

ЗНАКОМСТВО С CSS. АТТРИБУТЫ СТИЛЯ CSS

Наиболее простым является добавление атрибута **style** к HTML-элементу, например:

```
<p style="color: red">
```

Этот абзац будет выведен красным</p>

```
<p style="color: green">
```

Этот абзац будет выведен зеленым</p>

Этот подход (**inline style**) удобен в случаях, когда требуется одноразово изменить отображение элемента.

Элемент **style** может быть помещен внутри элемента head HTML-документа.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
```

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head>
```

```
...
```

```
<style type="text/css" media="screen">
```

```
p {
```

```
text-align: justify;
```

```
text-indent: 1cm;
```

```
}
```

```
.RedParagraph {
```

```
color: red;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p>
```

Текст обычного абзаца</p>

```
<p class="RedParagraph">
```

Этот абзац наследует форматирование обычного абзаца плюс красный цвет текста</p>

```
...
```

```
</body>
```

```
</html>
```

Элемент **style** имеет ряд атрибутов, из которых обязательным является лишь атрибут **type**, описывающий язык таблицы стилей (в примере **text/css**).

Атрибут **media** определяет, для какого носителя предназначена стилевая информация; возможно как единичное значение, так и разделенный запятыми список, например, `media=screen,print`.

Элемент **style** может находиться только в элементе **head** и встречаться там любое число раз. Внутри элемента **style** помещается стилевая информация, называемая **правилами**. Правила зависят от языка таблицы стилей. Правила CSS будут рассмотрены ниже.

Другой подход заключается в использовании внешних таблиц стилей. Этот подход будет рассмотрен в следующих лабораторных работах.

Стилевая информация каскадной таблицы стилей состоит из отдельных строительных блоков, называемых **правилами**. Каждое правило, в свою очередь, состоит из двух частей, **селектора** и **объявления стиля**, заключенного в фигурные скобки ({}).

Селекторы используются для отбора ("селекции") элементов HTML-страницы, к которым будет применено правило. Объявление состоит из пар имя свойства – значение, разделенных двоеточием (:). Пары между собой разделяются точкой с запятой (;).

```
h1 {  
    color: red;  
    font-size: large;  
}
```

Точка с запятой после последней пары необязательна, однако ее наличие считается хорошей практикой.

Задания:

1. Ввод заголовка и абзаца текста

В теле документа введите элемент заголовка первого уровня с произвольным текстом. Затем введите элемент абзаца - также с произвольным текстом.

2. Добавление таблицы каскадных стилей

Браузер отображает текст черным цветом на белом фоне, начиная с левого верхнего угла окна (точнее, как правило, делая отступ в 2 пиксела со всех сторон) во всю его ширину.

Измените этот стиль по умолчанию на собственный. Добавьте в раздел **head** элемент **style** с атрибутом **type="text/css"**. Добавьте в него пустое правило для элемента **body**:

```
<style type="text/css">  
body {  
}  
</style>
```

Теперь по порядку назначим при помощи этого правила элементу **body** следующий ряд свойств:

- цвет фона (неяркий)
- толстую серую рамку
- большой отступ - чтобы отдалить рамку от границ окна
- небольшое поле - чтобы отделить текст от рамки
- крупный размер шрифта (например, 20px)

- гарнитуру Comic Sans MS
- отступ первой строки

Уберете рамку справа и снизу, назначив свойствам **border-bottom** и **border-right** значение **none**.

3. Вставка изображения

Вставьте в документ элемент `img`.

4. Вставка элементов логического форматирования

Добавьте заголовок второго уровня с произвольным текстом и неупорядоченный список с двумя-тремя элементами. Просмотрите результат в браузере.

Вероятно, в список проникли отступы первых строк, определенные ранее для всех элементов в **body**. Очевидно, отступы первых строк должны быть определены не для всех элементов без разбора - достаточно, чтобы их имели абзацы и заголовки:

```
p, h1, h2 {
  text-indent: 40px;
}
```

Переопределите правило для отступа первой строки указанным образом.

Сделаем врезку в документе - блок текста, выделенный особым форматированием - как правило, гарнитурой и начертанием. Вставьте в документ элемент **blockquote** (цитату) с произвольным текстом из 1-2 предложений. Создайте правило для элемента **blockquote**, задающее для него другую гарнитуру (например, *Serif*) и начертание (например, *italic*).

Вставьте в документ последний, относительно длинный, абзац. Сделайте в нем выделения при помощи элементов логического форматирования **strong** и **span**. Определите какую-либо аббревиатуру (**abbr**).

Элементу **span** назначьте атрибут **class="greenspan"**. Добавьте правило стиля с селектором класса, делающее цвет текста в элементе этого класса зеленым и подчеркнутым:

```
.greenspan {
  color:Green;
  text-decoration:underline;
}
```

5. Создание рамки изображения

Добавьте стилевое правило для элемента **img**, включающее тонкую сплошную серую рамку и поле:

```
img {
  padding: 10px;
  border: 1px solid #777;
}
```

6. Центрирование изображения с подписью

Поместите изображение и его подпись (следующий элемент **p**) в секцию, т.е. создайте пустой элемент **div** и переместите в него элементы **img** и **p**.

Теперь позиционируем секцию горизонтально по центру страницы - для этого нужно явно задать ее ширину и затем установить автоматические отступы слева и справа. Ширина секции равна ширине изображения плюс удвоенный отступ изображения плюс удвоенная толщина рамки изображения.

