

ПЕРМСКИЙ  
ГОСУДАРСТВЕННЫЙ  
НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ

**Н. Н. Василюк**

**СОЗДАНИЕ  
И АДМИНИСТРИРОВАНИЕ  
САЙТОВ**



Пермь 2023

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное  
образовательное учреждение высшего образования  
«ПЕРМСКИЙ ГОСУДАРСТВЕННЫЙ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»

Н. Н. Василюк

## СОЗДАНИЕ И АДМИНИСТРИРОВАНИЕ САЙТОВ

*Допущено методическим советом  
Пермского государственного национального  
исследовательского университета в качестве  
учебного пособия для студентов, обучающихся  
по направлениям подготовки бакалавров  
гуманитарных и естественнонаучных направлений подготовки,  
не специализирующихся на изучении информатики  
и информационных технологий*



Пермь 2023

УДК 004.65(075.8)  
ББК 32.972.134я73  
В19

**Василюк Н. Н.**

В19 Создание и администрирование сайтов [Электронный ресурс] : учебное пособие / Н. Н. Василюк ; Пермский государственный национальный исследовательский университет. – Электронные данные. – Пермь, 2023. – 2,03 Мб ; 92 с. – Режим доступа: <http://www.psu.ru/files/docs/science/books/uchebnie-posobiya/Vasilyuk-Sozdanie-i-administrirovanie-sajtov.pdf>. – Заглавие с экрана.

ISBN 978-5-7944-4046-1

Учебное пособие содержит теоретические сведения, упражнения и задания для самостоятельной работы студентов по курсу «Создание и администрирование сайтов».

Издание предназначено для проведения практических занятий со студентами факультетов, не специализирующихся на изучении информатики и информационных технологий.

**УДК 004.65(075.8)**  
**ББК 32.972.134я73**

*Издается по решению ученого совета механико-математического факультета  
Пермского государственного национального исследовательского университета*

*Рецензенты:* кафедра управления персоналом АНО ВО «Гуманитарный университет» (рецензент – зав. кафедрой, канд. психол. наук, доцент **Т. Л. Сморкалова**);

зав. кафедрой общей и прикладной психологии АНО ВО «Гуманитарный университет», канд. психол. наук, доцент **Л. В. Тарасова**

ISBN 978-5-7944-4046-1

© ПГНИУ, 2023  
© Василюк Н. Н., 2023

## ОГЛАВЛЕНИЕ

<b>Введение</b> .....	4
<b>Методические указания</b> .....	4
Контрольные мероприятия .....	4
<b>Цели и задачи курса «Создание и администрирование сайтов»</b> .....	5
<b>Глава 1. Создание проекта сайта</b> .....	6
Планирование .....	6
Дизайн .....	11
Разработка макета сайта с помощью программы Figma.....	13
Создание сайта .....	22
Задания для самостоятельной работы к главе 1 .....	22
<b>Глава 2. Создание сайта с помощью HTML</b> .....	23
Теоретические сведения об HTML .....	23
Упражнения для изучения HTML .....	45
Задания для самостоятельной работы к главе 2 .....	60
<b>Глава 3. Создание сайта с помощью онлайн-конструкторов</b> .....	64
Google Sites .....	64
Конструктор сайтов Tilda.....	69
Конструктор сайтов UCoz.....	73
Конструктор сайтов Fo.ru.....	76
Задания для самостоятельной работы к главе 3 .....	78
<b>Глава 4. Создание сайта с помощью CMS</b> .....	83
Задания для самостоятельной работы к главе 4 .....	88
<b>Список использованной литературы</b> .....	91

## ВВЕДЕНИЕ

В результате изучения курса студент получает представление о возможностях создания сайтов с помощью языка разметки гипертекста (HTML), онлайн конструкторов и CMS.

Для освоения материала предлагаемого пособия необходимо иметь навыки работы:

- с ОС Windows;
- с файлами и папками в Windows;
- с графическими редакторами.

**Цель пособия** – сформировать навыки разработки сайтов, подготовить студентов к созданию и размещению собственных сайтов в сети Интернет.

По окончании изучения пособия студент должен научиться создавать:

- простейший проект сайта;
- сайт с помощью HTML;
- сайт с помощью разных онлайн-конструкторов;
- сайт с помощью одной из CMS.

## МЕТОДИЧЕСКИЕ УКАЗАНИЯ

### *Примерное содержание практических занятий (28 ч)*

№ п/п	Тема	Количество
1	Проектирование сайта. Подбор дизайна. Знакомство с Figma	4
2	Изучение языка разметки HTML. Создание страниц сайта	8
3	Создание сайта с помощью онлайн-конструкторов: Google Sites, Tilda, Ucoz и др.	8
4	Создание сайта с помощью CMS (Word Press, Joomla). Работа с Open Server или иным локальным сервером	8

### *Контрольные мероприятия*

Работа студентов по изучению курса оценивается по промежуточным самостоятельным заданиям и результатам прохождения итогового тестирования.

## ЦЕЛИ И ЗАДАЧИ КУРСА «СОЗДАНИЕ И АДМИНИСТРИРОВАНИЕ САЙТОВ»

Содержание учебно-методического комплекса «Создание и администрирование сайтов» охватывает такие темы, как основы HTML, основы планирования структуры сайта, доставка на сервер, создание и управление сайтом с помощью онлайн конструкторов, установка и управление сайтом с помощью CMS-системы, основы продвижения и оптимизации сайта, информационное и техническое обслуживание сайта. Содержание этой дисциплины приводится в табл. 1.

Таблица 1

### Содержание курса «СиАС»

№ п/п	Раздел содержания
1	История развития сети Интернет и ее устройство. Теоретические основы построения веб страниц
2	Проектирование и создание сайта: структура, дизайн, контент. Выбор площадки для развертывания
3	Веб сервера. Сборки веб серверов. Развертывание сайта
4	Создание сайта средствами HTML. Особенности создания и администрирования. Публикация подобного сайта в Интернете
5	Создание и управление сайтом на шаблонах бесплатных конструкторов (Google, wix, ucoz). Сравнительная оценка различных ресурсов
6	Системы управления контентом. Обзор существующих CMS и их предназначений. Создание сайта средствами Joomla! Работа с административной панелью. Публикация сайта в Интернете
7	Администрирование сайтов. Информационное и техническое обслуживание сайтов

Методика преподавания дисциплины «Создание и администрирование сайтов» с предложенным содержанием включает в себя такие методы обучения, как *репродуктивный, частично-поисковый и поисковый*.

Студенты учатся создавать и администрировать сайты как под руководством преподавателя, следуя инструкции (*репродуктивный метод обучения*), а затем самостоятельно, владея частичными знаниями о возможностях той или иной платформы или не владея ими совсем (*частично-поисковый и поисковый метод*).

# ГЛАВА 1. СОЗДАНИЕ ПРОЕКТА САЙТА

В процессе создания сайта можно выделить следующие этапы:

- планирование;
- дизайн;
- разработка.

Разделение является условным, поскольку планирование и разработка могут сливаться в один этап, который можно назвать «подготовительным перед разработкой». Рассмотрим предложенные три этапа.

## Планирование

Этап планирования можно разбить на несколько подэтапов, а именно:

- создание идеи;
- разработка структуры проекта;
- проработка макета проекта.

### *Создание идеи*

На этом этапе нужно определиться с тематикой проекта (сайта). Затем, в соответствии с выбранной темой, необходимо подобрать соответствующие материалы, например, текстовые, графические, аудио или видео.

### *Разработка структуры проекта*

После того, как мы определили тему проекта, собрали нужный материал, необходимо начать разработку структуры проекта. Структура проекта включает разделы сайта, в соответствии с которыми будет формироваться меню для навигации и создаваться дизайн проекта. На этом этапе также можно проводить классификацию материалов по темам и разделам.

### *Работа над макетом проекта*

После того, как мы определили структуру проекта, можно составить макет проекта. Для создания наброска можно использовать бумагу и ручку или любой графический редактор (пример см. на рис. 1). Отметим, что на данном этапе не выполняется отрисовка готового дизайн-макета, а лишь создается схематичный набросок, чтобы определить, как на сайте будут располагаться основные информационные блоки, графические элементы и пр.

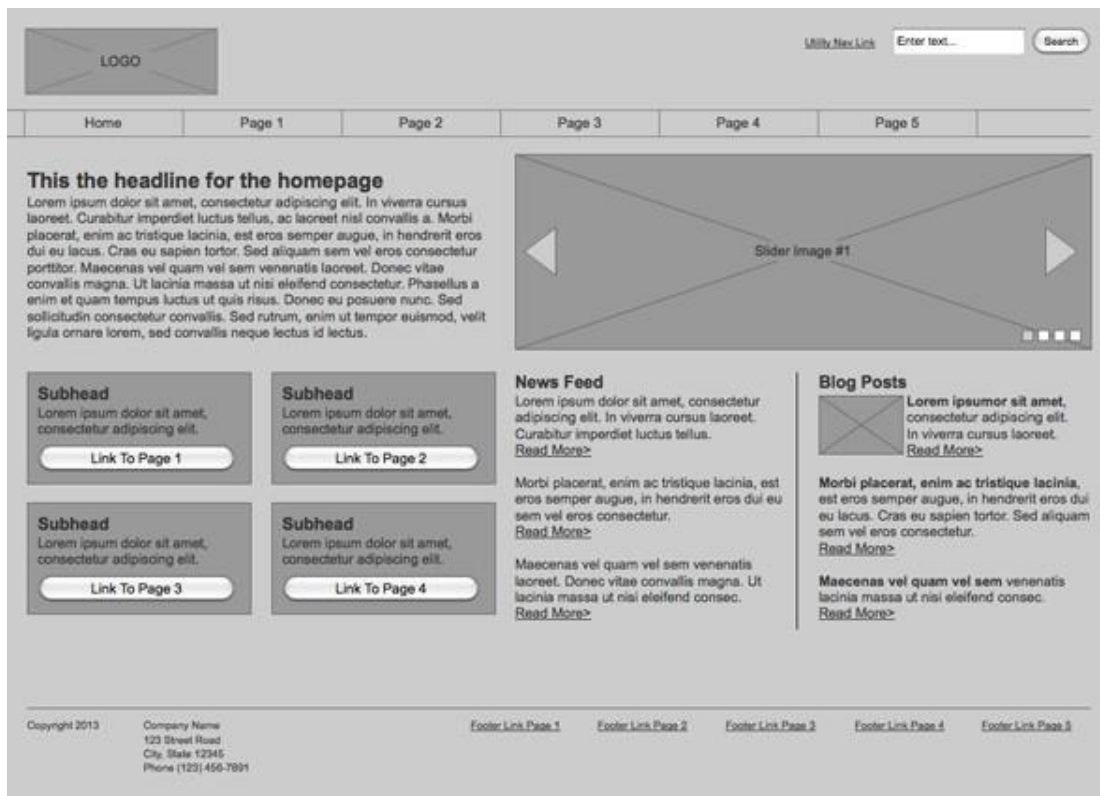


Рис. 1. Пример проекта сайта, выполненный в графическом редакторе

### **Основные элементы страницы**

Основные элементы страницы перечислены ниже (см. рис. 2). Разумеется, не каждая веб-страница должна иметь в составе все перечисленные элементы: *содержащий блок (container), логотип, навигация, контент, футер (нижний колонтитул), свободное пространство.*

Свободное пространство – это некое понятие, о котором не стоит забывать при составлении макета страницы, и тогда готовый проект не будет выглядеть как нагромождение блоков. Остальные элементы рассмотрим отдельно.

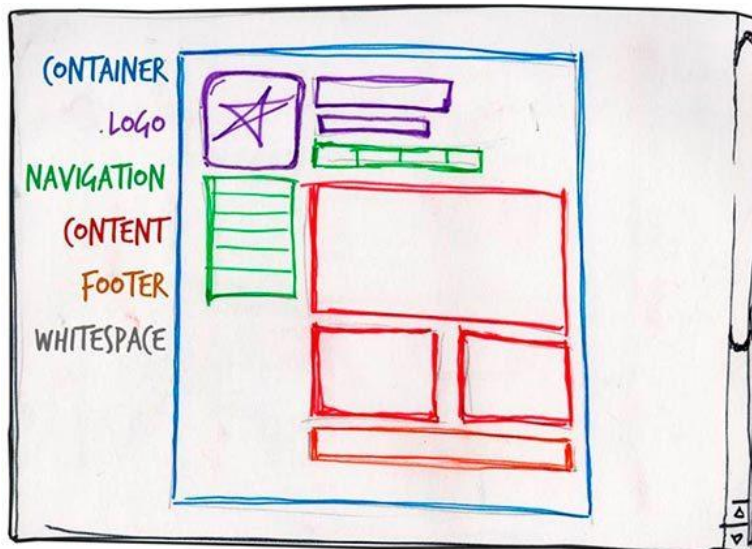


Рис. 2. Пример проекта сайта с основными элементами страницы



### ***Содержащий блок (контейнер)***

Роль контейнера на странице выполняет элемент *body*, или же элемент *div*. Ширину содержащего блока можно сделать резиновой (*fluid*) или фиксированной (*fixed*).

### ***Логотип***

Логотип – это текстовая или графическая составляющая проекта, выделяющая его среди других. Чаще всего он расположен в верхнем левом углу страницы или посередине (в зависимости от макета).

### ***Навигация***

На основной навигационной панели расположены ссылки на основные разделы сайта. Навигационная панель зачастую находится в верхней части страницы (сами элементы навигации могут располагаться вертикально или горизонтально).

### ***Контент***

Контент – это основная составляющая веб-страницы, то, ради чего создается сайт. Он доминирует в дизайне страницы, занимая самое большое пространство. Помимо текста контент практически всегда подкрепляется графикой.

### ***Нижний колонтитул (footer)***

Располагается внизу страницы и, как правило, может содержать информацию о правообладателе, контактные и юридические данные, ссылки на основные разделы сайта (иногда дублирует основную навигацию), ссылки на социальные сети, форму обратной связи и пр.

### ***Резиновый и фиксированный макет***

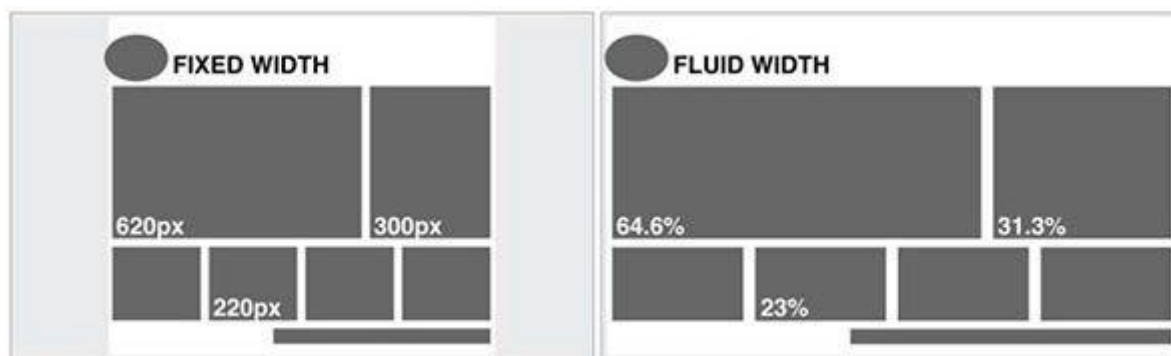


Рис. 3. Пример фиксированного и резинового макетов сайта

Создание фиксированного макета подразумевает, что независимо от разрешения экрана пользователя созданный сайт всегда будет занимать одинаковую ширину.

Создание резинового макета подразумевает, что страница сайта будет стараться занять всё доступное ей пространство на экране пользователя, подстраиваясь под разрешение.

### Модульная сетка

Перед составлением схемы проекта следует рассмотреть понятие модульной сетки. Модульная сетка – это разделение страницы на отдельные колонки по вертикали. При разработке дизайн макета именно по этой сетке выстраивается контент.

Наиболее популярной системой считается модульная сетка *960 Grid System* (<http://960.gs>), которая максимально делит страницу на 12, 16 и 24 колонки. Максимум в ширине сетка имеет 960 пикселей. Данное решение основано на том, что на момент создания сетки большинство мониторов имели разрешение не меньше 1024 на 768 пикселей.

Создание макета на основе модульной сетки помогает ускорить процесс разработки дизайна сайта.

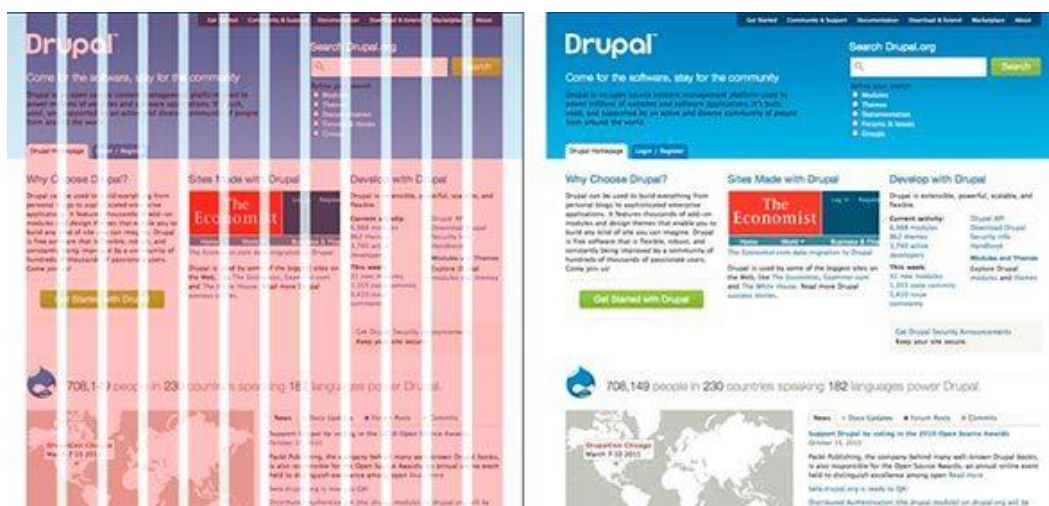


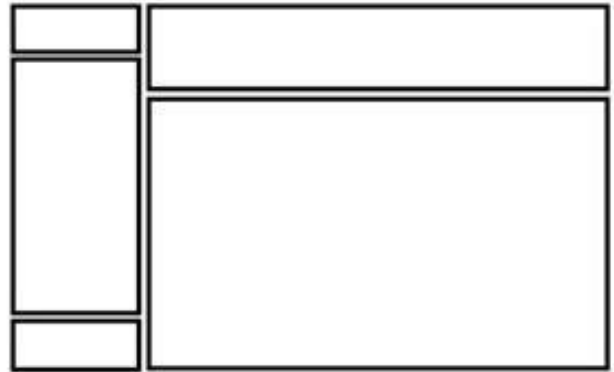
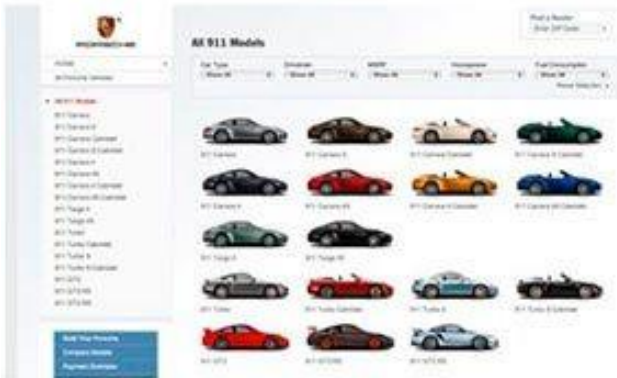
Рис. 4. Пример модульной сетки

Отметим, что при разработке «резинового» макета страницы существует понятие максимальной ширины, основанное на удобстве восприятия информации. Если разрабатываемый сайт не имеет максимального значения по ширине, то на больших мониторах информация будет сильно растягиваться, что обуславливает неудобство чтения. Чаще всего ограничиваются шириной в 1280 пикселей.

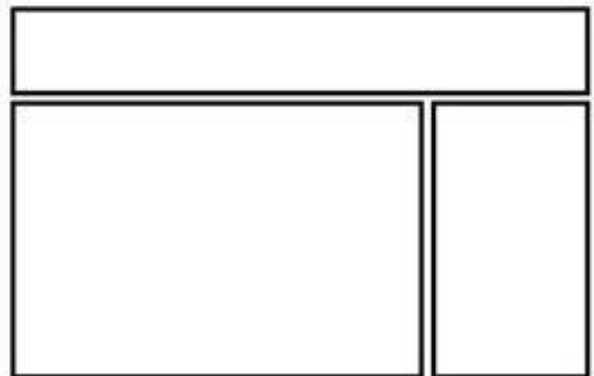
## Макеты веб-страниц

Среди многообразия макетов веб-страницы можно выделить четыре наиболее распространённых:

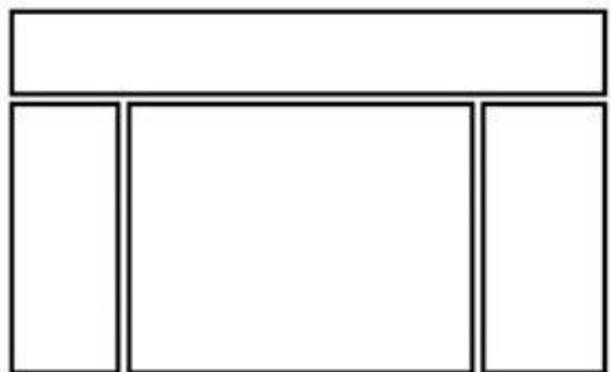
- **Навигация в левом столбце**



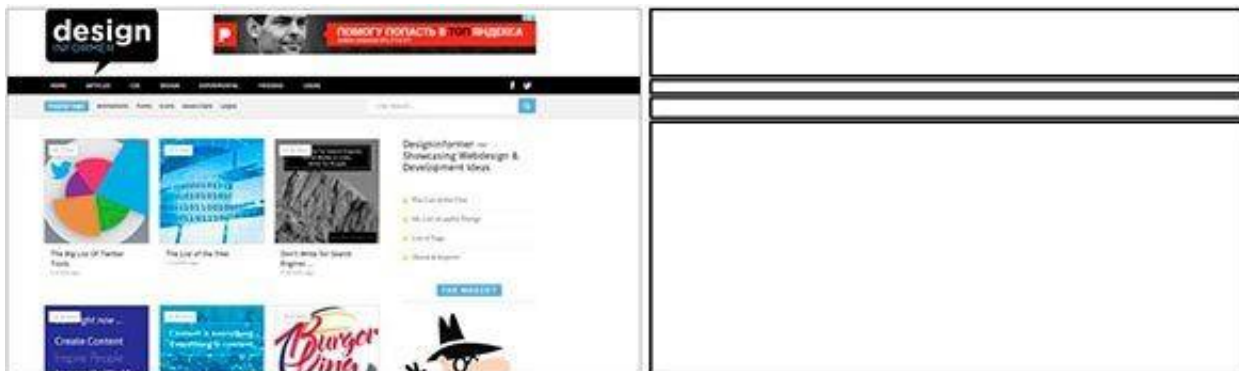
- **Навигация в правом столбце**



- **Навигация в трёх столбцах**



- **Горизонтальная навигация.** На данном этапе сайты с таким типом навигации составляют большинство. Удобство такого подхода легко объяснить тем, что в данном случае остаётся больше пространства для контента, составляющего разрабатываемый сайт.



Разумеется, этот список макетов нельзя назвать исчерпывающим, поэтому перед разработкой макета и дизайна страницы необходимо рассмотреть различные сайты с примерами других работ, а также сайты с аналогичной тематикой.

### *Принцип «Mobile First»*

С учётом тенденций последних лет данный подход все чаще встречается при разработке дизайна сайтов. Тенденция такова, что почти 60% пользователей Всемирной паутины используют для доступа в сеть мобильные устройства, поэтому правилом хорошего тона становится разработка не только десктопной версии сайта, но и мобильной версии. При использовании данного подхода разработка макета сайта, дизайна и вёрстки начинается с мобильной версии, а затем уже прорабатываются макеты для других разрешений: добавляются блоки, баннеры, дополнительные элементы дизайна и пр.

