



E-kursuse
Multimeedia II (veebidisain)
materjalid

Aine maht 3 EAP

Ljubov Feklistova, Lidia Feklistova (Tartu Ülikool), 2013

Содержание

НЕДЕЛЯ 1: ОСНОВЫ ВЕБ-ДИЗАЙНА.....	4
Предисловие.....	4
Интернет или WWW?.....	4
Что такое веб-дизайн?	6
Этапы веб-дизайна.....	7
Основные элементы	9
Принципы веб-дизайна	11
Избегай типичных ошибок!.....	12
Использованный материал	12
НЕДЕЛЯ 2: ОНЛАЙН РЕДАКТОР ВЕБ-СТРАНИЦ WEEBLY	14
Предисловие.....	14
Что такое Weebly?	14
Создание аккаунта	15
Первичные настройки. Панель меню	17
"Шапка" и макет веб-страницы.....	21
Элементы редактора.....	23
Основные (Basic)	24
Структура (Structure).....	27
Медиа (Media)	29
Коммерция (Commerce)	30
Дополнительно (More)	30
Сайт в WWW.....	33
НЕДЕЛЯ 3: ЯЗЫК РАЗМЕТКИ HTML (I).....	35
Предисловие.....	35
HTML язык.....	35
Структура HTML документа	37
Элементы HTML.....	39
Работа с текстом	40
Работа с изображениями	43
Работа с гиперссылками	44
НЕДЕЛЯ 4: ЯЗЫК РАЗМЕТКИ HTML (II).....	49
Предисловие.....	49
Фоновое оформление	49
Работа со списком.....	51
Работа с таблицей	53
Работа с формой.....	57
Обработка событий	59

Валидация.....	61
Справочники	61
НЕДЕЛЯ 5: КАСКАДНЫЕ ТАБЛИЦЫ СТИЛЕЙ CSS (I)	62
Предисловие.....	62
Каскадные таблицы стилей CSS	62
Подключение стилей.....	64
Синтаксис	66
Типы селекторов.....	67
Работа с текстом	69
Свойства шрифтов.....	75
Цвет.....	78
Что делать, если стили работают некорректно?	79
НЕДЕЛЯ 6: КАСКАДНЫЕ ТАБЛИЦЫ СТИЛЕЙ CSS (II).....	80
Предисловие.....	80
Работа со списками.....	80
Работа с границами.....	81
Поля и отступы	85
Поплавки	87
НЕДЕЛЯ 7: КАСКАДНЫЕ ТАБЛИЦЫ СТИЛЕЙ CSS (III)	89
Предисловие.....	89
Фон.....	89
Псевдоклассы	91
Псевдоэлементы	95
Позиционирование	97
Курсоры	99
Возможности CSS3.....	102

НЕДЕЛЯ 1: ОСНОВЫ ВЕБ-ДИЗАЙНА

Предисловие

Название книги: Основы веб-дизайна

Основные цели работы с книгой:

- выяснить различия между терминами "Интернет" и "Всемирная паутина - WWW";
- дать определение понятию "веб-дизайн";
- познакомиться с этапами и принципами веб-дизайна;
- рассмотреть возможности использования основных элементов.

Рассматриваемые темы:

- Интернет или WWW?
- Что такое веб-дизайн?
- Этапы работы.
- Основные элементы.
- Принципы веб-дизайна.
- Избегай типичных ошибок!

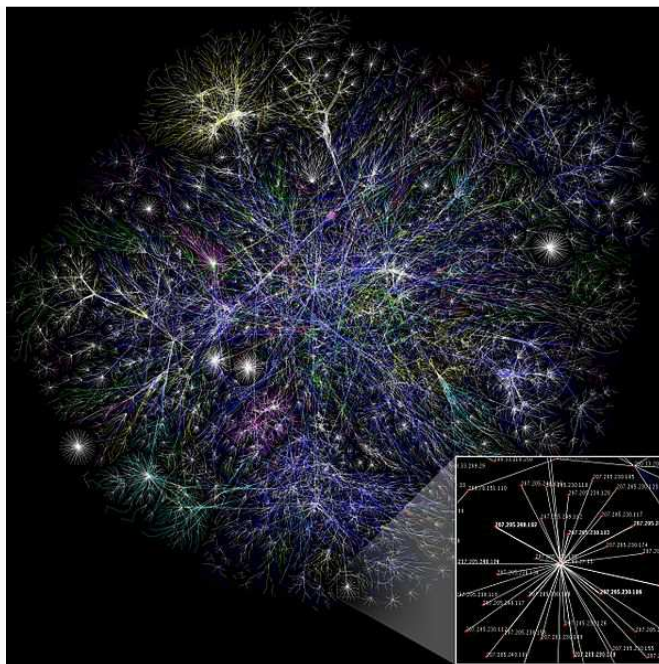
Результатом проработки данного материала является умение учащегося:

- разбираться и охарактеризовывать термины "Интернет", "Всемирная паутина", "веб-дизайн";
- выделять основные этапы работы по созданию веб-сайта;
- называть и кратко характеризовать основные элементы и принципы веб-дизайна;
- избегать характерных ошибок новичков, выбирать оптимальные варианты для решения поставленных задач.

Интернет или WWW?

Сегодня мы чаще всего ищем информацию в Интернете. Однако редко кто из нас задумывается о том, почему это стало возможным, как организован процесс обмена данными. Многие даже считают, что Интернет - это тоже самое, что и WWW. Так ли это на самом деле, зачем и когда появился Интернет, насколько он популярен и как передаётся информация в сети мы выясним на страницах данной главы.

Интернет - это соединение многомиллионного числа компьютеров с помощью кабелей в единую систему. Ниже представлен схематический вид малой части современного Интернета.



Источник:

http://commons.wikimedia.org/wiki/File:Internet_map_1024.jpg?uselang=ru

WWW (англ. яз. *World Wide Web*, Всемирная паутина) - это по сути файлы (текстовые, графические, мультимедийные...), которыми мы пользуемся, чтобы получить ответы на интересующие нас вопросы. Чтобы эта информация дошла до конечного пользователя и нужен Интернет. В большинстве своём ресурсы паутины представляют собой гипертекст (англ.яз. *hypertext*), который может содержать различного вида информацию и, конечно же, ссылки на другие страницы. Именно гипертекстовый документ принято называть веб-страницей (англ. яз. *webpage*). Для просмотра таких страниц нужна специальная программа - браузер (англ. яз. *browser*). Сегодня наибольшей популярностью пользуются следующие браузеры: *Google Chrome*, *Mozilla Firefox*, *Internet Explorer*, *Opera*. Для того чтобы гипертекст отображался корректно на любом компьютере, используется язык разметки HTML (англ. яз. HyperText Markup Language).

"Взаимосвязь между WWW и Интернет можно представить следующим образом: WWW – это информационный ресурс, в то время как Интернет - это средство передачи данных по сети. WWW - коллекция документов и программ для их просмотра, а Интернет - это система компьютеров, объединенных в сети с помощью кабелей и сигналов, опоясавших весь мир." [1]

Можно попытаться провести некоторую аналогию на примере оплаты покупки в магазине: у тебя на счету есть денежные накопления (WWW), но чтобы воспользоваться банковской карточкой для оплаты товара, необходим терминал (Интернет) для ввода PIN-кода.

В предложенном ниже видео (автор: Александр Фазлымов, https://www.youtube.com/watch?v=eT5mg2sMe08&feature=player_embedded) даётся краткий обзор истории Интернета и Всемирной паутины.

Пользователи Интернета

Количество пользователей Интернета постоянно растет. По данным мировой статистики Интернета [2] в июне 2012 года около 2,5 миллиардов человек пользовались услугами Интернета, то есть каждый третий житель планеты Земля. Составленный на основании

приведённых статистических данных рейтинг ТОП-5 стран, где проживают самые большие интернет-сообщества, выглядит следующим образом:

- Китай - 538 000 000 человек;
- США - 245 203 319 человек;
- Индия - 137 000 000 человек;
- Япония - 101 228 736 человек;
- Бразилия - 88 494 756 человек.

Стоит обратить внимание и на процентное соотношение количества пользователей Интернета к численности населения той или иной страны. Здесь ТОП-5 стран, составленных на конец 2011 года [3], состоит в большинстве своём из европейских государств:

- Исландия - 97,8% населения страны пользуется Интернетом;
- Норвегия - 97,2 %;
- Швеция - 92,9 %;
- Фолклендские острова - 92,4%;
- Люксембург - 91,4%.

Как устроен Интернет, каким образом происходит передача данных в сети между компьютерами говорится в предложенном далее видео (автор: Medialab, https://www.youtube.com/watch?feature=player_embedded&v=0O7iE1-PjMo).



- статья "Основные концепции Интернет";
- статья "Интернет";
- статья "Всемирная паутина".

Что такое веб-дизайн?

Если брать крайне узкое значение слова **"веб-дизайн"**, то можно ограничиться следующим - это проектирование структуры и внешнего вида веб-страницы. Веб-страницы, объединённые между собой единой тематикой, целью написания и связанные между собой гиперссылками, называются **веб-сайтами**. Все они создаются с целью донести до посетителя веб-страниц определённую информацию. Для достижения поставленных задач необходимо учитывать графическое оформление, удобство в ориентировании (навигации) и использовании сайта.

Сегодня в понятие **"веб-дизайн"** вкладывается более широкий смысл.

"Возможно, это не совсем верно, но на сегодняшний день, говоря «веб-дизайн», подразумевают не только профессиональное графическое оформление сайта. В этом понятии собрано несколько вполне самостоятельных направлений, которые и складываются в итоге в то, что мы называем «веб-дизайн». Это и графический интерфейс сайта, который мы видим сразу после того, как страница загрузилась в браузере, и проектирование системы навигации, и контент-проект сайта, то есть – его информационное наполнение, которое также должно учитываться при создании нового сайта. Это и загадочное «юзабилити» (*usability*), которому так и не подобрано до сих пор достаточно верного перевода с английского и обозначающее удобство использования, дружелюбность и понятность интерфейса, который мы видим на экране и многое другое... Веб-дизайн сегодня превратился в самостоятельное явление просто потому, что для занятия дизайном в Интернет недостаточно одного знания программ, предназначенных для работы с графикой и художественных способностей. Конечно, собственно программированием, особенно решением сложных задач, занимаются профессиональные

программисты. Но и работа дизайнера невозможна без чёткого представления технологий Интернета. Дизайнер должен чётко представлять, какими путями можно решить поставленную задачу или воплотить в действующую модель свои идеи, что влечёт за собой необходимость разбираться и в языках программирования, и в некоторых тонкостях серверных технологий. Он должен представлять себе, каким образом осуществляется передача данных в сети, как и в каком виде его разработки достигнут своего конечного пользователя – посетителя разрабатываемого сайта.

В конечном итоге, простая с первого взгляда задача «рисования» обрастает огромным количеством дополнительных требований, в основном – технического характера, имеющими с художественной стороной мало общего.

В итоге можно сказать, что веб-дизайн – это именно и есть создание сайта. Сочетание художественных способностей и широких знаний современных компьютерных технологий позволяют создавать сайты, которые запоминаются и заставляют посетителя возвращаться на них вновь." [4]

Таким образом, веб-дизайнер должен не только обладать художественным вкусом (выбор композиции, цветовой гаммы, шрифтов и т.д.), но и иметь представление о стандартах HTML и CSS (рассматриваются далее в данном курсе). От его работы зависит, насколько быстро посетитель получит нужную информацию, сориентируется на сайте, будет ли информация представлена атрактивно, какое впечатление на пользователя произведёт веб-сайт в целом и т.д.



На сайте BigIdeas опубликована [статья "Как определиться с качеством веб-дизайна: примеры и советы"](#), где приведены примеры и описания удавшихся веб-дизайнерских решений.



- [статья "История веб-дизайна"](#).

Этапы веб-дизайна

Результатом качественной разработки веб-страницы является доступность материала. Можно выделить **пять основных этапов**, которые сегодня относятся к веб-дизайну и помогают качественно решить поставленные задачи [5]. В рамках каждого из них имеется своя классификация, которая позволяет лучше понять суть веб-дизайна.

1. Назначение:

- *цель создания (функциональность) сайта* - распространение (рекламной) информации, интернет-продажи, сетевые игры...;
- *потенциальная аудитория* - для кого предназначен сайт;
- *объёмы и актуальность информации* - количество предлагаемой информации, её обновление и соответствие действительности.

2. Содержимое:

- *информация* - текст, знакомящий с деятельностью фирмы, описание товара, контактные данные... (конечно же составлено без грамматических и орфографических ошибок, материалы однозначно понятны для всех посетителей сайта);
- *вид информации* - текст, иллюстрации, видео или аудиоматериал, анимация (в меру, чтобы не раздражать внимание пользователя);
- *структура* - где, что, в каком порядке располагается на странице (сколько щелчков мышкой надо сделать, чтобы добраться до определённого пункта), удобство, логичность и лёгкость ориентирования на сайте и использования меню, оптимально состоящего из 5-7 пунктов (здесь на помощь приходит язык разметки HTML).

3. Зрительные образы:

- *работа с пространством* - сколько места выделить под ту или иную часть, определение местоположения основных элементов (обычно создаётся с помощью HTML и CSS);
- *выбор графических элементов* - визуальные образы кнопок, иконок, украшений, панели навигации... (также с помощью HTML и CSS).

4. Технология:

- *программное обеспечение* - выбор методов, приёмов, средств позволяющих реализовать задуманное (различные варианты HTML, CSS, JavaScript, PHP);
- *следование стандартам* для того, чтобы содержимое сайта отображалось правильным образом вне зависимости от используемых браузеров (программ для просмотра веб-страниц).

5. Доставка:

- *быстрота и качество* получения информации конечным пользователем - зависит от работы как создателя сайта, так и от возможностей клиента;
- *аппаратное обеспечение* - технические данные сервера, где будет храниться информация.



Образец общего макета работы над созданием веб-сайта для туроператора (выполнены первые три этапа).

- **Цель сайта** – сайт, знакомящий с возможными местами отдыха в Европе и Азии, имеющий в том числе онлайн-связь с консультантом, позволяющий при желании заказать путёвки;
- **Потенциальная аудитория** – отдыхающие средней возрастной группы (35 – 55 лет);
- **Объёмы и актуальность информации** – указывается описание тура (описание туристической программы, продолжительность, включённые экскурсии, цены, форма заказа, условия оплаты, рекламный видеоролик), обновляется перед каждым новым туристическим сезоном. Возможность оставить комментарии, мнения, добавить фотографии после возвращения из поездки. Наличие возможности связаться с туроператором (контактные данные).

Макет главной страницы

Логотип и слоган (текстура)		Поиск Иконка соц. сетей
Кнопка-меню 0 (Стартовая, Добро пожаловать, о фирме)	Общая информация • ¾ пространства • светло-оранжевый фон • тёмного цвета буквы • классический шрифт без засечек • размер чуть увеличенный • несколько ознакомительных иллюстраций	
Кнопка-меню 1 (Общее описание туров)		
Раскрывающаяся кнопка-меню 1.1 (Туры в Европу)		
Раскрывающаяся кнопка-меню 1.2 (Туры в Азию)		
Кнопка-меню 2 (Фотогалерея)		
Кнопка-меню 3 (Делимся впечатлениями)		
Кнопка-меню 4 (Спрашивайте – отвечаем!)		
Нижний колонтитул (текстура)		

Макет второстепенных страниц

Логотип и слоган (позволяет перейти на главную страницу, текстура)		Поиск Иконка соц. сетей
Название текущей страницы		
Кнопка-меню 0 (Стартовая, Добро пожаловать, о фирме)	Конкретная информация (текст, иллюстрации, видео-ролики, формы) • ¾ пространства • иконки-флаги для навигации • светло-оранжевый фон • тёмного цвета буквы • классический шрифт без засечек • размер чуть увеличенный	
Кнопка-меню 1 (Общее описание туров)		
Раскрывающаяся кнопка-меню 1.1 (Туры в Европу)		
Раскрывающаяся кнопка-меню 1.2 (Туры в Азию)		
Кнопка-меню 2 (Фотогалерея)		
Кнопка-меню 3 (Делимся впечатлениями)		
Кнопка-меню 4 (Спрашивайте – отвечаем!)		
Нижний колонтитул (текстура)		



- статья "Веб-дизайн";
- статья "Что необходимо учитывать при проектировании сайта".

Основные элементы

При оформлении страницы веб-дизайнеру чаще всего приходится работать со следующими элементами [6]:

- **пространство** - поле самой веб-страницы, где и будет размещена информация. Может быть двухмерное и трёхмерной, где объекты могут иметь глубину и объём;
- **линия** - элемент дизайна, который может быть прямым, кривым, горизонтальным, вертикальным, волнистым, пунктирным и т.д., с помощью которого определяются силуэты и очертания;
- **фигура** - внешнее очертание предмета, расположенное в одной плоскости (не объёмное). Различают геометрические (круг, квадрат, прямоугольник, треугольник) и органические фигуры (контуры облака, животного, машины и др.);
- **цвет** - используется для создания общего эмоционального настроения веб-сайта, для выделения особенно важных моментов. При выборе цвета желательно исходить из возраста, социальных предпочтений возможного посетителя сайта. В сравнительной таблице указаны характеристики различных цветов, их влияние на настроение посетителей веб-страницы [7].

Цвет	Что символизирует?	Какие негативные чувства может вызвать?
Красный	Страсть, агрессия, сила, кровь, динамика, энергия	Раздражает, повышает давление, усугубляет стресс
Зелёный	Природа, здоровье, свежесть,	Вызывает ощущение вялости и

	спокойствие, расслабление, покой	однообразия
Оранжевый	Энтузиазм, открытость, дружелюбие, радость, счастье	Чувство невежества, ложь
Синий	Мир, надёжность, уверенность в себе, спокойствие, доверие, мощь	Ощущение холода, депрессии, безразличия
Чёрный	Власть, вечность, глубина, утончённость, сложность, формальность, тайна, протест	Чувство печали, страха, зла, гнева, несчастья
Белый	Простота, совершенство, чистота, свет, безопасность, единство, саморазвитие, мир	Чувство холодности и отстранённости
Фиолетовый	Богатство, влияние, скрытые знания, благородство, тайна, интуиция, магия, чувство собственного достоинства	Чувство печали
Жёлтый	Интеллект, оптимизм, радость	Чувство лени и зависти
Коричневый	Дружба, надёжность, мир	Ощущение скуки, глупости
Розовый	Признательность, восхищение, здоровье, молодость, нежность, чувство прекрасного	Наивность, слабость, плаксивость.

В рамках одного сайта не стоит использовать более 4-6 цветов, иначе веб-страницы станут очень пёстрыми, трудными для восприятия. Не стоит забывать и о правиле контрастности и гармонии. Для подбора подходящей комбинации можно воспользоваться онлайн цветовым кругом (например, [Color Scheme Designer](#)). Данная услуга позволяет определиться с базовым цветом, задать количество используемых при оформлении сайта цветов, просмотреть возможные результаты и т.д.

- **текстура** - используется для придания форменного (дерево, пластик, вода, шерсть...) фона веб-странице;
- **шрифт** - начертания букв, используемых для передачи информации (дизайнеру стоит использовать единый стиль на всех веб-страницах сайта). Об использовании различных шрифтов (с засечками, рублёных, декоративных и др.) рассказывается в статье [Н. Клевер "Классификация шрифтов в веб-дизайне и особенности их использования"](#).
- **светотень** - игра светлых и тёмных тонов в рамках одного объекта;
- **форма** - созданный с помощью игры светотени или объединении нескольких фигур объёмный объект, для которого можно определить ширину, высоты и глубину;
- **размер** - величина объекта. Изменяя размер, можно обратить внимание на нужную часть страницы, выделить объект из остального потока информации.



- статья "Элементы и принципы дизайна".

Принципы веб-дизайна

После того как выбраны основные элементы для создания качественной веб-страницы, необходимо их правильно соединить между собой. Картина в целом должна смотреться гармонично, быть единым целым и не разделяться на части, т.е. составлять **композицию**. К базовым основам создания подходящей композиции любой веб-страницы относятся [8]:

- **баланс** -

"Это равновесие взаимодействующих или противоположных сил в композиции. В такой композиции не возникает ощущения, что какая-то её часть доминирует над всем остальным. Баланса можно достичь с помощью правильного размещения объектов, размеров объекта и по цвету." [8]

- **контраст** - противопоставление (формы, объёмов, цветовой гаммы и т.д.) различных частей композиции опять же с целью привлечь большее внимание на конкретный объект (выделить основное меню, главную информацию и др.). Незаурядный дизайн позволяет выделить текущую веб-сайта из многомиллионного количества сайтов;
- **значимость и подчинённость** - определение центральной, главной, смысловой нагрузки (сути) всей композиции - момента, на который пользователь будет сразу же обращать своё внимание. Может быть задана цветом, размером, формой объекта. Дополнительные, второстепенные, части будут находиться в подчинении у главной мысли веб-сайта;
- **направление внимания** - плавное перемещение взгляда посетителя по элементам страницы - чем привычнее расположение вещей, тем легче воспринимается информация. Сюда же может относиться и удобство в использовании панели меню. Если придерживаться негласного правила (меню располагается сверху или с левой стороны экрана), то посетитель интуитивно быстрее найдёт необходимую информацию;
- **пропорции** -

"Это отношение отдельной части ко всему объекту, а также соотношение отдельных частей друг с другом." [8]

- **масштаб** - отношение размера конкретного объекта к другим частям веб-страницы в контексте общей композиции;
- **повторение и ритм** - использование одних и тех же элементов (фигур, цвета, шрифта и т.д.) дизайна на страницах сайта с целью объединить единой логикой между собой различные страницы и помочь посетителю лучше сориентироваться на веб-сайте;
- **единство в разнообразии** - подчинение различных элементов дизайна единой мысли для достижения общей слаженной композиции, использование общего стиля оформления всего веб-сайта (единая линия дизайна прослеживается на всех страницах).

Следуя этим принципам, можно создать веб-страницу, где все элементы связаны между собой, в работе сайта прослеживается единый дизайн и посетитель легко ориентируется в информационном потоке.

Далее предложенное видео (автор: Анна Лауринен, https://www.youtube.com/watch?feature=player_embedded&v=z-buLxThHXw) позволяет закрепить ранее пройденный материал по основам веб-дизайна.



- статья Н. Клевер "Как правильно использовать математические закономерности в дизайне сайтов";
- статья "Принципы дизайна";
- статья С. Яценко "Ваш сайт глазами пользователей. Схемы, которым следует большинство посетителей при просмотре сайтов".

Избегай типичных ошибок!

Новичкам зачастую кажется, что чем больше информации, оформленной различными стилями (цвет, шрифт, размер), вычурных кнопок, многоуровневых меню с анимированными подпунктами располагается на веб-сайте, тем больше он заинтересует посетителя. На самом же деле это не всегда производит достойное впечатление - скорее всего человек постарается побыстрее закрыть такую веб-страницу и найти такой сайт, где нужная информация представлена в спокойном, сдержанном, не раздражающем стиле. Поэтому, потратив немного времени на изучение типичных ошибок, допускаемых в веб-дизайне, можно сократить путь поиска оптимальных решений на будущее.

- статья "Ошибки начинающих дизайнеров";
- статья "10 главных ошибок построения сайта";
- статья сайта SmashingMagazine.com "10 наиболее популярных ошибок юзабилити, которые Вы обязаны знать и не повторять";
- статья "Веб-дизайн с нуля. Часть 2" - посвящена исключительно вопросам оформления текста.

Следуя описанным в статьях нехитрым рекомендациям, можно достичь хороших результатов. Выполняя практические задания данного электронного курса, ты постепенно будешь совершенствовать свои навыки. Всё возможно, если практиковаться и не сдаваться! Удачной практики!

Использованный материал

- [1] Школа интернет-профессионалов, Что такое www (world wide web) (онлайн) <http://www.inetprofy.ru/component/content/article/8-articles/74--www-.html> (29.07.2013)
- [2] Internet World Stats, TOP 20 countries with the highest number of Internet users (онлайн) <http://www.internetworldstats.com/top20.htm> (29.07.2013)
- [3] Internet World Stats, TOP 50 countries with the highest Internet penetration rate (онлайн) <http://www.internetworldstats.com/top25.htm> (29.07.2013)
- [4] Что такое веб-дизайн? Сегодня в это понятие имеет большое по широте охвата значение (онлайн) <http://www.perfect-design.ru/stati17.htm> (29.07.2013)
- [5] Основы Web-дизайна (онлайн) <http://elearn.oknemuan.ru/?p=8&id=37> (30.07.2013)
- [6] Н. Клевер, Основные элементы дизайна, участвующие в создании композиции (онлайн) <http://www.designonstop.com/webdesign/article/osnovnye-elementy-dizajna-uchastvuyushhie-v-sozdanii-kompozicii.htm> (30.07.2013)
- [7] Н. Клевер, Психология цвета и его воздействие на эмоциональное состояние (онлайн) <http://www.designonstop.com/webdesign/article/psixologiya-cveta-i-ego-vozdjeystvie-na-emocionalnoe-sostoyanie.htm> (30.07.2013)

[8] Н. Клевер, Основные принципы создания гармоничной композиции в дизайне (онлайн)
<http://www.designonstop.com/webdesign/article/osnovnye-principy-sozdaniya-garmonichnoj-kompozicii-v-dizajne.htm> (30.07.2013)

НЕДЕЛЯ 2: ОНЛАЙН РЕДАКТОР ВЕБ-СТРАНИЦ WEEBLY

Предисловие

Название книги: Онлайн редактор веб-страниц *Weebly*

Основные цели работы с книгой:

- создать аккаунт для работы с *Weebly*;
- изменить первичные настройки будущего веб-сайта;
- использовать различные элементы редактора для расположения и добавления информации на веб-страницы;
- использовать средств общения, получение обратной связи от посетителя сайта;
- публикация работы во Всемирной паутине.

Рассматриваемые темы:

- Что такое *Weebly*?
- Создание аккаунта.
- Первичные настройки. Панель меню.
- "Шапка" и макет веб-страницы.
- Элементы редактора.
- Сайт в WWW.

Результатом проработки данного материала является умение учащегося:

- оформлять веб-страницы;
- добавлять материал на сайт;
- использовать различные средства коммуникации;
- делать сайт доступным для посетителей.

Что такое Weebly?

Weebly [произносится как *уибли*] - это редактор (конструктор, среда), который позволяет создать веб-сайты, состоящие из множества веб-страниц. Он доступен во Всемирной паутине по адресу <http://www.weebly.com>. Это означает, что дизайнер не должен что-либо устанавливать на свой компьютер, и вся работа происходит посредством браузера (лучше всего использовать *Mozilla Firefox* или *Google Chrome*). Используя конструктор, пользователь не должен обладать знаниями о языке разметки гипертекстов HTML (при желании продвинутые пользователи смогут всё-таки внести изменения в HTML-код). Основной принцип работы заключается в выборе подходящего элемента интерфейса (виджета) и его перемещении с помощью мышки в нужное место на веб-странице. При выборе того или иного виджета и происходит создание макета будущего сайта. Внутри макета можно добавлять не только текстовую информацию, но и иллюстрации, видео-материалы, организовывать работу форума и интернет-магазина, менять структуру каждой страницы веб-сайта и многое другое. Готовая работа размещается на сервере weebly.com бесплатно.

Пользуясь редактором *Weebly*, всего за несколько минут можно создать простой веб-сайт с интересным фоном, приветствием, иллюстрацией, сделать работу доступной всем пользователям Всемирной паутины. И это реально - смотрите видео [*располагается в учебных материалах в среде Moodle*]!

Ниже представлен англоязычный ролик (https://www.youtube.com/watch?v=aBdOgTj7x7o&feature=player_embedded), рассказывающий о возможностях конструктора *Weebly*.

Создание аккаунта

Для работы в редакторе *Weebly* разработчику сайта необходимо создать один раз **аккаунт**. Для этого существует две возможности:

- воспользоваться существующим аккаунтом Facebook;
- создать уникальный логин в самом *Weebly*.

Для начала необходимо открыть браузер и перейти по адресу <http://www.weebly.com>. Далее пользователь может выбрать удобный для работы язык (см. *Рисунок 1 - правый нижний угол*).

