. Если в тексте нет ни того, ни другого символа, браузер не сможет создать переносы и отобразит текст одной строкой. Если она шире окна браузера, то неминуемо появится горизонтальная полоса прокрутки.

Тег <p> определяет текстовый абзац. Тег <p> является блочным элементом, всегда начинается с новой строки, абзацы текста идущие друг за другом разделяются между собой отбивкой. Величиной отбивки можно управлять с помощью стилей. Если закрывающего тега нет, считается, что конец абзаца совпадает с началом следующего блочного элемента.

Синтаксис:

```
<p>Текст</p>
```

Атрибуты:

`align ="left | center | right | justify"` - определяет выравнивание текста.

Тег <img> предназначен для отображения на веб-странице изображений в графическом формате GIF, JPEG или PNG. Этот тег имеет единственный обязательный атрибут src, который определяет адрес файла с картинкой. Если необходимо, то 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>.

Рисунки также могут применяться в качестве карт-изображений, когда картинка содержит активные области, выступающие в качестве ссылок. Такая карта по внешнему виду ничем не отличается от обычного изображения, но при этом оно может быть разбито на невидимые зоны разной формы, где каждая из областей служит ссылкой.

Синтаксис:

```
<img src="URL" alt="альтернативный текст">
```

Атрибуты:

`align="bottom | left | middle | right | top"` - определяет как рисунок будет выравниваться по краю и способ обтекания текстом.

`alt="текст"` - альтернативный текст для изображения. Этот тег предназначен для того, чтобы пользователь даже если изображение не грузится, мог получить информацию о том, что на нем изображено.

border="целое число в пикселях" -толщина рамки вокруг изображения.

height ="целое положительное число в пикселях или процентах" - высота изображения.

hspace="целое положительное число в пикселях" - горизонтальный отступ от изображения до окружающего контента.

ismap - говорит браузеру, что картинка является серверной картой-изображением.

longdesc ="URL" - указывает адрес документа, где содержится аннотация к картинке.

src="URL" - путь к графическому файлу.

vspace="целое положительное число в пикселях" - вертикальный отступ от изображения до окружающего контента.

width="целое положительное число в пикселях или процентах" - ширина изображения.

usemap="#идентификатор" - ссылка на тег <map>, содержащий координаты для клиентской карты-изображения.

Тег <map> служит контейнером для элементов <area>, которые определяют активные области для карт-изображений. Такие области устанавливают невидимые зоны на изображении, являющиеся ссылками на HTML-документы. Цель использования тега <map> - в связывании тега <img> с клиентской картой-изображением. Эта связь определяется применением единого идентификатора как в теге <img>, задаваемого атрибутом usemap, так и в теге <map>, устанавливаемого атрибутом name.

Синтаксис:

```
<map name="имя">
  <area атрибуты>
</map>
```

Атрибуты:

name="идентификатор" - имя карты-изображения.

Элемент <area> определяет активные области изображения, которые являются ссылками. Рисунок с привязанными к нему активными областями называется в совокупности картой-изображением. Такая карта по внешнему виду ничем не отличается от обычного изображения, но при этом оно может быть разбито на невидимые зоны разной формы, где каждая из областей служит ссылкой. Тег <area> задает форму области, ее размеры, устанавливает адрес документа, на который следует сделать ссылку, а также имя окна или фрейма, куда браузер будет загружать документ. Этот тег всегда располагается в контейнере <map>, который связывает координаты областей с изображением.

Несколько областей могут перекрывать друг друга, сверху будет та, которая в коде HTML располагается выше.

Синтаксис:

```
<area href="URL">
```

Атрибуты

accesskey ="клавиша" - переход к области с помощью комбинации клавиш.

alt="текст" - альтернативный текст для области изображения.

coords ="координата 1, координата 2, ..." - координаты активной области.

href="URL" - задает адрес документа, на который следует перейти.

hreflang ="код языка" - указывает язык документа, на который ведет ссылка.

nohref - область без ссылки на другой документ.

shape = "circle | default | poly | rect" - форма области.

tabindex="любое целое положительное число, начиная с нуля" - задает последовательность перехода между элементами с помощью клавиши Tab. Значения выстраиваются последовательно и переход между элементами происходит от меньшего значения к большему.

target="_blank | _self | _parent | _top" - имя окна или фрейма, куда браузер будет загружать документ.

type ="MIME тип" - устанавливает MIME-тип документа, на который ведёт ссылка.

3.2.7. Форматирование текста.

Тег <font> представляет собой контейнер для изменения характеристик шрифта, таких как размер, цвет и гарнитура. Хотя этот тег до сих пор поддерживается всеми браузерами, он считается устаревшим и от его использования рекомендуется отказаться в пользу стилей.

Синтаксис:

```
<font>Текст</font>
```

Атрибуты:

color = "название цвета | 16-ричный код цвета" - устанавливает цвет текста.

face = "название гарнитуры шрифта" - определяет гарнитуру шрифта.

size="1 | 2 | 3 | 4 | 5 | 6 | 7 | +1| -1" - задает размер шрифта в условных единицах.

Тег <b> устанавливает жирное начертание шрифта. Допустимо использовать этот тег совместно с другими тегами, которые определяют начертание текста.

Синтаксис:

```
<b>Текст</b>
```

Тег <i> устанавливает курсивное начертание шрифта. Допустимо использовать этот тег совместно с другими тегами, которые определяют начертание текста.

Синтаксис:

```
<i>Текст</i>
```

Тег <u> Добавляет подчеркивание к тексту. Этот тег осуждается спецификацией HTML, взамен рекомендуется использовать стили.

Синтаксис:

```
<u>Текст</u>
```

Тег <s> отображает текст как перечеркнутый. Этот тег аналогичен тегу <strike>, но в отличие от него имеет сокращенную форму записи подобно тегам <b>, <i> и <u>. Взамен этого тега рекомендуется использовать стили.

Синтаксис:

```
<strike>Текст</strike>
```

Тег <marquee> создает бегущую строку на странице. На самом деле содержимое контейнера <marquee> не ограничивается строками и позволяет перемещать (скролировать) любые элементы веб-страницы - изображения, текст, таблицы, элементы форм и

т.д. Перемещение можно задать не только по горизонтали, но и вертикали, в этом случае указываются размеры области, в которой будет происходить движение. Первоначально тег <marquee> был предназначен только для браузера Internet Explorer, но современные версии других браузеров также понимают и поддерживают этот тег. Этот тег не входит в спецификацию HTML.

