

# ДИНАМИЧЕСКИЙ HTML



- Динамический HTML (dynamic HTML или DHTML) не является каким-то особым языком разметки страниц. Это всего лишь термин, применяемый для обозначения HTML-страниц с динамически изменяемым содержанием
- Реализация DHTML основывается на: непосредственно HTML, каскадных таблицах стилей (cascade style sheets – CSS) и языке сценариев (JavaScript или VBScript)

# Каскадные таблицы стилей (Cascade Style Sheets – CSS)



- Каскадные таблицы стилей предоставляют удобный способ добавления к web-страницам сложных атрибутов форматирования
- С их помощью можно определить гарнитуры и размеры шрифтов, способ выравнивания и другие характеристики, до мельчайших деталей определить параметры текста, цветов, графики, линий, цитат и т.д.
- Можно создать таблицу стилей и использовать её для всех документов, расположенных на сервере, что придаст стройность и строгость всему web-сайту
- Таблица стилей – это набор правил, который управляет форматированием тэгов HTML в web-документе

# Правило каскадных таблиц стилей



- Любое правило каскадных таблиц стилей состоит из двух частей: селектора и определения

**селектор {определение}**

- Селектор – любой тэг HTML
- Определение также состоит из двух частей: свойства и его значения, разделенных знаком двоеточия (:)
- В одном правиле можно задать несколько определений, разделённых символом точка с запятой (;)

**селектор {свойство1: значение1;  
свойство2: значение2; ...}**



- Синтаксис правил каскадных таблиц стилей не чувствителен к регистру (селекторы, свойства и их значения можно задавать как строчными, так и прописными буквами, или в смешанном порядке)
- Любая грамматическая ошибка приводит к тому, что правило пропускается анализатором браузера, и никакого предупреждающего сообщения не появляется

# Встраивание таблиц стилей в документ



## 1. Связывание

Таблица стилей храниться в отдельном файле  
Присоединяется с помощью тэга <LINK> в разделе  
HEAD

```
<LINK REL="stylesheet" TYPE="text/css"  
      HREF="mystyles.css">
```



## 2. Внедрение

Таблица стилей задаётся в стилевом блоке в разделе HEAD

Используется тэг <STYLE>

```
<STYLE TYPE="text/css">  
селектор {свойство: значение}  
...  
</STYLE>
```



### 3. **Импортирование**

В тэге <STYLE> можно импортировать внешнюю таблицу стилей

```
STYLE TYPE="text/css">  
@import: URL (адрес css);  
</STYLE>
```

### 4. **Встраивание в тэги документа**

Каждый тэг HTML имеет параметр STYLE, в котором можно задать значения его свойств в соответствии с синтаксисом каскадных таблиц стилей

```
<H1 STYLE= "color: red">  
заголовок отображается красным цветом  
</H1>
```

# Группирование



- Для уменьшения размеров таблиц стилей можно группировать разные элементы страницы HTML в виде списка через запятую, если для них задаётся одно правило

**H1 {font-family: Arial}**

**H2 {font-family: Arial}**

**H3 {font-family: Arial}**

- Можно задать в виде

**H1, H2, H3 {font-family: Arial}**



- Можно группировать свойства элемента

```
H1 {font-weight: bold}  
    H1 {font-size: 14pt}  
    H1 {font-family: Arial}
```

- Можно задать в виде

```
H1 {font-weight: bold; font-size: 14pt; font-family: Arial}
```

- Некоторые свойства имеют собственный синтаксис группирования, связанный с заданием значений нескольких свойств в одном

```
H1 {font: bold 14pt Arial}
```

# Селекторы



- Правила каскадных таблиц стилей влияют на отображение всех элементов заданного типа в документе

```
<STYLE TYPE="text/css">  
  H1 {color: red}  
</STYLE>
```

- Для одного тэга можно создавать различные классы

```
<STYLE TYPE="text/css">  
  H1.red {color: red}  
  H1.blueBgrd {color: red; background-color: blue}  
</STYLE>
```

# Использование классов



- В тексте документа ссылка на соответствующий класс задаётся в параметре CLASS

```
<H1 CLASS="red">
```

**Красный шрифт**

```
</H1>
```

```
<H1 CLASS="blueBgrd">
```

**Красный шрифт на синем фоне**

```
</H1>
```



- Если класс должен применяться ко всем элементам документа, то в селекторе задаётся имя класса с лидирующей точкой без указания конкретного элемента

```
<STYLE TYPE="text/css">  
  .red {color: red}  
  .blueBgrd {color: red; background-color: blue}  
</STYLE>
```

```
<H1 CLASS="red">  
  Красный шрифт  
</H1>  
<P CLASS="blueBgrd">  
  Красный шрифт на синем фоне  
</P>
```



- В качестве селектора можно использовать уникальное идентификационное имя элемента, предваряя его символом #

```
<STYLE TYPE="text/css">
```

```
#par24 {letter-spacing: 1 em}
```

```
H1#form3 {color: red; background-color: blue}
```

```
</STYLE>
```

```
<H1 ID=form3>Красный шрифт на синем фоне</H1>
```

```
<P ID=par24>
```

```
Разреженные слова в абзаце
```

```
</P>
```

# Свойства форматирования элементов



- В каскадных таблицах стилей все доступные свойства форматирования элементов в документе HTML разбиты на 8 категорий
  1. Шрифт
  2. Цвет и фон
  3. Текст
  4. Блок
  5. Визуальное форматирование
  6. Печать
  7. Фильтры и переходы
  8. Псевдоклассы и другие свойства

# Единицы измерения, используемые для задания значений свойств



Относительные единицы измерения	Абсолютные единицы измерения
<b>em</b> - высота элемента шрифта	<b>in</b> - дюйм (1 in = 2.54 cm)
<b>ex</b> - высота буквы	<b>cm</b> – сантиметр
<b>px</b> - пиксел	<b>mm</b> – миллиметр
<b>%</b> - процент	<b>pt</b> - пункт (1pt= 1/72 in)
	<b>pc</b> - пика (1 pc = 12 pt)