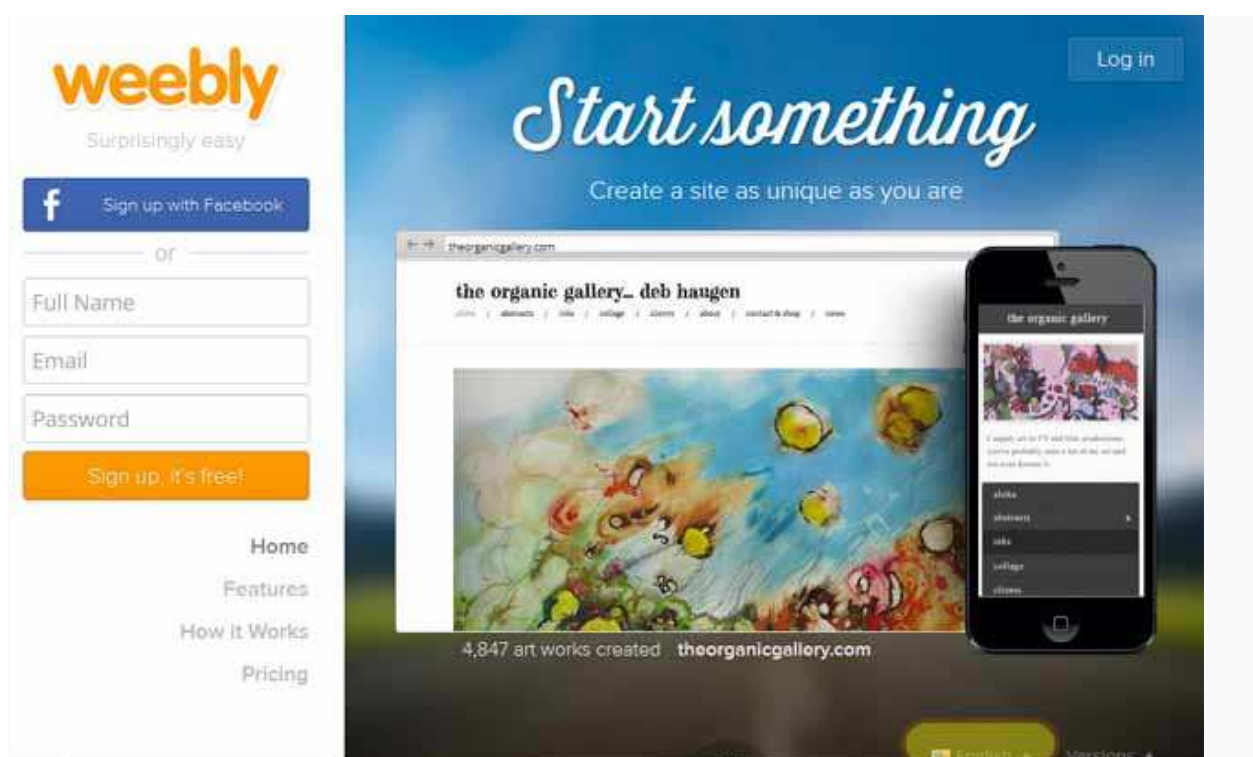


Рисунок 1: Главная страница редактора Weebly с возможностью изменения языка интерфейса.

В данном случае выбирается русский язык. К сожалению, перевод ещё не завершён. Поэтому в книге будут использоваться два языка.

Регистрация с помощью Facebook

С левой стороны главной страницы *Weebly* имеется кнопка **Зарегистрироваться с учётной записью Facebook** (см. *Рисунок 2 - левая сторона*).



Рисунок 2: Создание аккаунта.

Нажав на него, пользователь вводит свои данные и даёт согласие на использование своего аккаунта при работе в *Weebly*.

Создание уникального логина

Для создания нового аккаунта необходимо заполнить три поля, расположенных с левой стороны на главной странице *Weebly* (см. *Рисунок 2*). Адрес электронной почты необходимо указать верно - именно на него будет приходить информация о новом пароле, если забыть старый.



Выбери подходящий для тебя способ и создай учётную запись для работы в редакторе *Weebly*.



В дальнейшем вход в редактор *Weebly* будет осуществляться через расположенную в правом верхнем углу главной страницы кнопку **Log in** (см. *Рисунок 3*) и вводом данных аккаунта (Facebook или созданного уникального логина).



Рисунок 3: Вход в редактор Weebly.



Один и тот же аккаунт может быть связан с несколькими сайтами. Для создания нового веб-сайта необходимо после входа в редактор нажать кнопку **Добавить сайт** (см. *Рисунок 4*).

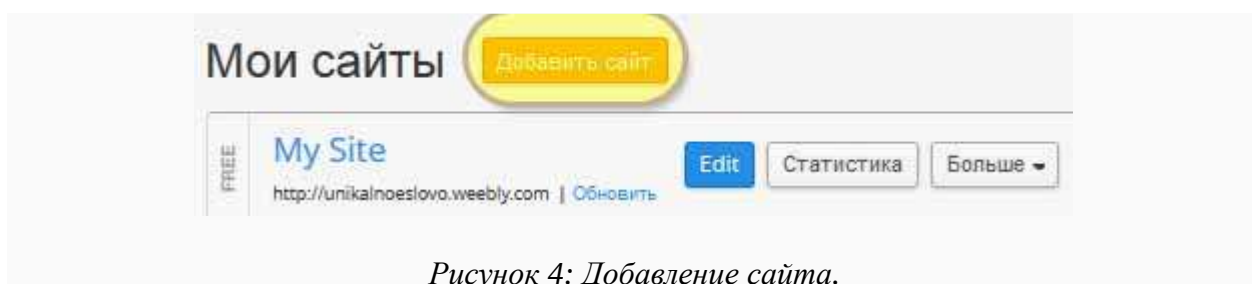


Рисунок 4: Добавление сайта.

Первичные настройки. Панель меню

Выбор фона

После создания аккаунта пользователя автоматически перенаправляют на страницу, где можно выбрать фон и цветовую гамму будущего веб-сайта. На выбор предлагается более 120 готовых вариантов. В зависимости от шаблона при наведении мышкой на него появляется возможность изменить используемые цвета кнопок, фона (см. *Рисунок 1*). Нажав кнопку **Выбрать** (англ. яз. *Choose*), можно зафиксировать выбор шаблона.

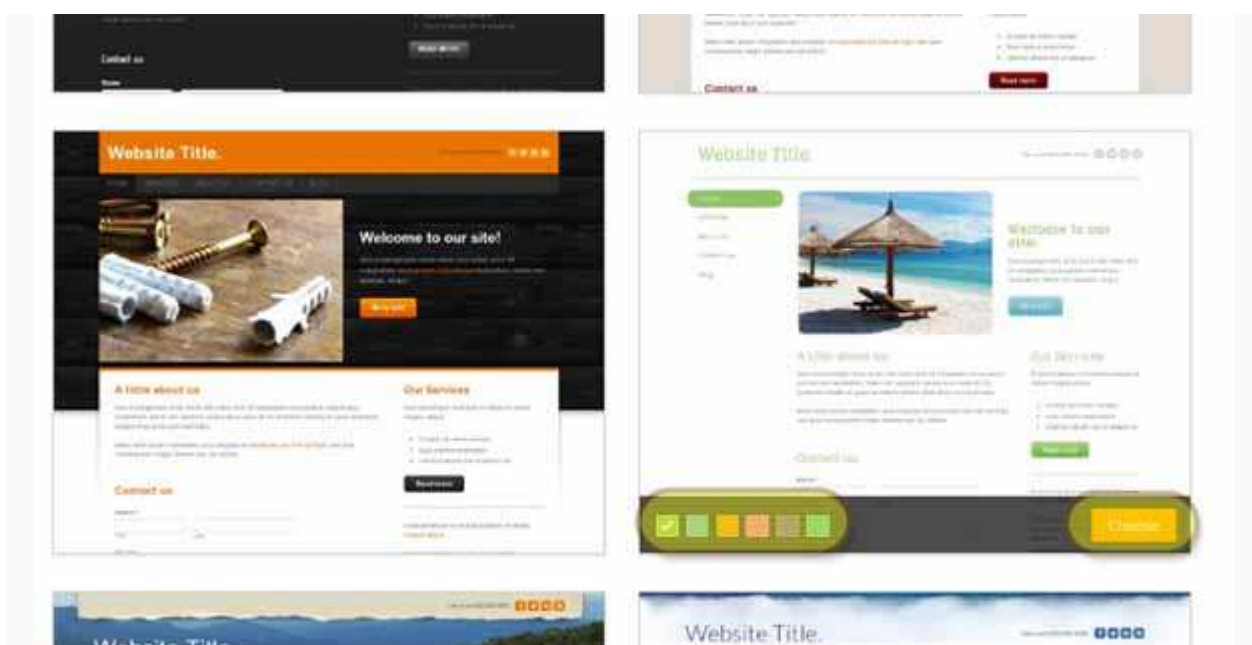


Рисунок 1: Выбор шаблона и изменение цветовой гаммы.



Выбери фон и цветовую гамму для своего пробного веб-сайта.

Если был выбран фон с иллюстрацией в верхней части страницы, то в дальнейшем её можно заменить собственным. При работе в редакторе *Weebly* разработчик в любой момент может внести изменения в свою работу. Поэтому сейчас выбранный фон можно будет при желании потом изменить.

Выбор веб-адреса

Следующим шагом в первичных настройках является определение веб-адреса, где будет располагаться создаваемый веб-сайт. Мы воспользуемся бесплатной версией и поэтому введём уникальное слово латиницей (или цифры) в первую ячейку (см. *Рисунок 2*). В итоге наш адрес будет вида *UnikalnoeSlovo.weebly.com*. Слово должно быть уникальным,

так как не может существовать два сайта с одним и тем же адресом (Интернет-системы не смогут их отличить, не смогут понять какую страницу необходимо показать пользователю). Примером уникального слова может быть твоё имя и фамилия (Koljalvanov), имя и дата рождения (Kolja1011), псевдоним (Nicolaus) и т.д. Как уже упоминалось ранее, изменения в свою работу можно внести в любой момент. Это касается и изменения веб-адреса.

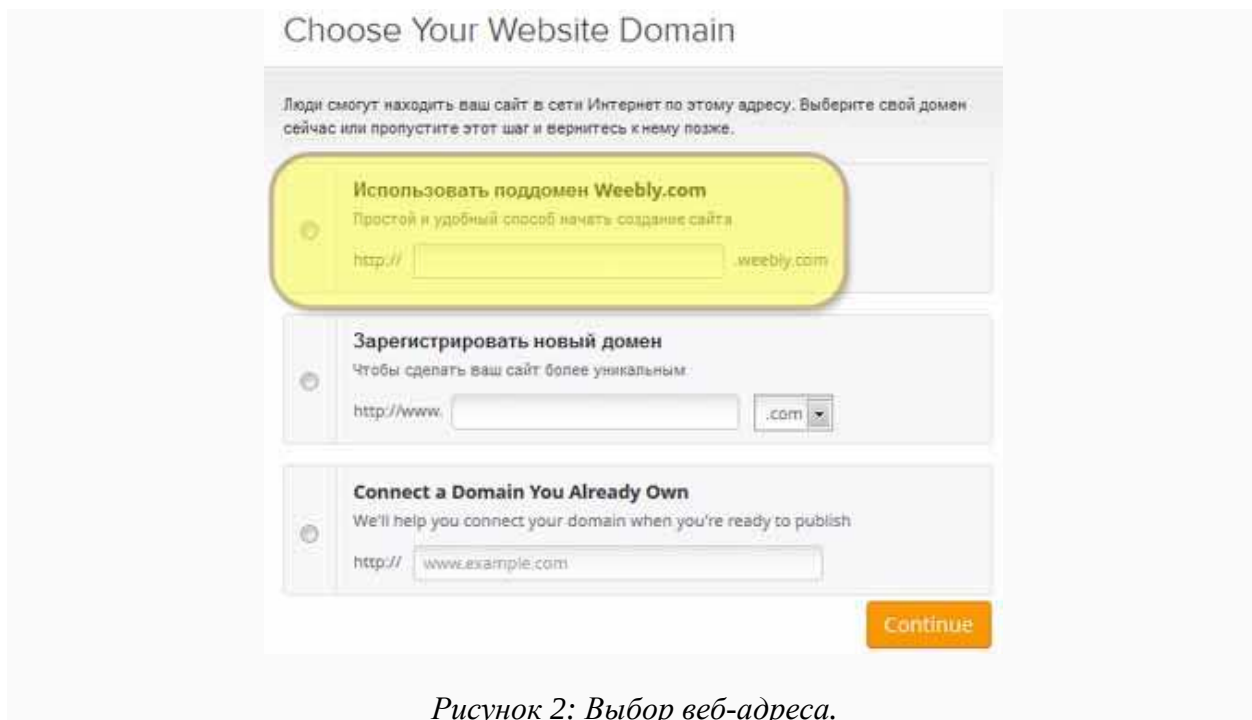


Рисунок 2: Выбор веб-адреса.

После того как разработчик веб-сайта выбрал уникальное слово (цифры), необходимо нажать кнопку **Продолжить** (англ. яз. *Continue*). При желании можно просмотреть короткое англоязычное видео, демонстрирующее возможности редактора Weebly.



Подумай и запиши веб-адрес для своего сайта в бесплатной зоне weebly.com.

Спланировать сайт

Под видео есть кнопка **Спланировать сайт**, нажав на которую можно приступить к указанию целей, основных тем веб-сайта (см. главу "Этапы веб-дизайна" из книги темы 1).

Создать сайт. Панель меню

Можно сразу же нажать кнопку **Создать сайт**. В таком случае пользователь попадает на первую, главную страницу своего будущего веб-сайта. Это и есть рабочее поле, с которым и предстоит работать. Для начала рассмотрим панель меню, которая располагается наверху (см. Рисунок 3).

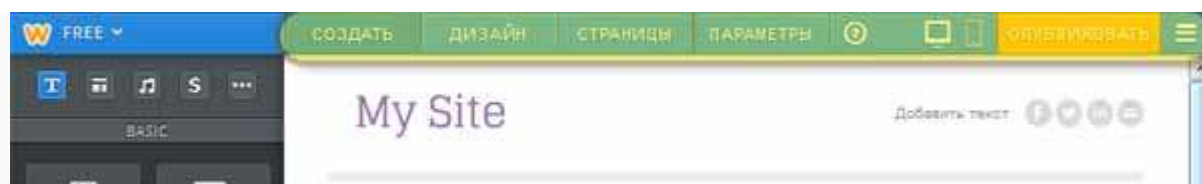


Рисунок 3: Панель меню редактора Weebly.

Меню Создать - для перехода в режим составления веб-страницы (возможность добавить текст, иллюстрации и т.д.). Рассматривается подробнее в последующих главах текущей книги.

Меню Дизайн - позволяет изменить фон, его цветовую гамму, настроить единый стиль используемых шрифтов, их цветов и размеров для разных частей веб-сайта (название сайта, заголовки частей, текст) и т.д. См. *Рисунок 4*.

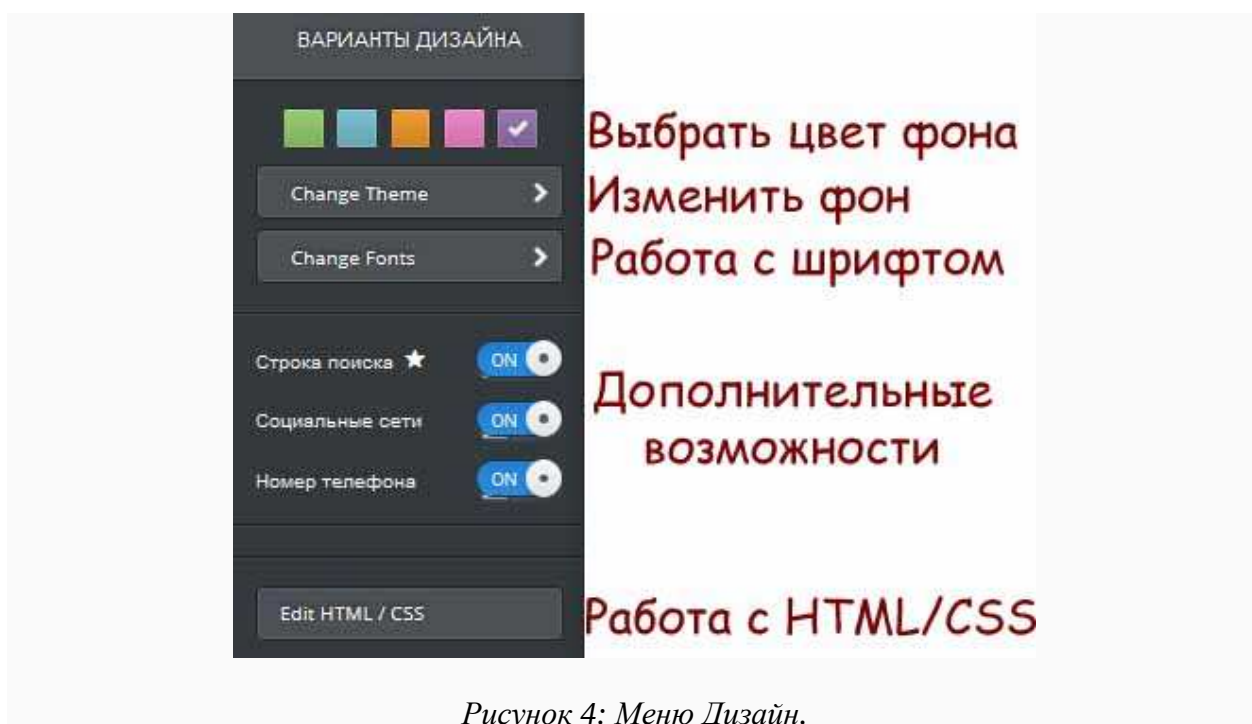


Рисунок 4: Меню Дизайн.



При желании можешь изменить настройки фона и шрифта для своего пробного веб-сайта.

Меню Страницы - предназначено для управления страницами веб-сайта (см. *Рисунок 5*).

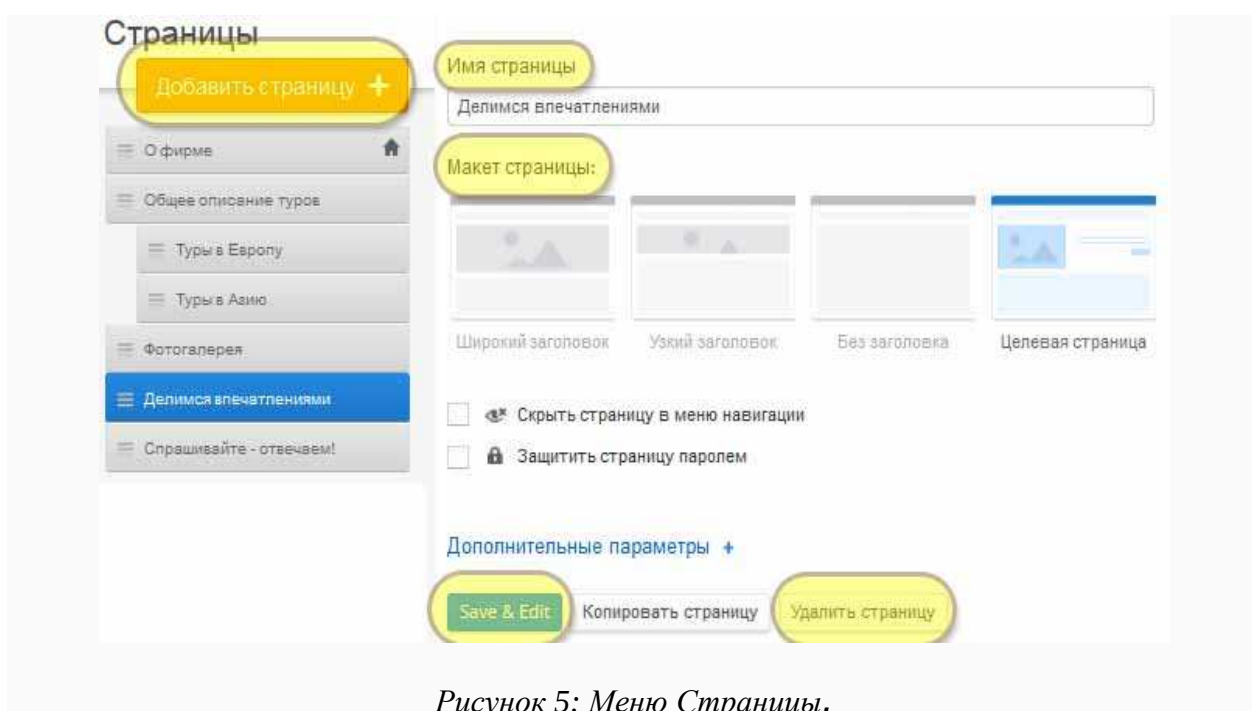


Рисунок 5: Меню Страницы.

- кнопка **Добавить страницу** - позволяет добавить ещё одну страницу на веб-сайт, при этом можно выбрать её тип: стандартная, в виде блога или же это будет ссылка на внешний источник информации. С правой стороны есть поля **Имя страницы** и **Макет страницы**, которые позволяют соответственно задать название новой странице и определиться с её общим внешним видом (более подробно о работе с макетом написано в главе 5 текущей книги). Добавленная страница появится в меню (навигации) с левой стороны. Если страница оказалась лишней, то, щёлкнув на её название, с правой стороны необходимо нажать кнопку **Удалить страницу**;
- для изменения последовательности названий в меню пользователь должен нажать и, удерживая левой кнопкой мышки нужный пункт, перетащить его вверх/вниз. Если необходимо создать подменю, то название меню следует перетаскивать вправо. Соответственно, движение влево вернёт название из пункта подменю в основное.
- по умолчанию уже есть первая стандартная страница с названием *Home* - именно на эту страницу попадают пользователи, когда в строке адреса браузера записывают адрес UnikalnoeSlovo.weebly.com. Название можно поменять, если щёлкнуть на слово *Home* и воспользоваться расположенной с правой стороны полем **Имя страницы**;
- для завершения работы необходимо нажать кнопку **Сохранить и Редактировать** (англ. яз. *Save & Edit*), после чего разработчик попадает обратно на главное рабочее поле. В зависимости от выбранного шаблона панель меню, состоящая из названия добавленных только что страниц, будет располагаться наверху или с левой стороны (см. *Рисунок 6*).

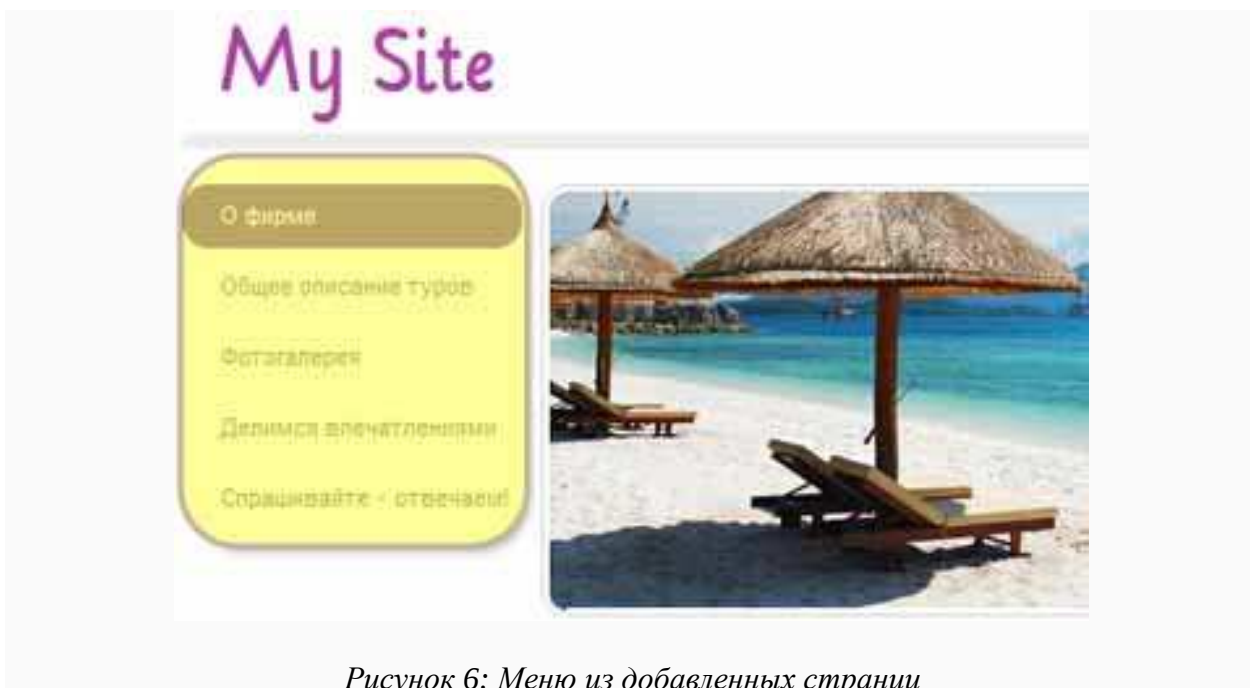


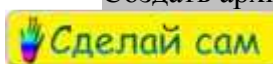
Рисунок 6: Меню из добавленных страниц



Сделай сам Переименуй страницу *Home* на Приветствие. Добавь 2 новых (под)страницы, используя разные макеты.

Меню Параметры - позволяет изменить общие настройки сайта, указать описание сайта, добавить соавторов и многое другое. Далее приведено описание основных, наиболее часто используемых, функций. Некоторые функции являются платными и будут помечены красной звёздочкой (*).

- **пункт Общий** - именно здесь можно изменить веб-адрес сайта (поле **Адрес сайта**), определиться с названием сайта и будет ли оно видно (поле **Заголовок сайта**). Поле **Значок сайта*** позволяет задать собственную иконку в адресной строке браузера. При желании можно закрыть некоторые страницы на пароль (поле **Пароль к сайту***). Поле **Навигация** поможет сгруппировать панель меню веб-сайта, если количество (под)меню очень большое и не помещается на экране. Те, кто хочет показать свой сайт в Facebook'e, должны воспользоваться полем **Публикация в Facebook**.
- **пункт Оптимизация для поисковых машин** - позволяет дать краткое описание сайта (поле **Описание сайта**) и ввести ключевые слова (поле **Ключевые слова, метатэги**), по которым сайт будет искаться во Всемирной паутине. Если разработчик по какой-то причине не хочет, чтобы сайт поисковые машины работали с сайтом, то необходимо воспользоваться полем **Спрятать сайт от поисковых машин**.
- **пункт Электронная коммерция** - позволяет организовать работу интернет-магазина, если нажать кнопку **Использовать PayPal**.
- **пункт Мобильные устройства** - воспользовавшись полем **Включить мобильную версию**, можно адаптировать веб-сайт для просмотра с мобильного устройства.
- **пункт Редакторы** - над разработкой веб-сайта могут работать несколько человек (кнопка **Добавить редакторов**).
- **пункт Код заголовка** - позволит сделать резервную копию всего веб-сайта (кнопка **Создать архив**).



При необходимости измени веб-адрес. Обязательно укажи заголовок сайта, небольшое описание сайта и введи 2-3 ключевых слова.



- нажав этот символ, дизайнер может вновь перейти к планированию сайта, получить дополнительную информацию по работе с конструктором *Weebly* (на английском языке) или сообщить разработчикам редактора о технической ошибке.



- переключение между видами веб-сайта: вид веб-сайта для компьютера или мобильного устройства.

Меню-кнопка Опубликовать - до нажатия на эту кнопку созданный веб-сайт не виден во Всемирной паутине. Этой кнопкой необходимо воспользоваться и тогда, когда вносятся изменения в содержание или оформление веб-страницы (даже если изменено одно слово). Посетитель должен видеть самую последнюю информацию.



- позволяет начать работу в полноэкранном режиме, перейти к общему списку веб-сайтов дизайнера (с одним аккаунтом может быть связано несколько сайтов), изучить статистику текущего сайта (частично бесплатно) или купить домен для собственного сайта*.

Далее рассмотрим возможности редактора *Weebly* по работе с "шапкой" веб-сайта.

"Шапка" и макет веб-страницы

Шапка

Верхнюю часть любой веб-страницы называют "шапкой". По умолчанию здесь расположен текстовый логотип веб-сайта (см. *Рисунок 1*), но редактор *Weebly* позволяет её оформить в соответствии с пожеланиями разработчика.



Рисунок 1: Текстовый логотип веб-сайта.

При наведении курсором мышки на имеющийся вариант можно с помощью всплывающей подсказки поменять логотип на графический (англ. яз. *Logo*). Добавленную иллюстрацию (см. *Рисунок 2*) можно редактировать. Работа с иллюстрациями будет описана в главе 6.1 текущей книги.