Синтаксис:

```
<marquee>...</marquee>
```

***Атрибуты*:**

behavior ="alternate | scroll | slide" - задает тип движения содержимого контейнера <marquee>.

bgcolor = "название цвета | 16-ричный код цвета" - задает цвет фона под содержимым элемента.

direction="down | left | right | up" - указывает направление движения содержимого контейнера <marquee>.

height="целое положительное число в пикселях | в процентах" - высота области прокрутки.

hspace="целое положительное число в пикселях" - горизонтальные поля вокруг контента.

loop="число повторений" - задает, сколько раз будет прокручиваться содержимое.

scrollamount="целое положительное число в пикселях" - скорость движения контента.

scrolldelay="целое положительное число в миллисекундах" - величина задержки в миллисекундах между движениями.

truespeed - отменяет встроенный ограничитель скорости при низких значениях атрибута scrolldelay.

vspace="целое положительное число в пикселях" - вертикальные поля вокруг содержимого.

width="целое положительное число в пикселях | в процентах" - ширина области прокрутки.

Тег <tt> отображает текст моноширинным шрифтом.

Синтаксис:

```
<tt>Текст</tt>
```

Тег <Hx>. HTML предлагает шесть заголовков разного уровня, которые показывают относительную важность секции, расположенной после заголовка. Так, тег <h1> представляет собой наиболее важный заголовок первого уровня, а тег <h6> служит для обозначения заголовка шестого уровня и является наименее значительным. По умолчанию, заголовок первого уровня отображается самым крупным шрифтом жирного начертания, заголовки последующего уровня по размеру меньше. Теги <h1>,...,<h6> относятся к блочным элементам, они всегда начинаются с новой строки, а после них другие элементы отображаются на следующей строке. Кроме того, перед заголовком и после него добавляется пустое пространство.

Синтаксис:

```
<h1>Заголовок первого уровня</h1>
```

Атрибуты:

align ="left | center | right | justify" - определяет выравнивание заголовка.

Тег <big> увеличивает размер шрифта на единицу по сравнению с обычным текстом. В HTML размер шрифта измеряется в условных единицах от 1 до 7, средний размер текста, используемый по умолчанию, принят 3. Таким образом, добавление тега <big> увеличивает текст на одну условную единицу. Допускается применение вложенных тегов <big>, при этом размер шрифта будет больше с каждым уровнем.

На размер шрифта влияет не только заданный атрибут size тега <font>, но и выбор гарнитуры шрифта. Шрифт Arial выглядит крупнее, чем шрифт Times, а шрифт Verdana чуть больше шрифта Arial.

Синтаксис:

```
<big>Текст</big>
```

Тег <small> уменьшает размер шрифта на единицу по сравнению с обычным текстом.

Синтаксис:

```
<small>Текст</small>
```

Тег <sub> отображает шрифт в виде нижнего индекса. Текст при этом располагается ниже базовой линии остальных символов строки и уменьшенного размера.

Синтаксис:

```
<sub>Текст</sub>
```

Тег <sup> отображает шрифт в виде верхнего индекса. Шрифт при этом отображается выше базовой линии текста и уменьшенного размера.

Синтаксис:

```
<sup>Текст</sup>
```

Тег <hr> рисует горизонтальную линию, которая по своему виду зависит от используемых параметров, а также браузера. Тег <hr> относится к блочным элементам, линия всегда начинается с новой строки, а после нее все элементы отображаются на следующей строке.

Синтаксис:

```
<hr>
```

Атрибуты:

align ="left | center | right" - определяет выравнивание линии.

color = "название цвета | 16-ричный код цвета" - цвет линии.

noshade - рисует линию без трехмерных эффектов.

size ="целое положительное число в пикселях" - толщина линии.

width="целое положительное число в процентах" - ширина линии.

Тег <blockquote> предназначен для выделения длинных цитат внутри документа. Текст, обозначенный этим тегом, традиционно отображается как выровненный блок с отступами слева и справа (примерно по 40 пикселов), а также с отбивкой сверху и снизу.

Синтаксис:

```
<blockquote>Текст</blockquote>
```

Тег <cite> помечает текст как цитату или сноску на другой материал. Такое выделение удобно для изменения стиля текста через CSS, а также применяется для разделения кода HTML на структурные элементы. Браузеры обычно устанавливают текст внутри контейнера <cite> курсивом.

Синтаксис:

```
<cite>Текст</cite>
```

Тег <dfn> . Как правило, в документе, когда упоминается новый термин, он выделяется курсивом и дается его определение. При использовании этого термина в дальнейшем, он считается уже известным читателю. Тег <dfn> применяется для выделения таких терминов при их первом появлении в тексте.

Браузеры отображают содержимое контейнера <dfn> с помощью курсивного начертания.

Синтаксис:

```
<dfn>Текст</dfn>
```

***Тег *** предназначен для акцентирования текста. Браузеры отображают такой текст курсивным начертанием.

Синтаксис:

```
<em>Текст</em>
```

***Тег *** предназначен для акцентирования текста. Браузеры отображают такой текст жирным начертанием.

Синтаксис:

```
<strong>Текст</strong>
```

Для отображения символов, которых нет на клавиатуре применяются специальные знаки, начинающиеся с амперсанда (&) и заканчивающиеся точкой с запятой (;).

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.

Маркированный список определяется тем, что перед каждым элементом списка добавляется небольшой маркер, обычно в виде закрашенного кружка. Сам список формируется с помощью контейнера <ul>, а каждый пункт списка начинается с тега <li>, как показано ниже. Отступы сверху, снизу и слева от списка добавляются автоматически.

Синтаксис:

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

Атрибуты <ul>:

`type="disc | circle | square"` - устанавливает вид маркера списка.

Атрибуты <li>:

`type="disc | circle | square"` - устанавливает вид маркера списка.

Нумерованный список представляет собой набор элементов с их порядковыми номерами. Вид и тип нумерации зависит от атрибутов тега <ol>, который и применяется для создания списка. Каждый пункт нумерованного списка обозначается тегом <li>, как показано ниже.

Синтаксис:

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

Атрибуты <ol>:

`type="A | a | I | i | 1"` - устанавливает вид маркера списка.

`reversed` - нумерация в списке становится по убыванию (3,2,1).