# Шрифты



Шрифты различаются по

- внешнему виду (начертанию)
- размеру
- стилю (прямой, курсив или наклонный)
- "жирности" отображения (нормальный, полужирный)

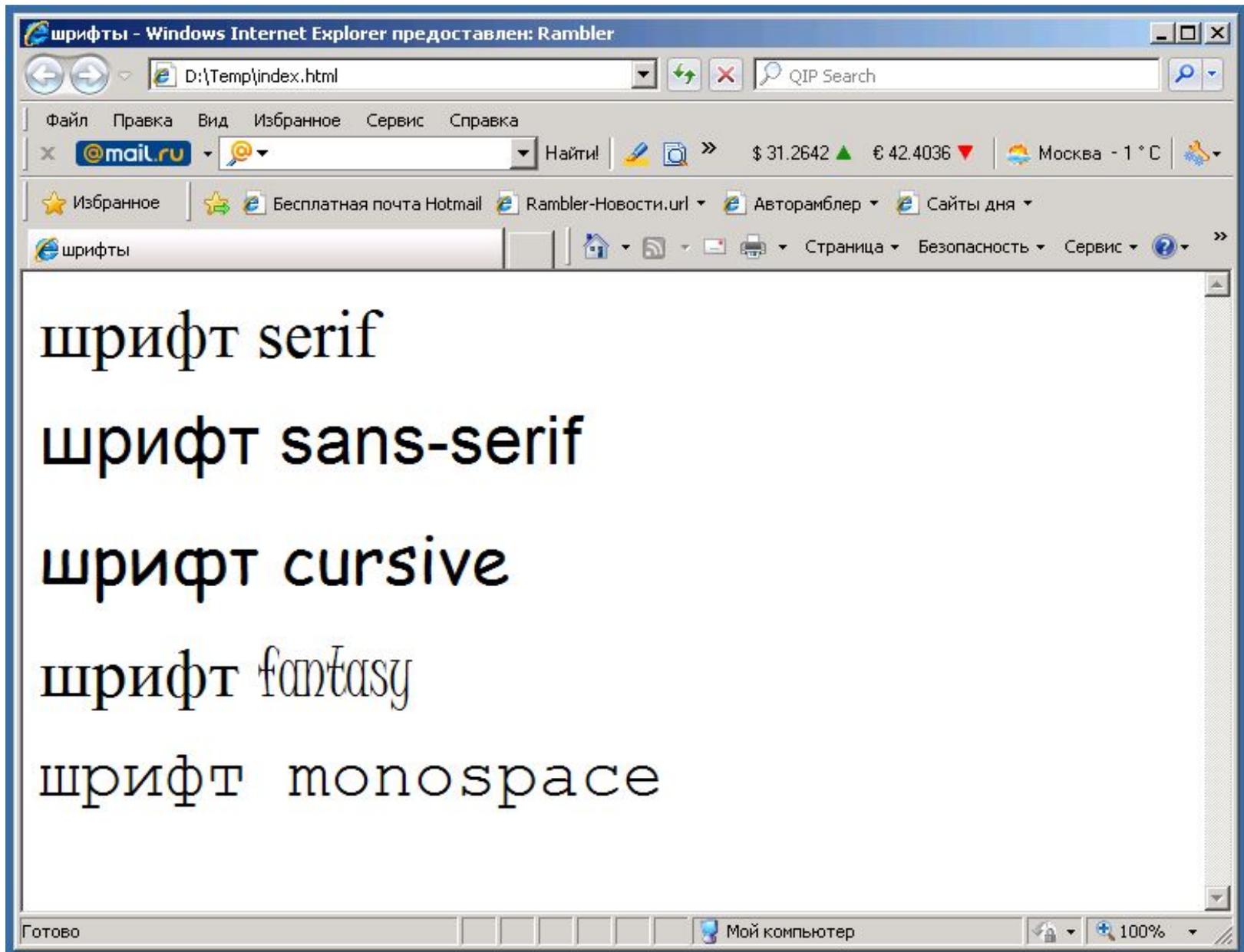
# Определение шрифта



- Свойство **font-family** задаёт приоритетный список шрифтов

**body {font-family: Impact, "Times New Roman", serif}**

- типовые семейства шрифтов
  1. serif (например, Times)
  2. sans-serif (например, Helvetica)
  3. cursive (например, Zapf-Chancery)
  4. fantasy (например, Western)
  5. monospace (например, Courier)



