Normalize.css reset.css всегда эти файлы добавлять в проект и ставить в самом верху путь к ним!!! Они сбрасывают левые стили со страницы…

1-отступы задаются направо и вниз (margin-right u margin-bottom)

2-элемент сам себя не толкает, а толкает соседний левый или верхний элемент (сосед)

3-Если нет соседнего элемента сверху или слева, мы используем padding top или padding left ‘у родителя’

4-Последнему элементу группы нужно обнулить отступ margin с помощью псевдоклассов (selector: last-child {margin: 0px;})

<!doctype html> хтмл5

<html>

<head> Начинается название веб-страницы

    <meta charset="utf-8"> говорит браузеру о применяемой кодировке utf-8 на странице

    <title>Book Toni</title> заглавие страницы

//  <link type="text/css" rel="stylesheet" href="../lounge.css" />

@media screen and (min-device-width: 481px) { #guarantee { margin-right: 250px; }} Таким образом, эти правила будут применяться, если ширина экрана окажется больше 480 пикселов…

link- ссылка на внешнюю информацию / Атрибут rel устанавливает взаимосвязь между HTMLфайлом и тем, на что вы ссылаетесь. Вы ссылаетесь на таблицу стилей, поэтому используется значение stylesheet. /

    <style type="text/css"> Поскольку вы будете применять CSS, вам необходимо задать тип «text/css»

        body {

            background-color: #d2b48c; Здесь задаются стили страницы Задает желто-коричневый цвет фона страницы.

            margin-left: 20%; Устанавливает размер левого и правого отступов: по 20 % от ширины страницы для каждого.

            margin-right: 20%;

border: 2px dotted black; Определяет, что граница вокруг содержимого будет пунктирной, и выделяет ее черным цветом.

            font-family: Arial, Helvetica, sans-serif; Определяет, какой шрифт использовать для текста.

         padding: 10px 10px 10px 10px; Создает поля вокруг основного текста на страниц

}

display: block; превращение строчного элемента в блочный

в строчный элемент нельзя: вкладывать блочный элемент, сделать вертикальные отступы, задать ширину

display: inline-block; одновременно и строчный и блочный элемент

border-style: solid; сначало выбираем вид рамки, это важно!!!! Допустим сплошная

background-image: url(../img/01-Logo.svg); вставка изображений через стили css

.section-description\_icon\_box img:last-child { -применяем псевдоклас last-child ко всем элементам img

    margin-right: 0px;} -селектор псевдокласса

img {} – селектор тега

.section {} -селектор класса

#logo {} -селектор идентификатора id

\*{} селектор всех элеметов

Padding -это внутренний отступ

Border -это рамка

Margin -это внешний отступ

a:hover псевдокласс : hover -наведение курсора

li:last-child – последний элемент

    </style>

</head>

<body> Начинается основная часть страницы

    <a href="#конец">в конец</a> создания ссылки, Атрибут href задает назначение ссылки.

<a target="\_blank" href="http://wickedlysmart.com/buzz" открытие ссылки в новом окне отдельном

    <h1 id="начало">На скутере по США</h1> id ставит якорь по названием "начало"

    <p> абзац

        Дневник моих

    </p>

    <h2>

        2 июня, 2012

    </h2>

    <img src="http://zoloto-russkoe.ru/journal/images/segway2.jpg" alt="изображение скутера"> alt-когда исчезает изображение, остается описание его

    <p>

        Первый <em> день моего </em> путешествия Используется курсивный шрифт

        <ul> если <ul> промаркирует элементы

            <li>сотовый телефон,</li>

            <li> iPod,</li>

        </ul>

        Только все самое необходимое: <q>Путешествие в тысячу </q> выделение цитат вместо кавычек

