

Учреждение образования
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

WEB-ДИЗАЙН И РАЗРАБОТКА МУЛЬТИМЕДИЙНЫХ ИЗДАНИЙ

Учебно-методическое пособие для студентов
специальности 1-47 01 01 «Издательское дело»
очной и заочной форм обучения

Минск
2021

УДК 004.9(075.8)(0.034.2PDF)
ББК 32.97я73
В26

Рассмотрено и рекомендовано к изданию редакционно-издательским советом Белорусского государственного технологического университета.

Составители:
Н. И. Потапенко
А. Н. Щербакова

Рецензенты:
кандидат физико-математических наук, доцент, заведующий кафедрой логистики и информационно-математических дисциплин УО «БИП – Университет права и социально-информационных технологий» *А. В. Остапенко*;
кандидат педагогических наук, доцент, заведующая кафедрой информационных технологий и моделирования экономических процессов УО «Белорусский аграрный технический университет» *О. Л. Сапун*

Web-дизайн и разработка мультимедийных изданий : учеб.-метод. пособие для студентов специальности 1-47 01 01 «Издательское дело» очной и заочной форм обучения / сост. : Н. И. Потапенко, А. Н. Щербакова. – Минск : БГТУ, 2021. – 100 с.

Пособие содержит лабораторный практикум и материалы, отражающие основные требования, предъявляемые к курсовой работе, ее тематике и содержанию, порядку оформления и защиты.

Предназначено для оказания помощи в выполнении лабораторных и оформлении курсовых работ студентами специальности 1-47 01 01 «Издательское дело», также может использоваться студентами других специальностей полиграфического комплекса.

УДК 004.9(075.8)(0.034.2PDF)
ББК 32.97я73

© УО «Белорусский государственный технологический университет», 2021

СОДЕРЖАНИЕ

ПРЕДИСЛОВИЕ	5
Лабораторная работа № 1 ПОИСК В ЭЛЕКТРОННЫХ БИБЛИОТЕЧНЫХ КАТАЛОГАХ И СИСТЕМАХ.....	6
Лабораторная работа № 2 СОЗДАНИЕ И ОФОРМЛЕНИЕ HTML-ДОКУМЕНТА	9
Лабораторная работа № 3 СПИСКИ	15
Лабораторная работа № 4 ТАБЛИЦЫ	20
Лабораторная работа № 5 ИЗОБРАЖЕНИЯ	23
Лабораторная работа № 6 HTML. ВЕРСТКА СТРАНИЦЫ. БЛОКИ. СЕМАНТИКА	26
Лабораторная работа № 7 ГИПЕРССЫЛКИ. НАВИГАЦИЯ.....	29
Лабораторная работа № 8 АНИМАЦИЯ И ТРАНСФОРМАЦИЯ НА CSS3	37
Лабораторная работа № 9 ВВОД И ВЫВОД ДАННЫХ В JAVASCRIPT. ВЫЧИСЛЕНИЯ	41
Лабораторная работа № 10 ФУНКЦИИ И СОБЫТИЯ	45
Лабораторная работа № 11 ФОРМЫ.....	50
Лабораторная работа № 12 РАБОТА С БИБЛИОТЕКОЙ JQUERY.....	57
Лабораторная работа № 13 СИСТЕМЫ УПРАВЛЕНИЯ КОНТЕНТОМ	62
Лабораторная работа № 14 БИБЛИОТЕКА WOW.JS, ANIMATE.JS, TEXTILATE.JS	65
Лабораторная работа № 15 СЛАЙДЕРЫ НА WEB-СТРАНИЦЕ	70

Лабораторная работа № 16 СОЗДАНИЕ СПОЙЛЕРОВ	72
Лабораторная работа № 17 СОЗДАНИЕ ВИДЕО, АНИМАЦИИ. ВИДЕОМОНТАЖ.....	73
Курсовая работа РЕКОМЕНДАЦИИ ПО ОРГАНИЗАЦИИ И ВЫПОЛНЕНИЮ РАБОТ	75
Приложение А. Регистрационная форма	90
Приложение Б. Пример дизайн-макета.....	95
Приложение В. Образец титульного листа	96
Приложение Г. Форма листа задания.....	97
СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ.....	99

ПРЕДИСЛОВИЕ

Учебная дисциплина «Web-дизайн и разработка мультимедийных изданий» предназначена для реализации на первой ступени высшего образования по специальности 1-47 01 01 «Издательское дело». Актуальность ее обусловлена высокой степенью проникновения информационных технологий в издательское дело. Предназначена для формирования у студентов базовых знаний об использовании современных информационных компьютерных технологий для обработки мультимедийной информации, реализации веб-проектов и публикации их в сети интернет, что позволит в дальнейшем эффективно применять полученные знания на практике.

В учебно-методическом пособии представлен лабораторный практикум по дисциплине, который включает 17 лабораторных работ и методические указания по разработке курсовой работы.

Практикум охватывает все темы дисциплины и позволяет овладеть технологиями разработки мультимедийных изданий в свете понятий современного веб-дизайна и интернет-программирования.

По учебному плану на лабораторные работы отводится 54 ч. В представленном пособии для каждой работы сформулирована цель, а также рекомендуемая длительность ее выполнения. В лабораторной работе приведены краткие теоретические сведения, необходимые для успешного ее выполнения, и типовые примеры.

Методические рекомендации к курсовой работе содержат описание всех этапов ее выполнения, информацию о структуре курсовой работы, наполнении разделов и требованиях к оформлению пояснительной записки. Курсовая работа выполняется в 7-м семестре.

Технической базой дисциплины являются персональные компьютеры, современные периферийные средства с необходимым установленным программным обеспечением.

Лабораторная работа № 1

ПОИСК В ЭЛЕКТРОННЫХ БИБЛИОТЕЧНЫХ КАТАЛОГАХ И СИСТЕМАХ

Цель работы: отработать навыки технологии поиска в электронных библиотечных каталогах; ознакомиться с ИР и ИС в регистре информационных ресурсов.

Время выполнения: 2 ч.

Задачи работы:

1. Приобрести навыки технологии поиска в электронных каталогах библиотек (задание № 1).
2. Создать библиографический указатель по приведенным ниже темам (каждый студент выбирает индивидуальную тему и отражает ее в названии файла практической работы) (задание № 2).
3. Заполнить регистрационную форму на ИР (прил. 1).

Источники для выполнения работы:

Официальные интернет-ресурсы коллекций онлайн-ресурсов.

Национальная библиотека Беларуси <http://nlb.by>

Центральная научная библиотека имени Якуба Коласа Национальной академии наук Беларуси <http://csl.bas-net.by>

Библиотека Белорусского государственного технологического университета <https://bibl.belstu.by>

Фундаментальная библиотека Белорусского государственного университета <http://www.bsu.library.by>

Президентская библиотека Республики Беларусь <http://preslib.org.by>

Республиканская научно-техническая библиотека <http://rntb.org.by>

Российская государственная библиотека <http://www.rsl.ru/>

Российская национальная библиотека <http://www.nlr.ru:8101/>

Государственная публичная научно-техническая библиотека России <http://www.gpntb.ru/>

Библиотека по естественным наукам <http://www.benran.ru>

Российская государственная библиотека по искусству <http://www.liart.ru/>

Государственная публичная историческая библиотека <http://www.shpl.ru/>

Библиотека технической литературы <http://www.listlib.narod.ru>

Национальная библиотека Германии <http://www.ddb.de/>

Британская библиотека <http://portico.bl.uk/>

Королевская библиотека Дании <http://www.kb.dk/>
Библиотека Конгресса США <http://lcweb.loc.gov/>
Национальная библиотека Финляндии <http://www.nrl.fi/>
Королевская библиотека Швеции <http://www.kb.se>



Задание 1

1. Найдите историю создания библиотеки БГТУ. Отрадите количество книг в печатном, электронном варианте, перечислите основные сервисы, доступные посетителям.

2. Используя ресурсы электронных библиотек, найдите и сохраните в отчете ссылки на текст произведения И. С. Тургенева «Записки охотника».

3. Подберите учебные материалы по курсу «Типографика». Перечислите список вузов Беларуси, Российской Федерации и СНГ (одна любая страна СНГ на выбор), в которых преподается подобный курс.

4. Найдите рейтинг самых читаемых авторов и ссылки на новые поступления в наиболее популярных электронных библиотеках. Укажите, с каким из поисковых серверов вам было комфортнее работать при выполнении пункта 4 задания № 1.



Задание 2

Составьте библиографический указатель (не менее 5 источников (статья, книга, видеофильм и т. п.)). Оформите в виде таблицы:

Библиографический указатель по теме _____

№	Название источника (.)	Название библиотеки и вид электронного каталога, в котором найден источник
1		

! Просто ссылки на статьи в интернете не подходят! Только источники из каталога библиотек.

Перечень тем для создания библиографического указателя:

1. Значение и место национальных и мировых информационных ресурсов в информационном обществе.

2. Важнейшие вехи политики белорусского государства в области информатизации и формирования информационного общества.

3. Система государственной регистрации НИР, ОКР, и ОТР в Республике Беларусь.
4. Правовые основы формирования информационного ресурса.
5. Информационные ресурсы государственной статистики Республики Беларусь.
6. Информационные ресурсы порталов и сайтов государственных органов.
7. Информационные ресурсы государственных архивов.
8. Структура информационных ресурсов сети интернет.
9. Модели «электронного правительства» зарубежных стран.
10. Сущность и виды информационных потребностей.
11. Типографика
12. Теория цвета.
13. Производство бумаги.
14. Старинные книги и фолианты – самые древние книги.



Задание 3

Используя один из источников (сайт факультета), заполните форму регистрации на ИР (прил. А).



Контрольные вопросы

1. Что такое информационные ресурсы?
2. Дайте определение понятию «информационная система».
3. Что называют библиографическим указателем и для чего он предназначен?

Лабораторная работа № 2

СОЗДАНИЕ И ОФОРМЛЕНИЕ HTML-ДОКУМЕНТА

Цель работы: изучить структуру HTML-документа, научиться создавать и оформлять страницу в формате HTML.

Время выполнения: 2 ч.



Теоретические сведения

Web-страницы можно создавать в простейшем текстовом редакторе Блокнот, а также используя Notepad++, Brackets, HTML-Online, Atom и множество др. При написании кода в Блокнот необходимо сохранить файл с расширением *.html*.

Любая web-страница структурно разбивается на две части: заголовок и тело. В заголовке указывается служебная информация о всей странице, а в теле web-документа описывается ее содержимое вместе с правилами оформления.

```
<html>
  {
    <head>
      {
        <title> Заголовок документа </title>
        <meta charset="unicod">
      }
    }
  {
    <body>
      {
        Тело документа
      }
    }
  }
</html>
```

В заголовок документа могут входить: тег, отображающий наименование web-страницы, тег стилевого оформления (*style*), тег выполняемого сценария (*script*) и метаданные.

Между тегами `<TITLE>` и `</TITLE>` указывается название страницы, которое отображается в строке заголовка окна при просмотре страницы в браузере.

Для внедрения метаданных в web-страницу применяется тег `<META>`. Тегов `<META>` может быть несколько, и они сообщают браузеру

неотображаемую на экране служебную информацию о web-странице. Чаще всего тег имеет следующий вид:

```
<META name="Author" content="Ivanov Petr">
```

Для указания поисковой машине набора ключевых слов для поиска необходимо вставить следующую конструкцию:

```
<META name="keywords" content="животные, зоопарк, природа, фауна">
```

С помощью тега <META> можно указать срок годности web-страницы:

```
<META http-equiv="Expires" content="Tue, 20 Aug 2020 14:25:00 GMT">
```

Для определения кодовой страницы и указания на то, что в документе содержится текст в формате HTML (*с раскладкой на русском языке*), необходим следующий тег в разделе заголовка:

```
<META charset="UTF-8">
```

Для комментариев рекомендовано пользоваться следующей конструкцией:

```
<!-- Комментарий -->
```

Внутри комментариев не должны встречаться угловые скобки. В открывающей части пробелы не допускаются. В закрывающей части пробел возможен между – и угловой скобкой >.

В теле документа между тегами <body> и </body> заключается информация, которая определяет web-страницу. Это та часть документа, которую разрабатывает автор и которая отображается браузером. Внутри элемента <body> можно использовать все элементы, предназначенные для дизайна web-страницы, также внутри него могут задаваться стилевые свойства для оформления страницы. Эти же свойства действительны для большинства тегов

```
<body style="color:green"; "padding-left: 1cm">
```

! Однако это не очень хорошая практика, и мы ее применим только на этой лабораторной работе, чтобы разобраться, как структурируется информация на странице и как мы ее оформляем.

Некоторые стилевые свойства

ЦВЕТ ТЕКСТА:

color = {color: red | color: #FF4400}, где red – название цвета; #FF4400 – шестнадцатиричный код цвета.

ЦВЕТ ФОНА

Стилевое свойство **background-color** – цвет фона элемента. Может применяться практически к любому элементу.

Значения:

background-color: цвет – явное задание цвета.

background-color: transparent – фон прозрачный.

background-image: url('images/flower.jpg').

background-repeat: repeat-x (возможно repeat-y | no-repeat | repeat).

РАЗМЕР ШРИФТА

Задается в процентах, сантиметрах, пикселах.

Стилевое свойство **font-size** – размер шрифта.

Стилевое свойство **line-height** – высота строки (интервал между строками).

font-weight: bold – жирный шрифт.

font-weight: normal – нормальный (нежирный) шрифт.

font-weight: bolder – жирнее, чем у контейнера. Также допустимо значение **lighter** – светлее (менее жирно), чем у контейнера. Не рекомендуется использовать.

font-weight: число – явное задание жирности. Могут быть указаны значения 100, 200, 300, 400, 500, 600, 700, 800, 900. Их трактовка зависит от браузера, поэтому их не рекомендуется использовать.

font-style: italic – курсивный шрифт. Его начертание в общем случае отличается от обычного, если используемый шрифт содержит символы такого начертания.

font-style: oblique – наклонный шрифт. Его начертание совпадает с обычным, символы обычного начертания просто наклоняются вправо.

font-style: normal – обычный шрифт.

font-family: serif – шрифт с засечками.

font позволяет задать несколько реквизитов шрифта сразу одной записью.

```
<p style="font: italic bold 20px/200% Arial">
```

РАМКИ

Стилевое свойство **border-style** – стиль рамки.

Стилевое свойство **border-top-style** – стиль верхней линии рамки.

Стилевое свойство **border-right-style** – стиль правой линии рамки.

Стилевое свойство **border-bottom-style** – стиль нижней линии рамки.

Стилевое свойство **border-left-style** – стиль левой линии рамки.

border-style задает стиль всей рамки или отдельной линии рамки.

Может применяться практически к любому элементу.

border-style: solid – сплошная рамка.

border-style: double – двойная рамка.

border-style: dotted – пунктирная рамка.

border-style: dashed – штриховая рамка.

border-style: groove – трехмерная выемка.

border-style: hidden – скрытая рамка

border-style: none – нет рамки.

ОБЪЯВЛЕНИЕ

```
<p style="border-style: solid; border-color: red">
```

УПРАВЛЕНИЕ РАЗМЕРАМИ

Стилевое свойство **width** – ширина содержимого.

Стилевое свойство **height** – высота содержимого.

Свойства размеров могут применяться практически к любому блочному элементу. Задают ширину и высоту содержимого блочного элемента. Если значение ширины и / или высоты не задано, оно считается заданным как **auto**.

Значения:

width: размер – ширина явно задана. При указании размера в процентах процент высчитывается относительно ширины контейнера.

width: auto – ширина подбирается автоматически.

Стилевое свойство **text-align** – горизонтальное выравнивание содержимого. Может применяться практически к любому блочному элементу.

Значения:

text-align: left – выравнивание по левой границе.

text-align: right – выравнивание по правой границе.

text-align: center – выравнивание по центру.

text-align: justify – выравнивание по ширине.

Таблица цветов представлена на рисунке.

















	Black = "#000000"		Green = "#008000"
	Silver = "#C0C0C0"		Lime = "#00FF00"
	Gray = "#808080"		Olive = "#808000"
	White = "#FFFFFF"		Yellow = "#FFFF00"
	Maroon = "#800000"		Navy = "#000080"
	Red = "#FF0000"		Blue = "#0000FF"
	Purple = "#800080"		Teal = "#008080"
	Fuchsia = "#FF00FF"		Aqua = "#00FFFF"

Таблица цветов

ТЕГИ ДЛЯ РАБОТЫ С ОФОРМЛЕНИЕМ ТЕКСТА

**
** – разрыв строки. Тег **
** непарный.

<p> – новый абзац (параграф) (закрывающий тег не обязателен).

<P ALIGN={LEFT, RIGHT, CENTER, JUSTIFY}>

<h1>...</h1>, <h2>...</h2>, ..., <h6>...</h6> – заголовки.

<pre>...</pre> – определяет заранее форматированный текст.

<hr> – горизонтальная линия (непарный).

<cite>...</cite> – цитата (отметка цитат, названий источников).

<dfn>...</dfn> – определение, обычно отображается курсивом.

... – выделение важных фрагментов текста, обычно отображается курсивом.

... – выделение очень важных фрагментов текста.

АТРИБУТЫ

align (выравнивание)={left, right, center}.

width =(ширина в пикселах или процентах).

size =(толщина в пикселах).

noshade (без тени).

color =(цвет).



Порядок выполнения работы

Самостоятельно создайте новый документ. Имя страницы – ваша фамилия.