# Стиль шрифта



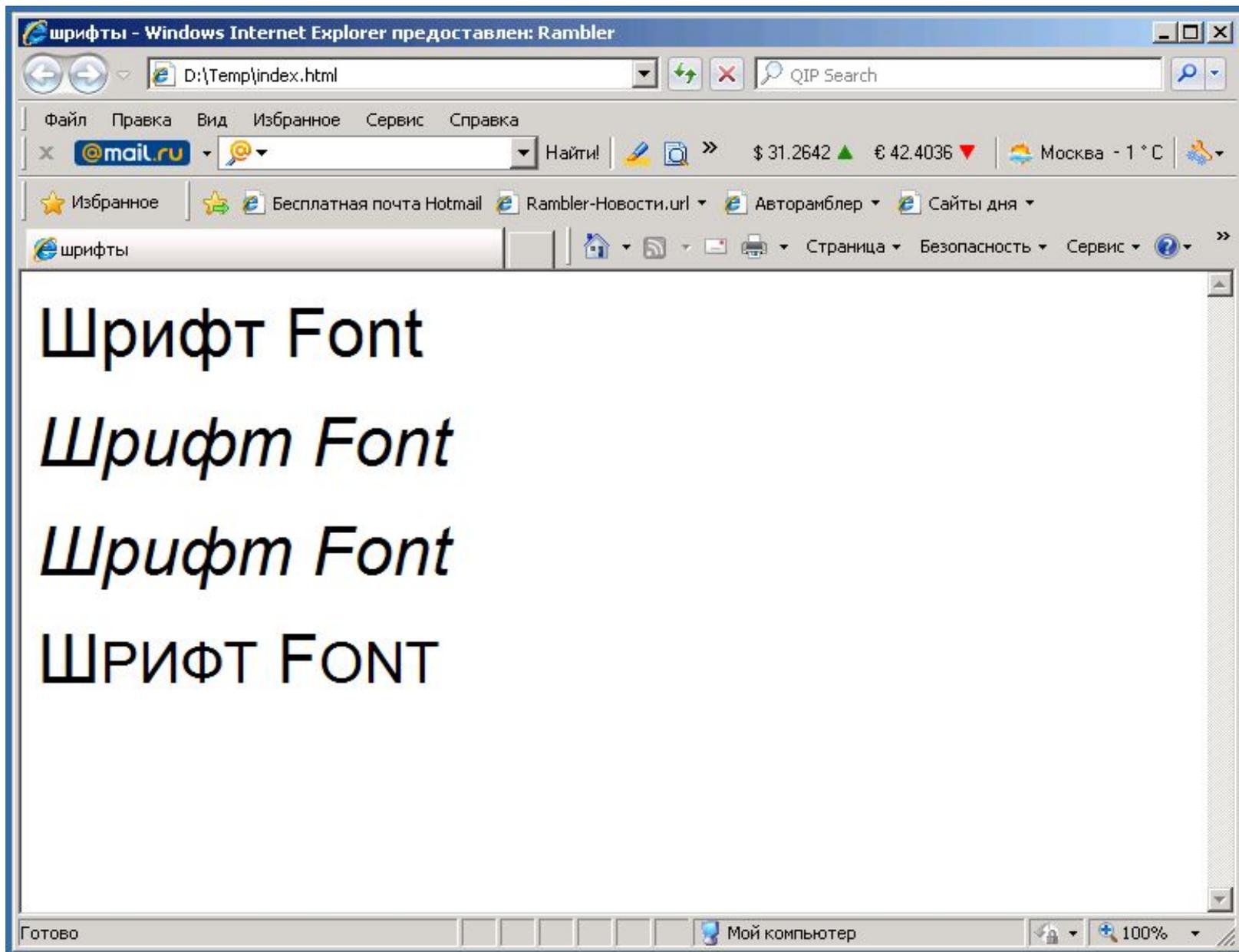
- Свойство **font-style** определяет стиль шрифта
  1. normal
  2. italic
  3. oblique
- **font-variant** - шрифт капитель (small-caps)

H1 {font-style: normal}

H2 {font-style: italic}

H3 {font-style: oblique}

H4 {font-variant: small-caps}



Шрифт Font

*Шрифт Font*

*Шрифт Font*

ШРИФТ FONT

# Жирность шрифта

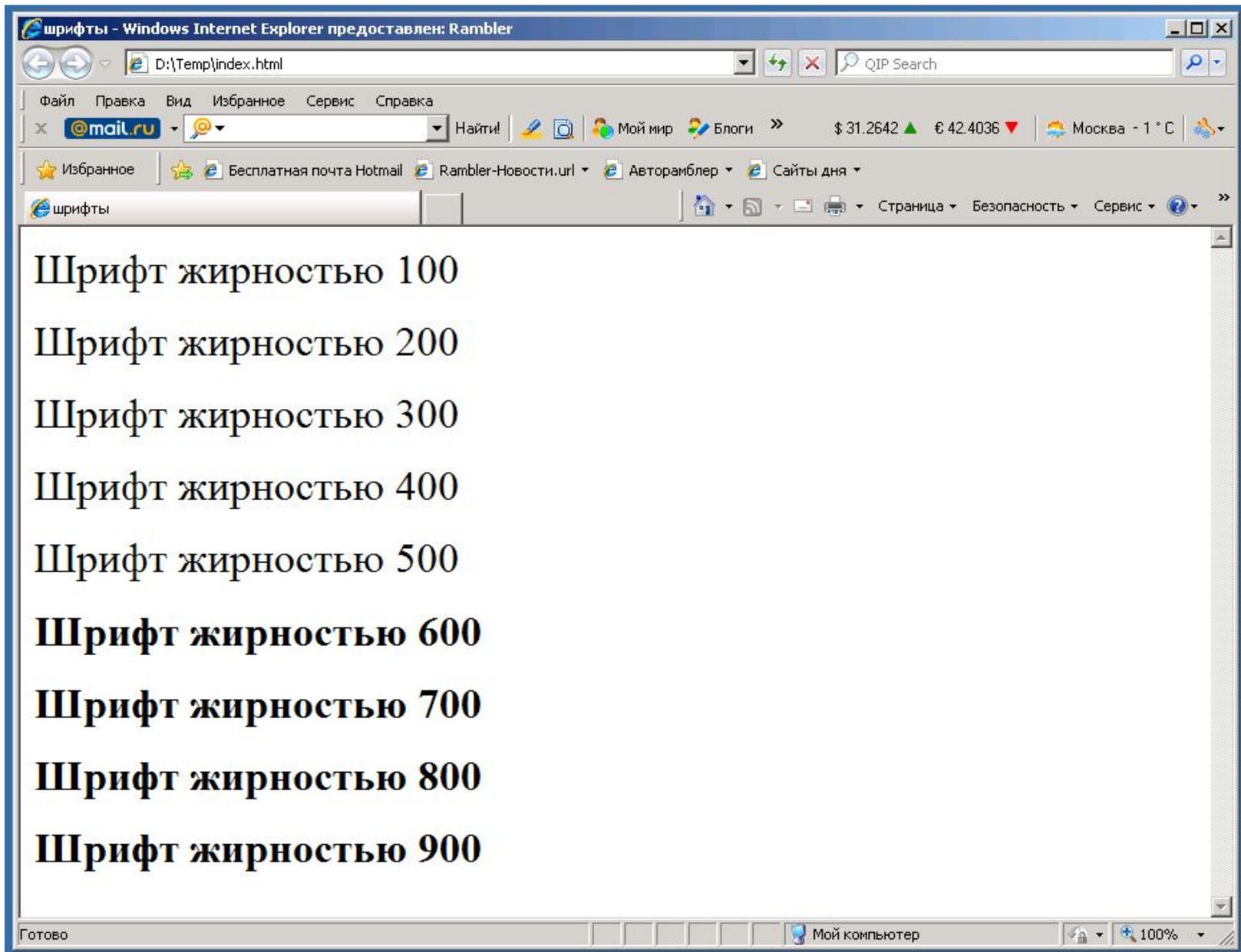


- Свойство **font-weight** выбирает из заданного семейства шрифт определённой жирности
- Числовые значения (9 категорий): 100 (самый бледный), 200, 300 .. 900 (самый жирный)
- Ключевые слова: normal, bold