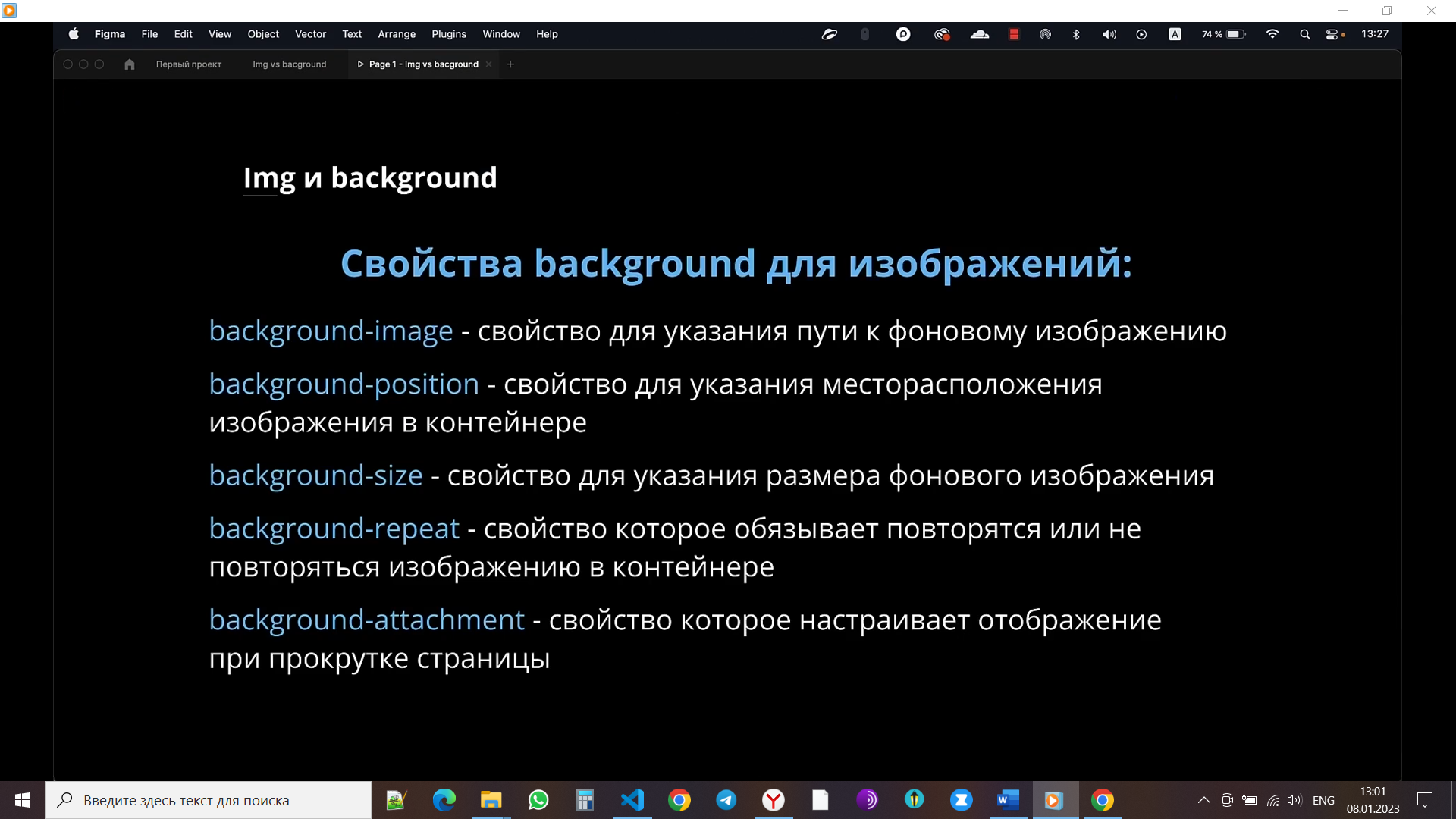
    </p>

    <h2>

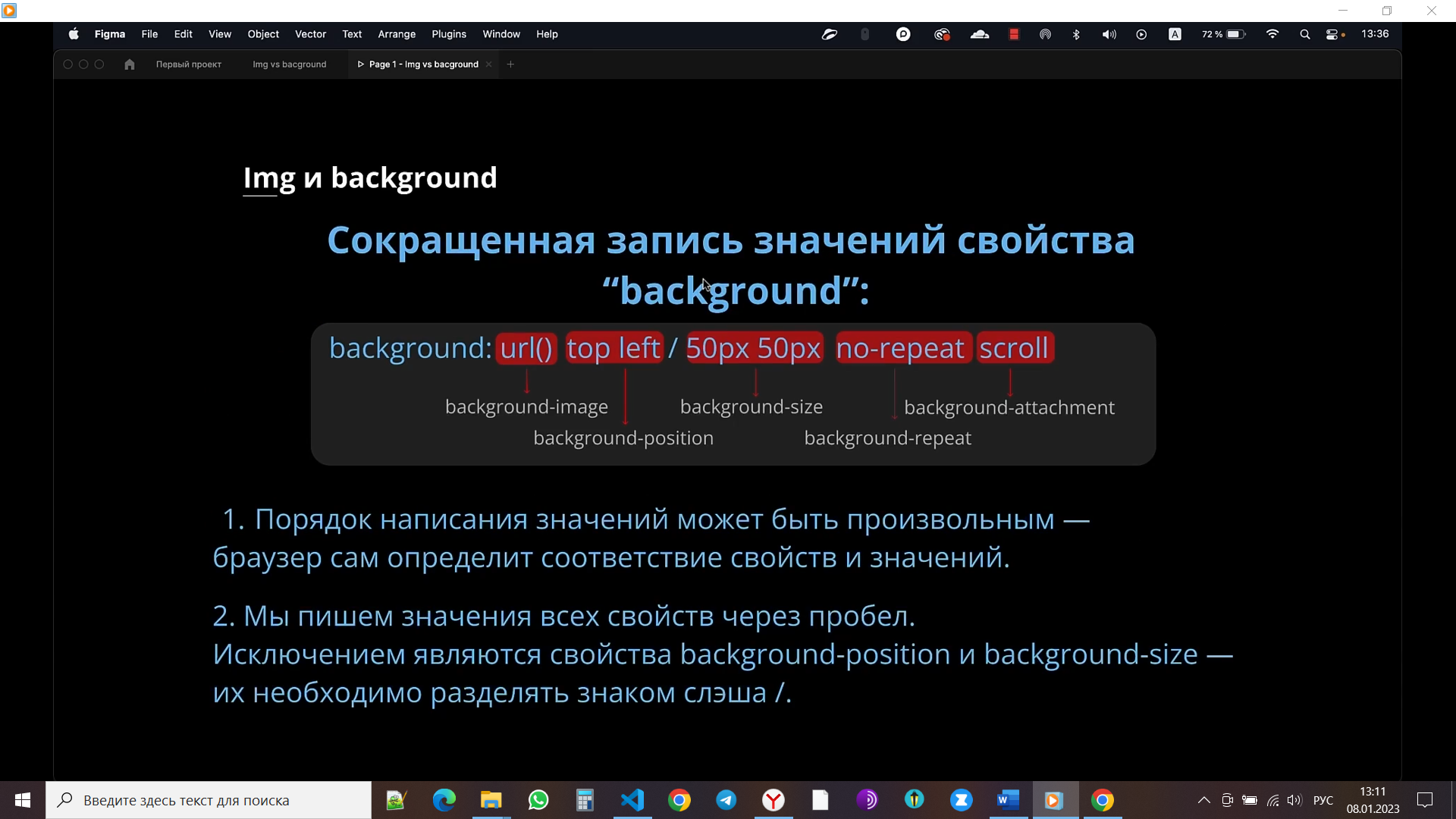
        14 июля, 2012

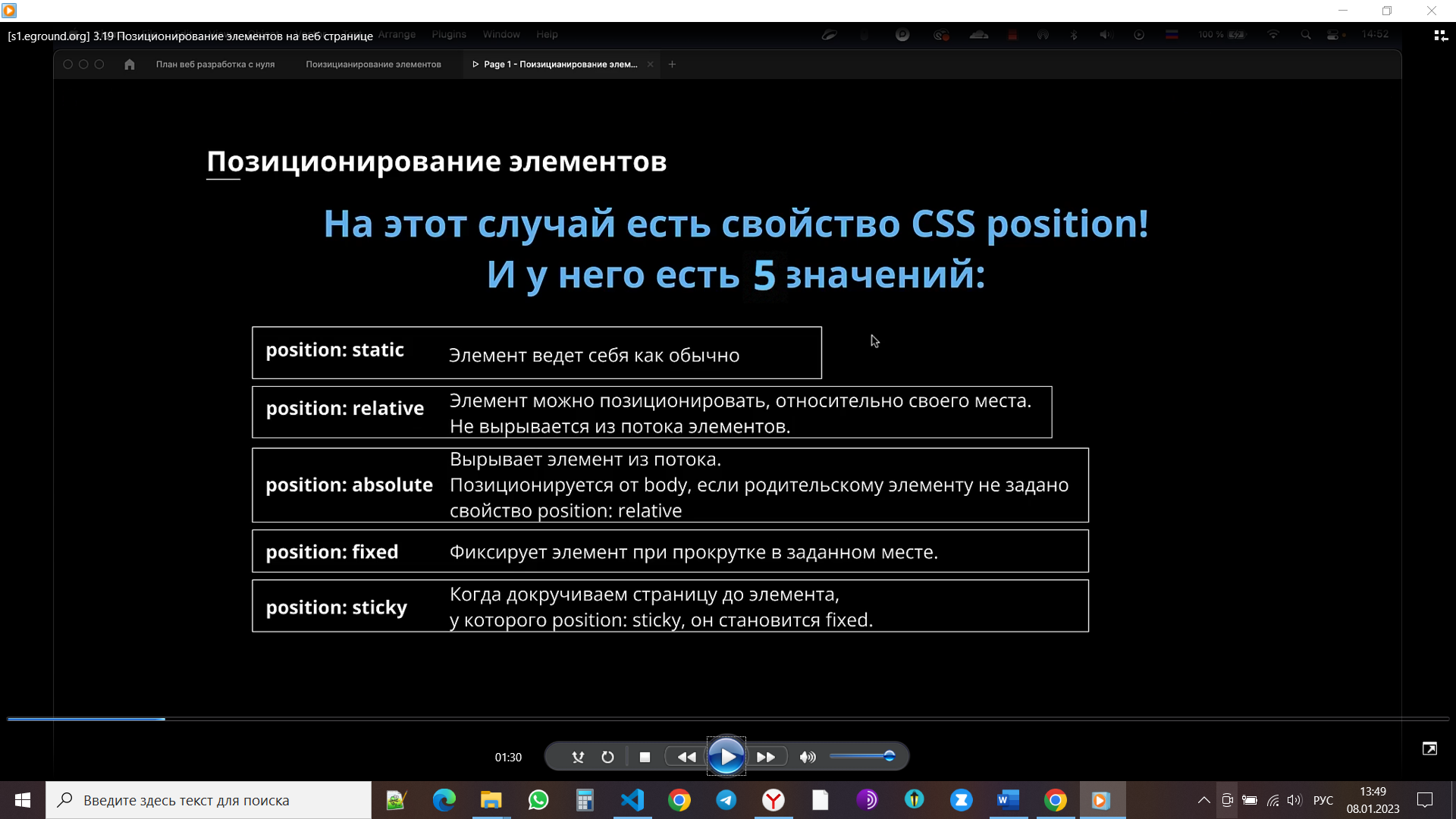
    </h2>

    <p class="rts"> типа якоря в css пишем: p.rts {color: yellow } и этот абзац будет желтого цвета



background-size: cover; - подгоняет изображение под ширину сайта….!