### **Дизайн**

После создания макета проекта можно приступить к созданию дизайн-макета непосредственно: определим цветовую гамму проекта.

Один из способов определения основного цвета в проекте – это составление *mood board*. Для этого нужно выписать все синонимы, связанные с темой проекта, а затем каждый синоним набирается в поиске по картинкам в Google или Yandex. На основе найденных изображений выписываются цвета, которые чаще всего встречаются на них, чтобы выяснить, каких цветов больше.

Полученные цвета позволяют определить визуальное восприятие нашего проекта, вызывая у пользователя соответствующие чувства.



Рис. 5. Пример Mood board

Для работы с выбранным цветом и составления палитры цветов нашего сайта можно использовать следующие инструменты:

- **Color Scheme Designer 3** (<http://colorschemedesigner.com/csd-3.5/>).

Помимо выбора цветовой схемы данный сервис позволяет посмотреть пример того, как выбранные цвета будут смотреться на сайте.



Рис. 6. Сервис Color Scheme Designer 3

- **Adobe Color CC** (<https://color.adobe.com/ru/>). Данный ресурс позволяет создавать палитры на основе загруженных изображений (которые, например, могли появиться у нас на этапе составления mood board). Также данный сервис обладает большим архивом палитр других пользователей.

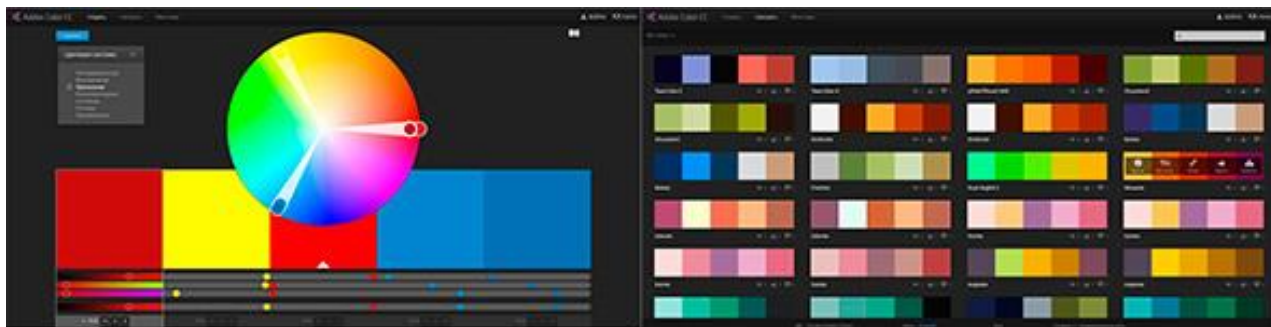


Рис. 7. Сервис Adobe Color CC

- **COLORlovers** (<http://www.colourlovers.com/>) – обширное сообщество, где можно подобрать различные палитры.

Отметим, что при подборе цветов для палитры нужно выбирать как минимум 2 контрастирующих цвета. Достижение нужного контраста между цветами – необходимое условие для того, чтобы получить хороший интерактивный дизайн.

### **Разработка макета сайта с помощью программы Figma**

*Figma* (Фигма) – это графический онлайн-редактор для совместной работы. В нём можно создать прототип сайта, интерфейс приложения и обсудить правки с коллегами в реальном времени. В Фигме можно отрисовать элементы интерфейса, создать интерактивный прототип сайта и приложения, иллюстрации, векторную графику. Многие дизайнеры делают в ней макеты сайтов для онлайн-конструктора сайтов Тильда.

Далее уточним некоторые понятия. *Прототип* – это модель сайта или приложения. Он позволяет оценить, как люди будут пользоваться продуктом. Чтобы создать прототип сайта, дизайнер отрисовывает экраны и создаёт связи между ними.

*Элементы интерфейса* – это внешний вид продукта. В Фигме можно создать кнопки, иконки, формы обратной связи и настроить эффекты: сделать кликабельные кнопки, раскрыть списки, создать анимацию для блоков и др.

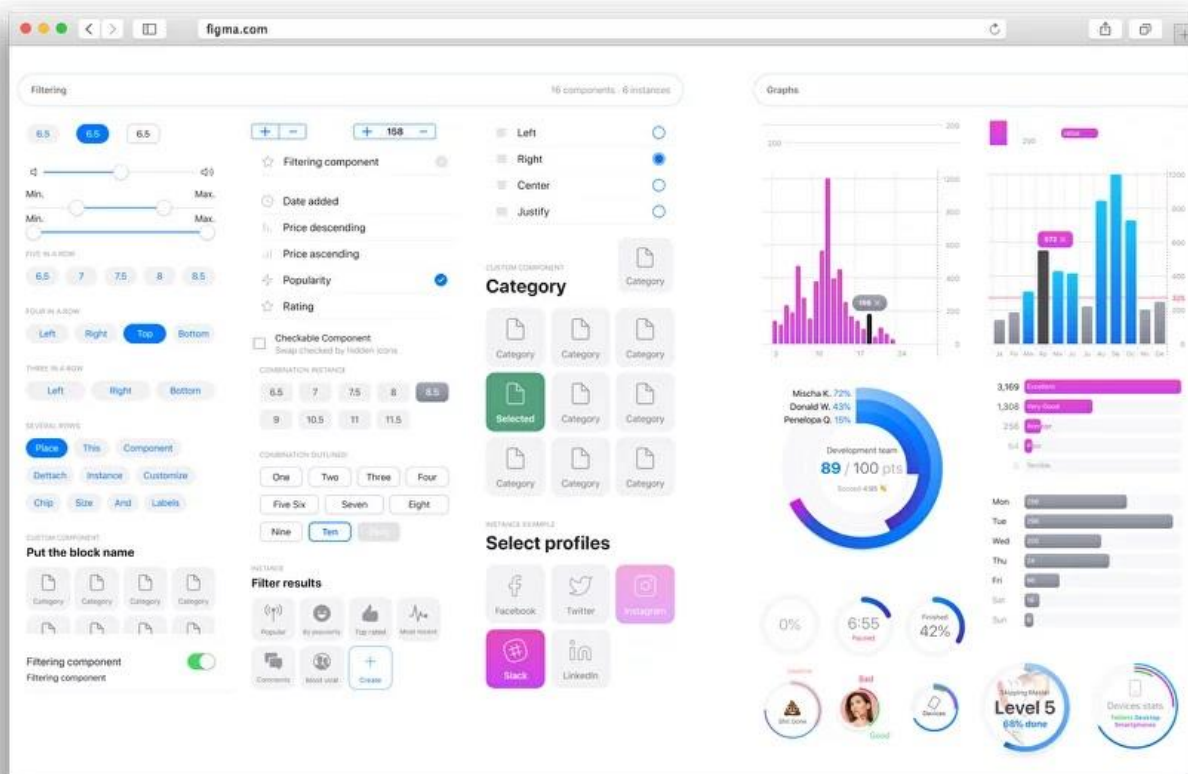


Рис. 8. Примеры интерфейса, созданного в Фигме

В Фигме есть основные инструменты для работы с векторными объектами, позволяющие экспортировать дизайн в формат SVG, импортировать векторные объекты из Adobe Illustrator или редактора Sketch.

### ***Работа в Фигме***

В Фигме можно работать из браузера или скачать приложение Фигма на компьютер. Если дизайнер пользуется разными устройствами, то нужно выбрать версию для браузера: зайти на сайт Фигмы [figma.com](https://figma.com) и зарегистрироваться.

Можно также бесплатно скачать и установить Фигму на свой компьютер, что поможет избавиться от многочисленных вкладок браузера перед глазами. Для этого нужно перейти по ссылке [figma.com/downloads](https://figma.com/downloads) и выбрать версию программы для iOS или Windows.

### ***Интерфейс Фигмы***

После регистрации пользователю доступны две рабочие области – графический редактор и менеджер файлов, в последнем можно создавать проекты и изменять настройки профиля.

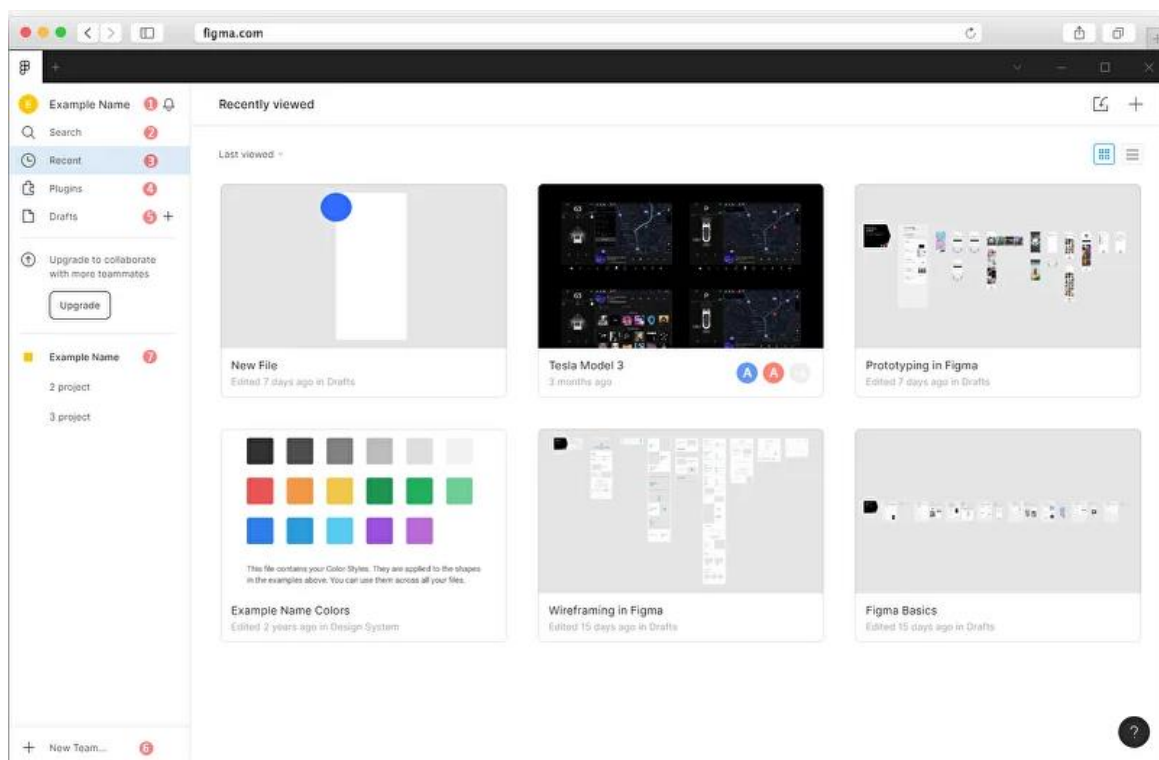


Рис. 9. Окно Figma в браузере

Далее рассмотрим пункты меню менеджера файлов.

- **Профиль** (*Example Name* – в том случае, когда программа открывается впервые). В настройках профиля можно загрузить аватар, сменить имя, пароль и email, на который приходят оповещения. Кроме того, там можно обновить тариф и удалить аккаунт.
- **Поиск** (*Search*). Файлы и проекты, в которых вы участвуете, можно найти через поиск *Search*. Чтобы было легче найти файлы, с которыми нужно работать в будущем, необходимо давать им разные имена.
- **Последние файлы** (*Recent*). В Фигме файлы сохраняются автоматически. Все файлы, которые открывал пользователь, можно увидеть во вкладке *Recent*. Фигма сохраняет историю изменений каждого файла, т.е. можно легко восстановить резервную копию своей работы.
- **Плагины** (*Plugins*). В пункте **Plugins** собраны расширения, которые помогут ускорить работу в Фигме. Например, с плагином **Unsplash** можно находить стоковые иллюстрации, не выходя из Фигмы. А с **Iconify** в распоряжении дизайнера будут 40 000 векторных иконок на все случаи жизни. Плагины отсортированы по категориям «Рекомендуемые», «Популярные» и «Установленные». Чтобы увидеть все плагины, нужно нажать кнопку **Browse all plugin**.
- **Новый файл** (*Drafts*). Создать новый файл в Фигме можно через пункт **Drafts** или **New File** в правом углу экрана. По умолчанию файл называется **Untitled**.




- **Команда** (*New Team*). Чтобы начать совместный проект, следует создать команду: нажать **New Team** и придумать название. Можно пригласить участников по email и обозначить их роли: только просмотр или редактирование файла. На бесплатном тарифе права редактора могут быть у двух человек, включая самого пользователя.

- **Проекты** (*Example Name* – в том случае, когда программа открывается впервые). По умолчанию при добавлении команды (см. выше) Фигма создаёт для неё новый проект. Проект – это папка. В ней команда хранит все файлы, которые относятся к одному проекту. Это удобно, если в каждом проекте – десятки файлов. На бесплатном тарифе можно создать максимум три проекта. Чтобы создать дополнительный проект внутри команды, нажмите **New Projects**. Внутри проекта можно создать новые файлы или перетащить туда графические файлы в форматах SVG, PNG, JPE.

### *Как устроен редактор файлов Фигмы*

В новом файле пользователю доступны три области: рабочая область, панель инструментов и панель слоёв. Рассмотрим важные функции пунктов меню.

Меню открывается при нажатии на кнопку  в левом верхнем углу экрана.

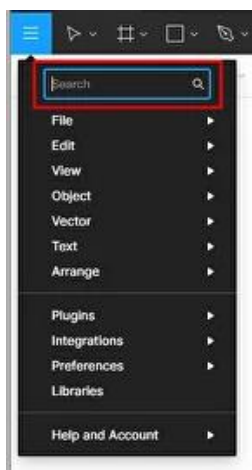


Рис. 10. Панель выпадающего меню в программе Figma

### *Поиск*

Поиск *Search* ищет пункты меню, а не элементы, которые создавал дизайнер. Данный пункт меню пригодится пользователю, если ему нужна конкретная команда, но он забыл, где именно эта команда находится.

## ***Панель File***

Инструменты панели помогают применить действие к файлу, с которым работает дизайнер. Например, с помощью *New file from Sketch* можно импортировать файлы, созданные в графическом редакторе Скетч.

Фигма – это инструмент для совместной работы, она сохраняет изменения файла автоматически. Можно сохранить изменения немедленно, нажав *Save to Version History* или посмотреть историю изменений коллег, нажав *Show Version History*.

Чтобы сохранить файл Фигмы в pdf, используйте *Export Frames to PDF*.

## ***Панель Edit***

Панель позволяет работать с элементами дизайна в файле. Чтобы применить действие к элементу, его нужно выделить. Например, объект можно скопировать как код CSS, SVG или PNG-файл с помощью *Copy As*.

Функция *Paste Over Selection* позволяет разместить скопированный элемент в левом верхнем углу другого объекта.

Все новые объекты, созданные в файле, по умолчанию серого цвета. Бывают ситуации, когда нужно создать множество объектов с другим цветом. Используйте *Set Default Properties*: скопируйте цвет нужного объекта один раз и все последующие объекты будут созданы с заданными свойствами.

Окрашивайте объект в нужный цвет пипеткой *Pick Color*.

Группа команд *Select All with* подсвечивает все похожие объекты в файле: с одинаковыми свойствами, заливкой, шрифтами, эффектами и т. д.

## ***Панель View***

Панель находится слева в меню редактора и дублируется справа. Она отвечает за масштабирование макета и навигацию по нему.

Одной из самых полезных является функция *Pixel Preview*. Фигма – это векторный редактор, поэтому дизайнер видит ровные векторные линии. Подключая просмотр объекта в пикселях, можно увидеть, как элемент будет выглядеть на экране.

Масштабировать макет можно с помощью функций *Zoom In/Zoom Out*.

Чтобы показать линейки и вытянуть направляющие в Фигме, используйте функцию *Rules*.

Чтобы расширить рабочую область и получить больше пространства для вдохновения, можно скрыть панель слоёв и интерфейс командами *Layers Panel, Show/Hide UI*.

## Панель Preferences

Панель определяет навигацию в файле. Первые три пункта: *Snap to Geometry*, *Snap to Pixel Grid*, *Snap to Objects* – настройки привязки. Они могут помочь слегка примагнитить элементы и избежать «кривой» сетки.

Функция *Highlight Layers on Hover* подсвечивает мелкие элементы дизайна при наведении, такие как кнопки, буквы, иконки, значки.

Включенная функция *Keyboard Zooms into Selection* может приближать/удалять макет относительно элемента, выбранного на экране.

## Инструменты и возможности Фигмы

Рассмотрим панель инструментов для работы с графикой и выравнивания объектов, выбора шрифтов, создания эффектов для слоёв и фигур.

### Фреймы

Фрейм или артборд – основной элемент дизайна в Фигме. Это законченный документ, который может быть страницей сайта или экраном мобильного приложения.

Можно задать размер фрейма самостоятельно или выбрать готовые размеры экранов популярных устройств (ноутбуков, часов, планшетов) в панели инструментов справа.

Фрейм объединяет объекты внутри себя. При создании дизайна внутри фрейма в панели свойств слева будут добавляться слои. Слои в Фигме – это содержимое вашего фрейма: объекты, текст, фотографии.

Можно сгруппировать отдельные объекты во фрейм или разбить его на объекты сочетаниями клавиш (см. табл. 2).

Таблица 2

### Сочетания клавиш для группировки или разбиения объектов во фрейме

Windows	MacOS	Действие
Ctrl + Alt + G	Cmd + Opt + G	Объединить объекты во фрейме
Ctrl + Shift + G	Cmd + Shift + G	Разбить фрейм на объекты

Чтобы выровнять объекты во фрейме по оси X/Y или расстояние между элементами, используйте панель справа.

Объекты внутри фрейма можно группировать. Это можно сделать из меню *Object* → *Group Selection* или сочетанием клавиш *Ctrl (Cmd) + G*. Изменения в

слоях объектов можно увидеть сразу – они объединились в группу. Теперь можно перемещать все объекты разом.

Дополнительно можно ограничивать поведение элементов во фрейме. Для этого нужно выбрать объект во фрейме и воспользоваться инструментом *Constraints* в панели свойств справа.

Фрейм – это законченный дизайн, его можно скачать одним файлом. При сохранении следует убедиться, что все объекты находятся внутри фрейма. Если случайно переместить слой объекта за пределы фрейма, можно потерять часть дизайна. Перед скачиванием проверьте превью файла, и выберите нужный формат.

### ***Модульная сетка***

Сетка в Фигме помогает упорядочить все элементы дизайна во фрейме. Чтобы легко адаптировать дизайн от одного устройства к другому, можно использовать в Фигме 12 колоночную модульную сетку (*Bootstrap*).

Для настройки сетки следует нажать «+» в блоке *Layout Grid*, задать количество колонок, их цвет и прозрачность. Можно настроить тип сетки – сделать её адаптивной, выровнять по центру или краю.

Чтобы контролировать отступы, нужно применить дополнительную сетку, например, создать одну колонку слева или справа нужной ширины, сделать неброский цвет.

### ***Векторные формы***

Инструментом *Shape Tool* можно создавать векторные объекты, отрисовывать элементы интерфейса, например, иконки.

Основные векторные объекты – это *прямоугольник, линия, треугольник, стрелка, эллипс, звезда*. Пользователь может вставлять объекты произвольной формы или удерживать клавишу **Shift** и нарисовать объект правильной формы (квадрат или окружность). Если нажать **Alt** – объект растягивается из центра.

Панель свойств объектов находится справа.

### ***Кривые***

*Pen Tool* позволяет рисовать кривые линии в Фигме и несложные векторные формы: иконки и графику. Дополнительная опция *Bend Tool* используется для скругления кривых или *Paint Bucket* для заливки закрытого контура.

## Изображения

В макет можно добавить одно или несколько изображений через панель File, инструмент Place Image или просто перетащить их с рабочего стола.

В Фигме фото вставляется как *shape*, а не как отдельный объект. По сути, изображением заливается фрейм – прямоугольник. Поэтому можно изменить заполнение фрейма параметрами Fill, Fit, Crop, Tile

С включенным параметром Fill изображение заполняет весь фрейм. При этом сложно соблюсти пропорции изображения, и картинка может обрезаться.

С включенным параметром Fit изображение отображается во фрейме полностью. Если пропорции фрейма не совпадают с пропорциями картинки, в нём появится пустое пространство.

С включенным параметром Crop можно приблизить нужный ракурс изображения, обрезав «лишние» части.

С включенным параметром Tile есть возможность создать паттерн или узор.

Чтобы добавить изображению градиент, нужно воспользоваться знаком «+» в панели свойств. Выберите стиль градиента: *Linear* (Линейный), *Radial* (Радиальный), *Angular* (Угловой) или *Diamond* (Ромбовидный).

Векторные объекты можно применять к фотографиям и использовать их как маски. Чтобы сделать маску в Фигме, создайте объект, перенесите слой ниже фотографии. Выделите оба слоя и выберите панель *Object* → *Use as mask* (горячие клавиши для Windows: **Ctrl + Alt + M**, для Mac OS: **Cmd + Opt + M**).

Можно загружать в Фигму SVG-файлы и работать с ними. SVG-файлы можно загрузить с компьютера либо переносить прямо из Adobe Illustrator или Sketch с помощью горячих клавиш **Cntrl + C** и **Cntrl + V**.

Обычно файлы такого типа вставляются как фрейм. Если необходимо, то можно перейти в слои, объединить векторные объекты в группу клавишами **Ctrl+G** и перетянуть их выше фрейма.

## Эффекты и маски

Фигма позволяет работать с заливкой фреймов и текстовых слоёв. Инструменты заливки находятся в левой панели свойств. Пользователь может выбрать цвет, прозрачность, указать конкретные значения цвета.

В Фигме шесть типов заливки: сплошной цвет (*Solid*), линейный градиент (*Linear*), радиальный градиент (*Radial*), угловой градиент (*Angular*), радиальный с четырьмя лучами (*Diamond*), изображение (*Image*). По умолчанию для объекта выбран режим ровной заливки *Solid*.

При переключении на *Linear* появится градиент с осью, вдоль которой изменяется цвет. По умолчанию у него есть две точки, одна из которых прозрачная. Добавьте цвета ползунком или укажите значение цвета в поле Hex-кода.

У линейного градиента есть третья неприметная ручка. Если зажать **Alt (Opt)** и потянуть за неё, цветовой переход отрисовывается перпендикулярно ей.

Есть возможность поменять расположение крайних точек градиента. Двигая ползунки, пользователь может настроить угол поворота градиента и скорость перехода от одного цвета к другому.

Если необходимо добавить несколько цветов в градиент, то кликните по шкале над палитрой. Чтобы удалить цвет – нажмите **Delete**.

Добавьте объектам заливки с разными градиентами и сочетайте фигуры друг с другом. Можно использовать *Layer* – режим наложения, накладывать объекты друг на друга для интересного решения.

### **Текст**

По умолчанию пользователю доступна библиотека шрифтов *Google Fonts*. Если нужно загрузить свои шрифты в Фигму, то необходимо скачать десктопную версию Фигмы или установить *Font Installers* (Windows/Mac OS), чтобы загрузить их в браузер.

Инструменты для работы с текстовым слоем стандартные: *начертание, размер, выравнивание текста, высота строки, отступ между параграфами и красная строка*.

В меню *Advanced Type* есть три типа изменения размера текстового блока: 1) *Width*, где ширина подстраивается под контент, 2) *Fixed*, чтобы задать блок фиксированной ширины, 3) *Height*, чтобы подстроить высоту текстового блока под его контент.

Пользователю лучше создавать отдельные текстовые блоки для заголовка и основного текста, поскольку так их удобнее редактировать.

В Фигме можно перевести шрифт в кривые и создать необычный объект вашего дизайна. Нужно выбрать текстовый слой, пункт *Flatten* и дважды кликнуть на текст.

Изменённый шрифт можно сохранить как SVG-файл и вставить в дизайн.

### **Компоненты**

Компоненты в Фигме помогают применять изменения к группе элементов. Это экономит время дизайнера при изменениях макета. Чтобы превратить объект или группу объектов в компонент, выделите их и нажмите *Create Component (Ctrl+Alt+K)*.