`start ="целое положительное число"` - задаёт число, с которого будет начинаться нумерованный список.

Атрибуты <li>:

`type="disc | circle | square"` - устанавливает вид маркера списка.

`value="целое положительное число"` - число, с которого будет начинаться нумерованный список.

Список определений состоит из двух элементов - термина и его определения. Сам список задается с помощью контейнера `<dl>`, термин - тегом `<dt>`, а его определение - с помощью тега `<dd>`.

Синтаксис:

```
<dl>
  <dt>Термин 1</dt>
    <dd>Определение 1</dd>
  <dt>Термин 2</dt>
    <dd>Определение 2</dd>
</dl>
```

3.2.8. Гиперссылки.

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 совсем 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.

Тег <a> является одним из важных элементов HTML и предназначен для создания ссылок. В зависимости от присутствия атрибутов `name` или `href` тег `<a>` устанавливает ссылку или якорь. Для создания ссылки необходимо сообщить браузеру, что является ссылкой, а также указать адрес документа, на который следует сделать ссылку. В качестве значения атрибута `href` используется адрес документа (`URL`, Universal Resource Locator, универсальный указатель ресурсов), на который происходит переход.

Синтаксис:

```
<a href="URL">...</a>
<a name="идентификатор">...</a>
```

Атрибуты:

`accesskey = "Цифра (0-9) | латинская буква (a-z)"` - позволяет активировать ссылку с помощью некоторого сочетания клавиш с заданной в коде ссылки буквой или цифрой. Браузеры при этом используют различные комбинации клавиш. Например, для accesskey="s" работают следующие сочетания: Internet Explorer: Alt + S; Chrome: Alt + S; Opera: Shift + Esc, S; Safari: Alt + S; Firefox: Shift + Alt + S.

`charset="utf-8 | windows-1251 | ..."` - указывает кодировку текста, на который ведет ссылка.

coords="координаты" - устанавливает координаты активной области. Набор координат определяется формой «горячей области», которая задается атрибутом shape. Отсчет координат обычно ведется от левого верхнего угла объекта или изображения и указывается в пикселах. Для прямоугольника (shape="rect") определяется четыре координаты - X1, Y1, X2, Y2. Для окружности (shape="circle") определяется три координаты - координаты центра окружности (X, Y) и ее радиус (R). Для полигона (многоугольника) (shape="poly") последовательно указываются координаты каждой вершины (X1, Y1, X2, Y2, …).

href ="Http://…" - задает адрес документа, на который следует перейти.

hreflang=" en | rus | ..." - идентифицирует язык текста по ссылке.

media = "all | braille | handheld | print | screen | speech | projection | tty | tv " - указывает тип носителя документа, на который ведёт ссылка.

name="любой текст с учетом регистра" - устанавливает имя якоря внутри документа.

rel = " answer | chapter | co-worker | colleague | contact | details | edit | friend | question | ... " - отношения между ссылаемым и текущим документами.

rev = "любая текстовая строка" - отношения между текущим и ссылаемым документами. Обратен атрибуту rel.

shape = "circle | default | poly | rect " - задает форму активной области ссылки для изображений.

tabindex = "любое целое положительное число, начиная с нуля " - определяет последовательность перехода между ссылками при нажатии на кнопку <Tab>. Значения выстраиваются последовательно и переход между ссылками происходит от меньшего значения к большему.

target = " _blank | _self | _parent | _top " - имя окна или фрейма, куда браузер будет загружать документ. (_blank - загружает страницу в новое окно браузера; _self - загружает страницу в текущее окно (это значение задается по умолчанию); _parent - загружает страницу во фрейм-родитель, если фреймов нет, то это значение работает как _self; _top - отменяет все фреймы и загружает страницу в полном окне браузера, если фреймов нет, то это значение работает как _self.

title="текст" - добавляет всплывающую подсказку к тексту ссылки.

type="text/css" - сообщает браузеру, какой синтаксис использовать, чтобы правильно интерпретировать стили. Указывает MIME-тип документа, на который ведёт ссылка.

Атрибут href определяет URL (Universal Resource Locator, универсальный указатель ресурса), иными словами, адрес документа, на который следует перейти, а содержимое контейнера <a> является ссылкой. Текст, расположенный между тегами <a> и , по умолчанию становится синего цвета и подчеркивается.

Если указана ссылка на файл, которого не существует, например, его имя в атрибуте href набрано с ошибкой, то такая ссылка называется битая. Битых ссылок следует категорически избегать, поскольку они вводят посетителей сайта в заблуждение.

По умолчанию, при переходе по ссылке документ открывается в текущем окне или фрейме. При необходимости, это условие может быть изменено атрибутом `target` тега `<a>`.

Изображение так же может быть ссылкой.

Синтаксис:

```
<a href=" URL изображения">...</a>
```

Любая ссылка является встроенным элементом, поэтому для нее действуют те же правила, что и для встроенных элементов. А именно, нельзя размещать внутри тега `<a>` блочные элементы, но допустимо делать наоборот, и вкладывать ссылку в блочный контейнер.

3.2.9. Вставка медиа.

Тег <bgsound> определяет музыкальный файл, который будет проигрываться на веб-странице при ее открытии. Громкость, продолжительность звучания музыки и другие характеристики определяются с помощью атрибутов тега, а также могут управляться через скрипты. Этот тег должен размещаться только в контейнере `<head>`.

Тег `<bgsound>` не входит в спецификацию HTML и при его использовании код не пройдет валидацию. К тому же пользователи, как правило, отрицательно относятся к музыке, которая играет на сайте, поэтому использовать эту возможность следует с осторожностью и по необходимости.

Синтаксис:

```
<head>
  <bgsound src="URL">
</head>
```

Атрибуты:

`balance="число"` - управляет балансом звука между правой и левой колонками. Целое число от -10000 до 10000. Значение ноль служит для баланса громкости, отрицательные числа увеличивают громкость в левой колонке, а положительные – в правой.

`loop="целое положительное число"` - устанавливает, сколько раз проигрывать музыкальный файл.

`src ="URL"` - путь к музыкальному файлу.

`volume = " целое число"` - задает громкость звучания музыки. Целое число от -10000 до 0. Ноль соответствует максимальной громкости звучания.

Тег <source> Вставляет звуковой или видеофайл для тегов `<audio>` и `<video>`. Обобщенно такие файлы называются медийными.