\* {box-sizing: border-box;} \*- для всех тегов

z-index: -5; зет индекс работает только если есть пункт position: если этого не будет то зет индекс не будет работать

background: linear-gradient(180deg, rgba(244, 213, 178, 0.8) 0%, rgba(234, 156, 65, 0.8)100%); делаем прозрачный блок с прозрачностью 0.8 (80%) и с переходом 0% и 100%

margin-bottom: делает отступ блока, вставляем в верхний блок или в левый, т.к. блоки сами себя не двигают!!

font-size размер текста

margin: 0 auto; -выравниваем БЛОЧНЫЙ элемент по центру!!

 Я видел парочку :

    </p>

    <blockquote> выделение абзаца с отступом впаро

        Если вы не заметите <br> проезжающие мимо машины, <br> то они могут сбить вас. <br> Одно мгновение — <br> и бесконечность… <br> разрыв строки

    </blockquote>

    <p>

        Я определенно <em> не хотел, </em> чтобы на меня  выделение текста

        <dl>

            <dt>Знаки Burma Shave</dt>

            <dd>Самое известное шоссе в сети автомагистралей Соединенных Штатов.</dd>

        </dl>

    </p>

    <h2 id="20 августа">

        20 августа, 2012

    </h2>

    <img src="segway1.jpg" width="48" hight="100"> Атрибут src определяет имя файла с изобр  width и hight ширина и высота рисунка

    <p>

        Итак, я уже проехал 1200:

        <ol> если <ol> пронумерует элементы

            <li>Вала-Вала, штат Вашингтон,</li> нумерация

            <li id="конец">Трут-орКонсекуэнсес, штат Нью-Мексико</li>

        </ol>

        <a href="#начало">в начало</a> ссылка на якорь для перехода в начало

    </p>

Ознакомьтесь с <a href="mission.html"

  title="Ознакомьтесь с важной задачей“>нашей задачей</a> title-подсвечивает описание ссылки когда мышку подводишь

<dt>Знаки Burma Shave</dt>  название абзаца со сдвигом влево

    <dd>Дорожные знаки, распространенные в США      описание со сдвигом вправо

        в 1920–1930-е годы и рекламирующие товары

        для бритья.</dd>

«Элемент <html>очень важен» что бы это напечаталось надо так- Элемент &lt ; html &gt; очень важен - & (амперсанд).

/\* и \*/ комментарий внутрь можно помещать...

<p id="footer">Пожалуйста</p>

id. Он применяется исключительно для того, чтобы присвоить имя одному элементу

</body>

</html>

**PNG бывает трех видов: PNG-8 PNG-24 и PNG-32 в зависимости от количества цветов** - без потерь качества для многоцвет логотипов и с прозрачностью

**GIF может выводить рисунки с количеством цветов, не превышающим 256.** -без потерь качества, прозрач для одного цвета

**JPEG- «с потерями качества не работает с прозрачностью, для фоток хорош**

color- для задания цвета текста элемента

font-weight- Задает толщину шрифта

font-style: italic курсив шрифт

font-weight: bold жирный шрифт

clear: right; Здесь мы добавляем новое свойство в правило для нижнего колонтитула. Оно говорит, что с правой стороны элемента не может быть никакого плавающего содержимого.

left- вы говорите элементу, где расположить его левый край

line-height- задает межстрочный интервал в текстовых элементах

top- Определяет позицию верхнего края элемента

background-color- определяет фоновый цвет элемента

background-image имеет строго определенное назначение — устанавливать фоновые рисунки для элементов

background-repeat: no-repeat; фон рисунок без повторений

background-position: top left; вверху слева

border- можно нарисовать рамку вокруг элемента. Она может быть сплошной, пунктирной и т. д

padding- Если вам необходим отступ между краем элемента и его содержимым, применяйте свойство padding

margin - Если вам необходим поля между краем элемента и его содержимым, применяйте свойство margin

font-size- Делает шрифт больше или меньше

text-align- выравнивания текста по левому краю, по центру или по правому краю

letter-spacing- Позволяет задавать интервал между символами в пределах элемента

font-style- Это свойство применяется для выделения текста курсивом

list-style- позволяет контролировать то, как будут выглядеть пункты списка

background-image- чтобы поместить изображение под элементом

p { color: black;} Это правило для любого элемента ‹p›

.greentea { color: green; }Это правило для любого члена класса greentea. Оно более приоритетно.

p.greentea { color: green; }Это правило только для абзацев из класса greentea, поэтому оно еще более приоритетно.

font-size: 14px В данном случае мы задаем размер шрифта 14 пикселов в высоту

font-size: 150%; данный шрифт в высоту будет составлять 150 % от другого шрифта

(

a:link {

    color: #b76666; задает форму и цвет подчеркивания линии ссылки

    text-decoration: none;

    border-bottom: thin dotted #b76666;

}

a:visited { color: #675c47; text-decoration: none; border-bottom: thin dotted #675c47; }

)

display: inline; Здесь мы изменяем значение display каждого из элементов списка с block на inline, так что теперь до и после этих элементов не будет разрывов строки; они все будут располагаться в одной строке на странице подобно обычным строчным элементам.

<aside> который предназначен для размещения дополнительного содержимого страницы

<section> используется для группировки взаимосвязанного содержимого

<nav> для навигационного меню

<article> Мы поместили каждый блог-пост (h1, h2 и т.д.)в его собственный элемент

<mark> Данный элемент служит для выделения частей текста. Он является почти тем же самым,

что и чернильный маркер!

<progress> Нужно отобразить ход выполнения задачи? Например, «выполнено 90%»? Тогда используйте данный элемент<footer> Данный элемент определяет нижний колонтитул раздела либо всегодокумента целиком

<meter> Нужно отобразить измерение в том или ином диапазоне? Например, если речь идет о термометре со шкалой от 0 до 212, показывающем уличную температуру в 90 градусов по Фаренгейту? Как жарко!

<time> – это элемент для размещения значения времени, даты или даты/времени (например, 21 января, в 2 часа дня)

<canvas> Данный элемент используется для отображения на странице графики и анимации, генерируемых при помощи JavaScript

<header> Используйте данный элемент для разделов с верхними колонтитулами либо для верхнего колонтитула всего документа целиком

<nav> Применяйте данный элемент для группировкиссылок, используемых для навигации по сайту.

<footer> используется для группировки такого содержимого, как информация о документе, юридические сведения и данные об авторских правах, которые обычно размещаются внизу страницы или раздела.

‹table› - начало таблицы

‹tr› - первая строка

‹th› — это название столбца таблицы

‹td› содержит данные одной ячейки

<caption> Название отображается в окне браузера. По умолчанию в большинстве браузеров оно отображается над таблицей

border-spacing Расстояние между ячейками задается свойством

border-collapse, позволяющее объединить границы между ячейками так, чтобы расстояния между ячейками не было вовсе.

nth-child способ задать цвет для каждой из строк таблицы? Он основан на применении псевдокласса пример(p:nth-child(even) { background-color: red; })

rowspan, объединяет строки

colspan объединяет столбцы

disc — тип маркеров, используемый по умолчанию.