Компоненты можно увидеть по цвету слоя – Фигма помечает их фиолетовым. Кроме цвета родительский компонент помечается иконкой с 4 ромбами, а

дочерние компоненты – иконкой с одним ромбом. Все изменения родительского компонента будут отражаться на дочерних компонентах.

Пользователь может менять настройки дочернего компонента, делая его относительно самостоятельным. Например, можно вручную задать компоненту размер, цвет, обводку, отличные от родительского компонента.

При этом связь между компонентами не потеряется. Можно внести новый объект в родительский компонент, и он отобразится в дочерних компонентах, даже измененных ранее.

Чтобы восстановить нарушенную связь между родительским и дочерним компонентом, следует выделить дочерний компонент, нажать иконку возврата действия и сбросить параметры, которые были изменены вручную.

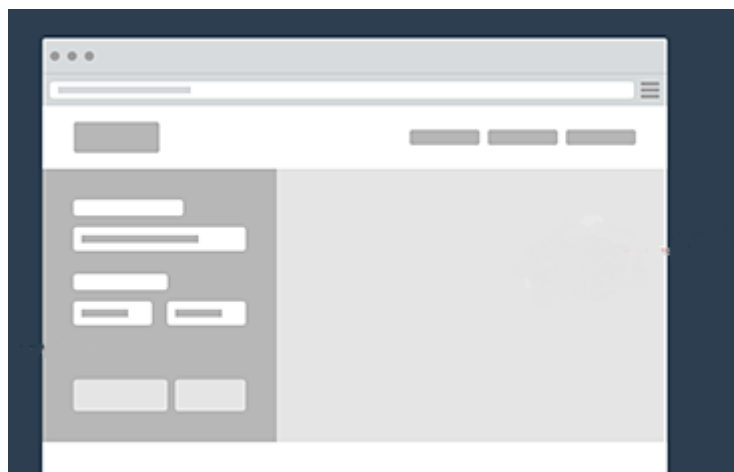
Для работы с компонентами Фигма разработала функцию *Auto Layout*. Если скопировать кнопку с функцией *Auto Layout*, копии также будут динамически изменяться в зависимости от контента.

## Создание сайта

На сегодняшний день существуют три основных метода создания сайтов. А именно: написание сайтов на языке гипертекстовой разметки HTML, создание сайтов при помощи CMS-движка, а также через онлайн конструктор. Эти способы создания сайта рассмотрим в последующих главах.

### Задания для самостоятельной работы к главе 1

1. Создайте в Figma проект (wireframe) по образцу (см. рисунок ниже).



2. Разработайте набросок макет-проекта для сайта произвольной тематики. Для отрисовки наброска используйте бумагу и ручку (пример см. на рис 2).
3. Создайте по своему наброску проект в программе Figma.

## ГЛАВА 2. СОЗДАНИЕ САЙТА С ПОМОЩЬЮ HTML

### Теоретические сведения об HTML

HTML-документ представляет собой текстовый файл с расширением **.htm** или **.html**. Такой документ может быть создан в простейшем текстовом редакторе *Блокнот*.

#### *Структура HTML-документа. Цвета и специальные символы*

Для форматирования текста, задания структуры документа, встраивания ссылок, графики и объектов в HTML-документах используются специальные кодовые слова, которые называются *дескрипторами разметки* или *тэгами* (tags). Тэги можно вводить как строчными, так и прописными буквами, но обязательно на английском языке.

#### Тэги HTML

Тэг HTML состоит из следующих друг за другом в определенном порядке элементов:

- левой угловой скобки <;
- необязательного слэша /, который означает, что тэг является *конечным тэгом, закрывающим некоторую структуру*. Таким образом, в этом контексте Вы можете читать символ / как *конец*;
- имени тэга, например TITLE или PRE;
- обязательных или необязательных, если даже тэг может иметь их, атрибутов. Тэг может быть без атрибутов или сопровождаться одним или несколькими атрибутами, например: ALIGN=CENTER. Атрибуты содержатся только в открывающем тэге;
- правой угловой скобки >.

**Примеры тэгов:** <head>, </body>, <HR>, <H1 ALIGN=LEFT>

#### Элементы HTML

Большинство, но не все тэги HTML, спарены так, что за открывающим тэгом следует соответствующий закрывающий тэг, а между ними содержатся текст или другие тэги, например:

```
<H1>Foreword</H1>,
<title>Заголовок окна</title>.
```

Такие тэги называются *парными*. Различают также *непарные* тэги, т.е. такие, для которых закрывающий тэг необязателен. Например, <HR>.



## Атрибуты

Атрибуты могут располагаться лишь в открывающем тэге! Для каждого тэга определяется множество возможных атрибутов. Большинство тэгов допускает один или несколько атрибутов, однако их может и совсем не быть. Атрибут состоит из расположенных в следующем порядке элементов:

- имени атрибута, например, WIDTH;
- знака равенства (=);
- значения атрибута, которое задается строкой символов, например, "80".

## Структура HTML – документа

Web-страница начинается с тэга <HTML> и заканчивается тэгом </HTML>.

HTML-документ состоит из двух частей: *заголовка* и *тела*.

```
<HTML>
<HEAD>
Содержание заголовка
</HEAD>
<BODY>
Содержание тела документа
</BODY>
</HTML>
```

Рассмотрим каждый раздел отдельно.

## Раздел HEAD

Может содержать следующие элементы:

<TITLE> Задает имя HTML- документа, это имя отображается в заголовке окна обозревателя;

<STYLE> Служит для локального описания каскадных листов стилей;

<SCRIPT> Используется для добавления сценариев, например, JavaScript.

## Тэг Style

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

Синтаксис тэга Style:

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

### Раздел <BODY> или тело страницы

В теле HTML-документа располагается содержимое самой Web-страницы. Эта главная часть документа находится между дескрипторами <BODY> и </BODY>. Там размещаются текст, изображения, таблицы, гиперссылки, аудио- или видео-объекты, и т.д.

### Атрибуты элемента <BODY>

В тэге <BODY> можно установить параметры страницы – атрибуты дескриптора <BODY>. Примеры атрибутов приводятся в табл. 3. В табл. 4 приводятся стандартные цвета, используемые для форматирования текста и других элементов страницы. Цвета можно задавать как с помощью текстового названия – например, *green*, так и с помощью шестнадцатеричного обозначения – например, *#0FF0*.

Таблица 3

### Атрибуты тэга Body и их применение

Атрибут	Пример применения
<b>bgcolor</b> =цвет фона	<BODY bgcolor=green>
<b>text</b> =цвет текста	<BODY text=green>
<b>background</b> =”имя фоновой иллюстрации”	<BODY background=”pic.gif”> (если картинка лежит в одном каталоге со страницей)
<b>(bgproperties=”fixed”)</b>	<BODY background=”Pictures/pic.gif”> (.././pic.gif – если на 2 уровня вверх)
<b>link</b> =цвет гиперссылки	<BODY link=#EE82EE>
<b>alink</b> =цвет активной гиперссылки	<BODY alink=#FF6347>
<b>vlink</b> =цвет просмотренной гиперссылки	<BODY vlink=#DDA0DD>

Стандартная таблица цветов

Имя цвета	Цвет			Русское название
	красный	зеленый	синий	
Aqua	#0	ff	ff	Бирюзовый
Black	#0	0	0	Чёрный
Blue	#0	0	ff	Синий
Fuchsia	#ff	0	ff	Розовый
Gray	#80	80	80	Серый
Green	#0	80	0	Зелёный
Lime	#0	ff	0	Светло-Зелёный
Maroon	#80	0	0	Коричневый
Navy	#0	0	80	Ультрамариновый
Olive	#80	80	0	Оливковый
Purple	#80	0	80	пурпурный
Red	#ff	0	0	красный
Silver	#c0	c0	c0	серебряный
Teal	#0	80	80	темно-зелёный
Yellow	#ff	ff	0	жёлтый
White	#ff	ff	ff	белый

### *Форматирование текста. Заголовки.*

Практически любой документ начинается с заголовка или названия статьи. В HTML заголовки формируются специальными тэгами:

`<Hn>...</Hn>`, где n – число от 1 до 6, заголовки располагаются в порядке уменьшения шрифта. Между тэгами располагается текст, который нужно отформатировать.

#### **Примеры:**

`<h6>` Заголовок шестого уровня – самый маленький! `</h6>`;

`<h4>` Заголовок четвертого уровня – на уровне обычного текста! `</h4>`;

`<h1>` Заголовок первого уровня – самый большой! `</h1>`.

### **Создание нового абзаца**

Для этой цели применяется тэг `<P>`, закрывающий тэг не используется, так как открывающий `<P>` для нового абзаца, для предыдущего абзаца означает конец абзаца. **НО** если в `<P>` используется атрибут **Align**, то его закрывать **обязательно**.

## Атрибуты:

Выравнивание текста	<P>Текст
<b>Align="left"</b> по левому краю	<P ALIGN=LEFT>Текст</P>
<b>Align="right"</b> по правому краю	<P ALIGN=right>Текст</P>
<b>Align="center"</b> по центру	<P ALIGN=center>Текст</P>
<b>Align="justify"</b> по ширине	<P ALIGN=justify>Текст</P>

Для создания новой строки в абзаце используется тэг <BR>. Закрывающий тэг отсутствует.

## Изменение начертания текста

К одному и тому же тексту может применяться сразу несколько разных элементов. Но следует помнить, что закрывающие тэги должны располагаться в обратном порядке. Основные тэги для изменения начертания текста приведены в табл. 5.

Таблица 5

### Тэги, применяемые для изменения начертания текста

Элемент	Значение
<I>.....</I>	Курсив (Italic)
<B>...</B>	Выделение(BOLD)
<U>...</U>	Подчеркивание
<S>...</S>	Перечеркнутый текст
<STRONG>...</STRONG >	Сильное выделение текста
<SMALL>...</SMALL>	Уменьшенный шрифт
<SUB>...</SUB>	Подстрочные символы (нижний индекс)
<SUP>...</SUP>	Надстрочные символы (верхний индекс)

## Примеры:

<i>текст выделен курсивом</i>;

<u><b><sub>подчеркнутый полужирный нижний индекс </sub></b> </u>.

## Изменение размера, цвета и шрифта текста

Элемент <FONT> используется для изменения шрифта, цвета и размера текста. Атрибуты тэга Font с примерами их использования приведены в табл.6:

### Атрибуты тэга Font и их применение

<b>SIZE</b> задает размер шрифта	<code>&lt;font size=n&gt;...&lt;/font&gt;</code> , n=1..6, по возрастанию	<code>&lt;font size=5&gt; Текст &lt;/font&gt;</code>
<b>FACE</b> задает стандартное имя используемого шрифта. Можно ввести несколько шрифтов через запятую	<code>&lt;font face="имя шрифта"&gt;...&lt;/font&gt;</code> (по умолчанию – Times New Roman)	<code>&lt;font face="Courier, Tahoma"&gt;Шрифт Courier или Tahoma &lt;/font&gt;</code>
<b>COLOR</b> задает цвет текста	<code>&lt;font color=red&gt;...&lt;/font&gt;</code>	<code>&lt;font color=blue&gt;этот текст голубого цвета &lt;/font&gt;</code>

В качестве стандартных шрифтов можно использовать шрифты Microsoft Office.

### Тэг div

Элемент `<div>` является блочным элементом и предназначен для выделения фрагмента документа с целью изменения вида содержимого.

Как правило, вид блока управляется с помощью стилей. Чтобы не описывать каждый раз стиль внутри тэга, можно выделить стиль во внешнюю таблицу стилей, а для тэга добавить атрибут `class` или `id` с именем селектора.

Как и при использовании других блочных элементов, содержимое тэга `<div>` всегда начинается с новой строки. После него также добавляется перенос строки. Синтаксис тэга: `<div>...</div>`.

Атрибуты тэга `div`:

- `align` – задает выравнивание содержимого тега `<div>`.
- `title` – добавляет всплывающую подсказку к содержимому.

### Выделение фона текста цветом

Для этого используется парный тэг `<SPAN>`:

`<SPAN STYLE="BACKGROUND-COLOR:GREEN">ТЕКСТ</SPAN>`

### Горизонтальные линии

Тэг `<HR>` вставляет в HTML-документ горизонтальную линию, цветом, длиной и толщиной которой можно легко управлять. Тэг имеет следующие параметры:

- **Width** определяет длину линии в пикселах или процентах от ширины окна обозревателя;

- **Size** определяет толщину линии в пикселах;
- **Align** определяет выравнивание линии, может принимать значения: **left** – по левой стороне, **right** – по правой стороне, **center** – по центру документа;
- **Noshade** определяет отказ от трехмерности (объемности) линии;
- **Color** определяет цвет линии.

#### Примеры:

`<hr width="80%" size=20 Noshade>` – линия размером 80% и толщиной 20 пикселей;

`<hr width="100" align="center" color=blue>` – линия размером 100 пикселей, расположенная по центру, синего цвета.

#### *Создание гиперссылок. Вставка обычной гиперссылки*

Гиперссылки языка HTML, или просто ссылки, могут указывать на другую Web-страницу, или сайт, на изображение, на любой файл, на адрес электронной почты.

Обычная конструкция записи ссылки выглядит следующим образом:

`<A HREF="Имя и адрес страницы" Target="Окно" Title="Подсказка"> Название ссылки </A>`

Параметру **HREF** открывающего тэга `<A>` присваивается адрес любого файла в Интернет. Этот адрес может быть абсолютным, т.е. указанным полностью, например `http://lenininc.narod.ru/index.html`, или указанным относительно текущей позиции (*index.html*).

Параметр **Target** определяет, в каком окне (фрейме) следует загрузить указанный документ. Этот параметр может иметь следующие значения:

– **\_top** – загрузить содержимое файла, заданного ссылкой, на всем пространстве окна браузера (если до этого существовало разбиение на фреймы, оно исчезнет);

– **\_blank** – загрузить содержимое файла, заданного ссылкой, в новом окне;

– **\_self** – загрузить содержимое файла в то окно или фрейм, которое включает данную ссылку;

– **\_parent** – загрузить содержимое файла в «родительский» фрейм.

Параметр **Title** содержит текст подсказки, который будет появляться при наведении пользователем курсора мыши на ссылку.

#### Пример:

`<a href="search.html" target="_self" title="пример ссылки"> Открыть </a>`  
(документ **search.html** должен быть создан заранее).

Рассмотрим создание гиперссылки на другую страницу. Тэг записывается следующим образом:

```
<A HREF="Имя и адрес страницы" > Текст ссылки </A>
```

### **Примеры:**

```
<a href="personal.htm">Перейти к моей личной странице</a>;
```

```
<a href="C:\web\index.htm">Главная</a>.
```

Ссылкой также может быть сделан любой графический файл. Для этого вместо названия требуемой гиперссылки в документ следует поместить изображение, воспользовавшись стандартным тэгом его вставки. В этом случае параметр **Title** должен быть определен только у рисунка.

```
<a href="first.html" target="_blank"></a>
```

### **Создание закладки (ссылки на определенное место в документе)**

В HTML-документе можно расставить в требуемых местах закладки с обязательно разными именами, а в ссылках указывать на них. Теперь, щелкнув по подобной ссылке левой кнопкой мыши, пользователь попадет в то место документа, где данная закладка была определена.

```
<a Name="Имя_закладки"></a> – определение закладки;
```

```
<a href="#Имя_закладки">название раздела</a> – ссылка на закладку.
```

### **Ссылки на страницы собственного сайта**

Большинство ссылок, которые создаются, ведут на страницы одного и того же сайта: с главной – на страницы разделов, с них – на страницы с контентом и так далее. В этом случае нам понадобятся относительные ссылки, т.е. те, которые обращаются к какой-либо странице на том же сервере.

Если браузер не находит в ссылке символов «http://» , он выполняет поиск указанного документа на том же сервере. Путь к файлу – обозначение, призванное указать на определенный файл, он сообщает браузеру, где искать этот файл. Формат веб-путей к файлам основывается на конвенции Unix, согласно которой каталоги и имена файлов отделяются друг от друга слешем (/). Относительный путь к файлу описывает, как добраться по ссылке до нужного файла, начиная с позиции текущего документа.

Рассмотрим создание ссылок на примере сайта с книгами со следующей структурой:

- главная страница (index.html);
- страница с информацией о сайте (about.html);
- страница с контактной информацией (contacts.html);

- список книг по разным предметам (файлы `math.html`, `history.html`, `biology.html` в каталоге `books`);
- изображения, используемые на страницах (логотип `logo.png` и фон страниц `background.png` в каталоге `images`).

Если каталоги сайта представить в виде схемы, то в конечном счете она будет выглядеть как перевернутое дерево с корневым каталогом в вершине иерархии (рис. 11).

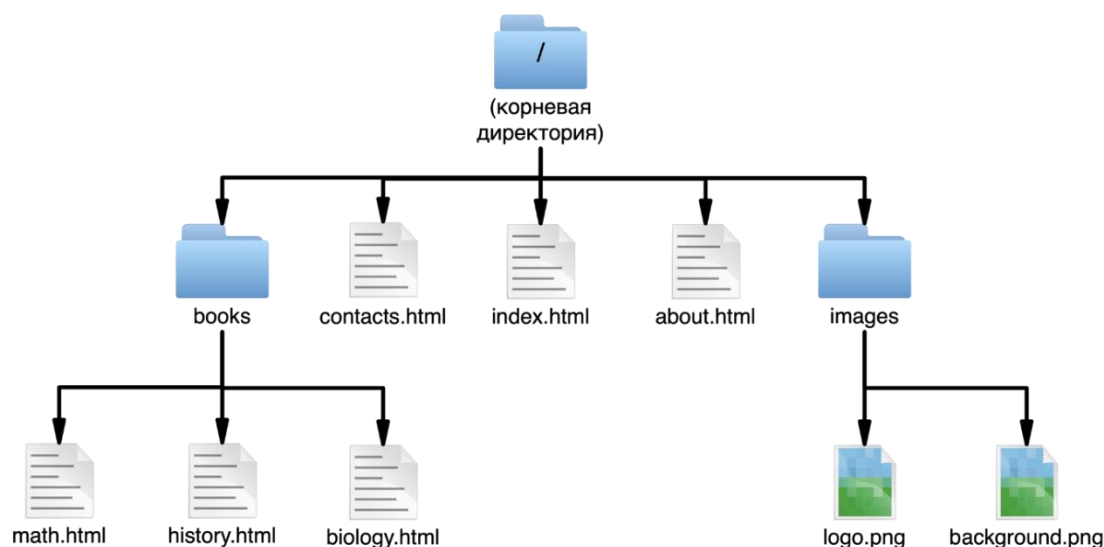


Рис. 11. Каталоги сайта в виде иерархической структуры

### Создание ссылок в пределах одного каталога

Наиболее простым относительным URL-адресом является ссылка на другой файл в пределах одного каталога. При создании ссылки на файл в том же каталоге нужно указать только имя файла (с расширением). Когда URL-адрес представляет собой лишь имя файла, сервер ищет его в текущем каталоге (т. е. в том, где находится HTML-документ со ссылкой).

В этом примере сделаем ссылку с главной страницы (`index.html`) на страницу с информацией о сайте (`about.html`). Оба файла находятся в одном корневом каталоге. Таким образом, нам достаточно просто указать имя файла (рис. 12):



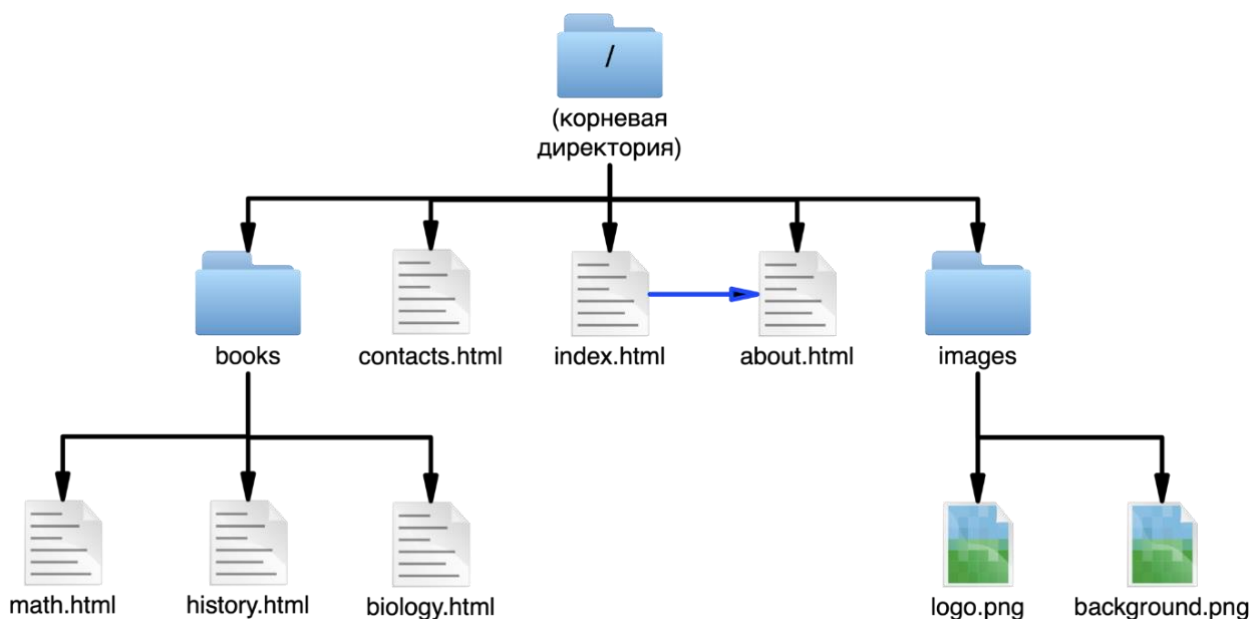


Рис. 12. Ссылки на страницы одного каталога

Для создания ссылки из файла `index.html` на файл `about.html` нам потребуется следующий код:

```
<a href="about.html">О проекте</a>
```

Сервер производит поиск файла в том же каталоге, что и документ, с которого осуществляется переход.

### Создание ссылки на файл во вложенном каталоге

А как быть, если файлы находятся в разных каталогах? Нужно указать браузеру направление поиска путем включения в URL-адрес полного пути к файлу. Давайте рассмотрим, как это сделать.

Возвращаясь к приведенному выше примеру, файлы с книгами лежат в подкаталоге с именем `books`. Тогда в файле `index.html` нам необходимо создать ссылку на файл `math.html` в каталоге `books`. URL-путь сообщает браузеру, что в текущем каталоге нужно найти каталог с именем `books`, а затем перейти к файлу `math.html` (рис. 13).

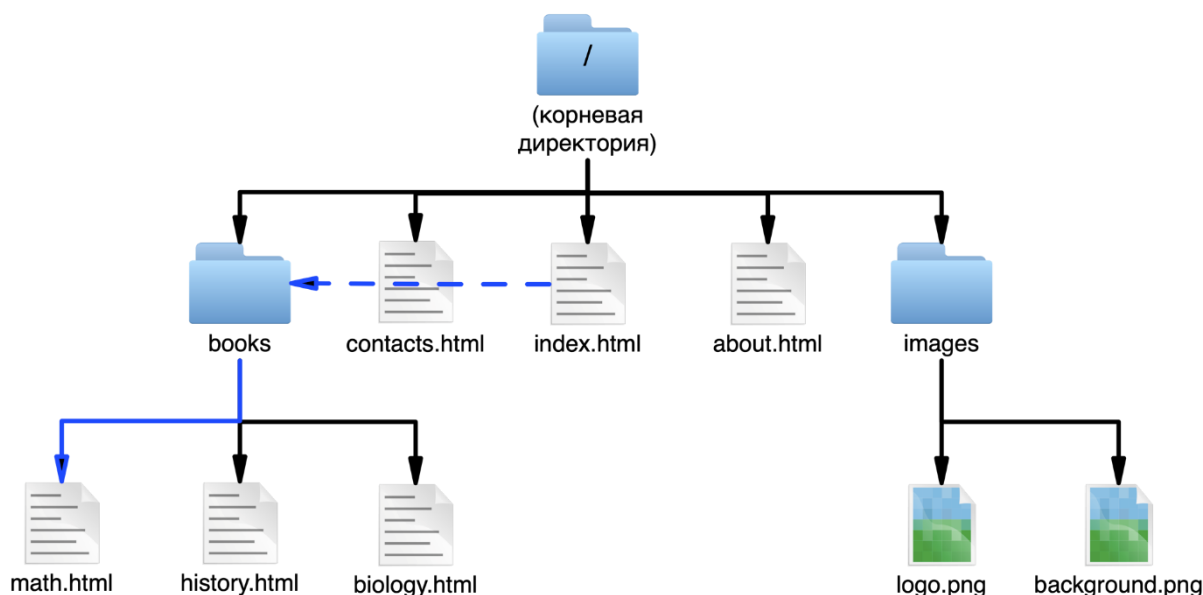


Рис. 13. Ссылка на страницу во вложенном каталоге

Для создания ссылки из файла `index.html` на файл `math.html` нам потребуется следующий код:

```
<a href="books/math.html">Математика</a>
```

Сервер ищет каталог `books` в каталоге с текущим документом.