Синтаксис:

```
<audio>
   <source src="URL">
</audio>
<video>
   <source src="URL">
</video>
```

Атрибуты:

`media="all | braille | handheld | print | screen | speech | projection | tty | tv"` - определяет устройство, для которого будет воспроизводиться файл.

`src="URL"` - адрес файла.

`type = "MIME-тип"` - MIME-тип медийного источника.

Тег <audio>. Добавляет, воспроизводит и управляет настройками аудиозаписи на веб-странице. Путь к файлу задается через атрибут `src` или вложенный тег `<source>`. Внутри контейнера `<audio>` можно написать текст, который будет выводиться в браузерах, не работающих с этим тегом.

Синтаксис:

```
<audio src="URL"></audio>
<audio> <source src="URL"> </audio>
```

Атрибуты:

`autoplay` - звук начинает играть сразу после загрузки страницы.

`controls` - добавляет панель управления к аудиофайлу.

`loop` - повторяет воспроизведение звука с начала после его завершения.

`preload="none | metadata | auto"` - используется для загрузки файла вместе с загрузкой веб-страницы.

`src="URL"` - указывает путь к воспроизводимому файлу.

Тег <video> - добавляет, воспроизводит и управляет настройками видеоролика на веб-странице. Путь к файлу задается через атрибут `src` или вложенный тег `<source>`.

Синтаксис:

```
<video>
  <source src="URL">
</video>
```

Атрибуты:

`autoplay` - видео начинает воспроизводиться автоматически после загрузки страницы.

`controls` - добавляет панель управления к видеоролику.

`height ="целое положительное число в пикселях или процентах"` - задает высоту области для воспроизведения видеоролика.

`loop` - повторяет воспроизведение видео с начала после его завершения.

`poster="URL"` - указывает адрес картинки, которая будет отображаться, пока видео не доступно или не воспроизводится.

`preload` - используется для загрузки видео вместе с загрузкой веб-страницы.

`src="URL"` - указывает путь к воспроизводимому видеоролику.

`width ="целое положительное число в пикселях или процентах"` - задает ширину области для воспроизведения видеоролика.

Элемент <object> сообщает браузеру, как загружать и отображать объекты, которые исходно браузер не понимает. Как правило, такие объекты требуют подключения к браузеру специального модуля, который называется плагин, или запуска вспомогательной программы. *Плагином* называется подключаемый к браузеру программный модуль, расширяющий возможности браузера. Например, к плагинам можно отнести поддержку Flash, QuickTime VR, VRML и т.д.

Синтаксис:

```
<object ></object>
```

Атрибуты:

align="absmiddle | baseline | bottom | left | middle | right | texttop | top" - определяет, как объект будет выравниваться на странице и способ его обтекания текстом.

archive="URL" - устанавливает путь к файлам, необходимым для работы объекта.

classid="URL" - адрес программы (приложения или плагина), которая работает с данным объектом, и будет запускать его.

code="имя класса Java-апплета" - имя объекта для его выполнения.

codebase="URL" - путь к папке с объектом, который указан атрибутом code или classid.

codetype="MIME-тип" - указывает на тип объекта, который задан атрибутом classid.

data="URL" - адрес файла для его отображения в окне браузера.

height="целое положительное число в пикселях или процентах" - высота объекта.

hspace="целое положительное число в пикселях" - горизонтальный отступ от объекта до окружающего контента.

tabindex="целое положительное число или 0" - определяет порядок перехода между элементами с помощью клавиши Tab.

type="MIME-тип" - MIME-тип объекта.

vspace="целое положительное число в пикселях" - вертикальный отступ от объекта до окружающего контента.

width ="целое положительное число в пикселях или процентах" - ширина объекта.

Тег <embed> также используется для загрузки и отображения объектов (например, видеофайлов, флэш-роликов, некоторых звуковых файлов и т.д.), которые исходно браузер не понимает.

Вид внедренного объекта зависит от установленных в браузере плагинов, типа загружаемого файла, а также от атрибутов тега <embed>. На рисунках показан вид воспроизведения в браузере

Синтаксис:

```
<embed width="..." height="..."></embed>
```

Атрибуты:

align = "absmiddle | baseline | bottom | left | middle | right | texttop | top" - определяет как объект будет выравниваться на странице и способ его обтекания текстом.

height="целое положительное число в пикселях или процентах" - высота объекта.

hidden="true | false" - указывает, скрыть объект на странице или нет.

hspace="целое положительное число в пикселях" - горизонтальный отступ от объекта до окружающего контента.

pluginspage="URL" - адрес страницы в Интернете, откуда можно скачать и установить плагин к браузеру.

src="URL" - путь к файлу.

type="MIME-тип" - MIME-тип объекта.

vspace="целое положительное число в пикселях" - вертикальный отступ от объекта до окружающего контента.

width ="целое положительное число в пикселях или процентах" - ширина объекта.

Тег <noembed> предназначен для отображения информации на веб-странице, если браузер не поддерживает работу с плагинами. Во всех остальных случаях содержимое контейнера <noembed> будет проигнорировано.

Синтаксис:

```
<noembed>Текст</noembed>
```

3.2.10. Каскадные списки стилей.

Стилем или ***CSS*** (Cascading Style Sheets, каскадные таблицы стилей) называется набор параметров форматирования, который применяется к элементам документа, чтобы изменить их внешний вид. Возможность работы со стилями издавна включают в развитые издательские системы и текстовые редакторы, тем самым позволяя одним нажатием кнопки придать тексту заданный, заранее установленный вид. Теперь это доступно и создателям сайта, когда цвет, размеры текста и другие параметры хранятся в определенном месте и легко «прикручиваются» к любому тегу. Еще одним преимуществом стилей является то, что они предлагают намного больше возможностей для форматирования, чем обычный HTML.