**position**: fixed; фиксированое размещение элемента

Начало формы

<form> Для создания формы применяется элемент

кнопка Submit (Отправить) (кнопка может также называться Submit Query (Послать запрос)).

**float** сначала берет элемент и смещает его влево или вправо, насколько это возможно (в зависимости от значения свойства). Затем все содержимое, находящееся под элементом, обтекает этот элемент

font-size: 1.2em; Допустим, вы таким образом задаете размер заголовков тогда размер будет составлять 1,2 размера шрифта родительского элемента, что в нашем случае означает в 1,2 раза больше, чем 14px, и примерно равняется 17px.

способ задать шрифты 2шт.

width: 200px; С помощью свойства width можно задавать ширину области содержимого элемента. В данном примере мы задаем ширину 200 пикселов.

text-align: center; выравнивание текста по центру

А что, если мне нужно что-то более сложное, например**<h2>, который является дочерним для элемента**, вложенного в элемент с идентификатором elixirs? #elixirs blockquote h2 { color: blue; }

1. Выберите ключевое слово (мы рекомендуем small или medium) и укажите его в качестве размера шрифта в правиле для body. Таким образом вы зададите размер шрифта, используемый на вашей странице по умолчанию.

2. Задайте размеры шрифтов для остальных элементов относительно размера шрифта элемента body, используя либо проценты, либо em (выбор остается за вами, так как, по сути, это два разных способа выполнения одного и того же действия).

пример body { font-size: small; }

h1 { font-size: 150%; }

h2 { font-size: 120%; }

font-weight насыщеность шрифтов-жирность

**Используйте ключевые слова, чтобы задать основной размер шрифта в правиле для body. В таком случае все браузеры смогут масштабировать размеры текста, если пользователи захотят, чтобы шрифт был больше или меньше.**

**ВИДЕО**

Атрибут preload обычно используется для тщательного контроля над загрузкой видео в целях оптимизации. Браузер решает, какой объем видео загружать, исходя из таких вещей, как, например, был ли задан атрибут autoplay, а также ориентируясь на пропускную способность канала пользователя. Вы можете изменить данное поведение, присвоив preload значение none (никакое видео не будет загружаться до тех пор, пока пользователь не нажмет на воспроизведение), либо значение metadata (метаданные видео будут загружаться, но без видеосодержимого), либо значение auto, которое позволит брау зеру самостоятельно принимать решение

font-style: italic; курсивный

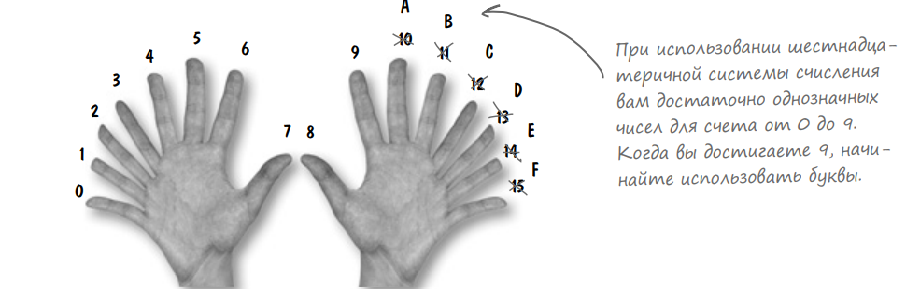
vertical-align: top; выравнивание всего содержимого в обеих ячейках по их верхнему краю

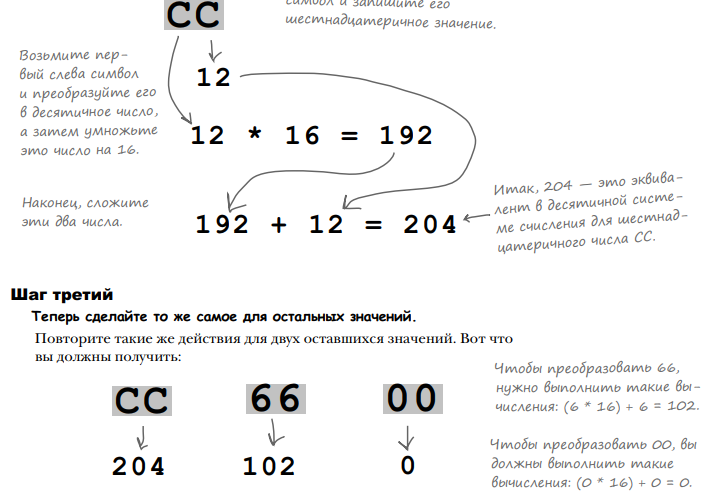
background-color: rgb(80%, 40%, 0%); 80 % от 255 — это 204, 40 % от 255 — это 102, и 0 % от 255 — это 0.

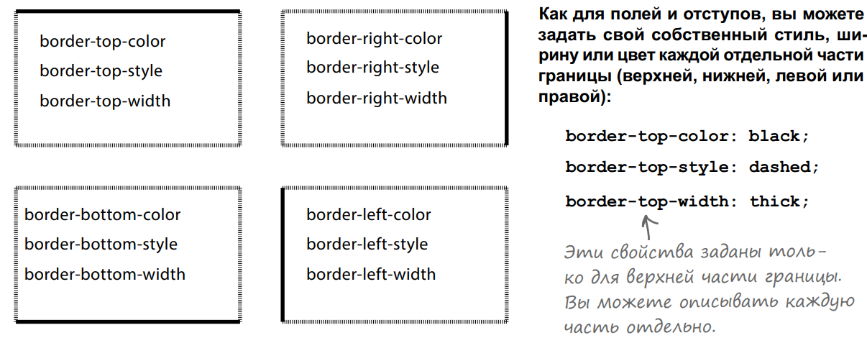
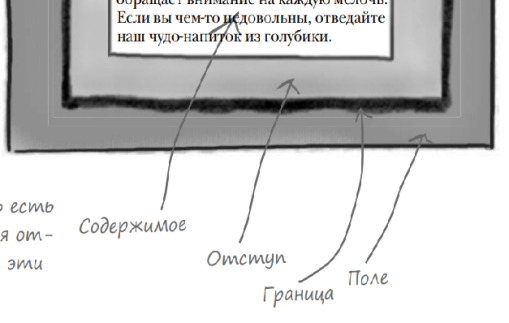
шестнадцатеричным код

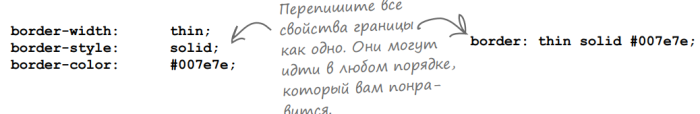
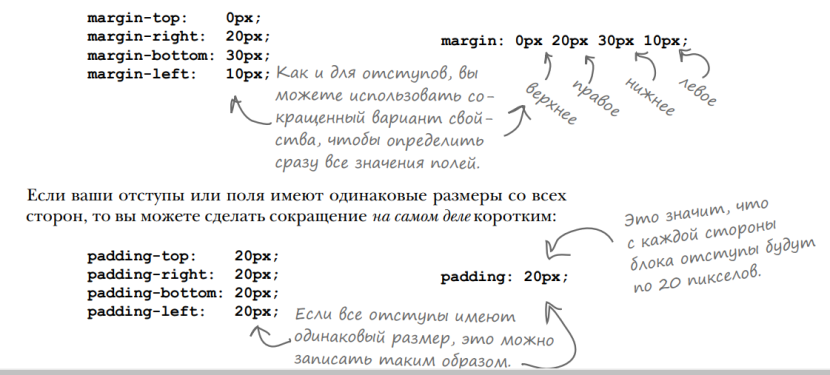
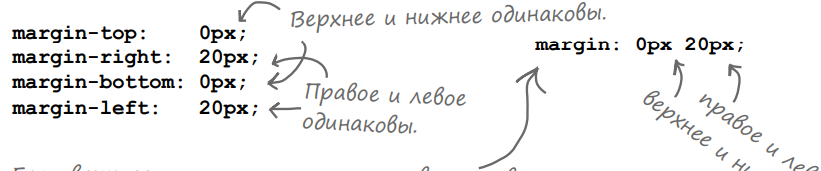
#cc6600 / сс красный/ 66 зеленый/ 00 синий