### Создание ссылки на вышестоящий каталог

На этот раз нам нужно выбрать противоположное направление и создать ссылку, ведущую со страницы с книгами по математике (`math.html`) обратно на главную страницу, расположенную на уровень выше.

Специально для этой цели в Unix существует условное обозначение «точка-точка-слеш» (`../`). Если путь к файлу начинается с символов `../`, это равносильно тому, что сказать браузеру: «вернись на уровень выше и следуй по пути к указанному файлу». Если сравнить это действие с поиском файлов на рабочем компьютере, то следует отметить, что символы `../` действуют так же, как щелчок мышью по кнопке «Назад» в файловом менеджере Проводник (Windows Explorer).

Поскольку файл `math.html` находится в подкаталоге `books`, то для перехода к файлу `index.html` нам необходимо вернуться на один уровень назад к корневому каталогу. Этот путь указывает браузеру на необходимость подняться на один уровень и затем найти файл `index.html` (рис. 14):

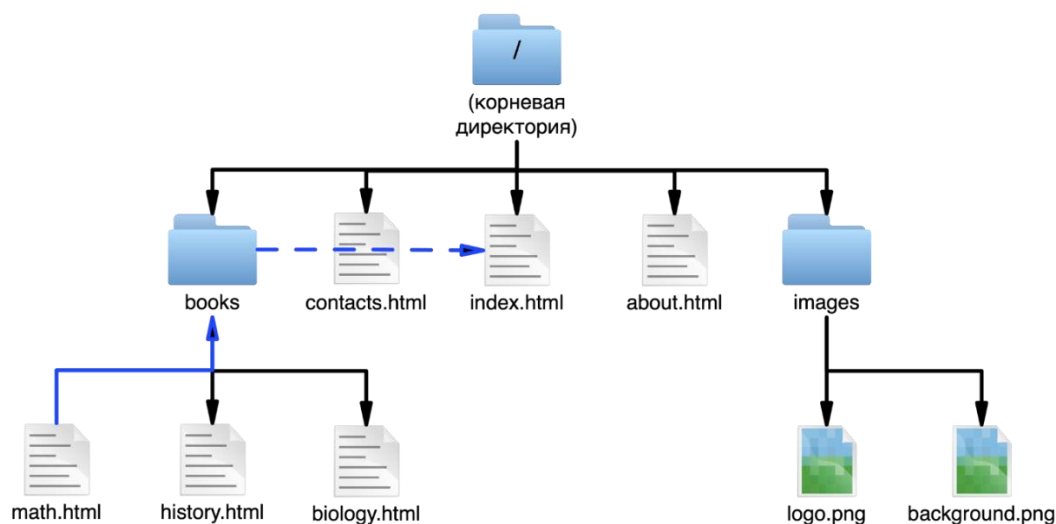


Рис. 14. Ссылка на страницу в вышестоящем каталоге

Для создания ссылки из файла `math.html` на файл `index.html` нам потребуется следующий код:

```
<a href="../../index.html">Главная страница</a>
```

### Пути к файлам относительно корня сайта

Как мы уже знаем, любой сайт имеет корневой каталог, в котором содержатся все остальные каталоги и файлы сайта. Все пути к файлам, которые мы рассматривали до этого момента, строились относительно документа со ссылкой. Существует альтернативный способ записи пути: начиная с корневого, следует перечислять имена каталогов до тех пор, пока не будет достигнут файл, на который должен осуществляться переход по ссылке. Такой путь называется *путем относительно корня сайта*.

Согласно принятому в Unix обозначению слеш (`/`) в начале пути к файлу вызывает обращение к корневому каталогу. Пример создания ссылки из файла `math.html` на главную страницу (`index.html`) можно упростить благодаря ссылке относительно корня сайта.

```
<a href="/index.html">Главная страница</a>
```

В указанной ниже ссылке путь относительно корня сайта передает следующую команду: «Зайди в самый верхний каталог этого сайта, открой каталог найди в нем файл `index.html`».

В целом, ссылки относительно корня сайта более предпочтительны по причине их большей гибкости.

Поскольку такая ссылка начинается с корневого каталога, она будет работать в любом документе независимо от того, где именно на сервере он находится. Ссылки относительно корня могут быть полезными в случае, если содержимое

сайта предполагается перемещать из каталога в каталог, а также в случае создания ссылок на динамические материалы. Кроме того, они облегчают процесс копирования и вставки ссылок из одного документа в другой.

Однако есть и обратная сторона медали: такие ссылки не будут срабатывать на локальном компьютере, поскольку обращаются к корню логического диска. Чтобы проверить работоспособность ссылок, необходимо выполнить загрузку сайта на удаленный сервер.

### ***Вставка изображения.***

Непарный тэг **<IMG>** позволяет вставить изображение в документ.

#### **Атрибуты тэга IMG**

**SRC** – имя вставляемого графического файла с расширением. В HTML-документ можно вставить изображения с расширениями **.gif, .jpg, .png (обязательный параметр)**;

**ALIGN** выравнивание изображения в документе;

**ALIGN=right, ALIGN=left** выравнивание изображения по правому или по левому краю (с обтеканием текстом);

**ALIGN=top** выравнивание текста по верхней кромке изображения;

**ALIGN=bottom** выравнивание текста по нижней кромке изображения;

**ALIGN=middle** выравнивание текста по центру изображения (без обтекания текстом);

**BORDER** – рамка в пикселах вокруг изображения, по умолчанию – 0;

**WIDTH** – ширина изображения в пикселах;

**HEIGHT** – высота изображения в пикселах;

**HSPACE** – горизонтальный отступ в пикселах;

**VSPACE** – вертикальный отступ в пикселах;

**ALT** – сообщение, которое выводится вместо изображения, когда оно не отображается. Также выдается в виде подсказки, при наведении курсора мыши;

**NAME** – имя изображения.

#### **Примеры:**

```
;
```

```
.
```

### ***Создание списков***

#### **Нумерованный список – элемент <OL>**

Нумерованный список начинается с **<OL>** и заканчивается **</OL>**. Между этими тэгами располагаются элементы списка, каждый из которых начинается с тэга **<LI>**.

```

<ol>
<li>первый элемент нумерованного списка
...
<li> ... элемент нумерованного списка
</ol>

```

<OL> может содержать атрибуты **TYPE** для задания типа используемой нумерации, **START** – для задания начального значения, с которого будет начинаться отчет:

<b>TYPE= A</b>	большие латинские буквы
<b>TYPE= a</b>	маленькие латинские буквы
<b>TYPE= 1</b>	арабские цифры
<b>TYPE= I</b>	большие римские цифры
<b>TYPE= i</b>	маленькие римские цифры

### Примеры:

```
<ol type=a start=2>...</ol>
```

```
<ol type=I >...</ol>
```

### Маркированный список – элемент <UL>

Маркированный список начинается с <UL> и заканчивается </UL>. Между этими тэгами располагаются элементы списка, каждый из которых начинается с тэга <LI>.

```

<ul>
<li>первый элемент маркированного списка
...
<li> ... элемент маркированного списка
</ul>

```

<UL> может содержать атрибуты **TYPE** для задания типа используемого маркера:

<b>TYPE= “disc”</b>	заштрихованный кружок
<b>TYPE= “circle”</b>	незаштрихованный кружок
<b>TYPE= “square”</b>	заштрихованный квадрат

### Примеры:

```
<ul type="disq" >...</ul>
```

```
<ul type="square">...</ul>
```

### Списки определений

Список определений начинается с `<DL>` и заканчивается `</DL>`. Между этими тэгами располагается название термина после тэга `<DT>`, а затем – определения, каждый из которых начинается с тэга `<DD>`.

```
<DL>  
  <DT>Термин  
  <DD>Определение 1  
  <DD>Определение 2  
</DL>
```

### Создание таблиц

Используя таблицы, можно создавать в HTML-документе такие эффекты, как размещение текста в нескольких колонках, состыковку картинки и фона, и др.

Таблица начинается открывающим тэгом `<table>` и завершается закрывающим тэгом `</table>`. Тэг `<table>` может иметь следующие параметры:

**WIDTH** – ширина таблицы в пикселях или процентах. По умолчанию ширина таблицы определяется содержимым ячеек;

**BORDER** – толщина рамки таблицы. По умолчанию таблица рисуется без рамки, т.е. `border = 0`;

**BORDERCOLOR** – цвет рамки;

**BGCOLOR** – цвет фона для всей таблицы;

**BACKGROUND** – графический фон для всей таблицы;

**CELLSPACING** – расстояние между рамками ячеек таблицы в пикселях;

**CELLPADDING** – расстояние в пикселях между рамкой ячейки и текстом;

**ALIGN** – определение расположения таблицы в документе. По умолчанию таблица прижата к левому краю страницы. Допустимые значения: **left**, **right**, **Center**;

**FRAME** – управление выводом внешней рамка таблицы. Допустимые значения: **void** – рамки нет (значение по умолчанию), **above** – только линия сверху; **below** – только линия снизу, **hsides** – линии сверху и снизу, **vsides** – только линии слева и справа, **lhs** – только линия слева, **rhs** – только линия справа, **box** – полная рамка, **border** – также полная рамка;

**RULES** – управление отображением линий, разделяющих ячейки таблицы. Допустимые значения: **none** – нет разделяющих линий (значение по умолчанию),

**groups** – линии только между группами рядов, **rows** – только между строками, **cols** – Только между столбцами, **all** – между всеми строками и столбцами.

Таблица может включать заголовок, который располагается между тэгами `<caption>...</caption>`. Этот тэг должен следовать непосредственно после тэга `<table>`. Тэг `<caption>` может содержать параметр **align**, который определяет его положение относительно таблицы: **top** – заголовок над таблицей по центру (по умолчанию), **left** – над таблицей слева, **right** – над таблицей справа, **bottom** – под таблицей по центру.

Теперь обратим внимание на строки и отдельные ячейки таблицы. Каждая строка таблицы начинается тэгом `<TR>` и завершается `</TR>`, а каждая ячейка таблицы начинается с тэга `<TD>` и завершается тэгом `</TD>`.

Данные тэги могут иметь следующие параметры:

**Align** – горизонтальное выравнивание текста в ячейках строки. Возможны следующие значения: **left** – выравнивание влево, **center** – выравнивание по центру, **right** – выравнивание вправо;

**Valign** – вертикальное выравнивание текста в ячейках строки. Возможны следующие значения: **top** – по верхнему краю, **center** – по центру, **bottom** – по нижнему краю;

**Bgcolor** – цвет фона строки или ячейки;

**Background** – графический фон строки или ячейки.

**Атрибуты, применяемые только для <TD>:**

**Width** – ширина ячейки в пикселях;

**Height** – высота ячейки в пикселях;

**Colspan** – растягивание (объединение) ячеек по горизонтали, например, если мы зададим атрибут как `<td colspan=2>`, то это будет означать, что ячейка растянется на два столбца таблицы;

**Rowspan** – растягивание (объединение) ячеек по строке, `<td rowspan=2>`, означает, что ячейка будет растянута на два строки таблицы;

**Nowrap** – присутствие этого параметра показывает, что текст должен размещаться в одну строку.

Если ячейка пуста, то рамка вокруг нее не отображается. Если рамка все же требуется вокруг пустой ячейки, то в эту ячейку надо ввести символьный объект «&nbsp;» (неразрывный пробел). Также следует иметь в виду, что тэги, задающие начертание шрифта – `<b>`, `<i>`, `<font color= ... size=...>`, необходимо повторять в каждой ячейке.

Рассмотрим пример – табл. 7.

## «Расписание»

День	Уроки		
Пн.	Матем.	Рус. яз.	География
Вт.	Физика	Химия	Матем.
Ср.	Ин.яз.	Литература	Труд

Параметры таблицы будут следующими. Шрифт – Garamond, цвет уроков во втором, третьем и четвертом столбцах – зеленый, остального текста – красный, фон во всей таблице – светло-голубой. Приведем код HTML для создания подобной таблицы:

```
<font face=Garamond>
<table border=5 bordercolor=pink width=80% bgcolor=aqua cellspacing=5 cellpadding=10>
<caption align=left>Расписание</caption>
<tr>
<td><font color=red>день<td colspan=3><font color=red>уроки
</tr>
<tr><td><font color=red>понед.
<td><font color=green>матем.
<td><font color=green>рус.яз.
<td><font color=green>география
</tr>
</tr>
<tr><td><font color=red>Вт.
<td><font color=green>физика.
<td><font color=green>химия
<td><font color=green>математика
</tr>
</tr>
<tr><td><font color=red>среда.
<td><font color=green>ин.яз.
<td><font color=green>литература
<td><font color=green>труд
</tr>
</tr>
</table>
</font>
```



## Фреймы

С помощью фреймов можно разделить окно браузера на части. В результате пользователь сможет изучать каждую часть страницы независимо от остальных ее частей. Браузер, поддерживающий фреймы, загружает разные элементы страницы в разные секции или фреймы своего окна.

Рассмотрим строение фреймов на примере. Файл `index.html` содержит следующий код:

```
<html>
<head>
<title></title>
</head>
<frameset cols="20%,75%">
<frame src="menu.htm" name="main">
<frame src="base.htm" name="bases">
</frameset>
<noframe>Установите себе браузер, который поддерживает
фреймы!</noframe>
</html>
```

В результате мы получим окно браузера, разделенное на два окна. Левое окно занимает 20% окна браузера и содержит страницу с названием `menu.htm`. Правое окно займет 75% окна браузера и в нем будет отображаться страница `base.htm`.

Между дескрипторами `<noframe>` и `</noframe>` находится текст, который будет отображаться в случае, если браузер пользователя не поддерживает фреймы.

При создании файла `menu.htm` в тэге `<head>` необходимо определить базовый элемент:

```
<head>
<base target="bases">
...</head>
```

Это означает, что ссылки, находящиеся в этом документе, должны открываться в окне, которое имеет имя `bases`.

Подробнее рассмотрим тэги `<FRAMESET>` и `<FRAME>`.

В тэге `<FRAMESET>` задается количество строк или столбцов (`ROWS` и `COLS` соответственно), а также их размеры и расположение.

Существует три способа задания размера строк и столбцов

1. **Задание пикселями.** Просто укажите высоту или ширину элемента в пикселях.

2. **Задание процентами.** Укажите, сколько процентов от всего размера окна обозревателя следует отдать данному фрейму. После цифр обязательно ставится знак %. Необходимо, чтобы все указанные процентные значения в сумме составляли 100.

3. **Использование символа \*.** С помощью символа \* элементу выделяется оставшаяся часть в окне. Например, «20%, \*» и «20%, 80%» – это одно и то же.

Тэг <FRAMESET> имеет следующий синтаксис:

```
<FRAMESET COLS="величины" ROWS="величины">
```

В тэге <FRAME> задаются параметры для каждого тэга в отдельности. Параметр **SRC** задает имя файла, который загрузится в данном фрейме. Параметр **NAME** задает имя данного фрейма. Это имя необходимо для того, чтобы впоследствии указать, к какому именно фрейму относится ссылка. Существует также параметр **SCROLLING**, позволяющий задать наличие у фрейма полос прокрутки. Может иметь значения YES, NO, AUTO. **NORESIZE** позволяет создавать фреймы, размеры которых в дальнейшем изменить невозможно.

```
<FRAME SRC="адрес" NAME="имя фрейма" SCROLLING="YES|NO|AUTO" NORESIZE>
```

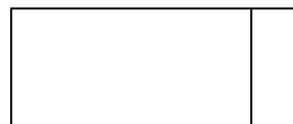
### Примеры:

1) <frameset cols="\*,140">

```
<frame src="first.htm" name="main">
```

```
<frame src="list.htm" name="list">
```

```
</frameset>
```

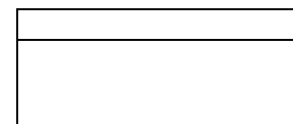


2) <frameset rows="100,\*">

```
<frame src="menu.htm" name="Frame1">
```

```
<frame src="base.htm" name="Frame2">
```

```
</frameset>
```

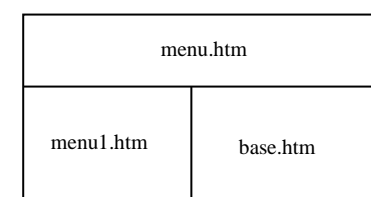


3) <frameset rows="20%,75%">

```
<frame src="menu.htm" name="Frame1">
```

```
<frameset cols="45%,55%">
```

```
<frame src="menu1.htm" name="Frame2">
```



```
<frame src="base.htm" name=" Frame3">
</frameset>
</frameset>
```

```
4) <frameset cols="30%,*">
<frame src="menu.htm" name="Frame1">
<frameset rows="20%,20%,*">
<frame src="page1.htm" name="Frame2">
<frame src="page2.htm" name="Frame3">
<frame src="page3.htm" name="Frame4">
</frameset>
</frameset>
```



#### Фреймы применяются в следующих случаях:

- 1) если необходимо, чтобы какая-то часть страницы все время находилась на экране (логотип, панель навигации);
- 2) если это необходимо для дизайнерских целей.

#### *Использование каскадных листов стилей*

Технология таблиц каскадных стилей – это набор различных правил оформления и форматирования, который может быть применен к различным элементам Web-страницы. Применяя технологию каскадных стилей, достаточно будет один раз описать свойства элементов и определить это описание как стиль, а в дальнейшем просто указывать, что элемент, который необходимо форматировать, должен принять свойства стиля, описанного ранее.

Можно также сохранить описание стиля не в тексте самой Web-страницы, а в отдельном HTML-файле, что позволит применять это описание стиля для любого количества документов.

Описания стилей размещаются между дескрипторами **<STYLE>... </STYLE>** и должны располагаться внутри заголовка **<HEAD>... </HEAD>** документа.

### Общий вид описания стиля:

```
<STYLE>
.имя_стиля{атрибут1:значение; атрибут2:значение;...}
<STYLE>
или
<STYLE>
<тэг>{атрибут1:значение; атрибут2:значение;...}
<STYLE>
```

**Пример:** описываем стиль под именем *Example*.

```
<STYLE>
.Example{font-family: Comic Sans MS, Tahoma;
Font-weight: bolder; font-size: 25pt;
Color: darkgreen}
</STYLE>
```

....

Далее, где-то в тексте вызываем описанный стиль:

```
<div class="Example">Пример</div>
```

При размещении описания стилей в отдельном файле между дескрипторами **<HEAD> ...</HEAD>** всех использующих эти стили HTML-документов следует добавить ссылку на CSS-файл.

```
<LINK REL=STYLESHEET TYPE="TEXT/CSS" HREF="example.css">
```

В данном примере файл `example.css` содержит описание применяемых стилей. Если он находится в другом каталоге, укажите к нему путь.

Создать CSS-файл можно в любом текстовом редакторе (но только не в Word). Потребуется лишь изменить расширение созданного текстового файла на `.css`. Кроме этого, в CSS-файле не должны указываться дескрипторы **<STYLE>...</STYLE>**.

Стиль для любого тэга можно определить отдельно. Для этого в этот тэг нужно добавить параметр **STYLE** и описать его стиль в кавычках. Как это сделать, показывает следующий пример:

```
<H3 STYLE="font-family:Verdana;font-size:150%; color:red">Текст</h3>
```

Рассмотрим наиболее важные свойства элементов, которыми можно управлять с помощью технологии CSS. Для удобства они собраны в табл. 8.

## Свойства элементов CSS

Свойства шрифта	
<b>Font-family</b>	Используется для указания шрифта или шрифтового семейства, которым будет отображаться элемент. Можно указать два шрифта через запятую, тогда если на компьютере пользователя нет 1 шрифта, будет отображаться 2 шрифт. P{ <b>font-family</b> :Verdana, sans-serif}
<b>Font-weight</b>	Определяет степень жирности шрифта с помощью трех параметров: <b>lighter</b> (обычный), <b>bold</b> (полужирный), <b>bolder</b> (жирный). Жирность шрифта также определяется цифрами от 100 до 900. B{ <b>font-weight</b> :bolder}
<b>Font-size</b>	Задаёт размер шрифта. Может указываться в процентах, пикселях, пунктах, дюймах, сантиметрах а также задаваться текстовыми значениями: <b>small</b> (небольшой), <b>medium</b> (средний), <b>large</b> (большой), <b>larger</b> (очень большой), <b>smaller</b> (очень маленький). H1{ <b>font-size</b> :250%} H2{ <b>font-size</b> :150 px} H3{ <b>font-size</b> :400 pt}
<b>Font-variant</b>	Задаёт вид текста (обычный либо написание заглавными буквами с размером строчных). Допустимые значения: <b>normal</b> , <b>small-caps</b> . По умолчанию используется <b>normal</b>
Свойства текста	
<b>Text-decoration</b>	Устанавливает эффекты оформления текста, такие как подчеркивание или перечеркивание. Допустимые значения: <b>none</b> (нет), <b>overline</b> (подчеркивание сверху), <b>underline</b> (подчеркивание снизу), <b>line-through</b> (перечеркивание) H4{ <b>text-decoration</b> :underline}
<b>Text-align</b>	Определяет выравнивание текстового элемента. P{ <b>text-align</b> : left}
<b>Text-indent</b>	Устанавливает отступ для первой строки текста. Чаще всего используется для создания параграфов с табулированной первой строкой. H1{ <b>text-indent</b> : 60 pt}
<b>Text-transform</b>	Задаёт режим трансформации текста. Допустимые значения: <b>capitalize</b> (делает заглавной первую букву каждого слова), <b>uppercase</b> (делает все буквы заглавными), <b>lowercase</b> (делает все буквы строчными), <b>none</b> (снимает все установки)
<b>Line-height</b>	Задаёт интервал между строками текста P{ <b>Line-height</b> : 50% }
<b>List-style</b>	Управляет отображением элементов списка. <b>Position</b> – расположение списка, <b>List-style-type</b> – вид маркера

<b>Свойства шрифта</b>	
<b>Vertical-align</b>	Расположение элемента по вертикали. Допустимые значения: <b>baseline</b> – выравнивание по образцу родительского элемента, <b>sub</b> – переводит элемент в нижний регистр, <b>super</b> – в верхний регистр, <b>top</b> – по верху самого высокого элемента строки, <b>bottom</b> – по низу самого низкого элемента строки, <b>text-top</b> – по верху текста, <b>text-bottom</b> – по низу текста, набранного шрифтом родительского элемента, <b>middle</b> – выравнивание по средней линии
<b>Цвет элемента и фона</b>	
<b>color</b>	Определяет цвет элемента. H3{color: red}
<b>background</b>	Устанавливает цвет фона для элемента
<b>Background-attachment</b>	Определяет наличие фиксации фона. Значения: <b>scroll</b> – фон прокручивается вместе с содержанием, <b>fixed</b> – неподвижный фон
<b>Background-image</b>	Задаёт рисунок фона. <b>None</b> или адрес расположения рисунка – URL(‘адрес файла’)
<b>Background-position</b>	Расположение фонового изображения. <b>Top, center, bottom, left, right</b> или цифровые значения
<b>Background-color</b>	Задаёт цвет фона, <b>transparent</b> (прозрачность) или заданный цвет
<b>Background-repeat</b>	Устанавливает вариант повторения фонового изображения. <b>Repeat</b> – картинка бесконечно повторяется по горизонтали и вертикали, <b>repeat-x</b> – повторяется по горизонтали, <b>repeat-y</b> – повторяется по вертикали, <b>no-repeat</b> – не повторяется.