Рисунок 2: "Шапка" страницы может содержать иллюстрацию или текст.

Также изначально "шапка" веб-сайта содержит иконки-ссылки на различные социальные сети, почтовые сервисы. При наведении курсора мышки на любую из них появится панель редактирования, которая позволит указать в данной части "шапки" соответствующие контакты, добавить или удалить иконку социальной сети (см. *Рисунок 3*). С правой стороны от них можно указать поясняющий текст.

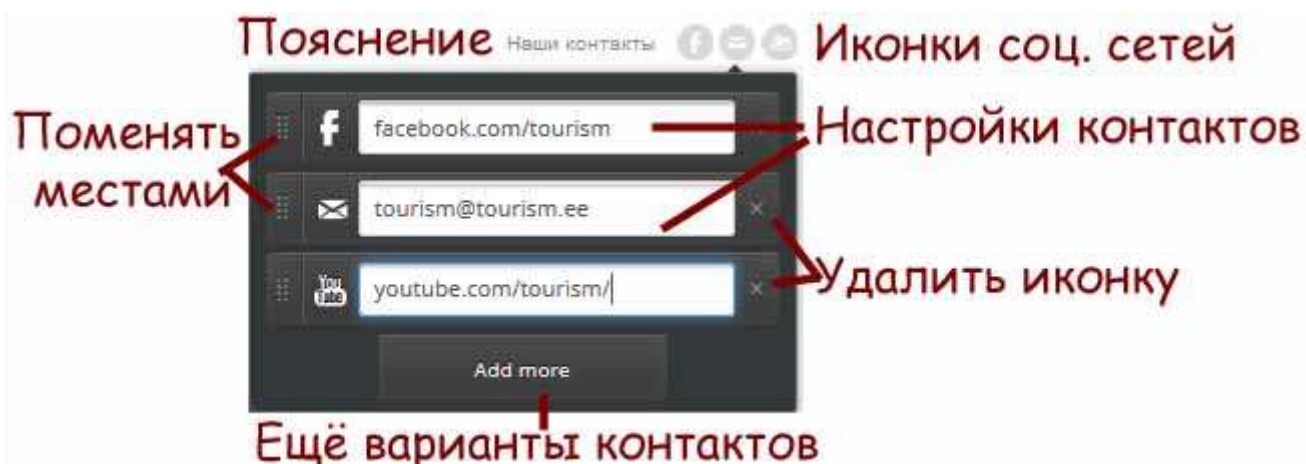


Рисунок 3: Информационная часть "шапки".

Выбранный логотип и контактные данные будут отображены на каждой странице веб-сайта.

Макет веб-страницы

В зависимости от выбранного макета страницы (см. главу 4 текущей книги) верхняя часть может также содержать дополнительную информацию:

- макет Широкий заголовок (англ. яз. *Tall header*) - большая иллюстрация;
- макет Узкий заголовок (англ. яз. *Short header*) - маленькая иллюстрация;
- макет Без заголовка (англ. яз. *No header*) - без иллюстрации;
- макет Целевая страница (англ. яз. *Landing page*) - иллюстрация, заголовок, пояснительный текст, кнопка.

Макет и иллюстрацию в начале страницы можно изменить, если навести на них курсор мышки (см. *Рисунок 4*).



Рисунок 4: Изменение макета Широкий заголовок и изображения на веб-странице.
Работа с иллюстрацией будет описана в главе 6.1 текущей книги. Редактор Weebly позволяет настроить макет для каждой отдельной страницы веб-сайта.



Измени "шапку" своего пробного веб-сайта по своему усмотрению. Используй различные макеты на веб-страницах сайта.

Элементы редактора

При работе в редакторе Weebly разработчик веб-сайта чаще всего использует меню **Создать** из верхней панели меню. Именно здесь располагаются все элементы-кнопки, которые позволяют добавлять различного вида информацию, выбирать её внешний вид и расположение.

Элементы разделены на 5 групп (см. *Рисунок 1*):

- основные (англ. яз. Basic);
- структура (англ. яз. Structure);
- медиа (англ. яз. Media);
- коммерция (англ. яз. Commerce);
- дополнительно (англ. яз. More).

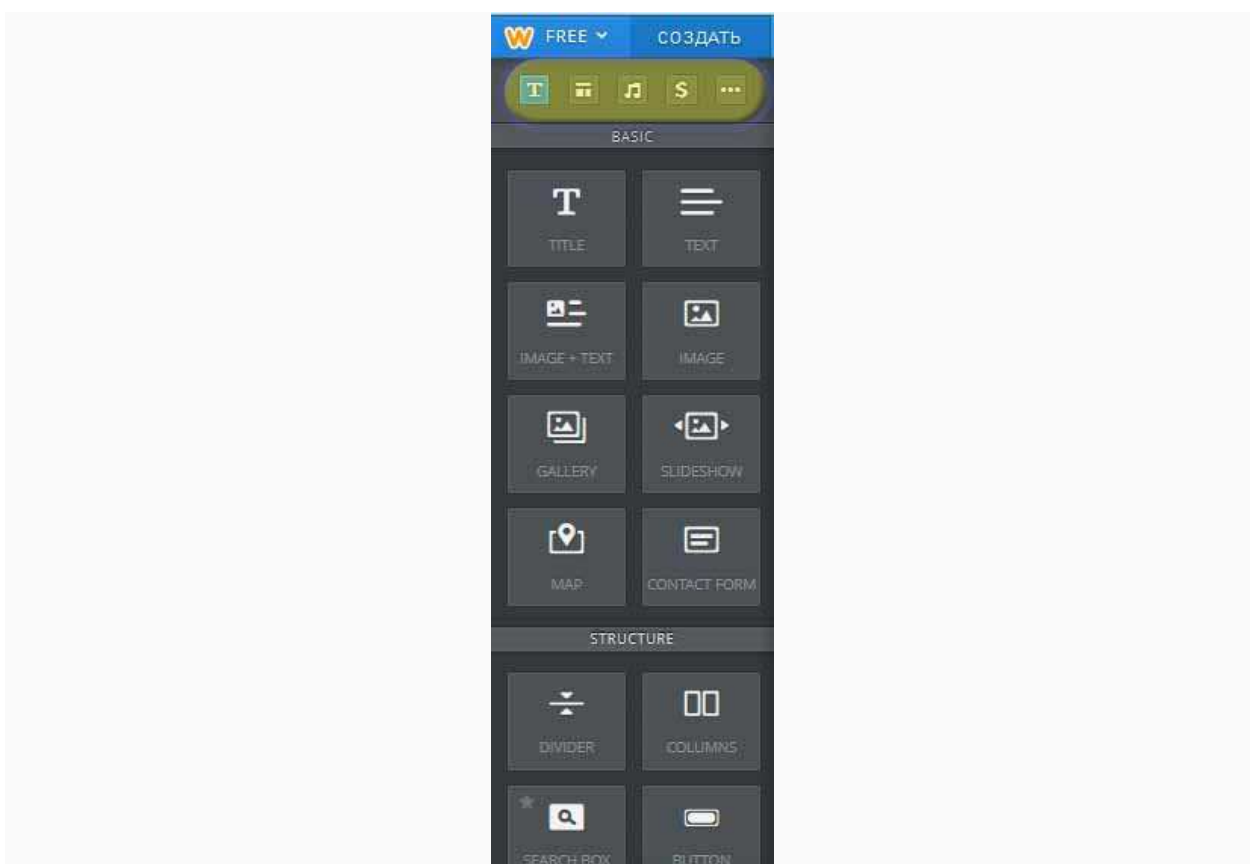


Рисунок 1: Элементы для разработки веб-страницы.

Большая часть элементов бесплатна. Если напротив названия имеется звёздочка, это означает платный вариант.

Принцип добавления элемента состоит в следующем:

1. щелчок и удерживание левой кнопкой мышки нужного элемента;
2. перенос элемента на рабочее поле (на саму веб-страницу справа);
3. ввод информации, её редактирование.

Таким образом, сначала необходимо подумать над дизайном, а потом уже добавлять материалы.

Расположенный на рабочем поле элемент можно (см. *Рисунок 2*):

1. перенести или скопировать на другую страницу в рамках одного веб-сайта;
2. поменять местами (вверх/вниз) с другим элементом;
3. удалить;
4. отредактировать, т.е. вписать текст, добавить иллюстрацию и т.д.

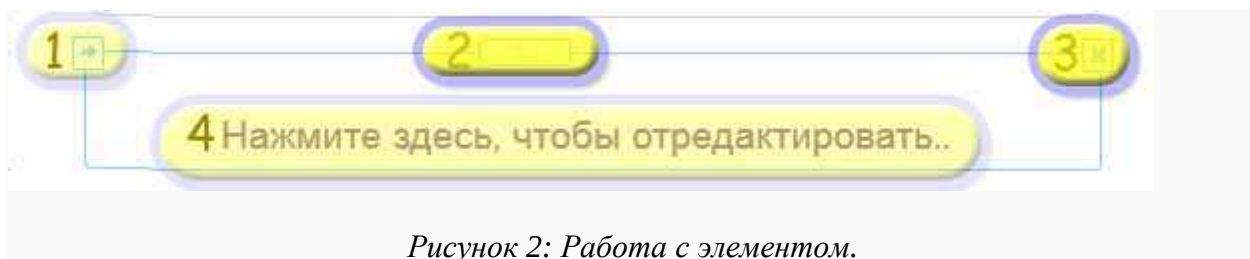


Рисунок 2: Работа с элементом.

Как правило, уже при добавлении непосредственно самой информации (будь то текст, иллюстрация, видео или другой тип информации) появляется дополнительная панель редактирования. Её содержание и возможности зависят от используемого элемента. Далее рассмотрим возможности каждой из групп.

Основные (Basic)

Группа элементов Основные предназначена для добавления текста и иллюстраций на веб-страницу. При желании можно добавить карту *Google* с указанием расположения офиса или достопримечательности, предложить заполнить форму с запросом (см. *Рисунок 1*).

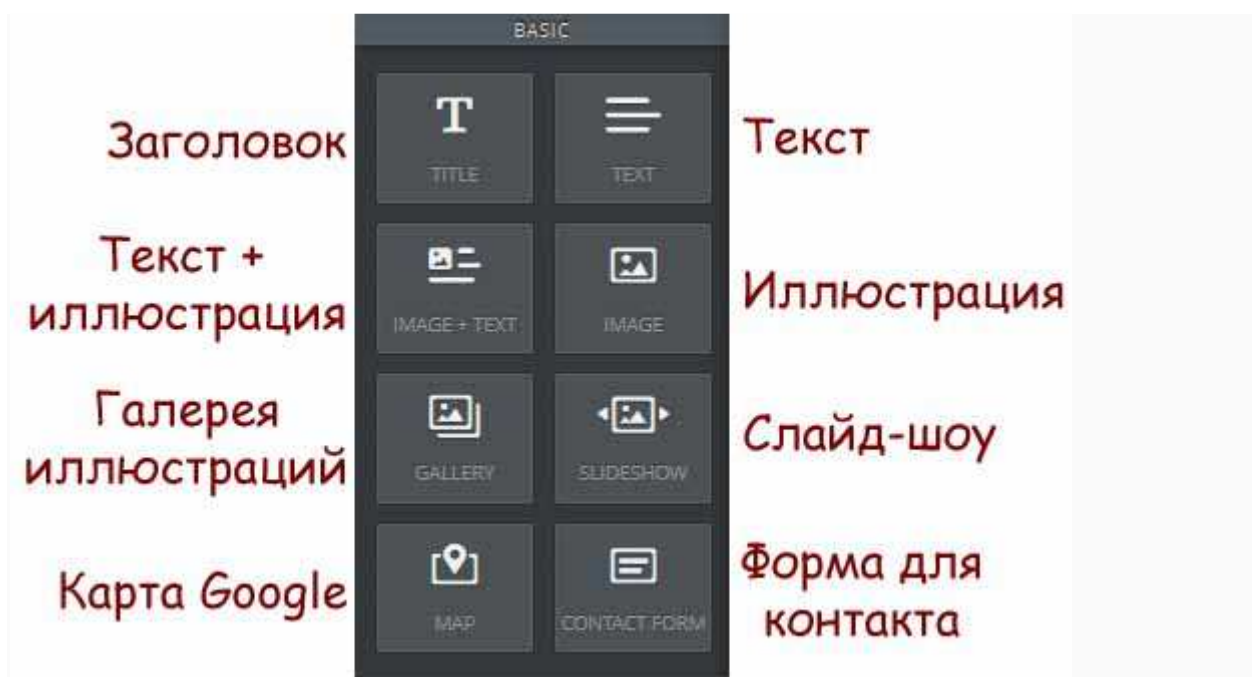


Рисунок 1: Группа элементов Основные (англ. яз. Basic).



Чтобы добавить заголовок части, необходимо перетащить на рабочее поле элемент Заголовок. При вводе любого текста появится панель редактирования (см. Рисунок 2), которая поможет его оформить (выбрать размер, цвет, выравнивание и т.д.). При наведении мышкой на ту или иную кнопку панели появится всплывающая подсказка.

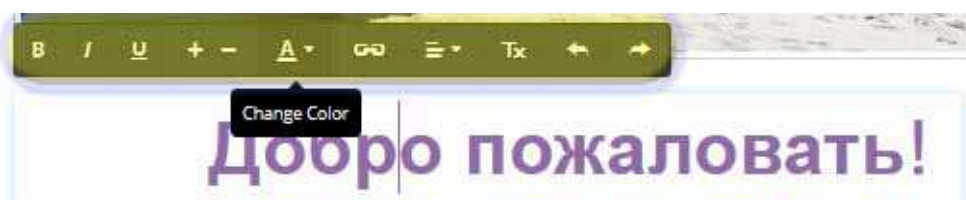


Рисунок 2: Элемент Заголовок и панель редактирования.



Конструктор Weebly позволяет использовать иллюстрации на страницах веб-сайта. Для этого следует перетащить на веб-страницу элемент Иллюстрация и нажать кнопку Загрузить иллюстрацию (англ. яз. Upload image). Далее следует выбрать, откуда загружается файл (см. Рисунок 3).

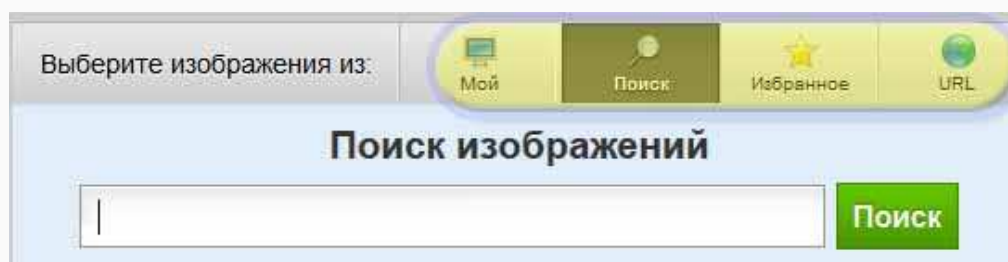


Рисунок 3: Добавление иллюстрации (вариант: поиск во Всемирной паутине).

Для изменения размера картинки необходимо навести на неё курсор мышки - в правом нижнем углу появится квадратик. Удерживая его, можно увеличить или уменьшить изображение. Добавленные фотографии можно также немного отредактировать. Для этого следует щёлкнуть на картинку и воспользоваться появившейся панелью (см. Рисунок 4).

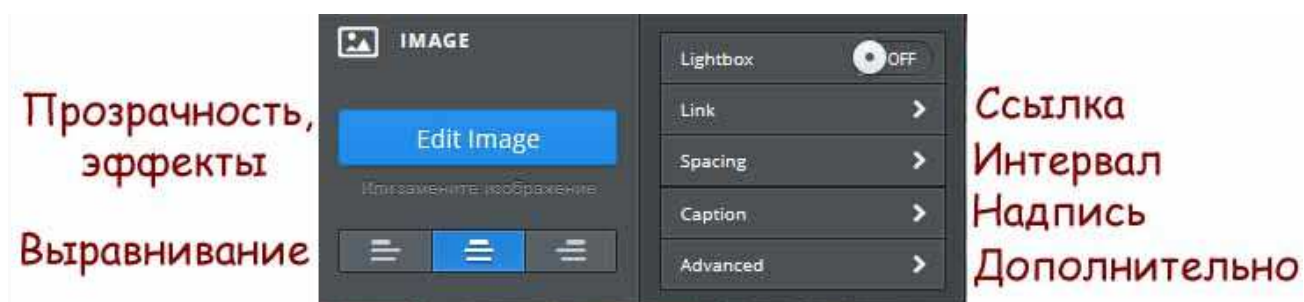


Рисунок 4: Редактирование иллюстрации.

На рисунке 5 представлен пример добавления и обработки иллюстрации средствами конструктора Weebly. Изменены размер и границы, применён эффект постепенной блеклости (англ. яз. *Fade*), добавлена надпись.



Рисунок 5: Отредактированная иллюстрация.



Элемент Карта Google позволит показать местоположение фирмы, достопримечательности, города. Первоначально на карте показан Сан-Франциско, но для изменения параметров необходимо щёлкнуть на карту и воспользоваться панелью редактирования (см. Рисунок 6).



Рисунок 6: Панель редактирования элемента Карта.

На рисунке 7 показано, где находится Тартуский университет. С помощью панели редактирования добавлен указатель масштаба, изменена кнопка карты на маленькую.



Рисунок 7: Местоположение Тартуского университета.



Видео [располагается в учебных материалах в среде Moodle].



Успел сам Подумай, какую информацию можно добавить с помощью элементов группы Основные на пробный веб-сайт. Используй 2-3 элемента.

Структура (Structure)

Группа элементов Структура позволяет добавить горизонтальную линию на страницу, разделить рабочее пространство на несколько столбцов, добавить кнопку и функцию поиска на страницу (см. *Рисунок 1*).

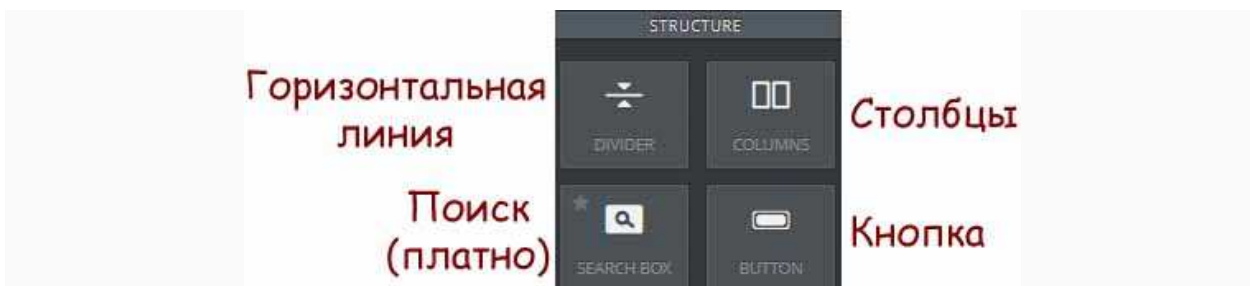


Рисунок 1: Группа элементов Структура (англ. яз. Structure).



Пример Иногда бывает необходимым расположить материал так, что с левой стороны было бы описание товара, а справа соответствующая иллюстрация/видео... Для этой цели используется элемент Столбцы. Стоит обратить внимание на тот нюанс, что Столбец связан с разделением рабочего поля на части (2,3,4 или 5), но сам по себе не добавляет информацию. Поэтому после переноса элемента Столбец на рабочее поле внутрь столбца надо поместить элемент Текст/Иллюстрация/Файл... и только тогда вносить материал (см. Рисунок 2).



Рисунок 2: Комбинированное использование двух элементов: столбца и текста.



Редактор Weebly позволяет использовать элемент Кнопка для создания определённой навигации на страницах веб-сайта. К сожалению, такую кнопку нельзя расположить в меню, а только непосредственно на веб-странице (например, под текстом или видео). Разработчик может выбрать местоположение (выравнивание) и стиль кнопки, определить, к какому файлу или на какую страницу (внутреннюю или внешнюю) посетитель будет перенаправляться. По умолчанию кнопка не имеет никакого существенного названия (кнопка текста), но его можно добавить. Для этого следует щёлкнуть на середину кнопки, удалить старое название и записать подходящий вариант (см. Рисунок 3).

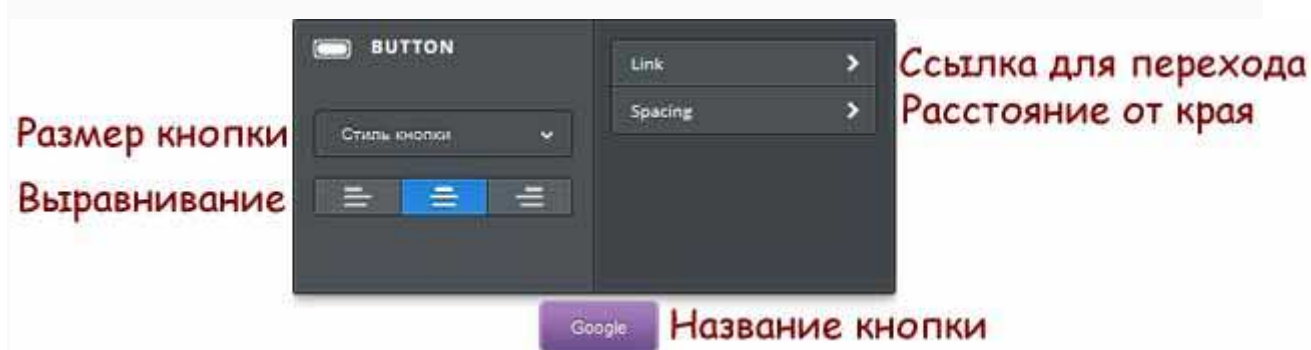


Рисунок 3: Панель редактирования элемента Кнопка.



Видео [располагается в учебных материалах в среде Moodle].



На своём пробном веб-сайте воспользуйся элементами Кнопка и Столбцы. Настройки выбери самостоятельно.

Медиа (Media)

Группа элементов Медиа предлагает возможность добавить различного типа файлы на страницы веб-сайта (см. Рисунок 1).



Рисунок 1: Группа элементов Медиа (англ. яз. Media).



Предположим имеется файл с расценками услуг. Такую информацию можно разместить на веб-странице либо с помощью элемента Документ, либо Файл. При выборе любого из них с помощью панели редактирования необходимо выбрать нужный файл с Вашего компьютера (кнопка Загрузить новый файл). После прохождения проверки на вирусы материал появляется на веб-странице. Элементы различаются тем, что Файл посетителю необходимо будет скачать на свой компьютер. В случае использования Документа информация будет сразу отображена на рабочем поле страницы в рамочке (см. Рисунок 2).

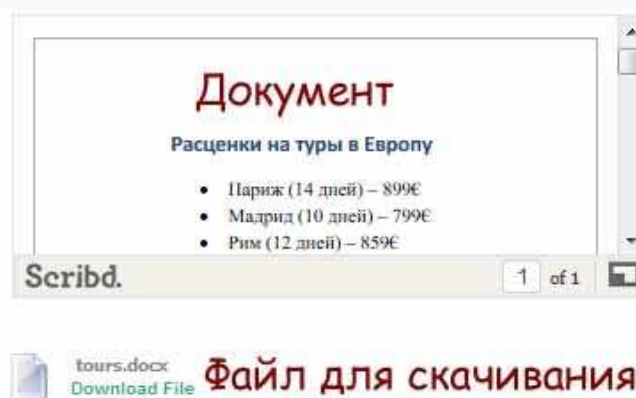


Рисунок 2: Элементы Документ и Файл.



Разместить интересное видео из YouTube на страницах собственного веб-сайта можно с помощью элемента Видео из YouTube. Прежде всего необходимо скопировать в буфер памяти компьютера точный адрес видео и затем вставить его в поле панели редактирования (см. Рисунок 3). Далее можно выбрать выравнивание видео относительно веб-страницы и указать размеры экрана для просмотра видео.

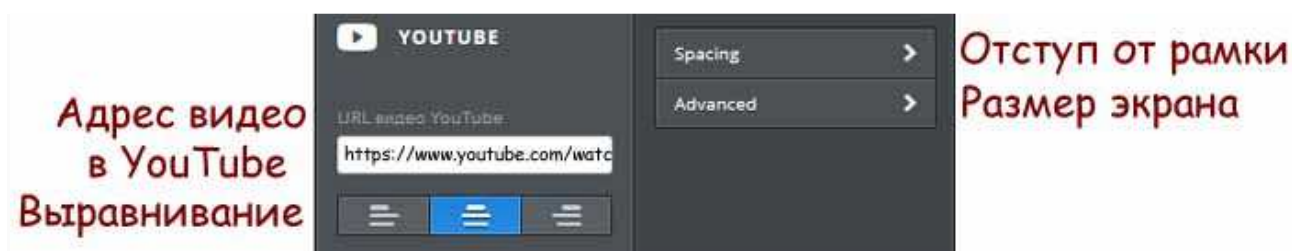


Рисунок 3: Панель редактирования элемента Видео из YouTube.



Видео [располагается в учебных материалах в среде Moodle].



Добавь на свой пробный веб-сайта интересное видео и любой файл (элемент выбери самостоятельно).

Коммерция (Commerce)

Элементы группы Коммерция позволяют использовать услуги интернет-магазина или получать доход от размещения рекламы на веб-сайте (см. Рисунок 1).

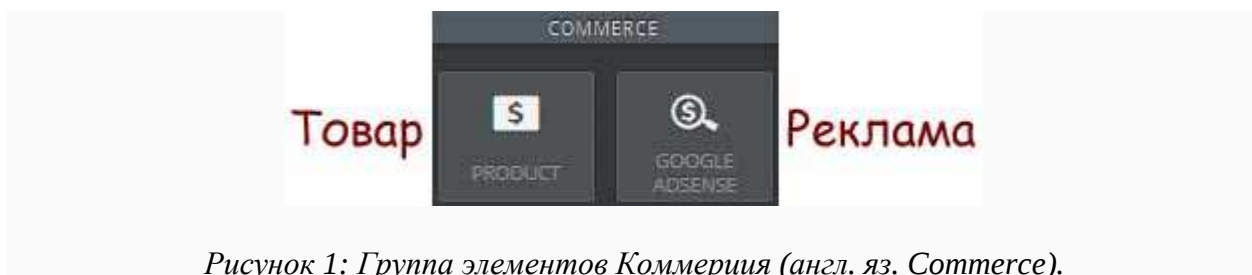


Рисунок 1: Группа элементов Коммерция (англ. яз. Commerce).

Элемент Товар предназначен для того, чтобы выложить изображение товара/услуги, дать оформленное описание и предложить посетителю сайта купить товар. Необходима предварительная настройка системы PayPal.

Элемент Реклама позволяет использовать систему GoogleAdSense. На веб-сайте будут размещаться рекламные баннеры-ссылки. Если пользователь перейдёт по ссылкам на рекламируемый товар/услугу/страницу, разработчик сайта будет получать определённую сумму денег.

Дополнительно (More)

К группе Дополнительно относятся элементы, позволяющие получить обратную связь от посетителя, добавить собственный программный код на веб-страницу, разместить иконки популярных социальных сетей (см. Рисунок 1).

