

Тема 10: Блочная модель документа

Все сайты можно разбить на блоки, которые в свою очередь могут состоять из блоков поменьше. Каждый блок подразумевает под собой начало с новой строки и при закрытии переход на новую строку. Вся блочная структура делается в файле HTML, а затем при помощи CSS определяется, как блоки будут отображаться, какие у них будут размеры и т.д. Почти каждый тег из себя изначально представляет блок, но по стандартам W3C есть определенные теги для блочной структуры, например вы уже ранее встречали тег `<div>`, который является универсальным блочным элементом и предназначен для группирования элементов документа с целью изменения вида содержимого через стили. Для этого добавляется атрибут `class` (подразумевает под собой объединение элементов со схожими свойствами) или `id` (назначается элементу с уникальными свойствами) с именем класса или идентификатора соответственно.

Но есть теги для блоков стандарта HTML5:

`<header>` - задаёт «шапку» сайта или раздела, в которой обычно располагается заголовок.

`<nav>` - задаёт навигацию по сайту. Если на странице несколько блоков ссылок, то в `<nav>` обычно помещают приоритетные ссылки. Также допустимо использовать несколько `<nav>` в документе.

`<main>` - предназначен для основного содержимого документа. На странице может быть только один `<main>` и он не должен располагаться внутри элементов `<article>`, `<aside>`, `<footer>`, `<header>` или `<nav>`.

`<section>` - задаёт раздел документа, может применяться для блока новостей, контактной информации, глав текста, вкладок в диалоговом окне и др. Раздел обычно содержит заголовок. Допускается вкладывать один тег `<section>` внутри другого.

`<article>` - задаёт содержание сайта вроде новости, статьи, записи блога, форума и др.

`<aside>` - представляет собой раздел страницы, который имеет косвенное отношение к содержимому страницы и может быть рассмотрен отдельно от этого содержимого. `<aside>` применяется для боковых панелей, рекламных блоков, ссылок на архив, меток и другой информации, которая отделена от основного содержимого страницы.

`<footer>` - задаёт «подвал» сайта или раздела, в нём может располагаться имя автора, дата документа, контактная и правовая информация.

Практическая работа №10.1

Давайте напишем структуру сайта, с которой в дальнейшем мы будем работать.

```
<!DOCTYPE html>
<html>
<head>
  <title>Мини сайт</title>
  <meta charset="utf-8">
  <link rel="stylesheet" type="text/css" href="css/style.css">
</head>
<body>
  <header>
    
    <h1>Ракета-ЗИП140</h1>
    <nav>
      <a href="#">Главная</a>
      <a href="#">Отзывы</a>
      <a href="#">Контакты</a>
    </nav>
  </header>
  <main>
    <section>
      <article>
        
      </article>
    </section>
    <section>
      <article>
        <h2>Предложение 1</h2>
        <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
        tempor incididunt ut labore et dolore magna aliqua. Quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo
        consequat.</p>
      </article>
      <article>
        <h2>Предложение 2</h2>
        <p>Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo
        consequat.</p>
      </article>
      <article>
        <h2>Предложение 3</h2>
        <p>Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod
        tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam,
        quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo
        consequat.</p>
      </article>
    </section>
  </main>
  <footer>
    <nav>
      <a href="#">Главная</a>
      <a href="#">Отзывы</a>
      <a href="#">Контакты</a>
    </nav>
    
  </footer>
</body>
</html>
```

Изображения могут быть любые, главное, чтобы логотип был в формате png, а остальные изображения в jpg(jpeg).

Также вы можете сделать верстку на div блоках с применением class и id.

Теперь давайте рассмотрим 3 свойства CSS, которые очень важны для блочной модели документа:

display - многоцелевое свойство, которое определяет, как элемент должен быть показан в документе (значение по умолчанию - *inline*).

Свойство может принимать значения:

block - элемент показывается как блочный. Применение этого значения для строчных элементов, например ``, заставляет его вести подобно блокам — происходит перенос строк в начале и в конце содержимого.

inline — элемент отображается как строчный. Использование блочных элементов, таких как `<div>` и `<p>`, автоматически создаёт перенос и показывает их содержимое с новой строки. Значение *inline* отменяет эту особенность, поэтому содержимое блочных элементов начинается с того места, где окончился предыдущий элемент.

inline-block - это значение генерирует блочный элемент, который обтекается другими элементами веб-страницы подобно строчному элементу. Фактически такой элемент по своему действию похож на встраиваемые элементы (вроде ``). При этом его внутренняя часть форматируется как блочный элемент, а сам элемент — как строчный.

inline-flex - элемент ведёт себя как строчный и выкладывает содержимое согласно флекс-модели.

flex — элемент ведёт себя как блочный и выкладывает содержимое согласно флекс-модели.

list-item - элемент выводится как блочный и добавляется маркер списка.

none - временно удаляет элемент из документа. Занимаемое им место не резервируется и веб-страница формируется так, словно элемента и не было. Изменить значение и сделать вновь видимым элемент можно с помощью скриптов, обращаясь к свойствам через объектную модель. В этом случае

происходит переформатирование данных на странице с учётом вновь добавленного элемента.

run-in - устанавливает элемент как блочный или строчный в зависимости от контекста.