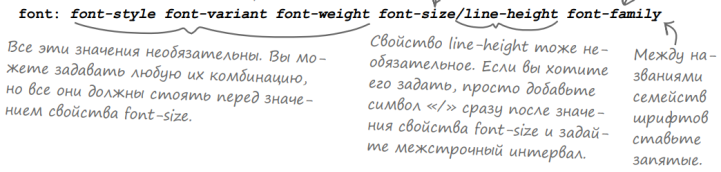
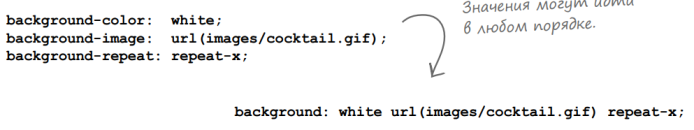
сс -12\*16=192/

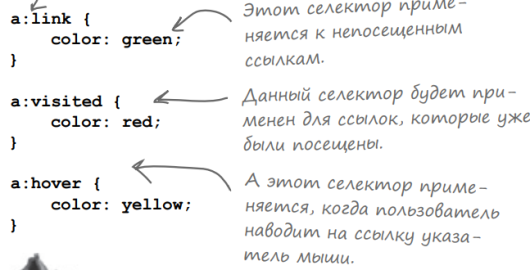
****

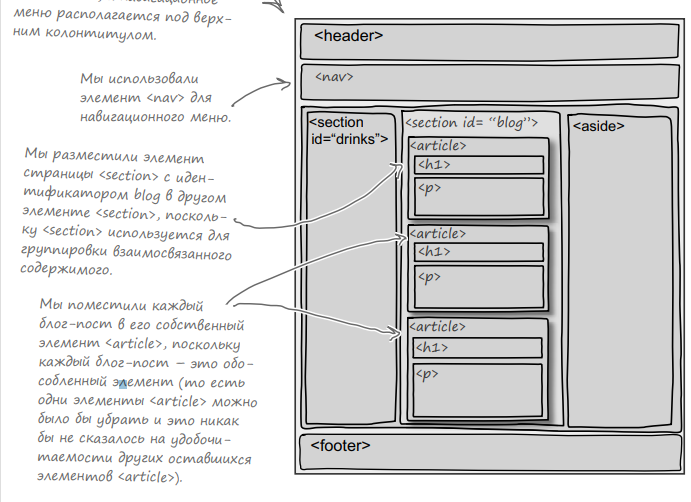
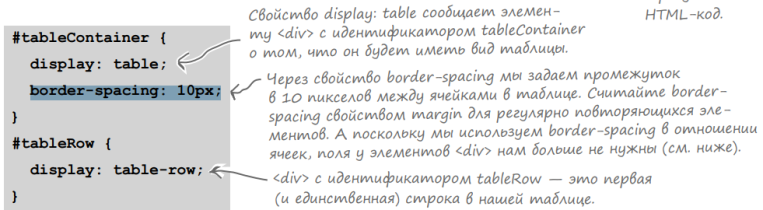
****

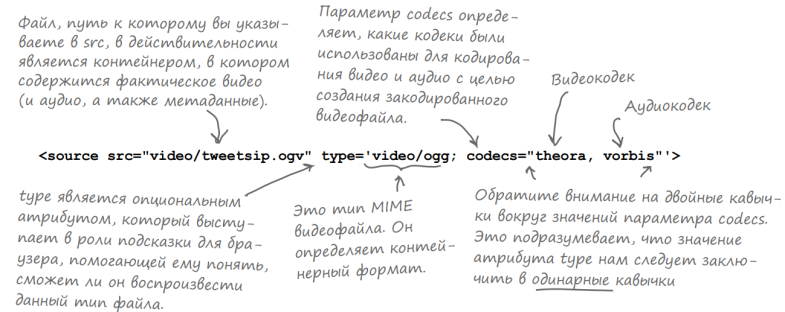
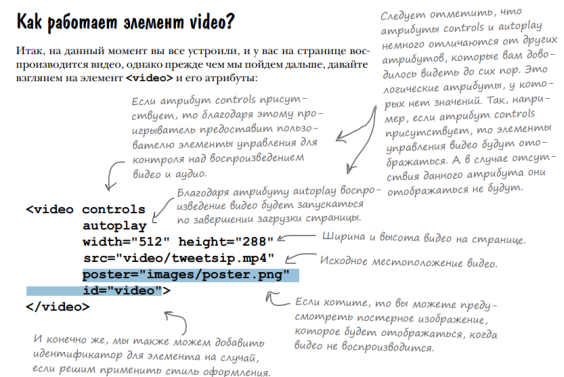
****

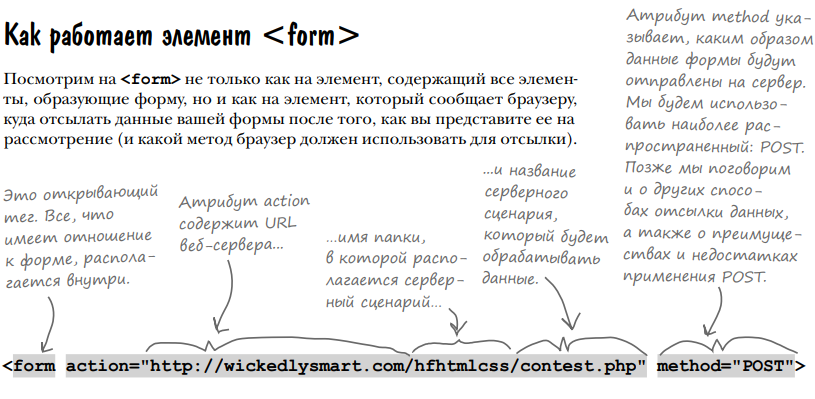
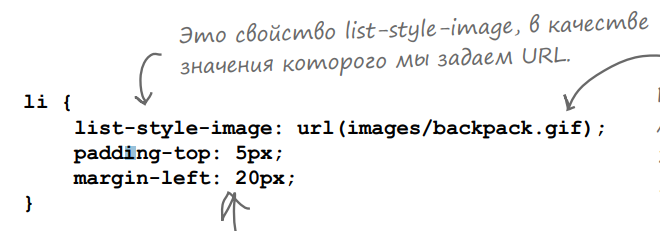
****