Единицы измерения – **Px** – пиксели, **cm** – сантиметры, **mm** – миллиметры, **pt** – пункты, **%** – проценты.

### Упражнения для изучения HTML

Прежде чем начать создавать страницы, необходимо создать папку, где Вы будете их хранить.

Создайте папку с именем *Web-страницы*, под руководством преподавателя скопируйте в нее папку с именем *Images* с необходимыми для работы рисунками.

Выполняя упражнения и задания, сохраняйте документы в папке *Web-страницы*, а рисунки копируйте в папку *Images*.

#### *Упражнение 1. Создание простейшей Web-страницы*

1. Запустите текстовый редактор *Блокнот* (*Пуск>Программы>Стандартные>Блокнот*).

2. Введите следующий текст в документ:

```
<HTML>
<HEAD>
<TITLE> Заголовок документа </TITLE>
</HEAD>
<BODY>
Содержание документа
</BODY>
</HTML>
```

(тэги можно вводить как большими, так и маленькими буквами, например, <HTML> или <html>).

3. Сохраните этот документ под именем first.htm (*Файл* > *Сохранить как*) в папке *Web-страницы*.

4. Откройте папку *Web-страницы* и найдите документ first.htm. Откройте его и обратите внимание на то, в какой программе он открывается, какой заголовок у окна и какой текст выводится в окне.

### ***Упражнение 2. Изучение приёмов форматирования абзацев***

1. Откройте документ first.htm в программе *Блокнот*. Сделать это можно с помощью щелчка правой кнопкой по документу, после чего в меню нужно выбрать *Открыть с помощью* > *Блокнот*.

2. Удалите весь текст, находящийся между тэгами <BODY> и </BODY>. Текст, который будет вводиться в последующих пунктах этого упражнения, необходимо поместить после тэга <BODY>, а его конкретное содержание может быть любым.

3. Введите заголовок первого уровня, заключив его между тэгами <H1> и </H1> (например, <h1>Привет! </h1>).

4. Введите заголовок второго уровня, заключив его между тэгами <H2> и </H2>.

5. Введите отдельный абзац текста, начав его с тэга <P>. Пробелы и символы перевода строки можно использовать внутри абзаца произвольно.

6. Введите тэг горизонтальной линейки <HR>.

7. Введите ещё один абзац текста, начав его с тэга <P>. У вас должно получиться примерно следующее:

```
<HTML>
<HEAD>
<TITLE> Заголовок документа </TITLE>
</HEAD>
<BODY>
<h1>Эта пара тэгов создает заголовок первого уровня</h1>
<h2>Эта пара тэгов создает заголовок второго уровня</h2>
<P>Этот тег начинает новый абзац.<br>Этот тэг начинает новую строку в абзаце.
<HR>
<r>Предыдущий тэг задает горизонтальную линию
</BODY>
</HTML>
```

8. Сохраните этот документ под именем paragraph.htm в папке *Web-страницы*.

9. Откройте файл paragraph.htm в браузере, выполнив двойной щелчок левой кнопкой мыши.

10. Посмотрите, как отображается этот файл. Установите соответствие между элементами кода *HTML* и фрагментами документа, отображаемыми на экране.

### ***Упражнение 3. Применение тэга STYLE для изменения стиля тэгов***

1. Запустите текстовый редактор *Блокнот* (*Пуск>Программы>Стандартные>Блокнот*).

2. Введите следующий текст в документ:

```
<html>
<head>
  <title>Тег STYLE</title>
  <style type="text/css">
    H1 {
      font-size: 120%;
      font-family: Verdana, Arial, sans-serif;
      color: blue;
    }
  </style>
</head>
<body>

  <H1>Заголовок 1 уровня</H1>

</body>
</html>
```



3. Сохраните этот документ под именем st1.htm (*Файл* > *Сохранить как*) в папке *Web-страницы*.

4. Откройте документ st1.htm в браузере. Обратите внимание на то, как выглядит заголовок первого уровня. Сравните его с заголовком из предыдущего упражнения.

5. Откройте документ st1.htm в программе *Блокнот*. Создайте свой стиль для тэгов H2 и P (для этого дополните существующий тэг STYLE).

6. Сохраните текстовый документ и откройте st1.htm в браузере. Проверьте, что созданные вами стили работают.

7. Закройте документ st1.htm.

#### **Упражнение 4. Создание гиперссылок**

1. Если это упражнение выполняется не сразу после предыдущего, откройте документ first.htm в программе *Блокнот*.

2. Удалите весь текст, находящийся между тэгами <BODY> и </BODY>. Текст, который будет вводиться в последующих пунктах этого упражнения, необходимо поместить после тэга <BODY>.

3. Введите фразу *Текст до ссылки*.

4. Введите тэг <A HREF=""first.htm">.

5. Введите фразу *Ссылка (перейти к документу first)*.

6. Введите закрывающий тэг </A>.

7. Введите фразу *Текст после ссылки*.

8. У вас должно получиться примерно следующее:

```
Файл  Правка  Формат  Вид  Справка
<HTML>
<HEAD>
<TITLE> Заголовок документа </TITLE>
</HEAD>
<BODY>
Текст до ссылки
<A HREF=""first.htm">Ссылка(перейти к документу first)</A>
Текст после ссылки.

</BODY>
</HTML>
```

9. Сохраните документ под именем link.htm.

10. Откройте файл link.htm из своей папки.

11. Убедитесь в том, что текст между тэгами <A> и </A> выделен как ссылка (цветом и подчеркиванием).

12. Щёлкните на ссылке и убедитесь, что при этом загружается документ, на который указывает ссылка, т.е. first.htm.

13. Щёлкните на кнопке *Назад* на панели инструментов браузера, чтобы вернуться к предыдущей странице. Убедитесь, что ссылка теперь считается «просмотренной» и отображается другим цветом.

### **Упражнение 5. Добавление изображения на Web-страницу**

1. Под руководством преподавателя скопируйте в папку *Images* файл cat.jpg (или выберите другой графический файл с небольшим рисунком и скопируйте его в папку *Images*).

2. Если это упражнение выполняется не сразу, а после предыдущего, откройте документ first.htm в программе *Блокнот*.

3. Удалите весь текст, находящийся между тэгами <BODY> и </BODY>. Текст, который будет вводиться в последующих пунктах этого упражнения, необходимо поместить после тэга <BODY>.

4. Введите произвольный текст (протяженностью две-три строки) и установите текстовый курсор в его начало.

5. Введите тэг <IMG SRC="Images\cat.jpg" ALIGN="BOTTOM">. Если Ваш файл называется по-другому, то его название должно быть написано вместо cat.jpg. Обратите внимание на расширение файла.

6. У вас должно получиться примерно следующее:

```
Файл  Правка  Формат  Вид  Справка
<HTML>
<HEAD>
<TITLE>Мой кот
<TITLE>
</HEAD>
<BODY>
А сейчас вы увидите фотографию моего любимого кота.
его зовут Мурзик и ему 2 года.
Он любит лежать на диване и смотреть телевизор.
<IMG SRC="Images\cat.jpg" ALIGN="Bottom">
</BODY>
</HTML>
```

Обратите внимание на заголовок *Мой кот*, он расположен между тэгами <TITLE> и </TITLE>. Этот заголовок окна обычно соответствует содержанию Web-страницы, выводимой в окне.

7. Сохраните документ под именем picture.htm.

8. Перейдите в свою папку и двойным щелчком мыши откройте файл picture.htm. Посмотрите на получившийся документ, обращая особое внимание на изображение.

9. Вернитесь в программу *Блокнот* и измените значение атрибута: ALIGN="TOP". Сохраните файл под тем же именем (*Файл* > *Сохранить*)

10. Вернитесь в браузер и щелкните на кнопке *Обновить* на панели инструментов. Посмотрите, как изменился вид страницы при изменении атрибутов.

11. Вернитесь в программу *Блокнот* и измените значение атрибута: ALIGN="LEFT". Сохраните файл под тем же именем.

12. Вернитесь в браузер и щелкните на кнопке *Обновить* на панели инструментов. Посмотрите, как изменился вид страницы при изменении атрибутов.

13. Вернитесь в программу *Блокнот* и добавьте в тэг <IMG> атрибуты: HSPACE=40 VSPACE=20. Должно получиться так:

```
<IMG SRC="Images/cat.jpg" ALIGN="LEFT" HSPACE=40 VSPACE=20>.
```

14. Сохраните файл под тем же именем.

15. Вернитесь в браузер и щелкните на кнопке *Обновить* на панели инструментов. Посмотрите, как изменился вид страницы при изменении атрибутов.

### ***Упражнение 6. Приёмы форматирования текста***

1. Если это упражнение выполняется не сразу после предыдущего, откройте документ first.htm в программе *Блокнот*.

2. Удалите весь текст, находящийся между тэгами <BODY> и </BODY>. Текст, который будет вводиться в последующих пунктах этого упражнения, необходимо поместить после тэга <BODY>, а его конкретное содержание может быть любым.

3. Введите произвольный абзац текста. Начните этот абзац с тэга <P>.

4. Введите тэги:

```
<P> <FONT SIZE="-2" FACE="ARIAL" COLOR="GREEN">.
```

5. Введите очередной абзац текста, закончив его тэгом </FONT>.

6. Введите еще тэги:

```
<P><FONT SIZE="6" FACE="Century" COLOR="Yellow">. Введите текст и закончите его тэгом </FONT>.
```

7. Начните новый абзац с тэга <P> и используйте в нем парные тэги: <B> (*полужирный шрифт*), <I> (*курсив*), <U> (*подчёркивание*), <S> (*вычеркивание*),

<SUB> (нижний индекс), <SUP> (верхний индекс). (например, <b> Этот текст будет выделен полужирным шрифтом</b>).

8. В следующем абзаце используйте парные тэги: <EM> (выделение), <STRONG> (сильное выделение).

9. У вас должно получиться примерно следующее:

```
*format – Блокнот
Файл Правка Формат Вид Справка
<HTML>
<HEAD>
<TITLE> Заголовок документа </TITLE>
</HEAD>
<BODY bgcolor=navy BACKGROUND="primer.jpg">
<BASEFONT SIZE=5 COLOR= "BROWN">
<P>Сейчас вы увидите примеры форматирования текста на Web-странице.
Этот текст выводится коричневым цветом и размером 5.
<P> <FONT SIZE="-2" FACE= "ARIAL" COLOR="GREEN">
Этот текст зеленого цвета, тип шрифта - ARIAL, размер на 2 меньше, чем базовый шрифт.
</FONT>
<P><FONT SIZE="6" FACE= "Century" COLOR="Yellow">
Этот текст желтого цвета, тип шрифта - Century , размер - 6.
</FONT>
<P><b>Этот текст будет выделен полужирным шрифтом</b>
<i>Этот текст будет выделен курсивом</i>
<u>Этот текст будет выделен подчеркиванием</u>
<S>Этот текст будет выделен полужирным шрифтом</S>
<sub>Этот текст будет представлен в виде нижнего индекса</sub>
<sup>Этот текст будет представлен в виде верхнего индекса</sup>
<P><em>этот текст будет выделен</em>
<strong>этот текст будет сильно выделен</strong>
</BODY>
</HTML>
```

10. Сохраните полученный документ под именем format.htm.

11. Перейдите в свою папку и двойным щелчком мыши откройте файл format.htm.

12. Изучите, как использованные элементы **HTML** влияют на способ отображения текста.

13. Вернитесь в программу *Блокнот* и измените документ так, чтобы элементы, задающие форматирование, были вложены друг в друга. Например, таким образом: <B><I>Текст</I></B>. Сохраните документ под тем же именем.

14. Вернитесь в браузер и щелкните на кнопке *Обновить* на панели инструментов. Посмотрите, как изменился вид страницы.

### Упражнение 7. Приёмы форматирования текста с тэгом `div`

1. Откройте новый документ в программе *Блокнот*.
2. Введите текст по образцу:

```
<html>
<head>
  <title>Тег DIV</title>
  <style type="text/css">
    .block1 {
      width: 200px;
      background: #ccc;
      padding: 5px;
      padding-right: 20px;
      border: solid 1px black;
      float: left;
    }
    .block2 {
      width: 300px;
      background: #fc0;
      padding: 5px;
      border: solid 5px blue;
      float: left;
      position: relative;
      top: 40px;
      left: -70px;
    }
  </style>
</head>
<body>
  <div class="block1">Текст 1 с оформлением block1</div>
  <div class="block2">Текст 1 с оформлением block2</div>
</body>
</html>
```

3. Сохраните полученный документ под именем `div.htm`.
4. Перейдите в свою папку и двойным щелчком мыши откройте файл `div.htm`. У вас должно получиться примерно такое содержимое страницы:

Текст 1 с оформлением  
block1

Текст 1 с оформлением block2

5. Обратите внимание на разницу в оформлении двух строк текста. Найдите в коде HTML те места, которые отвечают за фон текста, отступ, рамку вокруг текста, за ширину блока.

6. Откройте документ `div.htm` в *Блокноте*, измените некоторые параметры в коде (цвет или толщину рамки, ширину блока и т.п.), сохраните текстовый документ и обновите html-страницу.

### ***Упражнение 8. Создание цветного и графического фона страницы***

1. Если это упражнение выполняется не сразу после предыдущего, откройте документ `format.htm` в программе *Блокнот*.

2. Для создания цветного фона страницы введите в тэг `<BODY>` атрибут `BGCOLOR`: `<BODY BGCOLOR=navy>`. После знака «=» пишется название цвета страницы на английском языке или шестнадцатеричный код цвета. Если цвет *navy* не устраивает, можно вместо него ввести любой цвет из таблицы цветов.

3. Сохраните документ под тем же именем и нажмите на кнопку *Обновить* в браузере. При необходимости замените фоновый цвет на другой.

4. Для создания графического фона необходим фоновый рисунок. Под руководством преподавателя скопируйте в папку *Images* файл с названием `primer.jpg` (или скопируйте любой другой фоновый рисунок).

5. В тэг `<BODY>` поместить атрибут `BACKGROUND`: `<BODY BACKGROUND="Images /primer.jpg">`.

6. Сохраните документ под именем `format1.htm` в своей папке. Затем откройте документ в браузере и изучите его. При необходимости замените фоновый рисунок на другой.

### ***Упражнение 9. Приёмы создания списков***

1. Если это упражнение выполняется не сразу после предыдущего, откройте документ `first.htm` в программе *Блокнот*.

2. Удалите весь текст, находящийся между тэгами `<BODY>` и `</BODY>`. Текст, который будет вводиться в последующих пунктах этого упражнения, необходимо поместить после тэга `<BODY>`, а его конкретное содержание может быть любым.

3. Вставьте в документ тэг `<OL TYPE= "I">`, который начинает упорядоченный (нумерованный) список.

4. Вставьте в документ элементы списка, предваряя каждый из них тэгом `<LI>`.

5. Завершите список при помощи тэга `</OL>`:

```
<OL TYPE= "I">
<LI>Первый элемент нумерованного списка
<LI>Второй элемент нумерованного списка
<LI>Третий элемент нумерованного списка
</OL>
```

6. Сохраните полученный документ под именем list.htm.

7. Перейдите в свою папку и двойным щелчком мыши откройте файл list.htm.

8. Изучите, как упорядоченный список отображается в браузере, обращая особое внимание на способ нумерации, заданный при помощи атрибута «TYPE=».

9. Вернитесь в программу *Блокнот* и установите текстовый курсор после окончания введённого списка.

10. Вставьте в документ `<UL TYPE= "square">`, который начинает неупорядоченный (маркированный) список.

11. Вставьте в документ элементы списка, предваряя каждый из них тэгом `<LI>`:

```
<ul type="square">
<LI>Первый элемент маркированного списка
<LI>Второй элемент маркированного списка
<LI>Третий элемент маркированного списка
</UL>
```

12. Завершите список при помощи `</UL>`. Сохраните документ под тем же именем.

13. Вернитесь в браузер и щелкните на кнопке *Обновить* на панели инструментов. Посмотрите, как изменился вид страницы, обратив внимание на способ маркировки, заданный при помощи атрибута «TYPE=».

14. Перейдите в программу *Блокнот* и установите текстовый курсор после окончания введённого маркированного списка.

15. Вставьте в документ тэг `<DL>`, который начинает список определений, далее введите `<DT>`, после которого следует название термина. Каждое определение термина необходимо предварять `<DD>`.

```
<DL>
<DT>Название термина
<DD>Определение термина 1
<DD>Определение термина 2
</DL>
```

16. Завершите список при помощи `</DL>`. Сохраните документ под тем же именем.

17. Вернитесь в браузер и щелкните на кнопке *Обновить* на панели инструментов. Посмотрите, как выглядит при отображении Web-страницы список определений.

## Упражнение 10. Создание таблиц

1. Если это упражнение выполняется не сразу после предыдущего, откройте документ `first.htm` в программе *Блокнот*.

2. Удалите весь текст, находящийся между `<BODY>` и `</BODY>`. Текст, который будет вводиться в последующих пунктах этого упражнения, необходимо поместить после тэга `<BODY>` (в этом упражнении используется список номеров телефонов).

3. Введите `<TABLE BORDER="10" WIDTH="100%">`.

4. Введите строку: `<CAPTION ALIGN="TOP"> Список телефонов </CAPTION>`.

5. Первая строка таблицы должна содержать заголовки столбцов. Определите её следующим образом:

```
<TR BGCOLOR=YELLOW ALIGN="CENTER">
<TH> фамилия <TH> Номер телефона
```

6. Определите последующие строки таблицы, предваряя каждую из них тэгом `<TR>` и помещая содержимое каждой ячейки после `<TD>`.

7. Последнюю строку таблицы задайте следующим образом:

```
<TR> <TD ALIGN="CENTER" COLSPAN=2>На первом этаже здания имеется бесплатный телефон-автомат.
```

8. Завершите таблицу `</TABLE>`. Вот так выглядит полный код создания таблицы:

```
Файл  Правка  Формат  Вид  Справка
<HTML>
<HEAD>
<TITLE> Заголовок документа </TITLE>
</HEAD>
<TABLE BORDER="10" WIDTH="100%">
<CAPTION ALIGN="TOP"> Список телефонов </CAPTION>
<TR BGCOLOR=YELLOW ALIGN="CENTER">
<TH> фамилия <TH> Номер телефона
<TR><TD>Иванов<TD>321654
<TR><TD>Петров<TD>456987
<TR><TD>Сидоров<TD>456982
<TR> <TD ALIGN="CENTER" COLSPAN=2>
На первом этаже здания имеется бесплатный телефон-автомат
</TABLE>
</BODY>
</HTML>
```

9. Сохраните документ под именем `table.htm`.

10. Перейдите в свою папку и двойным щелчком мыши откройте файл `table.htm`.

11. Изучите, как таблица отображается в браузере, обращая особое внимание на влияние заданных атрибутов.



12. Измените атрибут WIDTH со 100% на 50%, т.е. WIDTH="50%". Сохраните изменения, запустите обозреватель и установите, как при этом изменяется внешний вид таблицы.

### ***Упражнение 11. Управление параметрами отображения таблицы***

1. Скопируйте в папку *Images* рисунок *Back10.jpg*.
2. Если это упражнение выполняется не сразу после предыдущего, откройте документ *first.htm* в программе *Блокнот*.
3. Удалите весь текст между `<body>` и `</body>`.
4. В документ после `<BODY>` введите следующий код:

```
<Font face=Century>
<TABLE bordercolor=yellow background="Images\Back10.jpg" frame=box
WIDTH="50%" cellspacing=30 cellpadding=10 rules=all>
<CAPTION ALIGN="TOP"> Список телефонов </CAPTION>
<TR BGCOLOR=YELLOW ALIGN="CENTER" VALIGN=CENTER>
<TH> Фамилия <TH> Имя
<TR VALIGN=top><TD><font color= maroon size=6>Иванов<TD><font
color=aqua size=5>Александр
<TR VALIGN=top ><TD><font color=orchid size=6>Петров<TD><font
color=navy size=5>Сергей
<TR VALIGN=top ><TD><font color=teal size=6>Сидоров<TD><font
color=orange size=5>Иван
<TR> <TD ALIGN="CENTER" COLSPAN=2 ><font color= violet size=5>
Список студентов, имеющих задолженности за летнюю сессию
</TABLE>
</Font>
```

5. Сохраните документ в своей папке под именем *table1.htm*.
6. Откройте документ *table1.htm* в браузере, обратите внимание на параметры отображения таблицы.
7. Откройте документ *table1.htm* в *Блокноте* и измените в `<TABLE>` значение параметров: *frame=above width="70%" rules=rows*. Сохраните изменения в *table1.htm*. Нажмите в браузере кнопку *Обновить*, обратите внимание на изменения во внешнем виде таблицы.
8. Измените значения параметров: *frame=below rules=cols*. Сохраните изменения в *table1.htm*. Нажмите в браузере кнопку *Обновить*, обратите внимание на изменения во внешнем виде таблицы.

9. Измените значения параметра: *frame=hsides*. Сохраните изменения в *table1.htm*. Нажмите в браузере кнопку **Обновить**, обратите внимание на изменения во внешнем виде таблицы.

10. Измените значения параметров: *frame=rhs rules=rows*. Сохраните изменения в *table1.htm*. Нажмите в браузере кнопку *Обновить*, обратите внимание на изменения во внешнем виде таблицы.

11. Измените толщину рамки таблицы, добавив в `<table>` параметр `<border=10>`, измените цвет рамки – `<bordercolor=>` на любой другой, кроме черного. Сохраните изменения в *table1.htm*. Нажмите в браузере кнопку *Обновить*, обратите внимание на изменения во внешнем виде таблицы.

### ***Упражнение 12. Создание описания фреймов***

1. Запустите текстовый редактор *Блокнот* (*Пуск > Программы > Стандартные > Блокнот*).

2. Введите следующий код:

```
<HTML>
<HEAD>
<TITLE>Описание фреймов</TITLE>
</HEAD>
<FRAMESET ROWS="60%,*">
<FRAME SRC="table.htm" NAME="one">
<FRAMESET COLS="35%, 65%" NORESIZE>
<FRAME SRC="first.htm" NAME="two">
<FRAME SRC="link.htm" NAME="three">
</FRAMESET>
</FRAMESET>
</HTML>
```

3. Сохраните этот документ под именем *frames.htm*.

4. Перейдите в свою папку и двойным щелчком мыши откройте файл *frames.htm*.

5. Изучите несколько созданных ранее документов, расположенных в отдельных фреймах.

6. Посмотрите, что происходит при изменении ширины окна обозревателя.

7. Проверьте, можно ли изменить положение границ фреймов методом перетаскивания при помощи мыши.

8. Щёлкните на ссылке, имеющейся в одном из фреймов, и посмотрите, как будет отображен новый документ.

9. Щёлкните на кнопке *Назад* на панели инструментов и убедитесь, что возврат к предыдущему документу не нарушает структуру фреймов.

### **Упражнение 13. Создание панели ссылок с помощью фрейма**

1. Запустите текстовый редактор *Блокнот* (*Пуск* > *Программы* > *Стандартные* > *Блокнот*).

2. Введите следующий код:

```
<HTML>
<HEAD>
<TITLE>Панель ссылок</TITLE>
</HEAD>
<FRAMESET COLS="30%,*">
<FRAME SRC="panel.htm" NAME="menu">
<FRAME SRC="first.htm" NAME="main">
</FRAMESET>
</HTML>
```

3. Сохраните документ под именем *frames2.htm* в своей папке.

4. Откройте документ *link.htm* в программе *Блокнот*.