**H1 {font-weight : normal}**

**H2 {font-weight : bold}**

**H3 {font-weight : 500}**



# Размер шрифта



- Свойство **font-size** определяет размер шрифта

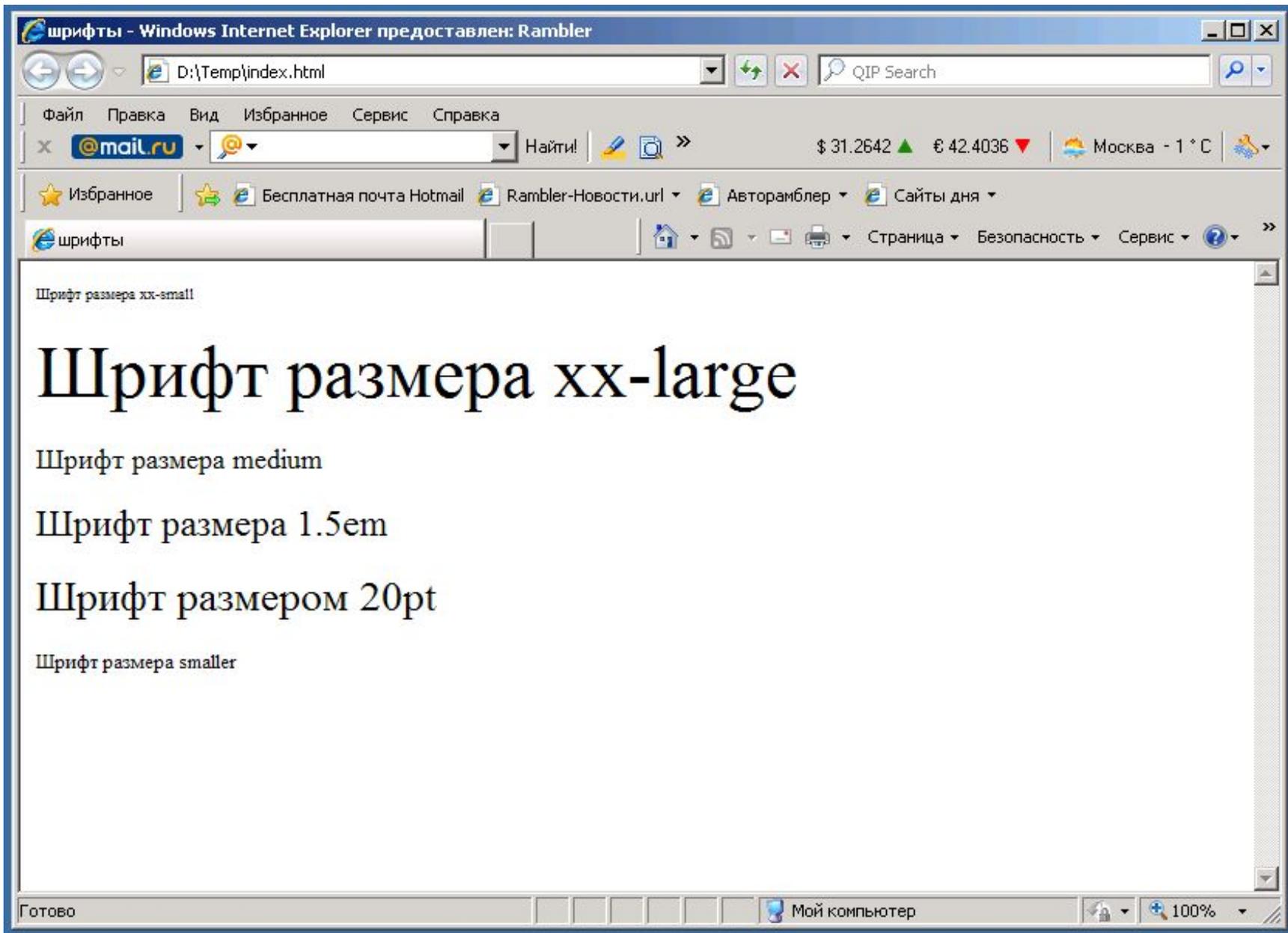
Абсолютное  
значение

- ключевые слова:  
xx-small, x-small,  
small, medium,  
large, x-large,  
xx-large
- абсолютное  
значение длины  
(pt)

Относительный  
размер

- ключевые слова:  
larger и smaller
- проценты
- относительные  
единицы длины  
(em)