Оформите отрывок следующим образом:

Заголовок – Н1.

Первый абзац – зеленый цвет шрифта, рамка – пунктирная зеленого цвета. Размер шрифта – 14 пт., выравнивание – по ширине.

Второй абзац отделен от первого линией.

Шрифт второго абзаца – красный, размер – 15 пт., выравнивание – по левому краю.

Михаил Александрович Берлиоз – выделить как важный фрагмент текста.

Подпись «Отрывок...» оформить как цитату.

Третий абзац – цвет шрифта – синий, рамка – сплошная синяя.

Глава 1. Никогда не разговаривайте с неизвестными

Однажды весной, в час небывало жаркого заката, в Москве, на Патриарших прудах, появились два гражданина. Первый из них, одетый в летнюю серенькую пару, был маленького роста, упитан, лыс, свою приличную шляпу пирожком нес в руке, а на хорошо выбритом лице его помещались сверхъестественных размеров очки в черной роговой оправе. Второй – плечистый, рыжеватый, вихрастый молодой человек в заломленной на затылок клетчатой кепке – был в ковбойке, жеваных белых брюках и в черных тапочках.

Первый был не кто иной, как Михаил Александрович Берлиоз, председатель правления одной из крупнейших московских литературных ассоциаций, сокращенно именуемой МАССОЛИТ, и редактор толстого художественного журнала, а молодой спутник его – поэт Иван Николаевич Поньрев, пишущий под псевдонимом Бездомный.

Попав в тень чуть зеленеющих лип, писатели первым делом бросились к пестро раскрашенной будочке с надписью: «Пиво и воды».

Отрывок из романа М. Булгакова «Мастер и Маргарита»

Сохраните документ *doc1.htm* в папке (диск D:/Фамилия).

Откройте созданный файл в браузере. Вы увидите, как выглядит созданный вами файл. Обратите внимание, как отображается текст.



Контрольные вопросы

1. Как создать простейшую web-страницу?
2. Что представляет собой структура HTML-документа?
3. В каком разделе страницы указывается заголовок?
4. Для чего используются мета-данные?

Лабораторная работа № 3

СПИСКИ

Цель работы: сформировать практические навыки по созданию списков; знать виды и элементы создания списков; уметь изменять вид маркера списка; использовать стилевые оформления при создании списков.

Время выполнения: 2 ч.



Теоретические сведения

Список – это упорядоченный набор некоторых элементов (слова, предложения и т. д.). При использовании соответствующих тегов разработчику не надо думать о нумерации пунктов. Если список дополняется новыми пунктами или укорачивается, нумерация корректируется автоматически.

Различают следующие виды списков: *маркированный*, *нумерованный* и *с определениями*.

Маркированный список

```
<UL>
  <LI> – элемент 1
  <LI> – элемент 2
  <LI> – элемент 3
</UL>
```

- – элемент 1
- – элемент 2
- – элемент 3

Вид маркера определяет параметр `TYPE=« . . »`, принимающий значения **disc (круг)**, **circle** или **round (окружность)**, **square (квадрат)**. В качестве маркеров можно использовать графические изображения: после указания `` размещаем тег ``.

Нумерованный список

```
<OL>
  <LI> – элемент 1
  <LI> – элемент 2
  <LI> – элемент 3
</OL>
```

1. – элемент 1
2. – элемент 2
3. – элемент 3

Каждый пункт маркируется элементом упорядоченной последовательности: арабскими или римскими числами, буквами латинского алфавита. Способ нумерации задается при помощи атрибута type.

type="1" – 1, 2, 3, 4.
type="i" – i, ii, iii, iv.
type="I" – I, II, III, IV.
type="a" – a, b, c, d.
type="A" – A, B, C, D.

Списки определений

В отличие от других типов списков каждый элемент списка определений состоит из двух частей: в первой части записывается определяемый термин, а во второй – текст в форме словарной статьи.

```
<DL>  
<DT> Минск  
<DD> Столица Беларуси  
<DT> Москва  
<DD> Столица России  
</DL>
```

Минск Столица Беларуси
Москва Столица России

Можно формировать вложенные списки. В данном случае будут использоваться комбинации трех видов тегов: , , <DL>.



Порядок выполнения работы

1. Разработайте следующие виды списков:

Необходимо предоставить следующие документы:

1. паспорт,
2. заявление,
3. справку о доходах.

Необходимо предоставить следующие документы:

- паспорт,
- заявление,
- справку о доходах.

Необходимо предоставить следующие документы:

- паспорт,
- заявление,
- справку о доходах.

Необходимо предоставить следующие документы:

1. паспорт
 - a. срок годности не менее 3 лет
2. заявление
 - a. на имя ректора
 - b. на имя декана
3. справку о доходах
 - a. за полгода
 - b. за последние три месяца

Необходимо предоставить следующие **документы**:

- ❖ паспорт,
- ❖ заявление,
- ❖ справку о доходах.

Основные рекомендации по осушению болот:

- i. Выбирайте правильное болото. Это самое важное правило, поэтому отнеситесь к нему особо внимательно.
- ii. Убедитесь, что в болоте никто не купается.
- iii. Осушайте болото быстро, без колебаний.

Оформите список определений:

Буква «ё»

Буква ё пишется в случаях, когда возможно неверное чтение или понимание слова (в том числе в именах собственных).

Местоимение «вы»

Местоимение вы при обращении к посетителям сайта пишется со строчной буквы.

Названия институтов, кафедр, специальностей и направлений

Названия институтов пишутся с прописной буквы. Склоняются.

Названия кафедр пишутся без кавычек строчными буквами. Склоняются.

Оформите сложный список:

1. Акроним «вуз»

– Слово вуз (сокращение от «высшее учебное заведение») пишется строчными буквами. Склоняется.

2. Слово «онлайн»

– Слово онлайн пишется по-русски, со строчной буквы, без дефиса.

3. Слово «Интернет»

– Слово Интернет пишется по-русски, с прописной буквы, без кавычек. Склоняется.

Пункт 1

А. Пункт 1.1

Пункт 1.1.1

Пункт 1.1.2

Пункт 1.1.3

В. Пункт 1.2

о Пункт 1.2.1.

о Пункт 1.2.2.

Пункт 2

2. Пункт 2.1

I Пункт 2.1.1.

II Пункт 2.1.2.

III Пункт 2.1.3.

3. Пункт 2.2

• Пункт 2.2.1.

• Пункт 2.2.2.

Пункт 3.

Определение пункта 3

Другое определение пункта 3

2. Сохраните как *doc3.html*.

3. Просмотрите в браузере, если все отображается корректно, переходим к пункту 7, если нет – исправляем.

4. Маркированный список оформить с использованием символов квадрат и окружность.

5. Нумерованный список оформить с использованием символов А, В, С.

6. Сохранить как *doc4.html*.



Контрольные вопросы

1. Какие виды списков предусмотрены в HTML?
2. Как изменить вид маркера в маркированном списке?
3. Что понимают под нумерованным списком?
4. Каким образом изменить вид маркера в нумерованном списке?
5. Как изменить нумерацию в нумерованном списке?
6. Из каких частей состоит список с определениями?
7. Что собой представляет вложенный список?
8. Каким образом можно построить вложенный список?

Лабораторная работа № 4

ТАБЛИЦЫ

Цель работы: сформировать практические навыки по созданию таблиц; знать элементы создания таблиц; уметь изменять вид таблиц; использовать стилевые оформления при создании таблиц.

Время выполнения: 2 ч.



Теоретические сведения

Таблицы являются средством форматирования данных на web-странице. Удобство таблиц в HTML в том, что браузер берет на себя заботу о прорисовке рамки таблицы. При создании таблиц используется принцип вложения: внутри основного элемента таблицы <TABLE> создается ряд элементов, определяющих строки <TR>, а внутри этих элементов размещаются элементы для описания каждой ячейки в строке <TD>, <TH>.

Ячейка <TH> используется для оформления заголовочных ячеек – обычно содержимое такой ячейки выводится жирным шрифтом и выровнено по центру. Пример создания таблиц представлен на рис. 1.

HTML-код	Отображение в браузере
<pre><table border=1> <tr> <th>англ</th> <th>рус</th> </tr> <tr> <td>red</td> <td>красный</td> </tr> <tr> <td>green</td> <td>зелёный</td> </tr> </table></pre>	
<pre><table border=1> <tr> <th>англ</th> <th>рус</th> </tr> <tr> <td style='width: 150px'>red</td> <td>красный</td> </tr> <tr> <td>green</td> <td style='background-color: green'> зелёный </td> </tr> </table></pre>	

Рис. 1. Пример создания таблиц

<TABLE>...</TABLE> – таблица, парный тег. Атрибут **border** тега <TABLE> – толщина рамок.

Парный тег <TR> внутри тега <TABLE> – строка таблицы.

Парный тег <TD> внутри тега <TR> – ячейка в строке таблицы.

Парный тег <TH> внутри тега <TR> – ячейка в строке таблицы.

Таблица <TABLE> должна содержать одну или несколько строк <TR>.

Каждая строка <TR> должна содержать одну или несколько ячеек <TD> или <TH>.

Пример (рис. 2):

Афоризмики		
СПИСОК нумерованный	Список маркированный	Список определений

Рис. 2. Пример таблица HTML

Код:

```
<TABLE BORDER="1" >
<TR>
  <TD COLSPAN="3" >Афоризмики</TD>
</TR>
<TR>
  <TD>Список нумерованный</TD>
  <TD>Список маркированный</TD>
  <TD>Список определений </TD>
</TR>
</TABLE>
```

С помощью CSS-стилей можно красиво оформить таблицу в соответствии с ее содержимым.



Задание

1. Разработайте следующие виды таблиц:

№	Фамилия	Адрес
1		
2		
3		

№	Фамилия	Адрес
1		
2		
3		

простая
таблица

имя	рост	вес
нина	160	70
коля	170	80
вера	155	47

имя	рост	вес
нина	160	70
коля	170	80
вера	155	47

2. Добавьте выравнивание, сделайте заливку фона таблицы.

1111		
222	33	55
	44	
666		77
		88

3. Раскрасьте ячейки таблицы.

№	Фамилия	Адрес
1		
2		
3		

Меню ресторана «Елки зеленые»		
Горячее		Цена
Суп	200 г	3 руб.
Борщ	200 г	3 руб.
Второе		
Котлета	100 г	1 руб.
Каша	200 г	1 руб.
Компот	Стакан 200 г	1 руб.

4. Сохраните как *doc5.html*.

5. Просмотрите, убедитесь, что все отображается корректно.



Контрольные вопросы

1. Какие теги обязательны при создании HTML-таблицы?
2. С помощью какого свойства задаются границы таблицы?
3. В чем отличие тегов <TD> и <TH>?
4. С помощью каких атрибутов можно объединить ячейки по горизонтали и вертикали?
5. Как задать цвет фона ячеек таблицы?

Лабораторная работа № 5

ИЗОБРАЖЕНИЯ

Цель работы: получить навыки оформления web-страниц с изображениями.

Время выполнения: 2 ч.



Теоретические сведения

Вставка изображения на страницу: ``

Атрибуты:

Свойство **align**="параметр" – определяет выравнивание рисунка. Это значение также влияет на то, как текст будет обтекать рисунок. Может принимать следующие параметры:

left – выравнивание по левому краю.

middle – выравнивание середины изображения по базовой линии текущей строки.

bottom – выравнивание нижней границы изображения по окружающему тексту.

top – верхняя граница изображения выравнивается по самому высокому элементу текущей строки.

right – выравнивание по правому краю.

alt="текст" – подсказка / описание картинка. Выполняет сразу две важные функции: выдает подсказку при наведении. Если в браузере отключены изображения, то выводится этот текст.

title="подсказка" – выводит текст при наведении на картинку.

border="ЧИСЛО" – задает толщину рамки, обтекающей изображение. Указывается числом пикселей.

bordercolor="#000" – цвет рамки.

```

```

height="число" – задает высоту изображения либо в пикселях, либо в процентах. Например, если изображение имеет размер 400×200, а мы укажем высоту 150 пикселей, то оно сжимается до 300×150 (на 25% меньше), т. е. пропорционально.

Свойство **width**="число" – задает ширину изображения либо в пикселях, либо в процентах. Например, если изображение имеет размер

1000×800, а ширина указывается 1200 пикселей, то оно автоматически расширяется на 20% до 1200×960.

Свойство **hspace="число"** – задает горизонтальный отступ изображения в пикселях от других объектов html.

Свойство **vspace="число"** – задает вертикальный отступ изображения в пикселях от других объектов html.

Можно использовать стилевые свойства:

margin-top: X px; (X – отступ сверху).

margin-bottom: Y px; (Y – отступ внизу).

margin-left: L px; (L – отступ слева).

margin-right: R px; (R – отступ справа).

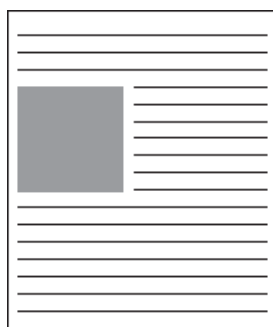
Пример, где картинка является ссылкой:

```
<a href="Куда_ведет_ссылка"></a>
 //округление углов.
```



Задание

1. Разработайте страницу, поместив на нее блок текста и изображение. Оформите следующим образом:

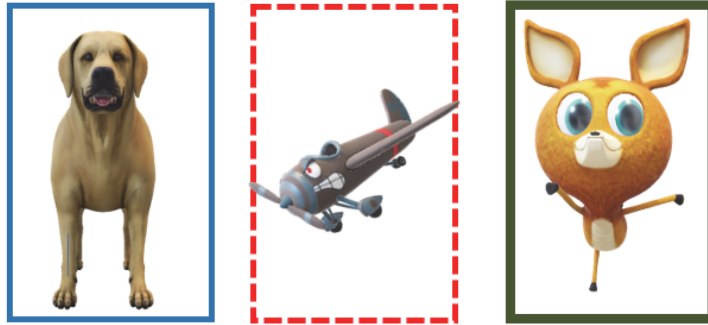


2. Разработайте страницу, поместив следующее:

Мои увлечения		
<i>Зима</i>	<i>Картинка с зимним пейзажем</i>	<i>1. Лыжи 2. Коньки 3. Скейтборд</i>
<i>Лето</i>	<i>Картинка с летним пейзажем</i>	– плавание – загорание – походы
<i>Осень</i>	<i>Картинка с летним пейзажем</i>	а) грибы б) рыбалка с) картошка

Примечание. Изображения должны быть одного размера.

3. Сделайте галерею:



Примечание. Блоки – одинаковые по ширине и высоте.
Картинки могут быть другие.
Позаботьтесь об альтернативном тексте
и всплывающих подсказках.



Контрольные вопросы

1. Обязательно ли закрывать тег ``?
2. Какой атрибут используется для выравнивания по левому краю?
3. Зависит ли правильное отображение рисунка от его местонахождения?

Лабораторная работа № 6

HTML. ВЕРСТКА СТРАНИЦЫ. БЛОКИ. СЕМАНТИКА

Цель работы: освоить приемы блочной и семантической верстки.
Время выполнения: 4 ч.



Теоретические сведения

Web-страница состоит из множества различных элементов, которые могут иметь сложную структуру, поэтому при ее создании возникает необходимость позиционировать эти элементы, стилизовать их так, чтобы они располагались на странице нужным образом. Возникает вопрос создания макета страницы и ее верстки.

Блочная верстка – это условное название способов и приемов верстки, при которой в большинстве web-страниц для разметки используется CSS-свойство **float**, а основным строительным элементом является элемент `<div>`, т. е. блок. Используя свойство **float** и элементы `<div>` или другие элементы, можно создать различные структуры страниц.

Пример создания двухколоночной web-страницы. Вверху и внизу будут стандартно шапка и футер, а в центре – две колонки: колонка с меню, или сайдбар, и колонка с основным содержанием.

В первую очередь определим все блоки. При работе с элементами, которые используют обтекание и свойство **float**, важен их порядок. Так, код плавающего элемента, у которого устанавливается свойство **float**, должен идти перед элементом, который обтекает плавающий элемент, т. е. блок сайдбара будет идти перед блоком основного содержания (рис. 4).

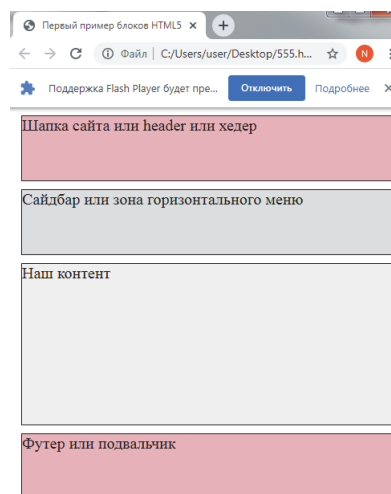
В HTML5 есть семантические теги для группировки контента конкретного типа.

Шапка сайта может помещаться в HTML5-тег `<header>`, а подвал – в похожий тег `<footer>`. Кроме того, существует еще множество других семантических тегов – `<nav>`, `<section>`, `<article>` и т. д. Данными тегами можно заменять обычные теги `<div>` для объединения элементов (рисунок).