Пример:

Использование страницы index.html с кодом

```
<!DOCTYPE html PUBLIC
"-//W3C//DTD XHTML 1.0 Transitional//EN"
http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
   <title>Пример </title>
  <meta http-equiv="Content-Type"
  content="text/html; charset=windows-1251" />
</head>
<body>
  <font color="#e14ccb"><b>Название раздела</b></font>
  <P> Текст</P>
</body>
</html>
```

Полностью аналогично использованию страниц index.html с кодом

```
<!DOCTYPE html PUBLIC
"-//W3C//DTD XHTML 1.0 Transitional//EN"
http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
   <title>Пример </title>
  <meta http-equiv="Content-Type"
  content="text/html; charset=windows-1251" />
  <link rel="stylesheet" href="style.css" type="text/css">
</head>
<body>
```

```
  <P class="razdel">Название раздела</P>
  <P> Текст</P>
</body>
</html>
```

И стилевого файла style.css с кодом

```
P.razdel{color: #e14ccb;
         font-weight: bolder;
        }
```

Так же можно описывать стилями параметры не для конкретного случая форматирования, а для всех элементов определенного типа. Так, например, для того, чтобы сделать начертание заголовков отличным от умолчального, в силевом файле можно прописать:

```
h1 {font-family: 'Times New Roman', Times, serif;
    font-size: 250%;
   }
```

Далее приведен список стилей по категориям.

Размеры, позиции и отступы.

`margin: [значение | проценты | auto] {1,4} | inherit`

Устанавливает величину отступа от каждого края элемента. Разрешается использовать одно, два, три или четыре значения, разделяя их между собой пробелом.

`margin-bottom: значение | auto | inherit`

Устанавливает величину отступа от нижнего края элемента.

`margin-left: значение | auto | inherit`

Устанавливает величину отступа от левого края элемента.

`margin-right: значение | auto | inherit`

Устанавливает величину отступа от правого края элемента.

`margin-top: значение | auto | inherit`

Устанавливает величину отступа от верхнего края элемента.

`bottom: значение | проценты | auto | inherit`

Устанавливает положение нижнего края содержимого элемента без учета толщины рамок и отступов.

`left: значение | проценты | auto | inherit`

Устанавливает положение левого края содержимого элемента.

`position: absolute | fixed | relative | static | inherit`

Устанавливает способ позиционирования элемента относительно окна браузера или других объектов на веб-странице.

`right: значение | проценты | auto | inherit`

Устанавливает положение правого края содержимого элемента.

`top: значение | проценты | auto | inherit`

Устанавливает положение верхнего края содержимого элемента.

`padding: [значение | проценты] {1, 4} | inherit`

Устанавливает значение полей вокруг содержимого элемента. Разрешается использовать одно, два, три или четыре значения, разделяя их между собой пробелом.

`padding-bottom: значение | inherit`

Устанавливает значение поля от нижнего края содержимого элемента.

`padding-left: значение | inherit`

Устанавливает значение поля от левого края содержимого элемента.

`padding-right: значение | inherit`

Устанавливает значение поля от правого края содержимого элемента.

`padding-top: значение | inherit`

`height: значение | проценты | auto | inherit`

Устанавливает высоту блочных или заменяемых элементов.

`width: значение | проценты | auto | inherit`

Устанавливает ширину блочных или заменяемых элементов.

`vertical-align: baseline|bottom|middle|sub|super|text-bottom|text-top|top|inherit | значение | проценты`

Выравнивает элемент по вертикали относительно своего родителя, окружающего текста или ячейки таблицы.

Таблицы.

`border-collapse: collapse | separate | inherit`

Устанавливает, как отображать границы вокруг ячеек таблицы.

`border-spacing: значение1 [значение2]`

Задает расстояние между границами ячеек в таблице.

Текст и шрифт.

`letter-spacing: значение | normal | inheritОпределяет интервал между символами в` пределах элемента.

`line-height: множитель | значение | проценты | normal | inherit`

Устанавливает интерлиньяж (межстрочный интервал) текста, отсчет ведется от базовой линии шрифта.

`text-align: center | justify | left | right | inherit`

Определяет горизонтальное выравнивание текста в пределах элемента.

`text-decoration: [ blink | line-through | overline | underline ] | none | inherit`

Добавляет оформление текста в виде его подчеркивания, перечеркивания, линии над текстом и мигания.

`text-indent: <значение> | <проценты> | inherit`

Устанавливает величину отступа первой строки блока текста.

`text-transform: capitalize | lowercase | uppercase | none | inherit`

Управляет преобразованием текста элемента в заглавные или прописные символы.

`white-space: normal | nowrap | pre | pre-line | pre-wrap | inherit`

Устанавливает, как отображать пробелы между словами.

`word-spacing: значение | normal | inherit`

Устанавливает интервал между словами.

`font-family: имя шрифта [, имя шрифта[, ...]] | inherit`

Устанавливает семейство шрифта, которое будет использоваться для оформления текста содержимого. Serif - шрифты с засечками (антиквенные), типа Times; sans-serif - рубленные шрифты (шрифты без засечек или гротески), типичный представитель - Arial; cursive - курсивные шрифты; fantasy - декоративные шрифты; monospace - моноширинные шрифты, ширина каждого символа в таком семействе одинакова (шрифт Courier).

`font-size: абсолютный размер | относительный размер | значение | проценты | inherit`

Определяет размер шрифта элемента.

`font-style: normal | italic | oblique | inherit`

Определяет начертание шрифта - обычное, курсивное или наклонное.

```
font-variant: normal | small-caps | inherit
```

Определяет, как нужно представлять строчные буквы - оставить их без модификаций или делать их все прописными уменьшенного размера.

```
font-weight:
bold|bolder|lighter|normal|100|200|300|400|500|600|700|800|900
```

Устанавливает насыщенность шрифта.

```
list-style-image: none | url('путь к файлу') | inherit
```

Устанавливает адрес изображения, которое служит в качестве маркера списка.

```
list-style-position: inside | outside
```

Определяет, как будет размещаться маркер относительно текста.

```
list-style-type: circle | disc | square | armenian | decimal | decimal-leading-zero | georgian | lower-alpha | lower-greek | lower-latin | lower-roman | upper-alpha | upper-latin | upper-roman | none | inherit
```

Изменяет вид маркера для каждого элемента списка.