Добавьте элементу **div** атрибуты **class="pic" style="width:689px"**: стилевое правило, центрирующее секцию, имеет смысл определить для множества (класса) секций; а ширина секции привязана к ширине изображения, поэтому атрибут ширины лучше определить во встроеном стиле. Для класса **pic** стиль должен выглядеть следующим образом:

```
.pic {  
    margin-left: auto;  
    margin-right: auto;  
}
```

Проверьте, что получилось в результате.

7. Добавление рамки и фона абзацу

Создайте стилевое правило для абзаца, помещенного в секцию с классом **pic**:

```
.pic p {  
}
```

Селектор **.pic p** означает: элемент **p**, находящийся на произвольном уровне вложенности в элемент класса **pic**.

Определите в этом стилевом правиле следующие свойства:

- выравнивание текста - по центру;
- поле - 5 пикселей;
- рамка - тонкая пунктирная (dashed) серая;
- шрифт полужирный;
- фон - серый (немного темнее общего фона).

8. Единицы размеров

Все размеры и позиционные расположения в CSS задаются в размерных единицах.

Абсолютные единицы задают точный размер, например, в сантиметрах или дюймах, относительные единицы вычисляются относительно каких-либо других свойств (например, размера монитора или листа бумаги) или размеров других элементов.

Допустимые абсолютные единицы:

- in – дюймы (2,54 см)
- cm – сантиметры
- mm – миллиметры
- pt – пункты (points, 1pt = 1/72in)

- pc – пики (picas, 1pc = 12pt)

Относительные единицы:

- em – размер на основе размера шрифта (атрибута font-size). В типографии em — это единица измерения, которая представляет высоту заглавной буквы М шрифта. В веб-дизайне 1 em — это высота базового шрифта в браузере, которая обычно составляет 16 пикселей (но пользователь может изменять ее). Если эта единица применяется для определения размера шрифта, то она имеет смысл относительной величины по отношению к размеру шрифта в родительском элементе.

Пример: left: 2.5em

- ex – размер буквы x
- px – вычисляется на основе разрешения монитора или принтера
- % – размер относительно другого, как правило, родительского элемента.

Например, ширина ячейки таблицы может быть выражена в процентах от ширины таблицы

Ниже приведен пример использования различных единиц измерения для задания ширины

```
div {
    border: 1px solid black;
}
#px {
    width: 200px;
}
#percent {
    width: 40%;
}
#em {
    width: 17em;
}
...
<div id="px">
    Ширина этой секции указана в пикселях
</div>

<div id="percent">
    Ширина этой секции указана в процентах
</div>

<div id="em">
    Ширина этой секции указана в em
</div>
```

Измените в примере:

- масштаб страницы;
- размер шрифта;
- ширину окна.

Проследите за шириной блоков.

9. Цвета и шрифт

Цвет в CSS может быть задан по имени или в виде шестнадцатеричного числа, определяющего интенсивности красного, зеленого и синего цветов (RGB схема).

Стандарт HTML 4.01 определяет только 16 цветов, спецификация CSS 2.1 добавляет еще один цвет; RGB схема позволяет задать свыше 2,000,000 цветов, от черного (#000000) до белого (#FFFFFF).

Следующий пример показывает способы задания красного цвета текста:

```
p { color: red; }  
/* стандартный цвет */  
p { color: #f00; }  
/* #rgb */  
p { color: #ff0000; }  
/* #rrggbb */  
p { color: rgb(255,0,0); }  
/* целые в диапазоне 0 – 255 */  
p { color: rgb(100%, 0.0%, 0%); }  
/* диапазон 0.0% - 100.0% */
```

Описание шрифта и форматирования текста состоит из целого ряда отдельных атрибутов. Многие термины здесь почерпнуты из типографской практики.

Font-family (гарнитура) задает название используемого шрифта. Поскольку нет полной уверенности, что указанный шрифт установлен и доступен в любом браузере и на любой клиентской машине, разработчик может задать список альтернативных шрифтов в порядке убывания предпочтительности, разделив список запятыми. Если название шрифта содержит пробелы, оно должно быть заключено в кавычки. Пример:

```
p {font-family: 'Times New Roman', Times, serif;}
```

Font-size (кегель) задает размер шрифта в абсолютных или относительных единицах или относительно пользовательских предпочитаемых размеров.

Font-weight задает насыщенность (жирность) шрифта. Возможные значения

- normal – обычный
- bold – жирный
- bolder – жирнее родительского
- lighter – светлее родительского

- 100 – самый светлый
- 200
- 300
- 400 – то же что normal
- 500
- 600
- 700 – то же что bold
- 800
- 900 – самый жирный

Установки жирности зависят от установленных на пользовательской машине шрифтов.

Часто пользователь не может увидеть различий между близкими значениями жирности.

Font-variant переключает шрифт между обычным (normal) и малыми заглавными буквами (small-caps).

```
p:first-line {font-variant: small-caps;}
```

Line-height задает межстрочный интервал в размерных единицах или процентах межстрочного интервала родительского объекта.

```
p {line-height: 8mm;}
```

Перечисленные выше атрибуты могут быть объединены в составной атрибут font в следующем порядке: font-style, font-variant, font-weight, font-size, line-height, font-family.

Пример:

```
p {
  font: italic normal 400 12px/14px Arial;
}
```

Эквивалентно

```
p {
  font-style: italic;
  font-variant: normal; /* значение по умолчанию можно опустить*/
  font-weight: 400; /* значение по умолчанию можно опустить*/
  font-size: 12px;
  line-height: 14px;
  font-family: Arial;
}
```

Значения **font-size** и **font-family** должны всегда присутствовать в объявлении **font** и быть расположены в установленном порядке. Если какое-либо из них будет пропущено, то все правило будет признано недействительным.

Практически примените рассмотренные свойства шрифта и текста к элементам **body** и **p** в созданном примере.