Второе свойство, которое нам необходимо изучить:

position - устанавливает способ позиционирования элемента относительно окна браузера или других объектов на веб-странице (значение по умолчанию - *static*).

Свойства принимает значения:

absolute - абсолютное позиционирование. Указывает, что элемент абсолютно позиционирован, при этом другие элементы отображаются на веб-странице словно абсолютно позиционированного элемента и нет. Положение элемента задаётся свойствами *left*, *top*, *right* и *bottom*, также на положение влияет значение свойства *position* родительского элемента. Так, если у родителя значение *position* установлено как *static* или родителя нет, то отсчёт координат ведётся от края окна браузера. Если у родителя значение *position* задано как *relative*, то отсчёт координат ведётся от края родительского элемента.

fixed - фиксированное позиционирование. По своему действию это значение близко к *absolute*, но в отличие от него привязывается к указанной свойствами *left*, *top*, *right* и *bottom* точке на экране и не меняет своего положения при прокрутке веб-страницы.

relative - относительное позиционирование. Положение элемента устанавливается относительно его исходного места. Добавление свойств *left*, *top*, *right* и *bottom* изменяет позицию элемента и сдвигает его в ту или иную сторону от первоначального расположения.

static - статичное позиционирование. Элементы отображаются как обычно. Использование свойств *left*, *top*, *right* и *bottom* не приводит к каким-либо результатам.

sticky - это сочетание относительного и фиксированного позиционирования. Элемент рассматривается как позиционированный относительно, пока он не пересекает определённый порог, после чего рассматривается как фиксированный. Обычно применяется для фиксации заголовка на одном месте, пока содержимое, к которому относится заголовок, прокручивается на странице.

Третье свойство *float* определяет, по какой стороне будет выравниваться элемент, при этом остальные элементы будут обтекать его с других сторон. Когда значение свойства *float* равно *none*, элемент выводится на странице как обычно, при этом допускается, что одна строка обтекающего текста может быть на той же линии, что и сам элемент (значение по умолчанию - *none*).

Принимает всего три значения:

left - выравнивает элемент по левому краю, а все остальные элементы, вроде текста, обтекают его по правой стороне.

right - выравнивает элемент по правому краю, а все остальные элементы обтекают его по левой стороне.

none - обтекание элемента не задаётся.

Практическая работа №10.2

Теперь попробуем провести первые манипуляции с ранее созданной версткой.

```
/* Обнуляем все отступы*/
* {
    margin: 0;
    padding: 0;
}
/* Назначаем шрифты, ширину, отображение на середине и фон страницы */
body {
    font-family: Helvetica, Tahoma, Arial;
    width: 80em;
    margin: 0 auto;
    background: url(img/main_bg.png) no-repeat grey fixed;
    background-size: cover;
}
header {
    background-color: rgba(0, 0, 255, .3);
}
/* Логотип */
header > img {
    display: inline-block;
    width: 10em;
    float: left;
}
/* Название организации */
header > h1 {
    display: inline-block;
    font-size: 7em;
    float: right;
}
/* Меню */
header > nav {
    width: 80em;
}
```

```

/* Пункты меню */
header > nav > a {
    text-decoration: none;
    font-size: 3em;
    color: black;
}
/* Посещенные ссылки */
header > nav > a:visited {
    color: black;
}
/* Контентная область */
main {
    background-color: rgba(0, 255, 0, .3);
}
main > section:first-child > article {
    position: relative;
}
/* Изображение слайдера */
main > section:first-child > article > img {
    width: 80em;
}
/* Псевдоэлемент для левой стрелки слайдера */
main > section:first-child > article::before {
    content: '<';
    position: absolute;
    color: black;
    background-color: rgba(255, 255, 255, .7);
    font-size: 5em;
    border-top-right-radius: 15px;
    border-bottom-right-radius: 15px;
    cursor: pointer;
}
/* Псевдоэлемент для правой стрелки слайдера */
main > section:first-child > article::after {
    content: '>';
    position: absolute;
    color: black;
    background-color: rgba(255, 255, 255, .7);
    font-size: 5em;
    border-top-left-radius: 15px;
    border-bottom-left-radius: 15px;
    cursor: pointer;
}
/* Блок предложений */
main > section:nth-child(2) {
    background-color: rgba(255, 255, 0, .3);
}
main > section:nth-child(2) > article {
    display: inline-block;
    width: 20em;
}

/* Подвал сайта */
footer {
    background-color: rgba(255, 0, 0, .3);
}
/* Меню подвала */
footer > nav {
    display: inline-block;
}
/* Пункты меню подвала */
footer > nav > a {
    text-decoration: none;
    font-size: 3em;
    color: black;
}
footer > nav > a:visited {
    color: black;
}
/* Логотип подвала */
footer > img {
    display: inline-block;
    width: 10em;
}

```


Проверка знаний

Переделайте структуру сайта на div блоки с использованием class и id, также переделайте селекторы в CSS в соответствии с вашей новой структурой.