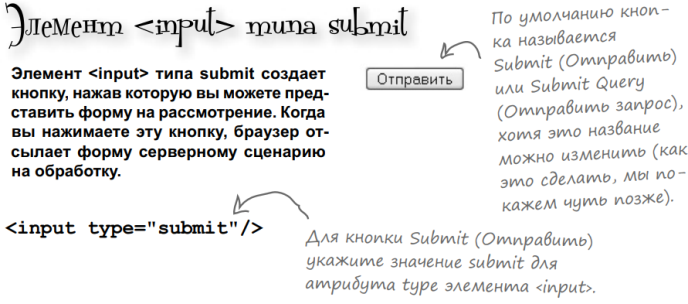
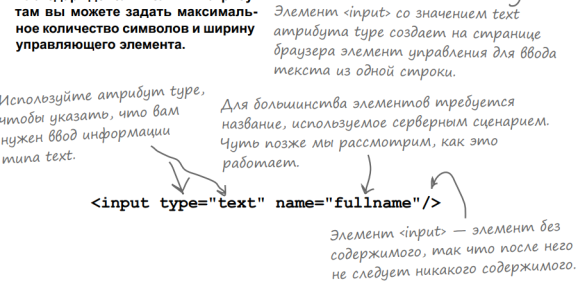
****

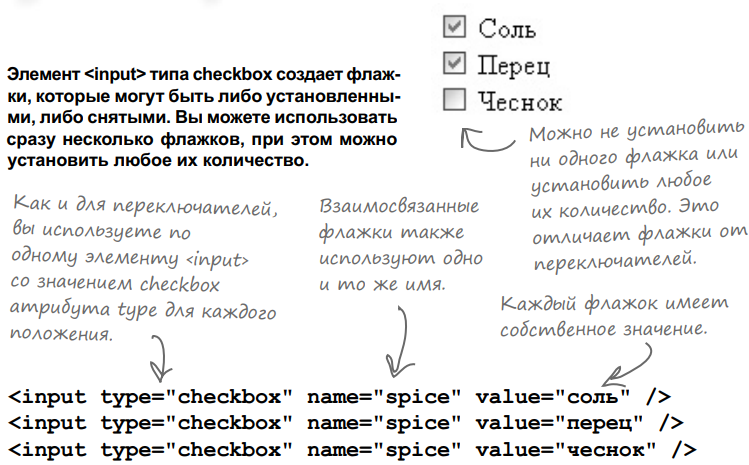
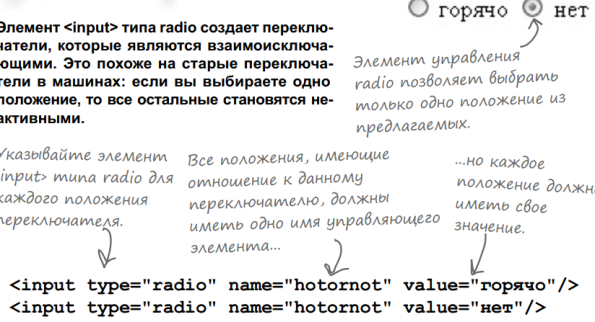
****

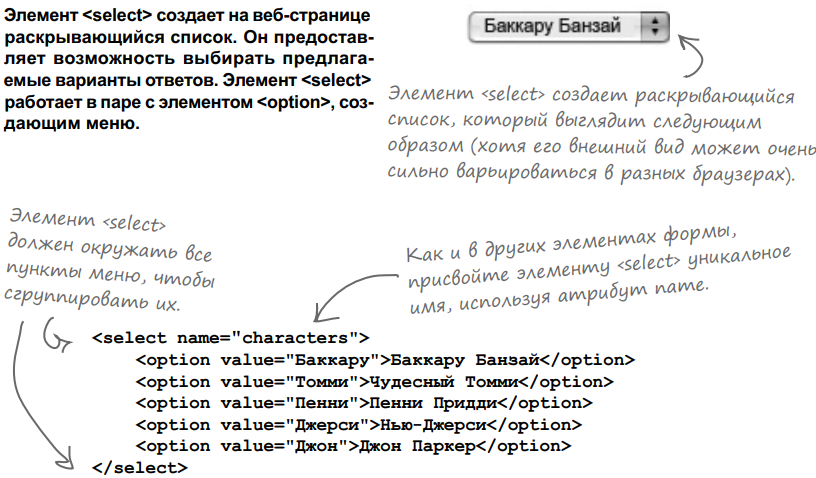
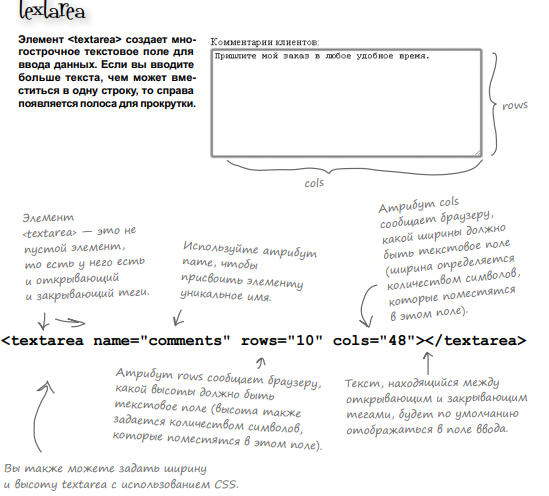
****

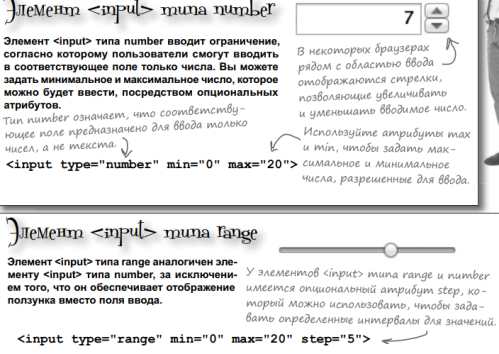
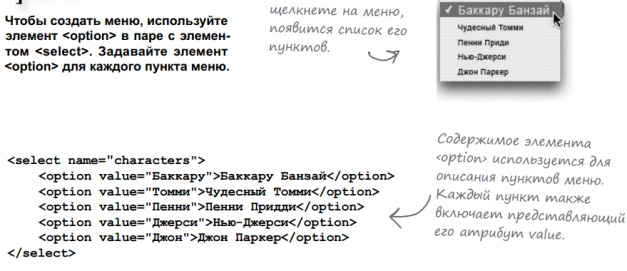
****

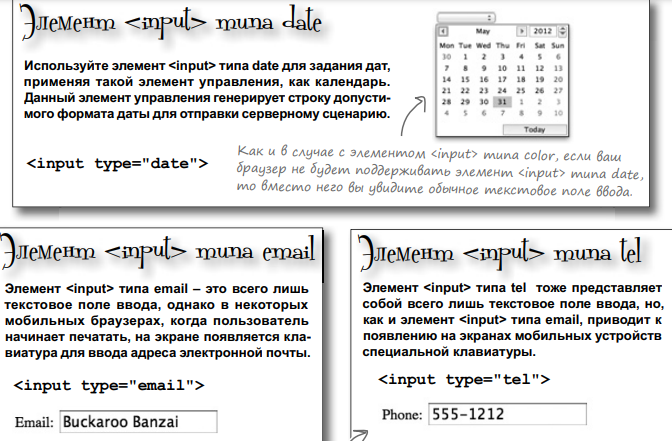
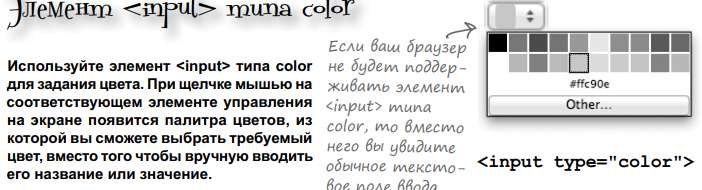
****