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="unicod">
    <title>Первый пример блоков HTML5</title>
    <style>
      div{
        margin: 10px;
        border: 1px solid black;
        font-size: 20px;
        height: 80px;
      }
      #header{
        background-color: #EBB3BC;
      }
      #sidebar{
        background-color: #ddd;
      }
      #main{
        background-color: #eee;
        height: 200px;
      }
      #footer{
        background-color: #EBB3BC;
      }
    </style>
  </head>
  <body>
    <div id="header">Шапка сайта или header или хедер</div>
    <div id="sidebar">Сайдбар или зона горизонтального меню</div>
    <div id="main">Наш контент</div>
    <div id="footer">Футер или подвальчик</div>
  </body>
</html>

```



Пример блочной верстки



Задание

1. Сверстайте страницу по макету:



В шапке разместите фамилию, телефон, значки социальных сетей. В зоне контента – рисунок и текст. В зоне меню – список из 4 позиций. В подвале – знак копирайтера, БГТУ, Издательское дело.

2. Изучите правила семантической верстки. Сверстайте макет по образцу:

```
<html>
<style>
header{
  text-align:center;
  background:#ffFA5F;
}
nav {
  float:left;
  width:200px;
  background:#FFAA00;
}
nav h2 {
  text-align:center
}
section {
  float:right;
  margin:10px auto;
  width:1250px;
  background:#0000ff
}
section h1 {
  color:#FFFFFF;
  padding:10px;
  text-align:center;
}
article {
  float:left;
  width:595px;
  margin:5px;
  padding-left:5px;
  text-align:center;
  background:#7BE7BE;
}
article h2 {
  width:250px;
  margin:10px auto;
  text-align:center;
  background:#FFFFFF;
}
footer {
  clear:both;
  height:40px;
  padding:10px;
  background:#8B4;
}
footer p {
  font-size:14px;
  text-align:center;
}
</style>
<header>
  <h1>Сайт о Ерунде</h1>
</header>
<nav>
  <h2>Меню Навигации</h2>
  <ul>
    <li><a href="#">Пойди туда</a></li>
    <li><a href="#">Пойди отсюда</a></li>
  </ul>
</nav>
<section>
  <h1>Блок о ерунде</h1>
  <article>
    <h2>Ерунда- это наше все</h2>
    <p>Ох, какая ерунда, эта ваша ерунда..</p>
  </article>
  <article>
    <h2>А еще есть ерунда</h2>
    <p>Вот это да...</p>
  </article>
</section>
<footer>
  <p>Авторские права на ерунду</p>
</footer>
```

Пример семантической верстки



Контрольные вопросы

1. Что такое блочная верстка?
2. Какой тег является основным строительным элементом веб-страниц?
3. Какие семантические теги в HTML5 вы знаете?

Лабораторная работа № 7

ГИПЕРССЫЛКИ. НАВИГАЦИЯ

Цель работы: освоить способы создания вертикального и горизонтального меню с гиперссылками.

Время выполнения: 4 ч.



Теоретические сведения

Навигация должна быть на любом сайте, даже если он одностраничный. Навигационные ссылки могут вести на различные страницы / разделы сайта либо перенаправлять к закладке (якорю), расположенной на текущей странице.

Навигация – это набор ссылок, упорядоченных и сгруппированных по смыслу. Различают *горизонтальное*, *вертикальное навигационное* меню. Меню, как правило, создается с использованием тега списка ``, где в каждом пункте `` содержится одна ссылка `<a>`. В HTML5 для навигации применяется тег `<nav>`, в который помещают теги ссылок.

Алгоритм создания меню

На HTML создаем каркас меню, а стилизация его происходит с помощью CSS. Стили задают вид меню.

Горизонтальное меню

Алгоритм создания горизонтального меню такой же, но только свойству **display** для элементов `` надо присвоить значение **inline**, чтобы пункты списка располагались друг за другом.

```
#navbar li {display: inline;}
```

Вертикальное меню

1. Создайте маркированный список, добавив к нему атрибут **id** с идентификатором **"navbar"**.

```
<ul id="navbar">
  <li><a href="#">Услуги</a></li>
  <li><a href="#">Акции</a></li>
```

```
<li><a href="#">Контакты</a></li>
<li><a href="#">Вакансии</a></li>
</ul>
```

2. Сбросьте установленные стили списка по умолчанию и задайте требуемую ширину.

```
#navbar {
  margin: 0;
  padding: 0;
  list-style-type: none;
  width: 150px;
}
```

3. Добавьте к ссылкам фоновый цвет, цвет текста, размер и насыщенность шрифта, уберите подчеркивание, добавьте небольшие отступы и переопределите отображение элемента `<a>` со строчного на блочный.

4. Добавьте левую и нижние рамки к пунктам списка. Сделайте их блочными. В этом случае ссылка занимает все доступное пространство пунктов списка, т. е. для перехода по ссылке больше не надо наводить курсор точно на текст.

```
#navbar a {
  background-color: #559444;
  color: #fff;
  padding: 15px;
  text-decoration: none;
  font-weight: bold;
  border-left: 5px solid #55AFFF;
  display: block;
}
```

5. Если необходимо изменить внешний вид ссылки при наведении на нее курсором, определите псевдокласс **:hover**.

```
#navbar a:hover {
  background-color: #666;
  border-left: 5px solid #3333FF;
}
```

Выпадающее меню

Алгоритм такой же, но применяем список в списке, комбинируя горизонтальное и вертикальное меню.

```
<ul id="navbar">
  <li><a href="#">Услуги</a></li>
```

```

<li><a href="#">Контакты</a></li>
<li><a href="#">Акции</a>
  <ul>
    <li><a href="#">Сегодня</a></li>
    <li><a href="#">В ноябре</a></li>
    <li><a href="#">Новогодние</a></li>
  </ul>
</li>
<li><a href="#">Вакансии</a></li>
</ul>

```

В CSS коде:

a) скрыть список с подпунктами, используя **display: none;**, чтобы они не отображались на web-странице все время.

b) при наведении на элемент `` список снова будет преобразован в блочный элемент:

```

#navbar ul {display: none;}
#navbar li:hover ul {display: block;}

```

c) убрать у обоих списков отступы и маркеры, установленные по умолчанию. Элементы списка с навигационными ссылками сделать плавающими, формируя горизонтальное меню, но для элементов списка, содержащих подпункты, задать **float: none;**, чтобы они отображались друг под другом.

```

#navbar, #navbar ul {
  margin: 0;
  padding: 0;
  list-style-type: none;
}
#navbar li {float: left;}
#navbar ul li {float: none;}

```

d) для того чтобы выпадающее подменю не смещало контент, расположенный под панелью навигации, вниз, задать пунктам списка **position: relative;**, а списку, содержащему подпункты, **position: absolute;** и добавить свойство **top** со значением 100%, чтобы абсолютно позиционированное подменю отображалось точно под ссылкой.

```

#navbar ul {
  display: none;
  position: absolute;
  top: 100%;
}

```

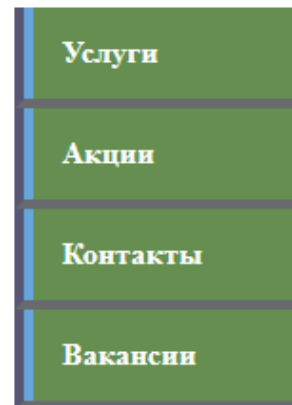
```
#navbar li {
    float: left;
    position: relative;
}
#navbar {height: 30px;}
```

Высота для родительского списка введена для того, чтобы он не был проигнорирован браузером, и контент, следующий за списком, обтекал меню. Браузеры не учитывают в качестве содержимого элемента плавающий контент.



Пример 1. Вертикальное меню

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>документ</title>
    <style>
      #navbar {
        margin: 0;
        padding: 0;
        list-style-type: none;
        width: 150px;
      }
      #navbar li {
        border-left: 5px solid #557;
        border-bottom: 5px solid #666;
      }
      #navbar a {
        background-color: #559444;
        color: #fff;
        padding: 15px;
        text-decoration: none;
        font-weight: bold;
        border-left: 5px solid #55AfFF;
        display: block;
      }
    </style>
  </head>
  <body>
    <ul id="navbar">
      <li><a href="#">Услуги</a></li>
      <li><a href="#">Акции</a></li>
```




```

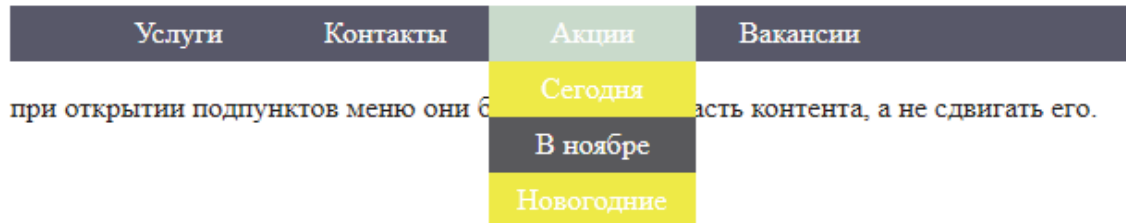
        <li><a href="#">Контакты</a></li>
        <li><a href="#">Вакансии</a></li>
    </ul>

</body>
</html>

```



Пример 2. Выпадающее меню



```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Документ</title>
    <style>
      #navbar ul{
        display: none;
        background-color: #ff0;
        position: absolute;
        top: 100%;
      }
      #navbar li:hover ul {display: block;}
      #navbar, #navbar ul{
        margin: 0; padding: 0;
        list-style-type: none;
      }
      #navbar {
        height: 30px;
        background-color: #556;
        padding-left: 35px;
        min-width: 480px;
      }
      #navbar li {
        float: left;
        position: relative;
        height: 100%;
      }
      #navbar li a {
        display: block;

```

```

padding: 6px;
width: 100px;
color: #fff;
text-decoration: none;
text-align: center;
}
#navbar ul li {float: none;}
#navbar li:hover {background-color: #cdc;}
#navbar ul li:hover {background-color: #555;}
</style>
</head>
<body>

<ul id="navbar">
  <li><a href="#">Услуги</a></li>
  <li><a href="#">Контакты</a></li>
  <li><a href="#">Акции</a>
    <ul>
      <li><a href="#">Сегодня</a></li>
      <li><a href="#">В ноябре</a></li>
      <li><a href="#">Новогодние</a></li>
    </ul>
  </li>
  <li><a href="#">Вакансии</a></li>
</ul>
  <p>при открытии подпунктов меню они будут скрывать часть
контента, а не сдвигать его.</p>
</body>
</html>

```



Пример 3. Оформление навигации в стиле семантической верстки

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="unicod">
    <title>Документ</title>
  </head>
  <body>
    <header>
      <h1>пример</h1>
    </header>
    <nav>
      <a href="page/1.html">Услуги</a> |
      <a href="page/2.html">Акции</a> |

```

```

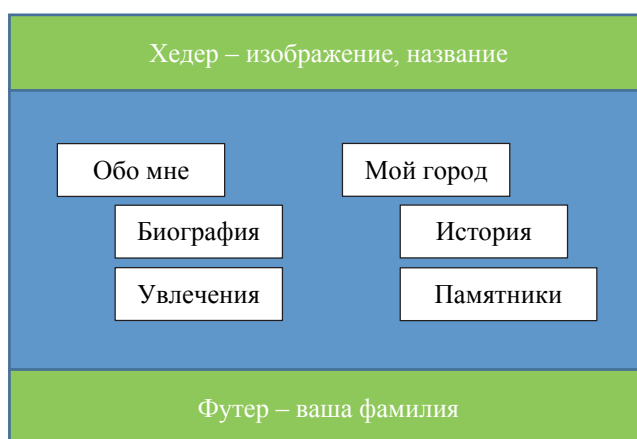
        <a href="page/3.html">Контакты</a> |
        <a href="page/4.html">Вакансии</a>
    </nav>
    <article>
        <h2>Добро пожаловать!</h2>
    </article>
</body>
</html>

```



Задание

1. Сверстайте страницу согласно приведенному макету (рисунок).



Макет для верстки

На страницах *Биография*, *Увлечения*, *История*, *Памятники* расположить текст, изображения, применить стилевое оформление для элементов страниц.

При верстке страницы:

- использовать блочную верстку;
- использовать семантическую верстку.

Отчет

Представить два сверстанных макета в различных стилях верстки.



Контрольные вопросы

- Перечислите методы верстки сайта.
- Какая разница между применением табличной и блочной верстки при отображении контента на странице?

3. Можно ли использовать фреймы при верстке сайта?
4. Дайте определение понятию «навигация на сайте».
5. Приведите алгоритм создания меню.
6. С помощью какого свойства можно изменить вид меню при наведении на него курсором мыши?
7. Для чего применяют блочное оформление для ссылок?
8. Что надо задать, чтобы список не был проигнорирован браузером и контент, следующий за списком, обтекал меню?
9. Почему мы сбрасываем свойства списка по умолчанию при использовании его в меню?
10. Для чего задаем пунктам списка **position: relative**?

Лабораторная работа № 8

АНИМАЦИЯ И ТРАНСФОРМАЦИЯ НА CSS3

Цель работы: получить базовые знания по использованию трансформаций и анимаций на CSS3.

Время выполнения: 2 ч.



Теоретические сведения

Преобразования в CSS представляют собой набор функций, которые позволяют определенным образом придавать элементу форму:

translate – перемещает элемент вдоль трех осей (x , y и z).

rotate – вращает элемент вокруг центральной точки.

scale – изменяет размер элемента.

skew – искажает элемент.

Свойства трансформации

Есть три доступных свойства для трансформации:

transform определяет, какая функция будет применяться (`translate`, `rotate`, `scale` и др.);

transform-origin позволяет изменять точку начала преобразования (работает как `background-position`);

transform-style для настройки 3D.

Обратите внимание, что в отличие от **background** и **border**, свойство **transform** не является сокращенным.

transform не прерывает поток независимо от вращения, масштабирования или перемещения, они не будут влиять на другие элементы.

Функция **translate()** позволяет перемещать элемент в плоскости (по осям x и y). Она принимает либо:

– один параметр: перемещает элемент вдоль оси x ;

– два параметра: первое значение для оси x , второе – для оси y .

Это похоже на использование относительного позиционирования с помощью значений **left** и **top**:

```
р {  
  animation: translating 4s linear 0s infinite;
```

```

    background: turquoise;
    color: white;
    height: 60px;
    line-height: 60px;
    margin: 0;
    position: absolute;
    text-align: center;
    width: 60px;
}

@keyframes translating {
    0% {transform: translate(0, 0);}
    25% {transform: translate(240px, 0);}
    50% {transform: translate(240px, 140px);}
    75% {transform: translate(0, 140px);}
    100% {transform: translate(0, 0);}
}
p {animation: translating 4s linear infinite;}

```

Функция **rotate()** позволяет вращать элемент вокруг неподвижной точки. По умолчанию вращение происходит вокруг центра элемента. Функция **rotate()** принимает только один параметр, который является значением угла и определяется в градусах (**deg**), градах (**grad**), радианах (**rad**) или в оборотах (**turn**) (один оборот эквивалентен полному кругу).

HTML:

```
<p> привет </p>
```

CSS:

```

p {
    animation: rotating 4s linear 0s infinite;
    background: slateblue;
    border-radius: 60px;
    color: white;
    height: 120px;
    line-height: 120px;
    margin: 0;
    text-align: center;
    width: 120px;
}

@keyframes rotating {
    0% {
        transform: rotate(0deg);
    }
    100% {
        transform: rotate(360deg);
    }
}

```

Функция **scale()** позволяет изменять размер элемента. Она может увеличить или уменьшить элемент. Функция принимает либо:

- один параметр: изменение размеров элемента одинаково по высоте и ширине;
- два параметра: первое значение изменяет размер элемента по горизонтали, второе – по вертикали.

Диапазон возможных значений:

- 1: элемент сохраняет свой первоначальный размер;
- 2: элемент удваивается в размере;
- 0.5: элемент уменьшается наполовину;
- 0: элемент в основном исчезает (так как его высота и ширина равны нулю);
- 1: элемент отражается.

```
@keyframes scaling {
  0%   { transform: scale(1); }
  20%  { transform: scale(2); }
  40%  { transform: scale(0.5); }
  60%  { transform: scale(0); }
  80%  { transform: scale(-1); }
  100% { transform: scale(1); }
}
p { animation: scaling 10s steps(1) 0s infinite; }
```

Функция **skew()** позволяет исказить элемент, сдвигая его стороны вдоль линий. Эта функция преобразования используется редко, поскольку ее последствия весьма непредсказуемы, а результат не обязательно привлекателен. Тем не менее давайте посмотрим, как это работает.

Как и **scale()**, функция **skew()** принимает либо:

- один параметр: элемент искажается по горизонтали;
- два параметра: первое значение искажает элемент по горизонтали, второе – по вертикали.

Как и **rotate()**, функция **skew()** принимает только значения угла, такого как градусы (**deg**).