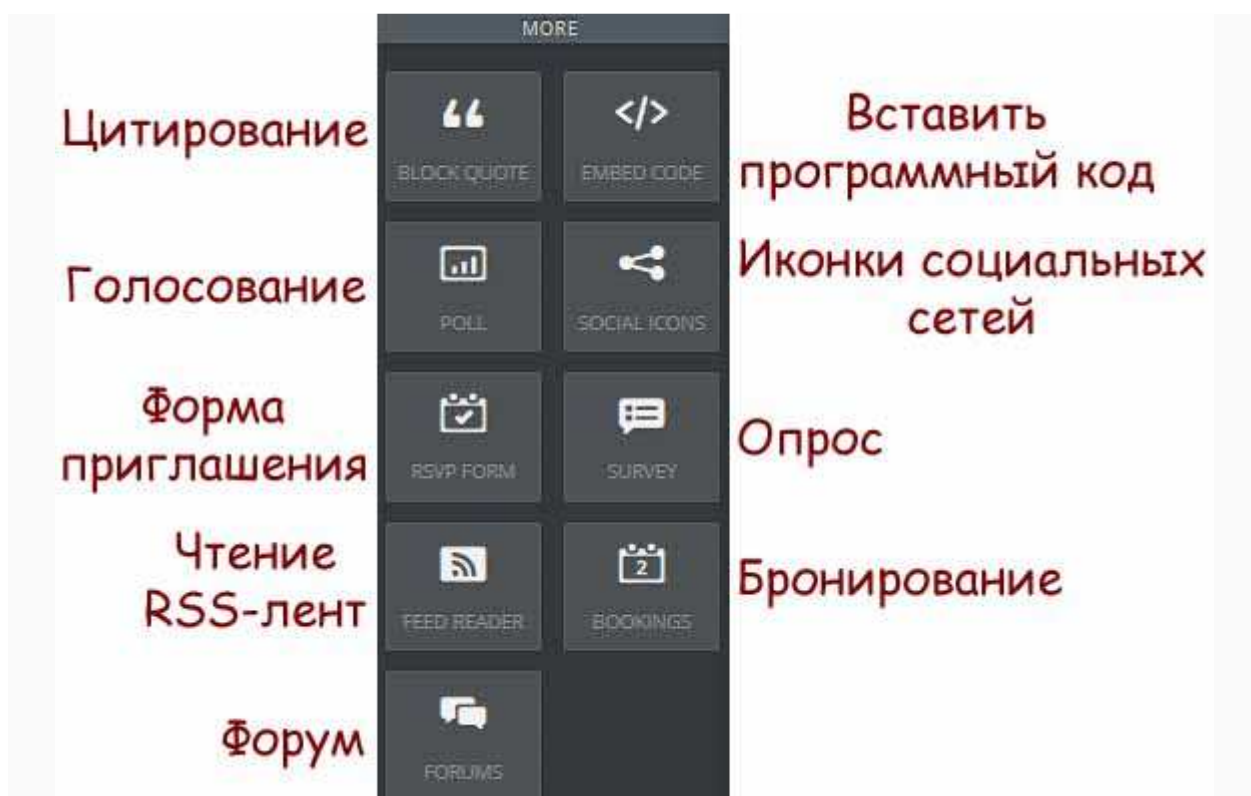


Рисунок 1: Группа элементов Дополнительно (англ. яз. More).



При использовании на страницах веб-сайта материалов других авторов необходимо соблюдать авторские права. Взятую информацию можно оформить с помощью элемента Цитирование, у которого также имеется панель редактирования. На рисунке 2 приведён результат работы с данным элементом.

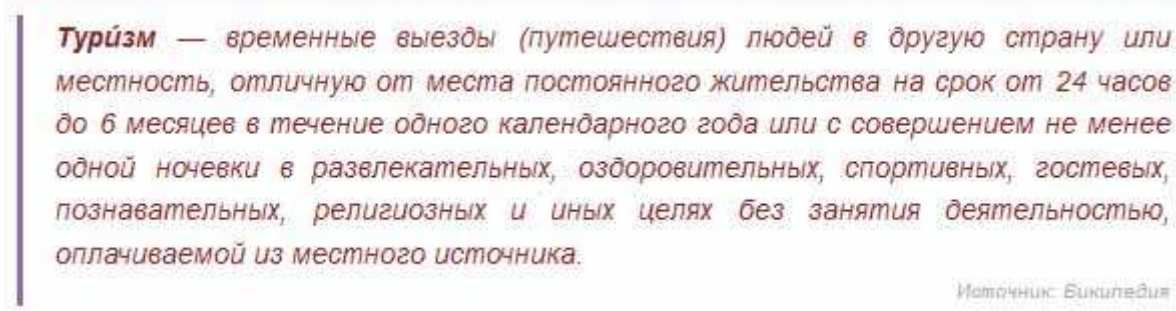


Рисунок 2: Оформленная цитата.



Редактор Weebly предлагает узнать мнение посетителей сайта с помощью элементов голосования, опрос или форум.

Если воспользоваться элементом **Голосование**, то можно узнать мнение других по заранее сформулированному вопросу. Посетителю необходимо будет выбрать один из предложенных вариантов. При активизации данного элемента потребуется настройка сервиса *Poll Daddy*.

Выбрав элемент **Опрос**, разработчик может самостоятельно составить опрос, который будет состоять из нескольких блоков (частей). Бесплатная версия позволяет использовать максимум 5 частей. По умолчанию предложенные части можно удалить (навести мышкой

на блок и нажать на расположенный в правом верхнем углу крестик). При работе с опросом на левой стороне экрана имеются различные возможности для его комплектации (см. *Рисунок 3*). Например, можно выбрать пункт, который позволит отвечающему ввести короткое слово-ответ, указать адрес своей электронной почты или выбрать вариант ответа из предложенных. Для этого необходимо перенести соответствующий элемент на рабочее поле, в опрос.



Рисунок 3: Возможности элемента Опрос.

Некоторые элементы потребуют дополнительной настройки. Например, чтобы указать варианты ответов при работе с пунктами Выпадающий список или Выбор..., необходимо воспользоваться панелью редактирования (см. *Рисунок 4*).

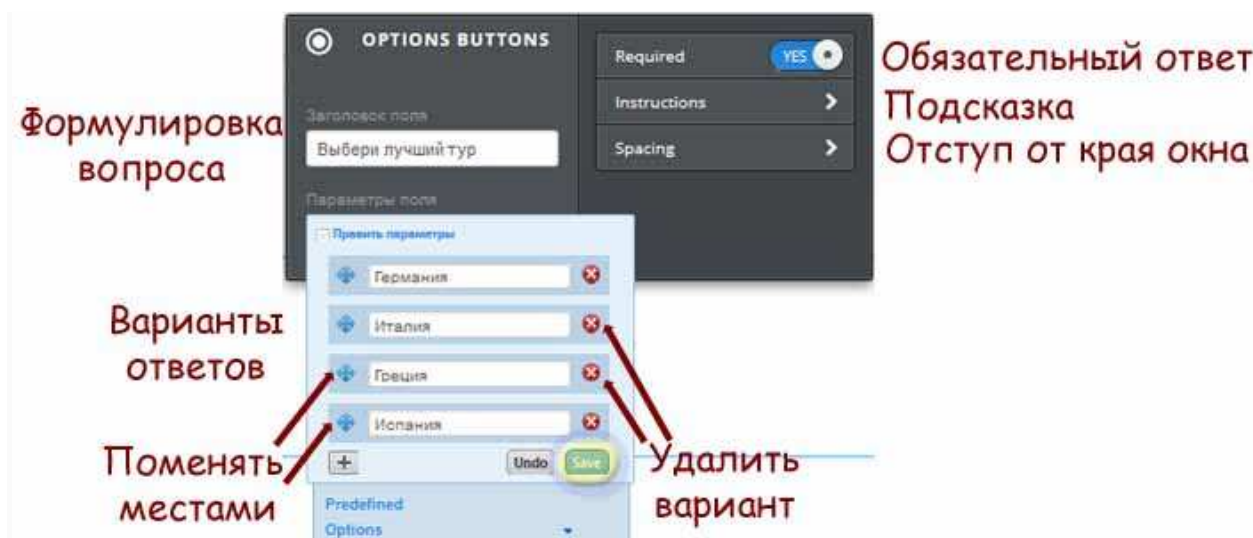


Рисунок 4: Панель редактирования элемента Выбор (1 вариант).

С результатами опроса можно познакомиться двумя способами:

1. ответы придут на электронный адрес, указанный при создании аккаунта редактора Weebly;
2. в роли разработчика сайта перейти к опросу и нажать кнопку **Просмотреть записи** (см. Рисунок 5).

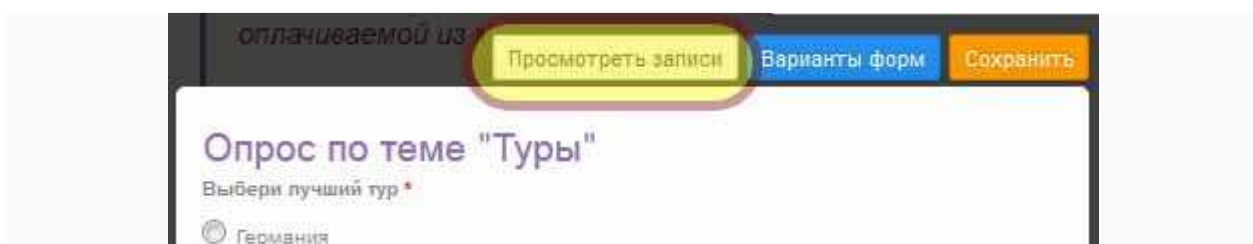
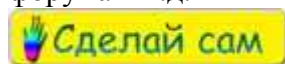


Рисунок 5: Изучение результатов опроса.

Обсуждение какого-либо вопроса можно организовать с помощью элемента **Форум**. Посетители сайта смогут оставить своё мнение, пообщаться с другими участниками форума и т.д.



По своему усмотрению добавь на свой пробный веб-сайт некоторые элементы из группы Дополнительно.

Сайт в WWW.

После того как все материалы собраны и оформлены, веб-сайт необходимо опубликовать - дать доступ посетителям к информации. Для этого разработчик должен нажать на расположенную в правом верхнем углу меню-кнопку Опубликовать. В первый раз редактор попросит ещё раз проверить будущий веб-адрес, указать категорию создаваемого сайта, ввести набор букв-цифр (англ. яз. *captcha*). Далее появится окно с адресом опубликованного веб-сайта (см. Рисунок 1).

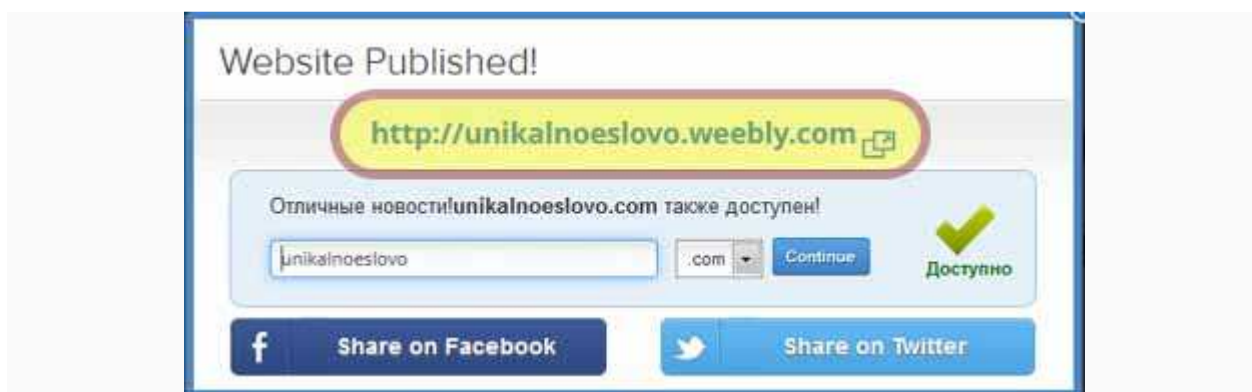


Рисунок 1: Сайт опубликован и доступен во Всемирной паутине.

Щёлкнув на адрес, разработчик увидит итоговый вариант своей работы. Именно такой веб-сайт и будет видеть будущий посетитель.

При необходимости внести изменения разработчик должен вернуться в редактор *Weebly*, подправить материалы и вновь нажать меню-кнопку Опубликовать. Обновлённая информация будет появляться только после того, как изменения будут опубликованы.



Опубликуй свой пробный веб-сайт и посмотри результат своей работы. При необходимости внеси изменения. При желании в форум темы №2 "Мой пробный веб-сайт" отправь ссылку на свой пробный веб-сайт. Однокурсники обязательно оценят твои старания!

НЕДЕЛЯ 3: ЯЗЫК РАЗМЕТКИ HTML (I)

Предисловие

Название книги: Язык разметки *HTML* (I)

Основные цели работы с книгой:

- дать определения основным терминам *HTML* языка;
- изучить возможности оформления текстовой информации;
- использовать тэги *HTML* языка для добавления иллюстраций и гиперссылок.

Рассматриваемые темы:

- *HTML* язык;
- Структура *HTML* документа;
- Элементы *HTML*;
 - Работа с текстом;
 - Работа с изображениями;
 - Работа с гиперссылками.

Результатом проработки данного материала является умение учащегося:

- различать изученные тэги и понятия;
- оформлять веб-документ, используя атрибуты элементов;
- использовать фотографии и гиперссылки в *HTML* файлах.

HTML язык

Сегодня многие из нас используют возможности Всемирной паутины для получения разнообразной информации. Но почему веб-страница, написанная автором из Германии или Аргентины, отражается корректно на мониторе пользователя в Эстонии? Как можно оформить материал красочно и интересно? Почему информация отражается правильно на планшете или на экране смартфона? Можно ли разместить на веб-странице фотографию для ссылки? ... Ответом здесь и сейчас может стать язык разметки гипертекстов *HTML*.

HTML (англ.яз. *HyperText Markup Language*) - это универсальный язык, на котором создаётся большинство документов, веб-страниц, для Всемирной паутины. Созданные с помощью языка разметки файлы имеют расширение *.html* или *.htm*. Они легко воспроизводятся **браузерами**, которые понимают язык *HTML*. Именно этот язык говорит браузеру о необходимости показать содержание страницы, структурировать материал заданным способом и т.д. Наиболее популярными браузерами на сегодняшний день являются *Google Chrome*, *Mozilla Firefox*, *Internet Explorer*. Благодаря таким программам каждый пользователь не должен знать особенности *HTML* языка, а может в удобной форме читать текст, рассматривать фотографии, прослушивать аудиофайлы, размещённые на интернет-страницах. Даже материалы данной книги в основе своей написаны на *HTML* языке. Если сделать над текущим текстом правый щелчок мышки и выбрать пункт Просмотреть исходный код (англ.яз. *View Page Source*), то можно увидеть *HTML* код для данной страницы в целом (см. Рисунок 1).

```

<!DOCTYPE html>
<html dir="ltr" lang="ru" xml:lang="ru">
<head>
  <title>Язык разметки HTML (I): HTML язык</
  <link rel="shortcut icon" href="https://mo
  <meta http-equiv="Content-Type" content="t
<meta name="keywords" content="moodle, Язык ра
<script type="text/javascript">
//<![CDATA[
var M = {}; M.yui = {};
M.pageloadstarttime = new Date();
M.cfg = {"wwwroot":"https://moodle.ut.ee","s
var yui2ConfigFn = function(me) {var p = me.pa
YUI_config = {"debug":false,"base":"https://
M.yui.loader = {modules: {}};

```

Рисунок 1. Часть HTML кода текущей страницы.

Изучая предложенную на веб-сайте информацию, пользователь может переходить с одной страницы на другую с помощью гиперссылок. Наличие переходов на *HTML* страницах и позволяет говорить о **гипертексте**. Сам язык разметки представляет собой набор определённых символов, указаний, которые передают браузеру информацию о том, как та или иная информация должна быть отображена, оформлена. Эти указания называются **элементами** и они всегда располагаются внутри угловых скобок < >. Когда элемент находится в угловых скобках, то можно говорить о **тэге**. Как правило, названия элементов напоминают соответствующие англоязычные слова. Например, тэг вида **<i>** будет браузером рассматриваться как указание начать написание текста курсивом (*i* от английского слова *italic*) или тэг **<table>** для работы с таблицей. Стоит обратить внимание, что в большинстве своём тэги парные, т.е. если в какой-то момент было дано указание записать текст курсивом, то где-то необходимо этот тэг будет закрыть с помощью наклонной линии /. Всё, что окажется внутри тэгов открытия и закрытия, т. е. сам материал, и будет оформлен указанными различными тэгами способами. Например, результат использования парных тэгов *HTML* языка **<i>** Пример **</i>** будет отображён в браузере как *Пример*. Также разрешено и практикуется использование одного тэга внутри другого, например, чтобы текст был подчёркнут и написан курсивом зелёного цвета. Поэтому можно говорить о **вложенных тэгах**. Здесь очень важно соблюдать порядок - тот тэг, который был открыт последним, должен быть закрыт первым.

Элементы могут иметь **атрибуты**, что позволяет ещё детальнее описать для браузера, как должен быть отображён материал. Например, для таблицы атрибутом может служить определение ширины столбцов, толщины линий и т.д.

В рамках прошлой темы ты уже познакомился с возможностью создавать веб-страницы с помощью онлайн редактора. В рамках данной книги ты узнаешь как правильно объявлять о начале работы с *HTML* документом, какие тэги существуют, для чего нужны атрибуты и многое другое. Но для того, чтобы собственными руками создавать такие файлы, тебе понадобится **редактор HTML**. Это специальная компьютерная программа, которая позволяет записывать и редактировать *HTML* страницы. В принципе, даже самый элементарный редактор *Notepad* подойдёт, но весь код будет написан в "чёрно-белом" варианте. Поэтому использование редактора *HTML* позволит лучше ориентироваться в коде (например, подсказка при ошибочном написании тэга; использование цветовой гамма для тэгов, атрибутов или текста; имеется возможность свернуть данные, которые находятся внутри одного тэга и т.д.). Такие редакторы бывают бесплатными и платными, одни имеют большую, более расширенную функциональность, другие же основные возможности. Небольшое сравнение приведено на рисунке 2.

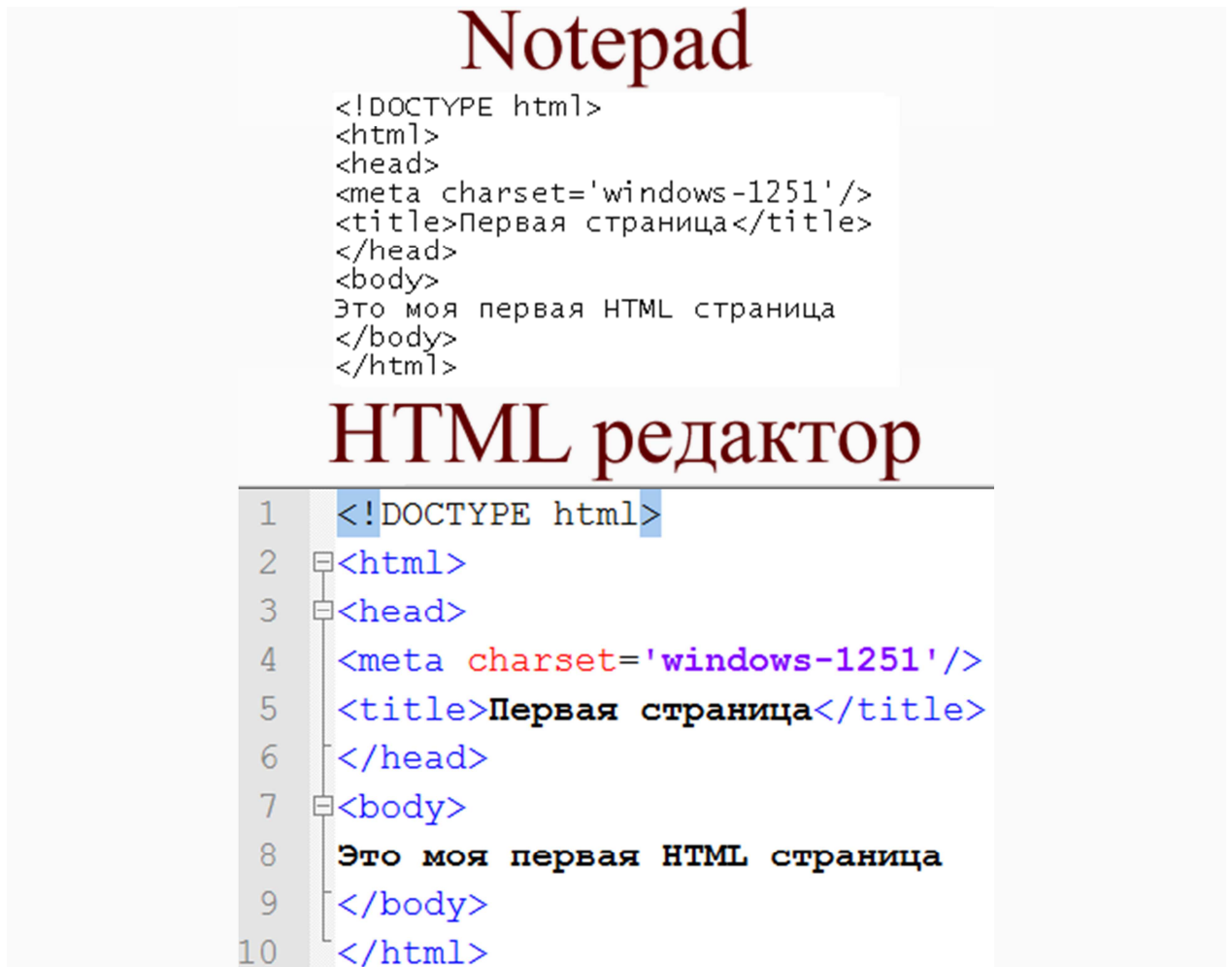


Рисунок 2: Работа с HTML кодом в разных редакторах.

При дальнейшем рассмотрении учебного материала будет использоваться бесплатный редактор *Notepad++*, который имеет русскоязычный интерфейс.



Перейди по предложенным ниже ссылкам и выполни самостоятельно:

- прочитай статью ["История создания и развития HTML"](#);
- прочитай статью ["Браузер"](#);
- скачать *HTML* редактор *Notepad++*.

Структура HTML документа

Структура любого документа, написанного на *HTML* языке, выглядит следующим образом:

1. объявление спецификации *HTML*;
2. объявление о начале работы языка *HTML*;
3. объявление о шапке;
4. содержание шапки;
5. объявление о закрытии шапки;
6. объявление о начале содержания документа;
7. содержание документа;
8. объявление о закрытии тела;
9. объявление о прекращении работы с *HTML* языком.

Объявление спецификации HTML

В начале любого документа необходимо заявить о версии *HTML*, которой будут придерживаться. Если в версии 4.01 было предусмотрено 3 спецификации, то в *HTML5* они упразднены до одного и выглядит следующим образом `<!DOCTYPE HTML>`.

Объявление о начале работы языка HTML

Чтобы браузер смог загрузить *HTML* документ должным образом, ему необходимо понять, что перед ним именно *HTML* документ. Поэтому любой код должен содержать тэг `<html>`.

Объявление о шапке и её содержание

В шапке может находиться информация о названии страницы, метаданных, стилях *CSS*, другие технические данные, которые в браузере пользователю не отражаются, но использование которых позволяет отобразить материалы корректно. Тэг для начала работы с шапкой имеет вид `<head>`. После этого следует название страницы, которое находится внутри тэгов `<title>Название страницы</title>`. Для метаданных, к которым может относиться информация о языковом коде отображения информации, ключевые слова и др., существует тэг без пары. Например, `<meta charset='windows-1251'/>` предполагает отображение информации корректно на кириллице.

Объявление о закрытии шапки

Если дальнейшая информация не относится к шапке, то её активность следует приостановить с помощью закрывающего тэга `</head>`.

Объявление о начале содержания документа

Всё содержание документа записывается в тело документа, открытие которого происходит с помощью тэга `<body>`. Всё, что написано и оформлено, начиная отсюда, будет отображено в браузере при посещении веб-страницы.

Содержание документа

Для корректного отображения информации в браузере существуют оформительские тэги, тэги работы с таблицами, формами и многие другие. Например, пара тэгов `<b> ... </b>` позволяет оформить текст полужирным шрифтом, а `<h1> ... </h1>` отобразит расположенные внутри этих тэгов информацию как заголовок первого уровня. Эти и многие другие тэги будут рассмотрены в материалах этой и следующей книг.

Объявление о закрытии тела

Когда необходимая информация записана и оформлена с помощью соответствующих тэгов, то тело документа следует закрыть с помощью закрывающего тэга `</body>`.

Объявление о прекращении работы с HTML языком

На самой последней строчке любого *HTML* документа располагается закрывающий тэг `</html>`.

Ниже приведённый рисунок 1 наглядно демонстрирует структуру элементарного *HTML* документа.

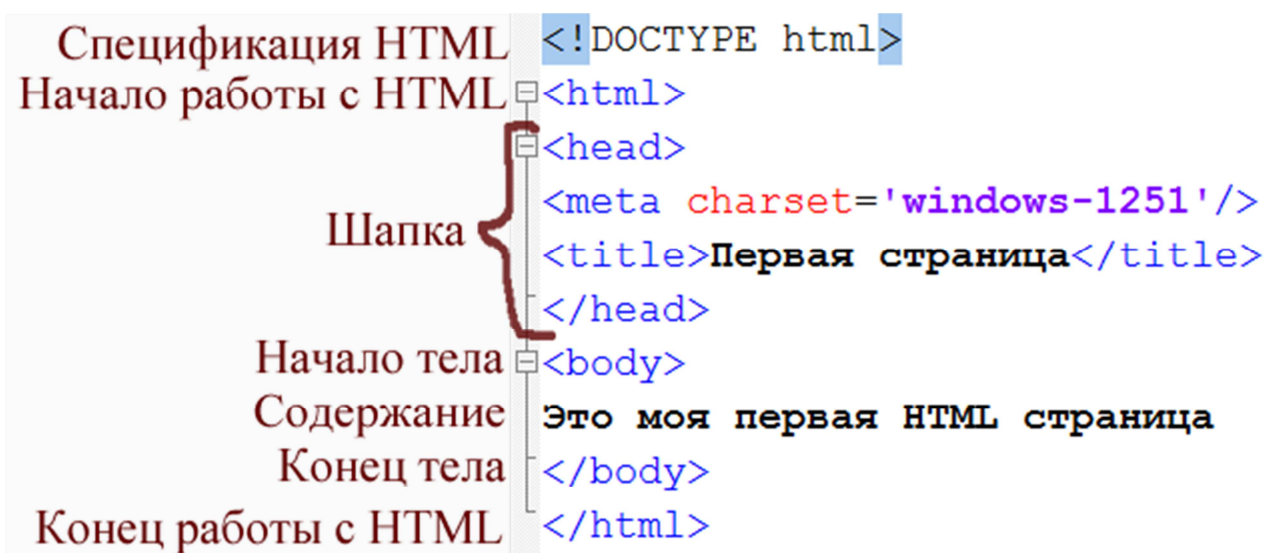


Рисунок 1. Структура простого HTML документа.

На рисунке 2 показан результат обработки такого кода браузером *Mozilla Firefox*.

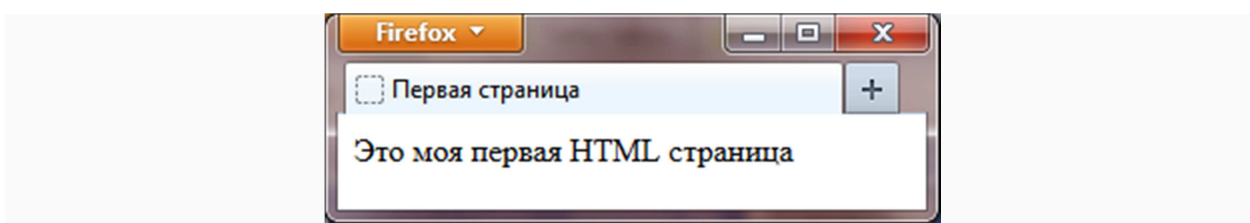


Рисунок 2. Отображение HTML кода браузером.



Открой *HTML* редактор *Notepad++* и создай заготовку для будущего веб-документа. Необходимый код приведён на рисунке 1. Сохрани файл как *tvoja_familija.html*. Чтобы посмотреть результат работы, перейди в редакторе *Notepad++* в меню *Запуск* и выбери один из предложенных вариантов по использованию браузера.

NB! Все имеющиеся в данной книге упражнения с пометкой *Сделай сам* выполняй в этом файле. Файл потом необходимо будет предоставить для оценивания.

Элементы HTML

Страницы, написанные на *HTML* языке, состоят из различных элементов. Названия элементов, в большинстве своём, схожи с англоязычными аналогами, что облегчает понимание синтаксиса. Например, для таблицы существует *HTML* элемент *table*, а для изображения *img*. При написании элемента не важен регистр букв, т. е. вариант *TABLE* и *table* будет одинаково правильно интерпретироваться браузерами. Все элементы располагаются в угловых скобках, что позволяет говорить о тэгах.

Элементы можно условно разделить на несколько групп. Есть такие, которые оформляют текст, создают списки, добавляют различные иллюстрации и таблицы, позволяют использовать гиперссылки и т.д. Далее в данной и следующей книге предложенная информация будет представлена именно по группам.