- **line-height** - высота строки



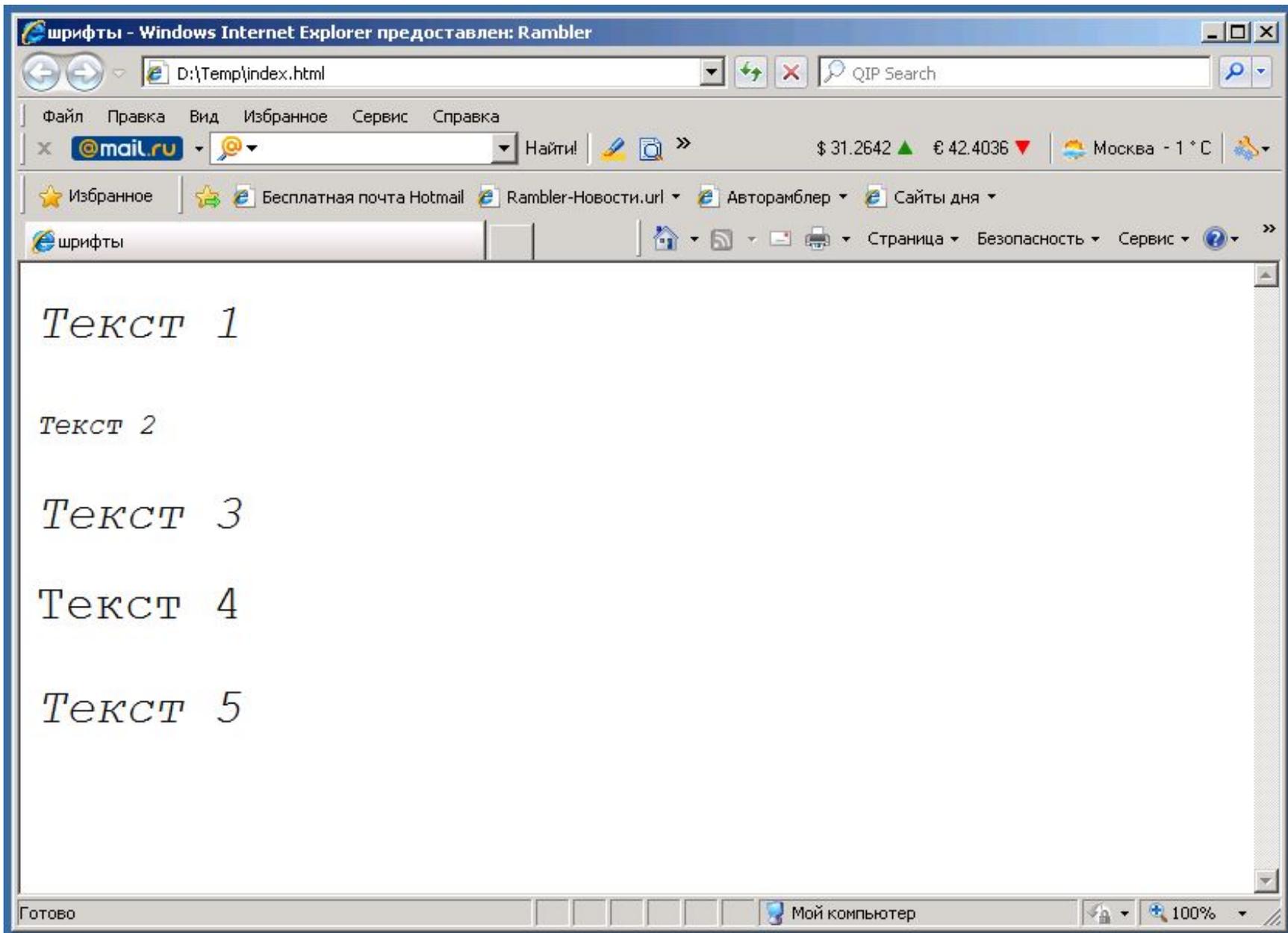


- Для сочетания всех возможных стилей для шрифта используется свойство **font**
- Все значения свойств задаются через пробелы в порядке font-style, font-variant, font-weight, font-size, line-height и font-family
- Первые три свойства могут не задаваться, что соответствует установке их значений в normal
- Размер шрифта и высота строки (свойство line-height) задаются через косую черту
- Элементы списка семейств шрифтов свойства font-family задаются через запятую

**P {font: oblique 12pt/14pt "Times Cyr", serif}**



```
<HTML>
  <HEAD>
    <TITLE>шрифты</TITLE>
    <STYLE TYPE="text/css">
      body {font: oblique 24pt/36pt "Courier New", monospace}
    </STYLE>
  </HEAD>
  <BODY>
    Текст 1
    <P STYLE = "font-size: 14pt"> Текст 2
    <P STYLE = "line-height: 24pt"> Текст 3
    <P STYLE = "font-style: normal"> Текст 4
    <P> Текст 5
  </BODY>
</HTML>
```



# Цвет и фон



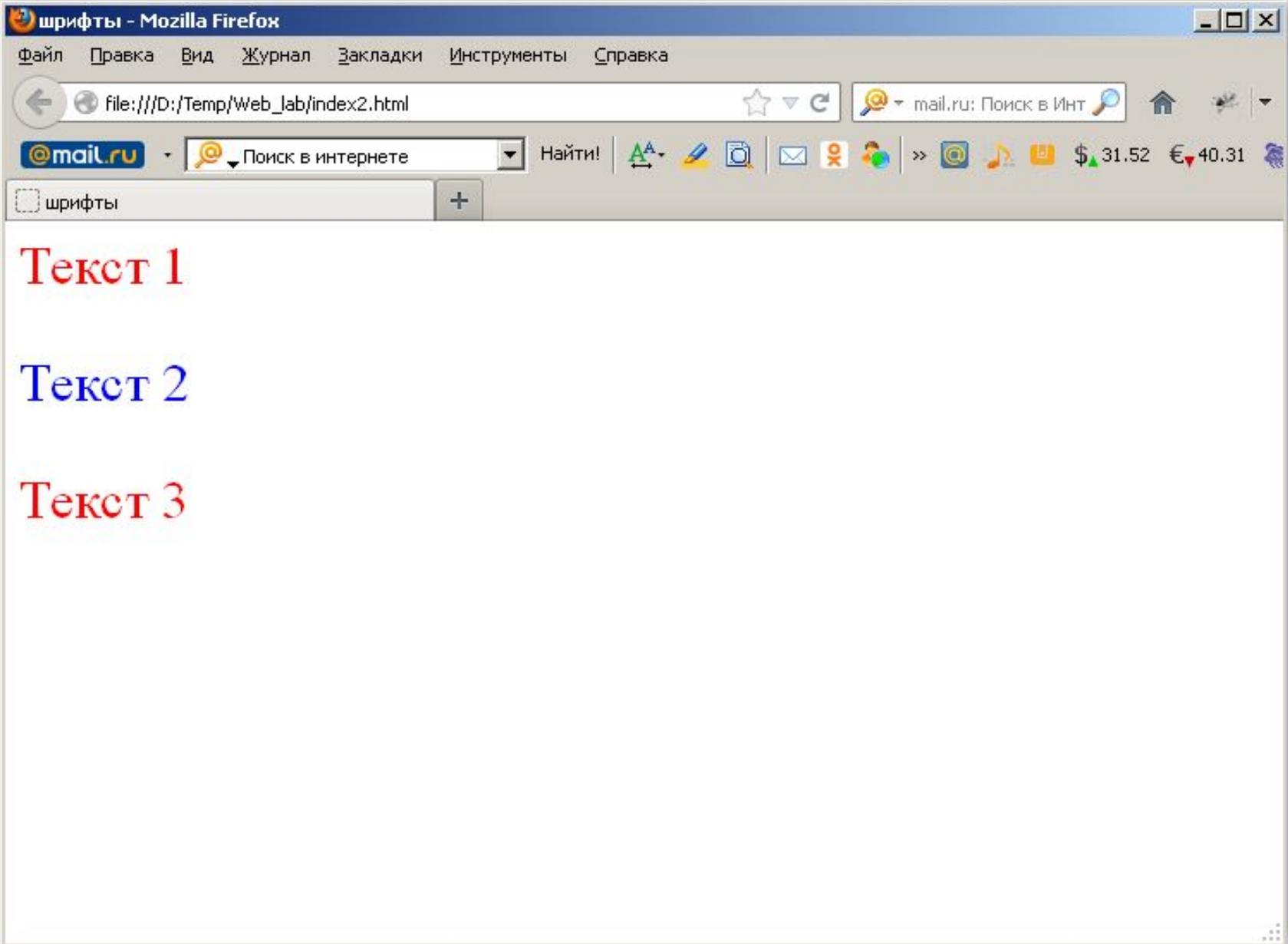
- цвет элемента
- фон элемента
  - цвет
  - изображение
    - положение
    - повторение
    - фиксация