5. В документ после `</A>` введите следующий код:

```
<BR><A href="list.htm" target="main"> Списки</A>
<BR><A href="table.htm" target="main"> Таблица</A>
<BR><A href="format.htm" target="main"> Формат</A>
```

6. Сохраните документ под именем *panel.htm* в своей папке.

7. Откройте документ *frames2.htm* в браузере. Проверьте, что ссылки, расположенные на панели слева, открывают соответствующие веб-страницы в правом окне. За это отвечает параметр *target="main"*, размещенный в ссылке.

8. Закройте документ *frames2.htm*.

### **Упражнение 14. Создание и применение листов стилей**

1. Откройте документ *first.htm* в *Блокноте*.

2. Измените имеющийся HTML-код в соответствии со следующим примером:

```

<html>
<head>
<title>Каскадные листы стилей</title>
<style>
.stil{font-family:Comic Sans MS;
font-weight:bolder; font-size: 24pt;
color:navy}
H1{font-size:30 pt; color:pink;
text-align:center; background:green}
</style>
</head>
<body>
Проверяем работу каскадных листов стилей
<div class="stil">Вот так применяется стиль stil</div>
Так применяется <h1>измененный стиль заголовка H1</h1>
</body>
</html>

```

Обратите внимание, что название класса стиля начинается с точки!

В этом примере создается произвольный стиль *Stil* со следующими параметрами: шрифт – Comic Sans, начертание полужирное, размер – 24 пт, цвет – темно-синий. Также изменяется значение H1. Теперь помимо заголовка 1 уровня этот тэг также задает размер 30 пт, розовый цвет шрифта, выравнивание по центру и зеленый фон для текста.

3. Сохраните введенный текст под именем cascade.htm в своей папке.

4. Откройте документ cascade.htm в браузере и изучите вид страницы.

5. Измените любые параметры в описании стиля *Stil*, сохраните изменения в документе cascade.htm, проследите за изменениями в форматировании страницы.

6. Откройте программу Блокнот, введите следующий текст:

```

.mystyle {
font-family:Century Gothic; font-size: 16pt;
text-align:center; letter-spacing: 2pt;
color: yellow; background:navy;}
H3 {
font-family:Verdana; font-size: 18pt;
text-align:right; font-weight: bolder;
text-decoration:underline;
color: red; }

```

7. Сохраните документ в своей папке под именем stili.css.

8. В этом файле описаны два стиля – *mystyle* и H3. Для их применения откройте в редакторе *Блокнот* произвольную Web-страницу из своей папки, например, страницу paragraph.htm.

9. Добавьте в заголовок <HEAD>...</HEAD> этой страницы следующую строку: <link rel=STYLESHEET type="text/css" href="stili.css">.

10. В теле страницы в любом месте примените оба стиля к произвольному тексту. Например,

<div class="mystyle"> Стиль mystyle </div>

<h3>Измененный стиль заголовка H3</h3>.

11. Сохраните изменения в документе, откройте страницу paragraph.htm в браузере. Изучите внешний вид страницы. Закройте документ.

12. Откройте другую страницу и примените оба этих стиля для форматирования ее элементов.

13. Создайте самостоятельно документ с названием format.css, где разработайте два собственных стиля – *casstyle* и *P*.

14. Примените оба этих стиля к любой ранее созданной Web-странице.

## Задания для самостоятельной работы к главе 2

При выполнении заданий страницы разместите в папке *Самостоятельная работа*, рисунки – внутри папки *Images*.

### Задание 1

Создайте две страницы «*Моя личная страница*» и «*Моя биография*». Сохранить их в своей папке.

1. Создайте web-страницу с заголовком «*Моя личная страница*»

2. В теле страницы поместите:

- фамилию, имя как заголовок первого уровня;
- с новой строки – факультет, курс, группа – как заголовок второго уровня;
- любимые предметы обычным текстом с новой строки;
- хобби – с новой строки;
- сохраните страницу в своей папке под именем personal.html.

3. Создайте web-страницу с заголовком «*Моя биография*»

4. В теле поместите (каждый пункт с новой строки):

- фамилию и имя – как заголовок второго уровня;
- дата рождения – как заголовок третьего уровня;
- место рождения;
- место проживания, в какой школе обучались, в каком году поступили в ПГНИУ;
- семейное положение;
- сохраните страницу в своей папке под именем biografia.html.

5. В нижней части страницы biografia.html добавьте изображение, создайте подпись к изображению.

6. Под изображением разместите гиперссылку на страницу personal.html с текстом «Назад на Мою личную страницу».

7. Сохраните изменения в документе biografia.html.

8. Просмотрите обе страницы в браузере, перейдите на personal.html и вставьте внизу этой страницы гиперссылку на страницу biografia.html с текстом «Хотите узнать обо мне больше? Прочитайте мою биографию!»

9. Проверьте как работают гиперссылки, убедитесь, что обе страницы сохранены в папку *Самостоятельная работа*.

## **Задание 2**

1. Создайте web-страницу с заголовком «Мой факультет».

2. Задайте на странице шрифт Garamond, цвет текста – purple.

3. В теле страницы разместите:

- заголовок первого уровня « ... факультет» (вместо точек вставьте название факультета), выровняйте по центру (<center>...</ center >);
- с новой строки – «Декан Ф.И.О.» – как заголовок второго уровня, выделите цветом, курсивом, шрифтом Arial;
- с новой строки – «Кафедры факультета» – как заголовок второго уровня, разместите по центру.
- ниже поместите список кафедр факультета, оформив его как маркированный список, выделите весь список зеленым цветом и шрифтом Courier.
- с новой строки – «Моя группа» – как заголовок второго уровня, разместите по центру.
- добавьте список студентов Вашей группы, оформив его как нумерованный список (фамилия, инициалы), выделить красным цветом и шрифтом Monotype Corsiva.

4. Внизу страницы разместите рисунок, вокруг него создайте рамку из пробелов.

5. Задайте на странице графический фон.

6. Поместите внизу страницы гиперссылки на страницы biografia.html и personal.html.

7. Сохраните страницу в своей папке под именем facultet.html.

8. На страницы biografia.html и personal.html добавьте цветной или графический фон, а также гиперссылку на страницу facultet.html.

9. Проверьте работу всех гиперссылок на страницах и правильность сохранения страницы facultet.html.

### Задание 3

1. Создайте страницу с заголовком «Мой город». Задайте произвольные шрифт и цвет. В теле страницы разместите:

- вверху страницы – «Мой город Пермь» – заголовок первого уровня, выделите цветом, выровняйте по центру;
- с новой строки – «Месторасположение города –Пермский край». Оформите как заголовок второго уровня;
- с новой строки – «Основан в 1723 году и является одним из крупнейших промышленных и культурных центров России»;
- с новой строки поместите таблицу «Наши театры» (заголовок выделите цветом, другим шрифтом и выровняйте по центру):

#### Наши театры

<i>Театр</i>	<i>Адрес</i>	<i>Телефон касс</i>
Театр оперы и балета им. П.И. Чайковского	ул. Коммунистическая, 25а	212-30-87
Драматический театр	ул. Ленина, 53	236-10-92
Театр кукол	ул. К. Маркса, 65	244-18-72
Театр юного зрителя	ул. Большевикская, 68	212-43-16
Театр «У Моста»	ул. Куйбышева, 11	237-51-66
<b>Ждем Вас в наших театрах!</b>		

- Задайте графический фон на странице;
- В теле страницы поместите 3 картинки на тему «Пермь». Также создайте гиперссылки на все три предыдущие страницы. Сохраните страницу в своей папке под именем perm.html.

2. На страницу personal.html добавьте гиперссылку на страницу perm.html с текстом «Немного о городе, в котором я живу».

3. Проверьте работу гиперссылок, убедитесь, что вы сохранили все страницы в своей папке.

### Задание 4

1. Создайте страницу с заголовком «Ссылки». В теле страницы разместите гиперссылки на все ранее созданные страницы. Задайте графический или цветной фон. Сохранить под именем mylinks.html.

2. Создайте две страницы с фреймами: страницу frame1.html с двумя горизонтальными фреймами и страницу frame2.html с двумя вертикальными фреймами.

3. В `frame1.html` отобразите любые две созданные ранее страницы, в `frame2.html` в правой части фрейма отобразите страницу `personal.html`, а в левой – `mylinks.html`.

4. Настройте фрейм так, чтобы при нажатии на гиперссылки в левом окне вызываемая страница открывалась в правой части окна.

### Задание 5

1. Откройте документ `personal.html`. Создайте в нем две каскадных стиля, например, для заголовка первого уровня `<H1>` и еще один – с произвольным названием. Опишите их в тэге `<STYLE>`.

2. Примените эти два стиля в тексте.

3. Создайте файл `stilcascad.css`, содержащий каскадные стили (не менее двух).

4. Примените эти стили к оформлению своих предыдущих страниц – `perm.html` и `facultet.html`.

### Итоговое задание

1. Выберите тему для создания web-страниц. Тема не должна совпадать со следующими темами: *Биография, Факультет, Пермь*.

2. Для создания страниц подберите текст (в электронном виде) и рисунки по выбранной теме.

3. Создайте и сохраните в отдельной папке не менее 4 страниц, удовлетворяющих следующим требованиям:

- a. главная страница должна носить название `index.html`;
- b. на главной странице должны находиться гиперссылки на все последующие страницы и на созданный фрейм;
- c. на каждой странице должны быть не менее 20 слов текста (отвечающих теме), не менее 2 иллюстраций, гиперссылка на главную страницу (кроме главной страницы);
- d. для оформления страниц создайте два каскадных стиля и примените их на каждой странице. Каждый стиль должен содержать не менее 5 атрибутов, и в обязательном порядке параметры **font, size, color**;
- e. для всех страниц должен быть задан единый графический фон;
- f. как минимум на двух страницах должны быть размещены списки и как минимум на одной странице – таблица.

4. Создайте фрейм, в котором можно будет одновременно просматривать все созданные страницы.

5. Создайте страницу, содержащую ссылки на все созданные страницы.

6. Создайте фрейм из двух окон, в котором в нижнем окне будут открываться страницы, при нажатии на гиперссылки в верхнем окне.



## ГЛАВА 3. СОЗДАНИЕ САЙТА С ПОМОЩЬЮ ОНЛАЙН-КОНСТРУКТОРОВ

*Конструктор сайта* – это специализированный онлайн-сервис, располагающий готовыми вариантами графического оформления для создаваемых сайтов. Как правило, эти сервисы предоставляют пользователю удобный и интуитивно понятный интерфейс.

Современный конструктор сайтов дает возможность практически любому человеку создать свой собственный Интернет-ресурс, при этом нет необходимости получать знания в области веб-программирования и дизайна<sup>1</sup>. Наиболее востребованными конструкторами сайтов являются Google Sites, Tilda, UKit, Fo.ru, и другие.

### Google Sites

Сайты Google (англ. *Google Sites*) – упрощённый бесплатный хостинг на базе структурированной вики. Может использоваться как часть Google Apps. Позволяет при помощи технологии wiki сделать информацию доступной для людей, которые нуждаются в её быстрой подаче. Пользователи сайта могут работать вместе, добавлять информацию из других приложений Google, например, Документы Google, Календарь Google, YouTube, Google Фото и из других источников.

#### *Обзор конструктора сайтов Google Sites*

Google Sites – бесплатный и простой в использовании конструктор сайтов. Он подходит для создания небольших личных и образовательных проектов. На сервисе нет инструментов электронной коммерции, да и в целом для решения бизнес-задач он подходит слабо. Сфера применения Google Sites – быстрая разработка информационных страниц.

Главный плюс Google Sites – бесплатность. Это отличный вариант для пользователей, которым нужно очень быстро собрать простую информационную страницу без финансовых вложений. Тот факт, что конструктор предоставляется корпорацией Google и глубоко интегрирован в её экосистему, тоже производит приятное впечатление.

---

<sup>1</sup> Как создать свой сайт бесплатно своими руками. URL: <https://www.internet-technologies.ru/how-to-create-a-website.html> (дата обращения: 01.04.2023).

## ***Google Sites: Редактор***

Для создания сайта на Google Sites не нужны специальные навыки. Если вы работали с другими сервисами Google (например, Документы, Презентации или Таблицы), то без труда разберётесь с интерфейсом конструктора.

Для работы с конструктором нужен только браузер. Просматривать проекты можно на любом устройстве (гарантируется работа только на текущих и предыдущих стабильных версиях браузеров Chrome, Safari, Edge, Firefox), а редактировать – только на компьютере. Для корректной работы сервиса требуется браузер Chrome версии 24 и старше или Firefox последней версии. Кроме того, в настройках обозревателя должны быть включены JavaScript и сохранение файлов cookie.

Платформа отчасти может быть интересна для создания личных страниц, страниц некоммерческих учреждений (школы, детские сады и т.п.), официальных страниц социальных групп, небольших портфолио, визиток и при должном усердии – лендингов.

### ***Простота использования***

Сайт настраивается в визуальном редакторе с поддержкой технологии Drag-and-Drop. Вы просто перетаскиваете секции и элементы, выстраивая из них нужную структуру страниц. Глубоко кастомизировать дизайн не получится – инструменты редактирования ограничены. Зато не нужно думать о том, что можно испортить шаблон – все элементы выстраиваются по сетке автоматически, отступы и размеры подгоняются (см. примеры сайтов на Google Sites).

Собрать сайт в Google Sites можно за несколько часов, а иногда и несколько минут (если это простой одностраничник). Специальные навыки не требуются. Если Вы умеете работать с другими сервисами Google и делали хоть раз презентацию, то будете чувствовать себя уверенно при работе с визуальным редактором Google Sites.

### ***Начало работы***

Для работы с Google Sites нужен аккаунт Google. После авторизации Вы попадёте в дашборд. Дальше есть два варианта: создать сайт с нуля или воспользоваться готовым шаблоном.

Если хотите собирать интерфейс с нуля, нажмите на «Пустой файл» в галерее шаблонов. Однако для первого опыта рекомендуем выбрать более или менее подходящий тематический шаблон – он поможет понять, как выстраивается интерфейс каждой отдельно взятой страницы.

Если вы выбрали создание проекта с чистого листа, он будет обозначен как «Сайт без названия». Если вы выбрали шаблон, то у него будет стандартное название – например, «Портфолио». Чтобы изменить имя проекта, нажмите на строку с названием и введите свой вариант. Сайт готов, осталось наполнить его контентом и опубликовать.

Второй вариант создания – из меню Google Диска. Нажмите на кнопку «Создать» -> пункт «Ещё» -> элемент «Google Сайты», и Вы сразу попадёте в интерфейс конструктора, как если бы начали с чистого листа.

### ***Редактирование сайта***

После создания сайта Вы попадаете в визуальный редактор. У него очень простая структура: рабочее поле, на котором в реальном времени отображаются все изменения, и панель управления с тремя вкладками (рис. 15).

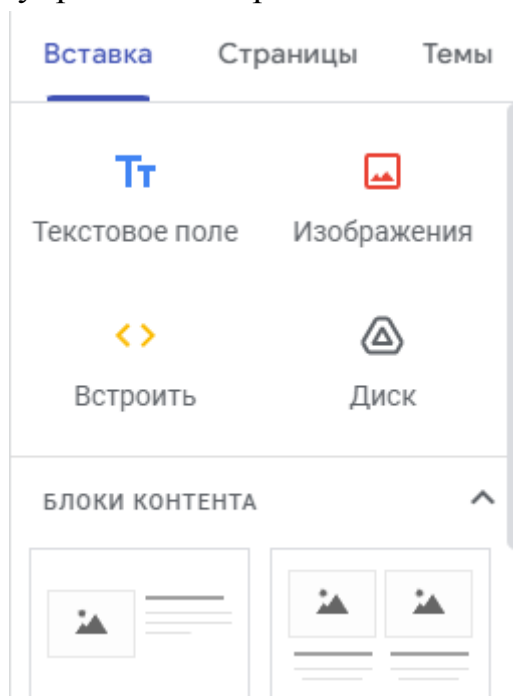


Рис. 15. Панель управления в Google Sites

1. *Вставка* – каталог блоков и элементов, из которых формируется интерфейс сайта. Здесь есть «Блоки контента» – это подготовленные схемы секций, внутри которых размещены изображения и текстовые блоки. Можно выбрать такой макет и гибко настроить его внешний вид: поменять местами элементы, удалить ненужные или добавить недостающие части, изменить размер и положение, заменить элементы на другие.

Можно не пользоваться готовыми секциями, а собрать интерфейс из отдельных элементов, которые можно гибко комбинировать. Например, добавить текстовое поле, а затем рядом с ним поставить изображение и разместить кнопку.

2. *Страницы* – здесь добавляются страницы и ссылки. По умолчанию они сразу отображаются на панели навигации сайта. Вы можете сделать любую страницу главной, создать полную её копию, изменить адрес, добавить подстраницу, удалить страницу.

3. *Темы* – варианты оформления сайта. В редакторе есть несколько готовых вариантов оформления, где можно менять цвет подложки и стиль шрифта. Изменения сохраняются автоматически.

В Google Сайтах поддерживается совместная работа над проектом. Можно пригласить других пользователей по адресу электронной почты или создать ссылку с правом редактирования черновика.

### ***Публикация сайта***

Прежде чем опубликовать сайт, убедитесь, что он полностью готов. Проще всего это сделать с помощью функции предварительного просмотра. Тут есть три режима: десктоп, планшет и смартфон. Проверьте, все ли элементы интерфейса отображаются верно.

Чтобы сайт стал общедоступным, нажмите на кнопку «*Опубликовать*». Появится окно, в котором нужно ввести веб-адрес. Обратите внимание: вам необходимо указать только последнюю часть адреса, например, `websitebuilders`. Полный адрес будет `https://sites.google.com/view/websitebuilders`.

Обратите внимание и на то, что общедоступный адрес может быть уже занят.

При публикации сайта вы также можете настроить его доступность. Например, можно пригласить других людей по электронной почте, чтобы они могли видеть сайт и вносить в него изменения.

Ещё одна полезная функция – управление индексацией. При публикации можно исключить сайт из поисковой выдачи. Эта настройка гарантирует, что контент со страниц не будет попадать в поисковые системы.

У Google Sites не слишком много возможностей по сравнению с другими конструкторами. Однако его функций вполне хватает для разработки простых сайтов.

### ***Интеграция с сервисами Google***

На сайты можно быстро вставлять файлы с Google Drive, Таблицы и Презентации, сделанные с помощью сервисов Google, видео с YouTube. Аналитика и инструменты веб-мастера подключаются в один клик. Вы также можете приглашать других авторов к разработке сайта, используя контакты из Gmail.

Рассмотрим некоторые возможности Google Форм:

- можно создавать не только простые поля для сбора контактных данных, но и выбирать даты, принимать файлы (они будут попадать в ваш Google Диск), предлагать множественный выбор из списка или из сетки элементов и т.п.;
- есть возможность организации тестирования с импортированием вопросов и ответов. Для тестов можно предусмотреть шкалу прогресса;
- варианты ответов можно перемешивать между собой, поэтому инструмент подойдёт даже для организации учебного процесса или для сборки небольших опросов. Ответы могут предусматривать назначение баллов и выставление итоговой оценки;
- историю ответов можно сохранить (по умолчанию они сохраняются в Google Таблицу) или выгрузить (в CSV-файл), также можно настроить отправку результатов респондентам;
- можно включить оперативные уведомления о наличии заполненных форм;
- Google Формы интегрируются с внешними приложениями, поэтому их функционал можно расширить. Например, добавить таймер обратного отсчёта, расчёт значений (онлайн-калькулятор), дополнительные поля для приёма заказа в интернет-магазинах и т.д.;
- в качестве внутреннего инструмента для сложных расчётов можно задействовать Google Apps Script (редактор скриптов и веб-приложений).

### ***Работа в команде***

Собрать команду можно двумя способами – с помощью индивидуальных приглашений по электронной почте и по ссылке на черновик с правом редактирования. При выборе второго способа нужно быть внимательным – каждый, у кого будет ссылка, получит неограниченное право на внесение изменений.

### ***История версий и копия сайта***

Версии сайта сохраняются автоматически с отображением даты внесения изменения и подсвечиванием различий. Также можно вручную сохранить копию сайта на Google Drive или отправить её другим пользователям.

Сайт при желании можно передать другим пользователям Google Диска («Открыть доступ», добавить нужного пользователя, в разделе «Уровни доступа» напротив нужного пользователя выбрать вариант «Владелец»).

## ***Библиотека изображений***

При создании сайта требуется много изображений, контентных и фоновых. Можно загружать их с компьютера, выбирать из встроенной галереи, добавлять из Google Drive, искать через Google и вставлять прямыми ссылками на сторонние источники.

Изображения после вставки анализируются специальным инструментом, и при необходимости алгоритм корректирует чёткость текста на их фоне. Вы можете поменять привязку картинки по отношению к блоку, внутри которого она используется: слева/справа/сверху, снизу/по центру, растянуть.

## **Конструктор сайтов Tilda**

Блочный конструктор сайтов, не требующий навыков программирования, позволяет создавать сайты, интернет-магазины, посадочные страницы, блоги и email-рассылки. Сайты на платформе собираются из готовых блоков, которые автоматически адаптируются под мобильные устройства и выделяются в смысловые категории.

### ***Что можно создать на Тильде***

Tilda – это инструмент для простой разработки:

- сайтов;
- лендингов товаров и услуг;
- интернет-магазинов;
- промостраниц, приглашений на события, презентаций;
- личного портфолио;
- корпоративных блогов;
- авторских блогов;
- спецпроектов.

В конструкторе есть спроектированные готовые блоки, из которых можно создавать сайт. Дизайнеры Тильды предлагают современные, трендовые решения для оформления элементов сайта, однако можно добавить интерактивные объекты и анимацию, не изучая программирование.

### ***Личный сайт или портфолио***

Если нужно собрать свои работы и презентовать их, то Тильда прекрасно подойдет. Сайт можно создать по готовому шаблону, наполнить контентом, подбирать дизайн и обложки, разделить по тематикам.

Тильду также можно использовать для публикаций анонсов событий. В конструкторе продумана типографика: подобраны ширина строк, интервалы и большой выбор шрифтов, в том числе авторских. Фото и видео можно быстро загрузить с диска, перетащить, встроить из YouTube.

### ***Сайт для мероприятия или промостраница***

Приглашения с анкетой для гостя нужны для выставок, конференций, лекций, фестивалей, церемоний вручения премий, спектаклей, вебинаров. Если планируется важное событие и нужно подготовить сайт с описанием, программой, расписанием, контактами – Тильда предоставит для этого все необходимые инструменты.

### ***Сайт обучающего проекта***

Используйте Тильду, если:

- для продажи курса нужна посадочная страница;
- вы хотите собрать всю учебную программу на сайте;
- вам нужно анонсировать старт образовательного марафона или набора на курс.

Добавьте на посадочную страницу описание программы, контакты, отзывы, тарифы с разными вариантами оплаты. Tilda дает возможность разместить все видео-уроки с YouTube-канала на сайте и собирать обратную связь от учеников.

### ***Плюсы и минусы Тильды***

#### Преимущества Тильды:

- недорогой способ разработки сайта;
- в функциях можно разобраться самостоятельно и обновлять содержание страниц без помощи программиста;
- создание сайта не занимает много времени, а лендинг можно разработать за один день;
- блоки легко адаптируются под разные версии сайта;
- с помощью Zero-блоков можно воплотить любую дизайнерскую идею, подчеркнуть индивидуальность;
- техподдержка в любой момент поможет разобраться в сложностях.

Видео, инструктажи на сайте и блоги содержат всю необходимую информацию по работе с функционалом и всегда доступны. Также специалисты Тильды регулярно добавляют ответы на все часто задаваемые вопросы.

### Недостатки Тильды:

- для несложного, не многостраничного сайта цена хостинга может быть неоправданно велика, а для интернет-магазина в Тильде есть только стандартные функции и интеграции. Задумайтесь, возможно, вам будет выгоднее обратиться к профессиональным разработчикам;
- если у вас нет хотя бы базовых дизайнерских навыков, а готовых блоков недостаточно, придется дополнительно нанимать стороннего специалиста или находить дизайнера через Тильду;
- места на хостинге иногда недостаточно. Картинки, видеоролики и другие элементы можно будет встроить в сайт через код, по-другому Тильда не сможет их загрузить из-за размера и внутренних ограничений.