Работа с текстом

Для оформления текста может быть использовано довольно большое количество тэгов. Можно изменять размер текста и его цвет, создавать бегущую строку, оформлять текстовые блоки и т.д. Тэги можно комбинировать между собой, т.е. текст может быть записан зелёным цветом и курсивом. Главное, соблюдать порядок - последний открывающий тэг должен быть первым закрыт. В данной подглаве также будет рассмотрено использование специальных символов, добавление которых с клавиатуры невозможно.

Для лучшего понимания материала используемые тэги выделены **жёлтым** фоном, возможные атрибуты записаны *курсивом*, *коричневым цветом*. Предложенный как пример синтаксис выделяется **светло-зелёным** фоном. Напротив большинства рассматриваемых тэгов приведён пример результата. Общий пример по рассмотренному в главе материалу будет приведён в конце страницы.

Оформление текста

`<h1> ... </h1>` работа с заголовком. Всего может быть 6 уровней. Пример заголовка первого уровня приведён на новой строке.

Заголовок первого уровня

`<i> ... </i>` выделение текста *курсивом*

`<b> ... </b>` выделение текста **жирным** шрифтом

`<u> ... </u>` подчёркивание текста

`<s> ... </s>` ~~зачёркивание~~ текста

`<sub> ... </sub>` использование нижнего индекса (например, H₂O)

`<sup> ... </sup>` использование верхнего индекса (например, a²)

`<small> ... </small>` написание текста уменьшенным

`<big> ... </big>` написание текста **увеличенным**

`<br>` переход на новую строку, закрывающего тэга не существует

`<font ...> ... </font>` пара тэгов предназначена для оформления текста с помощью выбранного шрифта, размера, цвета и др. По этой причине для элемента предусмотрен ряд уникальных атрибутов (можно использовать только для определённых элементов). Атрибуты записываются сразу за названием элемента и им присваивается некое значение, которое следует записывать в кавычках " ". Закрывать атрибуты отдельно не надо, достаточно закрыть с помощью наклонной линии / сам элемент. Иногда использование элемента без атрибута бывает бессмысленно. Далее приведено описание наиболее часто используемых атрибутов для элемента *font*:

- *color* для определения цвета текста. Значение можно задать с помощью шестнадцатеричного кода, где впереди располагается символ #, или англоязычного названия. Например, фиолетовый цвет может быть задан как #9900FF или слова *magenta*. Тогда итоговый синтаксис может быть вида `<font color = "#9900FF"> ... </font>` и результат написан **фиолетовым цветом**. Шестнадцатеричные коды цветов и их англоязычные названия можно найти на [странице Puzzleweb](#);
- *size* задаёт для вводимого текста его размер по шкале от 1 до 7. Стандартным считается 3. Синтаксис: `<font size = "4">...</font>` и результат будет записан размером текста 4 единицы;
- *face* позволяет выбрать шрифт для вводимого текста. Может быть *Times New Roman*, *Verdana*, *Comic Sans MS* и т.д. Синтаксис: `<font face = "Impact">...</font>`, а текст будет оформлен шрифтом Impact.

`<p ...> ... </p>` работа с абзацем. Наиболее часто используемым атрибутом является выравнивание по центру (англ.яз. *center*), левому (англ.яз. *left*) или правому (англ.яз. *right*) краю, по ширине (англ.яз. *justify*). Синтаксис для выравнивания по середине: `<p align = "center"> ... </p>`.

Пример выравнивания данного абзаца по центру.

`<hr />` отобразит горизонтальную линию на странице. При желании художественно оформить линию можно воспользоваться рядом атрибутов:

- *color* используется для выбора цвета линии, который может быть задан шестнадцатеричным кодом или англоязычным словом. Синтаксис: `<hr color = "#9900FF" />`;
- *width* позволяет выбрать длину линии в пикселях. Синтаксис: `<hr width = "500">`;
- *size* обозначает толщину линии в пикселях. Синтаксис: `<hr size = "15">`;
- *align* предусмотрен для выравнивания линии по центру, краям или ширине. Синтаксис: `<hr align = "right">`.

Ниже отображённая линия создана с помощью кода `<hr color = "#0000FF" width = "150" size = "5" />`



`<marquee> ... </marquee>` позволяет разместить на странице бегущую строку. При желании можно задействовать имеющиеся атрибуты, например:

- *scrollamount* задаёт скорость "бега" строки в секундах. Синтаксис: `<marquee scrollamount = "5"> ... </marquee>`;
- *loop* определяет количество раз прокрутки строки. Синтаксис: `<marquee loop = "3"> ... </marquee>`;
- *bgcolor* позволяет добавить цветовой фон для бегущей строки. Синтаксис: `<marquee bgcolor = "#006594"> ... </marquee>`.

Ниже имеющаяся бегущая строка создана с помощью кода `<marquee scrollamount = "5" loop = "13" bgcolor = "99FFFF">` Пример бегущей строки `</marquee>`.

[Пример недоступен в текстовой версии]

`<div> ... </div>` обозначение блока, оформление которого обычно связано с использованием таблицы стилей CSS (тема №5).

`<span> ... </span>` обозначение строки, которая оформляется с помощью таблицы стилей CSS (тема №5).



Ниже приведена часть HTML кода с использованием некоторых выше описанных тэгов.

```

<!DOCTYPE html>
<html>
<head>
<meta charset='windows-1251' />
<title>Первая страница</title>
</head>
<body>
<h1>Оформление текста с помощью <i>HTML</i></h1>
<font color = "brown" size = "4">
    С помощью данного языка можно оформить текст по своему усмотрению.
    Например, данный текст отображается коричневым цветом и размером 5.
</font>
<br>
<p align = "center">
<font color = "red"><big>Важно!</big></font>
    При использовании сразу нескольких тэгов порядок расположения
    закрывающих тэгов имеет значение.
</p>
<hr>
<marquee>Изучайте язык разметки гипертекста.</marquee>
</body>
</html>

```

Посмотреть веб-страницу [Доступно в учебных материалах курса]

Комментарии

Иногда бывает необходимым записать поясняющую информацию к какому-то участку кода или временно "отключить" *HTML* фрагмент (например, при тестировании работы различных частей). Для этого могут быть использованы комментарии. Тэг для них будет иметь вид `<!-- ... -->`, где вместо ... записываются те слова-комментарии, указывается та часть кода, которая не должна отображаться в браузере. При необходимости вновь активизировать строки кода данная пара тэгов удаляется.

Специальные символы

Если следует добавить в материал какой-то символ, которого не существует на клавиатуре, или прописать команды кода так, чтобы они не перешли в *HTML* код, то необходимо воспользоваться специальными символами (мнемониками). Каждый из них начитается знаком "&" и заканчивается знаком ";". Примеры использования некоторых таких символом приведены ниже:

- § - знак-символ §
- © - знак-символ ©
- ± - знак-символ ±
- ¾ - знак-символ ¾
- ü - знак-символ ü
- Θ - знак-символ Θ
- ⇒ - знак-символ ⇒
- - добавление пробела между словами. Если не использовать данный вариант, а просто нажимать несколько раз на пробел, то при выведении информации в браузере несколько нажатых пробелов будут отображены как один единственный.

Более подробный список мнемоник можно посмотреть в [статье "Мнемоники в HTML"](#).



Открой свой HTML файл. Между тегами `<body> ... </body>` запиши заголовок Моё мнение и дальше укажи своё первое мнение о языке разметки гипертекстов (объём: 8 - 10 предложений). Оформи добавленную информацию следующим образом:

- заголовок зелёного цвета третьего уровня;
- под заголовок добавь горизонтальную линию любого (но не чёрного) цвета, длиной 50 пикселей, толщину выбери самостоятельно;
- весь текст должен быть выровнен по ширине;
- три самых важных слова в тексте выдели жирным шрифтом оранжевого цвета;
- под текстом добавь бегущую строку, где:
 - фон был бы светлых тонов;
 - бегущий текст был бы любого цвета и содержал символ © + твоё имя.

Сохрани работу.

Работа с изображениями

Кроме текстовой информации на веб-странице могут размещаться фотографии. Для работы с ними существует элемент *img*, тэг которого не закрывается. У данного элемента есть ряд атрибутов:

- *src* используется для определения адреса изображения. Всегда будет заканчиваться одним из графических расширений (например, *.jpg*, *.gif*, *.png*). Файл может располагаться на другой веб-странице, а может быть загружен с компьютера пользователя. В последнем случае очень важно, чтобы *html*-файл и фотография находились в одной папке. Тогда путь ссылки на графический файл сохранится. Синтаксис: `<img src = "Paris.jpg">`;
- *alt* позволяет задать поясняющий текст, который будет отображаться в браузере, если по какой-то причине сама фотография не загрузится. Синтаксис: `<img alt = "Фотография Парижа">`;
- *title* будет показывать название фотографии, если на неё навести курсором мышки. Синтаксис: `<img title = "Так выглядит весенний Париж">`;
- *width* предназначен для определения ширины фотографии. Может быть задана в пикселях или процентах. По умолчанию при добавлении иллюстрации определяется само, но можно изменить величину самостоятельно. Если данное значение не совпадает с оригинальной шириной, то фотография будет либо сжата, либо растянута. Синтаксис: `<img width = "90%">`;
- *height* тоже самое, что и предыдущий атрибут, только работает с высотой фотографии. Синтаксис: `<img height = "110%">`;
- *align* позволяет задать расположение картинки относительно текста: сверху (англ.яз. *top*), по середине (англ.яз. *middle*), внизу (англ.яз. *bottom*), справа (англ.яз. *right*), слева (англ.яз. *left*). Синтаксис: `<img align = "left">`;
- *vspace* добавляет в пикселях пространство вверх и вниз от изображения (своего рода отступы). Синтаксис: `<img vspace = "25">`;
- *hspace* работает также как и предыдущий атрибут, только добавляется пространство слева и справа от фотографии. Синтаксис: `<img hspace = "25">`;
- *border* добавит рамку вокруг изображения. Толщина определяется в пикселях. Синтаксис: `<img border = "5">`.



Часть HTML кода, позволяющая добавить к тексту фотографию.

```

<!DOCTYPE html>
<html>
<head>
<meta charset='windows-1251' />
<title>Вторая страница</title>
</head>
<body>
<h3>Добавление изображения</h3>
<p align = "justify">
<font color = "#006600">
    К обычному тексту можно добавить фотографию. В зависимости
    от значения выбранного атрибута, она может размещаться
    справа, слева и т.д. от текста. Это позволяет посетителю
    веб-страницы одновременно и читать, и видеть иллюстрацию.
    Например, здесь и сейчас под текстом располагается
    фотография макета храма Св. Матьяша, сделанная автором курса
    во время отпуска. Размеры фотографии с помощью атрибутов уменьшены.
</font>
</p>
<p align = "center">
<img src = "Model.jpg" alt = "Макет храма Св. Матьяша"
    title = "Храм Св. Матьяша" width = "30%" height = "30%">
</p>
</body>
</html>

```

Посмотреть веб-страницу [Доступно в учебных материалах курса]



Открой свой файл tvoja_familija.html и дополни его следующим материалом:

- перед текстом, но после заголовка, добавь любую фотографию. Задай размеры фотографии 200 на 200 пикселей. Изображение должно располагаться по середине, находиться в рамочке. Обязательно воспользуйся атрибутами *alt* и *title*;
- после текста добавь фотографию-эмотикон, атрибуты для которой выбери самостоятельно.

Сохрани изменения.

Работа с гиперссылками

Текст можно называть гипертекстом, если он содержит ссылки на другие материалы. При этом в роли материалов могут выступать новые веб-страницы, файлы для скачивания или переходы на определённую часть на данной же веб-странице (прообраз содержания). Различают внутреннюю и внешнюю навигацию. Как правило, ссылки выглядят в виде синего подчёркивания. Однако при желании можно организовать переход при щелчке на фотографии. Парными тэгами, работающими с гиперссылками являются `<a> ... </a>`.

Основным атрибутом является *href*. Он позволяет задать адрес, куда необходимо перенаправить посетителя веб-страницы. При указании адреса не используется кириллица, пробелы или верхний регистр. Синтаксис: `<a href = "..."> *** </a>`. Вместо троеточия ... задаётся адрес веб-страницы, куда читатель будет перенаправлен, а вместо звёздочек *** указывается слово, символ, путь к фотографии (с помощью элемента *img*), на которую необходимо щёлкнуть, чтобы попасть на новую страницу.

Внешняя навигация

Синтаксис для внешней ссылки: `<a href = "http://www.ut.ee"> ... </a>`. Здесь вместо троеточия ... может быть использовано любое слово, предложение, символ или фотография. Как правильно прописать использование фотографии рассказывалось в подглаве 4.2 Работа с изображениями. Пример синтаксиса использования фотографии в виде ссылки: `<a href = "http://www.google.com"><img src = "MyGoogle.jpg" width = "10%" height = "10%"></a>`.

Для внешней навигации характерно использование ещё одного атрибута - *target*. Он позволяет выбрать окно, в котором будет открыт новый материал. Наиболее часто используемыми значениями являются открытие документа в данном окне (англ.яз. *top*) или в новом окне (англ.яз. *blank*). При этом перед каждым значением выставляется одно нижнее подчёркивание "_". Если не использовать данный атрибут, то по умолчанию материал будет открыт в текущем окне. Синтаксис: `<a target = "_blank"> ... </a>`.



Часть HTML кода с внешней навигацией. В роли гиперссылки используются и слово, и веб-адрес, и фотография.

```
<!DOCTYPE html>
<html>
<head>
<meta charset='windows-1251' />
<title>Третья страница</title>
</head>
<body>
<h3>Использование гиперссылок</h3>
<font color = "#990000">
    В тексте могут быть использованы гиперссылки на другие веб-страницы.
    Как правило, они выглядят как синие подчёркнутые слова. Например,
    нажми вот <a href = "http://www.google.com" target = "_blank">
    здесь </a>, чтобы перейти на страницу поисковой системы <i>Google</i>.
    Но можно при желании нажать на расположенное ниже лого и перейти на
    сайт свободной энциклопедии Википедия <i><font size = "2"> (Источник
    лого: <a href = "http://commons.wikimedia.org/wiki/
    File:Wikipedia-logo-v2.svg?uselang=ru" target = "_blank">
    http://commons.wikimedia.org/wiki/
    File:Wikipedia-logo-v2.svg?uselang=ru) </a></font></i>.
<br/>
<p align = "center">
<a href = "http://ru.wikipedia.org/wiki/" target = "_blank">
<img src = "http://upload.wikimedia.org/wikipedia/commons/thumb/8/80/
Wikipedia-logo-v2.svg/103px-Wikipedia-logo-v2.svg.png" width = "10%" height = "10%">
</a>
</p>
</font>
</body>
</html>
```

Посмотреть веб-страницу [Доступно в учебных материалах курса]

Внутренняя навигация

Иногда веб-страница может быть очень длинной, но её содержание можно поделить на несколько частей с соответствующими заголовками. Тогда в самом начале веб-страницы можно создать что-то наподобие содержания. Нажав на интересующий пункт, читатель не должен использовать боковую прокрутку, а автоматически попадает к началу выбранной части. Все действия происходят внутри уже открывшегося документа. Такая навигация называется внутренней. Чтобы воспользоваться такими возможностями *HTML* языка необходимо заголовки части снабдить соответствующей меткой, желательно латиницей, записанным ключевым словом или цифрой. На одной странице не может быть две одинаковой метки. Синтаксис для заголовка: `<a name = "ChastProKompjuter"> ... </a>`. Синтаксис для создания содержания в начале страницы обязательно должен содержать символ хэш #: `<a href = "#ChastProKompjuter"> ... </a>`. Логично было бы в конце каждой такой части добавлять отсылку обратно к содержанию. Для этого используется следующий синтаксис: `<a href = "#"> ... </a>`.



Использование внутренней навигации по длинному документу.

```

<!DOCTYPE html>
<html>
<head>
<meta charset='windows-1251' />
<title>Четвёртая страница</title>
</head>
<body>
<h3>Использование внутренней навигации</h3>
В нижеприведённом содержании выбери интересующую тебя часть и
щелкни на название. Ты автоматически переместишься к началу
данной части.
<br>
<a href = "#1chast">Первая часть</a>
<br>
<a href = "#2chast">Вторая часть</a>
<br>
<a href = "#3chast">Третья часть</a>
<br>
  <a name = "1chast">Начало первой части</a>
  Содержание первой части длинного документа. Предложения добавлены методом
  <i>copy-paste</i>. Содержание первой части длинного документа.
  Содержание первой части длинного документа.
  Содержание первой части длинного документа.
  Содержание первой части длинного документа.
  Содержание первой части длинного документа.
  Содержание первой части длинного документа.
  Нажми <a href = "#">здесь </a>, чтобы вернуться к содержанию.
  <br>
  <a name = "2chast">Начало второй части</a>
  Содержание второй части длинного документа. Предложения добавлены методом
  <i>copy-paste</i>. Содержание второй части длинного документа.
  Содержание второй части длинного документа.
  Содержание второй части длинного документа.
  Содержание второй части длинного документа.
  Содержание второй части длинного документа.
  Содержание второй части длинного документа.
  Нажми <a href = "#">здесь </a>, чтобы вернуться к содержанию.
  <br>
  <a name = "3chast">Начало третьей части</a>
  Содержание третьей части длинного документа. Предложения добавлены методом
  <i>copy-paste</i>. Содержание третьей части длинного документа.
  Содержание третьей части длинного документа.
  Содержание третьей части длинного документа.
  Содержание третьей части длинного документа.
  Содержание третьей части длинного документа.
  Содержание третьей части длинного документа.
  Нажми <a href = "#">здесь </a>, чтобы вернуться к содержанию.
</body>
</html>

```

Посмотреть веб-страницу [Доступно в учебных материалах курса]



Открой свой файл `tvoja_familija.html`. Дополни его следующими данными:

- после фотографии-эмотикона добавь заголовок "Использование *HTML*'а";
- под ним опиши, как у тебя проходило первое изучение возможностей языка разметки гипертекстов. Что на твой взгляд удалось лучше всего, над чем стоит больше поработать. Объем ответа - около 10 предложений;
- оформи добавленный текст по своему усмотрению;
- добавь новый заголовок Интересные сайты;
- найди и добавь на свою страницу различными способами не менее 3 ссылок на интересные веб-страницы;
- самостоятельно выясни значения атрибутов *link*, *alink*, *vlink* для элемента *body* и с помощью них дополни *HTML* код;
- все используемые в работе заголовки отметь метками для создания внутренней навигации;
- в самом начале документа добавь содержание для быстрого перехода к той или иной части работы.

Сохрани изменения. Готовый файл необходимо предоставить для оценивания (задание №3).

НЕДЕЛЯ 4: ЯЗЫК РАЗМЕТКИ HTML (II)

Предисловие

Название книги: Язык разметки *HTML* (II)

Основные цели работы с книгой:

- изучить возможности для графического и музыкального фонового оформления веб-страницы;
- получить практические навыки по созданию списков, таблиц и форм;
- разобрать несколько примеров обработки событий.

Рассматриваемые темы:

- Фоновое оформление;
- Работа со списком;
- Работа с таблицей;
- Работа с формой;
- Обработка событий;
- Валидация;
- Справочники.

Результатом проработки данного материала является умение учащегося:

- добавлять музыкальное и графическое сопровождение веб-страницы;
- использовать изученные элементы и их атрибуты для лучшего предоставления информации;
- применять некоторые обработки событий для создания интерактивности.

Фоновое оформление

Музыкальное сопровождение

Изучение материалов веб-страницы может сопровождаться прослушиванием мелодии. К выбору музыкального сопровождения необходимо подойти серьёзно. Слушать тяжёлую музыку и при этом стараться вникнуть в изучение материалов очень сложно. Для этого между тэгами шапки `<head> ... </head>` прописывается элемент *bgsound*. Закрывающего тэга для него не предусмотрено. Зато имеется несколько атрибутов:

- *src* указывает путь до соответствующего музыкального файла, который может быть сохранён во Всемирной паутине, а может быть и на компьютере пользователя. В случае последнего варианта файл необходимо будет разместить в той же папке, что и *HTML* файл, иначе браузер не сможет его проиграть. Используемая мелодия может быть с расширением *.wav*, *.au* или *.mid*. Синтаксис: `<bgsound src = "music.mid">`;
- *loop* позволяет определиться с количеством повторов для мелодии, которое задаётся целым положительным числом. Если использовать значение "-1", то

музыкальный файл будет проигрываться бесконечно. Синтаксис: `<bgsound loop = "2">`;

- `volume` задаёт громкость звучания от 0 (максимальное, используется по умолчанию) до "-10000" (минимальная громкость). Синтаксис: `<bgsound volume = "-5000">`.

NB! Использование данного тэга поддерживается только в браузерах *Internet Explorer* и *Opera*. Не предусмотрена панель управления аудиозаписью (пауза, продолжить).

Однако существует другой способ добавить музыкальное сопровождение и при этом дать посетителю веб-страницы возможность остановить звучание, выбрать громкость. Для этого используется элемент *embed*, где наиболее востребованными являются атрибуты *src* (для прописывания местоположения музыкального файла) и *autostart* (автоматический запуск проигрывание). Для автоматического начала проигрывания необходимо выставить значение *true*, если же пользователь должен сам нажать на кнопку Старт, то значение будет *false*. Синтаксис: `<embed src = "music.wav" autostart = "true"></embed>`.



Часть *HTML* кода, для просмотра веб-страницы с музыкальным сопровождением.

```
<!DOCTYPE html>
<html>
<head>
<meta charset='windows-1251' />
<title>Пятая страница</title>
</head>
<body>
<h1>Добавление музыкального сопровождения</h1>
    Используя тэг <i>&lt;embed&gt;</i>, автор веб-страницы
    может добавить музыкальный фрагмент. При этом у
    посетителя остаётся возможность самому приостанавливать
    звучание, выбирать громкость. Здесь для примера
    используется очень короткое звучание.
    <br>
    <embed src="Example_Mixed.wav" autostart = "false" ></embed>
</body>
</html>
```

Посмотреть веб-страницу [[Доступно в учебных материалах курса](#)]

Оформление фона

В большинстве случаев у веб-страницы имеется фон - определённый цвет, фоновая иллюстрация и т.д. Для этого необходимо указать некоторые атрибуты для элемента *body*. Если будет использоваться однородный цвет, то для этого применяется атрибут *bgcolor*, где с помощью шестнадцатеричного кода или просто соответствующего названия определяется цвет. Синтаксис: `<body bgcolor = "#FFFF99"> ... </body>`. Конечно, не стоит использовать очень яркие, едкие цвета, потому что очень быстро глаза читателя устанут и он поспешит закрыть веб-страницу. Также в виде фона может выступать картинка

(например, логотип фирмы). Тогда используется атрибут *background* и сразу же указывается путь до соответствующего фото. Чтобы картинка при прокрутке веб-страницы не скользила, а была зафиксирована, используется атрибут *bgproperties* со значением *fixed*. Синтаксис: `<body background = "logo.jpg" bgproperties = "fixed"> ... </body>`.

Стоит учитывать, что при изменении фона, цвет текста тоже необходимо скорректировать. По умолчанию предложенный чёрный цвет не всегда может подойти для выбранного фоновое решение. Чтобы цвет текста поменялся для всей веб-страницы (глобально), необходимо воспользоваться атрибутом *text*, который будет располагаться внутри тэга *body*. Значения для этого атрибута задаются с помощью шестнадцатеричного кода или названия цвета. Синтаксис: `<body text = "brown"> ... </body>`. Однако, если при работе с каким-то словом или предложением понадобится изменить его цвет на ещё какой-то (выделить термин, обозначить важный момент), то следует воспользоваться тэгом `<font color = "..."> ... </font>`. Тогда для этой части текста будет применён указанный непосредственно перед началом предложения цвет.



Данный *HTML* код демонстрирует использование фона для веб-страницы с глобальным изменением цвета для текста.

```
<!DOCTYPE html>
<html>
<head>
<meta charset='windows-1251' />
<title>Шестая страница</title>
</head>
<body bgcolor = "FFFFAA" text = "brown">
<h1>Изменение фона и цвета текста</h1>
    фон может быть задан либо цветовой гаммой,
    либо с помощью фотографии. Автор веб-
    страницы обязательно должен обратить
    внимание на используемый цвет текста,
    чтобы читателю было легче изучить
    предложенный материал.
</body>
</html>
```

Посмотреть веб-страницу [Доступно в учебных материалах курса]



В любом своём *HTML* документе поменяй фон с белого на любой другой. При необходимости измени цвет текста. По возможности добавь музыкальное сопровождение.

Работа со списком

Для предоставления, например, пошаговой инструкции или перечислении основных моментов может быть использован список. Рассмотрим два варианта, когда перед словом, фразой будет располагаться отметка или порядковый элемент.

Маркированный список

При работе с таким типом списка перед словом будет указан какой-то символ. Неупорядоченный список (где не используются порядковые номера, буквы) на английском языке будет *unordered list*, отсюда же и происходит название тэга `<ul>`, которым открывается список. Внутри этого тэга располагается тэг `<li>`, за которым и пишется необходимое ключевое слово или фраза списка. При желании пользователь может выбрать, какой символ будет использоваться. Для этого существует атрибут *type*, значения которого могут быть следующими:

- *square* (рус. яз. квадрат);
- *disc* (рус. яз. кружок);
- *circle* (рус. яз. окружность).

На самом деле вид символа может быть написан как в начале списка рядом с элементом *ul*, так и прописываться отдельно для каждого ключевого слова отдельно рядом с элементом *li*. Синтаксис: `<ul type = "circle"> <li> ... </li> <li> ... </li> </ul>`.

Нумерованный список

Для работы с упорядоченным, нумерованным типом списка необходимо использовать также два тэга: для начала списка `<ol>` (англ. яз. *ordered list*) и указания элемента списка `<li>`. Так же как и для маркированного списка с помощью атрибута *type* можно выбрать внешний вид элемента:

- 1 - для использования арабских цифр (1, 2, 3);
- I - для использования больших римских цифр (I, II, III);
- i - использование малых римских цифр (i, ii, iii);
- A - для нумерации большими прописными буквами (A, B, C);
- a - для нумерации маленькими строчными буквами (a, b, c).

При желании можно определить стартовую позицию списка. Для этого предусмотрен атрибут *start*, значение которого и обозначает начальную точку отсчёта. Синтаксис: `<ol start = "3">`.

Общий синтаксис для нумерованного списка выглядит так: `<ol type = "I"> <li> ... </li> <li> ... </li> </ol>`.

При работе с любым из типов списков после символа (диск, буква A и др.) может последовать не только слово, предложение, но и целые абзацы. Вся эта текстовая информация может быть оформлена любым из ранее изученным способом. Это означает, что можно изменять цвет, размер, использовать гиперссылки и т.д.

Для создания длинного перечня можно использовать комбинацию из маркированного и нумерованного списков. В этом случае необходимо быть очень внимательным, чтобы закрывающие тэги были расставлены верно.