# Цвет элемента



- **color** - цвет текста элемента

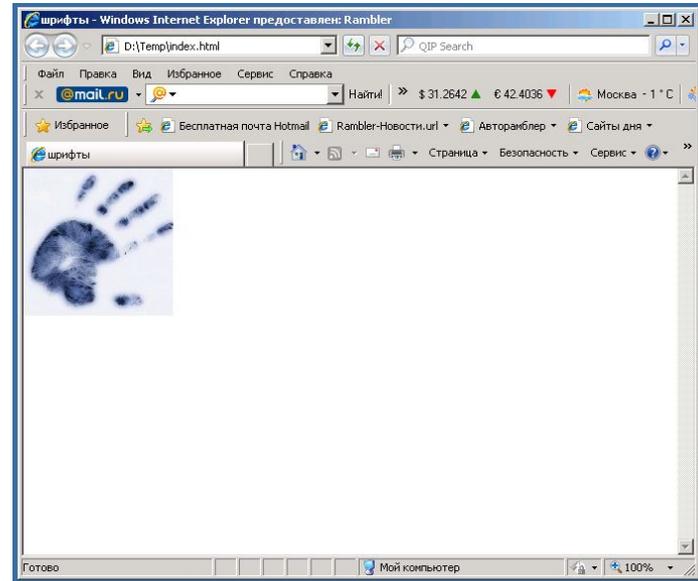
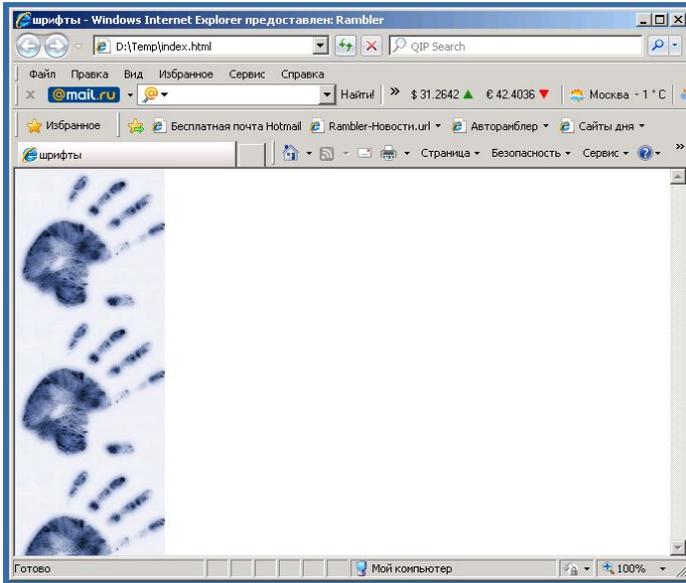
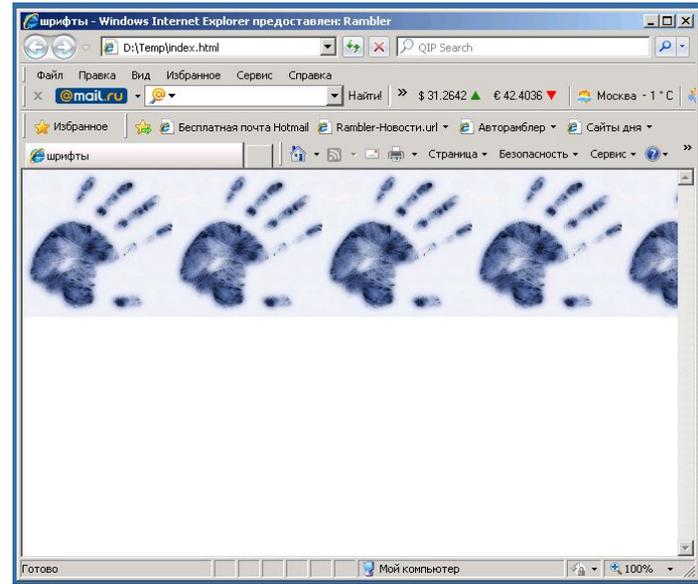
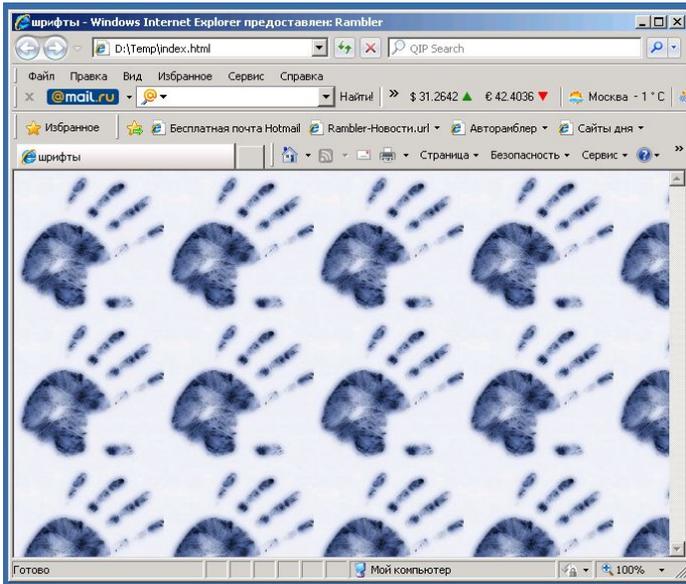
```
<HTML>
  <HEAD>
    <TITLE>шрифты</TITLE>
    <STYLE TYPE="text/css">
      body {font-size:24pt; color: red}
      p {color: blue}
    </STYLE>
  </HEAD>
  <BODY>
    Текст 1
    <P> Текст 2 </P>
    Текст 3
  </BODY>
</HTML>
```