### *Создание страниц сайта на Tilda*

Страницы на Тильде собирают из блоков с разным дизайном и интерфейсом. Блоки уже адаптированы под просмотр через мобильные устройства.

Чтобы создать страницу:

- зарегистрируйтесь и выберите тарифный план;
- нажмите на «Создать страницу»;
- выберите шаблон: Бизнес, Магазин, Событие, Контакты, Анкета;
- отредактируйте дизайн по своему вкусу или обратитесь к дизайнеру;
- добавьте контент;
- подключите интеграцию с социальными сетями и сервисами для работы кнопок и форм.

Чтобы редактировать конкретный блок, наведите на него указатель мыши и нажмите кнопку «Настройки» или «Контент». У каждого блока есть свои варианты для редактирования, чаще всего это: цвет, размер, диапазон видимости.

Перед публикацией задайте основные настройки страницы:

- заголовок;
- описание;
- адрес.

После редактирования внешнего вида и контента нажмите кнопку «Опубликовать» и проверьте все данные с разных устройств.



## ***Работа с внутренними страницами сайта***

Лучше использовать одинаковые шаблоны, чтобы посетители не отвлекались на разные стили в дизайне. Для этого добавляйте в шаблон нужную функцию. Каждую страницу адаптируйте под собственную задачу. Например, для портфолио добавьте галерею, а для описания продукта – кнопку «Купить».

## ***Создание страницы на Tilda с чистого листа***

Если ни один из шаблонов не подходит или проект слишком специфический, можно создать сайт с чистого листа. Для этого нажмите на кнопку «Создать страницу» и выберите «Пустая страница».

В библиотеке блоков с левой стороны представлены все возможные варианты. Задайте собственную последовательность и выберите необходимые блоки для своего проекта.

## ***Возможности Tilda***

Практически для любого проекта подойдут возможности Tilda. Рассмотрим самые основные.

### ***Дизайн***

Дизайн Tilda состоит из:

- готовых шаблонов: интернет-магазин, блог, событие, контактная информация;
- готовых блоков: меню, заголовок, преимущества, галерея, подвал;
- шрифтов и изображений;
- zero-блоков.

Дизайн сайта адаптируйте под себя: редактируйте готовые шаблоны или используйте zero-блоки для разработки с нуля. Изучите каталог экспертов Tilda Experts: там около тысячи дизайнеров, которые готовы создать для вас уникальный проект.

### ***Теги страниц***

В настройках страниц обязательно пропишите теги (ключевые слова), разные для каждой страницы (рис. 16).



Рис. 16. Добавление ключевых слов на страницу в Тильде

## Конструктор сайтов UCoz

Конструктор сайтов UCoz дает возможность разрабатывать современные сайты, а затем успешно заниматься их раскруткой и продвижением. У этого сервиса простой в освоении и интуитивно понятный интерфейс, при этом он включает в себя развитую систему аналитики.

### *Как создать сайт?*

С помощью данной инструкции каждый новичок, не разбирающийся в особенностях веб-разработок, сможет быстро, легко и бесплатно создать собственный сайт в системе uCoz. Для этого нужно выполнить три простых шага.

### *Шаг 1. Регистрация в системе*

Чтобы создавать сайты в системе uCoz, необходимо зарегистрироваться. Регистрация бесплатна, и после ее прохождения вы сможете создавать неограниченное количество сайтов.

- Перейдите на страницу: [uCoz.ru/register/](http://uCoz.ru/register/).
- Выберите удобный способ регистрации: через ввод e-mail и пароля или через социальные сети. Указывайте действующий e-mail, потому что на него придет письмо с подтверждением аккаунта.

После того как введен действующий адрес e-mail и придуман сложный пароль (или вы авторизовались с помощью социальных сетей), нажмите на кнопку «Продолжить» и Вы попадете на страницу под названием «Создание нового сайта». Нужно заполнить все пустые поля и, поставив внизу галочку, согласиться с условиями использования uID сообщества. Затем нужно нажать на кнопку «Регистрация».

Если все поля были заполнены корректно, откроется страница с информацией об успешной регистрации, а на указанный e-mail будет отправлено письмо. Теперь необходимо проверить свой почтовый ящик.

Найдите письмо с темой «uID – Добро пожаловать в наше сообщество!». Обратите внимание на то, что оно может попасть в папку «Спам». В полученном письме необходимо кликнуть на кнопку «Продолжить».

### ***Шаг 2. Настройка безопасности аккаунта***

Выполнив все действия, указанные выше, вы попадете на страницу с текстом «E-mail подтвержден!». Здесь вам необходимо ввести номер мобильного телефона и задать секретный вопрос для безопасной работы с сайтом (например, сайт или модуль нельзя будет впоследствии удалить, не ответив на секретный вопрос). Также вам нужно придумать ещё один пароль (**Пароль администратора**): он будет использоваться для входа в панель администратора (ее адрес [http://ваш\\_сайт/admin](http://ваш_сайт/admin)) на нашем будущем сайте. Пароль, придуманный нами чуть ранее, относился к uID профилю.

После того как Вы придумали ответ на секретный вопрос и задали пароль для панели администратора, нажмите на кнопку «Приступить к созданию». Затем Вы попадете на страницу, где необходимо выбрать адрес сайта. Именно по данному адресу посетители будут видеть сайт в Интернете. Придумайте свободный адрес сайта, наилучшим образом описывающий ваш проект, выберите для него доменную зону (ucoz.net, ucoz.org и т. д.) и нажмите на кнопку «Создать сайт» (рис. 17).

Рис. 17. Создание нового сайта в конструкторе UCoz

### ***Шаг 3. Создание и первичная настройка сайта***

Успешно выполнив все инструкции, которые написаны чуть выше, вы окажетесь на странице настроек конфигурации сайта.

Укажите название сайта, выберите его дизайн и язык. Если вы сомневаетесь в выборе, нажмите на кнопку «Продолжить», позднее вы сможете изменить эти настройки в любой момент.

Далее перед вами появится страница с выбором модулей, которые вы хотите использовать на сайте. Нужно поставить галочки напротив тех модулей, которые вам необходимы. Если вы не уверены в том, какой именно модуль вам нужен, посмотрите инструкцию по выбору модулей или нажмите «Продолжить». После создания сайта вы сможете отключить или добавить на сайт любой модуль системы uCoz.

После завершения основных настроек конфигурации сайта вы автоматически будете перенаправлены в панель управления вашим проектом (рис. 18).

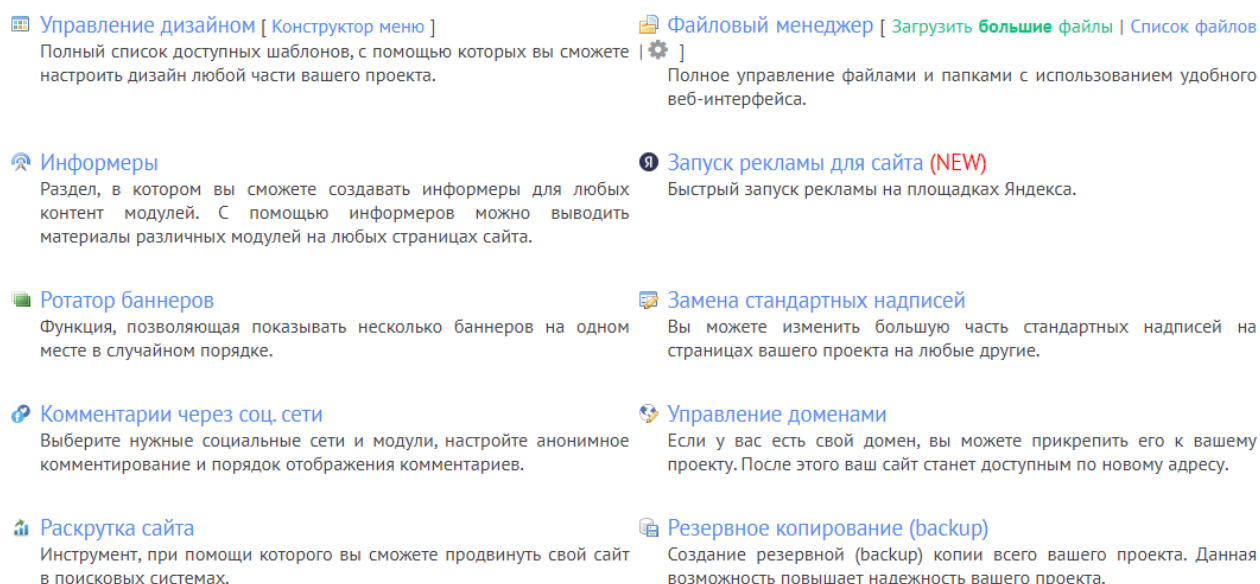


Рис. 18. Панель управления проектом в конструкторе UCoz

Обязательно сохраните данные, которые вам потребуются при работе над сайтом:

- адрес сайта и ссылка для входа в панель управления ([http://ваш\\_сайт/admin](http://ваш_сайт/admin));
- пароль для входа в панель управления (Пароль от ПУ, не путайте его с uID);
- ответ на секретный вопрос.

## Конструктор сайтов Fo.ru

Fo.ru – это бесплатная платформа, которая предоставляет любому пользователю возможность самостоятельно создавать сайты на основе доступных шаблонов (рис. 19). Возможности этого онлайн конструктора сайтов позволяют создавать качественные проекты без знания тонкостей HTML-верстки.

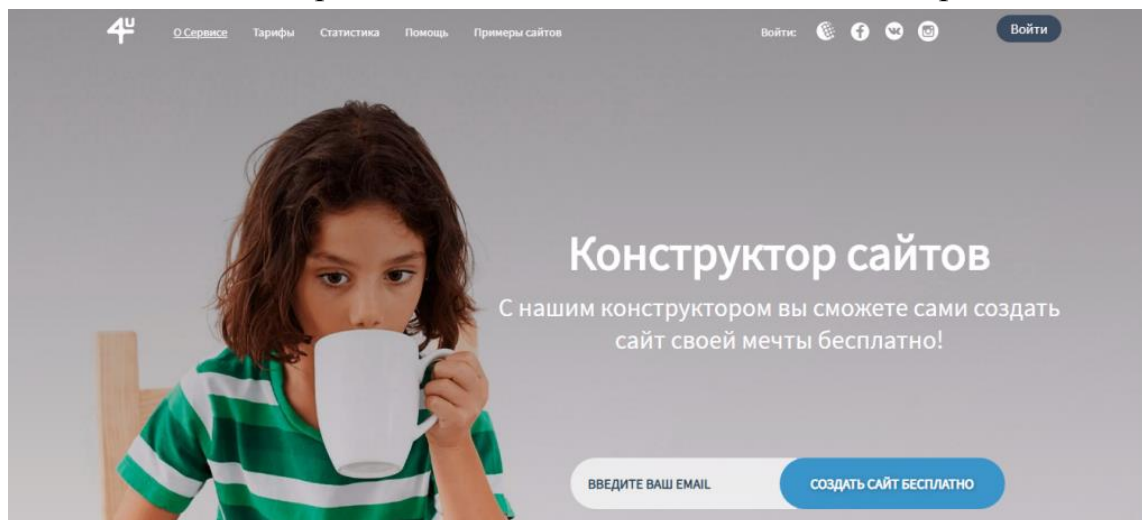


Рис. 19. Внешний вид конструктора Fo.ru

Fo.ru предназначен не только для создания сайтов (блогов, интернет-магазинов, лендинг пейдж), но и для их оптимизации. Пользователи этого сервиса получают возможность подключения собственного домена, редактирования мета-данных, создания robot.txt, размещения счетчиков и т.д. Сервис Fo.ru также предоставляет возможности для маркетинга сайтов.

Для того чтобы создать сайт посредством возможностей этого сервиса, необходимо зарегистрировать аккаунт на Fo.ru. Процедура регистрации очень простая и занимает несколько секунд. Пользователю необходимо только лишь указать адрес своей электронной почты. После этого на e-mail придет письмо со случайно сгенерированным паролем и требованием подтвердить регистрацию.

После завершения регистрации пользователь сможет приступить к созданию сайта.

Конструктор сайтов Fo.ru позволяет создать:

- интернет-магазин;
- блог;
- лендинг пейдж;
- сайт-визитку;
- сайт для продвижения бизнеса;
- сайт, посвященный любой тематике.

После выбора подходящего варианта пользователь окажется на странице с доступными готовыми шаблонами (см. рис. 20).

## Готовые шаблоны

Выберите подходящий шаблон и отредактируйте его или используйте новые возможности нашего сайта

[МАРКЕТ](#)

[КЛОНИРОВАТЬ САЙТ \(БЕТА\)](#)

[САЙТ ИЗ VK/ INSTAGRAM](#)

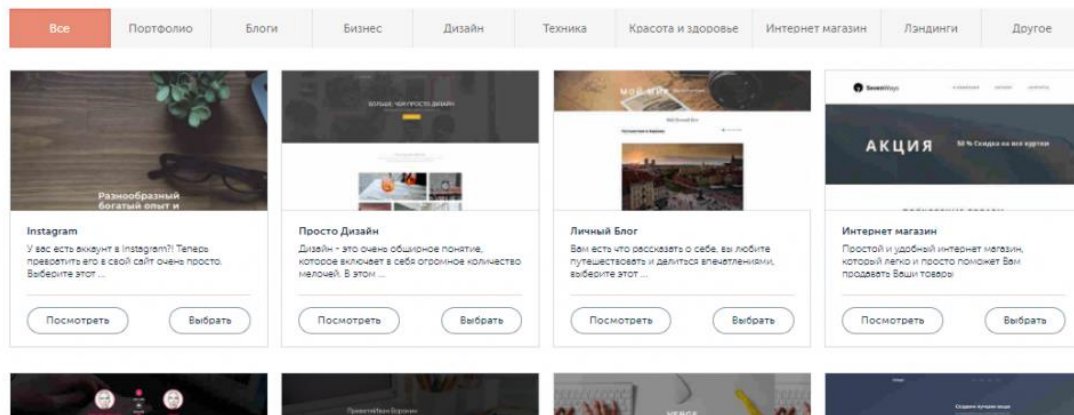


Рис. 20. Шаблоны для создания сайта в конструкторе Fo.ru

### *Шаблоны*

Этот бесплатный конструктор сайтов предоставляет доступ к десяткам шаблонов для создания сайтов любой направленности. У пользователей данного сервиса есть возможность предварительно просмотреть как будет выглядеть тот или иной шаблон путем нажатия на кнопку «Посмотреть».

Каждый из представленных шаблонов имеет описание, благодаря которому выбор пользователя существенно упрощается. Также все шаблоны разделены по разным категориям.

Для того чтобы использовать понравившийся шаблон, пользователю необходимо нажать на кнопку «Выбрать». В результате этого действия появится окно редактора, в котором будет доступна возможность настройки основных конфигураций сайта. Пользователь сможет выбрать подходящий стиль блока, шрифт, фон и анимацию текста.

Во вкладке контент разработчик сайта сможет загрузить изображения, а также настроить работу кнопок и ссылок. В этой вкладке также доступно редактирование текстового содержимого сайта.

### *Возможности*

Главной задачей этого сайта является обеспечение доступа к большому количеству инструментов для создания уникального сайта на html. Для реализации этой задачи Fo.ru предоставляет такие возможности:

- покупка доменов. Разработчики могут не только бесплатно создать свой сайт, но и подключить к нему домен с любым понравившимся именем. Стоит обратить внимание на то, что подключение доменов – это платная услуга;
- адаптивный дизайн. Этот конструктор сайта визитки автоматически адаптирует созданный ресурс под требования всех используемых сегодня платформ: ПК, смартфонов и планшетов;
- SEO-оптимизация. Сайты, разработанные на этой платформе, видны поисковикам с первой публикации. В случае необходимости для обеспечения SEO-оптимизации разработчики сайта могут добавить ключевые слова и фразы для поисковых систем;
- клонирование сайта. Эта услуга Fo.ru является в своем роде уникальной. Благодаря этому сервису пользователь может клонировать дизайн любого сайта;
- создание интернет-магазина. Этот конструктор сайтов для интернет-магазинов позволяет быстро настроить красивую онлайн-витрину и подключить все необходимые модули для совершения платежей в интернете. Пользователи этого сервиса могут подключить к своему интернет-магазину несколько способов оплаты: банковскими картами, биткоинами или посредством платежной системы WebMoney.

### **Задания для самостоятельной работы к главе 3**

#### **Задание по теме «Сайты Google»**

1. Подберите материал для создания трех-четырех страниц нового сайта, описывающего произвольную страну (например, общие сведения о стране, столица, язык, форма правления, достопримечательности).
2. Создайте сайт, состоящий из 1 главной и трех второстепенных страниц. На каждой странице разместите 10-15 строк текста.
3. Настройте *дизайн сайта* (тема, цвета и шрифты). На главной странице поместите панель навигации. На страницу «Столица» поместите карту Google с изображением этого города.
4. Добавьте на все страницы две-три фотографии и на одну страницу – произвольное видео.
5. Покажите созданный сайт преподавателю.

#### **Задание для работы с конструктором Tilda**

1. Создайте одностраничный сайт фирмы по продаже недвижимости «Кошкин дом» (см. рисунок в конце документа).
2. Для этого перейдите по ссылке <http://tilda.cc/ru/>. Изучите возможности конструктора (кнопка «Обзор»).

3. Нажмите на кнопку «Регистрация» в правом верхнем углу экрана. Заполните форму и нажмите кнопку «Зарегистрироваться».

4. Введите название проекта.

5. Создайте для будущего сайта новую страницу – кнопка «Создать новую страницу» справа.

6. Выберите подходящий шаблон.

7. Спроектируйте страницу, добавляя на нее новые блоки.

8. Для изменения страницы нажмите слева вверху кнопку «Контент». Исправьте заголовок и описание страницы, добавьте фоновую картинку.

9. Нажмите на кнопку «Сохранить и закрыть», просмотрите полученные изменения.

10. Покажите результат работы преподавателю.

11. Подберите информацию и создайте многостраничный сайт (не менее 3 страниц) по продаже недвижимости. Многостраничный сайт должен быть расширением одностраничного. Например, создайте отдельные страницы для некоторых объектов недвижимости.

12. Обязательно создайте меню для сайта. Добавьте на главную страницу блок «Меню» и настройте пункты меню для переходов на второстепенные страницы.

13. Покажите многостраничный сайт преподавателю.

### **Задание для работы с конструктором UCoz**

1. Перейдите по ссылке <http://www.ucoz.ru/register> и зарегистрируйтесь на сайте Ucoz.

2. Придумайте адрес сайта (англ. буквы) и создайте его. Тематика сайта – агентство «Универ» по подбору работы для студентов.

**Создание нового сайта**

Осталось выбрать запоминающийся адрес

Ваш сайт ucoz.net

Длина адреса от 2 до 30 символов

Согласен с Лицензионным соглашением, Договором о хранении учетных данных Держателя Карты и политикой конфиденциальности

Необходимо ваше согласие с правилами хостинга

**Создать сайт**

В любой момент вы сможете прикрепить к своему сайту домен вида "site.ru" (.com, .net или любой другой домен второго уровня).

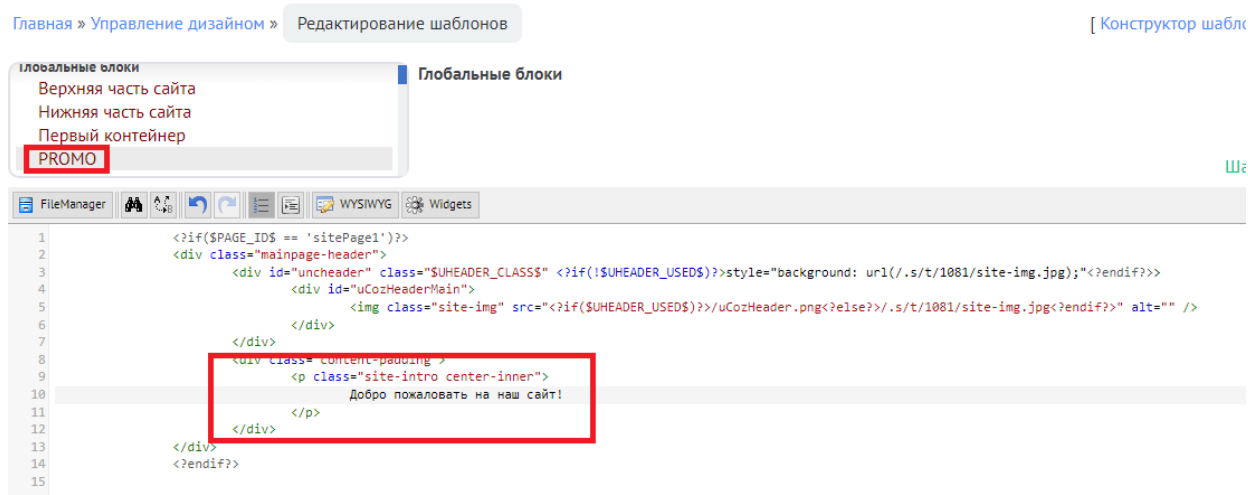


3. Выберите тип сайта – «Лендинг/визитка». Введите название – «Универ 2023». Нажмите кнопку «Создать сайт». Затем вы увидите «Панель управления», которую нужно закрыть.

4. Откройте в новой вкладке созданный сайт и изучите его.

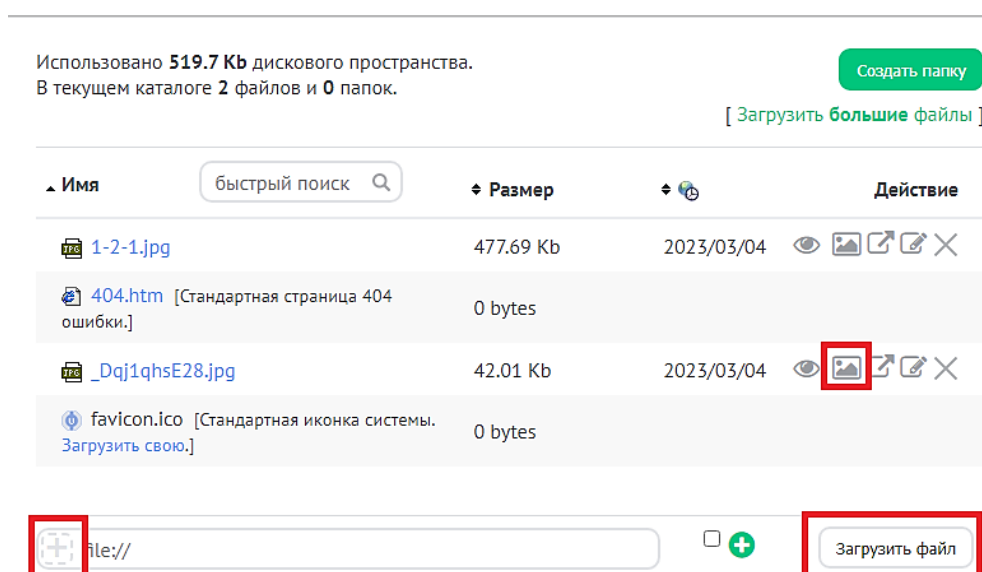
5. Для редактирования перейдите в раздел «Управление дизайном», далее в разделе "Редактор страниц" щелкните по ссылке «Страницы сайта».

6. Выберите раздел «Промо», в HTML-редакторе исправьте английский текст приветствия на русский (3–4 предложения об агентстве и его работе).



7. Чтобы сменить картинку на главной странице, выберите небольшое изображение и загрузите его в файловую систему. Для этого щёлкните по кнопке «Файловый менеджер» в правом нижнем углу.

8. Затем – по кнопке «+» в появившемся окне, после чего выберите картинку двойным щелчком. Нажмите на кнопку «Загрузить файл» и дождитесь загрузки.