Ниже приведённый *HTML* код позволяет посмотреть в действии работу со списками.

```

<!DOCTYPE html>
<html>
<head>
<meta charset='windows-1251' />
<title>Седьмая страница</title>
</head>
<body>
<h1>Работа со списками</h1>
<font color = "green">Маркированный список (устройства ввода информации)</font>
<br/>
<ul type = "circle">
<li>клавиатура</li>
<li>сканер</li>
<li>веб-камера</li>
</ul>
<font color = "blue">Нумерованный список (сохранение файла)</font>
<ol type = "1">
<li>перейти к меню файл</li>
<li>выбрать команду Сохранить файл как</li>
<li>указать место сохранения файла</li>
<li>дать имя файлу</li>
<li>нажать кнопку Сохранить</li>
</ol>
<br/>
</body>
</html>

```

Посмотреть веб-страницу [Доступно в учебных материалах курса]



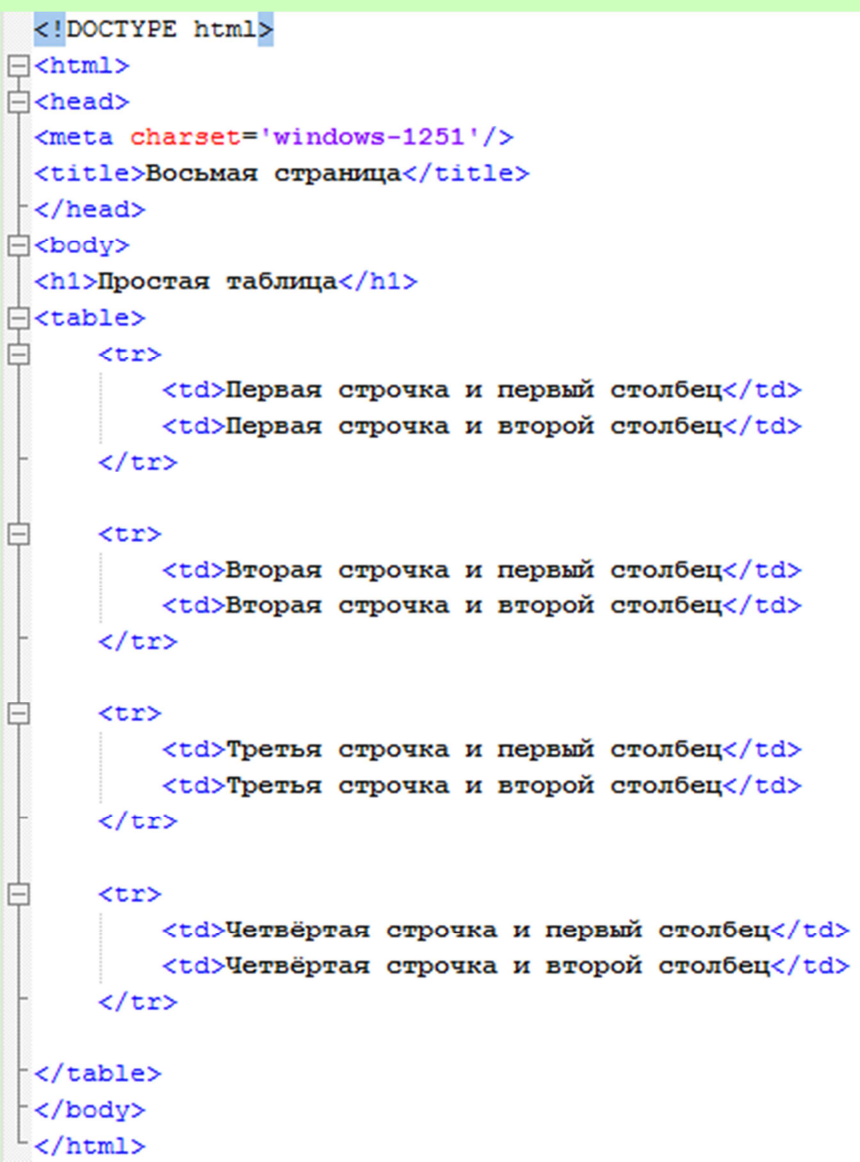
Свой *HTML* документ дополни нумерованным списком с указанием предметов, которые у тебя идут в школе по понедельникам. Маркированным списком запиши 4 - 5 ключевых фразы, которые у тебя ассоциируются с *HTML* языком.

Работа с таблицей

Иногда бывает необходимо разместить данные в таблице. Для этого используется общая пара тэгов `<table> ... </table>`, дальше необходимо определиться с количеством столбцов и строк. С помощью пары тэгов `<tr> ... </tr>` определяется строка, а для столбцов применяется пара тэгов `<td> ... </td>`. Стоит обратить внимание, что в языке *HTML* принято внутри строки указывать столько тэгов `<td> ... </td>`, сколько будет столбцов внутри данной строки. Схематично это можно представить как *начало таблицы* {строка[столбец1 столбец2 столбец3 ...], строка[столбец1 столбец2 столбец3 ...], ...} *конец таблицы*. Таким образом получается, что создание таблицы идёт по строчкам (создать строку - определиться со столбцами). На языке *HTML* синтаксис для приведённого выше примера будет иметь вид: `<table> <tr> <td> ... </td><td> ... </td><td> ... </td> </tr> <tr> <td> ... </td><td> ... </td><td> ... </td> </tr> </table>`.



Пример простой таблицы из 4 строчек и 2 столбцов, созданной с помощью *HTML* языка.



```

<!DOCTYPE html>
<html>
<head>
<meta charset='windows-1251' />
<title>Восьмая страница</title>
</head>
<body>
<h1>Простая таблица</h1>
<table>
<tr>
<td>Первая строка и первый столбец</td>
<td>Первая строка и второй столбец</td>
</tr>
<tr>
<td>Вторая строка и первый столбец</td>
<td>Вторая строка и второй столбец</td>
</tr>
<tr>
<td>Третья строка и первый столбец</td>
<td>Третья строка и второй столбец</td>
</tr>
<tr>
<td>Четвёртая строка и первый столбец</td>
<td>Четвёртая строка и второй столбец</td>
</tr>
</table>
</body>
</html>

```

Посмотреть веб-страницу [Доступно в учебных материалах курса]

Как видно из приведённого выше примера по умолчанию таблица строится без границ, но их толщину в пикселях можно задать с помощью атрибута *border*. Синтаксис: `<table border = "2"> ... </table>`. Атрибут *bordercolor* позволит определиться с цветом для рамки. Синтаксис: `<table bordercolor = "purple"> ... </table>`.

Для таблицы, строчек и столбцов существует ряд общих атрибутов, которые позволяют их оформить:

- *width* задаёт ширину для таблицы в пикселях или процентах. Синтаксис: `<table width = "15%"> ... </table>` или `<td width = "150"> ... </td>`;
- *height* задаёт высоту таблицы также в пикселях или процентах. Синтаксис: `<table height = "250"> ... </table>` или `<tr height = "7%"> ... </tr>`;
- *align* позволяет определить выравнивание текста по центру (англ.яз. *center*), левому (англ.яз. *left*) или правому краю (англ.яз. *right*). Синтаксис: `<table align = "center"> ... </table>`;

- *bgcolor* для определения фона. Синтаксис: `<table bgcolor = "#FFFFCC"> ... </table>` или `<tr bgcolor = "magenta"> ... </tr>`.

В случае необходимости "объединить" несколько ячеек можно воспользоваться атрибутами для столбцов:

- *colspan* позволяет определить, сколько столбцов необходимо между собой соединить, т.е. сколько места будет занимать одна большая ячейка. Синтаксис: `<td colspan = "3"> ... </td>`.
- *rowspan* задаёт количество строк, которые будет занимать объединённая ячейка. Синтаксис: `<td rowspan = "2"> ... </td>`.

Для определения заголовка для каждого столбца может быть использована пара тэгов `<th> ... </th>`. Текст такой ячейки обычно выделяется жирным шрифтом и располагается по центру.

Всё содержание таблицы может быть оформлено как было изучено ранее (цвет текста, размер шрифта) и содержать фотографии, гиперссылки, списки и т.д.



Пример использования различных изученных тэгов данной главы при работе с таблицей.

```

<!DOCTYPE html>
<html>
<head>
<meta charset='windows-1251' />
<title>Девятая страница</title>
</head>
<body>
<h1>Работа с таблицей</h1>
<table border = "3" bordercolor = "blue" width = "80%">

  <tr bgcolor = "yellow">
    <th width = "40%">Позитивные моменты</th>
    <th width = "40%">Отрицательные моменты</th>
  </tr>

  <tr>
    <td>Возможность предоставления информации с помощью таблицы</td>
    <td>Трудность в отслеживании порядка тэгов</td>
  </tr>

  <tr>
    <td>Возможность задать цвета линии и фона</td>
    <td>Сложность при объединении ячеек</td>
  </tr>

  <tr>
    <td colspan = "2" align = "center">
      <font color = "green">Ура :) Объединённая ячейка создана!</font><br/>

      <img src = "http://farm5.staticflickr.com/4032/4268841912_c2b34ca43c.jpg"
      alt = "HTML" title = "Огненный HTML" width = "15%" height = "10%" /> <br/>

      <small><i>Источник:
        <a href = "http://www.flickr.com/photos/nidhug/4268841912/"
        target = "_blank">
          http://www.flickr.com/photos/nidhug/4268841912/ </a></i>
      </small>
    </td>
  </tr>

</table>
</body>
</html>

```

Посмотреть веб-страницу [Доступно в учебных материалах курса]

Таблицы являются не единственным способом разграничить расположение материала на веб-странице. Можно сделать так, чтобы с левой стороны постоянно находилось что-то наподобие содержания, а с правой стороны было бы видно содержание выбранной темы. Для этого используются фреймы. Подробнее о них можно почитать [здесь](#).



С помощью *HTML* языка составь таблицу с твоим расписанием уроков в школе.

Работа с формой

Работа с формами позволяет наладить своего рода обратную связь с посетителями сайта. Читатель может записать комментарий в текстовое поле, выбрать из предложенных вариантов любимый продукт питания, нажать кнопку и т.д. Данные будут отправлены на сервер для последующей обработки. Этот процесс выходит за рамки данного курса, поэтому мы рассмотрим только принципы составления форм.

Для использования формы предусмотрена пара тэгов `<form> ... </form>`. Внутри них будут располагаться различные способы ввода, обозначенные элементом *input*. Для каждого контроля необходимо использовать атрибут *name*, который позволяет отличать одну группу контролов от другой, а также участвует при отправке данных на сервер для обработки. Вторым обязательным атрибутом является *type*, который и позволяет определиться со способом ввода, выбрать контрол. Наиболее популярными являются следующие типы:

- *text* позволяет посетителю сайта ввести в ячейку информацию. При этом можно задать ширину ячейки (атрибут *size*) и максимально разрешённое количество вводимых символов (атрибут *maxlength*). Синтаксис (результат приведён на рисунке 1): `<input type = "text" name = "language" size = "20" maxlength = "4">`.

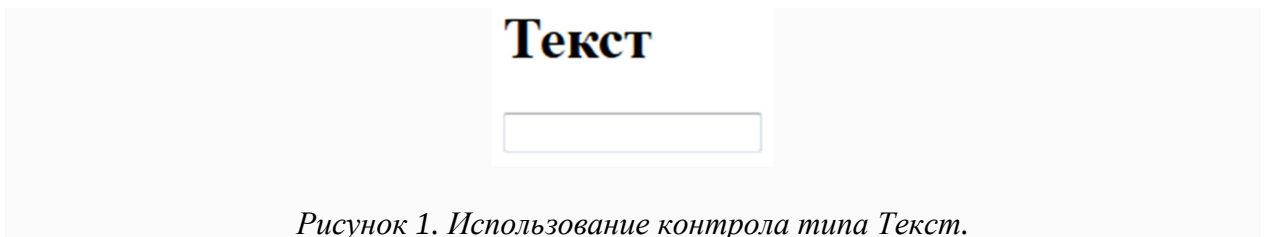


Рисунок 1. Использование контрола типа Текст.

- *password* используется исключительно для ввода пароля. Его отличие от текста заключается в том, что вместо букв, выводятся точки (чтобы никто не увидел вводимый пользователем пароль). Синтаксис: `<input type = "password" name = "parool">`;
- *radio* используется для выбора только одного значения из нескольких. Здесь атрибут *name* имеет для одной группы одинаковое значение, т.к. происходит выбор одного единственного ответа из всей группы. Сам вариант выбора задаётся с помощью атрибута *value*. Синтаксис (результат см. на рисунке 2):

```
<input type = "radio" name = "tags" value = "img"> img <br/>
<input type = "radio" name = "tags" value = "font"> font <br/>
<input type = "radio" name = "tags" value = "form"> form <br/>
<input type = "radio" name = "tags" value = "table"> table <br/>
```

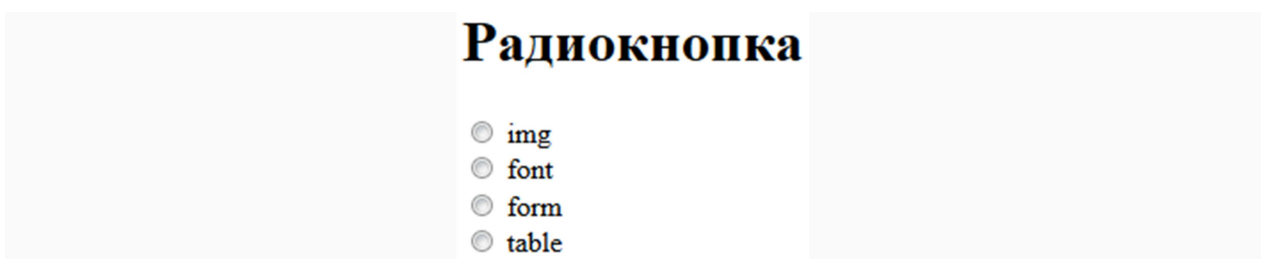


Рисунок 2. Использование контрола типа Радиокнопка.

- *checkbox* позволяет выбрать несколько или все значения, варианты ответов на один вопрос. Здесь также присутствуют атрибуты *name* и *value*, которые выполняют те же функции, что и для радиокнопок. Синтаксис (результат см. на рисунке 3):

```
<input type = "checkbox" name = "attributes" value = "color"> color <br/>
<input type = "checkbox" name = "attributes" value = "size"> size <br/>
<input type = "checkbox" name = "attributes" value = "face"> face <br/>
<input type = "checkbox" name = "attributes" value = "border"> border <br/>
```

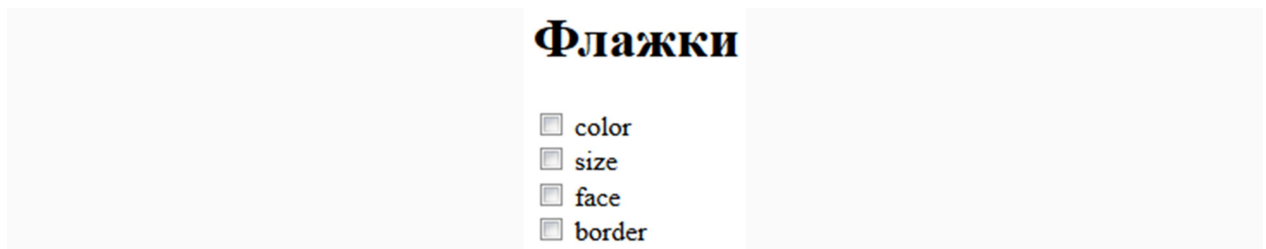


Рисунок 3. Использование контрола типа Флажок.

Если необходимо ввести довольно длинный комментарий, то логичнее использовать `<textarea>`, где с помощью атрибутов *cols* и *rows* можно задать количество столбцов и строк, соответственно. Синтаксис: `<textarea name = "comments" cols = "15" rows = "10">`.

Если пользователь должен выбрать что-то одно из выпадающего раскрывающегося списка, то для этого используется тэг `<select>`. В таком случае при щелчке на список перед пользователем имеется несколько вариантов, он щёлкает на подходящий и список сворачивается, но при этом активным, видимым становится выбранный вариант. Варианты выбора задаются с помощью элемента *option* с обязательным закрывающим тэгом и атрибутом *value*. Синтаксис (результат см. на рисунке 4):

```
<select name = "list">
<option value = "ol"> ol </option>
<option value = "table"> table </option>
<option value = "font"> font </option>
<option value = "img"> img </option>
</select>
```

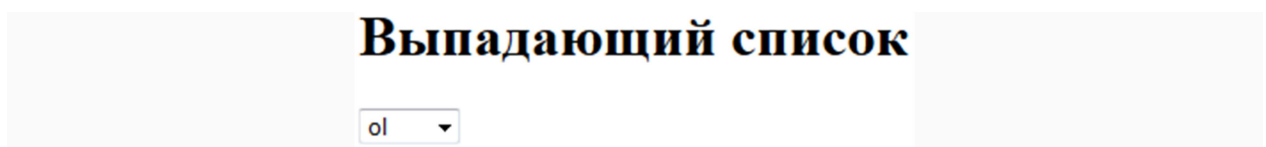


Рисунок 4. Использование контрола типа Выпадающий список.

Также могут использоваться кнопки различных видов:

- *submit* для отправки данных заполненной формы. Синтаксис: `<input type = "submit" value = "Send">`;
- *reset* для очистки, сброса всех данных. Синтаксис: `<input type = "reset" value = "Clear">`.



HTML код для изучения мнения посетителя о веб-странице.

```

<!DOCTYPE html>
<html>
<head>
<meta charset='windows-1251' />
<title>Десятая страница</title>
</head>
<body text = "darkgreen">
<h1>Работа с формой</h1>
<form>
Ваше имя
  <input type = "text" name = "ClientName" size = "20" maxlength = "50"> <br/>
Вы первый раз на нашей веб-странице? <br/>
  <input type = "radio" name = "FirstTime" value = "yes"> да <br/>
  <input type = "radio" name = "FirstTime" value = "no"> нет <br/>
Что Вам понравилось на нашей странице? <br/>
  <input type = "checkbox" name = "Likes" value = "text"> текст материала <br/>
  <input type = "checkbox" name = "Likes" value = "video"> видеоматериалы <br/>
  <input type = "checkbox" name = "Likes" value = "audio"> аудиоматериалы <br/>
  <input type = "checkbox" name = "Likes" value = "nothing"> ничего <br/>
Выберите самую интересную на Ваш взгляд тему. <br/>
  <select name = "Topic">
    <option value = "choose"> выберите тему </option>
    <option value = "image"> работа с изображением </option>
    <option value = "table"> работа с таблицей </option>
    <option value = "form"> работа с формой </option>
  </select><br/>
Пожалуйста, напишите Ваши пожелания. <br/>
  <textarea name = "wishes" cols = "50" rows = "10"></textarea><br/>
  <input type = "submit" value = "Send">
</form>
</body>
</html>

```

Посмотреть веб-страницу [Доступно в учебных материалах курса]



Дополни свой веб-документ формой, содержащей несколько контролов.

Обработка событий

Некоторые действия, события на веб-странице могут быть результатом деятельности человека. Создаётся некая интерактивность: посетитель навёл курсором мышки на картинку - появилось окно-сообщение, пользователь начал вводить данные в форму - всплыла подсказка и т.д. Таким образом, сначала необходимо определиться с событием, а потом с его следствием. Событие будет являться атрибутом элемента гиперссылки, которая ссылается на саму себя. Общий синтаксис будет иметь вид: `<a href = "#" событие-действие>`.

Некоторые события, которые могут запустить действие:

- *onkeydown* - нажатая клавиша клавиатуры;
- *onclick* - щелчок мышки;
- *onmouseover* - курсор мышки окажется над словом/изображением.

С полным перечнем событий можно ознакомиться в [этой статье](#).

Действия, которые могут быть запущены:

- *popup* - всплывающее окно;
- *alert* - предупреждение;
- *function ()* - прописанная ранее функция.



Ниже приведено несколько примеров работы с событиями.

```
<!DOCTYPE html>
<html>
<head>
<meta charset='windows-1251' />
<title>Одиннадцатая страница</title>
</head>
<body>
<h1>События и действия</h1>

Пользователь должен <font color = "red">навести мышкой </font> на слово
или изображение и появится окно-предупреждение<br/><br/>

<a href="#"
  onmouseover="alert ('Действие на событие')">Наводка мышкой на слово
</a><br/>

<p align = "center">
  <a href="#" onmouseover="alert ('Действие на событие')">
    <img src = "Model.jpg" width = "20%" height = "15%">
  </a>
</p><br/>

Пользователь нажмёт на слово или фотографию и появится интернет-страница
в окне с заданной шириной и высотой.<br/><br/>

<a href="#" onclick="Popup=window.open('http://www.google.com', 'Popup',
'width=700, height=300')">
Посмотреть веб-страницу
</a>
</body>
</html>
```

Посмотреть веб-страницу [[Доступно в учебных материалах курса](#)]

NB! Появление в примере нового окна со страницей Google может быть затруднено (окно в окне).

Полнофункциональное использование событий-действий возможно при использовании языка программирования *JavaScript*.

Валидация

Код написан человеком, для которого свойственно ошибаться. Заметить неточности, подправить код поможет валидатор (англ.яз. *validator*). Это программное обеспечение позволяет выявить ошибки в коде, сравнить используемые тэги со стандартами. Некоторые валидаторы способны даже подсказать как исправить ошибку, какой тэг лучше использовать. Как правило, такие программы позволяют проверить как всю веб-страницу целиком, так и небольшие кусочки программного кода.

Консорциум Всемирной паутины имеет собственный валидатор, который располагается по адресу <http://validator.w3.org>. Перейдя по этой ссылке, можно выбрать для проверки:

- опубликованную веб-страницу, указав её адрес;
- файл, который необходимо загрузить;
- часть кода, которую можно скопировать в рабочую область.

Когда все ошибки будут устранены, автор веб-страницы может быть уверен, что его материал будет отображён корректно в браузере.

Справочники

В двух книгах были рассмотрены основные тэги и их атрибуты. В мире языка разметки гипертекстов их, конечно же, намного больше. При желании, можете воспользоваться русскоязычными справочниками:

- Справочник ресурса htmlbook.ru;
- Справочник ресурса html.manual.ru;
- Справочник ресурса WebRemeslo.ru.

На англоязычной странице W3School также можно узнать информацию о различных тэгах, их атрибутах и возможных значениях.

НЕДЕЛЯ 5: КАСКАДНЫЕ ТАБЛИЦЫ СТИЛЕЙ CSS (I)

Предисловие

Название книги: Каскадные таблицы стилей CSS (I)

Основные цели работы с книгой:

- дать определения основным терминам;
- познакомить с синтаксисом стилей;
- описать разные способы включения стилей в веб-страницу;
- дать классификацию селекторам;
- изучить возможности оформления текстовой информации при помощи CSS.

Рассматриваемые темы:

- Каскадные таблицы стилей CSS;
- Подключение стилей;
- Синтаксис;
- Типы селекторов;
- Работа с текстом;
- Свойства шрифтов;
- Цвет;
- Что делать, если стили работают некорректно?

Результатом проработки данного материала является умение учащегося:

- использовать уместные термины при разговоре о каскадных таблицах стилей;
- целенаправленно выбирать и применять подключение каскадных таблиц на практике;
- быстро и эффективно оформлять текст веб-документа при помощи стилей;
- самостоятельно находить и исправлять ошибки форматирования.

Каскадные таблицы стилей CSS

На прошлых уроках речь шла о языке разметки гипертекста *HTML*. Его основная задача - логическая разметка текста (указать начало и конец заголовка, параграфа, блока; указать ссылку на фотографию, видео и т.д.); поэтому язык разметки гипертекста не предназначен на прямую для оформления веб-страниц. Косвенно, т.е. в ручную, можно задать цвет, размер, оформление и т.д. для каждого элемента отдельно (но не для всех разом). Добавляя один и тот же кусок *HTML*-кода в страничку, размер и время загрузки файла значительно увеличиваются.

Эти недостатки можно решить удобным и экономичным способом при помощи каскадных таблиц стилей, которые представляют собой альтернативу *HTML* тегам. **Каскадные таблицы стилей** (от англ. *Cascading Style Sheets*) - набор правил, описывающий внешний вид *HTML* документа. Набор правил, или **стиль**, применяется как одному элементу, так и ко всем одноименным элементам страницы. Таким образом, CSS может полностью контролировать работу элементов *HTML*-разметки.



Сравни два файла.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <meta charset='windows-1251' />
5    <title>HTML документ</title>
6  </head>
7  <body>
8    <h1>Оформление текста с помощью HTML</h1>
9    <font color = "brown"><b>
10      С помощью данного языка можно оформить текст по своему усмотрению.
11      Например, данный текст отображается коричневым цветом и размером 4. </b>
12    </font>
13    <br />
14    <p align = "center">
15      <font color = "red"><big>
16        Важно!
17      </big></font>
18      При использовании сразу нескольких тэгов порядок расположения закрывающих
19      тэгов имеет значение.
20    </p>
21    <font color = "brown"><b>
22      CSS позволяет упростить работу дизайнера при помощи единожды прописанных правил.
23    </font>
24  </body>
25  </html>
```

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4    <meta charset='windows-1251' />
5    <title>HTML документ</title>
6    <STYLE type="text/css">
7      h1{
8        color: black;
9      }
10     body{
11       color: brown;
12       font-weight: bold;
13     }
14     p{
15       text-align: center;
16       color: black;
17       font-weight: normal;
18     }
19     big{
20       color: red;
21     }
22   </style>
23 </head>
24 <body>
25   <h1>Оформление текста с помощью HTML</h1>
26     С помощью данного языка можно оформить текст по своему усмотрению.
27     Например, данный текст отображается коричневым цветом и размером 4.
28   <br />
29   <p>
30     <big>
31       Важно!
32     </big>
33     При использовании сразу нескольких тэгов порядок расположения закрывающих
34     тэгов имеет значение.
35   </p>
36   CSS позволяет упростить работу дизайнера при помощи единожды прописанных правил.
37 </body>
38 </html>

```

Наборы правил удобнее всего сохранять не в *HTML* документе, а в отдельном файле – это позволяет переиспользовать разработанные стили и применять документ к любому количеству веб-страниц. Более того, в подобный файл легче внести изменения, чем начинать работу с нуля. В отличие от *HTML*, изменение стиля *CSS*, правила которого прописаны в одном месте и применяются к нескольким веб-страницам, поможет избежать ошибок, сэкономить время на редактировании и сохранить единство всего документа.

Подключение стилей

Разработчик может подключить стили *CSS* к *HTML*-документу тремя способами:

1. внутри элемента *HTML*;
2. внутри блока `<head>` *HTML*-страницы;
3. в отдельном файле *CSS*.

Внутри элемента *HTML*

Стиль можно задать внутри самого элемента при помощи свойства **style**

```
<html>
<body>
  <span style="color:white;background-color:blue;font-variant:small-caps"> Белый текст на синем фоне, маленькими заглавными
буквами.</span>
</body>
</html>
```

БЕЛЫЙ ТЕКСТ НА СИНЕМ ФОНЕ, МАЛЕНЬКИМИ ЗАГЛАВНЫМИ БУКВАМИ.

Стоит отметить, что данный способ нарушает идеологию технологии (универсальность); применять его стоит в единичных случаях.

Внутри блока **<head>** *HTML*-страницы

Если заранее известно, что разрабатываемые правила будут использоваться только для создания одной страницы, а не целого портала, то их можно описать в заголовке **head** между тегами **<style type="text/css">** и **</style>**.

```
<html>
<head>
  <style type="text/css">
    h1 {text-align: center;
        color: red}
  </style>
</head>
<body>
  <h1>Этот заголовок имеет указанный выше стиль.</h1>
</body>
</html>
```

В отдельном файле *CSS*

Более универсальный способ оформления внешнего вида веб-страницы - это описать правила в отдельном файле и сохранить файл с расширением **.css** в той же папке, где и веб-страница. Для подключения стилевого файла к *html*-файлу пропишите тег **link** внутри **head**. В итоге получится два отдельных файла.

style.css	homepage.html
<pre> h1 {text-align: center; color: red} </pre>	<pre> <html> <head> <link rel="stylesheet" type="text/css" href="style.css"> </head> <body> <h1>Этот заголовок имеет указанный выше стиль</h1> </body> </html> </pre>

Синтаксис

Синтаксис CSS **правила** состоит из трех частей: селектора, свойства и значения:

```

селектор {
    свойство: значение;
}

```

Селектор — уникальное имя элемента или тег *HTML*, который определяет область применения правила. **Свойство** — общий атрибут, который следует изменить в документе (например, гарнитура, размер, ширина таблицы). Каждому свойству присваивается значение. Существует ряд рекомендаций, которым необходимо следовать при создании стилей.

1. Свойство и значение - объявление - разделяются двоеточием и помещаются внутри фигурные скобки: `p {color: white; }`
2. Если значение свойства состоит из нескольких слов, то оно помещается в кавычки: `h1 {font-family: "lucida calligraphy"}`
3. Если требуется определить более одного свойства, то необходимо разделить свойства точкой с запятой:

```

color: white;
background: blue;

```
4. Чтобы определения стилей было удобно читать, каждое свойство следует писать на отдельной строке, используя табуляцию:

```

h2{
    font-family: arial;
    margin-right: 20pt;
    color:#ffffff
}

```
5. При определении правил допускается группировка селекторов при помощи запятой.

```

p,table{
    font-size: 10px;
}

```

Типы селекторов

Существует несколько типов селекторов, у каждого из них свое предназначение и функции:

1. универсальные селекторы
2. селекторы элементов
3. селекторы классов
4. селекторы идентификаторов
5. селекторы псевдоклассов
6. селекторы псевдоэлементов
7. селекторы атрибутов
8. селекторы сестринских элементов
9. селекторы потомков (контекстные селекторами)
10. селекторы дочерних элементов

В данной лекции мы рассмотрим четыре первых типа; еще два типа будут описаны в следующей лекции.

