# МАТЕРИАЛ

## HTML

Давайте закрепим полученные знания на примере, и сделаем простую разметку для сайта, используя современный стандарт гипертекстовой разметки HTML 5.

Хочу сразу оговориться, что в этом примере будут использованы CSS свойства, которые необходимы для большей наглядности. Не переживайте если вы сейчас не поймете работу этих свойств, так как нас сейчас больше интересует правильная разметка документа, а подробное изучение CSS будет рассмотрено позже.

### Практика

Начнем с основного “каркаса” документа и потихонечку будем продвигаться к цели. Создайте новый html-файл, назовите его layout.html и добавьте базовую структуру страницы:

| <!DOCTYPE html> <html>  <head>  <title>Семантическая Разметка</title>  </head>  <body>    </body> </html> |
| --- |

Отлично, основа есть, давайте двигаться дальше. Добавим внутрь раздела BODY несколько элементов и подумаем зачем мы их сюда добавили:

| <body>  <header>HEADER</header>  <nav>NAV</nav>  <div id="page"></div>  <footer>FOOTER</footer> </body> |
| --- |

Один элемент здесь кажется лишним, но это не так. DIV нам нужен, чтобы оформить нашу страничку более выразительно. Используя его мы сможем применить к элементам внутри него некоторые стили.

Давайте посмотрим, что же будет внутри этого элемента с id=”page”:

| <div id="page">  <main>MAIN</main>  <aside>ASIDE</aside> </div> |
| --- |

Да, да это теги MAIN и ASIDE. Мы ведь хотим увидеть в результате не надписи, безобразно собранные в кучу, а четкую структуру.. мы должны почувствовать связь того, что написали в html-файле и того, что видем в окошке браузера и именно ради этой связи мы с вами и продолжим. Вот так нам нужно подправить в MAIN:

| <main>  MAIN  <article>ARTICLE</article> </main> |
| --- |

Main - это у нас основной контент, то содержимое страницы, которое представляет наибольшую важность и в данном случае наиболее важным у нас будет Статья, содержащая завершенную, независимую композицию.

Но не будет же у нас пустая статья, она же должна быть содержательной:

| <article>  ARTICLE  <header>HEADER</header>  <section>SECTION</section>  <section>SECTION</section>  <footer>FOOTER</footer> </article> |
| --- |

Так вот, наша статья будет состоять из нескольких частей: заголовок, две раздела группирующих связанное содержимое и подвал содержащий различные сноски и т.д.

Вот теперь статья выглядит более содержательно.

Перейдем к следующему тегу: ASIDE. Его мы тоже пустым не оставим. Давайте добавим туда пору элементов - HEADER и NAV. Должно получится вот так:

| <aside>  ASIDE  <header>HEADER</header>  <nav>NAV</nav> </aside> |
| --- |

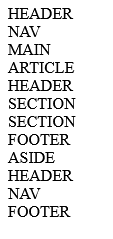
Здесь мы разместим некоторый контент отступающий от основного содержимого (MAIN),

Добавим сюда заголовок, указывающий на содержимое следующего за ним элемента, в данном случае - NAV. А вот он у нас, очевидно, должен содержать некоторые ссылки.

Ну вот и результат, мы закончили нашу структуру. Давайте посмотрим на код:

| <body>  <header>HEADER</header>  <nav>NAV</nav>  <div id="page">  <main>  MAIN  <article>  ARTICLE  <header>HEADER</header>  <section>SECTION</section>  <section>SECTION</section>  <footer>FOOTER</footer>  </article>  </main>  <aside>  ASIDE  <header>HEADER</header>  <nav>NAV</nav>  </aside>  </div>  <footer>FOOTER</footer> </body> |
| --- |

Выглядит отлично! Каждый элемент понятен и находится на своем месте. Теперь определенно стоит взглянуть на результат который нам покажет браузер:



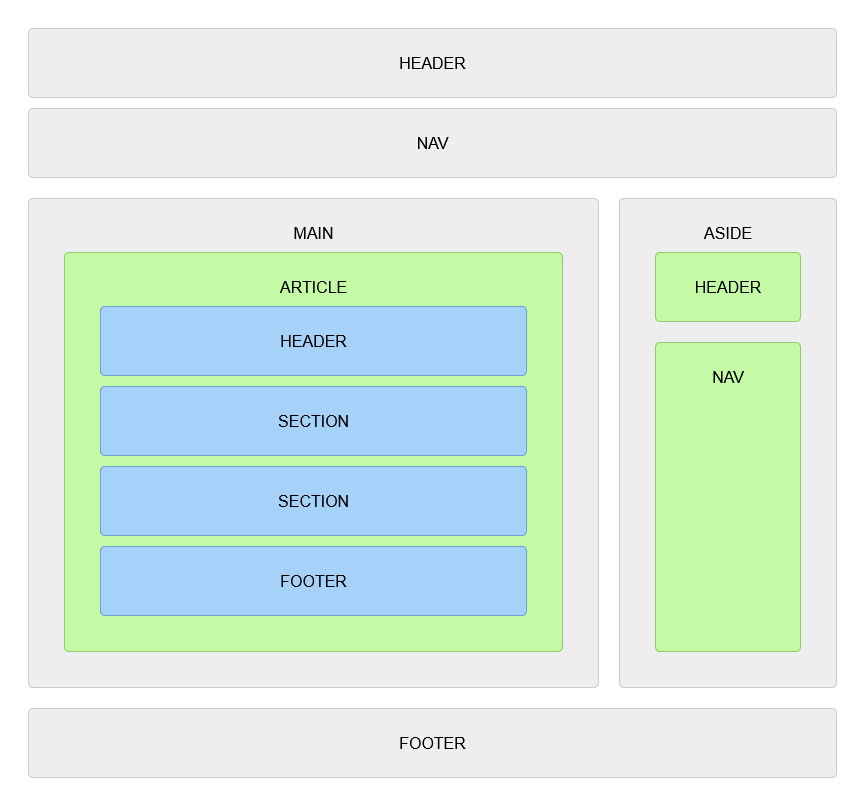
Хм… кажется это выглядит не так, как я описывал. Здесь определенно чего-то не хватает и мы сейчас с этим разберемся.

Добавим к нашей странички немножечко магии… CSS. Добавлять будем в самый низ раздела HEAD. Вот так:

| <head>  <title>Семантическая Разметка</title>  <style>  body {  font-family: sans-serif;  padding: 10px;  }  body \*:not(#page) {  padding: 25px;  margin: 10px;  background-color: #eee;  border: 1px solid #ccc;  border-radius: 5px;  text-align: center;  }  #page {  display: flex;  flex-direction: row;  flex-wrap: wrap;  }  #page > main {  flex: 1;  }  #page > aside {  display: flex;  flex-direction: column;  flex: 1;  max-width: 20%;  }  #page > aside > \* {  background-color: #c4f9a6;  border-color: #91ca71;  }  #page > aside > nav {  flex: 1;  }  #page > main > article {  background-color: #c4f9a6;  border-color: #91ca71;  }  #page > main > article > \* {  background-color: #a6d2f9;  border-color: #6ca1d0;  }  </style> </head> |
| --- |

Много здесь всего понаписано, конечно, но, результат стоит того. К тому же, мы скоро начнем знакомиться с CSS и его магическими возможностями.

Ну что, пора смотреть результат:



Совсем другое дело, вот теперь можно почувствовать связь HTML-кода с результатом в браузере.