

Знакомство с каскадными таблицами стилей

Окулова Виктория Викторовна
Учитель информатики

Знакомство с каскадными таблицами стилей

Оглавление

Занятие № 1. Основа CSS - селекторы	3
Занятие № 2. Оформление текста - работа со шрифтами	4
Занятие № 3. Оформление текста - работа с параметрами абзаца.....	5
Занятие № 4. Понятие блок, оформление блоков.	6
Занятие № 5. Создание каркаса сайта с помощью плавающих элементов.	7
Список использованных источников.....	8

Занятие № 1. Основа CSS - селекторы

- HTML – основа WEB-страницы, ее каркас, т.е. ЧТО должно быть на странице.
Стиль – это указание браузеру как отображать тот или иной элемент, т.е. КАК страница должна выглядеть.
Всё, что находится внутри тега <body>, браузер будет отображать в соответствии с описанными стилями.
- **Способы**
 - ✓ Вложение – прописывание в теге с помощью атрибута style
 - ✓ Встраивание – прописывание внутри тега Head с помощью элемента <style>
 - ✓ **Связывание** – размещение в отдельном файле style.css и прописывание внутри тега Head с помощью тега <link> (<link rel="stylesheet" type="text/css" href="style.css">)

Тег <link> - непарный. Атрибуты: rel="stylesheet" – подключение таблицы стилей, type="text/css" – тип подключаемого файла, href="style.css" – путь до таблицы стилей.

- **Основа** – селектор (имя тега, имя идентификатора, имя класса) и правило (записывается в {}).
Правило состоит из свойств и их значений, разделенных «;», свойство и значение разделяются «:».

```
Селектор { свойство1:значение;
           Свойство2:значение;
           .....
           свойствоN:значение;
        }
```

- **Селекторы**

Селекторы тегов	name{ }	позволяет назначить свойства всем тегам одного плана. name – имя тега
Селекторы идентификаторов	#name{ }	id-селектор позволяет изменить свойства одного элемента Самому элементу в html-документе надо дать имя id. name – имя идентификатора
Селекторы классов	.name{ }	позволяет изменять свойства выбранной группы элементов Элементам надо в html-документе дать имя класса. name – имя класса

Групповые селекторы используются для задания одинаковых правил для разных элементов.

Записываются селекторы через запятую, могут быть использованы все виды селекторов.

- **Задание**

1. Дан файл index.html с текстом, имеющим заголовок и разбитым на абзацы.
2. Написать файл style.css, в котором использовать
 - a. селекторы тегов (**name { }**) для всех абзацев и заголовка – установка цвета текста,
 - b. селекторы идентификаторов (**#name { }**) для нечетных абзацев – установка пунктирной рамки толщиной в 3 пикселя выбранного цвета,
 - c. селекторы классов (**.name { }**) для изменения свойств второго абзаца – установка рамки с сокращенной записью.
3. Использовать правила для цвета текста и рамок.
4. Связать файл index.html с файлом style.css.

- **Правила**

Цвет элемента	color:#1250FF;
Рамки (с 4 сторон)	border-width:2px; - ширина рамки border-style:solid dashed dotted и др.; - стиль рамки, solid – одинарная, dashed – пунктирная, dotted – в точку, double – двойная, groove – с тенью наружу, ridge – с тенью внутрь, inset – левая+верхняя, outset – правая+нижняя. Border-color:#235687; - цвет рамки
Сторона рамки	top bottom left right – приставки (например: border-top-width:2px); top – верхняя, bottom – нижняя, left – левая, right – правая
Сокращенная запись – толщина стиль цвет. Пример: border: 2px solid #235687;	

Занятие № 2. Оформление текста - работа со шрифтами

- **Правила**

Гарнитура Шрифта `font-family:font1, font2, type_font;`
 По умолчанию гарнитура шрифта – Times New Roman, font1 – название основного шрифта, font2 – название второго шрифта.
 type_font – тип шрифта, serif – с засечками, sans-serif – без засечек, monospace – моноширинный