```
@keyframes skewing {
  0%   { transform: skew(0deg); }
  20%  { transform: skew(10deg); }
  40%  { transform: skew(45deg); }
  60%  { transform: skew(90deg); }
  80%  { transform: skew(120deg); }
  100% { transform: skew(0deg); }
}
p { animation: skewing 10s steps(1) 0s infinite; }
```

Подробнее: <https://webref.ru/course/css-advanced/transform#content>.



Задание

1. При наведении курсора мыши происходит трансформация блока:



2. Используя свойство skew, оформите страницу со сдвигом текста:

Быть энтузиасткой сделалось ее общественным положением, и иногда, когда ей даже того не хотелось, она, чтобы не обмануть ожиданий людей, знавших ее, делалась энтузиасткой. Сдержанная улыбка, игравшая постоянно на лице Анны Павловны, хотя и не шла к ее отжившим чертам, выражала, как у избалованных детей, постоянное сознание своего милого недостатка, от которого она не хочет, не может и не находит нужным исправляться. В середине разговора про политические действия Анна Павловна разгорячилась. — Ах, не говорите мне про Австрию! Я ничего не понимаю, может быть, но Австрия никогда не хотела и не хочет войны! Она прельстят нас Россия одна

ее ароматический йодистый запах
вдоль и менижолоп манявтэршю
мётр ,но ,аеетох эн отот жакд йе влтой
ее хншванс йедоп йннаджко атунамбо эн
ваннаждс Йодистый запах
мннА эшп ан оннеотсоп рашаврн ладмт
мншанато ее х влш эн н ктох ,мановяП
хншванс лобен ү жек ,вллжрмн мтцэр
отолнн отэовс эннннгоо эоннеотсоп йэтэд
эн ,тэрех эн яно оторото то ,латвтэодэн
ясттвллвдпэн мннннн тндохэн эн н тэжом
энлсэрнтнгоп оуп вровотлвд энлсэрэс В
— ,эблнркротлвд яновояП янна внэтсйед
R !ондртвА оуп эннн этндровот эн жА
он ,атид тэжом ,оамнноп эн отэртн
тэрех эн н влэтэх эн влтоинн вндртвА
вннн вннннн Роллн лан тэвгач вннн шнннн

3. Организуйте вращение 3–4 шариков по кругу. При движении шарики меняют цвет и прозрачность:



Контрольные вопросы

1. Для чего используются CSS-трансформации?
2. Какая функция отвечает за вращение элемента вокруг центральной точки?
3. Какие свойства для трансформации вы знаете?
4. Для чего предназначена функция **skew()** и какие параметры она принимает?
5. С помощью какого правила задаются ключевые кадры анимации?
6. Как применить созданную анимацию к элементу на странице?

Лабораторная работа № 9

ВВОД И ВЫВОД ДАННЫХ В JAVASCRIPT. ВЫЧИСЛЕНИЯ

Цель работы: получить базовые знания по использованию языка программирования JavaScript.

Время выполнения: 2 ч.



Теоретические сведения

JavaScript – инструмент, предоставляющий web-дизайнерам возможности программирования и создания интерактива на страницах. JavaScript добавляется на web-страницы с помощью тега `<script>`.

В следующем примере на страницу выводится текст «Я научусь работать с JavaScript».

```
<script >
    document.write("Я научусь работать с
    JavaScript.");
</script>
```

JavaScript код также может храниться во внешнем текстовом файле с расширением `.js` и подключаться к страницам с помощью тега `<script>` следующим образом:

```
<script type="text/javascript" src="ex.js"></script>
```

JavaScript чувствителен к регистру букв.

Комментарии обозначаются `//` в начале строк в случае однострочного комментария. Многострочные комментарии начинаются с `/*` и заканчиваются `*/`.

Ввод и вывод данных

В JavaScript предусмотрены три стандартных метода для ввода и вывода данных: `alert()`, `prompt()` и `confirm()`.

Метод `alert()` позволяет выводить диалоговое окно (окно оповещения) с заданным сообщением и кнопкой ОК.

Метод **confirm()** позволяет вывести диалоговое окно (окно подтверждения) с сообщением и двумя кнопками – ОК и Отмена (Cancel). Используется в случаях, когда необходимо, чтобы пользователь подтвердил или отклонил что-либо.

Метод **prompt()** позволяет вывести на экран диалоговое окно (окно запроса) с сообщением, а также с текстовым полем, в которое пользователь может ввести данные. Используются в случаях, если от пользователя необходимо получить определенную информацию.

Арифметические операторы

Арифметические операторы используются для выполнения арифметических операций над переменными или значениями (табл. 1).

Таблица 1

Арифметические операторы

Оператор	Название	Пример
+	Сложение	$X + Y$
-	Вычитание	$X - Y$
*	Умножение	$X * Y$
/	Деление	X / Y
%	Остаток от деления	$X \% Y$
++	Увеличение на 1	$X++$
--	Уменьшение на 1	$Y--$

Операторы сравнения (табл. 2).

Таблица 2

Операторы сравнения

Оператор	Название	Пример
==	Равно	$X == Y$
!=	Не равно	$X != Y$
>	Больше, чем	$X > Y$
>=	Больше или равно	$X >= Y$
<	Меньше, чем	$X < Y$
<=	Меньше или равно	$X <= Y$

Логические операторы

Логические данные, обычно получаемые с помощью элементарных выражений, содержащих операторы сравнения, можно объединять в более сложные выражения. Для этого используются логические (булевы) операторы: **И (&&)**, **ИЛИ (||)**, **НЕ (!)**. Выражения с логическими операторами возвращают значение true или false.

Операторы условного перехода

Оператор условного перехода **if** позволяет реализовать структуру условного выражения если ..., то ..., иначе ...

Пример оператора if перехода:

```
if (sr_ball>=5) {  
    document.write("Стипендия есть");  
}  
else {  
    document.write("Стипендии нет");  
}
```

В фигурных скобках располагается блок кода – несколько выражений. Если в блоке используется одна команда, то фигурные скобки можно не писать.

Вызов математических функций осуществляется так **Math.sin(x)**.



Задание

Вариант – ваш номер по списку.

Вычисления оформить с применением условного оператора.

Результат вывести на страницу.

Вариант 1 $\frac{x^3 - 6x + 11}{x^2 - 8x + 12}, \text{ если } x \leq 6$	Вариант 2 $Y = \sin x + \exp(x),$ если x не равно π
Вариант 3 Вычислить площадь квадрата, и если $a > 5$ и площадь круга в противном случае	Вариант 4 $\frac{\sqrt{x} + 11}{x^2 - 7x + 10a},$ если x и a не равны 0
Вариант 7 $(\sin 3a + 11)/(x^2 - 8x),$ если $a \leq 2$ и x не равно 0	Вариант 8 $\sin x \cos y / 2\pi,$ если $x > 10$
Вариант 9 $\frac{x}{y} \sqrt{xy}, \text{ если } x \leq 1 \text{ и } y \text{ не равно } 0$	Вариант 10 $\operatorname{tg} x / \sqrt{x}, \text{ если } x \text{ не равно } 2$
Вариант 11 $\sqrt{xa} \sqrt{ya}, \text{ если } x > a^2$	Вариант 12 $\sin x \cos^{-1} y, \text{ если } \sin x \leq \pi$



Пример программы

```
<html>
<head>
  <meta charset="utf-8">
</head>
<body>
  <p>Начало документа...</p>
  <script>
    x = +prompt(' Введи x');
    if (x>8) y=Math.sin(x);
    else y=Math.cos(x);
    document.write("результат==", y );
  </script>
  <p>...Конец документа</p>
</body>
</html>
```

Отчет

Страница с разработанными скриптами. На страницу поместить задание.



Контрольные вопросы

1. Какой тег используется для добавления JavaScript на web-страницу?
2. Как создать однострочный комментарий в JavaScript?
3. Для чего предназначен метод **confirm()**?
4. Какие операторы сравнения вы знаете?
5. Что возвращают логические операторы **И (&&)**, **ИЛИ (||)**, **НЕ (!)**?
6. Для чего предназначен условный оператор **if**?
7. Как осуществляется вызов математических функций в JavaScript?

Лабораторная работа № 10

ФУНКЦИИ И СОБЫТИЯ

Цель работы: ознакомиться с работой функций и событий, освоить примеры работы с ними.

Время выполнения: 4 ч.



Теоретические сведения

Функции являются фундаментальными блоками JavaScript.

Функция – набор операторов, выполняющих специфическую задачу.

Для использования функции ее необходимо сначала определить.

В JS различают функции:

- именованные;
- анонимные;
- стрелочные.

Именованная функция

```
function имя(параметры) {  
    return выражение  
}
```

Пример: определить функцию, возвращающую сумму 2 чисел.

```
var a = +prompt ("введи a");  
var b = +prompt ("введи b");  
var s = sum(a,b); // вызов функции с фактическими параметрами  
alert(s);  
    function sum(x,y) { // определение функции  
        return x+y;  
    }
```

Анонимная функция

```
var имя = function(параметры) {  
    return значение  
}
```

Пример:

```
var sum = function(x,y) {
    return x+y
}
var a = prompt ("");
var b = prompt ("");
var s = sum(a,b) // вызов анонимной функции
```

Стрелочная функция

```
var имя =(параметры) => результат
```

Пример:

```
var sum =(x,y) => x+y;
```

Вызов:

```
var a = prompt ("");
var b = prompt ("");
var s = sum(a,b)
```

События

Практически все JavaScript-приложения выполняют те или иные действия, откликаясь на различные события.

Событие – это сигнал от браузера о том, что что-то произошло.

Различают события:

1. DOM-события, которые инициируются элементами DOM. Например, событие **click** происходит при клике на элементе, а событие **mouseover**, когда указатель мыши появляется над элементом.

2. События окна. Например, событие **resize** – при изменении размера окна браузера.

3. Другие события, например, **load**, **readystatechange**.

Именно DOM-события связывают действия, происходящие в документе, с кодом JavaScript, тем самым обеспечивая динамический web-интерфейс.

Назначение обработчиков

Для того чтобы скрипт реагировал на событие, нужно назначить функцию-обработчик. Обычно обработчики называют "**он+имя события**", например, **onclick**.

Обработчики всегда выполняются последовательно и в общем потоке. Это значит, что при установке обработчиков двух событий, которые возникают на элементе одновременно, например **mouseover** (мышь появилась над элементом) и **mousemove** (мышь двигается над элементом), их обработчики будут выполнены последовательно.

Существует несколько способов назначать обработчик на конкретное событие элемента.

1. Через атрибут HTML-тега

Обработчик события можно указать в виде inline-записи прямо в атрибуте *онсобытие*. Например, для обработки события **click** на кнопке **input** можно назначить обработчик **onclick** вот так:

```
<input id="b1" value="Нажми меня" onclick="alert('Спасибо!');" type="button"/>
```

Пример как назначить функцию:

```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="utf-8">
<script>
    function weather () {
        alert("Сегодня солнечно");
    }
</script>
</head>
<body>
<input type="button" onclick="weather()" value="Погода за
окном"/>
</body>
</html>
```

2. Через свойство объекта

Установка функции-обработчика через свойство *онсобытие* соответствующего элемента. Этот способ также работает в любом браузере с поддержкой JavaScript.

Для этого нужно:

1. Получить элемент **document.getElementById**.
2. Назначить обработчик свойству *он+имя*.

Пример установки обработчика события **click** на элемент с `id="myEl"`:

```
document.getElementById('myEl').onclick = function() {
```

```
        alert('Спасибо')
    }
    <input id="myEl" type="button" value="Нажми меня"/>
```

При нажатии на кнопку с `id= myEl` будет вызвана функция

```
function() {
    alert('Спасибо')
}
```

Сравните это с атрибутом. Там, наоборот, скобки нужны:

```
<input type="button" id="mybutton" onclick="doSomething()" />
```

Такое различие легко объяснить. Дело в том, что при назначении **onclick** в HTML браузер автоматически создает функцию-обработчик из содержимого кавычек. Получается, что последний пример – это, по сути, то же самое, что

```
document.getElementById('mybutton').onclick = function() {
    doSomething() // внутри автосозданной функции
}
```

Пример – установка обработчика на событие «содержимое окна загрузилось»:

```
window.onload = function() {
    alert('Документ загружен!')
}
```



Задание

1. По примеру создайте кнопку с именем «Приветствие», при нажатии на которую появляется вопрос «Как твоё имя», а затем всплывает сообщение «Приветствую тебя, *Имя*».

2. Создайте страницу, разместите на ней три абзаца текста, картинку. Назначьте им различные `id`.

Разместите кнопки ШРИФТ, ФОН, РАЗМЕР, при нажатии на которые будут меняться размер шрифта, фон страницы, размер картинки.

Например, строка поменяет цвет фона:

```
document.body.style.background = 'red';
```


Можно назначить событие напрямую элементу. Например, параграфу назначено изменение размера шрифта по событию **OnClick**.

```
<p Onclick="this.style.fontSize='24px'">Это просто текст.  
Для увеличения нажмите на него.</p>
```

3. Сделайте так, чтобы при нажатии на одно слово в параграфе изменялся цвет и размер шрифта этого слова (*команды в событии Onclick можно перечислить через точку с запятой*).



Контрольные вопросы

1. Что такое функция в JavaScript?
2. Какие функции бывают?
3. Что такое событие в JavaScript и какие они бывают?
4. Как назначить обработчик на конкретное событие элемента?
5. Каким образом получить элемент web-страницы по его **id**?

Лабораторная работа № 11

ФОРМЫ

Цель работы: научиться создавать на web-странице формы, изучить возможные элементы формы и их свойства, а также способы доступа к элементам формы.

Время выполнения: 2 ч.



Теоретические сведения

HTML формы могут содержать такие элементы ввода, как текстовые поля, флажки, радиокнопки, кнопки отправления и др.

Формы также могут содержать списки выбора, многострочные текстовые поля, метки и др.

Для создания формы в HTML применяется тег `<form>`.

Элементы ввода используются для приема пользовательских данных. Они отличаются друг от друга в зависимости от значения атрибута **type**.

Текстовое поле `<input type="text"/>` определяет однострочное текстовое поле, в которое пользователь может вводить различную информацию.

Поле пароля `<input type="password"/>` определяет поле для ввода пароля.

Флажок `<input type="checkbox"/>` дает возможность пользователям выбирать несколько пунктов с предварительно заполненной информацией из группы.

Радиокнопка `<input type="radio"/>` позволяет выбрать только один пункт с предварительно заполненной информацией из группы.

Кнопка отправления `<input type="submit"/>` предназначена для отправления на сервер введенных данных. Адрес, на который будут пересылаться данные формы, указывается в атрибуте тега `form` – **action**. Если данный атрибут отсутствует, данные будут отправлены на текущую страницу.

Текстовая область `<textarea>` позволяет ввести многострочный текст.

Список `<select>` создает выпадающий список, элементы которого определяются с помощью тега `<option>`. Атрибут **multiple** указывает, что в выпадающем списке могут быть выбраны одновременно несколько элементов.

Заголовки в формах `<fieldset>` позволяют сгруппировать желаемую часть формы и затем с помощью тега `<legend>` установить желаемое заглавие.

В HTML5 представлены несколько новых типов элемента **input**. Они могут быть использованы для создания полей ввода, которые будут принимать только определенный вид данных: `color`, `date`, `datetime`, `datetime-local`, `email`, `month`, `number`, `range`, `search`, `tel`, `time`, `url`, `week`.

Для применения одного из новых типов следует указать его название в качестве значения у атрибута **type**:

```
<input type="email"/>
```

Некоторые типы полей (например, «email» и «tel») приводят к открытию на мобильных устройствах специальных клавиатур с ограниченным набором клавиш.

Добавление атрибута «**required**» для элемента `input`, `select` или `textarea` сообщит браузеру о том, что у данного поля должно быть значение. Воспринимайте это как красную звездочку*, которую мы видим в большинстве регистрационных форм.

Когда вы указываете атрибут **required** для полей типа `email` или `url`, браузер ожидает наличия определенного шаблона, по которому он сможет проверить информацию, но подобная проверка очень снисходительна и пропускает e-mail адреса вроде «z@zz».

Использование следующих семантических атрибутов для валидации может быть очень полезным:

1) **required** – у него не может быть значения. Просто теги, которые использует этот атрибут, не должны быть пустыми.

```
<input type="text" name="text" class="mytext" required>
```

Он может быть применен в следующих `input`: `url`, `email`, `text`, `password`, `date`, `month`, `week`, `tel`, `search`, `select`, `textarea`, `file`, `checkbox`, `time`, `number`;

2) **maxlength** – этот атрибут позволяет установить максимальную длину вводимого текста для текстового поля ввода. **Maxlength** может быть использован в следующих `input`: `textarea`, `password`, `url`, `tel`, `text` и `search`.

В этом `textarea` лимит символов будет до 350.