Универсальные селекторы

Универсальный селектор позволяет применить стилевое оформление сразу ко всем элементам веб-страницы. Селектор обозначается звездочкой.

```
* {
    text-align: center;
}
```

Селекторы элементов

В данном случае в качестве имен селекторов используют имена самих *HTML* тегов для того, чтобы уточнить оформление или переопределить форматирование.

```
b {
    font-weight: normal;
    font-style: italic;
}
```

*Этот текст написан при помощи тэга *

Этот пример показывает, что стиль отображения элементов может быть полностью изменен при помощи *CSS*, а технологии *HTML*-разметка носит декларативный характер.

Селекторы классов

Если нужно определить стиль так, чтобы один и тот же тег *HTML* в разных случаях отображался по-разному, то на помощь приходят **классы**. Класс описывается в виде `имя_класса {свойства}`. Класс включается в тег свойством `class="имя_класса"`. Это можно продемонстрировать на следующем примере:

```

<head>
  <style type="text/css">
    p {
      text-align: center;
      color: green
    }

    .author {
      text-align: justify;
      color: blue;
    }

    p.red{
      text-align: left;
      color: red;
    }

    p.black{
      text-align: left;
      color: black;
    }
  </style>
</head>
<body>
  <p>Этот параграф оформлен селектором элементов.</p>
  <p class="red">Этот параграф оформлен заданным селектором
классов.</p>
  <p class="black"> Этот параграф оформлен заданным селектором классов
  <i class="author"> и содержит универсальный </i> селектор классов.
  </p>
  <p class="author"> Универсальный селектор классов можно применить к
любому тегу.</p>
</body>
</html>

```

Этот параграф оформлен селектором элементов.

Этот параграф оформлен заданным селектором классов.

Этот параграф оформлен заданным селектором классов *и содержит универсальный* селектор классов.

Универсальный селектор классов можно применить к любому тегу.

Селекторы идентификаторов

Селекторы идентификаторов в какой-то мере похожи на селекторы классов, но есть несколько существенных отличий.

```
<head>
  <style type="text/css">
    #paks {font-weight: bold;}
  </style>
</head>
<body>
  <p id="paks">Этот параграф будет выделен полужирным
  шрифтом.</p>
  <p> Этот параграф НЕ будет выделен полужирным шрифтом.</p>
  <p id="paks">Дважды использовать этот id НЕ СЛЕДУЕТ,
  если именно к этому (третьему) параграфу надо будет
  обратиться при
  написании программного кода.</p>
</body>
</html>
```

Этот параграф будет выделен полужирным шрифтом.

Этот параграф НЕ будет выделен полужирным шрифтом.

Дважды использовать этот id НЕ СЛЕДУЕТ, если именно к этому (третьему) параграфу надо будет обратиться при написании программного кода.

Во-первых, предполагается, что селектор идентификатора выберет только **один элемент**. Это очень важное условие при написании скриптов (программ или приложений) - возможность обратиться и выбрать значение элемента по его **уникальному** идентификатору. Во-вторых, перед селекторами идентификаторов вместо точки ставится «решетка» (#). В-третьих, селектор идентификатора включается в тег свойством **id**. **NB!** Если не имеешь дела со скриптами, можно вполне ограничиться использованием селекторов класса.

Работа с текстом

В этой главе речь пойдет о стилях CSS, которые чаще всего применяются к форматированию текста.

Выравнивание текста

Для того чтобы выровнять текст в *HTML*, мы применяли к тегу атрибут `align`(выравнивание) и одно из его возможных значений, например, `center`(по центру): `<p align="center">текст по центру</p>`.

В CSS данную задачу берет на себя свойство `text-align`, которое выравнивает текст относительно окна браузера или элемента родителя (например, блока `div`) в зависимости от значения:

- **left** - выровнять текст по левому краю элемента (по умолчанию);
- **right** - выровнять текст по правому краю;
- **center** - выровнять текст по центру;
- **justify** - выровнять текст по обоим краям.

```
<head>
  <style type="text/css">
    p {text-align: justify;}
  </style>
</head>
<body>
<p>Этот параграф выровнен по
правому боку.</p>
</body>
</html>
```

Этот параграф выровнен по правому боку.

Декорация текста

Свойство **text-decoration** позволяет украсить текст следующими свойствами:

- **blink** - текст мигает;
- **line-through** - перечеркнутый текст ;
- **overline** - надчёркивание текста;
- **underline** - подчеркивание текста;
- **none** - текст без оформления.

Последнее значение очень часто используют для ссылок. По умолчанию ссылка подчеркивается снизу. Для того чтобы убрать этот декор, свойству **text-decoration** присваивается значение **none**.

```
<head>
  <style type="text/css">
    .minu_stiil {text-decoration:none;}
  </style>
</head>
<body>
<a href="http://www.ut.ee">Ссылка по
умолчанию.</a>
<br />
<a href="http://www.ut.ee"
class="minu_Stiil">Ссылка без декора</a>
</body>
</html>
```

Ссылка по умолчанию.
Ссылка без декора

Трансформация текста

Свойство **text-transform** преобразует символы в указанном текстовом блоке, делая их заглавными или прописными по одному из следующих правил:

- **none** - текст отображается без каких-либо изменений (по умолчанию) ;
- **capitalize** - каждое слово в тексте отображается с заглавного символа;
- **lowercase** - все символы преобразуются в нижний регистр;

- **uppercase** - все символы преобразуются в верхний регистр.

```
<head>
  <style type="text/css">
    p {style-transform:
uppercase}
  </style>
</head>
<body>
<p>Все символы
преобразуются в верхний
регистр.</p>
</body>
</html>
```

ВСЕ СИМВОЛЫ ПРЕОБРАЗУЮТСЯ В ВЕРХНИЙ РЕГИСТР.

Еще немного трансформаций

Базовая линия - это линия, на которой располагаются символы в текстовой строке. Например, буква "А" сидит прямо на этой линии, а вот строчная буква "у" сидит на ней же, но свесив ноги..



Источник: http://usabili.ru/news/2010/02/24/cross-browser_inlyine-block.html

Расположение букв в тексте можно изменить при помощи свойство **vertical-align** следующим образом:

- **baseline** - выравнивает базовую линию элемента по базовой линии родителя;
- **bottom** - выравнивает элемент по нижней части строки;
- **middle** - выравнивает середину элемента по базовой линии родителя и прибавляет половину высоты родительского элемента;
- **sub** - нижний индекс (размер шрифта не меняется);
- **super** - верхний индекс (размер шрифта не меняется);
- **text-bottom** - нижняя граница элемента выравнивается по нижнему краю строки;
- **text-top** - верхняя граница элемента выравнивается по верхнему краю строки;
- **top** - выравнивает элемент по верхней части строки.

Все значения задаются в процентах (%), пикселях (px), точках (pt), сантиметрах (cm), миллиметрах (mm) или любых других принятых в CSS единицах измерения, причем эти единицы могут принимать как положительные, так и отрицательные значения.

```

<html>
<head>
</head>
<body>
<font size="+3">А и Б </font>
<span style="vertical-align: +5px">сидели на трубе </span>
<span style="vertical-align: bottom">А упало </span>
<span style="vertical-align: top">Б пропало.. </span>
<span style="vertical-align: 50%">что осталось на
трубе?</span>
<hr>
формула воды: Н<span style="vertical-align: sub">2</span>О
<hr>
<span>н</span>
<span style="vertical-align: -10px">а</span>
<span style="vertical-align: -20px">и</span>
<span style="vertical-align: -30px">с</span>
<span style="vertical-align: -40px">к</span>
<span style="vertical-align: -50px">о</span>
<span style="vertical-align: -60px">с</span>
<span style="vertical-align: -70px">о</span>
<span style="vertical-align: -80px">к</span>
</body>
</html>

```

А и Б сидели на трубе А упало Б пропало.. что осталось на трубе?

формула воды: H₂O

н а и с к о с о к

Красная строка

Свойство **text-indent** задаёт отступ первой строки в текстовом блоке с левой стороны. Расстояние от левого края окна браузера или же элемента родителя может быть заданно в процентах от ширины окна браузера (%), в пикселях(px),сантиметрах (cm) или же других единицах измерения принятых в CSS.

```
<html>
<head>
  <style type="text/css">
    p {text-indent: 3cm;}
  </style>
</head>
<body>
<p>Демонстрация красной
строки.</p>
</body>
</html>
```

Демонстрация красной строки.

Пробелы и перенос строки

Обычно браузеры выводят текст на экран в виде сплошного текста, автоматически расставляя переносы строк и убирая лишние пробелы. Свойство **white-space** похож на тег **<pre>**, определяя, показывать или нет пробелы между символами, если таковых больше чем один, а так же разрешает или запрещает перенос строки.

- **normal** - лишние пробелы убираются, переносы строк определяются автоматически (значение по умолчанию);
- **nowrap** - запрещается автоматический перенос строки;
- **pre** - показывает текст в том виде, в котором он был набран; пробелы и переносы строки не удаляются.

```
<head>
  <style type="text/css">
    p {white-space: pre}
  </style>
</head>
<body>
<p>Текст показан так,
                               как был набран.</p>
</body>
</html>
```

Текст показан так,
как был набран.

Расстояние между словами

Свойство **word-spacing** задаёт расстояние между словами.

- **normal** - стандартное расстояние между словами (по умолчанию);
- **px** - расстояние задаётся в пикселях.

```
<head>
  <style type="text/css">
    .pos {word-spacing: 20px}
    .neg {word-spacing: -20px}
  </style>
</head>
<body>
<p class="pos">Положительное
значение</p>
<p class="neg">Отрицательное
значение</p>
</body>
</html>
```

Положительное значение

Отрицательное значение

Расстояние между символами

А вот свойство **letter-spacing** определяет расстояние между символами в тексте.

- **normal** - стандартное расстояние (по умолчанию);
- **px** - расстояние задаётся в пикселях.

```
<head>
  <style type="text/css">
    .pos {letter-spacing: 20px}
    .neg {letter-spacing: -2px}
  </style>
</head>
<body>
<p class="pos">Положительное значение</p>
<p class="neg">Отрицательное значение</p>
</body>
</html>
```

П о л о ж и т е л ь н о е з н а ч е н и е

Отрицательное значение

Интерлиньяж

Интерлиньяж - это расстояние между строками текста. Его можно задать свойством **line-height**:

- **normal** - стандартное значение по умолчанию;
- **%** - расстояние задается в процентах; за сто процентов берется высота шрифта;
- **0.5** - множитель (любое число больше нуля);
- **px** - расстояние в пикселях.

```
<head>
  <style type="text/css">
    .one_half {line-height: 150%}
    .half {line-height: 0.5}
  </style>
</head>
<body>
<p class="one_half">Первая строка. <br /> Вторая
строка.
</p>
<p class="half">Третья строка. <br /> Четвертая
строка.</p>
</body>
</html>
```

Первая строка.
Вторая строка.
Третья строка.
Четвертая строка.

Свойства шрифтов

В этой главе речь пойдет о свойствах шрифта: типе, начертании, размере, полужирности и гарнитурах.

Стиль шрифта

Свойство **font-style** определяет наклон текста:

- **normal** - обычный по умолчанию;
- **italic** - курсив;
- **oblique** - наклонный.

```
<html>
<head>
</head>
<body>
<p style="font-style: italic">Это
курсив.</p>
<p style="font-style: oblique">А это
наклонный текст.</p>
<p style="font-style: normal">И чем,
спрашивается, они отличаются?</p>
</body>
</html>
```

Это курсив.

А это наклонный текст.

И чем, спрашивается, они отличаются?

Курсив - это шрифт взятый из библиотеки шрифтов, а **наклонный** текст - это результат работы алгоритма, где каждый символ слегка наклоняется в правую сторону.

Начертание шрифта

Свойство шрифта **font-variant** делает строчные буквы заглавными и уменьшенными:

- **normal** - нормальный по умолчанию;
- **small-caps** - все буквы заглавные и уменьшенные

```
<html>
<head>
  <style type="text/css">
    {font-variant: small-caps}
  </style>
</head>
<body>
<p>Демонстрация font-variant</p>
</body>
</html>
```

ДЕМОНСТРАЦИЯ FONT-VARIANT

Размер шрифта

Свойство **font-size** определяет размер шрифта. Размер шрифта может быть задан в процентах, пикселях или в любых других единицах измерения CSS.

- **xx-small** - очень очень маленький;
- **x-small** - очень маленький;
- **small** - маленький;
- **medium** - средний;
- **large** - большой;
- **x-large** - очень большой;
- **xx-large** - очень очень большой;
- **larger** - больше, чем размер шрифта родительского элемента;
- **smaller** - меньше, чем размер шрифта родительского элемента.

```
<html>
<head>
  <style type="text/css">
    .large {font-size: large;}
    .xxsmall {font-size: xx-small;}
    .pixels {font-size: 24px;}
  </style>
</head>
<body>
<p>
<span class="large">Демонстрация</span>
<span class="pixels"> размера</span>
<span class="xxsmall"> шрифтов.</span>
</p>
</body>
</html>
```

Демонстрация размера шрифтов.

Полужирность шрифта

Свойство **font-weight** определяет полужирность шрифта:

- **normal** - обычный шрифт;
- **bold** - полужирный шрифт;
- **bolder** - жирный шрифт относительно шрифта элемента родителя;
- **lighter** - тонкий шрифт относительно шрифта элемента родителя;
- **100...900** - условное числовое значение.

```
<html>
<head>
  <styletype="text/css">
    .bold {font-weight: bold;}
    .number {font-weight: 300;}
  </style>
</head>
<body>
<p>
<span class="bold">Демонстрация</span>
<span class="number"> полужирности.</span>
</p>
</body>
</html>
```

Демонстрация полужирности.

Семейство шрифта

Атрибут **font-family** - указывает на шрифт или гарнитуру. Указывать можно до трех гарнитур через запятую на случай, если на компьютере пользователя отсутствует нужный шрифт. Например, запись в стилевом описании `p {font-family: "Times New Roman", Arial, Verdana;}` - будет указывать браузеру пользователя, что данный параграф следует писать шрифтом Times New Roman; а если его нет в библиотеке, то следует использовать шрифт Arial; и уж если и его нет, тогда шрифт Verdana. Браузеру можно указывать не только какой то конкретный шрифт, но и обозначить предпочтительное семейство шрифтов из перечисленных ниже возможных:

- **serif** - шрифты с засечками;
- **sans-serif** - рубленые шрифты;
- **cursive** - курсивные шрифты;
- **fantasy** - декоративные шрифты;
- **monospace** - моноширные шрифты.

```
<html>
<head>
  <style type="text/css">
    p {font-family: fantasy, monospace,
cursive;}
  </style>
</head>
<body>
<p>
<p>Демонстрация шрифта</p>
</body>
</html>
```

Демонстрация шрифта

В заключении о шрифтах...

Шрифт, являясь несколько особенным атрибутом, может определять одновременно сразу несколько параметров. Предположим, необходимо написать стилевое описание шрифта для тега `<p>`, который должен быть: курсивом, жирным, иметь размер 20 пикселей и использовать шрифт **Arial**. Это можно сделать всего одной строчкой:

```
p {font: italic bold 20px Arial}
```

Цвет

Цвет элемента

Для того, чтобы изменить цвет текст, воспользуйся свойством **color**. Цвет можно задать одним из трех способов:

- **#ff0000** - шестнадцатеричное значение цвета *RGB*;
- **red** - именное значение цвета;
- **RGB(255,0,0)** - значение цвета *RGB*.

```
<html>
<head>
  <style type="text/css">
    p {color: #00ff00;}
  </style>
</head>
<body>
<p>
<p>Демонстрация цвета</p>
</body>
</html>
```

Демонстрация цвета

Цвет фона элемента

Свойство **background-color** определяет цвет фона. Цвет задается аналогичными способами:

- **#ff0000** - шестнадцатеричное значение цвета *RGB*;
- **red** - именное значение цвета;
- **RGB(255,0,0)** - значение цвета *RGB*;
- **transparent** - прозрачный фон по умолчанию.

```
<html>
<head>
  <style type="text/css">
    p {background-color: #0000ff;}
  </style>
</head>
<body>
<p>
<p>Демонстрация маркера.</p>
</body>
</html>
```

Демонстрация маркера

Что делать, если стили работают некорректно?

CSS не очень сложный язык; но работая с ним, можно допустить ошибки или применить стили, которые некоторые браузеры не умеют интерпретировать. Как и с *HTML*, браузер не выдает сообщения об ошибке; он просто игнорирует их и отображает страницу по-своему. Найти ошибки не всегда просто. Возможно:

- браузер не поддерживает стиль;
- стиль применен неправильно ;
- опечатка в имени стиля;
- т.д.

Если стили работают некорректно, сделай следующее.

1. Убедись, что файл `.css` находится в той же папке, что и страница (или в папке, определенной в атрибуте `href`).
2. Посмотри страницу в разных браузерах (*Internet Explorer*, *Mozilla Firefox*, *Opera*, *Google Chrome*, *Safari*).
3. Убедись, что все открывающие теги имеют соответствующие закрывающие теги (`<> ... </>`). Браузер должен знать, когда начинать стиль и когда заканчивать.
4. Убедись, что стили прописаны между тегами `<style>` и `</style>` в секции `<head>` или добавлен в веб-страницу при помощи тег `<link>`: `<link rel=stylesheet type="text/css" href="styles.css">`.
5. Убедись, что каждое описание стиля использует следующий формат:


```
tag1, tag2 {
    svoitvo1 : znachenie;
    svoistvo2 :znachenie;
}
```

 Разрыв строки не важен, но всегда отделяй селекторы запятой, а описания – точкой с запятой.
6. Убедись, что теги *HTML* или атрибуты не подавляют определений стиля. Например, даже если ваша таблица стилей определила шрифт `Arial` для тега `<p>`, тег `<font>` может поменять настройки.
7. Помни, что *HTML* элементы наследуют настройки стиля из окружающих их тегов. Например, если установлено свойство `color` в теге `<table>` как синий цвет, то все заголовки, текст и абзацы таблицы станут синими (если не указаны отдельные правила для этих тегов).
8. Если используешь в теге `class`, убедись, что в таблице стилей использована точка, но не в атрибуте класса:


```
<style>
.stiil{color:blue;
    font-size:7pt
}
</style>
<p class="stiil">Использование стилей.</p>
```
9. Проверь правильность установок для таблицы стилей по адресу <http://jigsaw.w3.org/css-validator/validator.html.en>

НЕДЕЛЯ 6: КАСКАДНЫЕ ТАБЛИЦЫ СТИЛЕЙ CSS (II)

Предисловие

Название книги: Каскадные таблицы стилей CSS (II)

Основные цели работы с книгой:

- получить практические навыки по созданию списков, таблиц и отступов;
- усовершенствовать предыдущие практические наработки.

Рассматриваемые темы:

- Работа со списком;
- Работа с границами;
- Поля и отступы;
- Поплавки.

Результатом проработки данного материала является умение учащегося:

- использовать изученные элементы и их атрибуты для лучшего предоставления информации.

Работа со списками

В данной главе речь пойдет о списках. Для обучения понадобится базовая информация о тегах: `<ul>`, `<ol>`, `<li>` - изложенная в главе о HTML.

Вид маркера в списке

Как ты помнишь, в *HTML* вид маркера в списке определял атрибут `type`. В *CSS* данную задачу берёт на себя свойство `list-style-type`, которое имеет свои стандартные значения, определяющие вид маркера:

- `disc` - диск (по умолчанию для `<ul>`);
- `circle` - полый круг;
- `square` - квадрат;
- `decimal` - арабские цифры (по умолчанию для `<ol>`);
- `lower-roman` - строчные римские цифры;
- `lower-alpha` - строчные буквы;
- `upper-roman` - заглавные римские цифры;
- `upper-alpha` - заглавные буквы;
- `none` - маркер отсутствует.

```

<html>
<head>
  <style type="text/css">
    ul {list-style-type: upper-roman;}
  </style>
</head>
<body>
<ul>Демонстрация списка
<li>Один</li>
<li>Два</li>
<li>Три</li>
</ul>
</body>
</html>

```

Демонстрация списка
 I. Один
 II. Два
 III. Три

Стиль обтекания маркера списком

Свойство **list-style-position** указывает браузеру на то, где должен находиться текст относительно его маркеров. По умолчанию текст находится в стороне от маркеров, но можно сделать так, что они будут обтекаться текстом.

- **outside** - маркер находится в стороне от списка (по умолчанию);
- **inside** - маркер обтекается текстом.

```

<html>
<head>
  <style type="text/css">
    ul {list-style-type:
upper-roman;
      list-style-position:
inside;
    }
  </style>
</head>
<body>
<ul>Демонстрация списка
<li>Один<br />Один</li>
<li>Два<br />Два</li>
<li>Три<br />Три</li>
</ul>
</body>
</html>

```

Демонстрация списка
 I. Один
 Один
 II. Два
 Два
 III. Три
 Три

Полезные советы:

Список - это легкий и удобный инструмент для структуризации данных. Кроме того, список (наряду с заголовками, выделенным текстом и картинками) - это элемент, который привлекают взгляд посетителя страницы; поэтому следует пользоваться им умело и ненавязчиво. Свойства CSS позволяют тебе создать практически любой дизайн маркеров.

Работа с границами

В этой главе речь пойдет о том, как сделать с помощью CSS рамку - бордюр, вокруг того или иного элемента. В HTML эта задача лежала на плечах атрибута **border**, однако его

можно было применить далеко не к каждому тегу. В CSS эту задачу берёт на себя одноимённое свойство **border** и значительно расширяет круг возможностей при работе с границами.

Стиль границы

Если в *HTML* бордюр мог быть только в виде сплошной линии вокруг элемента, то в *CSS* свойство **border-style** может принимать одно из следующих значений:

- **none** - граница отсутствует (по умолчанию);
- **dotted** - граница из точек;
- **dashed** - пунктирная граница;
- **solid** - сплошная граница;
- **double** - двойная граница;
- **groove** - граница "бороздка";
- **ridge** - граница "гребень";
- **inset** - вдавленная граница;
- **outset** - выдавленная граница.

```
<html>
<head>
  <style type="text/css">
    p {border-style: dotted;}
  </style>
</head>
<body>
<p>Точечная граница</p>
</ul>
</body>
</html>
```

Точечная граница

Стиль бордюра может быть задан как для всех сторон элемента одновременно, так и для каждой его стороны отдельно в зависимости от того, сколько значений присвоено свойству **border-style**. Таковых значений может быть от одного до четырёх по числу сторон элемента.

```
<html>
<head>
  <style type="text/css">
    p {border-style: dashed double
  ;}
  </style>
</head>
<body>
<p>Пунктирная и двойная граница</p>
</ul>
</body>
</html>
```

Пунктирная и двойная граница

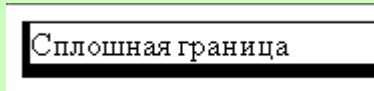
Толщина границы

Свойство **border-width** устанавливает ширину границы. Ширина бордюра может быть задана с помощью следующих аргументов:

- **thin** - тонкая граница;
- **medium** - средняя толщина границы;
- **thick** - толстая граница;
- а также в пикселях или любых других единицах измерения принятых в CSS (**cm**, **mm**, **px**, **pt**, **%**).

По аналогии со стилем, толщина бордюра тоже может иметь от одного до четырёх значений.

```
<html>
<head>
  <style type="text/css">
    p {
      border-style: solid;
      border-width: 1px 4px 7px;
    }
  </style>
</head>
<body>
<p>Сплошная граница</p>
</ul>
</body>
</html>
```



Сплошная граница

Цвет границы

Цвет рамки или её сторон по отдельности определяется свойством **border-color**. Цвет бордюра может иметь следующие значения:

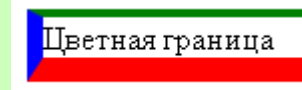
- **#ff0000** - шестнадцатеричное значение цвета *RGB*;
- **red** - именное значение цвета;
- **RGB(255,0,0)** - значение цвета *RGB*;
- **transparent** - прозрачная граница.

Ну и так же, как и в случаях с толщиной и стилем, цвет бордюра тоже может иметь от одного до четырёх цветовых значений.

```

<html>
<head>
  <style type="text/css">
    p {
      border-style: solid;
      border-width: 4px 8px 12px;
      border-color: #008000 RGB(0,0,255) red;
    }
  </style>
</head>
<body>
<p>Цветная граница</p>
</ul>
</body>
</html>

```



Границы справа слева сверху и снизу отдельно

Для того, чтобы определить стиль, цвет и ширину бордюра для одной из сторон, пользуйся свойствами **border-bottom, border-left, border-right, border-top** (и их дочерними атрибутами).

border-bottom - определяет стиль, цвет и ширину нижней границы элемента

- **border-bottom-color** - устанавливает цвет нижней границы элемента;
- **border-bottom-style** - определяет стиль нижней границы элемента;
- **border-bottom-width** - определяет ширину нижней границы элемента.

border-left - определяет стиль, цвет и ширину левой границы элемента

- **border-left-color** - устанавливает цвет левой границы элемента;
- **border-left-style** - определяет стиль левой границы элемента;
- **border-left-width** - определяет ширину левой границы элемента.

border-right - определяет стиль, цвет и ширину правой границы элемента

- **border-right-color** - устанавливает цвет правой границы элемента;
- **border-right-style** - определяет стиль правой границы элемента;
- **border-right-width** - определяет ширину правой границы элемента.

border-top - определяет стиль, цвет и ширину верхней границы элемента

- **border-top-color** - устанавливает цвет верхней границы элемента;
- **border-top-style** - определяет стиль верхней границы элемента;
- **border-top-width** - определяет ширину верхней границы элемента.

```

<html>
<head>
  <style type="text/css">
    p {
      border-bottom-style: solid;
      border-bottom-width: 12px;
      border-bottom-color: red;
    }

    div {
      border-top: 5px dashed green;
    }
  </style>
</head>
<body>
<p>Первый пример.</p>
<div>Второй пример.</div>
</ul>
</body>
</html>

```

Первый пример.



Второй пример.



Поля и отступы

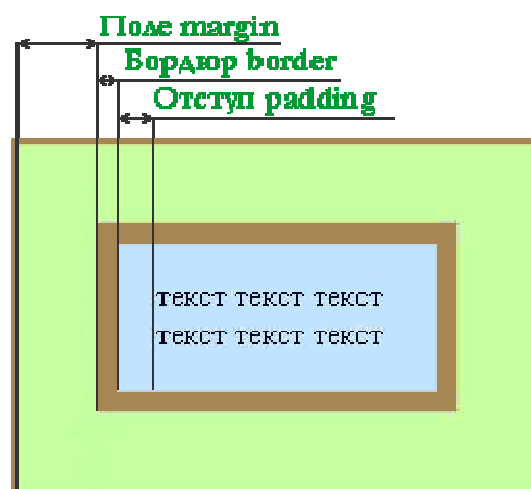
В этой главе речь пойдет о полях (свойство **margin**) и отступах (свойство **padding**). Эти два свойства весьма схожи, однако очень разные.

Свойство **margin** задает расстояние от внешней границы элемента до границы окна браузера или элемента родителя, т.е. границы того блока в который вложен элемент.

Отступ **padding** задает расстояние от внутренней границы элемента до его содержания (текста, картинки и т.д).

Расстояния **margin** и **padding** указываются:

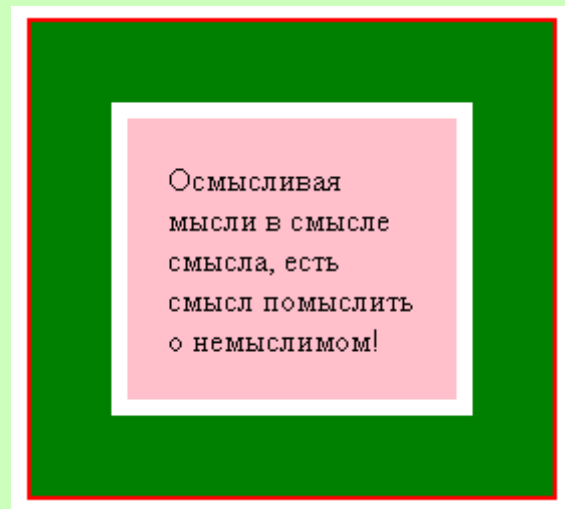
- **px** - в пикселях или любых других допустимых CSS единицах измерения;
- **%** - в процентах;
- **auto** - размер полей и отступов автоматически рассчитывается браузером.



```

<html>
<head>
<style type="text/css">
div.block1 {
  width: 260px;
  background-color: green;
  border: 2px solid red;
}
div.block2 {
  background-color: pink;
  border: 8px solid white;
  margin: 40px;
  padding: 20px
}
</style>
</head>
<body>
<div class="block1">
<div class="block2">
Осмысливая мысли в смысле смысла,
есть смысл помыслить о немыслимом!
</div>
</div>
</body>
</html>

```



Дополнительные свойства *margin* и *padding*

Свойства CSS **margin** и **padding** имеют ряд дополнительных свойств.