Normal style	
Arial, Arial, Helvetica, <i>sans-serif</i>	Lucida Sans Unicode, Lucida Grande, <i>sans-serif</i>
Arial Black, Arial Black, Gadget, sans-serif	Palatino Linotype, Book Antiqua ³ , Palatino, <i>serif</i>
Comic Sans MS, <i>Comic Sans MS</i> ⁵ , <i>cursive</i>	Tahoma, Geneva, <i>sans-serif</i>
Courier New, <i>Courier New</i> , <i>monospace</i>	Times New Roman, <i>Times New Roman</i> , Times, <i>serif</i>
Georgia ¹ , <i>Georgia</i> , <i>serif</i>	Trebuchet MS ¹ , <i>Trebuchet MS</i> , <i>sans-serif</i>
Impact, Impact⁶, Charcoal⁶, sans-serif	Verdana, <i>Verdana</i> , Geneva, <i>sans-serif</i>
Lucida Console, <i>Monaco</i> ⁵ , <i>monospace</i>	

Размер шрифта `font-size:14px;`
 % - проценты (100%), em – единицы (1), px – пиксели (16), pt - пункты (1/72 дюйма) (12),
 В скобках указаны базовые размеры
 Абсолютный размер - xx-small, x-small, small, **medium**, large, x-large, xx-large,
 относительный размер – larger, smaller,

Курсив `font-style:italic;`
 Перед отдельными фрагментами прописывается тег логического уровня , которому назначают класс

Насыщенность `font-weight:normal | bold | 100-900`
 Normal – нормальная (значение 400), или bold - полужирный шрифт, или числами от 100 до 900 через 100

Декор `text-decoration:underline | overline | line-through | none`
 Underline – подчеркнутый, overline – надчеркнутый, line-through – перечеркнутый, none – снятие подчеркивания

Интервал между словами `word-spacing:12px;` (пиксели) (игнорируется, если установлено выравнивание по ширине)

Расстояние между буквами `letter-spacing:2px;`

Изменение регистра `text-transform:uppercase | uppercase | capitalize | capitel;`
 uppercase – заглавные буквы (верхний регистр), uppercase - строчные буквы (нижний регистр),
 capitalize - каждая первая буква заглавная, capitel – строчные буквы как малые заглавные.

- **Задание**

1. Дополнить файл style.css правилами для заголовка, устанавливающими гарнитуру шрифта из семейства sans-serif, размер шрифта 18 пикселей, полужирного начертания, с увеличенным расстоянием между буквами, каждая первая буква заглавная.
2. Дополнить файл style.css правилами для всех абзацев, устанавливающими гарнитуру шрифта из семейства serif, размер шрифта 14 пикселей, курсивного начертания, с увеличенным расстоянием между словами.
3. Дополнить файл style.css правилами для четных абзацев, устанавливающими декор – подчеркивание, изменение регистра на заглавные буквы

Занятие № 3. Оформление текста - работа с параметрами абзаца

- **Правила**

Выравнивание	<code>text-align:left center right justify;</code>
Красная строка	<code>text-indent:20px;</code> (можно в %, пикселях, единицах)
Межстрочный интервал	<code>line-height:12px normal;</code> Или в px, или множитель от текущего размера шрифта

- **Фоновые цвета и изображения**

Все блоки изначально прозрачные.