```
<textarea name="message" id="message" cols="40" rows="6"
maxlength="350"></textarea>
```

Maxlength не покажет ошибку, но браузер не даст возможность пользователю ввести больше указанного числа символов.

3) **max**, **min** – атрибуты **min** и **max** указывают диапазон между минимальным и максимальным номерами. Используется в следующих `input`:

date, time, week, range, number и month. Следующий пример показывает возможный диапазон возраста:

```
<input type="age" name="age" min="10" max="18">
```

Если пользователь введет другой возраст, то получит сообщение об ошибке, при том что этот запрос не будет отправлен на сервер;

4) **step** – атрибут **step** можно использовать для числового интервала. В следующем примере в input number указываем минимальный и максимальный год и добавляем **step="4"**. Это означает, что при каждом изменении значение будет меняться с шагом 4 года:

```
<input type="number" name="leapyear" min="1972" max="2020" step="4">
```

5) **pattern** – атрибут паттерн использует регулярное выражение для валидации этого поля. Регулярное выражение представляет собой заранее определенный набор символов, которые образуют определенный шаблон. Используется для поиска строк, которые соответствуют шаблону, либо для обеспечения определенного формата, определенного шаблона.

Приведенный ниже пример требует от пользователей ввести пароль, который содержит минимально 8 символов и включает, по крайней мере, одну букву и одну цифру:

```
<input type="password" name="password" required pattern="^(?=.*[A-Za-z])(?=.*\d)[A-Za-z\d]{8,}$">
```

Пример формы (рисунок):

Данные о пользователе

Имя:

Фамилия:

Отчество:

ваш пол:

Мужской
 Женский

Анкета:

Какую специальность Вы считаете перспективной?

Издательское дело
 Химия
 Программирование
 Дизайн электронных и веб-изданий

Пример формы

Код для создания формы:

```
<html>
  <form>
    <fieldset>
      <legend>Данные о пользователе</legend>
      Имя: <br/><input type='text' name='firstname' /><br/>
      Фамилия:<br/> <input type='text' name='lastname' /><br/>
      Отчество: <br/><input type='text' name='lastname' /><br/>
      <p>ваш пол:</p>
      <input type='radio' name='s' value='m' /> Мужской<br/>
      <input type='radio' name='s' value='f' /> Женский
    </fieldset>
    <fieldset>
      <legend> Анкета: </legend>
      <p>Какую специальность Вы считаете перспективной?</p>
      <input type='checkbox' name='space' value='1' />
      Издательское дело <br/>
      <input type='checkbox' name='space' value='2' />
      Химия <br/>
      <input type='checkbox' name='space' value='3' />
      Программирование <br/>
      <input type='checkbox' name='space' value='3' />
      Дизайн электронных и веб-изданий <br />
    </fieldset>
  </form>
</html>
```

Доступ к элементам формы

С помощью JavaScript можно отправить данные формы на сервер, очистить ее, а также получить доступ к любому элементу формы для изменения его значений. Существует несколько способов обращения к формам с использованием объектной модели.

При создании формы автоматически создается массив **forms**. Для доступа к форме требуется указать номер элемента или имя формы, заданное параметром **name**.

```
<html>
<body>
  <form name=data>
    ...
  </form>
<script>
  alert (document.forms.length) // количество форм на
  странице
  alert (document.forms[0].name) // имя первой формы
```

```

    alert(document.forms.data.length) // количество элементов
в форме с именем data
    alert(document.forms["data"].length) // То же самое
    </script>
</body>
</html>

```

Нумерация элементов массива всегда начинается с нуля, поэтому обращение к первой форме будет **document.forms[0]**, ко второй – **document.forms[1]**.

Второй способ – через семейство **all**. Обращение к форме происходит как к элементу массива с именем, совпадающим с именем формы, – **document.all["data"]** или напрямую – **document.all.data**.

```

<html>
<body>
    <form name=data>
        ...
    </form>
<script>
    alert(document.all["data"].length)
    alert(document.all.data.name)
</script>
</body>
</html>

```

Обращение к элементам формы осуществляется посредством семейства **elements** или напрямую по имени элемента:

```

<html>
<body>
    <form name=data> <input type=text name=userName
value="Введите ваше имя">
    </form>
<script>
    alert(document.forms.data.elements.length) // Общее
число элементов в форме
    alert(document.forms[0].elements[0].value) // Зна-
чение первого элемента
    alert(document.forms["data"].userName.value) //
Значение элемента с именем userName
</script>
</body>
</html>

```

Нумерация массива **elements**, как и в случае с семейством **forms**, ведется с нуля, поэтому обращение к первому элементу формы

будет `elements[0]`. Для большого количества данных в форме значения элементов лучше получать по их имени.

Доступ к элементам формы можно также осуществить по их ID с помощью метода `document.getElementById()`.



Задание

1. Разработайте форму заказа печатного продукта.

Заголовок – название печатного центра.

Предусмотрите поля – Фамилия, имя, адрес, телефон (валидация), желаемая дата выполнения заказа, категория заказа (выбор: визитка, бланк, флаер, ...), кнопки СБРОСИТЬ, ОТПРАВИТЬ,

При нажатии на кнопку СБРОСИТЬ поля очищаются. Кнопкой ОТПРАВИТЬ введенные данные записываются в текстовый файл. Добавьте дату формирования заявки. При тестировании предусмотрите, чтобы каждая новая отправка формировала новый текстовый файл с другим именем.

2. Модифицируйте бланк заказа, добавив изображения продукции, а под изображением поле **checkbox** и количество заказа. Рядом – стоимость заказа (разработайте функцию вычисления заказа):

```
<html>
  <head>
    <script>
      function doit() {
        var greeting="Мне нравится "
        alert(greeting +
document.aform.color.value // значение из поля
NAME=color
        + " " + document.aform.geometr.value)
// значение из поля NAME=geometr
        var prich="Потому, что "
        alert(prich +
document.aform.prichina.value)
        alert("Количество букв в названии вашей
фигуры"
        + document.aform.geometr.value.length)
      }
    </script>
  </head>
  <body>
    <center>
      <form NAME="aform">
        Ваша любимая геометрическая фигура:
```

```
<INPUT TYPE="text" NAME="geometr"><p>
Ваш любимый цвет:
<INPUT TYPE="text" NAME="color"><p>
Почему вам нравится эта фигура:
<INPUT TYPE="text" NAME="prichina"><p>
<INPUT TYPE="button" VALUE="Отвечить"
onClick="doit()" >
</form>
</center>
</body>
</html>
```



Контрольные вопросы

1. Как создать форму HTML?
2. Чем поле типа `checkbox` отличается от поля типа `radio`?
3. Как создать выпадающий список с множественным выбором?
4. За что отвечает атрибут **required**?
5. Как получить доступ к элементу из формы?

Лабораторная работа № 12

РАБОТА С БИБЛИОТЕКОЙ JQUERY

Цель работы: получить базовые знания по использованию библиотеки jQuery, освоить приемы работы с ней.

Время выполнения: 4 ч.



Теоретические сведения

jQuery – библиотека, которая значительно упрощает и ускоряет написание JavaScript кода.

Девиз jQuery «write less, do more» (пиши меньше, делай больше) отражает ее главное предназначение.

jQuery позволяет создавать анимацию, обработчики событий, значительно облегчает выбор элементов в DOM и создание AJAX-запросов.

Данная библиотека работает со всеми браузерами.

Пример использования jQuery:

```
$(document).ready(function() { // это основа
  $(".:button").click(function() {
    $(".:button").hide(); //скрыть кнопку
    $("#wrap1").addClass("add");
    $("#wrap1").animate({height:280},2000);
    $("#wrap1").animate({width:400},2000);
    $("#wrap1").animate({padding:20},2000,function() {
      $("#text1").hide(2000,function() {
        $("#text2").show(2000;});
      });
    });
  });
});
```

Добавление jQuery на страницы

Для того чтобы начать использовать jQuery, необходимо:

1) скачать ее с официального сайта: выберите подходящую версию, нажмите на кнопку «Download (jQuery)»;

2) добавить ее на страницу в секцию head:

```
<script type="text/javascript"
src="путь к скачанному файлу/jquery.js">
</script>
```

Для тех, кто по каким-либо причинам не может (или не хочет) скачивать jQuery, предусмотрен альтернативный способ удаленного использования библиотеки, предоставленный компанией Google.

Для того чтобы использовать jQuery удаленно, в секцию head вставить следующий код:

```
<script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.5/jquery.min.js">
</script>
```

Команды jQuery

Код jQuery, как и код JavaScript, состоит из последовательно идущих команд. Команды являются основной структурной единицей jQuery.

Стандартный синтаксис jQuery команд:

```
$(селектор.метод());
```

1. Знак \$ сообщает, что символы, идущие после него, являются jQuery кодом.

2. Селектор позволяет выбрать элемент на странице.

3. Метод задает действие, которое необходимо совершить над выбранным элементом.

Методы в jQuery разделяются на следующие группы:

- манипулирования DOM;
- оформления элементов;
- создания AJAX запросов;
- создания эффектов;
- привязки обработчиков событий.

Символ \$

jQuery можно комбинировать с обычным JavaScript. Если строка начинается с \$ – это jQuery, если \$ в начале строки отсутствует – это строка JavaScript кода.

Пример:

```
function start() {
//Скроем абзац с id="hid" с помощью jQuery
    $("#p#hid").hide();
//Изменим содержимое абзаца с id="change"
    document.getElementById("change").innerHTML="Данное
содержимое было изменено с помощью JavaScript.";
}
```

Некоторые JavaScript библиотеки тоже используют \$ для обозначения своего кода. Для того чтобы избежать конфликта имен между ними, в jQuery предусмотрена команда **noConflict()**. С помощью команды **noConflict()** можно заменить знак \$ на любой другой.

Синтаксис:

```
var symbol=jQuery.noConflict();
```

Пример:

```
//Изменим стандартный символ $ на nsign
var nsign=jQuery.noConflict();
//Изменим цвет абзаца используя nsign вместо $
nsign(document).ready(function(){
    nsign("p").css("color","green");
});
```

Селекторы jQuery

С помощью селекторов можно выбирать элементы на странице для применения к ним определенных действий (таблица).

Примеры использования селекторов

Пример	Результат
<code>\$("#p")</code>	Будут выбраны все элементы p, которые находятся на странице
<code>\$(".par")</code>	Будут выбраны все элементы на странице с class="par"
<code>\$("#par")</code>	Будет выбран первый элемент на странице с id="par"
<code>\$(this)</code>	Позволяет выбрать текущий HTML элемент
<code>\$("#p.par")</code>	Будут выбраны все элементы p на странице с class="par"
<code>\$("#p#par")</code>	Будут выбраны все элементы p на странице с id="par"
<code>\$(".par, .header, #heat")</code>	Будут выбраны все элементы на странице со значениями атрибутов class="par", class="header" и id='heat'
<code>\$("#[src]")</code>	Будут выбраны все элементы на странице, имеющие атрибут src
<code>\$("#[src='значение']")</code>	Будут выбраны все элементы со значениями атрибута src="значение"
<code>\$("#[src\$='.gif']")</code>	Будут выбраны все элементы со значениями атрибута src, заканчивающимися на .gif

Пример	Результат
<code>\$("#div#wrap .par1")</code>	Будут выбраны все элементы с <code>class=par1</code> , которые находятся внутри элементов <code>div</code> с <code>id=wrap</code>
<code>\$(" :input ")</code>	Будут выбраны все <code>input</code> элементы на странице
<code>\$(" :тип ")</code>	Будут выбраны все <code>input</code> элементы с <code><input type='тип' /></code> . Например: <code>button</code> выберет все <code><input type='button' /></code>

Пример:

```
$(document).ready(function() {
//Установим размер шрифта всех абзацев равным 20 пикселям

$("p").css("fontSize", "20px");
//Изменим на зеленый цвет шрифта элемента с id=el2

$("#el2").css("color", "green");
//Изменим на красный цвет шрифта элемента с class=el3

$(".el3").css("color", "red");
//Сделаем жирным шрифт элементов с id=el2 и class=el3

$("#el2, .el3").css("fontWeight", "bold");
//Изменим на синий цвет текста кнопки

$(" :input ").css("color", "blue");
//Установим размер шрифта всех элементов, имеющих атрибут href равным
20 пикселям

$(" [href] ").css("fontSize", "20px");
//Изменим на зеленый цвет ссылки на www.wisdomweb.ru

$(" [href='http://www.wisdomweb.ru/'] ").css("color", "green");
});
```

**Задание**

1. Спроектируйте страницу (можно воспользоваться готовыми файлами из прошлых лабораторных работ), запросите цвет абзаца и по нажатию на кнопку ПЕРЕКРАСИТЬ (вызов функции JQ), перекрасьте в требуемый цвет.

2. На странице два абзаца текста. В тексте есть слово КОТ. По нажатию на слово КОТ слово исчезает и появляется картинка котика на месте слова. При нажатии на картинку она исчезает и появляется слово КОТ. Используем `span`, `show`, `hide`.



Контрольные вопросы

1. Для чего предназначена библиотека jQuery?
2. Как добавить библиотеку jQuery на страницы?
3. На какие группы подразделяются методы в jQuery?
4. Как выбрать все элементы на странице с `class="blu"`?
5. Каким образом выбрать все `input` элементы на странице?
6. Как избежать конфликта имен между JavaScript и jQuery?

Лабораторная работа № 13

СИСТЕМЫ УПРАВЛЕНИЯ КОНТЕНТОМ

Цель работы: освоить приемы работы с онлайн-построителем сайтов.

Время выполнения: 4 ч.



Теоретические сведения

Системы управления контентом – программное обеспечение, позволяющее создавать и публиковать интернет-ресурсы без глубокого знания HTML, CCS, JS.

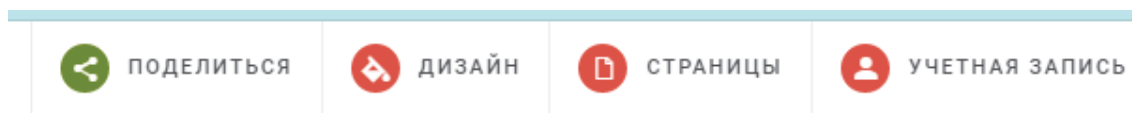
! **Внимание! НЕ используем коробочный вариант, работаем в режиме онлайн, т. е. нужен компьютер с интернетом.**



Порядок выполнения работы

1. Зайдите на <https://ru.simplesite.com/>
2. Нажмите вкладку НАЧАТЬ.
3. Задайте имя пользователя (это будущее имя вашего проекта и логин), придумайте и введите пароль.

Вкладки (рисунок).



Вкладки SimpleSite

Поделиться – предназначены для того, чтобы отправить сообщение по e-mail о вашем проекте.

Дизайн – позволяет выбрать или поменять дизайн.

Страницы – добавляем страницы в проект.

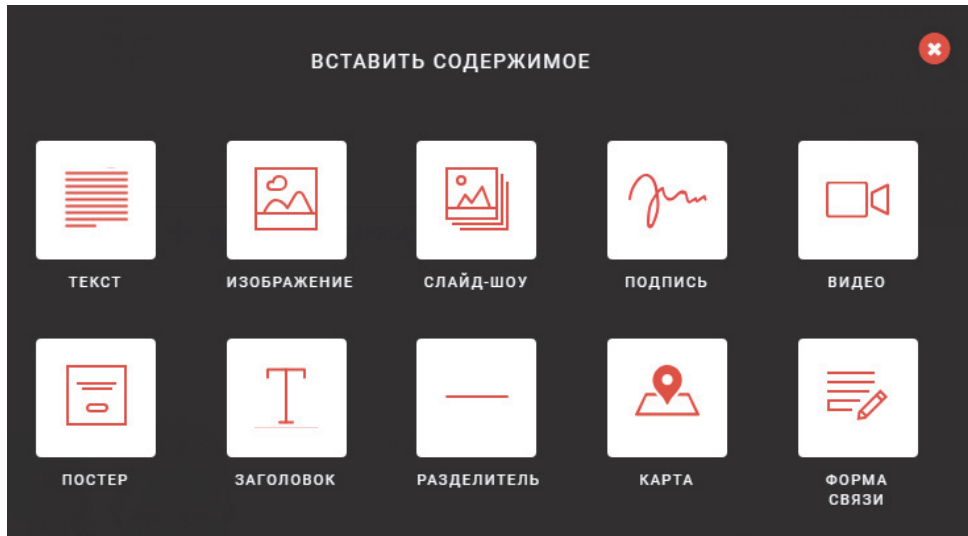
Учетная запись – управление вашей учетной записью.



Задание (по варианту)

Общие указания:

Создайте web-проект не менее 4 страниц, на которых должны быть разнообразные элементы.



Порядок выполнения работы

Выберите вариант, подберите материал, сохраните в отдельную папку, обработайте, соберите в сайт. Ссылку на собранный проект разошлите не менее чем четырем одноклассникам, получите отзывы по электронной почте, вставьте эти отзывы на отдельную страницу проекта ОТЗЫВЫ и пришлите ссылку на проект на почту для оценивания potapenko@belstu.by.

! Можно не придерживаться номера по списку, а выбрать тему по желанию. Однако в одной подгруппе темы не должны повторяться!!!

Варианты выбора:

- Блог писателя Н.
- Портфолио издателя детской литературы
- Сайт типографии
- Сайт магазина букиниста
- Сайт мастера кукол тильда
- Сайт филателиста
- Сайт магазина компьютерных игр

- Блог политика
- Сайт кинотеатра
- Сайт библиотеки
- Сайт нумизмата
- Сайт любителей бирдекали (подставки под пивные кружки)
- Сайт любителей кокофрамофилии (выпуски газет)
- Сайт любителей кошек (фелинофелия)
- Сайт, посвященный образу хитрого лиса (ренардофилия)
- Свободная тема

Ссылки

Виды коллекций:

<http://forumuuu.com/showthread.php?t=22125>

<https://ru.simplesite.com>



Не забываем указать ссылки, откуда брали материал для проекта!



Контрольные вопросы

1. Что такое системы управления контентом и для чего они предназначены?
2. Перечислите известные вам системы управления контентом.
3. Какие возможности предоставляет пользователю онлайн-построитель сайтов SimpleSite?

Лабораторная работа № 14

БИБЛИОТЕКА WOW.JS, ANIMATE.JS, TEXTILATE.JS

Цель работы: научиться применять анимационные эффекты при оформлении web-страниц, текста, блоков текста, скроллинге.

Время выполнения: 4 ч.



Теоретические сведения

WOW.js – это библиотека, которая позволяет включать анимацию на определенном этапе прокрутки страницы. Компактная и полностью самостоятельная, нет необходимости подключать jQuery или другие библиотеки.

Animate.CSS – скрипт, который непосредственно отвечает за саму анимацию. По сути wow.js берет анимации именно из этого скрипта. Всего видов анимаций порядка 70.

Подключение

```
<link rel="stylesheet" href="css/animate.css">
<script src="js/wow.min.js"></script>
```

Активация

```
<script>
    new WOW().init();
</script>
```

Требуется добавить ссылку на файл и инициализировать JS класс. Для активации блока при помощи библиотеки нужно прописать ему css класс wow:

```
<div class="wow bounceInUp">
    Content to Reveal Here
</div>
```

Библиотека подхватывает определенные атрибуты, позволяющие настроить действия:

data-wow-duration – изменить длительность анимации по времени.

data-wow-delay – изменить задержку при активации анимации.

data-wow-offset – дистанция от линии расположения блока до нижней видимой линии браузера, при которой запустится анимация.

data-wow-iteration – количество повторов анимации.

При инициализации можно использовать свои настройки библиотеки (все значения внизу – это значения по умолчанию).