# ФОН

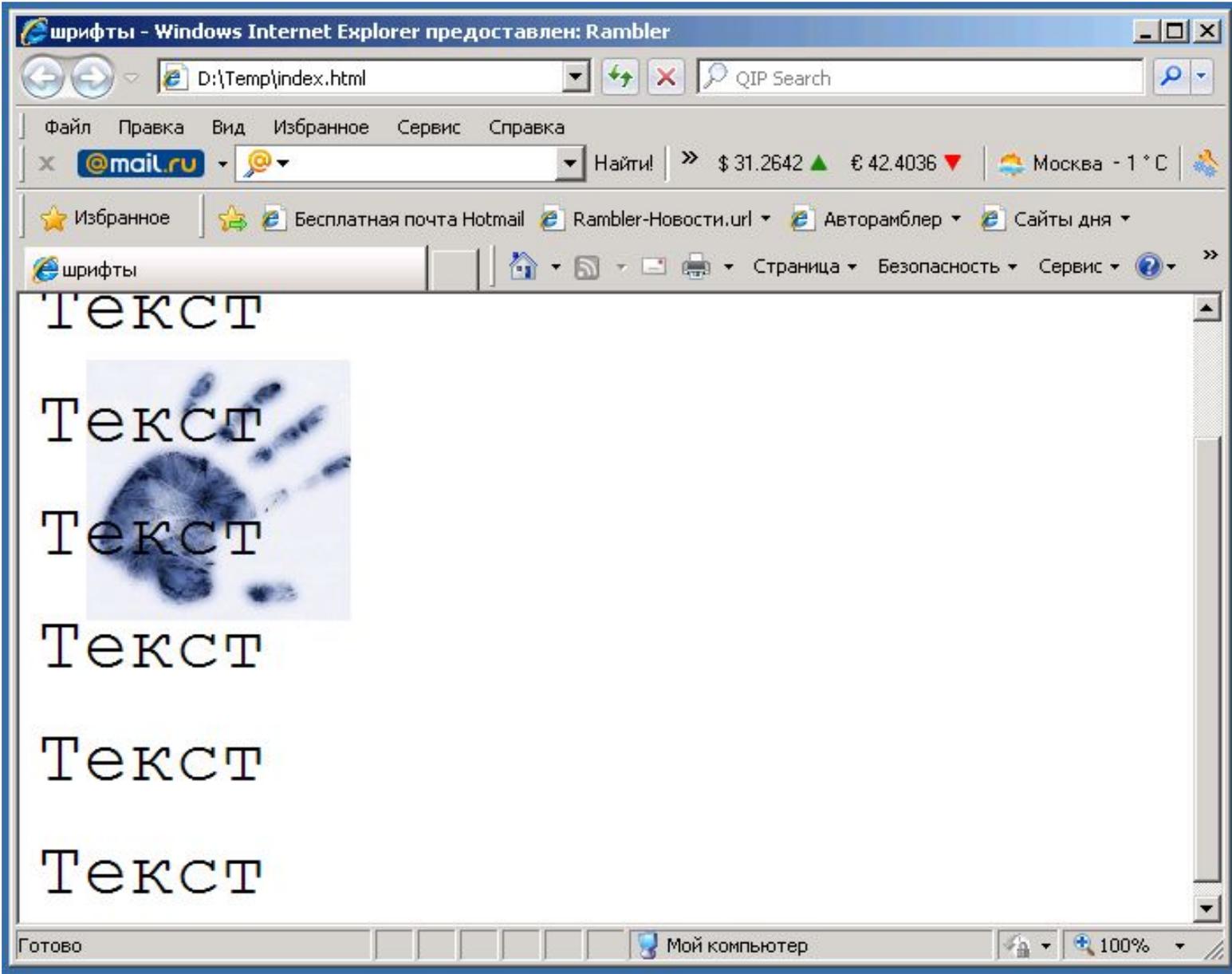


- **background-color** – цвет фона (начальное значение transparent - прозрачный)
- **background-image** – фоновое изображение
- **background-repeat** - повторяемость фонового изображения
  - repeat
  - repeat-x
  - repeat-y
  - no-repeat
- **background-attachment** – прокрутка фона
  - fixed
  - scroll





- **background-position** – начальное положение фонового изображения
  - ключевые значения: left, right, center, top и bottom
  - проценты
  - абсолютные единицы длины
- Свойство **background** позволяет одновременно устанавливать значения всех свойств фонового изображения
- **body {background: URL (pic1.jpg) no-repeat fixed 10mm 10mm}**