Цвет и фон.

```
background-attachment: fixed | scroll | inherit
```

Устанавливает, будет ли прокручиваться фоновое изображение вместе с содержимым элемента.

```
background-clip: [padding-box | border-box | content-box] [, [padding-box | border-box | content-box]]
```

Определяет, как цвет фона или фоновая картинка должна выводиться под границами.

```
background-color: цвет | transparent | inherit
```

Определяет цвет фона элемента.

```
background-image: url(путь к файлу) | none | inherit
```

Устанавливает фоновое изображение для элемента.

```
background-origin: [padding-box | border-box | content-box] [, [padding-box | border-box | content-box]]
```

Определяет область позиционирования фонового рисунка.

```
background-position: [left | center | right | <проценты> | <значение>] ||            [top | center | bottom | <проценты> | <значение>] | inherit
```

Задает начальное положение фонового изображения, установленного с помощью свойства background-image.

```
background-position-x: left | center | right | <проценты> | <значение>
```

Задает положение фонового изображения внутри элемента по горизонтали.

```
background-position-y: top | center | bottom | проценты | значение
```

Задает положение фонового изображения внутри элемента по вертикали.

```
background-repeat: no-repeat | repeat | repeat-x | repeat-y | inherit
```

Определяет, как будет повторяться фоновое изображение, установленное с помощью свойства background-image.

```
background-size: [ <значение> | <проценты> | auto ]{1,2} | cover | contain
```

Масштабирует фоновое изображение согласно заданным размерам.

`color: цвет | inherit`

Определяет цвет текста элемента.

После того, как каскадные списки созданы, можно приступить к созданию и заполнению остальных внутренних страниц.

3.2.11. Тестирование и отладка.

Кроссбраузерность - свойство сайта отображаться и работать во всех популярных браузерах идентично. Под идентичностью понимается отсутствие развалов верстки и способность отображать материал с одинаковой степенью читабельности. Понятие «кроссбраузерность» очень часто путают с попиксельным соответствием, что на самом деле является разными понятиями.

Кроссбраузерность остается одной из самых сложных проблем в веб-разработке. Следование веб-стандартам само по себе дает достаточно высокий уровень совместимости, но не все браузеры совершенны. Перед разработчиком стоит задача, обеспечения максимальной совместимости сайта с наиболее популярными браузерами, чтобы каждый посетитель мог с удобством его использовать.

Для тестирования на кроссбраузерность кроме того, что можно просто установить несколько браузеров, существуют веб-сервисы, которые либо предоставляют удаленный доступ к браузеру через VPN либо позволяют просмотреть скриншоты отображения сайта в различных браузерах (семейство Mozilla, Internet Explorer, Opera, Safari, мобильные браузеры), в различных операционных системах (Mac OS, Linux, Win). Эффект от внесенных в разметку сайта изменений можно увидеть сразу же несмотря на то, что на компьютере дизайнера нет множества браузеров. К таким сервисам относятся http://browsershots.org, http://netrenderer.com, http://broserlab.adobe.com, http://www.browsrcamp.com.

Также всегда следует учитывать, что для просмотра сайта может использоваться широкий диапазон устройств – мониторы различного разрешения, планшетные компьютеры, телефоны и т.д.

Учёт разрешения монитора, на котором посетители будут просматривать сайт, - важный составной момент как разработки сайта, так и редизайна сайта. В техническом задании на разработку сайта или редизайн сайта желательно оговорить отдельным пунктом основной диапазон разрешения мониторов, на которых должен просматриваться сайт.

Естественно если компьютер веб-дизайнер не поддерживает большое разрешение экрана, то посмотреть как выглядит какой-либо сайт при разрешении 1920 на 1200 у него не получится. При этом может помочь сервис http://viewlike.us/.

Также дизайнер не должен забывать о том, что многие посетители сайта имеют не слишком высокую скорость соединения с Интернет. Даже наличие "выделенной" линии не всегда гарантирует высокую скорость соединения с Интернет. Игнорирование этого факта при строительстве сайта приводит к тому, что не все из потенциальных клиентов могут просмотреть сайт.

Существует ряд специальных сервисов проверки битых ссылок, которые позволяют найти все ссылки на сайте, которые ведут на несуществующие страницы или файлы. Не требуется установка какой-либо программы проверки ссылок или регистрации - сканирование сайта и просмотр результатов производится в режиме онлайн. Сервис

полностью обойдет весь сайт по внутренним ссылкам и соберет подробную информацию о его структуре и ошибках, а также позволит просмотреть полученные результаты в формате Microsoft Excel, а также сгенерировать файл карты сайта sitemap.xml в формате Sitemaps для поисковых систем Google и Яндекс.

К таким сервисам относятся http://www.keeplinks.ru, linkchecker и др.

3.2.12. Загрузка на хостинг.

Хостинг (англ. hosting) - услуга по предоставлению вычислительных мощностей для физического размещения информации на сервере, постоянно находящемся в сети (обычно Интернет). Хостингом также называется услуга по размещению оборудования клиента на территории провайдера с обеспечением подключения его к каналам связи с высокой пропускной способностью.

Обычно под понятием услуги хостинга подразумевают как минимум услугу размещения файлов сайта на сервере, на котором запущено ПО, необходимое для обработки запросов к этим файлам (веб-сервер). Как правило, в услугу хостинга уже входит предоставление места для почтовой корреспонденции, баз данных, DNS, файлового хранилища на специально выделенном файл-сервере и т. п., а также поддержка функционирования соответствующих сервисов.

Основные бесплатные хостинги – это Narod.ru и Ucoz.ru.

Далее приведены выдержки из официального руководства пользователя хостинга Narod.ru.

Яндекс.Народ-это сервис для создания и бесплатного хостинга сайтов, хранения и обмена файлами. Для того чтобы создать сайт или загрузить файлы на Народе, требуется доступ к персональной части Яндекса. Если вы уже зарегистрированы на Яндексе, введите свои логин и пароль, перейдя по ссылке «Войти», расположенной в правой верхней части страницы.

Загрузка файлов на сайт

Загрузка файлов осуществляется в **Мастерской**, в разделе **«Загрузка файлов на сайт»**. Нажмите на кнопку «Browse» или «Обзор» и выберите на вашем жестком диске файл для загрузки в текущую папку, он должен появиться в поле рядом с кнопкой. Выберите до десяти таких файлов (размер каждого файла не должен превышать 5 Мб) и нажмите на кнопку «Загрузить файлы». Вмена загружаемых файлов не должны содержать недопустимые символы, например, символы «пробел», русские буквы и т.д. Имя файла должно состоять из латинских букв, цифр, знаков дефис и подчеркивание, может содержать точку.

Вы также можете загружать файлы на свой сайт при помощи FTP. Чтобы воспользоваться протоколом FTP, понадобится специальная программа, так называемый FTP-клиент. FTP-клиентом может служить интернет-браузер (MS Internet Explorer, Netscape Navigator), программа управления файлами FAR, специализированная программа CuteFTP и другие. FTP-клиент встроен также в некоторые HTML-редакторы, такие как HomeSite, Dreamweaver.