Фоновый цвет	<code>Background-color:#5621FF;</code> Фоновый цвет распространяется на содержимое и отступы. Рамка и внешние поля сами по себе, рамку раскрасить можно, поля – нет.
Контур для удобства верстки	<code>outline:1px solid #456879;</code>
Фон-изображение	<code>background-image:url();</code> url - путь относительный (относительно файла style.css или папки css) записывается в скобках
Повторение фона	<code>background-repeat:no-repeat repeat-x repeat-y repeat;</code>

- **Задание**

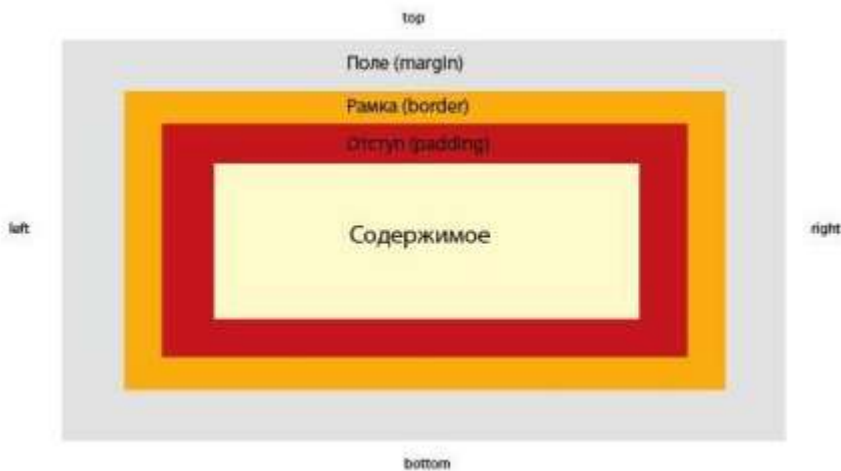
1. Дополнить файл style.css правилами для заголовка, устанавливающими выравнивание по центру, установить фоновый цвет.
2. Дополнить файл style.css правилами для всех абзацев, устанавливающими выравнивание по ширине, красную строку, межстрочный интервал с множителем 1.5.
3. Фоновое изображение скопировать в папку image.
4. Дополнить файл style.css правилом для тега <body>, устанавливающим фоновое изображение.

Занятие № 4. Понятие блок, оформление блоков.

• Блочные модели

Каждый тег для браузера (body – такой же блок, как и другие) – это блок.

Блочная схема: margin (поле), border (рамка), padding (отступ), content (содержимое).



Установка полей:

margin-top:20px; - верхнее поле, аналогично для остальных полей.

В сокращенной записи от top по часовой стрелке.

margin: 20px 10px 15px 5px;

Для выравнивания по центру –

margin: 20px auto 15px auto;

Для одинаковых противоположных полей –

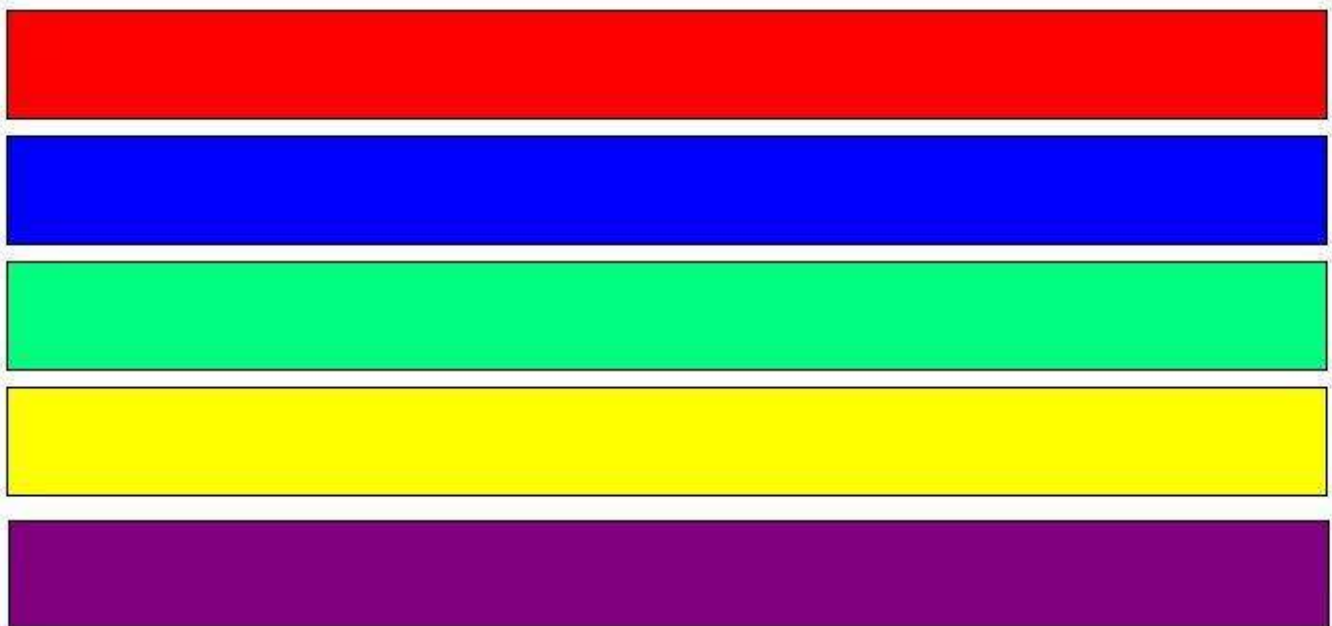
margin: 20px 10px;