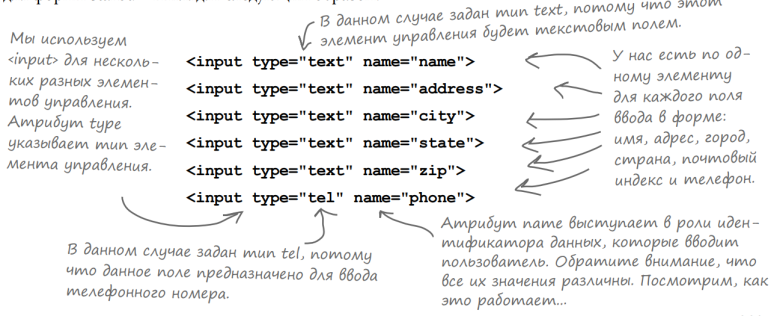
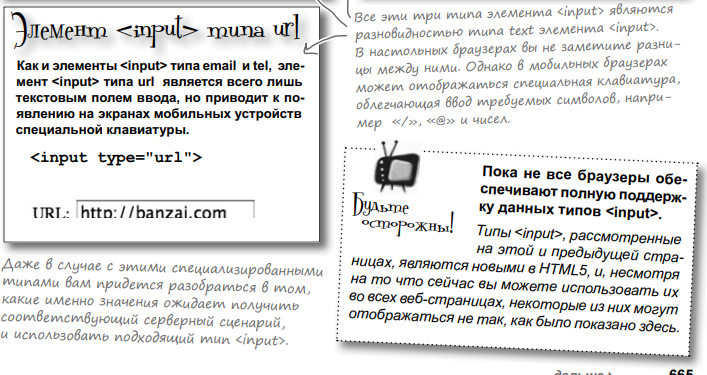
****

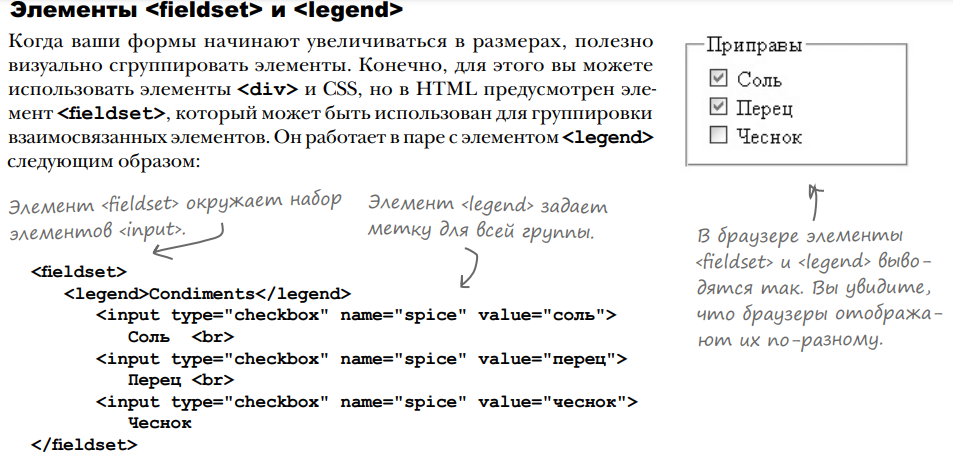
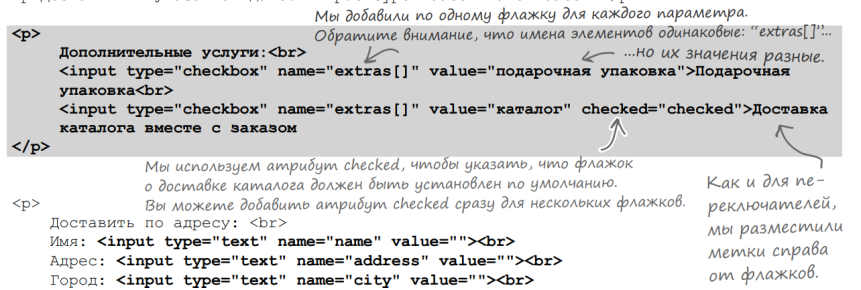
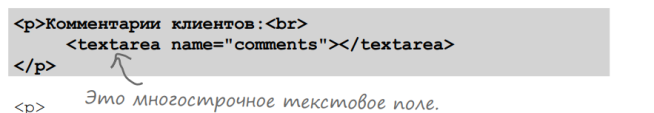
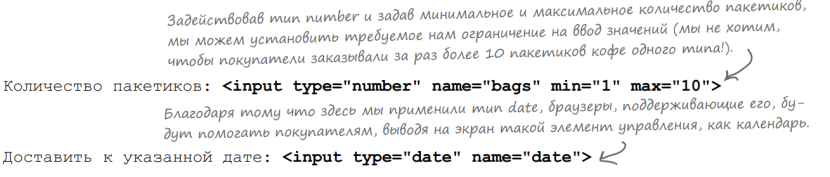
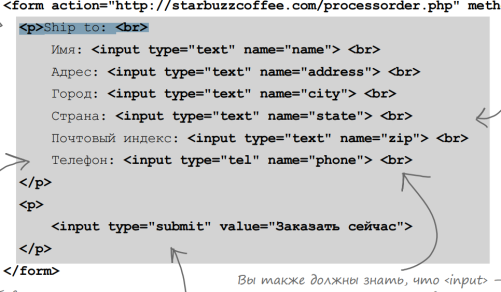
****

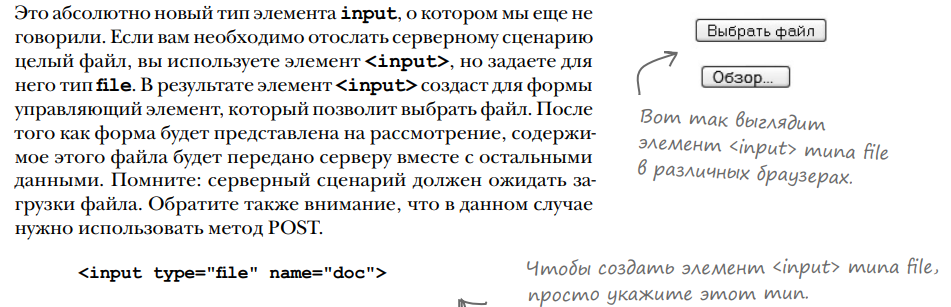
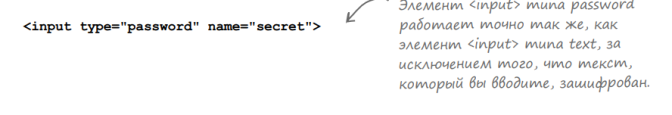
****