Сразу после загрузки файлов вы продолжаете видеть старую версию вашей мастерской. Обновить изображение можно при помощи кнопок Refresh (Reload) в меню вашего браузера, или клавиш Ctrl+R на клавиатуре. Эта команда перезагружает странички непосредственно с сервера, а не из кэша вашего компьютера.

Файлы, размещаемые на Народе, должны быть в кодировке Windows-1251. Кроме того, не надо использовать meta-тег charset.

Управление файлами на сайте

Все управление файлами осуществляется с помощью файл-менеджера в разделе **«Управление файлами» Мастерской**. Здесь вы сможете загружать готовые файлы или удалять их, создавать html-файлы самостоятельно, редактировать или переименовывать уже созданные, создавать и редактировать страницы по шаблону и т. д.

Открыв файл-менеджер, вы увидите список ваших файлов, для каждого из которых будет указано имя, описание, размер и дата последнего изменения. По умолчанию файлы разделены по типам: вначале выводятся страницы, созданные по шаблонам, затем нешаблонные html-страницы, далее - файлы-картинки и файлы иных видов. Кроме того, в самом верху, над списком файлов, выводится список подпапок текущей папки. Каждый тип объекта для удобства обозначен своим значком.

Все операции в файл-менеджере можно условно разделить на три группы: индивидуальные действия с объектами (файлами и папками), групповые операции и создание новых объектов. Индивидуальные действия (задание имени и описания объекта, управление счетчиками, редактирование html-кода) реализуются с помощью кнопки «Свойства», которая присутствует в каждой строчке файл-менеджера. При ее нажатии открывается всплывающее окно, напоминающее окно свойств в Windows. Отдельным свойствам файлов соответствуют закладки, отображаемые в левой части этого окна. При выборе закладки в правой части окна появляется форма для управления соответствующим свойством файла. Для сохранения измененных свойств служит кнопка «Сохранить» при ее нажатии страница с файл-менеджером обновляется.

Групповые действия - копирование, перемещение и удаление - выполняются с помощью одноименных кнопок, которые размещаются в верхней и нижней частях файл-менеджера; при этом для осуществления операций предварительно требуется выделить нужные файлы или папки, отметив соответствующие флажки. Создание новых папок, страниц по шаблону и html-файлов реализуется с помощью ссылок, которые расположены сразу над шапкой файл-менеджера. Там же находится ссылка «Настройки», которая открывает раздел, где можно изменить внешний вид и правила отображения файлов в файл-менеджере.

Работа с папками

Для перехода в одну из папок нажмите ссылку с ее именем. На экран будет выведено содержимое выбранной папки. Чтобы вернуться на один уровень вверх, нажмите значок с изображением открытой папки. При нахождении в одной из внутренних папок есть возможность перейти в любую из папок вышележащих уровней: в файл-менеджере всегда указывается полный путь к выбранной папке, причем каждый элемент этого пути является ссылкой, нажав которую, вы перейдете в соответствующую папку.

Чтобы создать новую папку, выберите ссылку «Создать папку», после чего укажите имя папки. Для переименования существующей папки воспользуйтесь кнопкой «Свойства», затем выберите закладку «Имя и описание», задайте новое имя (здесь же можно задать и словесное описание папки - для удобства) и нажмите кнопку «Сохранить».

Для удаления одной или нескольких папок отметьте нужные папки флажками и нажмите желтую кнопку «Удалить» (расположена на серой полосе над и под списком

файлов). Аналогичным образом можно скопировать или переместить нужные папки в другую папку, при этом папки копируются и перемещаются со всем их содержимым.

Как и в традиционных файл-менеджерах, которые служат для управления файлами на вашем компьютере, файл-менеджер Narod.ru позволяет удалять файлы, копировать, перемещать, переименовывать и редактировать. Удаление, копирование и перемещение реализованы как групповые операции. Это означает, что вначале следует выбрать один или несколько файлов, а затем нажать желтую кнопку с названием нужной операции. Переименование и редактирование файлов - индивидуальные операции, для их выполнения следует нажать кнопку «Свойства» напротив нужного файла и выбрать закладку, соответствующую требуемому действию. Закладка «Имя и описание» позволяет указать новое имя файла и задать ему словесное описание.

Редактирование файлов

Отредактировать html-код файла можно с помощью закладки «Текстовый редактор». Для того чтобы отредактировать страницу, созданную по шаблону, воспользуйтесь кнопкой «Редактировать» .

Для того чтобы посмотреть любой из файлов, представленных в файл-менеджере, просто нажмите на ссылку с его именем - файл при этом откроется в новом окне.

Вы можете изменить html-код своих страниц прямо на сайте, воспользовавшись онлайновым текстовым редактором. Чтобы вызвать его, нажмите кнопку «Свойства» напротив нужного файла и затем выберите закладку «Текстовый редактор». Чтобы вызвать редактор для страницы, созданной не по шаблону, или текстового файла, достаточно нажать кнопку «Редактировать» , которая находится слева от имени файла.

Редактор представляет собой текстовое поле, в котором отображается html-код или текст страницы и содержимое которого можно изменять (в этом и состоит редактирование), и под этим полем - несколько кнопок с действиями. Для того чтобы было удобнее работать с длинными строками, окно с текстовым редактором, как и любое другое окно в Windows, можно раскрыть на весь экран. После того как вы внесли изменения в текст, нажмите кнопку «Сохранить», и отредактированная страница запишется на ваш сайт.

По ходу внесения изменений удобно проверять, какой вид приобретает страница, - для этого служит кнопка «Пред. просмотр» страница при этом открывается в новом окне.

Если вы хотите отменить только что внесенные правки, пока еще не была нажата кнопка «Сохранить», нажмите кнопку «Отменить изменения», и тогда текст в поле станет таким, каким он был до начала редактирования.

3.3. Оформление пояснительной записки

Титульный лист пояснительной записки оформляется в соответствии с приложением 1.

В расчетно-пояснительной записке к курсовой работе должны быть:

- все эскизы и наработки созданные в процессе разработки концепции дизайна;
- подробный листинг кода всех страниц спроектированного сайта;
- скриншоты заглавной и внутренних страниц спроектированного веб-сайта.