Отступы (padding) устанавливаются аналогично.

width:400px; – ширина только для содержимого! height:200px; – высота содержимого. Если содержимое не помещается в устанавливаемых размерах, то overflow:hidden (спрятать)|auto (полосы прокрутки, если не помещается)|scroll (полосы прокрутки в любом случае)|visible;

• Задание

1. Создать 5 блоков <div> </div> в html-файле:
 - a. размер каждого блока по высоте 100px (пока нет заполнения, затем это свойство снимается и блок имеет высоту, занимаемую заполняющей его информацией), по ширине они займут отведенное им место
 - b. отступы (padding) до рамки 10px,
 - c. поле (margin) снизу 20px,
 - d. рамка 4px одинарная черного цвета, цвет заливки каждого блока подобрать самостоятельно.
2. Созданные блоки «обернуть» в блок <div> </div>, для которого задать ширину 800px, выравнивание по центру (auto).
3. Все свойства задать в присоединенном css-файле.



Занятие № 5. Создание каркаса сайта с помощью плавающих элементов.

Каркас с «шапкой», левом меню, содержимым и нижней частью страницы



Свойство плавающего элемента –
`float:left;` - выравнивание по левому краю
`float:right;` - выравнивание по правому краю
`clear:both;` - очистка обтекания по двум сторонам (слева и справа).
Для плавающего элемента обязательно надо указывать ширину (`width`) для корректного отображения разными браузерами

Для элемента (левое меню) указывается нужная ширина (например: 150px) и выравнивание по левому краю.

Для элемента (содержимое) указывается ширина, равная разности общей ширины с шириной левого меню и небольшого отступа между ними (например: $800 - 150 - 10 = 640$ px), а также левое поле, равное ширине левого меню + небольшой отступ (160px).

Перед блоком нижней части страницы надо вставить дополнительный блок, которому задать класс (в html-файле). В css-файле этому классу надо задать свойство очистки обтекания по двум сторонам.

Каркас с «шапкой», левом меню, правом меню, содержимым и нижней частью страницы



Для элемента (левое меню) указывается нужная ширина (например: 150px) и выравнивание по левому краю.

Для элемента (правое меню) указывается нужная ширина (например: 150px) и выравнивание по правому краю.

Для элемента (содержимое) указывается ширина, равная разности общей ширины с шириной левого и правого меню и небольшого отступа между ними (например: $800 - (150+150) - (10+10) = 480$ px), а также выравнивание по центру.

Перед блоком нижней части страницы надо вставить дополнительный блок, которому задать класс (в html-файле). В css-файле этому классу надо задать свойство очистки обтекания по двум сторонам.

Список использованных источников

<http://ab-w.net/> - самоучитель по сайтостроению

<http://evgenyropov.com/> - видеоуроки по HTML и CSS