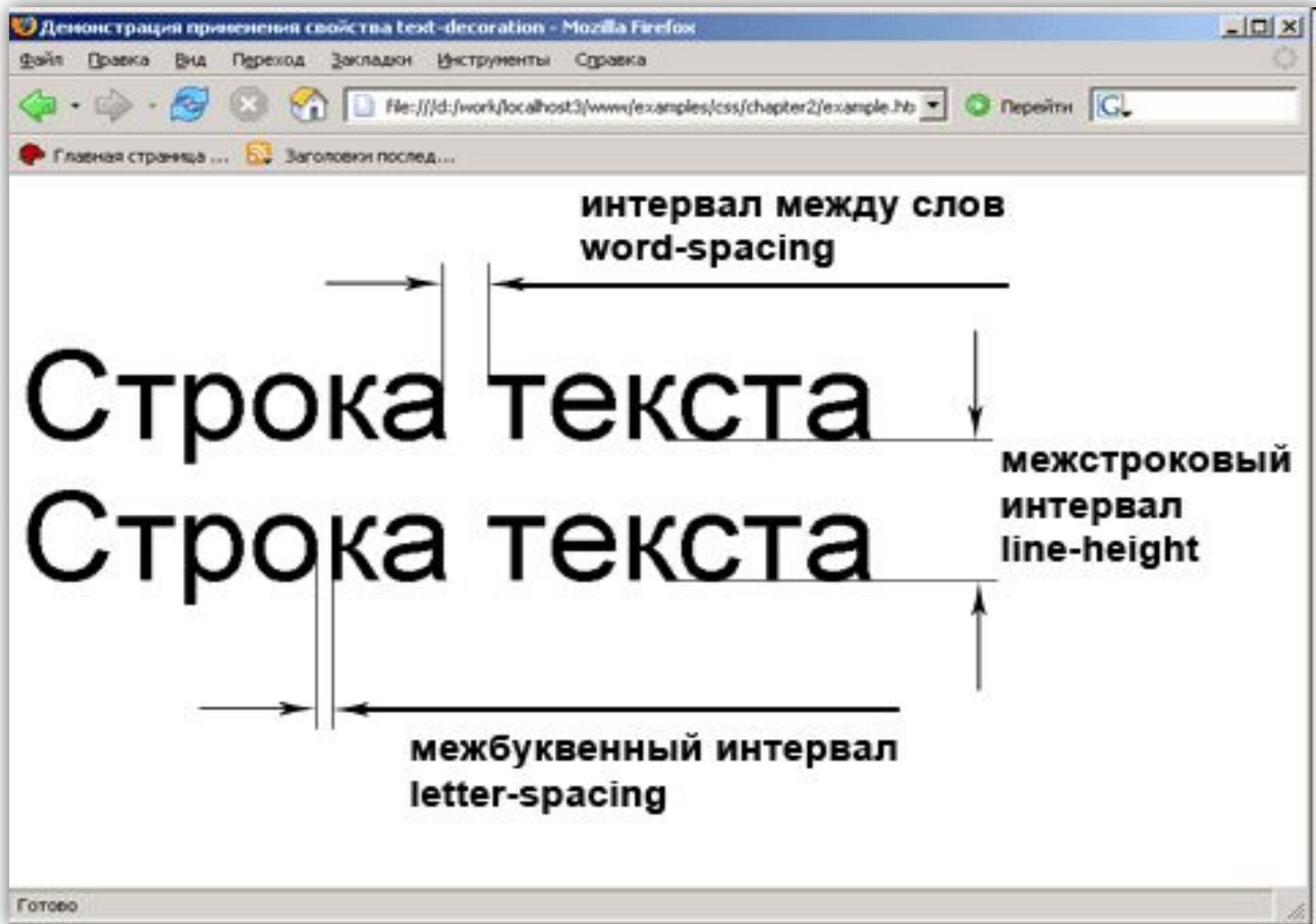
# Форматирование текста



- **Позволяет определить**
  - расстояние между словами
  - расстояние между буквами в словах
  - отступы и высота строк в абзацах



- **word-spacing** влияет на расстояние между словами
- **letter-spacing** влияет на расстояние между символами
- **line-height** – расстояние между базовыми линиями двух соседних строк
- **text-transform** – преобразование текста
  - capitalize – все слова отображаются с прописной буквы
  - uppercase – все буквы прописные
  - lowercase – все буквы строчные





- **text-decoration** - задаёт подчеркивание, надчеркивание или перечеркивание текста
  - underline – подчеркивание
  - overline – надчеркивание
  - line-through – перечеркивание
- **text-align** – выравнивание текста
  - left
  - right
  - center
  - justify
- **text-indent** – отступ первой строки



- **vertical-align** – положение элемента по вертикали относительно элемента-родителя
  - baseline – выравнивание базовой линии элемента по базовой линии родителя
  - middle – выравнивание средней точки элемента на уровне базовой линии родителя плюс половина ширины блока содержимого родителя
  - sub – элемент отображается в виде нижнего индекса
  - sup – элемент отображается в виде верхнего индекса
  - text-top – выравнивание верха элемента с верхом шрифта элемента-родителя
  - text-bottom – выравнивание низа элемента с низом шрифта элемента-родителя
  - top – выравнивание верха элемента с верхом самого высокого элемента строки
  - bottom – выравнивание низа элемента с низом самого низкого элемента строки

# Пример



```
<HTML>
  <HEAD>
    <TITLE>шрифты</TITLE>
  </HEAD>

  <BODY>
    <P>текст1
    <P STYLE="letter-spacing: 0.5em"> текст2
    <P STYLE="letter-spacing: 1em; text-transform: capitalize"> текст3
    <P STYLE="text-transform: lowercase"> тЕкСт4
    <P STYLE="text-decoration: underline; color: red">текст5
    <P STYLE="text-decoration: overline">текст6
    <P STYLE="text-decoration: line-through; color: blue">текст7
    <P STYLE="text-align: right; color:green">текст8
    <P STYLE="text-indent: 20">текст9
  </BODY>
</HTML>
```