```
wow = new WOW({
  boxClass: 'wow',           // название CSS класса для активации
  animateClass: 'animated', // название CSS класса анимации
  offset: 0,                 // отступ как в атрибуте data wow-offset, но для всех
                             // элементов сразу
  mobile: true,             // включение-выключение библиотеки на мобильных
                             // устройствах
  live: true                 // проверка новых элементов на странице, добавляемых
                             // динамически
})
wow.init();
```



Пример 1. Заголовок и абзац

При загрузке страницы заголовок (h1) появляется слева, а текст абзаца (div) справа.

```
<h1 class="wow animate slideInLeft">Первый заголовок</h1>
<div class="wow animate slideInRight"> Lorem ipsum dolor
sit amet...</div>
```



Пример 2. Скроллинг

```
var wow = new WOW({
  boxClass: 'wow', // класс, скрывающий элемент до момента
                  // отображения на экране (по умолчанию, wow)
  animateClass: 'animated', // класс для анимации элемента
                  // (по умолчанию, animated)
```

```

    offset: 0,      // расстояние в пикселях от нижнего края
браузера до верхней границы элемента, необходимое для начала
анимации (по умолчанию, 0)
    mobile: true,  // включение/отключение WOW.js на мобиль-
ных устройствах (по умолчанию, включено)
    live: true,    // поддержка асинхронно загруженных эле-
ментов (по умолчанию, включена)
    callback: function(box) {
        // функция срабатывает каждый раз при старте анимации
        // аргумент box – элемент, для которого была запущена
анимация
    },
    scrollContainer: null // селектор прокручиваемого кон-
тейнера (опционально, по умолчанию, window)
}
);
wow.init();

```



Пример 3. Анимация текста

Textillate.js – это jQuery-плагин, который объединяет возможности *animate.css* и *lettering.js* для легкого применения множества вариантов CSS3-анимации для любого текста.

Пример использования

Начнем с простой разметки:

```
<h1 class="tlt"> Мой текст</h1>
```

Для вызова плагина для этого текста JavaScript-код должен выглядеть следующим образом:

```
$(function () {
    $('.tlt').textillate();
})
```

В данном случае будет использоваться анимация с параметрами по умолчанию. Для изменения настроек по умолчанию можно использовать HTML data API

```
<h1 class="tlt" data-in-effect="rollIn">Заголовок</h1>
```

или передать в параметрах при инициализации

```
$('.tlt').textillate({ in: { effect: 'rollIn' } });
```

Можно вызвать `textillate.js` для анимации списка текстов при помощи следующей разметки:

```
<h1 class="tlt">
<ul class="texts">
  <li data-out-effect="fadeOut" data-out-shuffle="true">Пункт меню</li>
  <li data-in-effect="fadeIn">Второй пункт меню</li>
</ul>
</h1>
$('.tlt').textillate();
```

Обратите внимание, что вы можете контролировать параметры анимации для каждого пункта (``) при помощи **data api**.

Для использования `textillate.js`:

1. Определяем класс с именем `class="tlt"`.
2. Используем с заданием эффекта:

```
<h1 class="tlt" data-in-effect="bounce">Lorem ipsum...</h1>
и $('.tlt').textillate();
или следующим образом:
$('.tlt').textillate({ in: { effect: 'rollIn' } });
```

Чтобы начать использование `textillate.js`, необходимо подключить следующие библиотеки:

jQuery, `lettering.js`, `animate.css`

Примеры использования эффектов WOW и `textillate.js` приведены в папке WOW.



Задание

1. Скопируйте пример, объясните, как работает.

```
<div class="row">
  <div class="col-lg-4 col-md-6 wow animate
bounceInUp" data-wow-delay="200">
    <h1>Заголовок №1</h1>
    <div>Lorem ipsum dol...</div>
  </div>
```