У **margin** это:

- **margin-top** - поле от верхней границы элемента родителя;
- **margin-left** - поле от левой границы элемента родителя;
- **margin-right** - поле от правой границы элемента родителя;
- **margin-bottom** - поле от нижней границы элемента родителя.

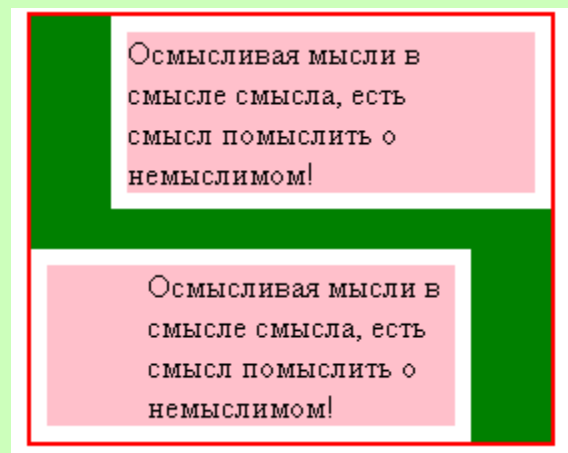
У **padding** соответственно:

- **padding-top** - отступ от верхней границы элемента до его содержания;
- **padding-left** - отступ от левой границы элемента до его содержания;
- **padding-right** - отступ от правой границы элемента до его содержания;
- **padding-bottom** - отступ от нижней границы элемента до его содержания.

```

<html>
<head>
<style type="text/css">
  div.block1 {
    width: 260px;
    background-color: green;
    border: 2px solid red;
  }
  div.block2 {
    background-color: pink;
    border: 8px solid white;
    margin-left: 40px;
    padding-right: 50px
  }
  div.block3 {
    background-color: pink;
    border: 8px solid white;
    margin-right: 40px;
    padding-left: 50px
  }
</style>
</head>
<body>
<div class="block1">
<div class="block2">
Осмысливая мысли в смысле смысла,
есть смысл помыслить о немислимом!
</div>
<br />
<div class="block3">
Осмысливая мысли в смысле смысла,
есть смысл помыслить о немислимом!
</div>
</div>
</body>
</html>

```



Поплавки

Эх, рыбалка... Закинуть бы сейчас удочку и последить за поплавком на свежем воздухе... Но нет, в этой главе речь пойдет о совсем других поплавках.

Обтекание элемента

Итак, свойство **float** при переводе с английского означает "поплавок" и определяет, по какой стороне выравнивается элемент и как обтекает другие элементы.

- **left** - выравнивает элемент по левому краю, остальные элементы обтекают его справа;
- **right** - выравнивает элемент по правому краю, остальные элементы обтекают его слева;
- **none** - выравнивание элемента не задается (по умолчанию)

```

<html>
<head>
  <style type="text/css">
    div {
      float: left;
    }
  </style>
</head>

```

```

<body>

```

```

<p>

```

```

<p>

```

```

<div></div>

```

Текст в этом блоке обтекает рисунок помещённый в контейнер `div` с правой стороны, так как сам блок с рисунком, с помощью свойства `float` с значением `left`, "всплывает" слева.

```

</p>

```

```

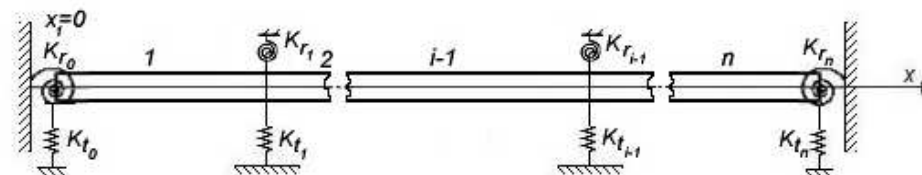
</body>

```

```

</html>

```



Текст в этом блоке обтекает рисунок помещённый в контейнер `div` с правой стороны, так

как сам блок с рисунком, с помощью свойства `float` с значением `left`, "всплывает" слева.

НЕДЕЛЯ 7: КАСКАДНЫЕ ТАБЛИЦЫ СТИЛЕЙ CSS (III)

Предисловие

Название книги: Каскадные таблицы стилей CSS (III)

Основные цели работы с книгой:

- добавлять и форматировать фон;
- дать определение псевдоклассам и псевдоэлементам;
- улучшить позиционирование объектов при помощи соответствующих стилей;
- применить различные стили курсора;
- прочитать о наиболее распространенных возможностях CSS3;
- применить на практике полученные теоретические знания по оформлению веб-страниц;
- дополнить предыдущие практические наработки по созданию своего веб-сайта.

Рассматриваемые темы:

- Фон;
- Псевдоклассы;
- Псевдоэлементы;
- Позиционирование;
- Курсоры;
- Возможности CSS3.

Результатом проработки данного материала является умение учащегося:

- выбирать и форматировать фон;
- создавать интерактивность при помощи псевдоклассов и курсоров;
- описать возможности CSS3.

Фон

В данной главе углубим наши знания в создании фона для веб-страницы.

Фоновое изображение

Для любого элемента (всей страницы, ячейки таблицы и т.д.) можно задать фоновое изображение с помощью CSS свойства: **background-image** и пути к картинке согласно синтаксису **url (путь к файлу/имя файла)**. Путь к файлу указывается в том случае, если рисунок находится в другой папке.

```

<html>
<head>
  <style
type="text/css">
  body {
    background-
image: url(oli.jpg);
  }

  p {
    color: white;
    font-size: 40px;
  }
</style>
</head>
<body>
<p>Демонстрация
фона.</p>
</body>
</html>

```



Если рисунок не заполняет собой все пространство элемента, то фон выкладывается "плиткой".

Фиксация фонового изображения

Если на странице или в блоке есть полоса прокрутки, то фоновое изображение будет прокручиваться вместе с остальным содержанием данного блока. Зафиксировать фоновое изображение позволяет свойство **background-attachment** и его значения:

- **fixed** - фиксированный фон;
- **scroll** - подвижный фон (по умолчанию).

Повторение фонового изображения

Если рисунок не заполняет собой все пространство элемента, то он повторяется и выкладывается "плиткой". Свойство **background-repeat** регулирует повторение фонового изображения:

- **no-repeat** - запретить повторение, показать изображение только один раз;
- **repeat** - повторять изображение (по умолчанию);
- **repeat-x** - повторять только по горизонтали;
- **repeat-y** - повторять только по вертикали.

Дополнительные возможности

С помощью нового свойства **background-size** можно указывать размер изображения выступающего в качестве фона. Указывать размер ширины и высоты фоновой картинке можно с помощью следующих значений:

- **px** - в пикселях или любых других единицах измерения принятых в CSS;
- **%** - в процентах от размеров элемента, которому назначается фон;

- **cover** – растягивает фоновое изображение на весь блок;
- **contain** – растягивает фоновое изображение с сохранением пропорций до тех пор, пока его ширина или высота не упрётся в края блока;
- **auto** - исходный размер изображения (по умолчанию); в случае если значение **auto** присвоено только одной из сторон, то её размер вычисляется относительно размера второй стороны с указанным размером.

Позиция фонового изображения

При помощи CSS свойства **background-position** можно управлять позицией фонового изображения в элементах, где оно задано с помощью **background-image**. Позицию верхнего левого угла изображения можно задать в процентах, пикселях, а также любых других единицах измерения CSS, с помощью двух значений, где первое значение - отступ от левой границы элемента, а второе значение - отступ от верхней границы элемента. Например, **background-position: 200px 100px;** означает, что фоновое изображение будет смещено по горизонтали на 200 пикселей от левой границы элемента и по вертикали на 100 пикселей от верхней границы элемента. Кроме числовых значений можно использовать следующие значения:

по горизонтали:

- **left** - расположить слева;
- **center** - расположить по центру;
- **right** - расположить справа;

по вертикали:

- **top** - расположить сверху;
- **center** - расположить по центру;
- **bottom** - расположить снизу.

В заключении

Перечисленные выше свойства можно объединить в одно при помощи ключевого свойства **background** и записать:

background: url(oli.jpg) fixed repeat-x center

Полезные советы

Не используйте "мегабайтные" изображения! Берегите время, деньги и нервы пользователей. Почти всегда можно значительно сжать картинку особо не проиграв в качестве изображения, поработав с ней в графических редакторах.

Псевдоклассы

Псевдоклассы - это особые свойства, которые позволяют добавить странице некую динамику и менять стиль элемента в зависимости от действий пользователя. Классическим примером применения псевдоклассов является ссылка, которая меняет свой цвет при наведении на неё курсором.

Список псевдоклассов:

- **active** - стиль для ссылки, которая становится активной, но переход по ней еще не совершен;
- **visited** - стиль для недавно посещённой ссылки;
- **link** - стиль для нечасто посещаемой ссылки;
- **hover** - стиль элемента, на который наведён курсор;
- **first-child** - стиль первого дочернего элемента.

Каждый псевдокласс будет отдельно описан ниже, а пока посмотрим на синтаксис. Для того чтобы применить тот или иной псевдокласс к элементу и определить его стиль, нужно следовать следующим правилам синтаксиса:

a: hover { color: #ff0000 }

где: **a** - селектор или тег, к которому применяется псевдокласс.

:hover - после двоеточия сам псевдокласс

{color: #ff0000} - блок объявления стилей в фигурных скобках.

Вся эта запись говорит о том, что если навести курсор на ссылку - то она станет красной.

Ссылки

Рассмотрим сразу три псевдокласса созданных для работы со ссылками.

- **active** - стиль активной ссылки (нажимая и удерживая мышкой на одну из не посещенных ссылок). НЕ работает в браузере *Internet Explorer* 6 и ниже.
- **visited** - стиль для посещённой ссылки;
- **link** - стиль для не посещенной ссылки.

```
<html>
<head>
  <meta http-equiv="Content-Type"
content="text/html; charset=windows-1251">
  <style type="text/css">
    a:link {color:red;}
    a:active {color:blue;}
    a:visited {color:green;}
  </style>
</head>
<body>
<a href="#1">Первая ссылка.</a>
<a href="#1">Вторая ссылка.</a>
</body>
</html>
```

Посмотреть [Доступно в учебных материалах курса]

Псевдокласс hover

Псевдокласс **hover** активизируется в том случае, если на элемент наведен курсор.

```
<html>
<head>
```

Посмотреть [Доступно в учебных материалах курса]

```

<meta http-equiv="Content-Type"
content="text/html; charset=windows-1251">
<style type="text/css">
    table {
        border: 1px solid black;
        border-collapse: collapse;

        td {
            border: 1px solid black;
        }
        tr:hover {
            background-color:red;
        }
</style>
</head>
<body>
<table>
<tr>
<td>ИВАНОВ</td><td>+</td><td></td>
</tr>
<tr>
<td>ПЕТРОВ</td><td></td><td>+</td>
</tr>
</table>
</body>
</html>

```

Следует отметить, что браузер *Internet Explorer 6* и его более ранние версии поддерживает псевдокласс **hover** только для ссылок - тег `<a>`, так что, к примеру, строки таблицы, при наведении на них курсора, в браузерах *IE 6* и ниже подсвечиваться не будут!

Псевдокласс first-child

Все элементы CSS можно определить как родительские или дочерние; элемент родитель может содержать в себе несколько дочерних элементов, ну например:

```

<div>- блок родитель
  <p></p> - первый дочерний элемент
  <p></p> - второй дочерний элемент
</div>- блок родитель

```

Так вот псевдокласс **first-child** определяют стиль первого дочернего элемента находящегося в родительском контейнере.

```

<html>
<head>
  <meta http-equiv="Content-Type"
content="text/html; charset=windows-1251">
  <style type="text/css">
    table {
      border: 1px solid black;
      border-collapse: collapse;
    }
    td {
      border: 1px solid black;
      background: green;
      color: white;
    }
    td:first-child{
      background: pink;
      color: white;
    }
  </style>
</head>
<body>
<table>
<tr>
<td>Иванов</td><td>+</td><td></td>
</tr>
<tr>
<td>Петров</td><td></td><td>+</td>
</tr>
<tr>
<td>Сидоров</td><td>+</td><td>+</td>
</tr>
</table>
</body>
</html>

```

Иванов	+	
Петров		+
Сидоров	+	+

Применяют данный псевдокласс так же и в тех случаях, если требуется задать разный стиль для первого и последующих элементов, например сделать "буквицу"- одну единственную в начале документа или обозначить первый абзац текста на всех страницах сайта , выделить первый пункт в списках...и т.д.

Полезные советы

- Псевдоклассы нельзя внедрять в *HTML* документ с помощью атрибута `style`. Можно только с помощью тега `<style>` в голове документа или внешнего CSS файла.
- Если селектор какого-либо псевдокласса явно не указывать, можно написать вот так:
:hover {color: #ff0}
 Это значит, что действие данного псевдокласса будет распространяться на все элементы документа.
- Браузер *IE6* и ниже игнорирует практически все псевдоклассы.

Псевдоэлементы

Псевдоэлементы - это особый вид свойств CSS, которые позволяют работать не над самим элементом, а над его отдельной частью.

Вот перечень всех псевдоэлементов:

- **:first-letter** - стиль первой буквы текстового блока;
- **:first-line** - стиль первой строки текстового блока;
- **:before** - добавляет содержимое до элемента;
- **:after** - добавляет содержимое после элемента;
- **:selection** - стиль выделенного пользователем текста.

Так же как и в случае с псевдоклассами, псевдоэлементы используются согласно следующего синтаксиса:

p:first-letter { color:#ff0000 }

где: **p** - селектор, к которому применяется псевдоэлемент;
:first-letter - после двоеточия псевдоэлемент;
{color:#ff0000} - блок объявления стилей в фигурных скобках.

Рассмотрим перечисленные псевдоэлементы по отдельности.

Стиль первой буквы

Псевдоэлемент **first-letter** задаёт стиль первой буквы в текстовом блоке.

```
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html;
  charset=windows-1251">
  <style type="text/css">
    p {
      font: 16px Arial;
    }
    p:first-letter {
      font: bold 24px Verdana;
      color:#ff0000
    }
  </style>
</head>
<body>
<p>Мало кто знает, как много надо знать для того, что бы знать, как
мало мы знаем.</p>
</body>
</html>
```

Мало кто знает, как много надо знать для того, что бы знать, как мало мы знаем.

Стиль первой строки

Псевдоэлемент **first-line** определяет стиль первой строки в текстовом блоке.

```
<html>
<head>
  <meta http-equiv="Content-
Type" content="text/html;
charset=windows-1251">
  <style type="text/css">
    p {
      font: 16px Arial;
    }
    p:first-line {
      font: bold 24px
Verdana;
      color:#ff0000
    }
  </style>
</head>
<body>
<p>Мало кто знает, как много
надо знать для того, что бы
знать, как мало мы знаем.<p>
</body>
</html>
```

Мало кто знает, как

много надо знать для того, что бы знать, как
мало мы знаем.

Контент

Псевдоэлементы **after** и **before** предназначены для вставки в страницу сайта содержания, которое изначально не указано в *HTML* документе. Содержание добавляется перед (**:before**) или после (**:after**) какого-либо элемента с помощью свойства **content**, которое собственно и определяет содержимое для вставки.

p:after {content: "Конец, а кто слушал молодец!"; }

Теперь после каждого параграфа будет добавляться надпись: "Конец, а кто слушал молодец!"

Значением свойства **content** может быть текст, полный адрес ссылки на рисунок, кавычки и символы юникода.

```

<html>
<head>
  <meta http-equiv="Content-
Type" content="text/html;
charset=windows-1251">
  <STYLE type="text/css">
    p {
      font: 16px Arial;
    }
    p:after {
      color: blue;
      content: " Конец, а
кто слушал молодец!";
    }
  </style>
</head>
<body>
<p>Мало кто знает, как много
надо знать для того, что бы
знать, как мало мы знаем.</p>
<p>Мало кто знает, как много
надо знать для того, что бы
знать, как мало мы знаем.</p>
</body>
</html>

```

Мало кто знает, как много надо знать для того, что бы знать, как мало мы знаем. **Конец, а кто слушал молодец!**

Мало кто знает, как много надо знать для того, что бы знать, как мало мы знаем. **Конец, а кто слушал молодец!**

Псевдоэлементы **after** и **before**, как и свойство **content** не поддерживаются браузерами *Internet Explorer 7* и ниже.

Позиционирование

Позиционированием - это месторасположения того или иного элемента на странице.

Абсолютное позиционирование

Абсолютно позиционированный элемент (**position: absolute**) выводится из общего потока и несмотря на другие элементы и их взаимное расположение, занимает указанное место на странице от края/краёв окна браузера. При таком способе позиционирования один элемент может накладываться поверх другого следующими свойствами CSS:

- **bottom** - расстояние от нижнего края окна браузера;
- **left** - расстояние от левого края окна браузера;
- **right** - расстояние от правого края окна браузера;
- **top** - расстояние от верхнего края окна браузера.

Данные расстояния могут быть заданы в пикселях, процентах или любых других принятых единицах измерения CSS.

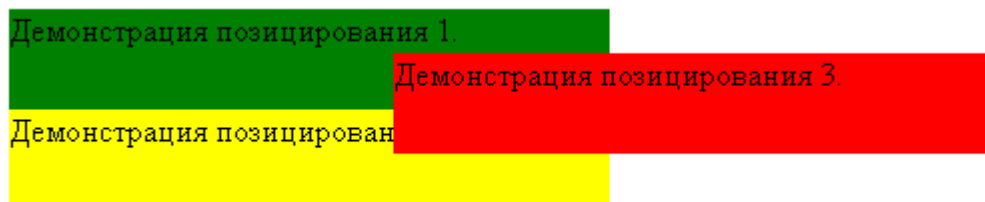
```

<html>
<head>
  <style type="text/css">
    div.blokk1 {
      background: green;
      width: 300px;
      height: 50px;
    }

    div.blokk2 {
      background: yellow;
      width: 300px;
      height: 50px;
    }

    div.blokk3 {
      background: red;
      width: 300px;
      height: 50px;
      position: absolute; left:200px; top: 30px;
    }
  </style>
</head>
<body>
<div class="blokk1">Демонстрация позицирования 1.</div>
<div class="blokk2">Демонстрация позицирования 2.</div>
<div class="blokk3">Демонстрация позицирования 3.</div>
</body>
</html>

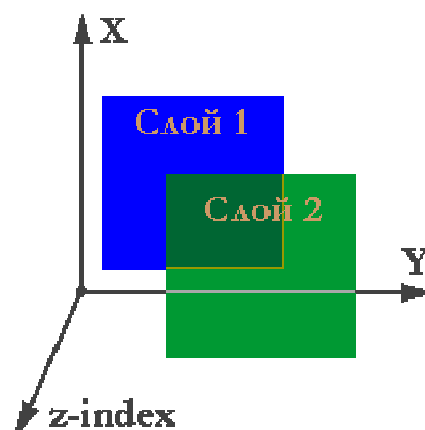
```



z-index

Позиционированные элементы могут накладываться один поверх другого, имитируя тем самым некую трёхмерность страницы, где каждый последующий наложенный друг на друга элемент является слоем. Свойство **z-index** позволяет веб-мастеру управлять позициями этих слоёв в глубину экрана (по оси Z), другими словами позволяет указывать браузеру какие элементы следует показывать на переднем плане, а какие на заднем. Значения свойства **z-index**:

- **auto** - элементы накладываются друг на друга том порядке, каком они указаны в коде *HTML* (по умолчанию);



В

- **целое число** - чем выше данное значение, тем более высокую позицию занимает элемент по отношению к тем элементам, значение которых ниже.

При равном значении **z-index**, на переднем плане находится тот элемент, который в коде *HTML* идет ниже остальных.

```
<html>
<head>
  <style type="text/css">
    body {background-color: green;}
    a {
      position: absolute;
      z-index: auto;
      top: 100px;
      border: black 1px solid;
    }
    a:hover {
      position: absolute;
      z-index:1;
      top: 80px;
      border: black 1px solid;
    }
  </style>
</head>
<body>
<h2>Проведите курсором по картам</h2>
<a href="#" style="left: 10px"></a>
<a href="#" style="left: 30px"></a>
<a href="#" style="left: 50px"></a>
<p style="position: absolute; left: 10px; top: 250px;">
В злосчастном IE6, в отличие от других браузеров,
псевдокласс <b>:hover</b> работает только для ссылок,
поэтому в качестве контейнера для рисунков/карт использую
никуда не ведущую ссылку. (<b>a href="#"</b>)</p>
</body>
</html>
```

Посмотреть [Доступно в учебных материалах курса]

Курсоры

Свойство CSS **cursor** позволяет установить нестандартный вид курсора для того или иного элемента - блока, текста, рисунка. Когда пользователь наведет курсор на такой элемент, курсор поменяет свой вид. Курсоры можно выбирать как стандартные, так и подгружать свои собственные пользовательские курсоры - файлы в формате *cur*, *ani* или *svg*.

Стандартные курсоры

Для того чтобы изменить внешний вид курсора свойству **cursor** необходимо указать одно из возможных значений:

- **auto**- курсор назначается автоматически в зависимости от специфики элемента (по умолчанию)
- **crosshair**- перекрестие
- **default**- встроенный(основной)
- **e-resize**- стрелка на "восток"
- **hand**- указатель аналогия pointer
- **help**- помощь
- **move**- перемещение
- **n-resize**- переместить на "север"
- **ne-resize**- переместить на "северо-восток"
- **nw-resize**- переместить на "северо-запад"
- **pointer**- указатель
- **progress**- продолжение операции
- **s-resize**- переместить на "юг"
- **se-resize**- переместить на "юго-восток"
- **sw-resize**- переместить на "юго-запад"
- **text**- текст
- **w-resize**- переместить на "запад"
- **wait**- ожидание

Кроме того некоторые браузеры поддерживают дополнительные формы курсоров:

- **all-scroll**- переместить во все стороны
- **col-resize**- переместить по горизонтали
- **no-drop**- указатель "нет доступа"
- **not-allowed**- нет доступа
- **row-resize**- переместить по вертикали
- **vertical-text**- вертикальный текст

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html;
charset=windows-1251">
  <style type="text/css">
    table {
                                border: 0px solid yellow;
                                border-collapse: collapse;
                                }
    td {
                                background: brown;
                                color: white;
                                font-weight: bold;
                                }
    td:first-child {
                                color: red;
                                }
```

```

        font-weight: bold;
    }
    .k1 {cursor: auto;}
    .k2 {cursor: crosshair;}
    .k3 {cursor: default;}
    .k4 {cursor: e-resize;}
    .k5 {cursor: hand;}
    .k6 {cursor: help;}
    .k7 {cursor: move;}
    .k8 {cursor: n-resize;}
    .k9 {cursor: ne-resize;}
    .k10 {cursor: nw-resize;}
    .k11 {cursor: pointer;}
    .k12 {cursor: progress;}
    .k13 {cursor: s-resize;}
    .k14 {cursor: se-resize;}
    .k15 {cursor: sw-resize;}
    .k16 {cursor: text;}
    .k17 {cursor: w-resize;}
    .k18 {cursor: wait;}
    .k19 {cursor: all-scroll;}
    .k20 {cursor: col-resize;}
    .k21 {cursor: no-drop;}
    .k22 {cursor: not-allowed;}
    .k23 {cursor: row-resize;}
    .k24 {cursor: vertical-text;}
</style>
</head>
<body>
<table>
<tr class="k1"><td>auto </td><td>- курсор назначается автоматически
</td></tr>
<tr><td></td><td>в зависимости от специфики элемента.(по
умолчанию)</td></tr>
<tr class="k2"><td>crosshair</td><td>- перекрестие</td></tr>
<tr class="k3"><td>default</td><td>- встроенный(основной)</td></tr>
<tr class="k4"><td>    e-resize</td><td>- стрелка на
"восток"</td></tr>
<tr class="k5"><td>    hand</td><td>- указатель аналогия
pointer</td></tr>
<tr class="k6"><td>    help</td><td>- помощь</td></tr>
<tr class="k7"><td>    move</td><td>- перемещение</td></tr>
<tr class="k8"><td>    n-resize</td><td>- переместить на
"север"</td></tr>
<tr class="k9"><td>    ne-resize</td><td>- переместить на "северо-
восток"</td></tr>
<tr class="k10"><td>    nw-resize</td><td>- переместить на "северо-
запад"</td></tr>
<tr class="k11"><td>    pointer</td><td>- указатель</td></tr>
<tr class="k12"><td>    progress</td><td>- продолжение
операции</td></tr>
<tr class="k13"><td>    s-resize</td><td>- переместить на
"юг"</td></tr>
<tr class="k14"><td>    se-resize</td><td>- переместить на "юго-
восток"</td></tr>
<tr class="k15"><td>    sw-resize</td><td>- переместить на "юго-

```

```

запад"></td></tr>
<tr class="k16"><td>      text</td><td>- текст</td></tr>
<tr class="k17"><td>      w-resize</td><td>- переместить на
"запад"></td></tr>
<tr class="k18"><td>      wait</td><td>- ожидание</td></tr>
<tr class="k19"><td>      all-scroll</td><td>- переместить во все
стороны</td></tr>
<tr class="k20"><td>      col-resize</td><td>- переместить по
горизонтали</td></tr>
<tr class="k21"><td>      no-drop</td><td>- указатель "нет
доступа"></td></tr>
<tr class="k22"><td>      not-allowed</td><td>- нет доступа</td></tr>
<tr class="k23"><td>      row-resize</td><td>- переместить по
вертикали</td></tr>
<tr class="k24"><td>      vertical-text</td><td>- вертикальный
текст</td></tr>
</table>
</body>
</html>

```

Посмотреть [Доступно в учебных материалах курса]

Возможности CSS3

В данном разделе рассмотрим еще несколько дополнительных и интересных свойств CSS, которые работают во всех современных браузерах: *IE9+*, *Opera 10+*, *Firefox 3.6+*, *Chrome 12+*, *Safari 5+*.

Тень текста

CSS3 позволяет имитировать тень, падающую от текста, при помощи свойства **text-shadow**. Для этого надо задать как минимум два из четырех значений:

- **Первое значение** - определяет сдвиг тени относительно текста по горизонтали, указывается в пикселях или любых других единицах измерения CSS, может иметь как положительное, так и отрицательное значение, является обязательным параметром;
- **Второе значение** - определяет смещение тени относительно текста по вертикали, может быть как положительным, так и отрицательным; обязательный параметр.
- **Третье значение** - делает тень тени, определяет радиус сглаживания тени, делая её более чёткой или же наоборот размытой: чем большее значение, тем больше расфокусируется отбрасываемая от текста тень; данный параметр не является обязательным, по умолчанию равен 0.
- **Четвёртое значение** - задаёт цвет тени, необязательный параметр; в случае если он не указан, тень будет того же цвета что и сам текст.

```

<html>
<head>
  <style type="text/css">
    .minu_stiil{
      text-shadow: 15px 10px 3px
blue;
    }
  </style>
</head>
<body>
<p class="minu_stiil">Демонстрация тени.</p>
</body>
</html>

```

Демонстрация тени.
Демонстрация тени.

Текст может также отбрасывать не одну, а сразу несколько теней одновременно с разными параметрами! Для этого следует указать группы параметров для каждой тени отдельно через запятую.

```

<html>
<head>
  <style type="text/css">
    .minu_stiil{
      font-size: 20px;
      text-shadow: 15px 20px
3px blue, -15px -15px 3px green, 20px 1px
3px yellow;
    }
  </style>
</head>
<body>
<p class="minu_stiil">Демонстрация
тени.</p>
</body>
</html>

```

Демонстрация тени
Демонстрация тени.
Демонстрация тени.

Прозрачность

Свойство **opacity** позволяет делать прозрачным элемент сайта. Степень прозрачности элемента определяется значением от **0** (полностью прозрачный) до **1** (непрозрачный).

```
<html>
<head>
  <style type="text/css">
    .stiil1{
      opacity:0.1;
    }
    .stiil2{
      opacity:0.5;
    }
    .stiil3{
      opacity:1;
    }
  </style>
</head>
<body>
<table><tr><td>
</td>
<td></td>
<td></td>
</tr>
</table>
</body>
</html>
```

Посмотреть [Доступно в учебных материалах курса]