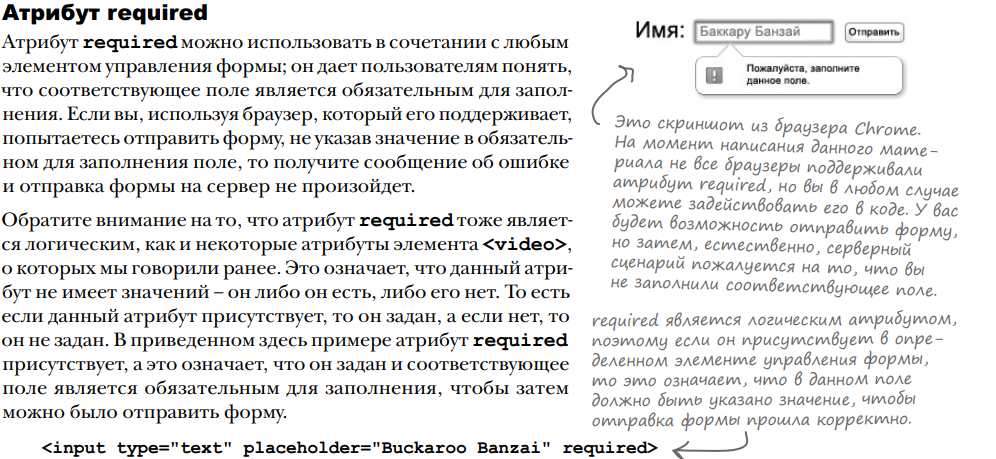
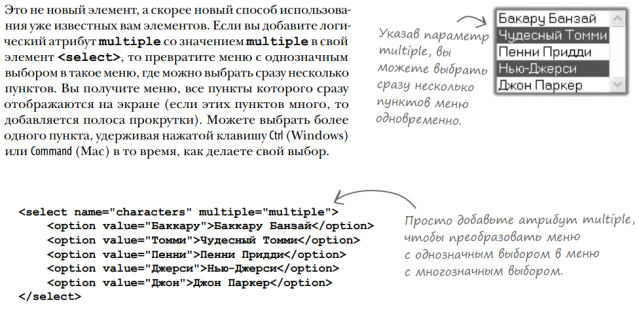
****

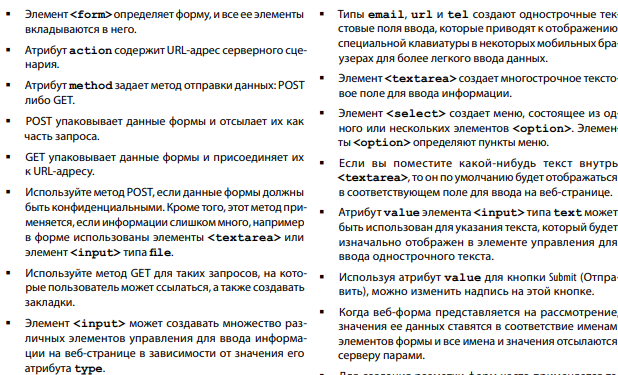
****

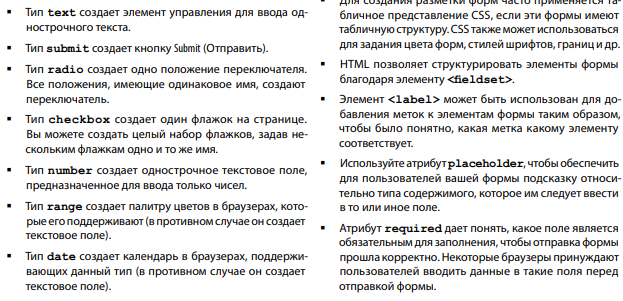
****

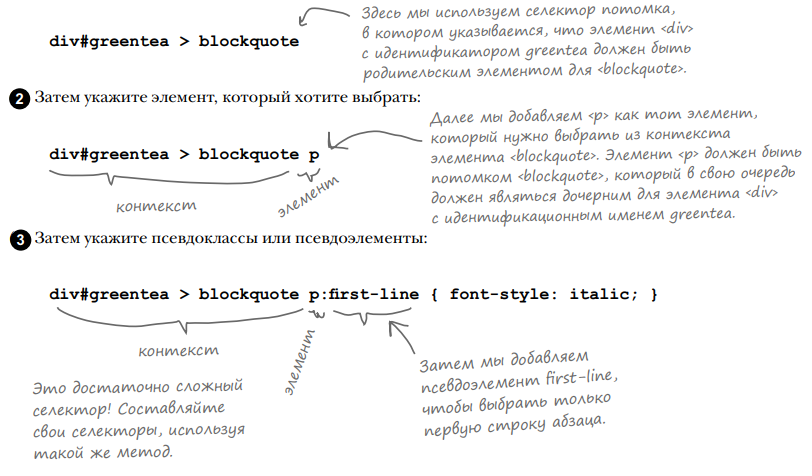
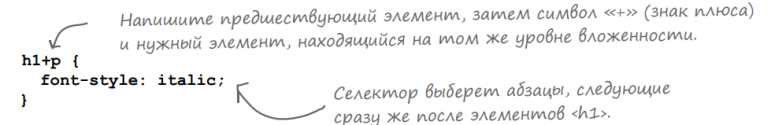
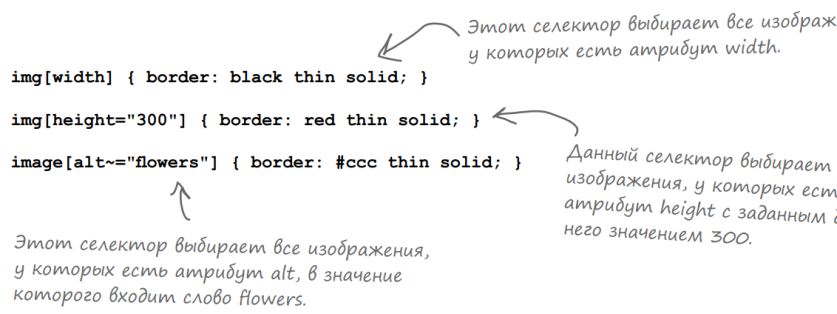
****

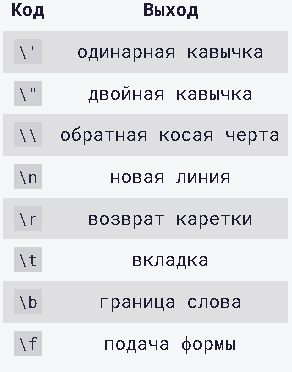
****

****

****

****

****

****

const arr = [

[1, 2, 3],

[4, 5, 6],

[7, 8, 9],

[[10, 11, 12], 13, 14]

];

arr[3];

arr[3][0];

arr[3][0][1];

arr[3]есть [[10, 11, 12], 13, 14], arr[3][0]есть [10, 11, 12]и arr[3][0][1]есть 11.

const arr1 = [1, 2, 3];

arr1.push(4);

const arr2 = ["Stimpson", "J", "cat"];

arr2.push(["happy", "joy"]);

arr1теперь имеет значение [1, 2, 3, 4]и arr2имеет значение ["Stimpson", "J", "cat", ["happy", "joy"]

const threeArr = [1, 4, 6];

const oneDown = threeArr.pop();

console.log(oneDown);

console.log(threeArr);

Первый console.logотобразит значение 6, а второй отобразит значение [1, 4].

const ourArray = ["Stimpson", "J", ["cat"]];

const removedFromOurArray = ourArray.shift();

removedFromOurArrayбудет иметь значение строки Stimpson, и ourArrayбудет иметь ["J", ["cat"]]

const ourArray = ["Stimpson", "J", "cat"];

ourArray.shift();

ourArray.unshift("Happy");

После shift, ourArrayбудет иметь значение ["J", "cat"]. После unshift, ourArrayбудет иметь значение ["Happy", "J", "cat"]

function myTest() {

const loc = "foo";

console.log(loc);

}

myTest();

console.log(loc);

Вызов myTest()функции отобразит строку fooв консоли. Строка console.log(loc)(вне myTestфункции) выдаст ошибку, так как locне определена вне функции.