```
<div class="col-lg-4 col-md-6 wow animate
bounceInUp" data-wow-delay="400">
  <h1>Заголовок №2</h1>
  <div>Lorem ipsum dolor sit amet, ct...</div>
</div>
<div class="col-lg-4 col-md-6 wow animate
bounceInUp" data-wow-delay="600">
  <h1>Заголовок №3</h1>
  <div> Lorem ipsum dolor sit amet, t...</div>
</div>
</div>
```

Атрибут **data-wow-delay**, заданный у каждой колонки, позволяет активировать блоки по очереди.

2. Изучите и проверьте работу **WOW** и **textillate.js**. Для этого создайте 2 страницы с текстом (не менее 4 абзацев). Примените несколько эффектов.

3. Переделайте 2 страницы следующим образом:

а) первая страница содержит две колонки текста (не табличная верстка), для заголовков и текста использовать эффекты из библиотеки **textillate.js**;

б) вторая страница содержит три блока текста, заголовок, изображение, при загрузке страницы применить эффекты для заголовков, текста или блоков текста из библиотеки **WOW**.



Контрольные вопросы

1. Что такое **WOW.js**? Как активизировать ее на web-странице?
2. Что собой представляет **Textillate.js**, для чего предназначен и как использовать?
3. Дайте определение **Animate.CSS**.

Лабораторная работа № 15

СЛАЙДЕРЫ НА WEB-СТРАНИЦЕ

Цель работы: научиться по готовым шаблонам конструировать слайдер по своей теме.

Время выполнения: 4 ч.



Теоретические сведения

Слайдер – элемент web-дизайна, состоящий из блока определенной ширины, содержащего изображения, которые в автоматическом или ручном режиме сменяют друг друга. Слайдеры могут размещаться на главной странице, на других тематических страницах, в футере. Функция слайдеров заключается в добавлении изображений в текст и более выгодный показ данных сайта. Главное преимущество слайдера – показ информации визуалью. Как известно, интерес к движению заложен еще в подсознании, поэтому слайдеры и стали обязательной частью любого дизайна.

Рассмотрим подробнее элементы слайдера: средства навигации, экран, маркеры с информацией о количестве слайдов. Смена слайдов происходит в трех режимах – автоматический, ручной и смешанный. В автоматическом режиме слайд за слайдом меняется через 1–3 с. При ручном режиме частота смены слайдов зависит от желания и действий пользователя. В смешанном типе изображение автоматически меняется через 5–25 с, поэтому при желании можно поменять картинку одним нажатием.

Количество изображений-слайдов для слайдера может быть любым. Так, стандартный набор состоит из 3–5 изображений. Это может быть либо обычное изображение, либо дополняться текстом с информацией, таблицами и ссылками. В слайдер иногда вставляют анимацию или небольшое видео. *Маркеры* – активные элементы (стрелки, кружочки), предназначенные для ручного управления или показа (кружочки) количества слайдов в слайдере.

В большинстве случаев слайдеры работают по аналогичным алгоритмам.

В функциях, обеспечивающих работу слайдера, указана скорость, с которой перелистываются слайды, устанавливаются условия показа

слайдов (предыдущий / последующий или по номеру), задается внешнее оформление слайдов и дополнительные эффекты перелистывания. Вывод слайдов в требуемом месте web-страницы прописан в HTML-коде, за визуальное оформление отвечает CSS, а за работу Javascript – сценарий, например функции библиотеки jQuery.



Задание

Для создания собственных слайдов воспользуйтесь заготовками слайдеров. В начале необходимо подготовить изображения на любую тему (4–5 изображений). Обратите внимание на их формат и размеры. Они должны быть идентичны. Далее разберите готовый пример по ссылке, указанной ниже. Разработайте web-страницу – заголовок, слайдеры, сделанные по пунктам 1–4, футер – ваша фамилия и номер группы. Остальные элементы декора на ваше усмотрение.

1. <https://codepen.io/lucasmotta/full/ujKbF> – адаптивная бесконечная карусель. Вставить свои изображения.

2. <https://amazingcarousel.com/examples/javascript-carousel-with-images-and-videos-id8/> – карусель JavaScript с изображениями и видео. Видео взять любое из свободных источников.

3. <https://amazingcarousel.com/examples/jquery-testimonial-slider-id14/> – карусель с несколькими элементами управления.

4. Слайдер с эффектами WOW <https://wowslider.com/jquery-slider-carousel-mac-stack-demo.html>.

5*. Круговой слайдер <https://baijs.com/tinycircleslider/> (это бонус, работа по желанию).



Контрольные вопросы

1. Для чего применяются слайдеры на web-странице?
2. С помощью каких средств можно создать слайдер?
3. Какие основные элементы слайдера?
4. Можно ли создать слайдер на «чистом» CSS?
5. Какой слайдер будет загружаться быстрее – разработанный только с применением CSS и на JS?

Лабораторная работа № 16

СОЗДАНИЕ СПОЙЛЕРОВ

Цель работы: получить навыки создания и оформления web-страниц с использованием спойлеров.

Время выполнения: 2 ч.



Теоретические сведения

Спойлер – визуальная линейка, которая организует большой (или не очень) текст / изображения / видео / анимацию в раскрывающиеся блоки. Для электронных изданий – это очень интересный и полезный инструмент. Можно провести аналог с содержанием, когда по гиперссылке открывается фрагмент текста или осуществляется переход на следующую страницу. Здесь видны закладки-блоки. При нажатии ниже открывается блок информации. При повторном нажатии скрывается. Примеры спойлеров можно посмотреть по ссылке: <http://shpargalkablog.ru/2013/04/details-html.html>.



Задание

Используя материал лабораторной работы № 7, оформите содержимое в виде спойлеров. Используйте два вида спойлеров на ваше усмотрение. Используйте готовые решения <http://shpargalkablog.ru/2013/04/details-html.html>, адаптировав их под свой контент.



Контрольные вопросы

1. Что такое спойлер? Дайте другие названия этого термина.
2. Какие технологии применяются для разработки спойлера?
3. Всегда ли оправдано использование спойлера?
4. В каких случаях применяется этот инструмент?

Лабораторная работа № 17

СОЗДАНИЕ ВИДЕО, АНИМАЦИИ. ВИДЕОМОНТАЖ

Цель работы: получить навыки создания и оформления коротких видеороликов, баннеров, анимационных плакатов.

Время выполнения: 6 ч.

Указания к выполнению работы

1. Можно скачать <https://www.adobe.com/ru/products/premiere/free-trial-download.html> Adobe Premier и установить у себя на компьютере, но *версия действует бесплатно 7 дней*.

2. Можно воспользоваться онлайн сервисами:

a) <https://www.kizoa.ru/Movie-Maker>

b) Camtasia Studio <http://relizua.com/besplatnaya-multimedia/436-skachat-camtasia-studio-na-russkom.html>

c) Видеоредактор <https://videoredaktor.ru/>

d) Movavi Video Editor 14 – многофункциональный видеоредактор

e) Видеоконструктор <https://supa.ru/>



Задание

1. Разработайте короткий видеоролик – тема может быть любая, должны присутствовать наложение текста, эффекты, переходы, озвучка или фоновая музыка. Продолжительность – 1–2 мин.

2. Разработайте анимационный баннер (плакат) – реклама продукта, услуги.

3. Разместите на web-странице. Оформление страницы – заголовок, фон страницы, предусмотрите запуск ролика не в автоматическом режиме. Добавьте кнопку «аудио» для запуска музыки.

4. На странице отобразите сценарии ролика и баннера и короткий отчет – этапы и технологии создания вашего проекта.

Общие указания

1. Разработайте (придумать сценарий).

a) название, цель – что мы хотим показать;

б) ключевые кадры – начало – отображается вид1, затем вид2, ..., окончание – видX;

в) переход между кадрами – текст – , смена кадров – эффект ...;

г) заикливание – ролик прокручивается один раз, бесконечно, несколько раз.

2. Реализация – с помощью чего вы монтировали ролик, как использовали монтажную линейку, какие эффекты применили, какой текст и как вы его накладывали.

3. Формат, в котором сохранили ролик.

4. Страница с внедренным роликом.



Контрольные вопросы

1. Какие видеоредакторы вы знаете? Их достоинства и недостатки.

2. Перечислите виды баннеров и их назначение.

Курсовая работа

РЕКОМЕНДАЦИИ ПО ОРГАНИЗАЦИИ И ВЫПОЛНЕНИЮ РАБОТ

Курсовая работа – это вид самостоятельной учебной работы и контроля качества обучения студента на определенных этапах выполнения учебного плана по специальности. Выполнение курсовых работ предусматривается типовыми учебными планами в целях закрепления, углубления и обобщения знаний, полученных студентом в процессе обучения, развития способности самостоятельного и творческого мышления. Курсовая работа содержит обоснованное решение практической задачи (электронное мультимедийное издание), вытекающее из анализа выбранного объекта и ситуации.

В процессе подготовки курсовой работы студенту следует:

- определить актуальность темы курсовой работы, степень ее разработки в интернете, сформулировать цели и предполагаемую целевую аудиторию;

- изучить аналоги электронных изданий, опубликованных в интернете или доступных в ЭБС, провести сравнительный анализ, выделив положительные и отрицательные черты;

- разработать логотип, слоган (при необходимости);

- обосновать типографику и цветовое решение;

- спроектировать прототип и разработать дизайн-макет;

- определить уровень мультимедиа и ее формы на web-проекте;

- описать инструменты для верстки web-проекта и используемые технологии;

- сверстанный web-проект проверить на валидаторе;

- сформулировать выводы.

Все курсовые работы выполняются в сроки, предусмотренные соответствующим учебным планом под руководством преподавателя университета.

Курсовая работа является формой текущей аттестации студентов по дисциплине учебного плана специальности. Дифференцированный зачет (отметка по 10 балльной системе) по курсовой работе учитывается при определении общего числа экзаменов и зачетов на соответствующем этапе обучения. Темы курсовых работ вносятся в зачетные книжки студентов.

Особенностью курсовых работ является следующее:

1. *Выбор темы курсовой работы.* Списки тем курсовых работ формируются и утверждаются на кафедре «Информатика и web-дизайн». Студент может согласовать с руководителем и другую тему курсовой работы, обосновав свой выбор.

2. *Студент обязан в процессе написания курсовой работы:*

- консультироваться с руководителем по срокам выполнения работы, плану, структуре, выбранным технологиям;
- осуществить необходимые работы по проектированию и верстке web-проекта в соответствии с методикой подготовки курсовых работ;
- представить в установленные сроки курсовую работу в печатном и электронном варианте;
- подготовить доклад по результатам курсовой работы.

3. *Руководитель обязан:*

- совместно со студентом составить план работы;
- объяснить студенту методику подготовки курсовой работы;
- консультировать студента по выбору научной и технической литературы, использованию технологий проектирования и верстки web-проекта;
- периодически проверять процесс выполнения курсовой работы, при необходимости – корректировать направление работ.

Цель курсовой работы – применение знаний в практической деятельности по решению задач предметной области, приобретение и закрепление умений, полученных на лабораторных занятиях, и формирование навыков по решению типовых задач.

На выполнение курсовой работы согласно учебному плану студенту отводится 40 ч.

В состав курсовой работы входят:

- пояснительная записка;
- электронное приложение (мультимедийное электронное издание или web-сайт).

Задачами курсовой работы являются закрепление основ и углубление знаний принципов и приемов проектирования и создания мультимедийного web-проекта. В ходе выполнения курсовой работы у студента должно формироваться представление об этапах проектирования и создания мультимедийных электронных изданий.

Организация выполнения курсовой работы

Продолжительность выполнения курсовой работы – 17 недель. Каждый студент выполняет индивидуальное задание. Список типовых заданий на курсовую работу приведен в прил. В. Каждый студент обязан

посетить не менее 5 консультаций по вопросам выполнения курсовой работы, предъявляя на предварительный просмотр преподавателю отдельные результаты web-проекта и пояснительной записки. По результатам курсовой работы студентом в соответствии с требованиями оформляется пояснительная записка и сдается преподавателю на проверку. Если все недочеты, выявленные при консультациях, устранены, студент защищает курсовую работу, по результатам защиты выставляется итоговая оценка. Если пояснительная записка не удовлетворяет поставленным требованиям, то она возвращается студенту на доработку. При выполнении курсовой работы студент обязан посетить консультации и зачет у руководителя, как минимум, следующие этапы работы:

- 1) обоснование темы, выбора платформы и инструментария для создания сайта;
- 2) разработка web-дизайна сайта;
- 3) описание программной реализации web-сайта;
- 4) тестирование сайта;
- 5) выводы по полученным результатам.

Каждый этап может представляться на проверку преподавателю по частям во время консультаций. Консультации проводятся 1 раз в 2 недели.

Защита курсовой работы

Защита курсовой работы является особой формой проверки глубины осмысления изученного материала, эффективным способом выработки и демонстрации навыков научных выступлений, умения кратко, аргументировано и четко излагать представленный в курсовой работе материал. Защита курсовой работы принимается комиссией в составе не менее двух человек. В состав комиссии входит непосредственный руководитель курсовой работы. Состав комиссии, порядок и формы ее работы определяются распоряжением заведующего кафедрой и доводятся до сведения студентов, руководителей курсовых работ и членов комиссии.

Формой аттестации по курсовым работам является дифференцированный зачет и оценивается по десятибалльной шкале оценки знаний. Отметка за курсовую работу выставляется по результатам защиты в ведомость и зачетную книжку студента (неудовлетворительная оценка – только в ведомость) и заверяется подписями членов комиссии. После защиты, не позднее 10 дней после окончания экзаменационной сессии, ответственный сотрудник кафедры готовит опись курсовых работ, которые хранятся на кафедре на протяжении года после окончания срока обучения. По истечении срока хранения составляется акт о выделении к уничтожению документов, не подлежащих хранению.

Содержание и структура курсовой работы

Общий объем работы – минимум 25 страниц печатного текста формата А4 (без приложений).

Структурными элементами курсовой работы являются:

- титульный лист (прил. В);
- задание на курсовую работу (прил. Г) оформляется на бланке установленной формы и утверждается заведующим кафедрой до начала выполнения работ;

– реферат;

– содержание:

Введение

1. Аналитический обзор

1.1. Цели создания ЭИ

1.2. Обзор аналогов (не меньше 4 схожих изданий)

1.3. Технологии реализации web-проекта

2. Проектирование

2.1. Описание страниц проекта

2.2. Структурная схема проекта

3. Дизайн

3.1. Цветовая схема

3.2. Типографика

3.3. Логотип

3.4. Дизайн-макет

4. Разработка и верстка

4.1. Обработка контента

4.2. Видео и анимация

4.3. Верстка

4.4. Тестирование

Заключение

Список использованных источников

Приложения

Рекомендации по содержанию курсовой работы

Реферат должен быть выполнен в соответствии с ГОСТом и содержать:

- сведения об объеме курсовой работы;
- количество страниц пояснительной записки с указанием количества рисунков, таблиц, использованных источников и приложений;
- перечень ключевых слов;
- текст реферата.

В начале текста реферата указывается Курсовая работа: количество листов пояснительной записки, количество рисунков, приложений, источников.

Пример:

Курсовая работа: 23 с., 6 рис., 2 прил., 15 источников.

Перечень ключевых слов должен включать от 5 до 15 слов или словосочетаний из текста курсовой работы, которые в наибольшей мере последовательно характеризуют его содержание и обеспечивают возможность информационного поиска. Ключевые слова приводятся в именительном падеже, единственном числе, прописными буквами в строку через запятые, без переноса слов, записываются с начала строки без абзацного отступа. Точка в конце перечня не ставится.

Текст реферата должен отражать сокращенное изложение содержания курсовой работы с основными фактическими результатами.

Слово «Реферат» записывают в виде заголовка, размещенного по центру текста, с первой прописной буквы.

Объем текста реферата составляет 500–800 знаков (не более одной страницы).

Во *Введении* раскрывается актуальность выбранной темы и дается оценка современного состояния предметной области, с которой связано выполнение курсовой работы. Актуальность web-проекта может быть связана с недостаточным количеством web-проектов выбранной тематики, устаревшим (непривлекательным) дизайном, отсутствием современных элементов дизайна, привлекательностью для потенциальных посетителей сайта. Также приводится краткая аннотация глав пояснительной записки.

Раздел 1. «Аналитический обзор»

1.1. Цели создания ЭИ

1.2. Обзор аналогов (не меньше 4 схожих изданий)

1.3. Технологии реализации web-проекта

В пункте «*Цели создания ЭИ (web-проекта)*» формулируются цели создания проекта, перечисляются задачи, которые следует учитывать при создании web-проекта. Цель – это результат, который необходимо получить с помощью web-проекта за определенный период времени. Например:

- познакомить с творчеством автора X через восприятие мультимедиа контента;
- предоставить услуги онлайн;
- дать возможность изучить географию с использованием мультимедийного контента.

Четко поставленные цели позволят определить параметры структуры сайта, его внешний вид, схему и содержимое каждой страницы.

Пункт *«Обзор аналогов»*. В этом разделе представляется аналитический обзор изданий схожей тематики. Анализируется дизайн, используемые технологии, функционал. Описываются их сильные и слабые стороны. В курсовой работе рекомендуется исследование не менее четырех изданий. В этом разделе можно привести скриншоты главных страниц издания. Обязательно указать ссылки на источники получения информации (интернет-адрес, ссылка на ЭБС).

Пункт *«Технологии реализации web-проекта»*. Описываются и характеризуются технологии для обработки контента, мультимедиа-компонентов, сборки готового проекта в единое издание. Не более двух страниц.

Раздел 2. «Проектирование»

2.1. Описание страниц проекта

2.2. Структурная схема проекта

В подразделе *«Описание страниц проекта»* составляется описание для каждой страницы:

- функциональность;
- информация, которая будет на ней присутствовать;
- цели, которые она выполняет;
- точки входа на страницу и выхода из нее.

Дается описание элементов хедера и футера.

В подразделе *«Структурная схема проекта»* определяется модель организации контента или, другими словами, структура сайта. Существуют следующие основные виды моделей:

Одна страница. Подходит для сайтов с ограниченным содержимым, узкой целью (промо-сайты, лонгриды, персональные страницы).

Линейная структура. Имеет сквозную навигацию, количество страниц на сайте не более 10. Это может быть энциклопедия, портфолио типографии, обучающий ресурс (электронный учебник, энциклопедия, электронная книга), небольшой интернет-магазин с ограниченным видом товаров или услуг (см. рис. Г1), сайт типографии.

Строгая иерархия. Переход на следующую страницу можно сделать только со страницы-родителя (электронные книги, учебники с заданной последовательностью прохождения тем, презентации, портфолио (см. рис. Г2)).

Ромашка. Такая структура предполагает возврат на главную страницу с любой целевой страницы. Чаще всего встречается в web-приложениях, на образовательных сайтах (см. рис. Г3).

Многомерная иерархия. Одна из наиболее распространенных структур. Предусматривается наибольшее количество элементов навигации, каждая страница доступна отовсюду (см. рис. Г4).

В этом подразделе приводятся схемы логической взаимосвязи будущих страниц проекта. Рекомендуется для этого использовать специализированные программы для построения логических схем, такие как <https://www.xmind.net/>, <https://www.mindmeister.com/ru/>, <https://coggle.it/>, <https://www.mindjet.com/>, <https://itunes.apple.com>, <https://www.thebrain.com/>.

Некоторые из программ условно бесплатны, поэтому не забывайте сохранять копии созданных логических схем на компьютере.

Далее определяется навигация и используемые компоненты.

Навигация – это возможность перехода между страницами сайта. Навигационная система определяет успешность сайта, чем она понятнее, тем лучше для посетителей.

Критерии качественной навигации:

– простота – все элементы пользовательского интерфейса должны быть хорошо видимыми и понятными для пользователя. Считается, что правило «не больше трех кликов» должно обеспечивать путь к любой странице сайта;

– доступность – с любой страницы сайта должна быть понятная пользователю возможность перейти на необходимый раздел или вернуться на главную страницу;

Раздел 3. «Дизайн»

3.1. Цветовая схема

3.2. Типографика

3.3. Логотип

3.4. Дизайн-макет

Дизайн сайта – это совокупность графических элементов, шрифтов и цветов, реализованных на сайте. Основная задача дизайна сайта – объединение всех информационных блоков и формирование у посетителя приятного впечатления.

При разработке дизайн-макетов страниц электронного мультимедийного издания учитываем следующее:

Типовые страницы – это шаблоны или макеты, на базе которых создаются страницы сайта, обладающие схожими характеристиками, например:

– страница текста;

– страница товаров в каталоге – содержит сквозные элементы в виде шапки и подвала, а также дополнительные блоки: боковую колонку с

фильтрацией товаров по заданным параметрам, список товаров в конкретной категории и блок персональных предложений;

- страница развернутой темы – электронная книга, учебник, журнал;
- страница персоналий – содержит графический блок – фотография, текстовый блок – биография, научные интересы, адресный блок, блок обратной связи;

Уникальные страницы не имеют повторений. Это может быть главная страница, страница «Контакты», «Мы на карте», «Об авторе» и т. д.

Вспомогательные страницы. Это страницы, которые могут быть не обязательными для просмотра всеми, например:

- страница 404 (ошибки);
- страница результатов поиска;
- страницы входа и регистрации.

При разработке макетов страниц учитываем правила и законы композиции.

Композиция – это определенное сочетание элементов сайта и их гармоничное расположение. При составлении композиции, т. е. выделении текстовых и графических регионов на макете, помним о логических ударениях. **Логическое ударение** – психолого-аппаратные приемы, направленные на привлечение внимания пользователя к определенному объекту. Психологическое действие логических ударений связано с уменьшением времени зрительного поиска и фиксации оси зрения по центру главного объекта.

Учитывая специфику тем электронных мультимедийных изданий, логотипа в web-проекте может и не быть.

Логотип – оригинальное изображение, отражающее сущность компании, организации, товарного знака. Выделяют три группы изображений логотипа:

- текстовые;
- знаковые;
- комбинированные.

Цвета логотипа должны повторяться в цветовом оформлении сайта.

Типографика – графическое оформление текста средствами представления, набора и в соответствии с правилами конкретного языка. Качество представления текста – очень важный критерий успешности сайта.

Базовые правила типографики на web-сайте:

- минимальное количество шрифтов. Не рекомендуется использовать более 2–3, уделить внимание размерам. Оптимальный вариант гарнитур – один. В случае использования более одного шрифта необходимо проследить, чтобы они имели схожее начертание и совпадение по ширине;

– стандартные шрифты (Arial, Calibri, Trebuchet) отображаются во всех браузерах. Авторские шрифты требуют загрузки на сервер и дополнительных действий по их правильному отображению в браузерах;

– визуальный комфорт чтения заключается в длине строки, т. е. количества символов в строке. Оптимальным считается для десктопов – 60–75 символов в строке, для мобильных устройств – 30–40. Ширина текстового блока выставляется в единицах измерения em или пикселях;

– универсальная гарнитура для всех устройств. Текст должен одинаково хорошо читаться на любом устройстве;

– для основного текста не используются заглавные буквы;

– межстрочный интервал (интерлиньяж) должен быть на 30% больше высоты самих символов;

– соотношение контрастов согласно рекомендациям W3C должно быть следующим:

а) мелкий текст имеет соотношение контрастов между текстом и его фоном не менее 4,5 : 1,0;

б) крупный текст (шрифт 14 жирный / 18 нежирный и выше) имеет соотношение контрастов между текстом и его фоном не менее 3 : 1;

– избегать использования мигающего текста, так как он может вызывать негативное отношение у посетителей сайта.

В этом разделе приводится обоснование выбранных шрифтов и их связь с выбранным стилем web-дизайна в целом, даются наборы начертаний.

Цветовая схема. Цвета должны и могут устанавливать влияние на эмоции посетителей. Они являются сильным фактором воздействия на них. При выборе цветовой схемы руководствуемся рекомендациями по теории цвета.

Определяются цветовые схемы, дается их описание для шрифтов, фона, дополнительных элементов и акцентов.

Фон может быть:

– одноцветный;

– текстура;

– градиент.

Необходимо указать желаемые параметры.

Чтобы дизайн сайта смотрелся гармонично, не разваливаясь на отдельные составляющие, необходимо следовать базовым принципам дизайна:

– акцент – акцентирование внимания на ключевых элементах сайта. Определяются главные, ключевые элементы сайта и регионы для них;

– контраст – визуальная дифференциация нескольких элементов. Основные элементы на сайте четко выделяются с помощью изменения размера шрифта, добавления цветов, замены расположения элементов и т. д.;

– баланс – распределение визуальной нагрузки за счет правильно сбалансированного распределения пустого пространства, графики, текста, цвета шрифтов и фона;

– точность – размеры идентичных блоков должны совпадать пиксель в пиксель, все элементы обязаны как можно ближе подгоняться под границы страницы и максимально эффективно задействовать свободное пространство;

– стиль – четкая цветовая палитра, отличительные линии, очертания и сбалансированный набор шрифтов;

– удобство восприятия – посетитель, придя на сайт, должен понимать, где он и как ему двигаться дальше (продуманная навигация).

Указать тип элемента и способ его начертания, например:

Основной текст – Times New Roman, 14 пт, цвет – черный (#000000).

В разделе 2.3. «Дизайн-макет» представляются дизайн-макеты web-проекта, объединившие прототипы, навигацию, логотип, типографику и цветовое решение web-страниц согласно определенному стилю web-дизайна и законам композиции.

Дается описание ключевых моментов для последующей web-верстки.

Web-дизайн не должен мешать восприятию и читабельности текста и должен быть разработан таким образом, чтобы, несмотря на все графические и мультимедийные элементы, не увеличивать скорость загрузки страниц сайта. Примеры дизайнов-макетов приведены в прил. Б.

Раздел 4. «Разработка и верстка»

4.1. Обработка контента

4.2. Видео и анимация

4.3. Верстка

4.4. Тестирование

В подразделе «Обработка контента» приводится краткое описание подготовки текстовой и графической информации. Указываются размеры шрифтов для основного и вспомогательного текстов, эффекты для текста. Для изображений приводится описание технологий подготовки графического контента для публикации. Указываются программные средства, используемые при подготовке, форматы и размеры файлов.

В подразделе «*Видео и анимация*» приводятся сценарии для видео, анимации, скриншоты подготовки видео, наложения эффектов. Для анимации – фрагменты программного кода с описанием и комментариями. Мультимедиа – контент или содержание, которое одновременно передается в разных формах: звук, анимированная компьютерная графика, видеоряд. Например, в одном объекте-контейнере может содержаться текстовая, аудиальная, графическая и видеоинформация, а также, возможно, способ интерактивного взаимодействия с ней.

Анимация на web-сайте может быть представлена в различных вариациях и технологиях.

Для анимации элементов используем:

- созданные в сторонних редакторах gif-файлы;
- Java-апплеты;
- трансформации на CSS;
- анимации на CSS;
- JavaScript и JQuery;
- тег canvas;
- тег SVG;
- flex-блоки на CSS.

В подразделе «*Верстка*» указываются используемые технологии верстки (блочная, семантическая) и библиотеки. Дается пояснение, какие особенности библиотек применяются для создания сайта.

JavaScript библиотеки (*JQuery, WOW, Animate, Textillate* и др.) ускоряют процесс разработки web-приложений, обеспечивают простоту создания сложных элементов интерфейса, предоставляют средства для анимации элементов и многое др.

В этом разделе можно привести фрагменты кода с комментариями программной реализации ключевых элементов, например реализация галереи, параллакс-эффекта и пр. Фрагменты могут быть представлены как в тексте пояснительной записки, так и вынесены в приложения. Листинги сопровождаются комментариями.

Различают:

- заголовочный комментарий – содержит название проекта, авторство, используемые технологии, основную идею, дату последнего изменения;
- процедурные комментарии – поясняют основные содержательные блоки.

Подраздел «*Тестирование*». В ходе тестирования проверяются навигация, интерфейс, текстовое и графическое наполнение и web-дизайн, все те компоненты, от которых зависит эффективность работы ресурса.

Тестирование помогает проверить, все ли этапы были качественно реализованы. В случае выявления слабых мест, неудобных логических связей, программных сбоев проводится корректировка, меняется текст, web-дизайн и т. д. Успешное тестирование – необходимый показатель, без которого невозможно оценить, насколько грамотно было выполнено создание сайта.

Валидация сайтов – это проверка специальными программами на соответствие HTML-кода страниц сайта и CSS-кода современным стандартам. Валидация позволяет исправить своевременно ошибки кода, уплотнить код, сделать сайт понятным для большинства браузеров. Рекомендуется использовать совмещенный валидатор <https://validator.w3.org>. Он позволяет проверить как опубликованный сайт, так и локальный файл.

В пояснительную записку вносится описание процесса валидации и их результатов.

В *заключении* формулируются краткие выводы по проделанной работе, дается оценка степени выполнения поставленных задач, полученных проектных разработок.

Список использованных источников содержит библиографический список сведений об источниках, использованных при выполнении курсовой работы. В случае интернет-источника указываем дату и режим доступа. Для курсовой работы рекомендуется использовать не менее 15 источников. Ссылки на ресурсы Википедии не использовать.

В *приложения* рекомендуется включать материалы, связанные с выполнением курсовой работы (дизайн-макеты, таблицы, графики, рисунки), если они затрудняют восприятие материала, а также листинг кода разрабатываемого web-сайта. Серьезными недостатками работы признается неполное соответствие материалов работы избранной теме, отсутствие последовательности изложения материала, нечеткие формулировки.

Требования к оформлению пояснительной записки

Пояснительная записка выполняется в формате Microsoft Word печатным текстом формата А4, печать односторонняя согласно стандарту ГОСТ 7.32. Поля: правое – 3,0, левое – 1,5, верх – низ – 2,0, ориентация – книжная, абзацный отступ – 1,25 см, междустрочный интервал – 1,0; шрифт Times New Roman, 14 кегль; стиль «Обычный», выравнивание текста по ширине. Выделения в тексте – курсивом. Заголовки – стиль «Строгий».

Основной текст набирается по правилам набора без ошибок. Все предлоги должны быть соединены неразрывным пробелом с относящимся к ним текстом. В качестве тире используется специальный символ

«короткое тире». В тексте применяется автоматический перенос слов. Для вставки ссылки на источник литературы используются квадратные скобки из латинского алфавита, например ссылка на первый источник согласно списку использованных источников [1]. Нумерация в списке использованных источников сквозная. Первому источнику соответствует первый, упомянутый в тексте. Оформление списка – согласно СТП БГТУ 002-2007.

Перечисления оформляются с использованием знака короткого тире, в случае многоуровневого перечисления сначала используются знаки короткого тире, затем строчные буквы кириллицы, отделяемые от последующего текста круглой скобкой, далее – нумерационное перечисление, также отделяемое круглой скобкой.

Если вводятся какие-то сокращения или условные обозначения, то они должны быть оформлены соответствующим списком условных сокращений или обозначений и прописаны после первого упоминания в тексте в круглых скобках. Например, *пояснительная записка курсовой работы (ПЗ) включает титульный лист, задание, реферат, содержание, введение и т. д.*

Для выравнивания текста по высоте допускается использовать междустрочный интервал в пределах одной страницы в значении Множитель от 0,99 до 0,95 и от 1,01 до 1,05.

Все рисунки должны иметь порядковую нумерацию в пределах раздела, к которому они относятся, и предварительно обозначены в записке. Если размер рисунка не позволяет вставить его сразу после абзаца и его уменьшение до нужных размеров приводит к потере информации, то допускается его перенос на следующую страницу. Но в этом случае свободное поле листа заполняется последующим текстом, а сам рисунок размещается в начале следующей страницы по правилам оформления рисунков. Для самого рисунка рекомендуется использовать стиль Рисунок, а для подрисуночной подписи – стиль Подрисуночная подпись в текущем шаблоне.

Допускается ссылка на рисунок, размещаемый в приложении. В этом случае само приложение нумеруется прописными буквами кириллицы (например, ПРИЛОЖЕНИЕ А), а ссылка на рисунок в приложении соответствует его порядковому номеру в этом приложении. Например, форма регистрации (создание аккаунта) приведена на рисунке А5, т. е. рисунок размещается в прил. А и является пятым по счету в этом приложении.

При оформлении таблиц размер шрифта должен быть 14 пт. Допускается поворот таблицы по часовой стрелке на 90°, в этом случае она должна быть размещена на отдельной странице, следующей после первого

упоминания этой таблицы. Оставшийся до следующей страницы интервал заполняется текстом, следующим по смыслу.

Для ссылки на таблицу применяется то же правило, что и для ссылки на рисунок. Например, информация о параметрах используемого графического материала для главной страницы сайта приведена в табл. 4.3. Последующий текст должен быть отделен от таблицы интервалом в 14 пт.

Для оформления заголовка таблицы и ее продолжения используйте стиль Заголовок таблицы. Чтобы отделить текст от таблицы – стиль Текст после таблицы.

Формулы, при их наличии, набираются в редакторе формул. Для этого используется Меню – Вставка – Объект... – MathType 6.0 Equation или Microsoft Equation 3.0. Формулы оформляются в соответствии со стилем Формула. Для того чтобы она стала строго по центру в начале строки, нажимается клавиша [Tab], а после набора формулы еще раз [Tab].

Список использованных источников включает все использованные информационные источники в порядке появления ссылок на них в тексте, помещается после изложения текстового материала перед приложением, нумеруется арабскими цифрами без точки и печатается с абзацного отступа. Список источников оформляется по ГОСТ 7.1.

Пояснительная записка распечатывается на формате А4 с одной стороны, листы нумеруются внизу посередине, начиная со страницы 3, пропуская титульный лист и лист задания. Все листы пояснительной записки, включая графическую часть и электронное приложение, оформленное указанным способом, подшиваются в папку. К защите не принимаются не скрепленные воедино листы.

Требования к оформлению электронного приложения

Электронное приложение представляет собой CD / DVD / флеш диск, который содержит следующие каталоги / папки:

Web-проект. Содержит все необходимые папки с разработанным web-сайтом. Сюда же копируются вспомогательные плагины, программы для корректного отображения web-сайта.

Документация. Содержит файл с текстом пояснительной записки.

Файл READ.me. Текстовый файл, содержащий название проекта, автора, сведения по запуску web-сайта.

CD / DVD / флеш диск на титульной стороне содержит наклейку (прил. Е) и крепится в папку с пояснительной запиской.

В случае сдачи проектов на флеш-диске составляется описание содержимого флеш-диска.

Критерии оценки курсовой работы

Перед предъявлением работы к защите необходимо проверить:

- соответствие названия темы курсовой работы, указанной на титульном листе, названию, утвержденному заведующим кафедрой;
- идентичность заголовков в оглавлении и в работе, а также их общую редакционную согласованность;
- правильность подкладки листов (их последовательность и размещение относительно корешка);
- правильность нумерации рисунков, таблиц, приложений;
- общую редакционную согласованность таблиц и надписей;
- наличие ссылок на рисунки, таблицы, приложения, литературу;
- правильность ссылок;
- наличие подписи студента на титульном листе;
- отсутствие пометок и элементов оформления в непечатном виде;
- наличие сквозной нумерации страниц и соответствие ей содержания;
- наличие электронной версии на электронном носителе.

Примерный перечень тем курсовых работ

1. Электронное мультимедийное издание «Жизнь и творчество Анны Ахматовой».
2. Электронное издание «Поэты Серебряного века».
3. «Путешествие по Березинскому заповеднику». Лонгрид.
4. Лелуш Ламперуж (Lelouch Lamperouge). Лонгрид.
5. Произведение А. Шолохова «Тихий дон» в книгах и на экране.
6. Web-сайт «Все о героях произведения Хаяо Миядзаки «Ходячий замок»».
7. Электронная кулинарная книга.
8. Web-сайт типографии «Ух, напечатаем!»
9. Web-сайт «История типографики».
10. Электронная газета «Экология и мы».

ПРИЛОЖЕНИЕ А

Регистрационная форма

Утверждена постановлением Министерства
связи и информатизации Республики Беларусь
05.08.2009 № 32

ЗАЯВКА на государственную регистрацию информационного ресурса

Заявитель _____

Прошу (нужное подчеркнуть)

зарегистрировать	изменить
------------------	----------

 информационный ресурс (далее – ИР)

Регистрационный номер ИР _____

Краткое наименование ИР

--

Полное наименование ИР

--

Сведения о собственнике (владельце) ИР

Идентификационные сведения:

Наименование ЮЛ¹/ФИО ИП²

--

--

Адрес

--	--

Индекс

Область, город, район,
улица, дом, квартира

Учетный номер плательщика

--

Свидетельство о государственной регистрации собственника (владельца):

Номер	
Дата выдачи	

¹ ЮЛ – юридическое лицо

² ИП – индивидуальный предприниматель

Государственный орган (организация)

Телефон Факс
E-mail

Сведения о разработчике ИР

Идентификационные сведения:

Наименование ЮЛ/ФИО ИП

Адрес

Индекс Область, город, район, улица, дом, квартира

Учетный номер плательщика

Свидетельство о государственной регистрации разработчика:

Номер	<input type="text"/>
Дата выдачи	<input type="text"/>

Государственный орган (организация)

Телефон Факс
E-mail

Сведения об авторах ИР

Фамилия, собственное имя и отчество автора (соавторов)

<input type="text"/>
<input type="text"/>
<input type="text"/>
<input type="text"/>
<input type="text"/>

Сведения о подразделении, осуществляющем ведение ИР

Наименование подразделения

Фамилия, инициалы администратора ИР	<input type="text"/>		
Телефон	<input type="text"/>	E-mail	<input type="text"/>

Общие сведения об ИР

Тип ИР

Рубрикация ИР

Содержательная характеристика ИР

Объем ИР Мб записей (для ИР типа «база данных») Язык(и)

Год создания Охват – временной интервал с г. по г.

Источник формирования ИР

--

Обновление

Да	Нет
----	-----

 Период содержательного обновления

Разработка и взаимодействие ИР

ИР является результатом выполнения научно-исследовательской или опытно-конструкторской работы (далее – НИОКР) (нужное подчеркнуть)	Да	Нет
--	----	-----

Наименование НИОКР	<input type="text"/>
--------------------	----------------------

Источники финансирования ИР (нужное подчеркнуть)

Республиканский бюджет	Местный бюджет	Внебюджетные фонды
Собственные средства	Средства иных юридических лиц	Иностранные гранты

Входит в состав информационной системы

Взаимодействует с другими ИР (нужное подчеркнуть) Да Нет

Краткие наименования ИР, взаимодействующие с данным ИР

Пользовательские характеристики ИР

Программная среда работы с ИР

Размещение (местоположение или носитель) ИР (нужное подчеркнуть)

Интернет Локальная сеть Автономный компьютер CD/DVD Другое

Интернет-адрес

Услуги (нужное подчеркнуть) Разовое обращение Фрагмент ИР Поставка ИР Другое

Поставка: формат носитель

Ограничения по доступу к информации (нужное подчеркнуть) Да Нет

Коммерческая тайна	Персональные данные	Патентная заявка	Служебная тайна
--------------------	---------------------	------------------	-----------------

Требования к обеспечению безопасности ИР (нужное подчеркнуть) Да Нет

Состав информационных объектов и идентификаторов доступа

Наименование информационного объекта	Описание информационного объекта	Идентификатор

Заявитель _____

(подпись)

М.П.

(инициалы, фамилия)

Дата «__» _____ 20__ г.

ПРИЛОЖЕНИЕ Б

Пример дизайн-макета

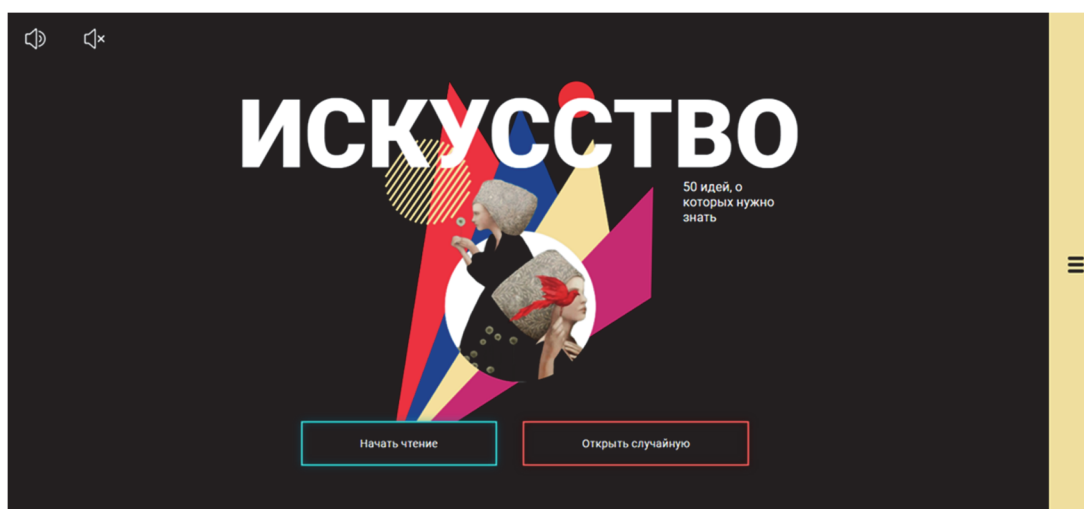


Рисунок А1 – Пример дизайн-макета

Работа Аннейчик И. А., студентка специальности «Издательское дело», 2020 г. Тема курсовой работы: «Разработка электронного издания “Искусство. 50 идей, о которых нужно знать” С. Ходж».



Рисунок А2 – Пример дизайн-макета

Работа Жуковой А. к курсовой работе «Электронная библиотека для детей», студентка специальности «Дизайн электронных и web-изданий», 2020 г.

ПРИЛОЖЕНИЕ В

Образец титульного листа

УО «Белорусский государственный технологический университет»

Факультет информационных технологий
Кафедра «Информатика и web-дизайн»
Специальность 1-47 01 01 «Издательское дело»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА КУРСОВОЙ РАБОТЫ

по дисциплине «WEB-дизайн и разработка мультимедийных изданий»
тема « _____ »

Исполнитель

Студент ___ курса ___ группы

подпись, дата

инициалы и фамилия

Руководитель

подпись, дата

инициалы и фамилия

Курсовая работа защищена с оценкой

Руководитель _____
(подпись)

(отметка и дата)

инициалы и фамилия

Минск 2021

ПРИЛОЖЕНИЕ Г

Форма листа задания

Учреждение образования
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий
Кафедра информатики и веб-дизайна
Специальность 1-47 01 01 «Издательское дело»

Утверждаю
Заведующий кафедрой
_____ Д. М. Романенко
«__20__» _____ 02 _____ 20__ г.

Задание
по курсовой работе
дисциплина «WEB-дизайн и разработка мультимедийных изданий»

Студенту _____

1. Тема « »
2. Сроки защиты «_1_»__ _____ 2021_ г.
3. Содержание пояснительной записки курсовой работы. Пояснительная записка по курсовой работе должна быть выполнена в текстовом редакторе Word. Она должна включать:
 - а) титульный лист;
 - б) задание по курсовой работе;
 - в) введение;
 - г) описание поставленной в работе задачи и этапов ее выполнения;
 - д) литературу;
 - е) оглавление. Форма представления выполненной курсовой работы: в печатном виде необходимо представить пояснительную записку. В электронном виде – файлы выполненной работы, при необходимости файлы фреймворков, библиотек, пояснительную записку.
4. Консультанты _____

5. Календарный график работы

1)	Аналитический обзор Определить цели и задачи работы. Описать предметную область. Сделать обзор не менее 4 аналогов. Обосновать выбор программных средств и технологий для реализации проекта.	Выполнить до
2)	Проектирование Разработать структуру проекта. Описать функционал и содержание страниц.	Выполнить до
3)	Дизайн. Определить стиль проекта, разработать логотип, разработать или выбрать шрифтовое решение, определить цветовое решение. Разработать дизайн-макеты (уникальные и типовые страницы, 404 страница и пр.).	Выполнить до
4)	Разработка и верстка Обработка контента, создание видео и внимания. Описание верстки, CSS, и JS –скриптов. Валидация (https://codebeautify.org/cssvalidate , https://validator.w3.org) и ее результаты (обзор ошибок и способы их устранения).	Выполнить
5)	Подготовка пояснительной записки.	Выполнить до
6)	Защита проекта.	

Дата выдачи задания «_ 0 _» _____ 2021 г.

Руководитель _____
(подпись) _____ ФИО

Задание принял к исполнению _____ ()

СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

1. Гарретт, Д. Web-дизайн. Элементы опыта взаимодействия / Д. Гарретт. – СПб.: Символ-плюс, 2015. – 192 с.
2. Дакетт, Д. HTML и CSS. Разработка и дизайн web-сайтов / Д. Дакетт. – М.: Эксмо, 2018. – 208 с.
3. Дакетт, Д. HTML и CSS. Разработка и дизайн web-сайтов / Д. Дакетт. – М.: Эксмо, 2015. – 480 с.
4. Кирсанов, Д. Web-дизайн: книга Дмитрия Кирсанова / Д. Кирсанов. – СПб.: Символ-плюс, 2015. – 376 с.
5. Кирсанов, Д. Web-дизайн: книга Дмитрия Кирсанова / Д. Кирсанов. – М.: Символ, 2015. – 368 с.
6. Киселев, С. В. Web-дизайн: учеб. пособие / С. В. Киселев. – М.: Academia, 2018. – 416 с.
7. Киселев, С. В. Web-дизайн: учеб. пособие / С. В. Киселев. – М.: Academia, 2017. – 312 с.
8. Киселев, С. В. Web-дизайн: учеб. пособие / С. В. Киселев. – М.: Academia, 2018. – 384 с.
9. Киселев, С. В. Web-дизайн / С. В. Киселев. – М.: Academia, 2017. – 152 с.
10. Киселев, С. В. Web-дизайн / С. В. Киселев. – М.: Academia, 2019. – 285 с.
11. Макнейл, П. Web-дизайн. Книга идей web-разработчика / П. Макнейл. – СПб.: Питер, 2017. – 480 с.
12. Нильсен, Я. Web-дизайн: книга Якоба Нильсена / Я. Нильсен. – М.: Символ, 2015. – 512 с.
13. Потапенко, Н. Основы веб-дизайна / Н. Потапенко, М. Кудлацкая, А. Олеферович. – Минск: БГТУ, 2019. – 90 с.
14. Сырых, Ю. Современный web-дизайн. Настольный и мобильный / Ю. Сырых. – М.: Диалектика, 2019. – 384 с.
15. Гурский, Юрий. Компьютерная графика Photoshop CS5, CorelDRAW X5, Illustrator CS5. Трюки и эффекты / Юрий Гурский, Андрей Жвалевский, Владимир Завгородний. – М.: Питер, 2017. – 704 с.
16. Стьюер, Шерон. Креативное мышление в Photoshop. Новый подход к цифровому искусству / Шерон Стьюер. – М.: ИТ Пресс, 2017. – 272 с.

Учебное издание

WEB-ДИЗАЙН И РАЗРАБОТКА МУЛЬТИМЕДИЙНЫХ ИЗДАНИЙ

Учебно-методическое пособие

Составители:

Потапенко Наталья Ивановна
Щербакова Алина Николаевна

Редактор *Е. И. Гоман*
Компьютерная верстка *А. Н. Петрова*
Корректор *Е. И. Гоман*

Издатель:

УО «Белорусский государственный технологический университет».
Свидетельство о государственной регистрации издателя,
изготовителя, распространителя печатных изданий
№ 1/227 от 20.03.2014.
Ул. Свердлова, 13а, 220006, г. Минск.