9. Далее найдите картинку в списке загруженных файлов и нажмите на значок "HTML-код картинки". Скопируйте получившийся код.

10. Закройте окно загрузки и вставьте скопированный код вместо соответствующего тэга на странице «ПРОМО».

11. Сохраните изменения, перейдите на вкладку с сайтом и нажмите «Обновить».

12. Перейдите в «Редактор страниц» и щёлкните по ссылке «Управление страницами сайта».

13. Отредактируйте страницу «Информация о сайте» с помощью редактора (значок «глаз»).

14. Создайте новую страницу «Вакансии» с 1-2 вакансиями в произвольном формате.

15. Проверьте, появилась ли новая страница в меню.

16. Добавьте страницу «Сотрудники», вставьте 1-2 фотографии сотрудников.

17. Покажите результаты работы преподавателю.

### **Задание для работы с конструктором Fo.ru**

1. Перейдите по ссылке fo.ru и нажмите на кнопку «Создать сайт бесплатно».

2. На странице «Вход» щёлкните по ссылке «Зарегистрироваться».

3. После регистрации перейдите в раздел «Тарифы» и подключите тариф «Базовый».

4. Затем перейдите в раздел «Мои сайты» и нажмите кнопку «Создать сайт».


5. Выберите шаблон «Доставка выпечки», будем создавать сайт магазина-пекарни «Колобок».

6. Отредактируйте заголовок и фоновое изображение, используя кнопку «Контент».


7. Измените блок с доставкой, смените город и стоимость доставки. Отредактируйте время приема заказов и время приготовления.

8. Отредактируйте слайдер с картинками (Контент, Слайдер, Управление слайдером). Оставьте 6 слайдов, разместите там изображения ватрушек, круасанов, пирожков, сэндвичей, роллов и кап-кейков. Зайдите в пункт «Настройки» и поменяйте шрифт блока, добавьте анимацию изображений на свой выбор.

9. После слайдера добавьте на страницу новый блок – прайс. Для этого

щёлкните по значку , а затем на панели слева выберите пункт «Прайс».

10. Опишите стоимость тех товаров, изображения которых были добавлены на прайс. Для добавления новой карточки зайдите в пункт «Контент», «Динамические карточки» и щелкните по кнопке внизу – «Добавить карточку».

11. Просмотрите созданный сайт с помощью кнопки  в верхней части страницы.

12. Переделайте раздел «Успейте заказать» в «Форму заказа»: добавьте поля для имени, телефона и адреса заказчика, поле «Отзыв» переделайте в «Заказ».

13. Покажите выполненную работу преподавателю.

## ГЛАВА 4. СОЗДАНИЕ САЙТА С ПОМОЩЬЮ CMS

CMS – это программное обеспечение, которое позволяет управлять контентом сайта: создавать и публиковать записи, размещать виджеты, менять дизайн, редактировать различные элементы, добавлять функционал и т. д. Систему управления контентом сайта также называют движком.

CMS обычно состоит из двух основных компонентов:

- приложения для управления контентом (CMA) в качестве внешнего пользовательского интерфейса, позволяющего пользователю добавлять, изменять и удалять контент с веб-сайта без вмешательства веб-мастера,
- приложение доставки контента (CDA), которое компилирует контент и обновляет веб-сайт.

### *Назначение CMS*

Назначение CMS — упрощение процесса создания и управления сайтом. Чтобы добавить новый раздел, опубликовать новость или поменять дизайн, не нужно нанимать целый штат веб-разработчиков. Достаточно иметь одного человека, который разбирается в создании сайтов. Ему даже не нужны специальные знания и навыки программирования, так как движки позволяют взаимодействовать с ресурсом при помощи простого интерфейса и визуального редактора WYSIWYG (what you see is what you get — что видишь, то и получишь).

Все, что формирует сайт как таковой, создается и редактируется с помощью системы управления CMS. Конечно, сверстать веб-сайт можно и вручную. Но тогда вам понадобится действительно профессиональный программист. Сам процесс создания ресурса займет значительно больше времени, а реализовать некоторые отдельные функции (которые легко добавляются в CMS) будет либо затруднительно, либо невозможно.

### *Основные функции CMS:*

- предоставление инструментов для создания содержимого, организация совместной работы над содержимым;
- управление содержимым: хранение, контроль версий, соблюдение режима доступа, управление потоком документов;
- публикация содержимого;
- представление информации в виде, удобном для навигации, поиска.

В системе управления содержимым могут находиться самые различные данные: документы, фильмы, фотографии, номера телефонов, научные данные и т.д.

Такая система часто используется для хранения, управления, пересмотра и публикации документации.

Контроль версий является одной из важных возможностей, когда содержимое изменяется группой лиц.

### ***Какие бывают CMS. Популярные системы***

#### *WordPress*

Бесплатная свободно распространяемая система управления содержимым сайта с открытым исходным кодом, которая изначально позволяла создавать только сайты для блогов. Но на сегодня ее функционал расширен. Теперь он включает CMS для Wiki, интернет-магазинов, корпоративных сайтов, блогов, информационных порталов.

WordPress – самая распространенная на сегодняшний день CMS. По данным независимого агентства Web Technology Surveys за март 2022 г., около 64,8 % всех сайтов в мире использует CMS WordPress. Вот главные особенности ее успеха:

- бесплатный «движок» – CMS WordPress распространяется по открытому лицензионному соглашению (GNU GPL), так что его можно свободно использовать даже в коммерческих целях;
- неограниченные возможности – с помощью WordPress можно создать интернет-магазин, личный блог, корпоративный сайт, информационный портал, отраслевой ресурс, галерею мультимедиа;
- гибкая настройка внешнего вида и функциональности – доступны платные и бесплатные шаблоны, с помощью плагинов можно решать технические задачи, обеспечивать необходимую функциональность;
- простота администрирования – чтобы работать с WordPress, не нужны специальные знания, все интуитивно понятно. Пример администраторской панели приведен на рис. 21;
- многие компании предоставляют услугу автоматической установки WordPress на хостинг – не нужно самому искать на официальном сайте дистрибутив, скачивать его и заниматься установкой. Например, Beget предлагает установить на выбор WordPress или Joomla. Для установки CMS достаточно пары кликов.

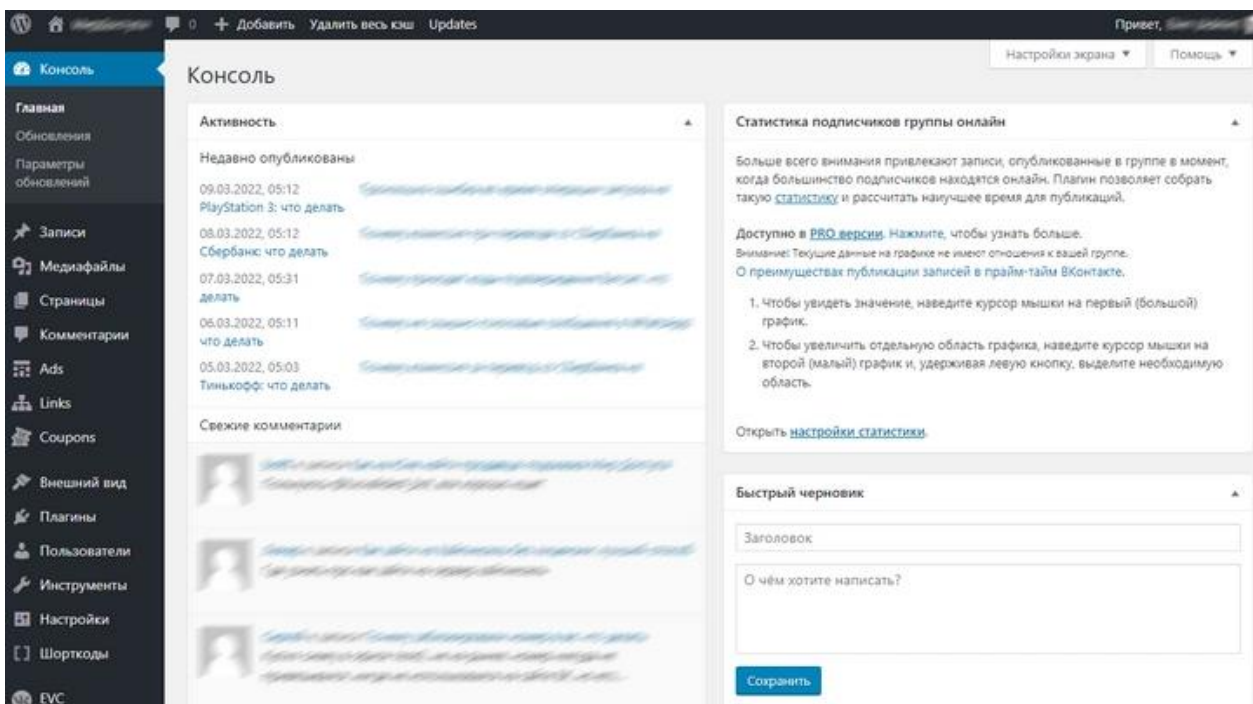


Рис. 21. Администраторская панель в WordPress

### «1С-Битрикс»

Данный движок больше предназначен для «тяжелых» коммерческих сайтов. Если сравнивать с WP, функционал в области коммерции шире. Но использование 1С не является бесплатным. Зато платформа хорошо справляется со сложными задачами, и благодаря чему она часто используется владельцами интернет-магазинов.

### Drupal

Данный движок является бесплатным и при этом полнофункциональным. Способен решать сложные задачи наряду с простыми.

### Joomla

Бесплатный движок Joomla используют порядка 11% сайтов в Рунете. Система популярна не только в странах СНГ, но и во всем мире и входит в ТОП-5 лидирующих CMS.

Как и WordPress, Joomla! подходит для всех типов сайтов. При этом не имеет значение масштаб: система одинаково оптимальна для небольших страниц и крупных площадок.

### *Безопасность и производительность Joomla*

На базовом уровне движок в достаточной мере защищен от несанкционированного доступа. Новые инструменты, которые повышают безопасность, появляются с каждым обновлением.

Однако, как и в случае с другими бесплатными CMS, установка сторонних расширений снижает уровень защищенности, поэтому статистика взломов неутешительна.

Также бывают проблемы с производительностью, когда Joomla! используют для крупных проектов.

### *Удобство использования Joomla*

Чтобы разобраться, как работает CMS Joomla, не нужно долгое время разбираться в возможностях системы и вырабатывать специальные навыки. Установка и настройка системы не вызывают проблем даже у новичков, поэтому движок считается достойной альтернативой WordPress.

### *Системные требования Joomla:*

- PHP 4.2.x или выше;
- MySQL 3.23.x или выше;
- Apache 1.13.19 или выше.

В качестве Web-сервера может использоваться также MS IIS (Internet Information Server) версии 5.0 и выше. Необходимо удостовериться в том, что используемая версия PHP имеет встроенную поддержку MySQL, XML и Zlib.

Joomla может использоваться совместно со всеми наиболее известными браузерами, а именно: Internet Explorer (версия 5.5+), Netscape, и Mozilla. Именно с ними секция администрирования наиболее совместима.

Некоторые WYSIWYG-редакторы, например htmlArea, работают только с Internet Explorer.

Базовая версия Joomla! отлично справляется с управлением сайта и содержит все необходимые функции для создания качественного сайта.

Для многих людей основная сила Joomla! заключается в возможности создания прикладных дополнений и расширений к базовой версии системы.

Тысячи разработчиков со всего мира создают универсальные расширения (компоненты, модули, мамботы) специально для Joomla! Доступны следующие расширения:

- динамические генераторы форм;
- бизнес или организационные каталоги;
- управление документами;
- галереи изображений и мультимедиа;
- системы электронной коммерции;
- форумы и чаты;
- календари;

- блоги;
- почтовые рассылки.

### Модули Joomla!

Модули расширяют возможности Joomla! и дают программам новую функциональность. Модули – небольшие объекты содержимого сайта, которые могут быть отображены в любом месте шаблона, если это будет разрешено.

Модули очень легко устанавливаются в «Панели управления» (рис. 22).

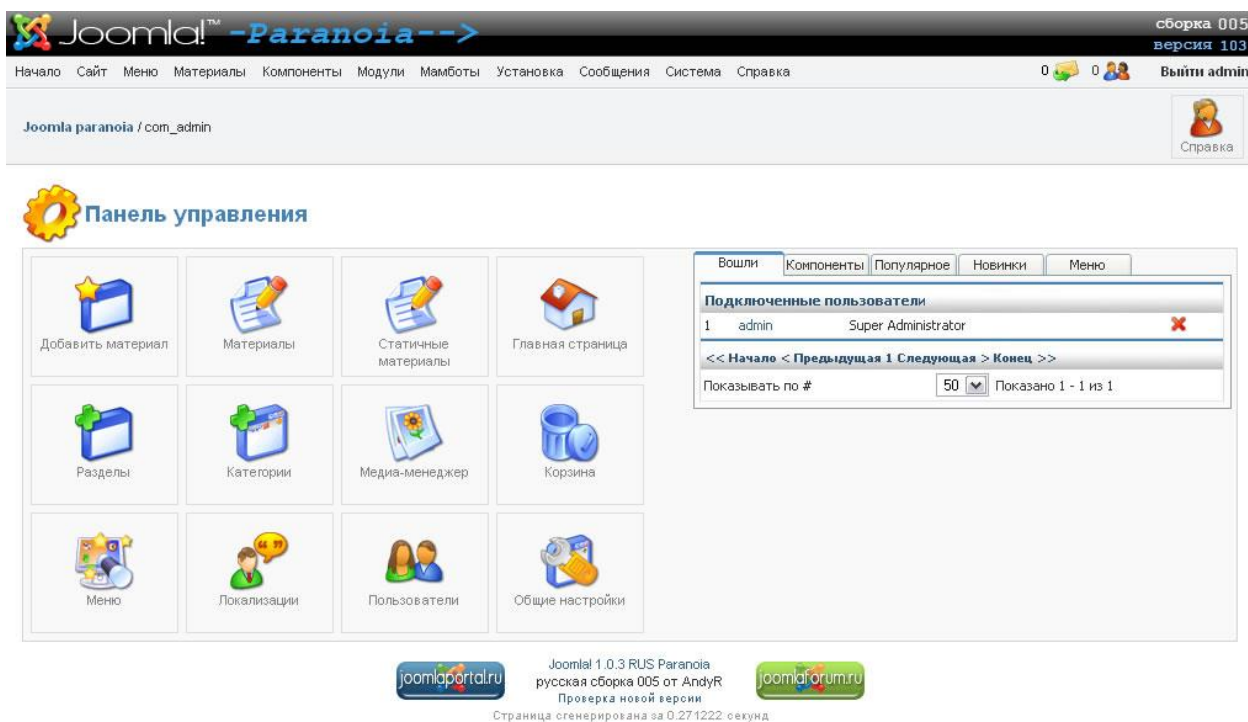


Рис. 22. Панель управления в CMS Joomla

Обычно Joomla! уже включает в себя несколько модулей:

- Главное меню (Main Menu),
- Верхнее меню (Top menu),
- Выбор шаблона (Template Chooser),
- Опросы (Polls),
- Счетчик нажатий и т.д.



## Задания для самостоятельной работы к главе 4

### Задание 1

1. Запустите *Denwer* (ярлык **Start Denwer** на рабочем столе).
2. Запустите любой браузер, перейдите по ссылке <http://localhost/mysite.ru/administrator>
3. Введите в панели администрирования логин и пароль – **admin**. Выберите язык – *русский*.
4. Просмотрите установленный шаблон сайта – ссылка *Мой сайт* в правом верхнем углу.
5. Загрузите и установите новый шаблон для Вашего сайта (*шаблон нужно взять у преподавателя*). Для загрузки зайдите в раздел *Расширения, Расширения, Установка*.
6. Проверьте его установку, изучите главную страницу сайта.
7. Установите еще два-три шаблона. Из всех установленных шаблонов выберите тот, который Вам нравится больше всего и установите его по умолчанию для всех страниц.
8. Определите на главной странице позицию, куда можно вставить баннер. *Баннер нужно взять у преподавателя*.
9. Чтобы просмотреть позиции Вашего сайта, необходимо к его адресу добавить значение «*?tp=1*» и перейти по указанному адресу. Например, *site.ru/?tp=1*.
10. Выберите и установите баннер для своего сайта (<https://sitedelkin.ru/kak-ustanovit-banner-na-joomla.html>). Обратите внимание на размеры баннера.
11. Отредактируйте главную страницу (<https://joomla-abc.ru/uroki/glavnaya-stranitsa-sajta-joomla-vse-nastrojki-glavnoj-stranitsy>). Добавьте произвольное изображение и два абзаца текста об агентстве по продаже недвижимости «Кошкин дом».
12. Покажите выполненную работу преподавателю.

### Задание 2

1. Запустите **Denwer** (ярлык *Start Denwer* на рабочем столе).
2. Запустите любой браузер, перейдите по ссылке <http://localhost/mysiteN.ru/administrator>.
3. Введите в панели администрирования логин и пароль – **adminN**.
4. Создайте и настройте отображение статической главной страницы (<https://joomla-abc.ru/uroki/glavnaya-stranitsa-sajta-joomla-vse-nastrojki-glavnoj->

stranitsy). Добавьте произвольное изображение и один-два абзаца текста об агентстве по продаже недвижимости «Кошкин дом».

5. Создайте 2 категории материалов: Продажа, Покупка (как сделать – <http://fairheart.ru/joomla/osnovy-joomla/materialy-joomla.html>, <http://vsetyrabota.ru/uroki-joomla/175-urok-10-sozdanie-kategorii-v-joomla>).

6. Загрузите 6 изображений (размером не более 2 МБ) для следующих объектов: *Лавочка, Киоск, Памятник, Остановка, Будка, Стадион*.

7. Создайте материалы для каждого объекта из упоминаемых в п.6 (10-15 слов текста + изображение). Добавьте категорию «*Продажа*».

8. Создайте вертикальное меню *Продаем*, добавив туда ссылки на все созданные материалы (подробнее – см. <http://vsetyrabota.ru/uroki-joomla/176-urok-11-sozdanie-menyu-v-joomla-3>).

9. Самостоятельно создайте три материала с объектами, которые агентство желает купить. Для этих объектов создайте еще одно меню «*Приобретем*».

10. Покажите выполненную работу преподавателю.

### Задание 3

1. Запустите **Denwer** (ярлык *Start Denwer* на рабочем столе).

2. Запустите любой браузер, перейдите по ссылке <http://localhost/mysiteN.ru/administrator>.

3. Введите в панели администрирования логин и пароль – adminN.

4. Подберите 3-4 картинки на тему «Недвижимость» примерно одного размера.

5. Установите внешний модуль «STCarousel»: *Расширения, Расширения, Установка*. Загрузите модуль, не распаковывая архива.

6. Просмотрите страницу своего сайта с определением позиций: <http://localhost/mysiteN.ru/&tp=1>. Выберите место для карусели изображений и запомните его название.

7. Перейдите в раздел *Расширения, Модули*. Откройте только что установленный модуль. Перейдите в раздел *Изображения*. Загрузите по очереди отобранные изображения.

8. Перейдите в пункт *Привязка к пунктам меню*. Выберите вариант *На всех страницах*.

Сохранить Сохранить и закрыть Сохранить и создать Сохранить как копию Закрыть

Заголовок \* stcarousel

Модуль Привязка к пунктам меню **Изображения** Каталог Настройки Дополнительные параметры Права доступа

### stcarousel

Сайт

Карусель с навигацией в виде точек

Заголовок  
Показать Скрыть

Позиция  
Введите или выберите пози...

Состояние  
Не опубликовано

Начало публикации

Завершение публикации

9. Перейдите обратно в пункт *Модуль*. Выберите справа **Позицию** и установите переключатель **Состояние** в положение *Опубликовано*.

10. Обратите внимание на то, что в разделе *Настройки* можно задать дополнительные параметры отображения модуля.

11. Просмотрите свой сайт и покажите результат работы преподавателю.

12. Установите счетчик посещаемости на свой сайт (<https://www.liveinternet.ru/code>, <http://atempl.com/schetchiki>).

13. Для создания счетчика выберите позицию на сайте (где-нибудь внизу) и создайте новый модуль: *Расширения, Модули*, кнопка **Создать**.

14. Выберите тип модуля – *HTML-код*. Укажите заголовок *Счетчик*, в основное окно вставьте код из файла *Код\_счетчика*. В самом низу нажмите кнопку *Выключить редактор*.

15. Справа выберите позицию для счетчика и нажмите вверху на кнопку «Сохранить и закрыть». Просмотрите сайт на наличие счетчика.

16. Добавьте на сайт кнопки для социальных сетей, например, для перехода на vk.com (используйте генератор кнопок <https://inter-net.pro/btn-generator>). Если нет подключения к сети Интернет, то можно взять код из файла *Код\_кнопки*.

17. Задайте все параметры для кнопки и скопируйте ее html-код. Далее действуйте, как в ситуации со счетчиком, – создайте модуль для HTML-кода и т.д.

18. Покажите выполненную работу преподавателю.

## СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Информатика: базовый курс / С.В. Симонович и др. СПб.: Питер, 2001. 640 с.
2. Дригалкин В.В. HTML в примерах. Как создать свой Web-сайт. М.: Изд. дом «Вильямс», 2004. 192с.
3. Гид по Фигме для начинающих веб-дизайнеров. [Электронный ресурс]. URL: <https://tilda.education/articles-figma?ysclid=lbutnxbdwo113477946> (дата обращения: 01.04.2023).
4. Как создать сайт? База знаний UCoz. [Электронный ресурс]. URL: <https://www.ucoz.ru/help/start/kak-sozdat-svoj-sajt-v-ucoz> (дата обращения: 01.04.2023).
5. Как создать свой сайт бесплатно своими руками. URL: <https://www.internet-technologies.ru/how-to-create-a-website.html> (дата обращения: 01.04.2023)
6. Обзор конструктора сайтов Google Sites. [Электронный ресурс]. URL: <https://superbwebsitebuilders.com/ru/obzor-konstruktora-sajtov-google-sites/?ysclid=lc96hf28nu429524688> (дата обращения: 01.04.2023).
7. Обзор конструктора сайтов Fo.ru [Электронный ресурс]. URL: [https://club.cnews.ru/blogs/entry/import\\_obzor\\_konstruktora\\_sajtov\\_fo\\_ru\\_6e30?ysclid=lfe33a6698398124842](https://club.cnews.ru/blogs/entry/import_obzor_konstruktora_sajtov_fo_ru_6e30?ysclid=lfe33a6698398124842) (дата обращения: 01.04.2023).
8. Самоучитель HTML [Электронный ресурс]. URL: <http://htmlbook.ru/> (дата обращения: 01.04.2023).
9. Создание веб-сайта. Курс молодого бойца. [Электронный ресурс]. URL: <https://habr.com/ru/post/273795/> (дата обращения: 01.04.2023).
10. Тильда (конструктор сайтов): обзор возможностей и инструкция по настройке [Электронный ресурс]. URL: <https://www.calltouch.ru/blog/tilda-konstruktor-sajtov-obzor-vozmozhnostej-i-instrukciya-po-nastrojke/?ysclid=lfcoqg9xvs614522656> (дата обращения: 01.04.2023).
11. CMS (Content Management System) – Система управления контентом. [Электронный ресурс]. URL: <https://w-owl.com/cms-content-management-system-sistema-upravleniya-kontentom-dvizhok> (дата обращения: 01.04.2023).

*Учебное издание*

**Василюк Надежда Николаевна**

## **Создание и администрирование сайтов**

Учебное пособие

Редактор *Н. И. Стрекаловская*

Корректор *А. В. Цветкова*

Компьютерная верстка: *Н. Н. Василюк*

---

Объем данных 2,03 Мб

Подписано к использованию 09.11.2023

---

Размещено в открытом доступе

на сайте [www.psu.ru](http://www.psu.ru)

в разделе НАУКА / Электронные публикации  
и в электронной мультимедийной библиотеке ELiS

Управление издательской деятельности  
Пермского государственного  
национального исследовательского университета  
614068, г. Пермь, ул. Букирева, 15