Пояснительная записка оформляется в соответствии с ЕСКД и ГОСТ 2.105-79 (см. приложение 3).

Примечание: Допускается не делать рамку на листах пояснительной записки.

4. БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1) Курушин, В.Д. Дизайн и реклама : самоучитель / В.Д.Курушин .— М. : ДМК Пресс, 2006.— 272с.

2) Пауэлл Т. Web-дизайн : пер.с англ. / Т.Пауэлл .— 2-е изд. — СПб. : БХВ-Петербург, 2005 .— 1072с.

3) Вин, Д. Искусство web-дизайна : Самоучитель / Дж.Вин; Пер.с англ.О.Кузнецова .— СПб.и др. : Питер, 2002 .— 224с.

4) Комолова, Н.В. Компьютерная верстка и дизайн / Н.В. Комолова .— СПб. : БХВ-Петербург, 2003 .— 512с.

5) Все о работе с цветом в Web : пособие для дизайнеров и программистов .— Минск : Харвест, 2007 .— 320с.

6) Сеймур-Коэн, Seymour Cohen L. Секреты дизайнера.Профессиональные приемы в Adobe Photoshop 7 и Adobe Illustrator 10 : пер.с англ. / Л.Сеймур-Коэн .— 4-е изд. — М. : КУДИЦ-ОБРАЗ, 2003 .— 192с.

7) Яцюк, О.Г. Основы графического дизайна на базе компьютерных технологий / О.Г.Яцюк .— СПб. : БХВ-Петербург, 2004 .— 240с.

8) Джонсон, Johnson J. Web-дизайн:типичные ляпы и как их избежать : пер.с англ. / Д.Джонсон .— М. : КУДИЦ-ОБРАЗ, 2005 .— 400с.

9) Кирсанов, Д. Веб-дизайн:книга Дмитрия Кирсанова / Д.Кирсанов .— СПб. : Символ-Плюс, 2001 .— 376с.

ПРИЛОЖЕНИЕ 1
Образец оформления титульного листа записки

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Тульский государственный университет»

Политехнический институт
Факультет транспортных и технологических систем

Кафедра «Технология полиграфического производства и защиты информации»

КУРСОВАЯ РАБОТА
по дисциплине
"Web-дизайн"
на тему:
«Создание сайта-визитки»

Выполнил ст. гр._____
_____ (И.О. Фамилия)
«_____» _____20__ г.

Проверил._____
_____ (И.О. Фамилия)
«_____» _____20__ г.

Тула 20__

ПРИЛОЖЕНИЕ 2
Образец оформления бланка задания

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Тульский государственный университет»

Политехнический институт
Факультет транспортных и технологических систем

Кафедра «Технология полиграфического производства и защиты информации»

ЗАДАНИЕ

на выполнение курсовой работы (проекта)
по дисциплине _____

Студенту гр. _____ _____
индекс группы фамилия, и. о.

Тема
_____
_____

Рекомендуемая литература_____
_____
_____

Дата выдачи задания _____

Срок защиты _____

Задание принял _____ _____
подпись студента фамилия, и. о.

Задание выдал _____ _____
подпись преподавателя фамилия, и. о.

Пояснительная записка пояснительная записка набирается на компьютере в текстовом редакторе типа *Лексикон* (под MS DOS) или *WORD* (под Windows).

При использовании *WORD*, текст набирается шрифтом *Times New Roman (Cyr)* величиной 14 пунктов с одинарным интервалом. Формат бумаги - А4. Абзацный отступ - 1,25 см. Все поля страницы – по 2 см, переплет – 1 см. Текст на странице выравнивается по ширине.

Таблицы желательно располагать на странице без разрыва, а в случае переноса на другую страницу – дублируется шапка таблицы.

Рисунки располагаются по тексту пояснительной записки. В порядке исключения рисунки могут быть выполнены на отдельных листах белой бумаги, либо на кальке черной тушью или пастой. Рисунки имеют подрисуночную надпись и нумерацию – либо сквозную, либо по разделам.

Формулы следует выполнять в редакторе *Microsoft Equation* со следующими размерами:

обычный ..... 18 пт;
крупный индекс ..... 14 пт;
мелкий индекс ..... 12 пт;
крупный символ ..... 24 пт;
мелкий символ ..... 10 пт.

Шрифты: *Times New Roman (Cyr), Symbol*.

Ссылки на литературу даются в квадратных скобках.

Желательно проверять орфографию и грамматику текста пояснительной записки перед распечаткой – для этого в редакторах имеются специальные опции!

Второй страницей пояснительной записки (первая – титульный лист) является аннотация. На этом листе необходимо выполнить рамку и основную надпись как для текстовых документов (см. спецификацию).

Остальные требования к пояснительной записке – см. ГОСТы на выполнение текстовых документов.

Образец выполнения титульного листа прилагается (приложение 1). В библиотеке кафедры имеется его электронная версия.

СПИСОК НОРМАТИВНОЙ ДОКУМЕНТАЦИИ

Стандарты ЕСКД по правилам выполнения чертежей и схем и на условные графические обозначения.

Общие правила выполнения некоторых документов.

1. ГОСТ 2.102-68. ЕСКД Виды и комплекты конструкторской документации;
2. ГОСТ 2.104-68. ЕСКД Основные надписи(1-1-73)*) ;
3. ГОСТ 2.105-79. ЕСКД Основные требования к текстовым документам;
4. ГОСТ 2.106-68. ЕСКД Текстовые документы;

ПРИЛОЖЕНИЕ 4
Бланк ТЗ на разработку веб-сайта

Техническое задание на проект

_____

	Подпись	ФИО
Разработал		
Утвердил		

Общие сведения
Название сайта

_____

Целевая аудитория

_____

Цели и задачи, которые должен решать сайт

_____

Возможные сайты-конкуренты

_____

Тип сайта/портала

_____

Языковые версии

_____

Желаемый адрес сайта

_____

Требования к дизайну
Стиль сайта

_____

Впечатление, которое сайт должен производить на пользователя

_____

Что пользователь должен в результате посещения сайта
увидеть

_____

почувствовать

_____

сделать

_____

Есть ли фирменный стиль/бренд бук

_____

Обязательные элементы для всех страниц

_____

Примерная желаемая цветовая гамма

_____

Основные требования к графическому дизайну

_____

Использование анимации

_____

Структура разделов сайта

Содержание разделов сайта